

nShield Security World

API introductory guide

03 September 2026

© Entrust Corporation. All rights reserved.

Table of Contents

1. Additional useful documentation . . . 2
2. Security World Software default directories . . . 3
    2.1. Utility help options . . . 4
    2.2. Further information . . . 4
    2.3. Security advisories . . . 5
    2.4. Contacting Entrust nShield Support . . . 5
3. nShield architecture . . . 6
    3.1. Security World Software modules . . . 6
    3.2. Security World Software server . . . 6
    3.3. Stubs and interface libraries . . . 7
    3.4. Using an interface library . . . 7
    3.5. Writing a custom application . . . 8
    3.6. Acceleration-only or key management . . . 8

Entrust provides the following toolkits and libraries, and supporting documentation, to help developers write applications that use nShield modules:

- nShield PKCS #11 library
- Microsoft CryptoAPI (MSCAPI) toolkit
- Microsoft Cryptography API: Next Generation (CNG) toolkit
- nCipherKM JCA/JCE cryptographic service provider toolkit

These toolkits, like the application plug-ins supplied by Entrust, use the Security World para digm for key storage. For an introduction to Security Worlds, see nShield Security World v13.9.6 Management Guide.

The available documentation helps you to build an application that uses an nShield key-man agement module to accelerate cryptographic operations and protect cryptographic keys through a standard interface rather than the full nCore API.

These guides assume that you are familiar with the concept of the Security World. They are intended for experienced programmers and assume that you are also familiar with the fol-lowing documentation:

- The nCore Developer Tutorial, which describes how to write applications using an nShield module.
- The *nCore API Documentation* (supplied as HTML with the product, see Further infor-mation), which describes the nCore API.

This introductory document provides basic information about the toolkits and the Security World software archictecture. It is designed to be used alongside the toolkit-specific guides.

1. Additional useful documentation

Refer to nShield Security World v13.9.6 Management Guide and nShield v13.9.6 HSM User Guide for additional information about Security Worlds and nShield HSMs.

2. Security World Software default directories

The default locations for Security World Software and program data directories on English-language systems are summarized in the following table:

Directory Name	Environment Variable	Windows Server 2016	Linux
nShield Installation	NFAST_HOME	C:\Program Files\nCipher\nfast	/opt/nfast/
Key Management Data	NFAST_KMDATA	C:\ProgramData\nCipher\Key Management Data	/opt/nfast/kmdata/
Dynamic Feature Certificates	NFAST_CERTDIR	C:\ProgramData\nCipher\Feature Certificates	/opt/nfast/femcerts/
Static Feature Certificates		C:\ProgramData\nCipher\Features	/opt/nfast/kmdata/features/
Log Files	NFAST_LOGDIR	C:\ProgramData\nCipher\Log Files	/opt/nfast/log/



By default, the Windows %NFAST_KMDATA% directories are hidden directories. To see these directories and their contents, you must enable the display of hidden files and directories in the **View** settings of the **Folder Options**.



Dynamic feature certificates must be stored in the directory stated in the default directories table.

The directory shown for static feature certificates is an example location. You can store those certificates in any directory and provide the appropriate path when using the Feature Enable Tool. However, you must not store static feature certificates in the dynamic features certificates directory. For more information about feature certificates, see Optional features.

The absolute paths to the Security World Software installation directory and program data directories on Windows platforms are stored in the indicated nShield environment variables at the time of installation. If you are unsure of the location of any of these directories, check the path set in the environment variable.

The instructions in this guide refer to the locations of the software installation and program data directories by their names (for example, Key Management Data) or:

- For Windows, nShield environment variable names enclosed in percent signs (for example, %NFAST_KMDATA%).

- For Linux, absolute paths (for example, `/opt/nfast/kmdata/`).

`NFAST_KMDATA` cannot be a symbolic link.

If the software has been installed into a non-default location:

- For Windows, ensure that the associated nShield environment variables are re-set with the correct paths for your installation.
- For Linux, you must create a symbolic link from `/opt/nfast/` to the directory where the software is actually installed. For more information about creating symbolic links, see your operating system's documentation.

2.1. Utility help options

Unless noted, all the executable utilities provided in the `bin` subdirectory of your nShield installation have the following standard help options:

`-h|--help` displays help for the utility

`-v|--version` displays the version number of the utility

`-u|--usage` displays a brief usage summary for the utility.

2.2. Further information

The API guides form one part of the information and support provided by Entrust.

The *nCore API Documentation* is supplied as HTML files installed in the following locations:

- Windows:
  - API reference for host: `%NFAST_HOME%\document\ncore\html\index.html`
  - API reference for SEE: `%NFAST_HOME%\document\csddoc\html\index.html`
- Linux:
  - API reference for host: `/opt/nfast/document/ncore/html/index.html`
  - API reference for SEE: `/opt/nfast/document/csddoc/html/index.html`

The Java Generic Stub classes, nCipherKM JCA/JCE provider classes, and Java Key Manage ment classes are supplied with HTML documentation in standard Javadoc format, which is installed in the appropriate `nfast\java` or `nfast/java` directory when you install these classes.

2.3. Security advisories

If Entrust becomes aware of a security issue affecting nShield HSMs, Entrust will publish a security advisory to customers. The security advisory will describe the issue and provide rec ommended actions. In some circumstances the advisory may recommend you upgrade the nShield firmware and or image file. In this situation you will need to re-present a quorum of administrator smart cards to the HSM to reload a Security World. Because of this, you should consider the procedures and actions required to upgrade devices in the field when deploying and maintaining your HSMs.



The Remote Administration feature supports remote firmware upgrade of nShield HSMs, and remote ACS card presentation.

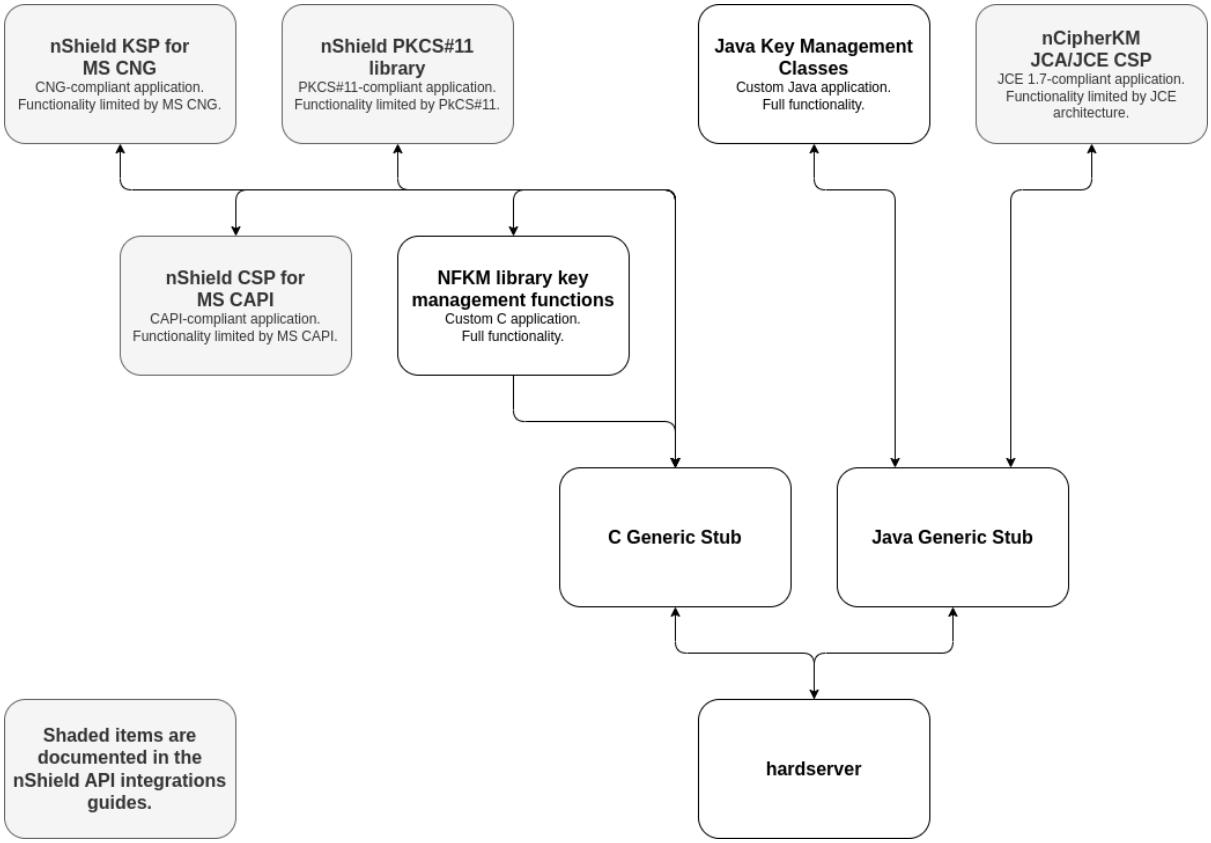
We recommend that you monitor the Announcements & Security Notices section on Entrust nShield, https://trustedcare.entrust.com/, where any announcement of nShield Security Advisories will be made.

2.4. Contacting Entrust nShield Support

To obtain support for your product, contact Entrust nShield Support, https://trustedcare.entrust.com/.

3. nShield architecture

This chapter provides a brief overview of the Security World Software architecture. The following diagram provides a visual representation of nShield architecture and the documentation that relates to it.



3.1. Security World Software modules

nShield modules provide a secure environment to perform cryptographic functions. Key-management modules are fitted with a smart card interface that enables keys to be stored on removable tokens for extra security. nShield modules are available for PCI buses and also as network-attached Ethernet modules (nShield Connect).

3.2. Security World Software server

The Security World Software server, often referred to as the `hardserver`, accepts requests by means of an interprocess communication facility (for example, a domain socket on Linux or named pipes or TCP/IP sockets on Windows).

The Security World Software server receives requests from applications and passes these

to the nShield module(s). The module handles these requests and returns them to the server. The server ensures that the results are returned to the correct calling program.

You only need a single Security World Software server running on your host computer. This server can communicate with multiple applications and multiple nShield modules.

3.3. Stubs and interface libraries

An application can either handle its own cryptographic functions or it can use a cryptographic library:

- If the application uses a cryptographic library that is already able to communicate with the Security World Software server, then no further modification is necessary. The application can automatically make use of the nShield module.
- If the application uses a cryptographic library that has not been modified to be able to communicate with the Security World Software server, then either Entrust or the cryptographic library supplier need to create adaption function(s) and compile them into the cryptographic library. The application users then must relink their applications using the updated cryptographic library.

If the application performs its own cryptographic functions, you must create adaption function(s) that pass the cryptographic functions to the Security World Software server. You must identify each cryptographic function within the application and change it to call the nShield adaption function, which in turn calls the generic stub. If the cryptographic functions are provided by means of a DLL or shared library, the library file can be changed. Otherwise, the application itself must be recompiled.

3.4. Using an interface library

Entrust supplies the following interface libraries:

- Microsoft Cryptography API: Next Generation (CNG)
- Microsoft CryptoAPI (CAPI)
- PKCS #11
- nCipherKM JCA/JCE CSP

Third-party vendors may supply nShield-aware versions of their cryptographic libraries.

The functionality provided by these libraries is the intersection of the functionality provided by the nCore API and the functionality provided by the standard for that library.

Most standard libraries offer fewer key-management options than are available in the nCore API. However, the nShield libraries do not include any extensions to their standards. If you want to make use of features of the nCore API that are not offered in the standard, you should convert your application to work directly with the generic stub.

On the other hand, many standard libraries include functions that are not supported on the nShield module, such as support for IDEA or Skipjack. If you require a feature that is not sup ported on the nShield module, contact Support because it may be possible to add the fea- ture in a future release. However, in many cases, features are not present on the module for licensing reasons, as opposed to technical reasons, and Entrust cannot offer them in the interface library.

3.5. Writing a custom application

If you choose not to use one of the interface libraries, you must write a custom application. This gives you access to all the features of the nCore API. For this purpose, Entrust provides generic stub libraries for C and Java. If you want to use a language other than C orJava, you must write your own wrapper functions in your chosen programming language that call the C generic stub functions.

Entrust supplies several utility functions to help you write your application.

3.6. Acceleration-only or key management

You must also decide whether you want to use key management or whether you are writing an acceleration-only application.

Acceleration-only applications are much simpler to write but do not offer any security bene fits.

The Microsoft CryptoAPI, Java JCE, PKCS #11, as well as the application plug-ins, use the Security World paradigm for key storage.

If you are writing a custom application, you have the option of using the Security World mechanisms, in which case your users can use the command-line utilities supplied with the module for many key-management operations. This means you do not have to write these functions yourself.

The NFKM library gives you access to all the Security World functionality.