The Scute Manual

Version 1.6.1-beta10
11 August 2020

g10 Code GmbH

This is *The Scute Manual* for Scute version 1.6.1-beta10 and was last updated 11 August 2020. Scute is a PKCS#11 provider on top of GnuPG.

Table of Contents

1 Introduction

Scute is a PKCS #11 implementation for the GnuPG Agent using the GnuPG Smart Card Daemon. Currently, OpenPGP and PIV cards are supported.

Scute enables use of the OpenPGP smart card or a PIV smart card in applications supporting PKCS #11 compliant security tokens. The main application at this time is client authentication in Mozilla-based web browsers. In the future, other applications will be supported.

1.1 Getting Started

This manual documents the Scute module, how it can be used for common applications supported by it, and how it can be extended and improved by programmers. It is thus a user manual as well as a developer manual.

The reader is assumed to possess basic knowledge about cryptography in general, and public key cryptography in particular. The underlying cryptographic engines that are used by the library are not explained, but where necessary, special features or requirements are provided.

This manual can be used in several ways. If read from the beginning to the end, it gives a good introduction into the module and how it can be used in an application. Forward references are included where necessary. Later on, the manual can be used as a reference manual to get just the information needed about any particular application of the module.

1.2 Features

Scute is currently the only implementation of PKCS #11 for the OpenPGP smart card. Apart from that, it offers a couple of other benefits:

it's free software
: Anybody can use, modify, and redistribute it under the terms of the GNU General Public License (see [Library Copying], page 23).

it's built to grow
: Although Scute initially provided a single function, client authentication using OpenPGP smart cards in Mozilla-based web browsers, it was built with the intention of supporting other applications as well in the future.

it's easy
: Building and installing Scute is easy, and preparing smart cards for use with Scute is a snatch using the GnuPG 2 framework. The integration of Scute into the application is seamless.

1.3 Overview

Scute is a security device that implements the PKCS #11 interface for security tokens. Applications which know how to use the PKCS #11 interface to access security tokens for cryptographic operations can use Scute to access the OpenPGP smart card. An important example of such an application is the Firefox web browser by the Mozilla project, which uses the Mozilla Network Security Services library (NSS).

Scute itself does not include a driver for the smart card itself. Instead, it uses the GnuPG 2 framework to access the smart cards and associated data like certificates. Scute acts as the glue between the application and GnuPG 2.

Currently supported usages are client authentication over HTTPS with Firefox (allowing users to authenticate themselves to a remote web service without entering their log-in information), email signing with Thunderbird, and document signing with LibreOffice.

2 Preparation

To use Scute, you first have to install the software. You also have to prepare each card you want to use with Scute before it can be used. Furthermore, you need to configure the application to make use of Scute for cryptographic operations. This chapter explains each of these steps in detail.

2.1 Prerequisites

There are two types of dependencies for Scute: compile-time dependencies and run-time dependencies. The compile-time dependencies only need to be fulfilled when Scute is compiled and installed. The run-time dependencies need to be fulfilled when Scute is used in an application.

Scute depends, in addition to the essential build utilities, on the following packages at build time:

`libgpg-error`
: Scute uses the GnuPG 2 framework for error handling, so it depends on the GPG error library. The minimum version required is 1.38.

`libassuan`
: Scute uses the GnuPG framework for communication with the GPG Agent, so it also depends on the Assuan library. The minimum version required is 2.5.0.

At run-time, in addition to the run-time versions of the above libraries, you also need the following packages installed and configured:

GnuPG
: Scute uses the GnuPG 2 framework to access the OpenPGP card and for certificate management. The minimum version required is 2.2.0. For full functionality, in particular for use with the OpenVPN software, GnuPG version 2.3 is required.

Pinentry
: Pinentry is a dependency of GnuPG 2, so it also needs to be installed with it.

Firefox et al.
: Firefox is the first application supported by Scute. In the future, other applications may be supported. The applications are not dependencies of Scute, but Scute can not be used stand-alone, so you can not experience it without an application.

2.2 Building the Source

Scute does comply to the GNU coding standards and thus can be compiled and installed according to the generic installation instructions found in the source package in the file `INSTALL`. There are no Scute specific options to the configure script.

After installation, the `scute.so` module file can be found in the library directory of the installation path.

2.3 Certificate Preparation

To use an OpenPGP card with Scute, it first has to be initialized by generating or loading a key on the card, see the OpenPGP Card How-To. Then a certificate has to be created and imported into GPGSM. This task involves three steps: First, a certificate signing request (CSR) has to be created that matches the key on the card. This certificate signing request then has to be submitted to a certificate authority (CA), which will create the certificate and send it back to you. At last, the certificate has to be imported into GPGSM. This section will explain all of these steps in detail.

2.3.1 Creating a CSR

Before you start, make sure that the GPG Agent is running, see Section 2.1 [Prerequisites], page 3 and that your card is in the reader. There is no need to configure GPGSM, so you can create a CSR with the command:

```
$ gpgsm --gen-key > floppy-head.csr
Please select what kind of key you want:
   (1) RSA
   (2) Existing key
   (3) Existing key from card
Your selection? 3
```

As we create a certificate for the OpenPGP Card, the option "`[3] Direct from card`" should be selected.

```
Serial number of the card: 355F9746499F0D4B4ECEE4928B007D16
Available keys:
   (1) D53137B94C38D9BF6A199706EA6D5253 OPENPGP.1
   (2) B0CD1A9DFC3539A1D6A8B851A11C8665 OPENPGP.2
   (3) 53DB41052CC590A40B403F3E6350E5DC OPENPGP.3
Your selection? 3
Possible actions for a RSA key:
   (1) sign, encrypt
   (2) sign
   (3) encrypt
Your selection? 2
```

The only operation currently supported is client authentication. For this, the authentication key has to be selected. This is the third key on the card, so the options "`[3] OPENPGP.3`" and "`[2] sign`" should be chosen. Note that the key usage is only advisory, and the CA may assign different capabilities.

```
Enter the X.509 subject name: CN=Floppy Head,OU="Webserver Team",O="Snake Oil, Ltd",L=
Enter email addresses (end with an empty line):
> floppy.head@example.org
>
Enter DNS names (optional; end with an empty line):
>
Enter URIs (optional; end with an empty line):
>
Create self-signed certificate? (y/N) n
```

As a last step, the common name and e-mail address of the key owner need to be specified by you. The above are only an example for a fictious person working at a fictious company. DNS names are only meaningful for server certificates and thus should be left empty.

We have now entered all required information and gpgsm will display what it has gathered and ask whether to create the certificate request:

```
These parameters are used:
    Key-Type: card:OPENPGP.3
    Key-Length: 1024
    Key-Usage: sign
    Name-DN: CN=Floppy Head,OU="Webserver Team",O="Snake Oil, Ltd",L="Snake Town",ST="
    Name-Email: floppy.head@example.org

Proceed with creation? (y/N) y
Now creating certificate request.  This may take a while ...
gpgsm: about to sign the CSR for key: &53DB41052CC590A40B403F3E6350E5DC
```

GPGSM will now start working on creating the request. During this time you will be asked once for a passphrase to unprotect the authentication key on the card. A pop up window will appear to ask for it.

When it is ready, you should see the final notice:

```
gpgsm: certificate request created
Ready.  You should now send this request to your CA.
```

Now, you may look at the created request:

```
$ cat floppy-head.csr
-----BEGIN CERTIFICATE REQUEST-----
MIICCDCCAXECAQAwgYExCzAJBgNVBAYTAlhZMRUwEwYDVQQIEwxTbmFrZSBEZXNl
cnQxEzARBgNVBAcTClNuYWtlIFRvd24xFzAVBgNVBAoTDlNuYWtlIE9pbCwgTHRk
MRcwFQYDVQQLEw5XZWJzZXJ2ZXIgVGVhbTEUMBIGA1UEAxMLRmxvcHB5IEhlYWQw
gaAwDQYJKoZIhvcNAQEBBQADgY4AMIGKAoGBANWaM9YS89AOx3GX1Rua+4DUHwbL
wt0rBYdBddlabMMteVjUcOOhbFMirLpLAi1S8fUXNiy84ysOmFStmvSIXDsAgXq5
1ESOU4SNg2zEkPDF1WYJ5BFIXdYq9i2k5W7+ctV8PkKv3e5IeYXTa5qppIPD31de
gM8Qj7tKOhL/eNCfAgQAAQABoEUwQwYJKoZIhvcNAQkOMTYwNDAiBgNVHREEGzAZ
gRdmbG9wcHkuaGVhZEBleGFtcGxlLmNvbTAOBgNVHQ8BAf8EBAMCBsAwDQYJKoZI
hvcNAQEFBQADgYEAFC9q6+ib9YGCLB/2AlZR+/dvb+pEeXR1EbpV/dw/gjP1yPY6
29n8ZIDLUvQvNCtfCcXFxFimVSSB/KmFXXsJbM+NXQyT6Ocn34iHmkf9IVRMWQWg
ZBYfQVeXAd7XlxI6d1wXDLwD/26lTU/rH2JU6H1+zSfZxqwVC4Iu+kiN4Y8=
-----END CERTIFICATE REQUEST-----
$
```

2.3.2 Signing the CSR

The next step is to submit this certificate request to the CA, which can then create a certificate and send it back to you.

If, for example, you use the CA CAcert, then you can log into your account at the CAcert website, choose "Client Certificates -> New", check "Show advanced options", paste the above request block into the text field and click on "Submit". If everything works correctly, a certificate will be shown, which you can cut and paste into a new file `floppy-head.crt`.

Alternatively if, for example, you set up your own CA with OpenSSL, then you can create your own certificate by issueing a command similar `openssl ca -in floppy-head.csr -cert snakeoil-ca-rsa.crt -keyfile snakeoil-ca-rsa.key -out floppy-head.crt`. Please see the OpenSSL documentation for more details on how to set up and administrate a certificate authority infrastructure.

2.3.3 Importing the Certificate into GPGSM

Once the CSR has been signed, you should end up with a certificate file `floppy-head.crt`, which you then have to import into GPGSM. It is also recommended that you import the root certificate of the CA first in the same fashion.

```
$ gpgsm --import floppy-head.crt
gpgsm: certificate imported

gpgsm: total number processed: 1
gpgsm:               imported: 1
```

gpgsm tells you that it has imported the certificate. It is now associated with the key you used when creating the request. To see the content of your certificate, you may now enter:

```
$ gpgsm -K Floppy
/home/foo/.gnupg/pubring.kbx
---------------------------
Serial number: 10
       Issuer: /CN=Snake Oil CA/OU=Certificate Authority/O=Snake Oil, Ltd/L=Snake Town
      Subject: /CN=Floppy Head/OU=Webserver Team/O=Snake Oil, Ltd/ST=Snake Desert/C=XY
     validity: 2006-11-11 14:09:12 through 2007-11-11 14:09:12
     key type: 1024 bit RSA
  fingerprint: EC:93:A2:55:C6:58:7F:C9:9E:96:DB:12:6E:64:99:54:BB:E1:94:68
```

The option "`-K`" is used above because this will only list certificates for which a private key is available. To see more details, you may use "`--dump-secret-keys`" instead of "`-K`".

2.3.4 Loading the Certificate onto the Card

This step is optional. You may choose to store the certificate directly into your OpenPGP card. The benefit of doing so is that Scute will then be able to fetch the certificate from the card without having to look into the GPGSM store.

You need your certificate in the DER format. Export it from the GPGSM store with the following command:

```
$ gpgsm -o floppy-head.crt --export Floppy
```

Then, fire up the GnuPG card editor to transfer the certificate to the card (note that the `writecert` command is not listed in the editor's online help):

```
$ gpg2 --card-edit

Application ID ...: D2760001230102000000500001234000
[...]

gpg/card> admin
```

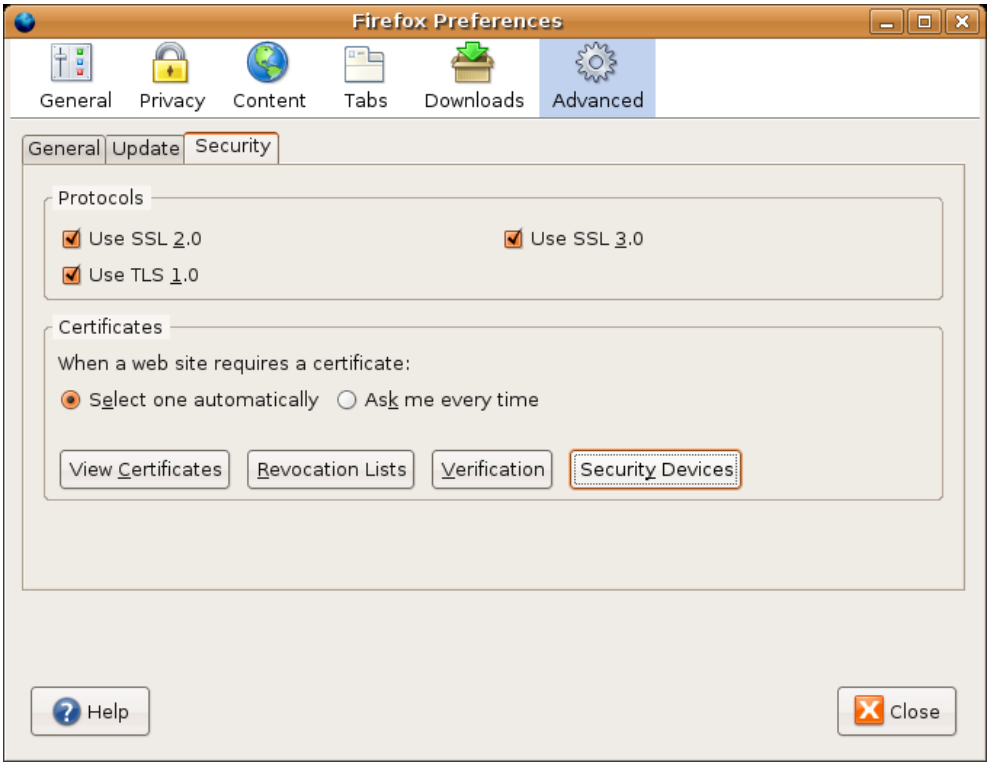
```
Admin commands are allowed

gpg/card> writecert 3 < floppy-head.crt

gpg/card> quit
```

3 Client Authentication

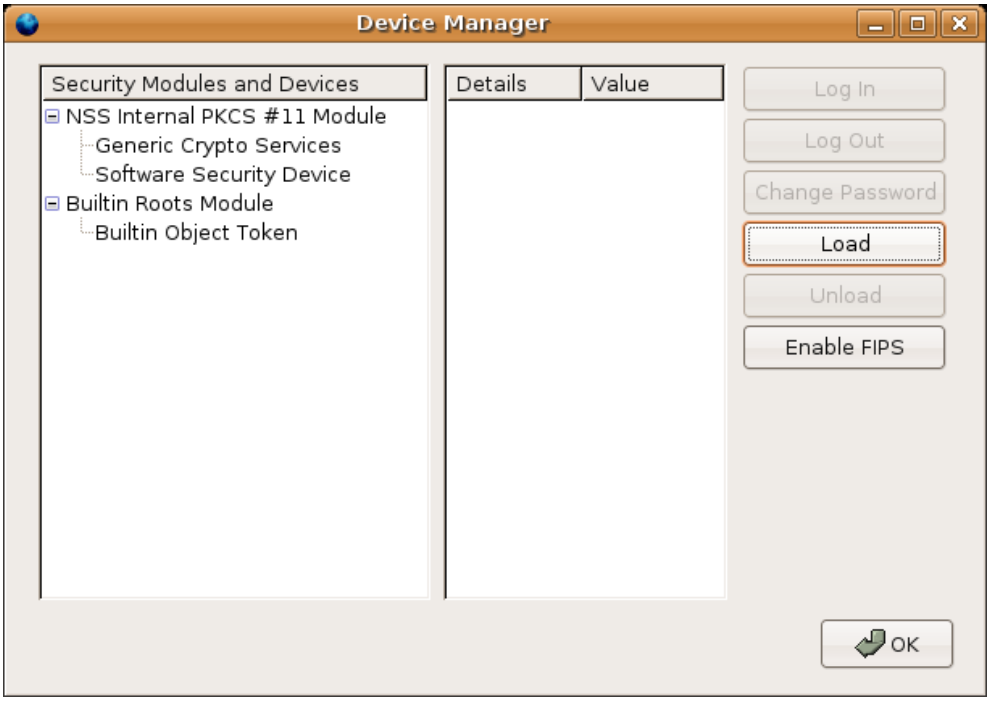
Scute allows you to authenticate yourself to a website securely without entering a username or password by simply using your OpenPGP card. Currently, only Mozilla-based browsers like Firefox are supported, although other applications using Mozilla NSS or supporting PKCS #11 modules may work.

3.1 Application Configuration

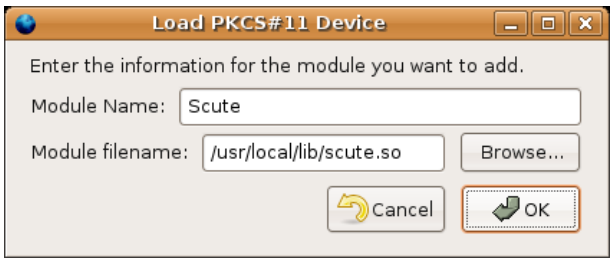
To prepare your application for use with Scute, you have to load the Scute module as a PKCS #11 module into the application. With Firefox, this can be done by choosing `Edit->Preferences` in the menu. In the preferences configuration dialog, you should select the `Advanced` configuration section, then the `Security` tab, and then select `Security Devices` in the category `Certificates`.



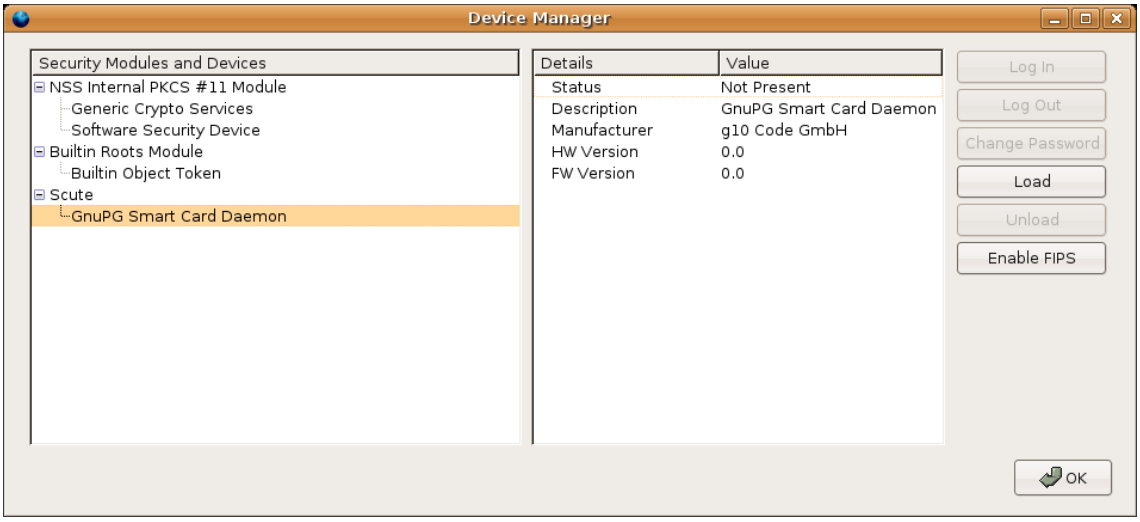
In the devices manager dialog, you can select `Load` to load a new PKCS #11 device.



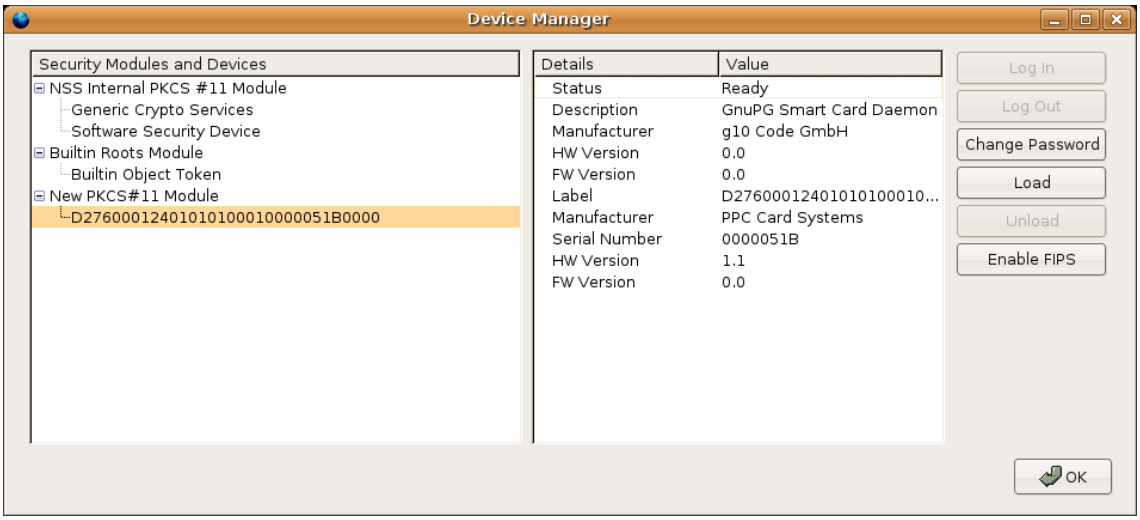
In the pop-up dialog that follows, you can give a module name (e.g. "`Scute`") and a module filename. The latter should correspond to the full file name of the installed Scute module file `scute.so`. The default installation path is `/usr/local/lib`, which would mean that you have to provide the file name `/usr/local/lib/scute.so`. If you or your system administrator installed Scute in a different location, you have to adjust the file name correspondingly.



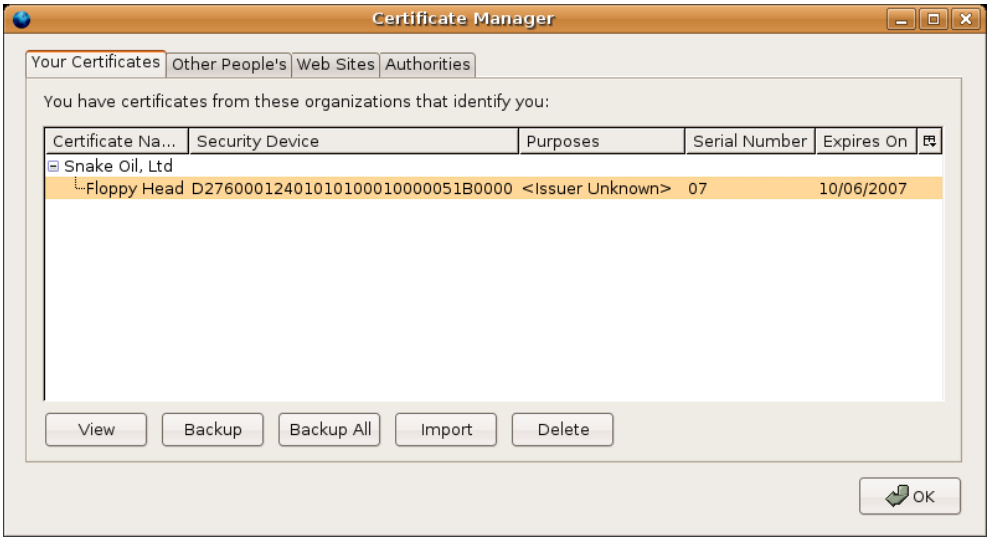
After confirming installation of the security device, a pop-up window should confirm that the module was successfully loaded, and an entry for the security device should appear in the device manager list of `Security Modules and Devices`.



When you insert the OpenPGP card for which you generated and imported a certificate earlier (see Section 2.3 [Certificate Preparation], page 4), the device manager should detect this security token and display some information about it in the `Details` list when you select it from the module list.



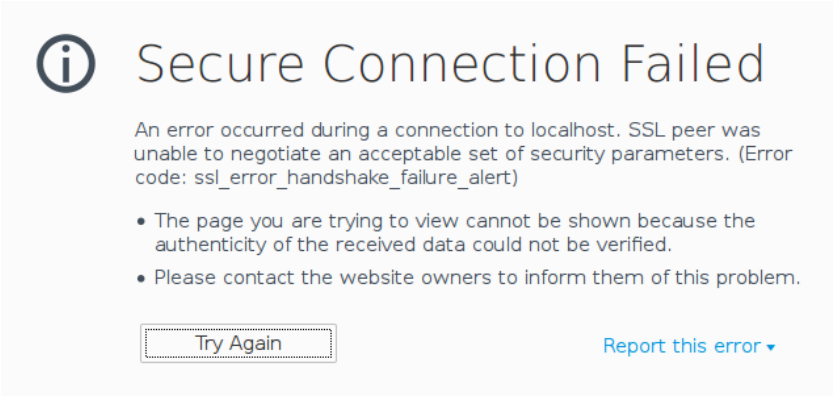
The client certificate will show up in the `Certificate Manager` under `Your Certificates`:



3.2 Authentication With Service

Before you access a web service which requires client authentication, for instance a fictious web service `https://example.com`, the OpenPGP card should be present. In this case, a pop-up window will appear that requests you to enter the PIN number protecting the authentication key on the OpenPGP card. After entering the PIN number, your browser will be authenticated to the server. If the server accepts your request and certificate, this is all which is required. You should leave the card in the reader as long as the connection persists. Depending on how aggressively GPG Agent caches your PIN number, you may have to enter the PIN number again later to keep up the connection to the server.

If the card is not present, or you enter the wrong PIN, or the server does not admit your certificate, you will get an error message. This error message is generated by the application and Scute can not influence it. Unfortunately, in Firefox (at least up to version 38.5.0), this error message is not very user friendly. For example, entering a bad PIN results in the following generic error message, and the `Try Again` button does not work as expected:

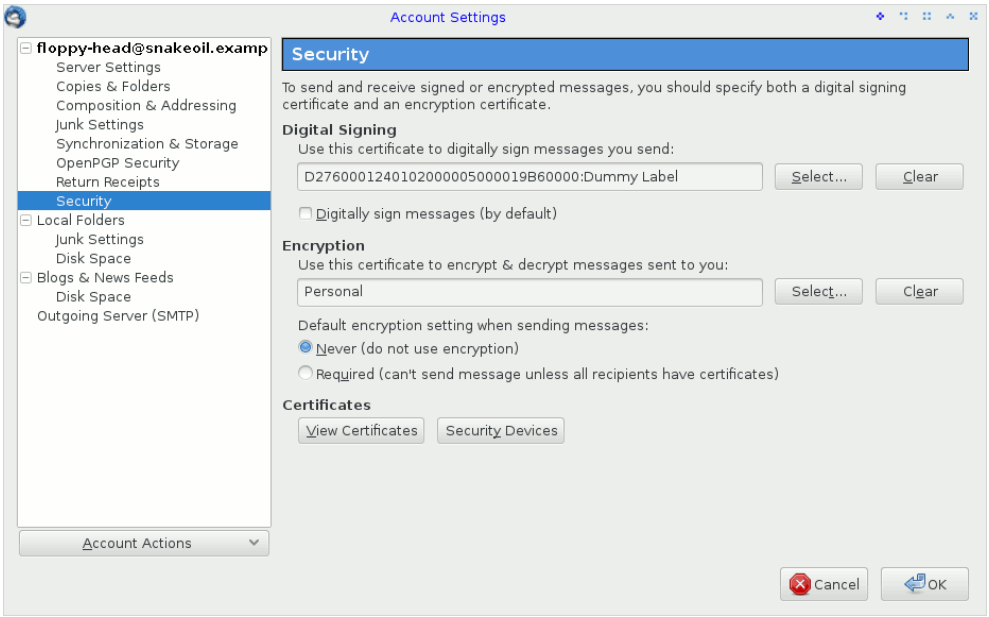


4 Email Signing

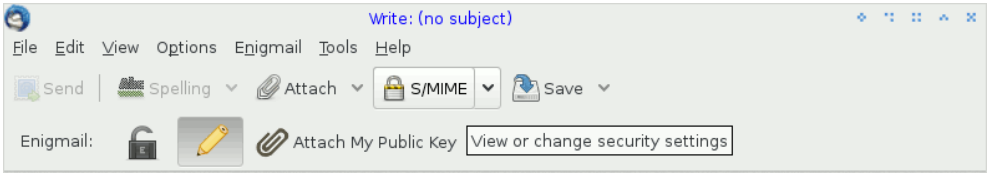
Scute also allows you to use your card-based X.509 certificate to sign your emails with the S/MIME signature format. This has been tested with Mozilla Thunderbird only, but should work with any mail client with support for PKCS #11 (notably GNOME Evolution).

You must first load the Scute module into your mail client. With Mozilla Thunderbird, the procedure is the same as the one described above for Mozilla Firefox.

Then, open your account configuration dialog (`Edit->Account Settings`), and in the `Security` tab, under the section `Digital Signing`, use the `Select...` button to associate your card-based certificate with your account.



When writing a new message, you may then use the `S/MIME` button and select `Digitally sign this message` in the popup menu. You will be prompted for your User PIN before the message is sent.



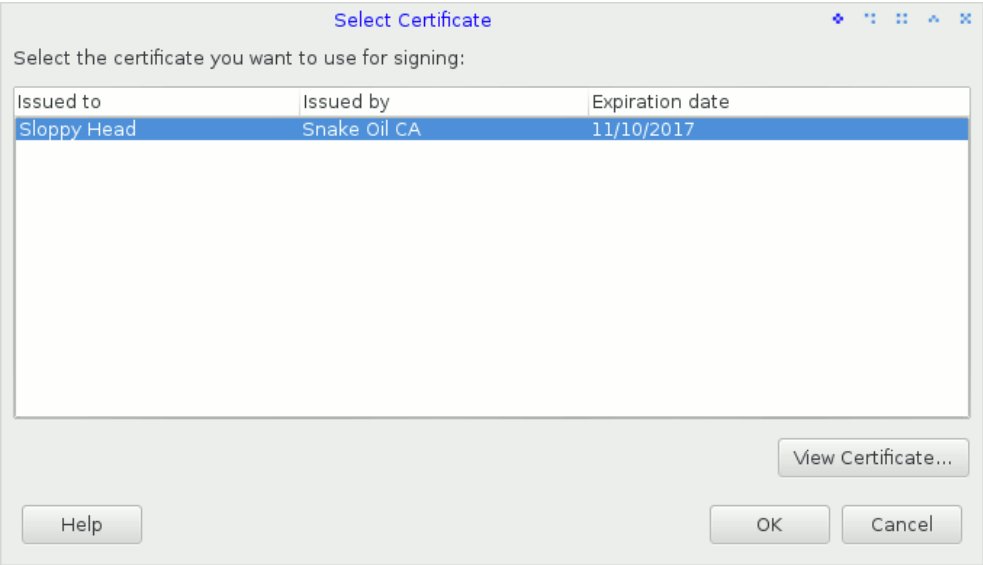
5 Document Signing

Scute can also be used with LibreOffice to sign OpenDocument files.

First, you must load the Scute module into Mozilla Firefox according to the above procedure. Then, configure LibreOffice to use Firefox's certificate store by defining the `MOZILLA_CERTIFICATE_FOLDER` environment variable to your Firefox profile directory.

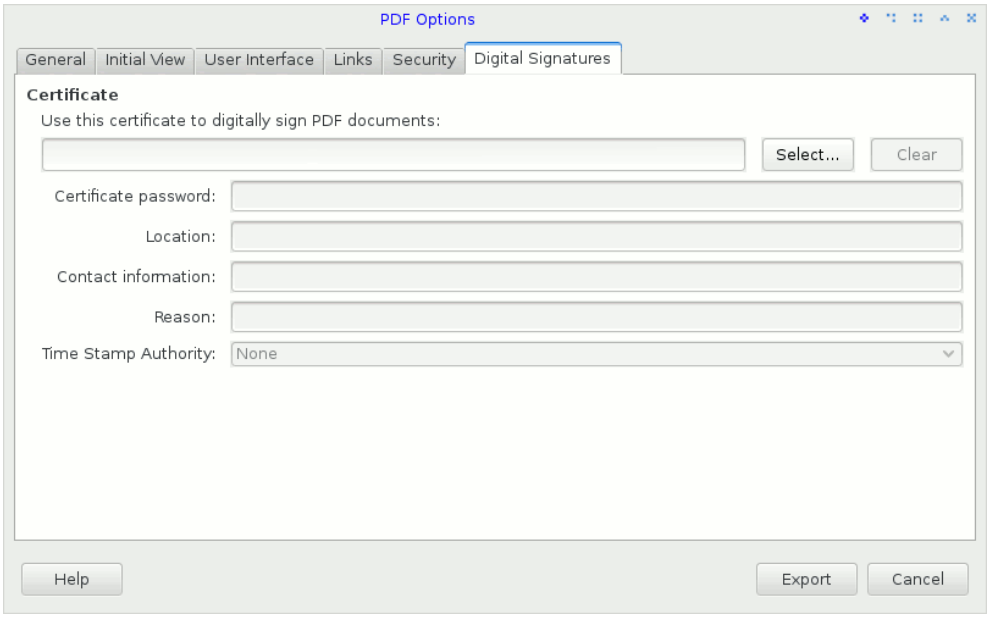
Then, to sign the document you are editing, select the `File->Digital Signatures...` menu option to open the `Digital Signatures` dialog.

Click the `Sign Document` button to open the certificate selection dialog. Select your card-based certificate, then validate. Enter your User PIN when prompted by GPG Agent.



You may also sign a PDF export of your document. Select the `File->Export as PDF...` menu option to open the `PDF Options` dialog. In the `Digital Signatures` tab, use the

`Select` button to open the certificate selection dialog as above. You will be prompted for your User PIN when you will click the `Export` button.



6 Troubleshooting

Symptom: Loading the Scute security device in the security device manager of Firefox fails with "Unable to load module".

Solution: Make sure that Scute is correctly installed, and that all libraries and executables are available. If you are using GnuPG 2.0 (instead of 2.1), you may need to make sure that the GPG Agent is running and can be found via the environment variable `GPG_AGENT_INFO`. See Section "Invoking GPG-AGENT" in *Using the GNU Privacy Guard*, for details on how to run the GPG Agent.

Symptom: Client authentication fails with "<example.com> has received an incorrect or unexpected message. Error code: -12227".

Solution: Make sure that the correct OpenPGP card is inserted and the certificate available in GPGSM. Check that the OpenPGP card is detected correctly in the security device manager and the corresponding certificate is displayed in the certificate manager of Firefox. See Section 3.2 [Authentication With Service], page 12.

Symptom: The OpenPGP card is detected and displayed in the security device manager in Firefox, but no corresponding certificate is displayed in the certificate manager of Firefox.

Solution: Make sure that the corresponding certificate is imported in GPGSM.

7 Internals

The following notes are intended for people interested in more technical details about Scute and its implementation. They give an overview about its scope and potential compatibility issues with applications.

7.1 Features and Limitations

Scute implements version 2.20 of the PKCS #11 specification.

The OpenPGP smart card application is supported in read-only mode.

The following functions are not supported:

`C_Initialize`
: No support for native thread package. Locking callbacks must be provided if multi-threaded operation is desired.

`C_WaitForSlotEvent`
: Not implemented. The interface as specified by PKCS #11 is broken anyway, as the function can not safely be canceled. Thus, we require polling.

`C_GetOperationState`
`C_SetOperationState`
: Not supported.

`C_InitToken`
`C_InitPIN`
`C_SetPIN` Not supported. No write operations are allowed. To configure the token, please use the tools accompanying the GnuPG software suite.

`C_Login`
`C_Logout` Not supported. No login into the token by the software is required. Passphrase queries are implemented by the use of GPG Agent and Pinentry.

`C_EncryptInit`
`C_Encrypt`
`C_EncryptUpdate`
`C_EncryptFinal`
`C_DigestInit`
`C_Digest`
`C_DigestUpdate`
`C_DigestKey`
`C_DigestFinal`
`C_VerifyInit`
`C_Verify`
`C_VerifyUpdate`
`C_VerifyFinal`
`C_VerifyRecoverInit`
`C_VerifyRec`
: Not supported. Only secret key operations are supported.

`C_DecryptInit`
`C_Decrypt`
: Not yet supported, but will be in the future.

`C_SignUpdate`
`C_SignFinal`
`C_DecryptUpdate`
`C_DecryptFinal`
: No progressive crypto-operations are supported.

`C_SignRecoverInit`
`C_SignRecover`
: Not supported.

`C_DigestEncryptUpdate`
`C_DecryptDigestUpdate`
`C_SignEncryptUpdate`
`C_DecryptVerifyUpdate`
: Dual-purpose cryptographic functions are not supported.

`C_GenerateKey`
`C_GenerateKeyPair`
`C_WrapKey`
`C_UnwrapKey`
`C_DeriveKey`
: Key management functions are not supported. Please use the tools accompanying the GnuPG software suite to generate and import keys for use with the token.

`C_SeedRandom`
: Not supported.

`C_CreateObject`
`C_CopyObject`
`C_DestroyObject`
`C_SetAttributeValue:`
: Only read-only operations are supported on objects.

`C_GetObjectSize`
: Not supported.

`CKO_CERTIFICATE`
: The label specifies the key on the card used (e.g. `OPENPGP.3`). The ID is the fingerprint.

`CKO_PRIVATE_KEY:`
: The `CKA_LOCAL` attribute can not be supported by the OpenPGP card. It is always set to false (as the key on the card may be copied to the card from an external source).

7.2 Developing Scute

Scute is single-threaded. There is a global lock that is taken in all entry points of Scute, except for `C_Initialize`, `C_Finalize`, `C_GetFunctionList`, and stubs.

Here are a couple of hints on how to develop PKCS #11 modules for Mozilla:

`libopensc2` ships with a `pkcs11-spy` library that can be loaded as a wrapper around the PKCS #11 library you want to use to log all functions invoked by Mozilla. Here is how to use it:

Set the `PKCS11SPY_OUTPUT` environment variable to a filename. `pkcs11-spy` appends its log messages at the end of this file. Set the `PKCS11SPY` environment variable to the filename of the PKCS #11 module you actually want to use. Start Mozilla within this environment.

There is a different, probably more powerful way to debug Mozilla PKCS #11 libraries. However, to be able to use it, you need to configure and compile the Mozilla NSS sources with `--enable-debug`. Instructions can be found at: https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS/nss_tech_notes

Here are a couple of links to more information about implementing a PKCS #11 module for Mozilla:

https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS/PKCS11_Implement
: Guidelines for implementors of PKCS #11 modules targeting Mozilla

http://www-archive.mozilla.org/projects/security/pki/pkcs11/
: PKCS #11 Conformance Testing

https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS
: The Mozilla NSS web page

7.3 Mozilla Compatibility

Mozilla has a bug that causes the wrong security device to be unloaded when unloading a security device. Also, the displayed list becomes corrupt. When closing and reopening the security device manager, the list displayed is correct, but in anyway the wrong security module is unloaded.

Library Copying

Version 2.1, February 1999

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program

by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the *Lesser* General Public License because it does *Less* to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

   A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
   a. The modified work must itself be a software library.
   b. You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
   c. You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
   d. If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

      (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply

to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

   Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

   This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

   If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

   However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

   When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

   You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

   a. Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

   b. Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

   c. Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

   d. If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

   e. Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

   For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components

(compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b. Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN

WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> *one line to give the library's name and an idea of what it does.*
> Copyright (C) *year name of author*
>
> This library is free software; you can redistribute it and/or modify it
> under the terms of the GNU Lesser General Public License as published by
> the Free Software Foundation; either version 2.1 of the License, or (at
> your option) any later version.
>
> This library is distributed in the hope that it will be useful, but
> WITHOUT ANY WARRANTY; without even the implied warranty of
> MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
> Lesser General Public License for more details.
>
> You should have received a copy of the GNU Lesser General Public
> License along with this library; if not, write to the Free Software
> Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307,
> USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the library
> 'Frob' (a library for tweaking knobs) written by James Random Hacker.
>
> *signature of Ty Coon*, 1 April 1990
> Ty Coon, President of Vice

That's all there is to it!