



Red Hat Certificate System 9

Enterprise Security Client を使用したスマート カードの管理

Red Hat Certificate System 9.7 向けに更新

Red Hat Certificate System 9 Enterprise Security Client を使用したスマートカードの管理

Red Hat Certificate System 9.7 向けに更新

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

法律上の通知

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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.

概要

本ガイドは、Certificate System サブシステムの通常のユーザーを対象としています。また、スマートカードのフォーマットおよび管理用のシンプルなインターフェイスである Enterprise Security Client を使用して、個人証明書および鍵を管理する方法を説明します。

目次

| | |
|--|-----------|
| 第1章 ENTERPRISE SECURITY CLIENT の概要 | 3 |
| 1.1. RED HAT ENTERPRISE LINUX、シングルサインオン、および認証 | 3 |
| 1.2. RED HAT CERTIFICATE SYSTEM および ENTERPRISE SECURITY CLIENT | 4 |
| 第2章 ENTERPRISE SECURITY CLIENT のインストール | 6 |
| 2.1. クライアントでサポートされるプラットフォーム | 6 |
| 2.2. 対応するスマートカード | 6 |
| 2.3. RED HAT ENTERPRISE LINUX での ENTERPRISE SECURITY CLIENT のインストールおよびアンインストール | 6 |
| 第3章 ENTERPRISE SECURITY CLIENT の使用 | 8 |
| 3.1. ENTERPRISE SECURITY CLIENT のトレイアイコン | 8 |
| 3.2. ENTERPRISE SECURITY CLIENT の起動 | 8 |
| 3.3. PHONE HOME の設定 | 9 |
| 3.4. 登録するユーザーの設定 | 12 |
| 3.5. スマートカードの管理 | 13 |
| 3.6. 問題の診断 | 22 |
| 第4章 WEB およびメールクライアントでのスマートカードの使用 | 28 |
| 4.1. トークンに対して SSL をサポートするブラウザの設定 | 28 |
| 第5章 ENTERPRISE SECURITY CLIENT の設定 | 30 |
| 5.1. ENTERPRISE SECURITY CLIENT 設定の概要 | 30 |
| 5.2. TPS を使用した SSL 接続の設定 | 33 |
| 5.3. 共有セキュリティーデータベースの使用 | 36 |
| 5.4. トークン操作の LDAP 認証の無効化 | 37 |
| 付録A 更新履歴 | 38 |

第1章 ENTERPRISE SECURITY CLIENT の概要

Enterprise Security Client は、スマートカードの管理を簡素化する Red Hat Certificate System のツールです。エンドユーザーは、セキュリティトークン (スマートカード) を使用して、シングルサインオン (SSO) アクセスやクライアント認証などのアプリケーションのユーザー証明書を保存できます。エンドユーザーには、署名、暗号化、およびその他の暗号化機能に必要な証明書および鍵が含まれるトークンが発行されます。

Enterprise Security Client は、Certificate System の完全なトークン管理システムの 3 番目の部分です。2 つのサブシステム (Token Key Service (TKS) および Token Processing System (TPS)) は、トークン関連の操作を処理するために使用されます。Enterprise Security Client は、スマートカードとユーザーがトークン管理システムにアクセスできるようにするインターフェイスです。

トークンの登録後、Mozilla Firefox や Thunderbird などのアプリケーションは、トークンを認識して、クライアント認証や S/MIME メールなどのセキュリティ操作に使用するように設定できます。Enterprise Security Client は、以下の機能を提供します。

- Gemalto 64K V2 や Safenet 300J Java スマートカードなどの Global Platform 準拠のスマートカードに対応します。
- セキュリティトークンを登録して、TPS で認識されるようにします。
- TPS でトークンを再登録するなど、セキュリティトークンを維持します。
- 管理対象トークンの現在のステータスに関する情報を提供します。
- トークンが失われた場合に別のトークンで鍵をアーカイブおよび復元できるように、TPS および DRM サブシステムによるサーバー側の鍵生成をサポートします。

1.1. RED HAT ENTERPRISE LINUX、シングルサインオン、および認証

ネットワークユーザーは、使用する各種サービスに複数のパスワードを送信する必要があります。たとえば、電子メール、Web 閲覧、組織のサーバー、およびネットワーク上のサーバーなどです。複数のパスワードを維持し、それらを入力し続けることは、ユーザーおよび管理者にとって困難です。**シングルサインオン** は、管理者が単一のパスワードストアを作成してユーザーが一度ログインし、単一のパスワードを使用してすべてのネットワークリソースに認証できるようにするための設定です。

Red Hat Enterprise Linux は、ワークステーションへのログイン、スクリーンセーバーのロック解除、Mozilla Firefox を使用して暗号化された Web ページへのアクセス、Mozilla Thunderbird を使用した暗号化された電子メールの送信など、複数のリソースのシングルサインオンをサポートします。

シングルサインオンは、ユーザーにとって便利であると同時に、サーバーおよびネットワークのセキュリティにおけるもう 1 つの層でもあります。シングルサインオンは、セキュアで効果的な認証を受け、Enterprise Security Client は Red Hat Certificate System が実装する公開鍵インフラストラクチャーに関連付けられます。

セキュアなネットワーク環境を確立するための基盤の 1 つは、ネットワークへのアクセス権限を持つユーザーにアクセスが制限されるようにすることです。アクセスが許可されると、ユーザーはシステムに対して **認証** できます。つまり、ユーザーはアイデンティティを検証できます。このような方法の 1 つとして **証明書** を表示することです。証明書が存在するエンティティを特定する電子ドキュメントが挙げられます。

これらの証明書はスマートカードに保存できます。ユーザーが挿入すると、スマートカードは証明書をシステムに提示し、ユーザーを識別して認証できるようにします。Red Hat Enterprise Linux のシングルサインオンに対する 2 つの認証方法の 1 つはスマートカード認証です。もう 1 つは Kerberos ベースの認証です。

スマートカードを使用したシングルサインオンには、以下の 3 つの手順があります。

1. ユーザーがスマートカードをカードリーダーに挿入します。これは、Red Hat Enterprise Linux のプラグ可能な認証モジュール (PAM) により検出されます。
2. システムは、証明書をユーザーエントリーにマッピングし、スマートカードで提示された証明書をユーザーエントリーに保存されている証明書と比較します。
3. 証明書がキー配布センター (KDC) に対して正常に確認されると、ユーザーはログインを許可されます。

Enterprise Security Client は、シングルサインオンの管理に含まれるスマートカードを管理します。

1.2. RED HAT CERTIFICATE SYSTEM および ENTERPRISE SECURITY CLIENT

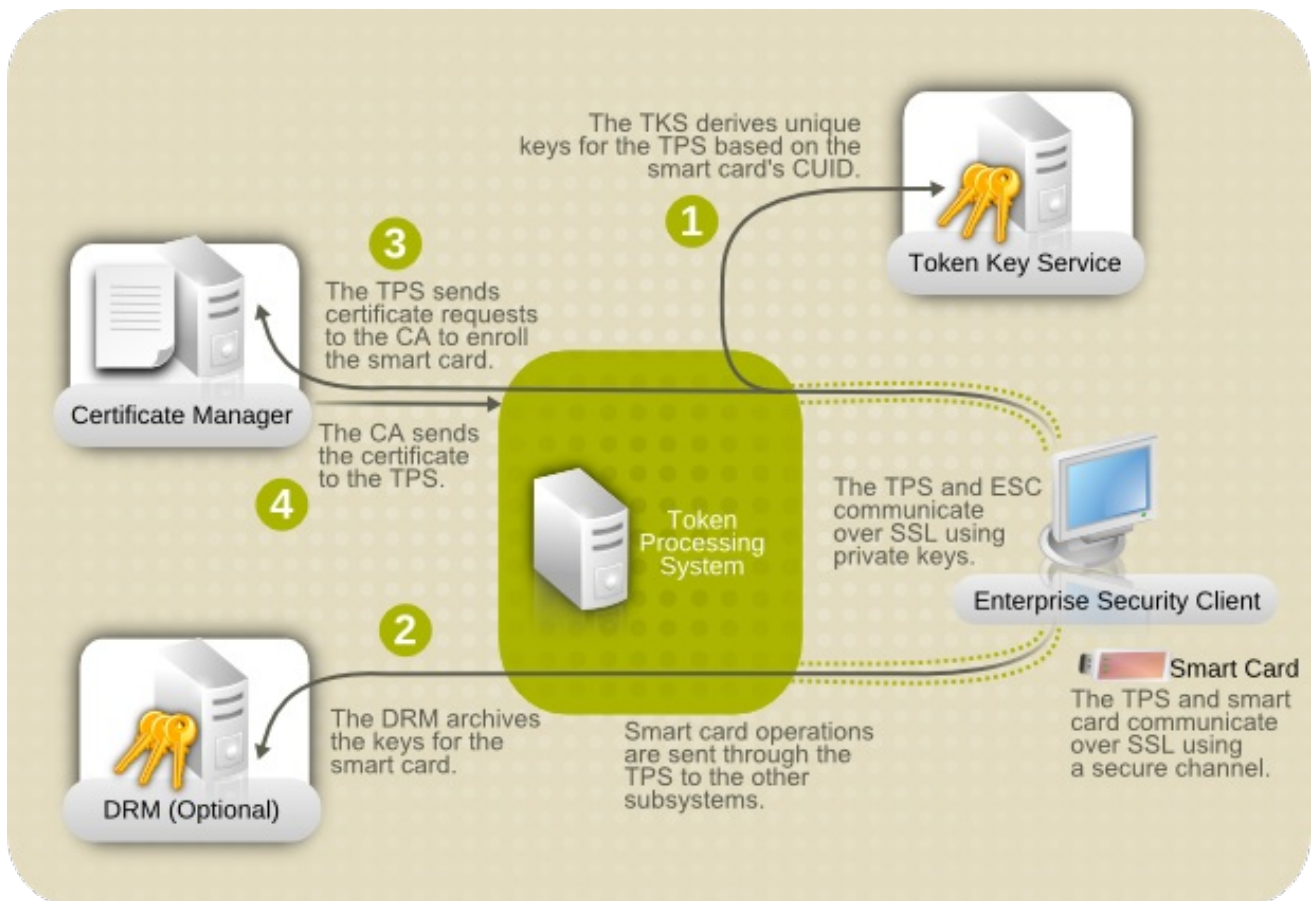
Red Hat Certificate System は、証明書および鍵の作成、管理、更新、取り消しを行います。Certificate System には、スマートカードを管理するために、キーの生成、証明書リクエストの作成、および証明書受け取りを行うトークン管理システムがあります。

2 つのサブシステム (Token Key Service (TKS) および Token Processing System (TPS)) は、トークン関連の操作を処理するために使用されます。Enterprise Security Client は、スマートカードとユーザーがトークン管理システムにアクセスできるようにするインターフェイスです。

4 つの Certificate System サブシステムの合計はトークンの管理に関与します。2 つはトークンの管理 (TKS および TPS)、もう 2 つは公開鍵インフラストラクチャー (CA および DRM) 内の鍵と証明書の管理に使用します。

- Token Processing System (TPS) はスマートカードと対話し、ユーザーやデバイスなどの特定のエンティティのキーと証明書を生成および保存できるようにします。スマートカード操作は TPS を経由して、証明書を生成する認証局やデータリカバリーマネージャーなどのアクションのために適切なサブシステムに転送され、キーをアーカイブおよび回復します。
- Token Key Service (TKS) は、TPS とスマートカード間の通信に使用される対称鍵を生成または取得します。TKS によって生成された鍵のセットは、カードの一意の ID を基にしているため一意です。鍵はスマートカードでフォーマットされ、スマートカードと TPS との間で通信の暗号化、または認証を行うために使用されます。
- 認証局 (CA) は、スマートカードに保存されているユーザー証明書を作成して破棄します。
- 必要に応じて、Data Recovery Manager (DRM) は、スマートカードのキーをアーカイブおよび復元します。

図1.1 Certificate System のスマートカードの管理方法



TPS は、[図1.1「Certificate System のスマートカードの管理方法」](#) に示すように、Red Hat Certificate System トークン管理システムを中心となるハブです。トークンは TPS と直接通信します。その後、TPS は TKS と通信して、TPS のトークン通信 (1) に使用できる一意の鍵のセットを取得します。スマートカードが登録されると、トークンに新しい秘密鍵が作成されます。キーのアーカイブを設定する場合は、これらのキーを DRM(2) でアーカイブできます。CA は証明書要求 (3) を処理し、トークンに保存する証明書を発行します。TPS は、これらの証明書を Enterprise Security Client (4) に戻します。これらはトークンに保存されます。

Enterprise Security Client は、TPS が安全な HTTP チャンネル (HTTPS) で各トークンと通信し、Certificate System とともに TPS を介して通信するパイプになります。

トークンを使用するには、Token Processing System がトークンを認識して通信できるようにする必要があります。必要な鍵と証明書でトークンを設定し、Certificate System にトークンを追加するために、トークンを最初に **登録** する必要があります。Enterprise Security Client は、トークンを登録するエンドエンティティーのユーザーインターフェイスを提供します。

第2章 ENTERPRISE SECURITY CLIENT のインストール

2.1. クライアントでサポートされるプラットフォーム

Enterprise Security Client インターフェイスは、Red Hat Enterprise Linux 7.3 以降のプラットフォームでサポートされています。

また、Red Hat Enterprise Linux 5 および 6 の最新バージョンでもサポートされます。これらのプラットフォームは Red Hat Certificate System 9 をサポートしませんが、これらのクライアントは Red Hat Certificate System 9 の TMS システムで使用できます。

2.2. 対応するスマートカード

詳細は、[Red Hat Certificate System 9 リリースノート](#) で該当するセクションを参照してください。

2.3. RED HAT ENTERPRISE LINUX での ENTERPRISE SECURITY CLIENT のインストールおよびアンインストール

2.3.1. ESC クライアントのインストール

Enterprise Security Client のインストールの最初の手順は、必要なパッケージをダウンロードすることです。パッケージを取得する方法は 2 つあります。

- カスタマーポータルから ISO イメージをダウンロードします。
- Red Hat **yum** ユーティリティーの使用

RPM を取得する方法として、以下のように **yum** コマンドラインユーティリティーを使用することが推奨されます。

```
# yum install esc
```

yum コマンドが正常に完了すると、必要な Enterprise Security Client RPM と依存関係がすべてインストールされ、使用できるようになります。



注記

yum ユーティリティーを使用して Enterprise Security Client をインストールした場合は、これ以上インストールする必要はありません。クライアントはすでにインストールされています。以下の手順では、CD イメージからインストールします。

1. **root** ユーザーとして、Enterprise Security Client パッケージをインストールします。

```
# yum install esc
```

Enterprise Security Client は、Red Hat Enterprise Linux 32 ビットシステムの **/usr/lib/esc-1.1.0** にあり、Red Hat Enterprise Linux 64 ビットシステムの **/usr/lib64/esc-1.1.0** にあります。**esc** シェルスクリプトが **/usr/bin/esc** にインストールされます。**esc** コマンドを実行すると、Enterprise Security Client を起動できます。

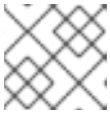
Enterprise Security Client for Linux は、スマートカードが挿入されるのを待機するデーモン(**escd**)を実

装します。登録されていないスマートカードが挿入されると、デーモンは自動的にクライアント UI を起動し、Enterprise Security Client ガイドに従って登録プロセスを進めます。また、**System Settings** を選択して **Smart Card Manager** を設定して、**System** メニューからクライアントを手動で起動することもできます。

2.3.2. ESC クライアントのアンインストール

1. すべての USB トークンを取り外します。
2. Enterprise Security Client の停止
3. **root** ユーザーとしてログインし、**rpm -ev** を使用して Enterprise Security Client RPM を削除します。

```
# yum remove esc
```



注記

お使いのバージョンに一致する RPM ファイルのバージョン番号を更新します。

4. インストールディレクトリーの残りのファイルを削除します。

第3章 ENTERPRISE SECURITY CLIENT の使用

以下のセクションでは、Enterprise Security Client を使用したトークンの登録、フォーマット、およびパスワードのリセットの操作に関する基本的な手順を説明します。

3.1. ENTERPRISE SECURITY CLIENT のトレイアイコン

多くのプログラムは、トレイまたは通知エリアでアイコンを維持し、これを使用してプログラムの動作を制御することができます。通常は、アイコンを右に移動したときにコンテキストメニューで行います。Enterprise Security Client は、スマートカードの挿入や削除などのエラーやアクションのツールチップなど、トレイアイコンを提供します。

図3.1 トークントレイアイコンおよびツールチップの例



デフォルト設定では、Enterprise Security Client が起動し、自動的にトレイに最小限に抑えられます。Red Hat Enterprise Linux では、Gnome の通知領域が有効になっている場合に限り、トレイアイコンが表示されます。

3.2. ENTERPRISE SECURITY CLIENT の起動

Enterprise Security Client の起動には、以下の 2 つの概念があります。Enterprise Security Client プロセスを開始して、挿入されたスマートカードまたはトークンを検出できるよう、通知なく実行される必要があります。スマートカードが挿入されたり、または手動で開くことができると、Enterprise Security Client のユーザーインターフェイスが自動的に開きます。

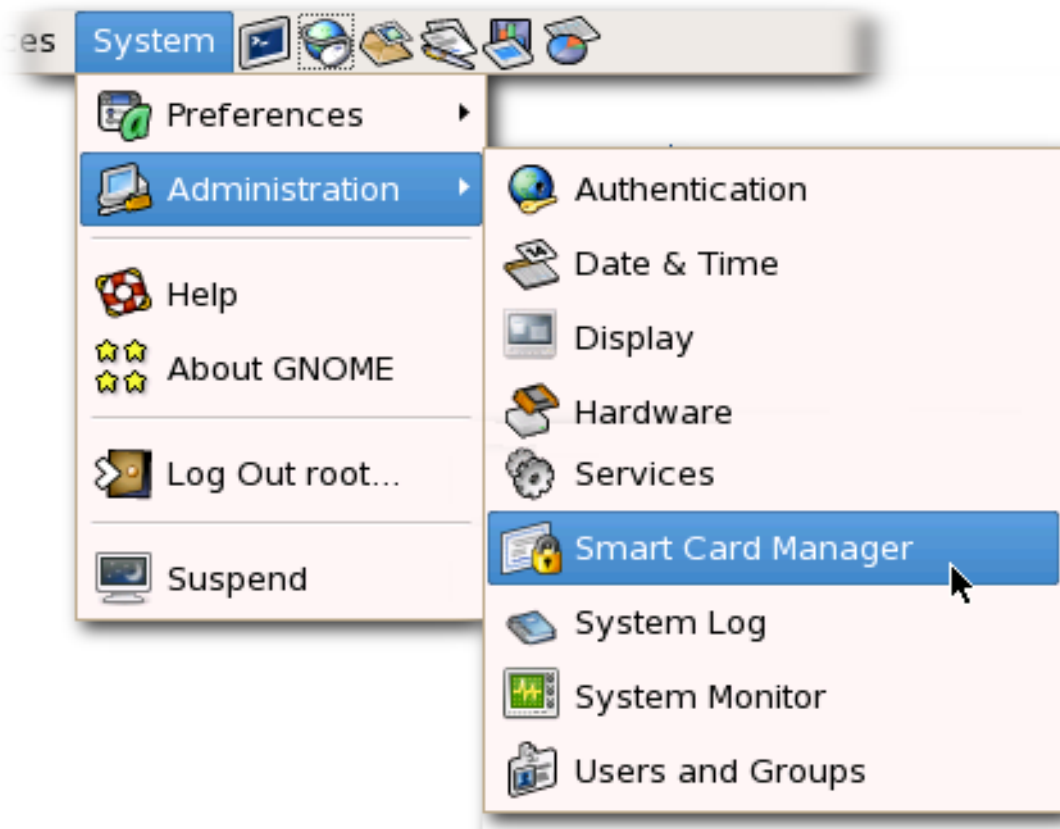
3.2.1. Red Hat Enterprise Linux で Enterprise Security Client を開く

コマンドラインから Enterprise Security Client デーモン(**escd**)を起動します。

```
esc
```

このデーモンはスマートカードを暗黙的にリッスンし、スマートカードが挿入されるとすぐに GUI を開きます。

Enterprise Security Client GUI を手動で開くには、**Applications**、**System Settings**、**Smart Card Manager** の順にクリックします。



3.3. PHONE HOME の設定

Enterprise Security Client の **Phone Home** 機能は、各スマートカード内の情報を、固有の TPS サーバーおよび Enterprise Security Client UI ページを示す情報に関連付けます。Enterprise Security Client が新しいスマートカードにアクセスするたびに、TPS インスタンスに接続して、Phone Home 情報を取得できます。

Phone Home はこの情報を取得してキャッシュします。この情報はローカルでキャッシュされているため、フォーマットされたスマートカードが挿入されるたびに TPS サブシステムと通信する必要はありません。

この情報はキーまたはトークンごとに異なる場合があります。つまり、異なる TPS サーバーおよび登録 URL を企業やカスタマーグループごとに設定できます。Phone Home を使用すると、Enterprise Security Client を手動で設定して正しいサーバーと URL を特定することなく、発行者や会社単位で異なる TPS サーバーを設定できます。

注記

TPS サブシステムが Phone Home 機能を使用できるようにするには、以下のように TPS 設定ファイルで Phone Home を有効にする必要があります。

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

3.3.1. Phone Home プロファイルについて

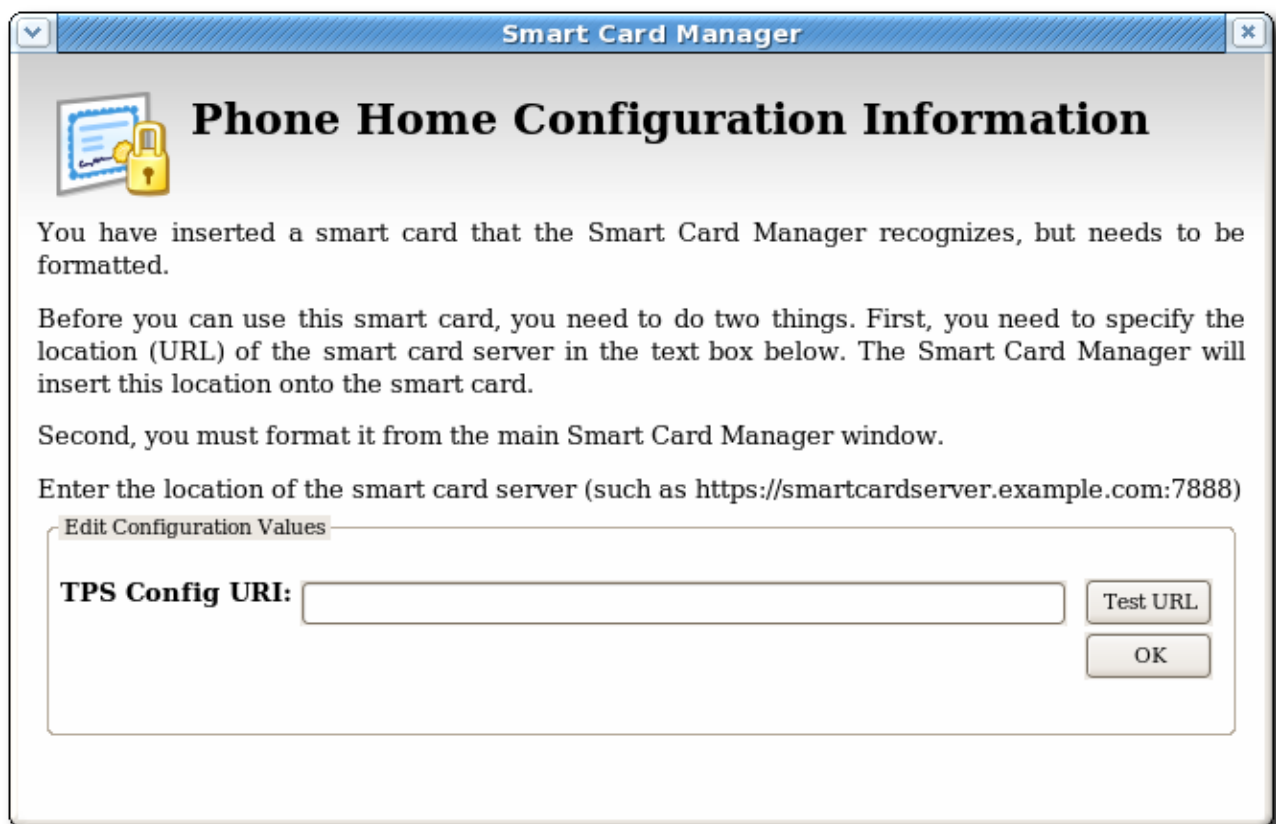
Enterprise Security Client は Mozilla XULRunner に基づいています。したがって、各ユーザーには、Mozilla Firefox および Thunderbird で使用されるユーザープロファイルと同様のプロファイルがあります。Enterprise Security Client は、設定ファイルにアクセスします。Enterprise Security Client が各

トークンの情報をキャッシュすると、情報はユーザーの設定ファイルに保存されます。次のエンタープライズセキュリティクライアントの起動時に、サーバーを再び接続する代わりに、設定ファイルから情報を取得します。

スマートカードが挿入され、Phone Home を起動すると、Enterprise Security Client は最初にトークンで Phone Home の情報をチェックします。トークンに関する情報がない場合、クライアントは **esc-prefs.js** ファイルで **esc.global.phone.home.url** パラメーターをチェックします。

トークンに Phone Home 情報が保存されておらず、グローバルな Phone Home パラメーターがない場合、[図3.2「Phone Home 情報のプロンプト」](#) に示すように、スマートカードが挿入されると、ユーザーは Phone Home URL の入力を求められます。その他の情報は、トークンのフォーマット時に提供され、保存されます。この場合、会社はユーザーに特定の Phone Home URL を提供します。ユーザーが URL を送信すると、フォーマットプロセスにより、残りの情報が Phone Home プロファイルに追加されます。フォーマットプロセスは、ユーザーに変わりません。

図3.2 Phone Home 情報のプロンプト



3.3.2. グローバル Phone Home 情報の設定

Phone Home は、セキュリティトークンがマシンに挿入されると自動的にトリガーされます。システムは、トークンから Phone Home URL を即座に読み込み、TPS サーバーと通信しようとしています。新しいトークンまたは以前にフォーマットされたトークンの場合、Phone Home 情報はカードで利用できない場合があります。

Enterprise Security Client 設定ファイル **esc-prefs.js** には、グローバル Phone Home URL のデフォルト設定を可能にするパラメーターがあります。このパラメーターは **esc.global.phone.home.url** で、デフォルトではファイルには含まれません。

グローバルの Phone Home URL を定義するには、以下を実行します。

1. 既存の Enterprise Security Client のユーザープロファイルディレクトリーを削除します。プロファイルディレクトリーは、スマートカードが挿入されると自動的に作成されます。Red Hat Enterprise Linux では、プロファイルディレクトリーは `~/.redhat/esc` になります。
2. **esc-prefs.js** ファイルを開きます。Red Hat Enterprise Linux (32 ビット) では、プロファイルディレクトリーは `/usr/lib/esc-1.1.0/defaults/preferences` になります。64 ビットシステムでは、これは `/usr/lib64/esc-1.1.0/defaults/preferences` です。
3. グローバルの Phone Home パラメーター行を **esc-prefs.js** ファイルに追加します。以下に例を示します。

```
pref("pref("esc.global.phone.home.url","https://localhost:8443/tps/phoneHome");");
```

URL は DNS およびネットワーク設定に応じて、マシン名、完全修飾ドメイン名、または IPv4 または IPv6 アドレスを参照できます。

3.3.3. Phone Home 情報の手動によるトークンへの追加

Phone Home 情報は、以下の 2 つの方法の 1 つでトークンに手動で配置できます。

- 推奨される方法は、情報がファクトリーでトークンに書き込まれることです。トークンが製造元から順序付けられると、会社は、送信時にトークンを設定する方法の詳細を提供します。
- トークンが空白である場合、企業の IT 部門は、トークンの小規模なグループのフォーマット時に情報を提供できます。

以下の情報は、`~/.redhat/esc/alphabetic_string.default/prefs.js` ファイルの各スマートカードの **Phone Home 機能**によって使用されます。

- TPS サーバーおよびポート以下に例を示します。

```
"esc.key.token_ID.tps.url" = "http://server.example.com:7888/nk_service"
```

- TPS 登録インターフェイス URL。以下に例を示します。

```
"esc.key.token_ID.tps.enrollment-ui.url" = "http://server.example.com:7888/cgi_bin/esc.cgi?"
```

- 発行先の会社名または ID。以下に例を示します。

```
"esc.key.token_ID.issuer.name" = "Example Corp"
```

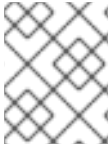
- Phone Home URL。以下に例を示します。

```
"esc.key.token_ID.phone.home.url" = "https://localhost:8443/tps/phoneHome"
```

- オプションで、登録したスマートカードが挿入されたときにアクセスするデフォルトのブラウザ URL。

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

prefs.js ファイルで使用されるパラメーターの詳細は、[???](#) に記載されています。



注記

これらのパラメーターの URL は DNS およびネットワーク設定に応じて、マシン名、完全修飾ドメイン名、または IPv4 または IPv6 アドレスを参照できます。

3.3.4. TPS を設定し、Phone Home を使用する

Phone Home を使用するように TPS が正しく設定されている場合に限り、Phone Home 機能と、それに使用される各種情報が利用できます。TPS が Phone Home に対して設定されていない場合、この機能は無視されます。Phone Home は、`/var/lib/pki-tps/cgi-bin/home` ディレクトリーの `index.cgi` で設定されます。これにより、Phone Home 情報が XML に出力されます。

例3.1「TPS Phone Home 設定ファイル」は、Phone Home 機能を設定する TPS サブシステムによって使用される XML ファイルの例を示しています。

例3.1 TPS Phone Home 設定ファイル

```
<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>
```

TPS 設定 URI は TPS サーバーの URL で、残りの Phone Home 情報を Enterprise Security Client に返します。この URL の例は **https://localhost:8443/tps/phoneHome** です。この URL は、必要に応じてマシン名、完全修飾ドメイン名、または IPv4 アドレスまたは IPv6 アドレスを参照できます。TPS 設定 URI にアクセスすると、TPS サーバーは、すべてのホームディレクトリー情報を Enterprise Security Client に返すように求められます。

Smart Card サーバーの URL をテストするには、**TPS Config URI** フィールドにアドレスを入力し、**Test URL** をクリックします。

サーバーに正常に接続すると、成功を示すメッセージボックスが表示されます。テスト接続に失敗すると、エラーダイアログが表示されます。

3.4. 登録するユーザーの設定

Token Processing System がインストールされると、設定の1つに、トークンの登録が許可されるユーザーが含まれる LDAP ディレクトリーがあります。この認証ディレクトリーに保存されているユーザーのみが、トークンの登録、フォーマット、または所有を許可されています。トークンまたはスマートカードの登録を試みる前に、操作を要求するのユーザーが LDAP ディレクトリーにエントリーがあることを確認してください。

TPS は、LDAP ディレクトリーの特定ベース DN を確認するように設定されています。これは、TPS の **CS.cfg** で設定されます。

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

ユーザーがトークンの登録を許可するには、ユーザーはベース DN の背後にある必要があります。

ユーザーにエントリーがない場合は、ユーザーにトークンを登録する前に、指定したベース DN の指定された LDAP ディレクトリーに管理者がユーザーを追加する必要があります。

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

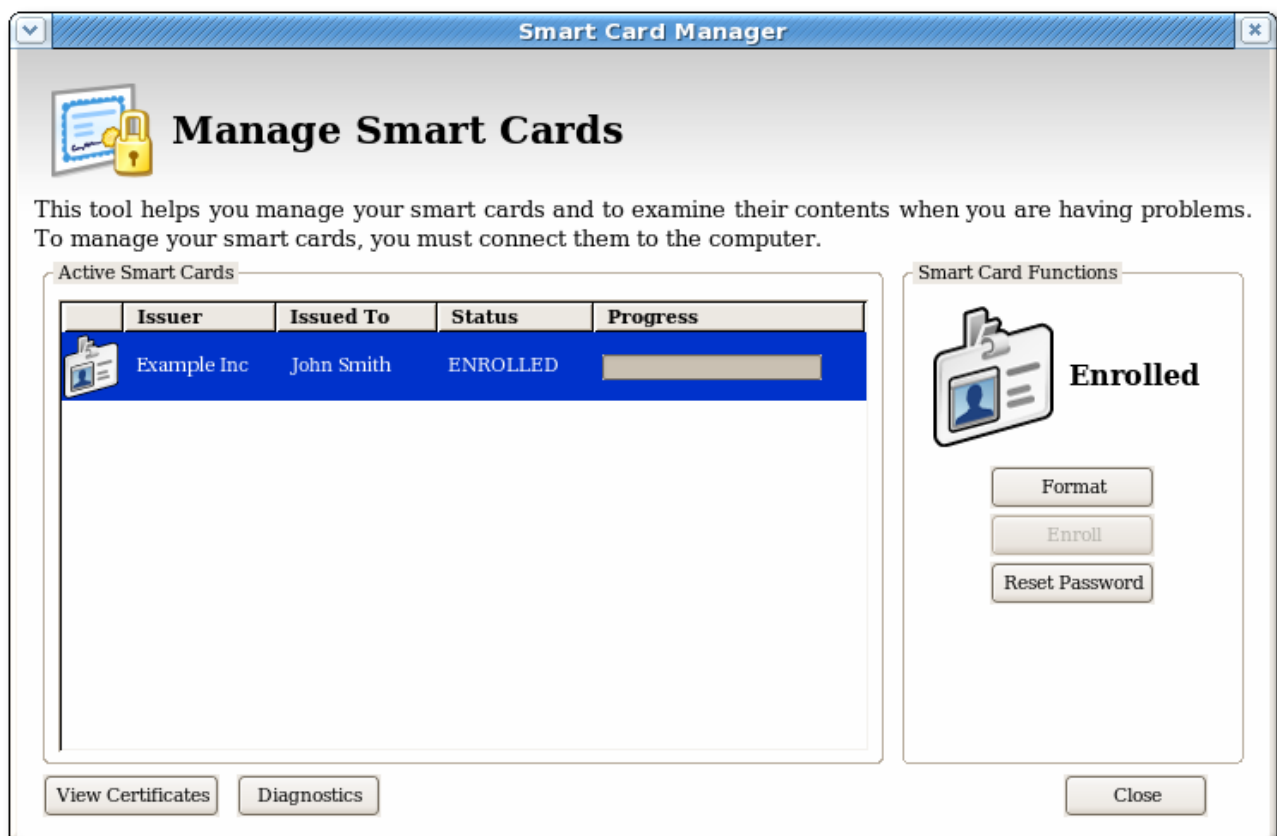
```
dn: uid=jsmith,ou=People,dc=example,dc=com
objectclass: person
objectclass: inetorgperson
objectclass: top
uid: jsmith
cn: John Smith
email: jsmith@example.com
userPassword: secret
```

3.5. スマートカードの管理

Manage Smart Cards ページを使用すると、トークンに保存されている暗号鍵のいずれかに適用できる多くの操作を実行できます。

このページを使用して、トークンのフォーマット、カードのパスワードの設定とリセット、およびカード情報の表示を行うことができます。その他の2つの操作(トークンの登録および診断ログの表示)は、**Manage Smart Cards** ページからもアクセスできます。これらの操作は他のセクションで扱われます。

図3.3 スマートカードページの管理



3.5.1. スマートカードのフォーマット

スマートカードをフォーマットすると、初期化されていない状態にリセットされます。これにより、以前に生成されたユーザーのキーペアがすべて削除され、登録時にスマートカードに設定されたパスワードが消去されます。

TPS サーバーは、新しいバージョンのアプリッキー鍵と対称キーをカードに読み込むように設定できます。TPS は、Red Hat Enterprise Linux 7.9 に同梱される CoolKey アプレットをサポートします。

スマートカードをフォーマットするには、以下を行います。

1. 対応しているスマートカードをコンピューターに挿入します。カードが **Active Smart Cards** テーブルに表示されることを確認します。
2. **Manage Smart Cards** 画面の **Smart Card Functions** セクションで、**Format** をクリックします。
3. TPS がユーザー認証用に設定されている場合は、認証ダイアログにユーザー認証情報を入力して、**Submit** をクリックします。
4. フォーマットプロセス中に、カードのステータスが BUSY に変更され、進捗バーが表示されます。フォーマットプロセスが完了すると、成功メッセージが表示されます。**OK** をクリックしてメッセージボックスを閉じます。
5. フォーマットプロセスが完了すると、**Active Smart Cards** の表に、UNINITIALIZED というカードステータスが表示されます。

3.5.2. スマートカードパスワードのリセット

カードを登録した後にユーザーがスマートカードのパスワードを忘れた場合は、パスワードをリセットできます。スマートカードのパスワードをリセットするには、次のコマンドを実行します。

1. 対応しているスマートカードをコンピューターに挿入します。カードが **Active Smart Cards** テーブルに表示されることを確認します。
2. **Manage Smart Cards** 画面の **Smart Card Functions** セクションで、**Reset Password** をクリックして、**Password** ダイアログを表示します。
3. **Enter new password** フィールドに新しいスマートカードパスワードを入力します。
4. **Re-Enter password** フィールドで新しいスマートカードパスワードを確認して、**OK** をクリックします。



5. TPS がユーザー認証用に設定されている場合は、認証ダイアログにユーザー認証情報を入力して、**Submit** をクリックします。
6. パスワードのリセットが完了するのを待ちます。

3.5.3. 証明書の表示

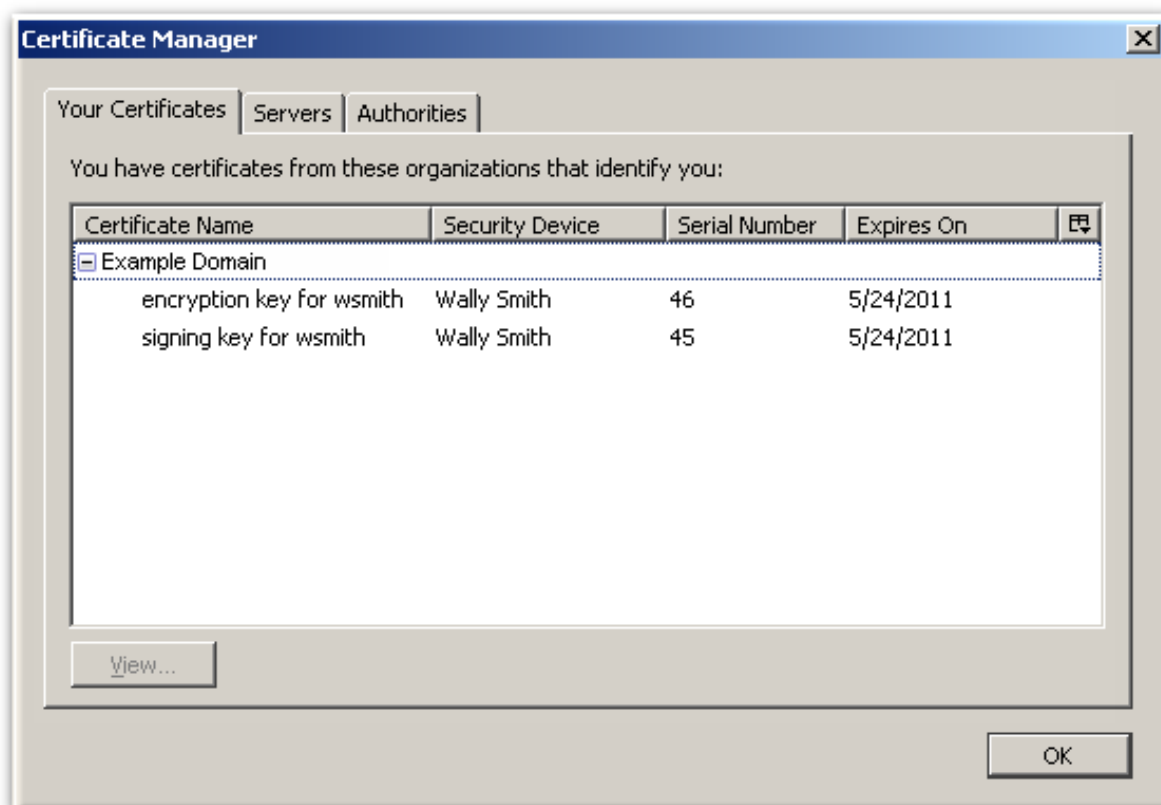
Smart Card Manager は、保存した鍵や証明書など、選択したスマートカードの基本情報を表示できます。証明書情報を表示するには、以下を行います。

1. 対応しているスマートカードをコンピューターに挿入します。カードが **Active Smart Cards** テーブルに表示されることを確認します。
2. 一覧からカードを選択し、**View Certificates** をクリックします。



これにより、シリアル番号、証明書のニックネーム、有効日など、カードに保存されている証明書の基本情報を表示します。

3. 証明書に関する詳細情報を表示するには、一覧から証明書を選択して **表示** をクリックします。



3.5.4. CA 証明書のインポート

Xulrunner Gecko エンジン、ブラウザーや Enterprise Security Client のようにクライアントがアクセスできる SSL ベースの URL に対する文字列制御を実装します。(Xulrunner フレームワークを介して) Enterprise Security Client が URL を信頼しない場合、URL にアクセスできなくなります。

SSL ベースの URL を信頼する1つの方法として、サイトの証明書を発行した CA の CA 証明書チェーンをインポートおよび信頼する方法があります。(もう1つは、「[サーバーの例外の追加](#)」のようにサイトに信頼 **セキュリティ例外** を作成することです)

スマートカードの証明書を発行する CA は、Enterprise Security Client アプリケーションによって信頼される必要があります。つまり、その CA 証明書を Enterprise Security Client にインポートする必要があります。

1. Web ブラウザーで CA のエンドユーザーページを開きます。

`https://server.example.com:9444/ca/ee/ca/`

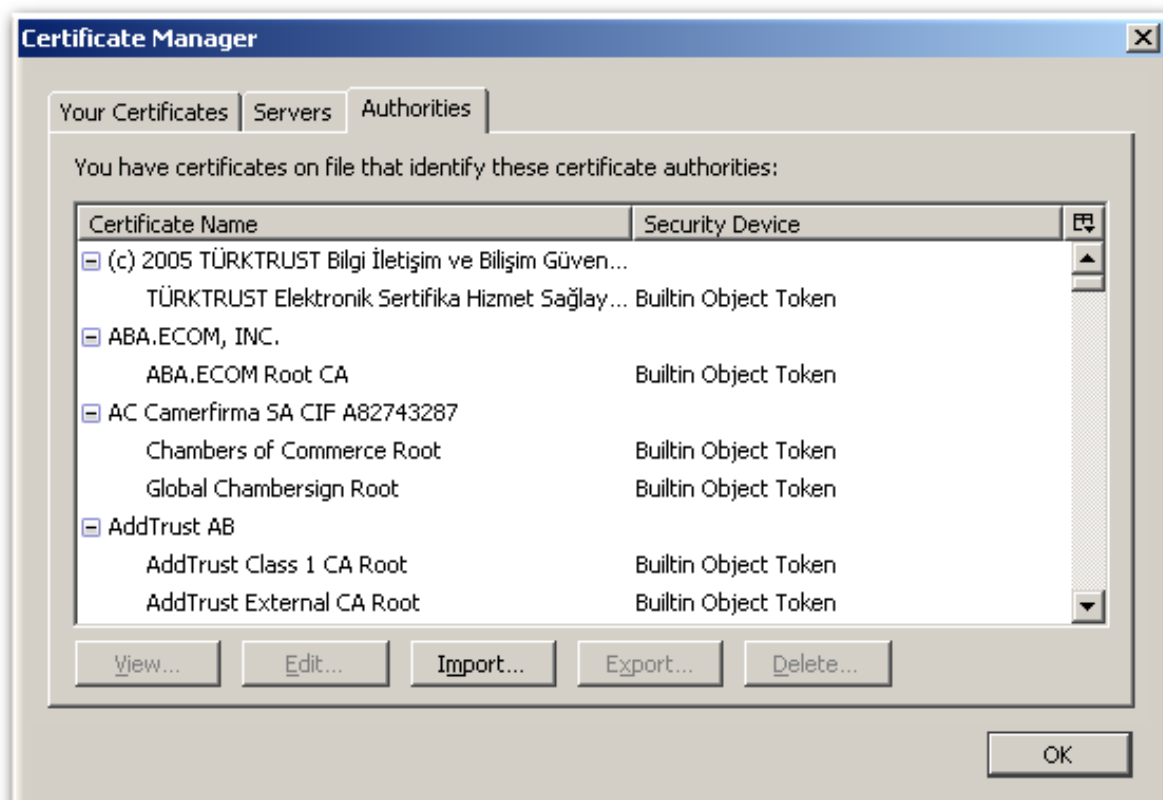
2. 上部の **Retrieval** タブをクリックします。
3. 左側のメニューで **Import CA Certificate Chain** リンクをクリックします。
4. チェーンをファイルとしてダウンロードするにはラジオボタンを選択し、ダウンロードしたファイルの場所と名前を書き留めます。
5. Enterprise Security Client を開きます。



6. 証明書の表示 ボタンをクリックします。



7. 認証 タブをクリックします。
8. インポート をクリックします。



9. CA 証明書チェーンファイルを参照し、これを選択します。
10. プロンプトが表示されたら、CA を信頼することを確認します。

3.5.5. サーバーの例外の追加

Xulrunner Gecko エンジン、ブラウザーや Enterprise Security Client のようにクライアントがアクセスできる SSL ベースの URL に対する文字列制御を実装します。(Xulrunner フレームワークを介して) Enterprise Security Client が URL を信頼しない場合、URL にアクセスできなくなります。

SSL ベースの URL を信頼する1つの方法として、サイトの証明書をインポートして、Enterprise Security Client がそれを認識するよう強制するサイトに信頼 **セキュリティー例外** を作成するものがあります。(もう1つのオプションは、「[CA 証明書のインポート](#)」のように、サイトの CA 証明書チェーンをインポートし、自動的に信頼するオプションです)。

スマートカードは、特別なセキュリティー例外を必要とする SSL 経由でサービスまたは Web サイトにアクセスするために使用できます。これらの例外は、Mozilla Firefox などのブラウザーで Web サイトの例外を設定するのと同様に、Enterprise Security Client で設定できます。

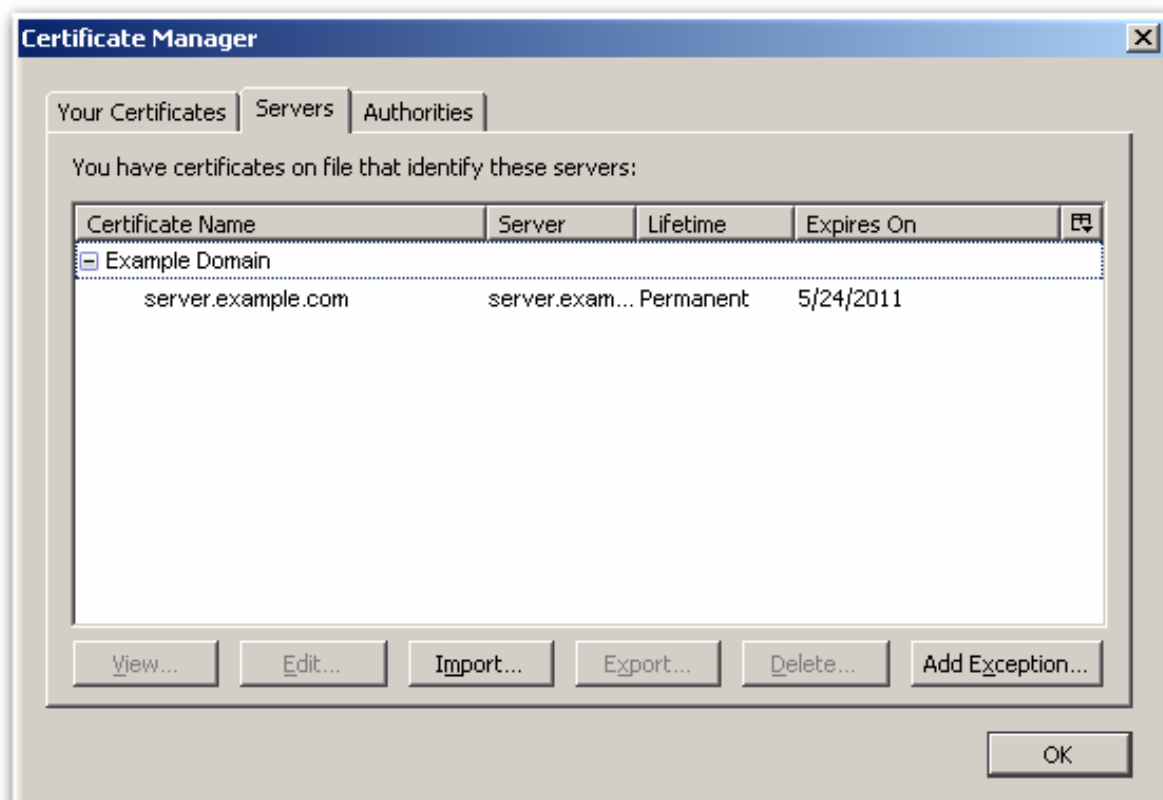
1. Enterprise Security Client を開きます。



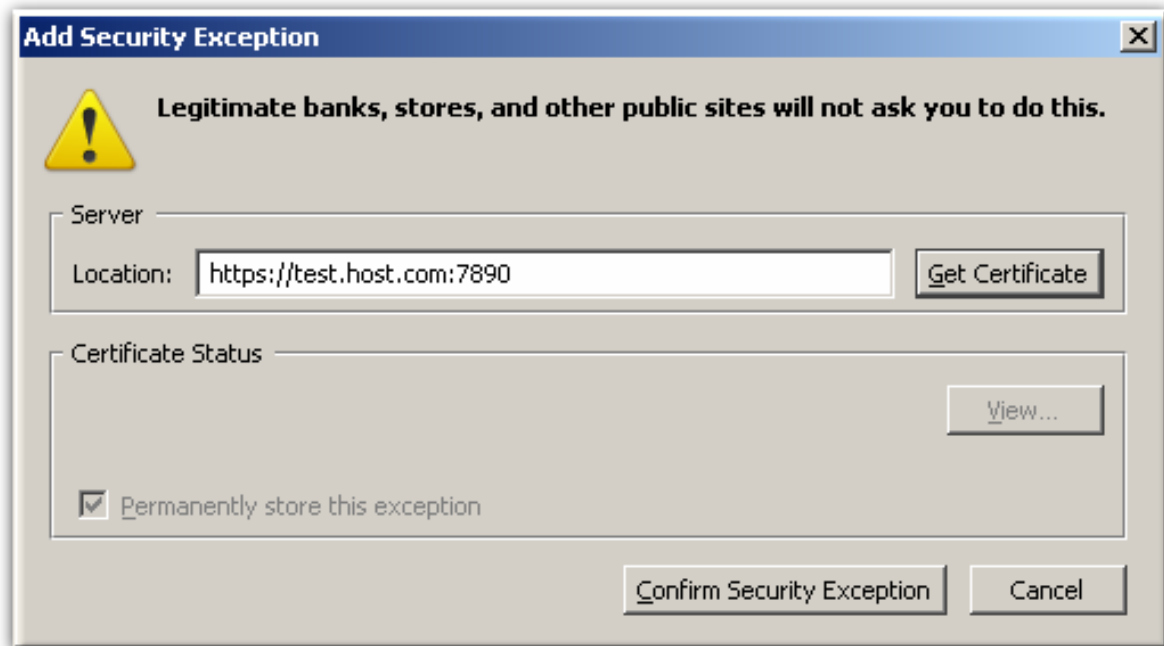
2. 証明書の表示 ボタンをクリックします。



3. **Servers** タブをクリックします。
4. **Add Exception** をクリックします。



5. スマートカードがアクセスに使用されるサイトまたはサービスの URL (ポート番号) を入力します。次に、**証明書の取得** ボタンをクリックして、サイトのサーバー証明書をダウンロードします。

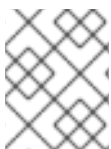


6. セキュリティー例外の確認をクリックして、許可されたサイトの一覧にサイトを追加します。

3.5.6. スマートカードの登録

ほとんどのスマートカードは、自動登録手順を使用して自動的に登録されます。**Manage Smart Cards** 機能を使用して、スマートカードを手動で登録することもできます。

ユーザーキーペアでトークンを登録する場合、トークンは SSL クライアント認証や S/MIME などの証明書ベースの操作に使用できます。



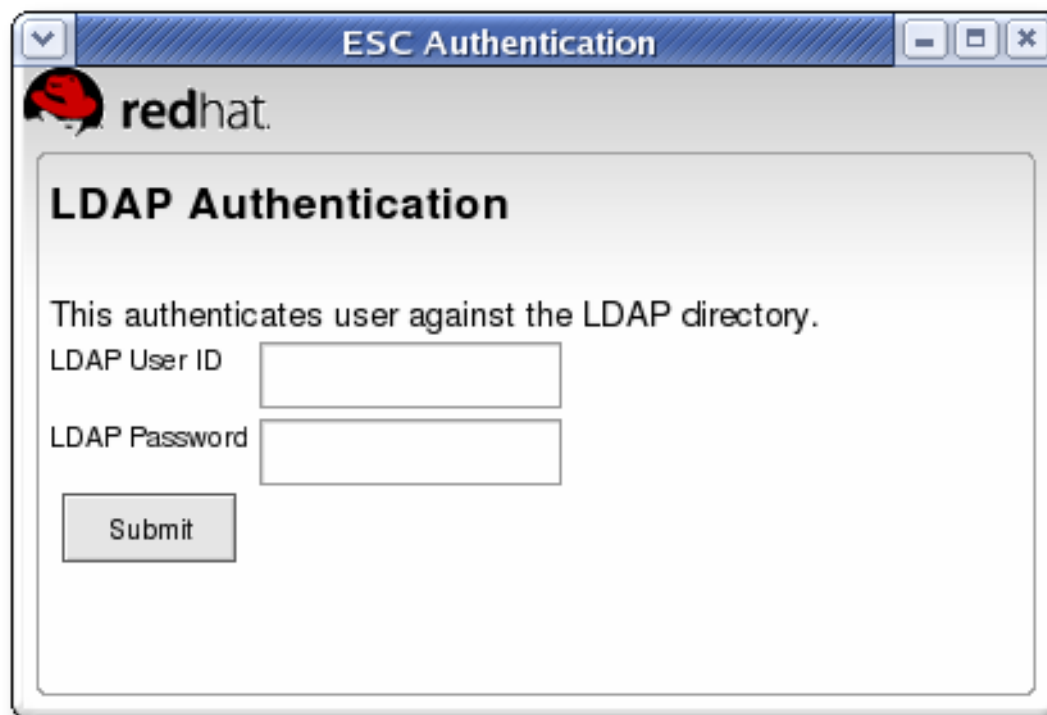
注記

TPS サーバーは、トークンが失われた場合のリカバリーのために、サーバー上でユーザーキーペアを生成し、DRM サブシステムでアーカイブするように設定できます。

スマートカードを手動で登録するには、以下を実行します。

1. 対応している未登録のスマートカードをコンピューターに挿入します。カードが **Active Smart Cards** テーブルに表示されることを確認します。
2. **Enroll** をクリックして、**Password** ダイアログを表示します。
3. **Enter a password** フィールドに新しいキーパスワードを入力します。
Re-Enter a password フィールドで新規パスワードを確認します。
4. **OK** をクリックして登録を開始します。
5. TPS がユーザー認証用に設定されている場合は、認証ダイアログにユーザー認証情報を入力して、**Submit** をクリックします。

TPS がキーを DRM にアーカイブするように設定されている場合は、登録プロセスでキーの生成およびアーカイブが開始されます。



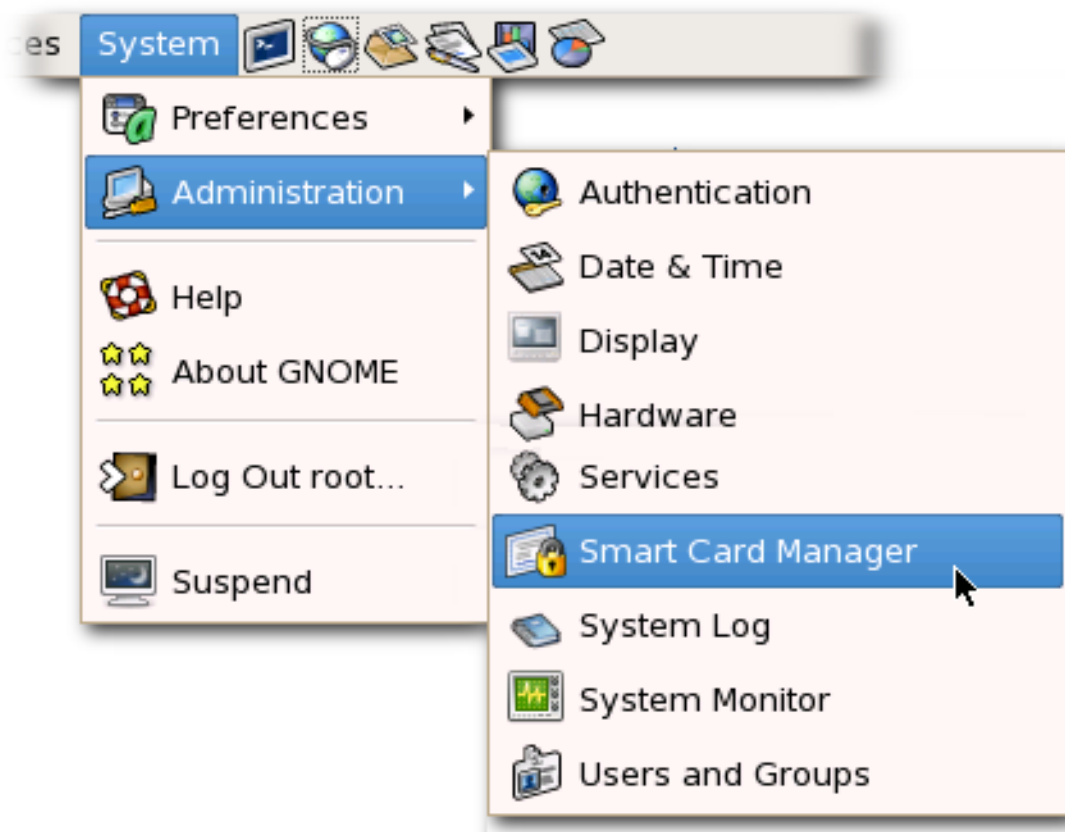
登録が完了すると、スマートカードのステータスが ENROLLED として表示されます。

3.6. 問題の診断

Enterprise Security Client には、基本的な診断ツールと、スマートカードの挿入や削除、カードのパスワードの変更など、エラーや一般的なイベントを記録する簡単なインターフェイスが含まれています。診断ツールは、Enterprise Security Client、スマートカード、および TPS 接続の問題について、ユーザーに特定して通知できます。

Diagnostics Information ウィンドウを開くには、次を実行します。

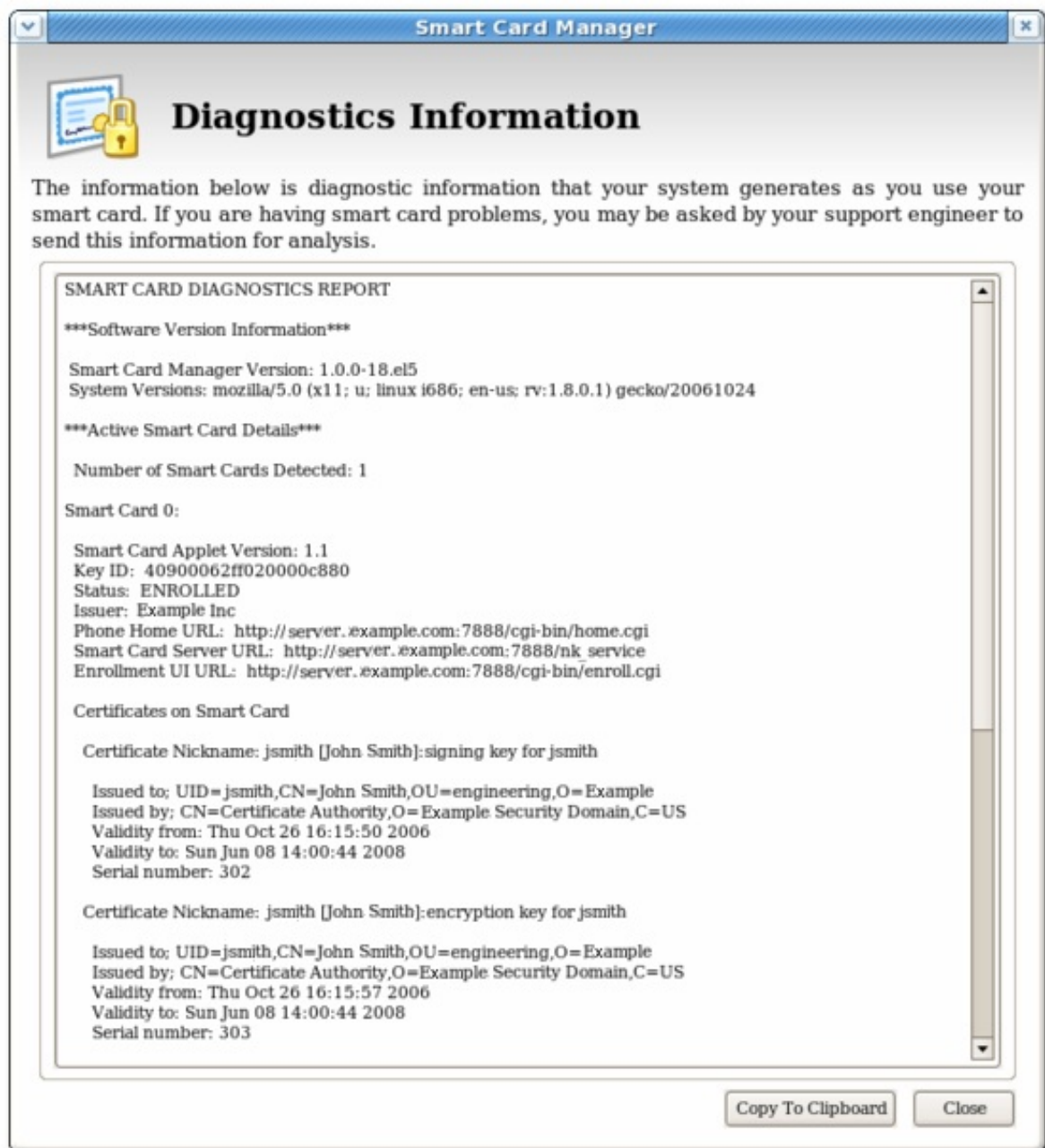
1. Enterprise Security Client を開きます。



2. 一覧から確認するスマートカードを選択します。
3. **診断** ボタンをクリックします。



4. これにより、選択したスマートカードの **Diagnostic Information** ウィンドウが開きます。



Diagnostics Information 画面には、以下の情報が表示されます。

- Enterprise Security Client のバージョン番号。
- クライアントが実行されている Xulrunner フレームワークのバージョン情報。
- Enterprise Security Client によって検出されたカードの数。

検出されたカードごとに、以下の情報が表示されます。

- スマートカードで実行しているアプレットのバージョン。
- スマートカードの英数字 ID。
- カードのステータス。以下の 3 つのいずれかになります。
 - NO_APPLET キーは検出されませんでした。
 - UNINITIALIZED。キーが検出されても、証明書は登録されていません。

- **ENROLLED**。検出されたカードが、証明書およびカード情報に登録されています。
- カードの Phone Home URL。これは、すべての Phone Home 情報を取得するための URL です。
- **Example Corp.**などのカード発行者名。
- カードの ATR (answer-to-reset) 文字列。これは、スマートカードの異なるクラスの識別に使用できる一意の値です。たとえば、以下のようになります。

```
3BEC00FF8131FE45A0000000563333304A330600A1
```

- TPS Phone Home URL。
- TPS サーバーの URL。これは Phone Home 経由で取得されます。
- TPS 登録フォーム URL。これは Phone Home 経由で取得されます。
- カードに含まれる各証明書に関する詳細情報。
- 最新の Enterprise Security Client エラーと一般的なイベントの稼働中のログ。

Enterprise Security Client は、2 種類の診断情報を記録します。これは、スマートカードによって返された **エラー** を記録し、Enterprise Security Client 経由で発生した **イベント** を記録します。また、スマートカード設定の基本情報を返します。

3.6.1. エラー

- Enterprise Security Client はカードを認識しません。
- 証明書の登録、パスワードのリセット、フォーマット操作など、スマートカードの操作中に問題が発生します。
- Enterprise Security Client は、スマートカードへの接続を失います。これは、**PCSC** デーモンとの通信で問題が発生した場合に発生する可能性があります。
- Enterprise Security Client と TPS 間の接続が失われます。

スマートカードは、TPS に特定のエラーコードを報告できます。これは、メッセージの原因に応じて、TPS の **tps-debug.log** ファイルまたは **tps-error.log** ファイルに記録されます。

表3.1 スマートカードのエラーコード

| 戻りコード | 説明 |
|------------|---------------------|
| 一般的なエラーコード | |
| 6400 | 特定の診断なし |
| 6700 | Lc の誤った長さ |
| 6982 | セキュリティーステータスが満たされない |
| 6985 | 使用条件が満たされない |

| 戻りコード | 説明 |
|---------------|--------------------|
| 6a86 | 間違った P1P2 |
| 6d00 | 無効な命令 |
| 6e00 | 無効なクラス |
| インストール読み込みエラー | |
| 6581 | メモリー障害 |
| 6a80 | データフィールドの誤ったパラメーター |
| 6a84 | 不十分なメモリー容量 |
| 6a88 | 参照データが見つからない |
| 削除エラー | |
| 6200 | アプリケーションを論理的に削除 |
| 6581 | メモリー障害 |
| 6985 | 参照データを削除できない |
| 6a88 | 参照データが見つからない |
| 6a82 | アプリケーションが見つからない |
| 6a80 | コマンドデータの値が正しくない |
| データ取得エラー | |
| 6a88 | 参照データが見つからない |
| ステータス取得エラー | |
| 6310 | より多くのデータが利用可能 |
| 6a88 | 参照データが見つからない |
| 6a80 | コマンドデータの値が正しくない |
| 読み込みエラー | |

| 戻りコード | 説明 |
|-------|-------------|
| 6581 | メモリー障害 |
| 6a84 | 不十分なメモリー容量 |
| 6a86 | 間違った P1/P2 |
| 6985 | 使用条件が満たされない |

3.6.2. イベント

- カードの挿入および削除、正常に完了した操作、エラーの原因となるカード操作などの単純なイベント。
- TPS から Enterprise Security Client にエラーが報告されます。
- NSS 暗号ライブラリーが初期化されています。
- その他の低レベルのスマートカードイベントが検出されています。

第4章 WEB およびメールクライアントでのスマートカードの使用

スマートカードの登録後、スマートカードは SSL クライアント認証および S/MIME メールアプリケーションに使用できます。PKCS #11 モジュールの名前は異なり、オペレーティングシステムによって異なるディレクトリにあります。

表4.1 PKCS #11 モジュールの場所

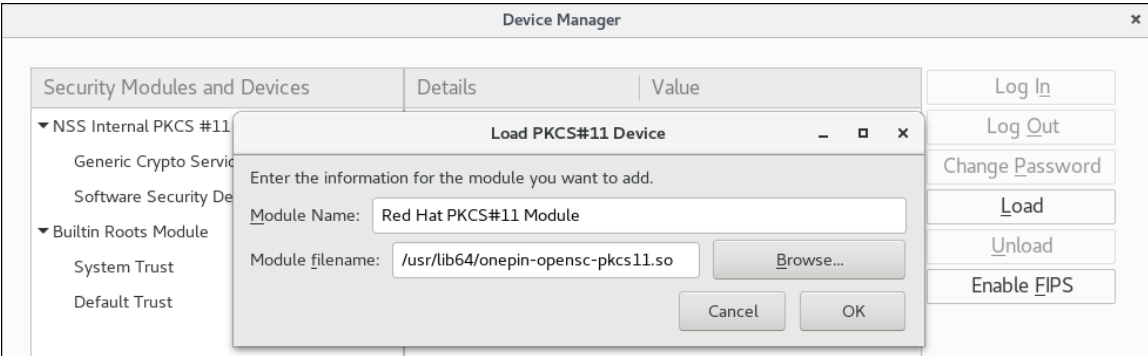
| プラットフォーム | モジュール名 | 場所 |
|--------------------------|-------------------------|-------------|
| Red Hat Enterprise Linux | onopin-opensc-pkcs11.so | /usr/lib64/ |

4.1. トークンに対して SSL をサポートするブラウザーの設定

トークンの SSL をサポートするように Firefox ブラウザーを設定するには、以下を実行します。

1. **Edit** メニューを開き、**Preferences** を選択します。

Firefox にメニューバーが表示されない場合は、**Alt** キーを押して一時的に表示します。
2. **Advanced** エントリで **Certificates** タブを選択し、**Security Devices** ボタンをクリックします。
3. PKCS #11 ドライバーを追加します。
 - a. **Load** ボタンをクリックします。
 - b. モジュール名を入力します。
 - c. **Browse** をクリックして、Enterprise Security Client PKCS #11 ドライバーライブラリーを選択し、**OK** をクリックします。



4. CA がまだ信頼されていない場合は、CA 証明書をダウンロードしてインポートします。
 1. CA で **SSL End Entity** ページを開きます。たとえば、以下のようになります。

https://server.example.com:9444/ca/ee/ca/
 2. **Retrieval** タブをクリックし、**Import CA Certificate Chain** をクリックします。
 3. **Download the CA certificate chain in binary form** をクリックし、**Submit** をクリックします。

4. 証明書チェーンを保存する適切なディレクトリーを選択し、**OK** をクリックします。
 5. **Edit > Preferences** をクリックして、**Advanced** タブを選択します。
 6. **証明書の表示** ボタンをクリックします。
 7. **Authorities** をクリックし、CA 証明書をインポートします。
5. 証明書信頼関係を設定します。
1. **Edit > Preferences** をクリックして、**Advanced** タブを選択します。
 2. **証明書の表示** ボタンをクリックします。
 3. **Edit** をクリックして、Web サイトへの信頼を設定します。

証明書は SSL に使用できます。

第5章 ENTERPRISE SECURITY CLIENT の設定

Enterprise Security Client は Mozilla XULRunner をベースとしており、Mozilla に組み込まれた設定機能を使用して、Enterprise Security Client の単純な設定が可能になりました。簡単な UI は、[3章Enterprise Security Client の使用](#) で説明され、最も重要な設定を管理します。



注記

Enterprise Security Client は、追加の設定なしに起動できます。

5.1. ENTERPRISE SECURITY CLIENT 設定の概要

Enterprise Security Client は中間フロントエンドで、ユーザー (およびそのトークン)、トークン処理システム、および認証局との間の接続を提供します。Enterprise Security Client は、以下の 2 つの異なるインターフェイスを提供します。

- XUL および JavaScript に基づくローカルインターフェイス
- CGI、HTML、および JavaScript に基づくリモートアクセスに使用できる web ホストインターフェイス

ローカルサーバーからアクセスされるプライマリー Enterprise Security Client ユーザーインターフェイスには、Mozilla XULRunner 技術が含まれています。XULRunner はランタイムパッケージで、ユーザーインターフェイス用の機能が充実した XML マークアップ言語である XUL をベースとするスタンドアロンアプリケーションをホストし、アプリケーションの HTML と比較していくつかの利点があります。

- 幅広い UI ウィジェットセット。およびプレゼンテーションにわたる優れた制御。
- クライアントマシンへのローカルマークアップにより、HTML よりも権限レベルが高くなります。
- 便利なプログラム論理スクリプトや XPCOM 技術を利用できるスクリプト言語、JavaScript。

Web ホストインターフェイスのすべてのファイルをカスタマイズおよび編集し、Enterprise Security Client の動作や外観を理由に合わせて変更できます。

Token Processing System とともに Enterprise Security Client は、各種ユーザーがさまざまなトークン登録パスを利用できるように、さまざまな **ユーザープロファイル** をサポートします。Enterprise Security Client と TPS は、証明書設定がさまざまなタイプのトークンにカスタム定義できるように、両者とも異なる **トークンプロファイル** もサポートしています。これらの設定はいずれも TPS で設定され、[『Red Hat Certificate System 計画、インストール、およびデプロイメントのガイド』](#) で説明されています。

5.1.1. 設定ファイルについて

Enterprise Security Client は、設定ファイルを使用して Mozilla アプリケーションと同様に設定されます。主な設定ファイルは **esc-prefs.js** で、Enterprise Security Client とともにインストールされます。2 つ目は、Mozilla プロファイルディレクトリーの **prefs.js** で、Enterprise Security Client の初回起動時に作成されます。

Enterprise Security Client は、サポートされるプラットフォームごとに Mozilla 設定を使用します。デフォルトの設定ファイルは、各プラットフォームの以下のディレクトリーにあります。

- Red Hat Enterprise Linux 32 ビットでは、これは **/usr/lib/esc-1.1.0/defaults/preferences/esc-prefs.js** にあります。

- Red Hat Enterprise Linux 64 ビットでは、これは `/usr/lib64/esc-1.1.0/defaults/preferences/esc-prefs.js` にあります。

esc-prefs.js ファイルは、Enterprise Security Client の初回起動時に使用するデフォルト設定を指定します。これには、TPS サブシステムに接続し、パスワードプロンプトを設定し、Phone Home 情報を設定するパラメーターが含まれます。各設定は、前に 単語で 囲まれ、パラメーターと値は括弧で囲まれます。以下に例を示します。

```
pref(parameter, value);
```

esc-prefs.js ファイルパラメーターは [表5.1「esc-prefs.js Parameters」](#) に記載されています。デフォルトの **esc-prefs.js** ファイルは [例5.1「デフォルトの esc-prefs.js ファイル」](#) に表示されます。

表5.1 esc-prefs.js Parameters

| パラメーター | 説明 | 注記およびデフォルト |
|-----------------------------|---|---|
| toolkit.defaultChromeURI | XUL Chrome ページへのアクセスに使用するエンタープライズセキュリティクライアントの URL を定義します。 | ("toolkit.defaultChromeURI", "chrome://esc/content/settings.xul") |
| esc.tps.message.timeout | TPS に接続するためのタイムアウト期間を秒単位で設定します。 | ("esc.tps.message.timeout", "90"); |
| esc.disable.password.prompt | パスワードプロンプトを有効にします。これは、スマートカードから証明書情報を読み取るのにパスワードが必要であることを意味します。 パスワードプロンプトはデフォルトで無効になっているため、Enterprise Security Client を使用できます。ただし、ある企業がセキュリティ担当者を使用してトークン操作を管理する場合など、セキュリティコンテキストでは、パスワードプロンプトを有効にして、Enterprise Security Client へのアクセスを制限します。 | ("esc.disable.password.prompt", "yes"); |

| パラメーター | 説明 | 注記およびデフォルト |
|---------------------------|--|--|
| esc.global.phone.home.url | <p>TPS サーバーへの接続に使用する URL を設定します。</p> <p>通常、Phone Home 情報はすでにアプレットを介してトークンに設定されます。トークンに Phone Home 情報がない場合 (TPS サーバーと通信する方法がない場合)、Enterprise Security Client はグローバルのデフォルト Phone Home URL をチェックします。</p> <p>この設定は、明示的に設定されている場合にのみチェックされます。この設定はクライアントでフォーマットされたすべてのトークンに適用されるため、このパラメーターを設定すると、すべてのトークンが同じ TPS をポイントするように強制されます。特定の動作が望ましい場合にのみ、このパラメーターを使用してください。</p> | <pre>("esc.global.phone.home.url", "http://server.example.com:7888/cgi-bin/home/index.cgi");</pre> |
| esc.global.alt.nss.db | <p>サーバー上のすべての Enterprise Security Client ユーザーが使用している共通のセキュリティーデータベースが含まれるディレクトリーを参照します。</p> <p>この設定は、明示的に設定されている場合にのみチェックされます。これが設定されていない場合、各ユーザーは共有データベースではなく、個別のプロファイルセキュリティーデータベースにのみアクセスします。</p> | <pre>prefs("esc.global.alt.nss.db", "C:/Documents and Settings/All Users/shared-db");</pre> |

例5.1 デフォルトの esc-prefs.js ファイル

このファイルのコメントは、例には含まれません。

```
#pref("toolkit.defaultChromeURI", "chrome://esc/content/settings.xul");
pref("signed.applets.codebase_principal_support",true); for internal use only

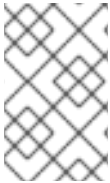
pref("capability.principal.codebase.p0.granted", "UniversalXPConnect"); for internal use only
pref("capability.principal.codebase.p0.id", "file:///"); for internal use only

pref("esc.tps.message.timeout","90");

#Hide the format button or not.
pref("esc.hide.format","no");

#Use this if you absolutely want a global phone home url for all tokens
#Not recommended!
#pref("esc.global.phone.home.url","http://test.host.com:7888/cgi-bin/home/index.cgi");
```

Enterprise Security Client の起動時に、システム上のユーザーごとに個別の一意のプロファイルディレクトリーを作成します。Red Hat Enterprise Linux では、これらのプロファイルは `~/.redhat/esc/ alphanumeric_string . default/prefs.js` に保存されます。



注記

Enterprise Security Client でユーザーの設定値に変更が必要となると、更新された値はデフォルトの JavaScript ファイルではなく、ユーザーのプロファイルエリアに書き込まれます。

表5.2 「PREFS.js パラメーター」 `prefs.js` ファイルの最も関連性の高いパラメーターを一覧表示します。このファイルの編集は複雑です。`prefs.js` ファイルは、Enterprise Security Client によって動的に生成および編集され、このファイルへの手動の変更は、Enterprise Security Client の終了時に上書きされます。

表5.2 PREFS.js パラメーター

| パラメーター | 説明 | 注記およびデフォルト |
|--|---|--|
| <code>esc.tps.url</code> | TPS への接続に使用する Enterprise Security Client の URL を設定します。これはデフォルトでは設定されません。 | |
| <code>esc.key.token_ID.tps.url</code> | TPS との通信に使用するホスト名とポートを設定します。 この Phone Home 情報がファクトリーでカードに書き込まれていない場合は、TPS URL、登録ページの URL、発行者の名前、および Phone Home URL を追加して、カードに手動で追加できます。 | <code>("esc.key.token_ID.tps.url" = "https://test.host.com:8443/tps/tps");</code> |
| <code>esc.key.token_ID.issuer.name</code> | トークンを登録する組織の名前を指定します。 | <code>("esc.key.token_ID.issuer.name" = "Example Corp");</code> |
| <code>esc.key.token_ID.phone.home.url</code> | TPS の Phone Home 機能にアクセスするために使用する URL を指定します。 トークンが Phone Home 情報を指定しない場合に、グローバル Phone Home パラメーターは、トークン登録で使用するデフォルトを設定します。このパラメーターを特定のトークン ID 番号に設定すると、指定の Phone Home パラメーターがそのトークンにのみ適用されます。 | <code>("esc.key.token_ID.phone.home.url" = "http://server.example.com:7888/cgi-bin/home/index.cgi?");</code> |

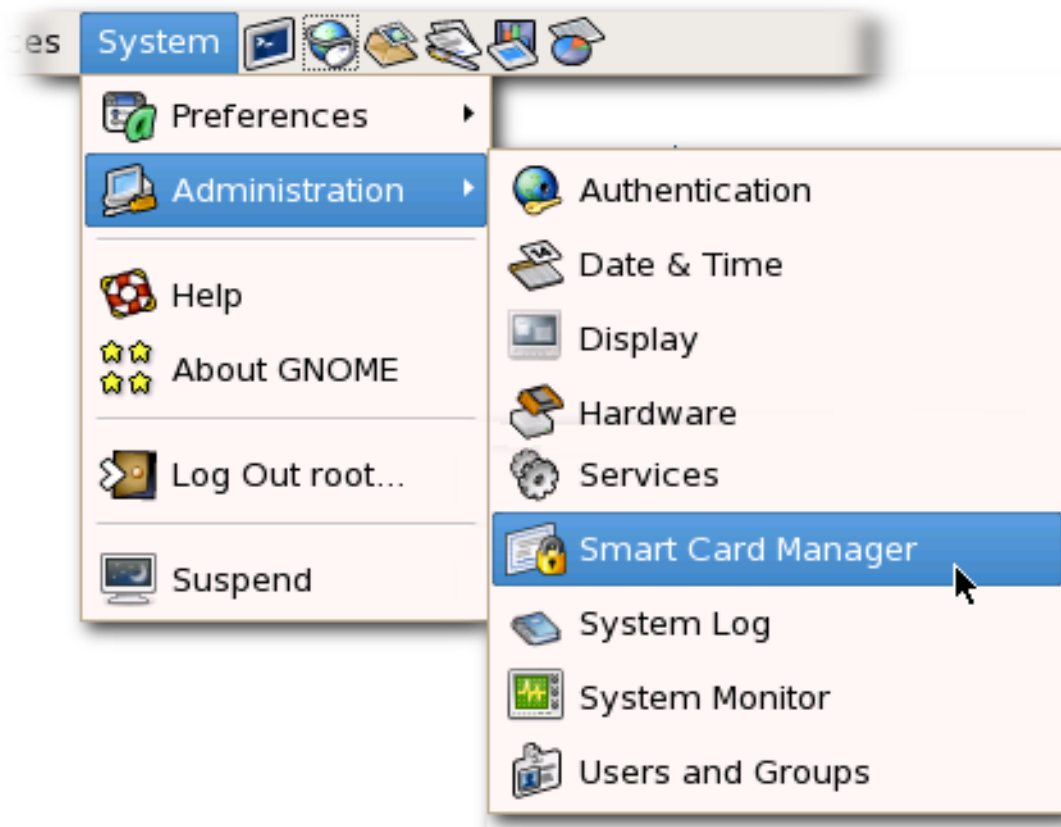
5.2. TPS を使用した SSL 接続の設定

デフォルトでは、TPS は標準の HTTP で Enterprise Security Client と通信します。多くの状況で好ましいことですが、HTTP over SSL (HTTPS) を使用して TPS クライアント通信をセキュアにすることも可能です。

Enterprise Security Client には、TPS 接続を信頼するために TPS の証明書を発行した CA の CA 証明書が必要です。そこから、Enterprise Security Client を設定して TPS の SSL 証明書に接続できます。

1. TPS が使用する CA 証明書をダウンロードします。
 - a. Web ブラウザーで CA のエンドユーザーページを開きます。

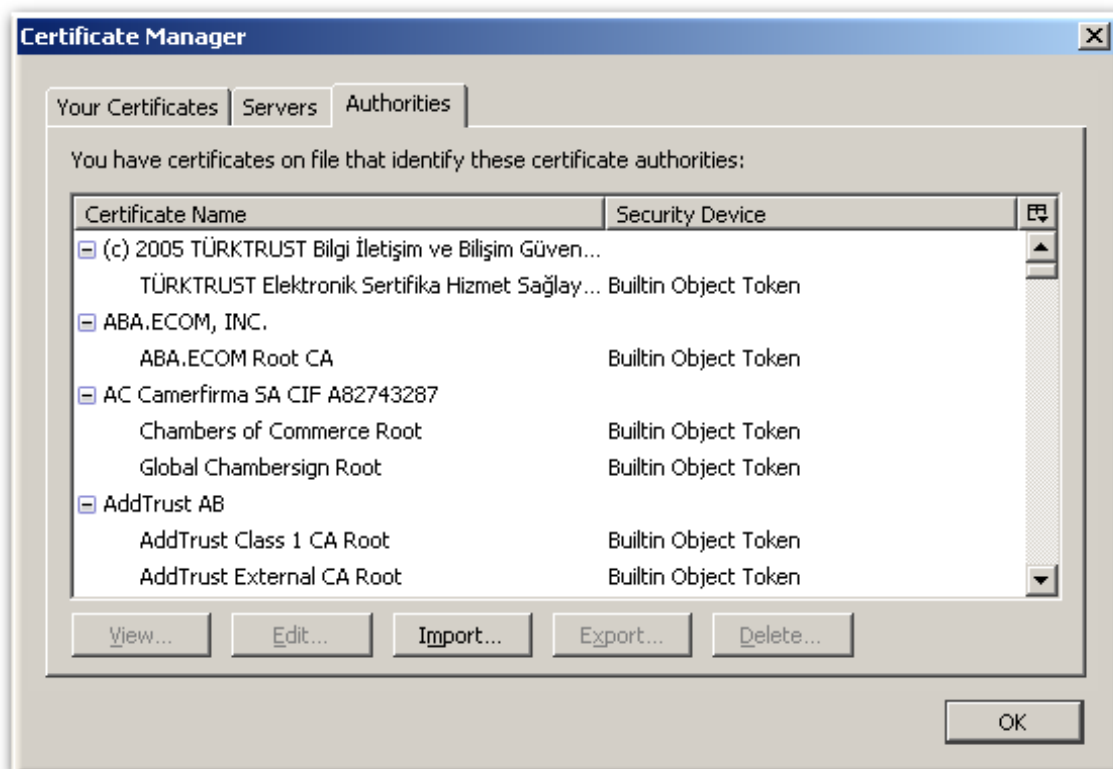
`https://server.example.com:9444/ca/ee/ca/`
 - b. 上部の **Retrieval** タブをクリックします。
 - c. 左側のメニューで **Import CA Certificate Chain** リンクをクリックします。
 - d. チェーンをファイルとしてダウンロードするにはラジオボタンを選択し、ダウンロードしたファイルの場所と名前を書き留めます。
2. Enterprise Security Client を開きます。



3. CA 証明書をインポートします。
 - a. **証明書の表示** ボタンをクリックします。



- b. 認証 タブをクリックします。
- c. インポート をクリックします。



- d. CA 証明書チェーンファイルを参照し、これを選択します。
- e. プロンプトが表示されたら、CA を信頼することを確認します。

- Enterprise Security Client は、SSL 経由で TPS と通信するように設定する必要があります。これは、Enterprise Security Client が TPS の接続に使用するデフォルトである **Phone Home URL** を設定して行います。
- 新しい空のトークンをマシンに挿入します。

空のトークンはフォーマットされていないため、既存の Phone Home URL がないため、URL は手動で設定する必要があります。フォーマットされたトークン (トークンは製造元または IT 部門によってフォーマット可能) にはすでに URL が設定されているため、Phone Home URL の設定を求めるプロンプトは表示されません。

- 新しい TPS URL に SSL ポート情報を入力します。以下に例を示します。

```
https://server.example.com:7890/cgi-bin/home/index.cgi
```

- テスト** ボタンをクリックして、TPS にメッセージを送信します。

リクエストが正常に行われると、クライアントは Phone Home URL が正常に取得されたことを示すダイアログボックスを開きます。

5.3. 共有セキュリティーデータベースの使用

Enterprise Security Client は通常、Enterprise Security Client に関連付けられた各ユーザープロファイルのキーと証明書用に新しい NSS セキュリティーデータベースを作成します。ユーザーが使用する Enterprise Security Client の証明書をインポートまたは信頼するたびに、そのプロファイルの NSS データベースにインポートされます。(これは、Web ブラウザーで、異なるセキュリティーデータベース、パスワードストア、およびプロファイルのブックマークを持つ異なるユーザープロファイルを持つ方法と似ています。)

複数の Enterprise Security Client ユーザーが1台のマシンでクライアントを使用する場合は、インスタンスが存在する可能性があります。この場合、ユーザープロファイルデータベースに加えて、Enterprise Security Client によって信頼される共通の共有セキュリティーデータベースがあることが理にかなっています。この共有セキュリティーデータベースには、TPS が使用する CA 署名証明書など、すべてのユーザーが共通する証明書が含まれます。

共有セキュリティーデータベースの使用は、デフォルトでは設定されていません。

- Enterprise Security Client の停止
- セキュリティーデータベースディレクトリーと、共有されるデータベースを作成します。Enterprise Security Client を設定する前に、データベースが存在し、クライアントが読み取り可能であり、クライアントによって使用される証明書が含まれている必要があります。

NSS データベースは、**certutil** コマンドを使用して作成できます。詳細は、https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS/tools/NSS_Tools_certutil などの **certutil** ドキュメントを参照してください。

- esc-prefs.js** ファイルを開きます。

```
vim /usr/lib/esc-1.1.0/defaults/preferences/esc-prefs.js
```

- 共有データベースを含むディレクトリーを参照する **esc.global.alt.nss.db** パラメーターを追加します。

```
prefs("esc.global.alt.nss.db", "/etc/pki/nssdb");
```

- Enterprise Security Client を再起動して、設定の変更を適用します。

5.4. トークン操作の LDAP 認証の無効化

デフォルトでは、トークン操作を要求する各ユーザーは LDAP ディレクトリーに対して認証されます。ユーザーにエントリーがある場合は、操作が許可されます。ユーザーにエントリーがない場合は、操作が拒否されます。

テスト目的や特定タイプのユーザーの場合は、LDAP 認証を無効にする方がより簡単で適切な方法になります。これは、Enterprise Security Client 設定では設定されませんが、トークン処理システム設定で設定されるため、TPS 管理者によって実行する必要があります。

- TPS サブシステムを停止します。

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

- TPS 設定ファイルを開きます。

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

- 認証パラメーターを **false** に設定します。

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

operation_type は、登録、フォーマット、またはピンリセットなど、LDAP 認証が無効になっているトークン操作です。ある操作タイプで認証を無効にしても、他の操作タイプでは無効になりません。

token_type はトークンプロファイルです。通常のユーザー、セキュリティ担当者、およびセキュリティ担当者が登録したユーザーには、デフォルトのプロファイルがあります。他の種類のユーザーや証明書のカスタムトークンタイプもあります。

たとえば、以下のようになります。

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

- TPS サブシステムを再起動します。

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

TPS 設定の編集は、『[Red Hat Certificate System 9 管理ガイド](#)』で説明されています。

付録A 更新履歴

改訂番号は本ガイドに関するものであり、Red Hat Certificate System のバージョン番号とは関係ありません。

| | | |
|---|------------------------|------------------------|
| 改訂 9.7-0 Red Hat Certificate System 9.7 のガイドを再公開しました。 | Fri Nov 20 2020 | Florian Delehay |
| 改訂 9.5-0 Red Hat Certificate System 9.5 ガイドのリリース | Tue Aug 06 2019 | Marc Muehlfeld |
| 改訂 9.4-0 Red Hat Certificate System 9.4 ガイドのリリース | Thu Oct 25 2018 | Marc Muehlfeld |
| 改訂 9.3-1 バージョン 9.3 『の場合：トークンに対して SSL をサポートするようにブラウザーの設定』を更新(coolkey パッケージを削除) | Tue Apr 10 2018 | Marc Muehlfeld |
| 改訂 9.2-0 Red Hat Certificate System 9.2 GA リリース | Tue Aug 01 2017 | Petr Bokoč |
| 改訂 9.1-1 非同期の更新 | Thu Mar 09 2017 | Petr Bokoč |
| 改訂 9.1-0 Red Hat Certificate System 9.1 リリース | Tue Nov 01 2016 | Petr Bokoč |
| 改訂 8.9-0 ステージングリリース。 | Thu Jul 02 2015 | Tomáš Čapek |