nShield Security World

Microsoft CNG Guide for nShield Security World v13.9.0

28 October 2025

Table of Contents

1. Introduction

Cryptography API: Next Generation (CNG) is the successor to the Microsoft Crypto API (CAPI) and its long-term replacement. The Security World Software implementation of Microsoft CNG is supported on Microsoft Windows Server 2016 and later releases. The nShield CNG providers offer the benefits of hardware-based encryption accessed through the standard Microsoft API, and support the National Security Agency (NSA) classified Suite B algorithms.

Before using the nShield CNG providers, run the nShield CNG Configuration Wizard to:

- configure HSM Pool mode for CNG as required.
- create a new Security World or specify an existing Security World to use.
- register the nShield CNG providers.
- configure the nShield CNG providers as default CNG providers for specific tasks.

For additional information, see the Microsoft CNG documentation: http://msdn2.microsoft.com/en-us/library/aa376210.aspx.

This guide describes the Microsoft Cryptography API: Next Generation (CNG) toolkit supplied by Entrust Security to help developers write applications that use nShield modules.

This toolkit, like the application plug-ins supplied by Entrust, uses the Security World paradigm for key storage. For an introduction to Security Worlds, see nShield Security World v13.9.0 Management Guide.

1.1. Read this guide if...

Read this guide if you want to build an application that uses an nShield key-management module to accelerate cryptographic operations and protect cryptographic keys through a standard interface rather than the full nCore API.

This guide assumes that you are familiar with the concept of the Security World. It is intended for experienced programmers and assumes that you are familiar with the following documentation:

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

1.2. Additional useful documentation

Refer to nShield Security World v13.9.0 Management Guide and nShield v13.9.0 HSM User Guide for additional information about Security Worlds and nShield HSMs.

1.3. 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.

1.4. 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.

1.5. Further information

This guide forms 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.

1.6. 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.

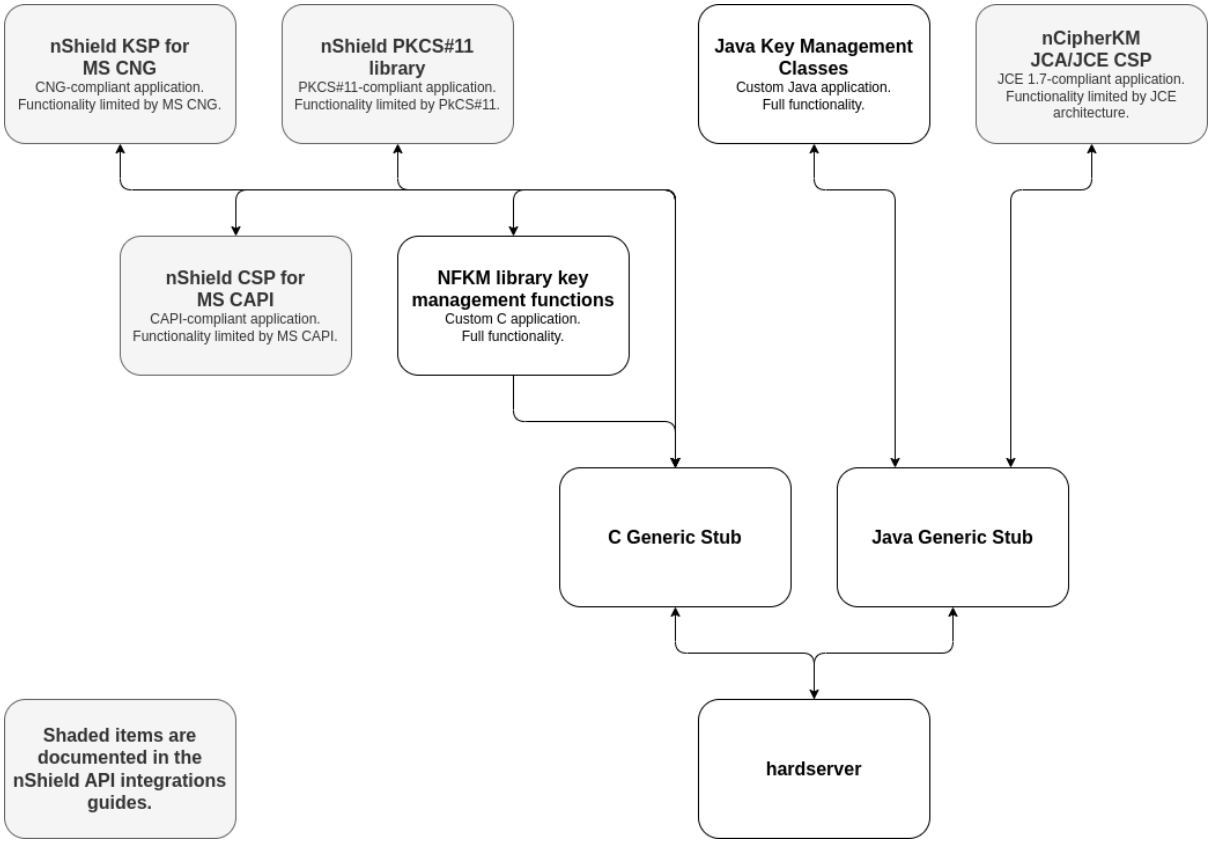
We recommend that you monitor the Announcements & Security Notices section on Entrust nShield, https://trustedcare.entrust.com/, where any announcement of nShield Security Advisories will be made.

1.7. Contacting Entrust nShield Support

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

2. 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.



2.1. 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).

2.2. 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.

2.3. 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.

2.4. 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.

2.5. 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.

2.6. 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.

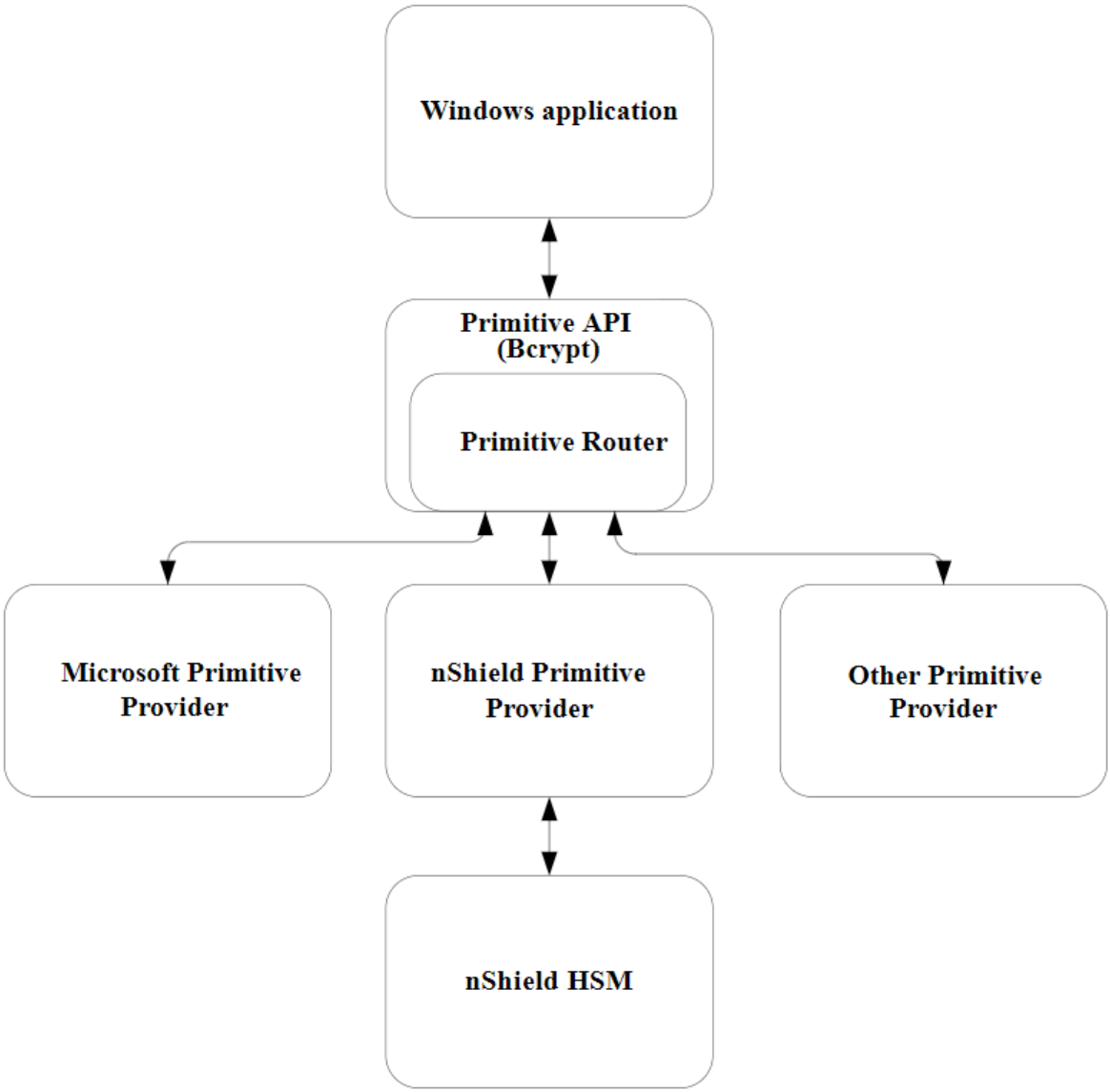
The Microsoft CryptoAPI, Java JCE, PKCS #11, as well as the application plug-ins, use the Security World paradigm for key storage.

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.

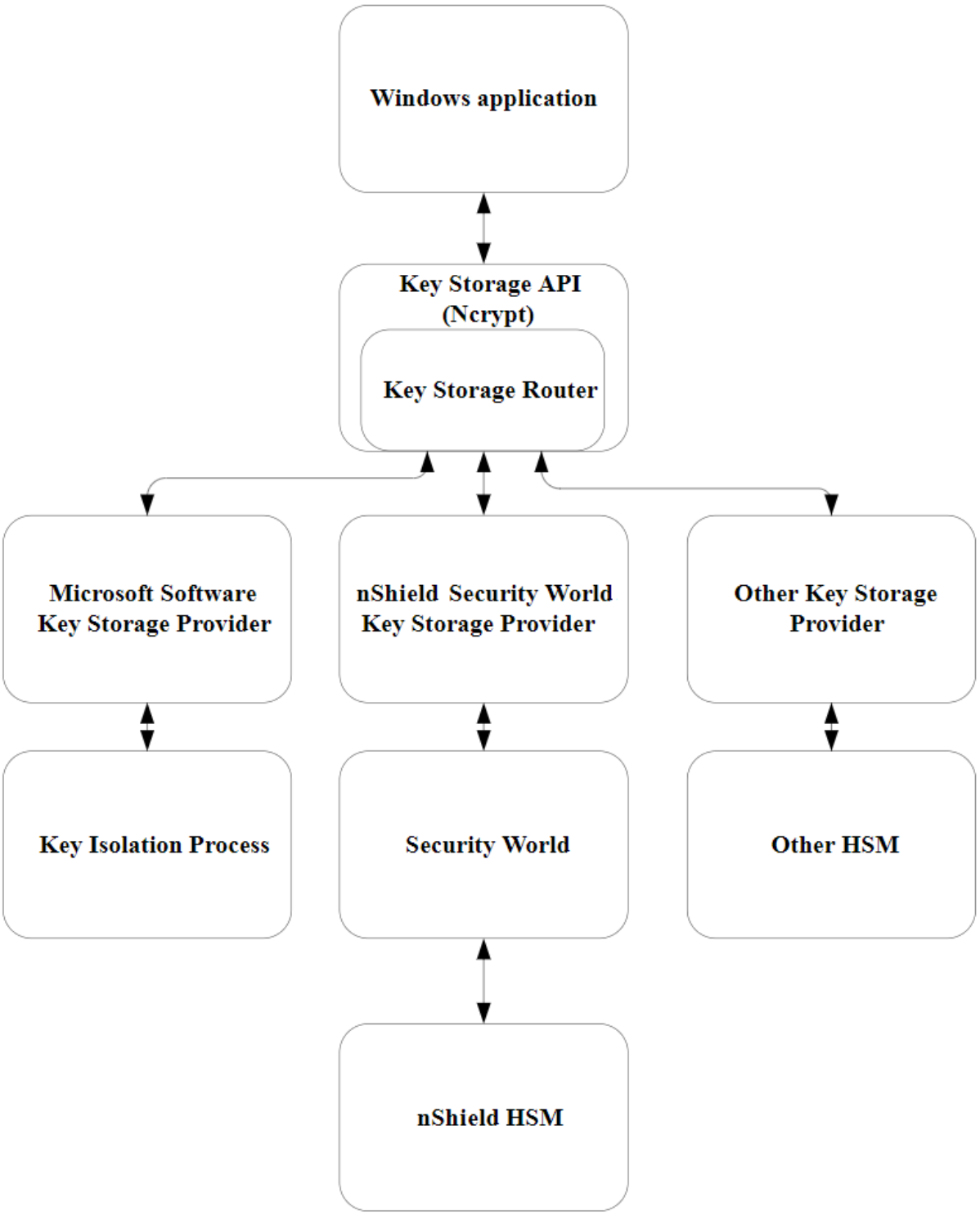
The NFKM library gives you access to all the Security World functionality.

3. CNG Architecture Overview

CNG handles cryptographic primitives and key storage through separate APIs. In both cases a Windows application contacts a router, which forwards the cryptographic operation to the provider that is configured to handle the request. For an illustration of communication between the architecture layers for cryptographic primitives, see the following diagram.



For an illustration of communication between the architecture layers for cryptographic key storage, see the following diagram.


Windows application
Key Storage API (Ncrypt)
Key Storage Router
Microsoft Software Key Storage Provider
nShield Security World Key Storage Provider
Other Key Storage Provider
Key Isolation Process
Security World
Other HSM
nShield HSM

4. Register the nShield CNG CSP

The DLL files that support the nShield CNG CSP are installed during product installation. However, you need to register the CNG CSP without removing the provider DLL files from your system.

You can unregister the nShield CNG CSP without removing the provider DLL files from your system. After unregistering, you can reregister the nShield CNG CSP, removing the files from your system. For more information, see Unregister or reregister the CNG CSP.

You can completely uninstall the nShield CNG CSP, removing the files from your system. After uninstalling, you must reinstall the files and then reregister the CNG CSP before you can use it. For more information, see Unregister or reregister the CNG CSP.

You can register the nShield CNG CSP with:

- CNG configuration wizard
- The `cngregister` command-line utility

To register the nShield CNG CSP, the hardserver must be running and able to communicate with at least one module. This requirement is normally fulfilled during the product installation process. You can check that this requirement is fulfilled by running the `enquiry` command-line utility and checking the output for details about the module.

4.1. Register the CNG CSP with the CNG configuration wizard

Entrust recommends using the CNG configuration wizard to register the nShield CNG CSP. The product installation process places a shortcut to the CNG configuration wizard in the Windows Start menu: **Start > Entrust nShield Security World**.

You can also perform the following actions with the CNG configuration wizard:

- Load existing Security Worlds
- Create Security Worlds
- Generate new Operator Card Sets (OCS)
- Configure the set-up parameters of the CNG CSP, including HSM Pool mode.
  With module firmware version 2.65.2 or later, if your application only uses module protected keys, you can use HSM Pool mode with multiple hardware security modules. HSM Pool mode exposes a single pool of HSMs and supports returning or adding a hardware security module to the pool without restarting the system. With a FIPS 140

Level 3 Security World, keys cannot be created in HSM Pool mode, however keys created outside HSM Pool mode can be used in HSM Pool mode. In audit logging worlds, HSM Pool mode is not supported.

- Configure user key namespacing
  Key namespacing allows multiple users to generate user keys of the same name by pre fixing the key blobs with the creating user's SID.
- Configure access controls for machine and user keys:
  - Read and write access for the creating user (built-in Administrators group for machine keys) only.
  - NFAST_KMLOCAL default ACLs (classic behaviour).
  - Read and write access for the creating user (built-in Administrators group for machine keys) and read access for a specified user or group SID.
  - User can specify the SDDL, either SACL or DACL, to be applied to the key blob by default.

To register the CNG CSP with the CNG configuration wizard, you must have already created a Security World and chosen a key protection method, either module-protection or OCS-protection. If you chose OCS-protection, you must also have already created an OCS before you can register the nShield CNG CSP with the CNG configuration wizard.

The CNG configuration wizard is not suitable for creating complex Security World setups. When used with network-attached HSMS, it is not suitable for creating Security Worlds with the unit. When creating such Security Worlds, or if you require more flexibility than the CNG configuration wizard provides, refer to nShield Security World v13.9.0 Management Guide.

If you use the CNG configuration wizard to create a Security World (and, if appropriate, an OCS), the wizard automatically prompts you to register the CNG CSP after you have fulfilled the necessary prerequisites.

You can also use the CNG configuration wizard to change an existing configuration at any time by running the wizard as usual and choosing the **Use the existing security world** option on the **Initial setup** screen.

To register the CNG CSP with the CNG configuration wizard after the necessary key-protection prerequisites have been fulfilled:

1. If the wizard is not already running:
   a. Run the wizard by double-clicking its shortcut in the Windows Start menu: **Start > Entrust nShield Security World**.

The wizard displays the welcome window.

b. Click the **Next** button.

The wizard allows you to configure HSM Pool mode for CNG.

c. Click the **Next** button.

If the prerequisite to create a Security World has been fulfilled, the wizard displays a confirmation screen.

d. Click the **Next** button.

The wizard displays a screen confirming that your Security World and (if you chose to create an OCS) an OCS have been created.

If you chose module-protection for your keys, the wizard does not confirm that an OCS has been created.

2. When the wizard has confirmed that it is ready to register the nShield CNG providers, click the **Next** button.

The wizard registers the nShield CNG CSP.

You cannot use the CNG configuration wizard to configure the nShield CNG providers for use as defaults. We recommend that you always use the nShield CNG providers by selecting them directly with the application that is using CNG.

When configuration of your nShield CNG CSP is complete, the wizard displays a confirmation screen.

4.2. Register the CNG CSP with cngregister

You can use the `cngregister` command-line utility to register the nShield CNG CSP manually even if you have not already created a Security World (or, if you choose OCS-protection for your keys, even if you have not already created an OCS).

To register the nShield CNG CSP with the `cngregister` command-line utility, run the command without specifying any options:

```
cngregister
```

You cannot use the `cngregister` command-line utility to configure the

> nShield CNG providers for use as defaults. We recommend that you always use the cngregister command-line utility, see Utilities.

4.3. Unregister or reregister the CNG CSP

You can use the `cngregister` command-line utility to unregister or reregister the nShield CNG CSP manually. You can also configure the defaults for protection type, user key name-spacing, and access control for user and machine keys when they are generated.

To unregister the nShield CNG CSP, run the command:

```
cngregister -U
```

This command unregisters the CNG CSP, but does not remove the provider DLL files from your system. For information about removing these files, see Uninstall or reinstall the CNG CSP.

> If any applications or services are using the nShield CNG providers for key storage or cryptography, unregistering the CNG CSP, you can rereg ister it at any time as long as the files have not been uninstalled from your system.

After unregistering the nShield CNG CSP, you can reregister it at any time as long as the files have not been uninstalled from your system. To reregister the nShield CNG CSP on your system, run the command:

```
cngregister
```

> You cannot use the `cngregister` command-line utility to configure the nShield CNG providers for use as defaults. We recommend that you always use the nShield CNG providers by selecting them directly with the application that is using CNG.

For more information about these command-line utilities, see Utilities.

4.4. Uninstall or reinstall the CNG CSP

To uninstall the nShield CNG CSP:

1. To remove any and all dependencies that you have set, run the command:

```
ncsvcdep -x
```

Always run `ncsvcdep` as a user with full administrative privileges.

2. Unregister the nShield CNG CSP on your system by running the command:

```
cngregister -U
```

This command unregisters the CNG CSP, but does not remove the provider DLL files from your system.

3. Uninstall the nShield CNG DLLs from your system:
   - On 32-bit versions of Windows, run the command:

```
cnginstall32 -U
```

   - On 64-bit versions of Windows, run the command:

```
cnginstall -U
```

To reinstall the nShield CNG CSP after you have previously uninstalled it:

1. Reinstall the nShield CNG CSP files on your system:
   - On 32-bit versions of Windows, run the command:

```
cnginstall32 -i
```

   - On 64-bit versions of Windows, run the command:

```
cnginstall -i
```

2. Reregister the nShield CNG CSP on your system by running the command:

```
cngregister
```

For more information about these command-line utilities, see Utilities

5. Supported Algorithms

This chapter lists the National Security Agency (NSA) classified Suite B algorithms supported by the nShield CNG providers.

The MQV algorithm is not supported by the nShield CNG providers.

Some mechanisms may be restricted from use in Security Worlds conforming to FIPS 140 Level 3. See Mechanisms for more information.

5.1. Signature interfaces (key signing)

Interface name	*Type of support*
RSA PKCS#1 v1	Hardware
RSA PSS	
DSA	
ECDSA_P224	
ECDSA_P256	
ECDSA_P384	
ECDSA_P521	

Hashes used with ECDSA must be of the same length or shorter than the curve itself. If you attempt to use a hash longer than the curve the operation returns `NOT_SUPPORTED`. In FIPS 140 Level 3 Security Worlds, curves must be of an approved type and length.

5.2. Hashes

Hash name	*Type of support*
SHA1	Hardware (HMAC only)/software
SHA256	
SHA384	
SHA512	
SHA224	Hardware (HMAC only, requires firmware version 2.33.60 or later)/software

Hash name	*Type of support*
MD5	Hardware (HMAC only)/software

5.3. Asymmetric encryption

Algorithm name	*Type of support*
RSA Raw (NCRYPT_NO_PADDING_FLAG)	Hardware
RSA PKCS#1 v1 (NCRYPT_PAD_PKCS1_FLAG)	
RSA OAEP (NCRYPT_PAD_OAEP_FLAG)	

5.4. Symmetric encryption

Algorithm name	*Type of support*
RC4	Hardware and Software
AES ECB,CBC	
DES ECB,CBC	
3DES ECB,CBC	
3DES_112 ECB,CBC	

5.5. Key exchange

Protocol name	*Type of support*
DH	Hardware
ECDH_P224	
ECDH_P256	
ECDH_P348	
ECDH_P521	

Elliptic curve cryptography algorithms must be enabled before use. Use the `fet` command-line utility with an appropriate certificate to enable a purchased feature. If you enable the elliptic curve feature on your modules after you first register the CNG providers, you must run the configuration wizard again for the elliptic curve algorithm providers to be regis-

> tered. For more information about registering the CNG providers, see Register the nShield CNG CSP.

5.6. Random Number Generation

Name	*Type of support*
RNG	Hardware

6. Migrate keys for CNG

Entrust provides functionality for migrating existing keys from other providers into the Security World Key Storage Provider. To identify installed providers, run the command:

```
cnglist --list-providers
```

To identify the keys that are available from a particular provider, run the command:

```
cnglist --list-keys --provider="ProviderName"
```

In this command, *ProviderName* is the name of the provider. The following command provides an example of identifying keys from the *Security World Key Storage Provider*:

```
cnglist --list-keys --provider="nCipher Security World Key Storage Provider"
MyApp Personal Data Key: RSA
CertReq-5eb45f6d-6798-472f-b668-288bc5d961da: ECDSA_P256 machine
WebServer Signing Key: DSA machine
ADCS-Root-Key: ECDSA_P521 machine
```

To list the keys available from the *Security World Key Storage Provider*, run the command `cnglist --list-keys` (without specifying the `--provider` option).

6.1. Importing a Microsoft CAPI key into the Security World Key Storage Provider

To import a Microsoft CAPI key into the Security World Key Storage Provider, first run the CAPI utility `csputils` to identify the existing CAPI containers and their key contents.

CAPI containers can contain either a signing key or a key exchange key, or both. The following example shows how to import both a signing key and a key exchange key from a Microsoft CAPI container:

```
cngimport -m --csp="Microsoft Strong Cryptographic Provider"
 -k "EXAMPLE_CAPICONTAINER"
    "EXAMPLE_IMPORTED_SIGNATURE_CAPICONTAINER"
    "EXAMPLE_IMPORTED_KEYEXCHANGE_CAPICONTAINER"
```

To check the success of the import, list the keys present in the *Security World Key Storage Provider*:

```
cnglist --list-keys
```

```
EXAMPLE_IMPORTED_SIGNATURE_CAPICONTAINER: RSA
EXAMPLE_IMPORTED_KEYEXCHANGE_CAPICONTAINER: DH
```

The following example command shows how to import a single signing key:

```
cngimport -m -s --csp="Microsoft Strong Cryptographic Provider"
 --key="EXAMPLE_CAPICONTAINER"
       "EXAMPLE_IMPORTED_SIGNATURE_ONLY_CAPICONTAINER"
```

Run the `cnglist` command with the `--list-keys` option to check the success of the key import:

```
cnglist --list-keys
EXAMPLE_IMPORTED_SIGNATURE_ONLY_CAPICONTAINER: RSA
```

The `cngimport` option `-m/--migrate` cannot be used to migrate nShield CAPI container keys to CNG. For information about importing nShield CAPI container keys into CNG, see Importing a Microsoft CNG key into the Security World Key Storage Provider.

6.2. Importing a Microsoft CNG key into the Security World Key Storage Provider

To import a Microsoft CNG key into the Security World Key Storage Provider, run the `cngimport` command as shown in the following example:

```
cngimport -m
 -k "EXAMPLE_RSA_1024"
    "IMPORTED_RSA_1024"
```

Run the `cnglist` command with the `--list-keys` option to check the success of the key import:

```
cnglist --list-keys
 IMPORTED_RSA_1024: RSA
```

The original key is not deleted from the provider from which it was imported:

```
cnglist --list-keys --provider="Microsoft Software Key Storage Provider"
 EXAMPLE_RSA_1024
```

Certain applications, such as Certificate Services, create keys using the Microsoft Software Key Storage Provider which cannot be exported.

> Attempting to import such a key into the nShield provider results in the following message:

```
cngimport -m -k WIN-KQ1Z6JMCUTB-CA WIN-ncipher-CA
Unable to continue.
This key can not be exported from Microsoft Software Key Storage Provider.
```

6.3. Importing a Security World key into the Security World Key Storage Provider

To import a Security World key into the Security World Key Storage Provider, run the `cngimport` utility as shown in the following example:

```
cngimport --import --key=nfkmsimple1 --appname=simple nfkmsimple1
Found key 'nfkmsimple1'
Importing NFKM key.. done
```

Run `cnglist` with the `--list-keys` option to confirm that the key has been successfully imported:

```
cnglist --list-keys
nfkmsimple1: RSA
```

To import an nShield CAPI container into the Security World Key Storage Provider, run the `csputils` command to identify the container name:

```
csputils -l
File ID    Container name       Container owner      DLL name   S X
=========  ===================  ===================  =========  = =
31e994f07  CONTAINER2           SYWELL\Administrato  ncsp       * *
3a2b082a8  CAPICONTAINER        SYWELL\Administrato  ncsp       * *
2 containers and 4 keys found.
```

> Run the `csputils` command with the `-l` and `-m` options to migrate an nShield CAPI machine container.

Identify the Security World key names of the keys in the container by running the `csputils` command as follows:

```
csputils -d -n CAPICONTAINER
Detailed report for container ID #3a2b082a8f2ee1a5acb756d5e95b09817072807a
Filename:       key_mscapi_container-3a2b082a8f2ee1a5acb756d5e95b09817072807a
Container name: CAPICONTAINER
User name:      SYWELL\Administrator
User SID:       s-1-5-21-352906761-2625708315-3490211485-500
CSP DLL name:   ncsp.dll
```

```
Filename for signature key is key_mscapi_ce51a0ee0ea164b993d1edcbf639f2be62c53222
  Key was generated by the CSP
  Key hash:    ce51a0ee0ea164b993d1edcbf639f2be62c53222
  Key is recoverable.
Key is cardset protected.
Cardset name:             nopin
     Sharing parameters:       1 of 1 shares required.
Cardset hash:             d45b30e7b60cb226f5ade5b54f536bc1cc465fa4
     Cardset is non-persistent.
Filename for key exchange key is key_mscapi_dbd84e8155e144c59cf8797d16e7f8bd19ac446a
  Key was generated by the CSP
  Key hash:    dbd84e8155e144c59cf8797d16e7f8bd19ac446a
  Key is recoverable.
Key is cardset protected.
Cardset name:             nopin
     Sharing parameters:       1 of 1 shares required.
Cardset hash:             d45b30e7b60cb226f5ade5b54f536bc1cc465fa4
     Cardset is non-persistent.
1 container and 2 keys found.
```

The key name to pass to the `cngimport` command `--key` option is the part of the key name that follows `key_mscapi_` in the output line that starts `Filename for signature key is key_mscapi_`.

For example, the signature key file name for `CAPICONTAINER` in the example shown above is `key_mscapi_ce51a0ee0ea164b993d1edcbf639f2be62c53222`, so `ce51a0ee0ea164b993d1ed-cbf639f2be62c53222` is the key name that should be passed to `cngimport`:

```
cngimport --import --key="ce51a0ee0ea164b993d1edcbf639f2be62c53222" --
appname="mscapi" Signature_Key_Imported_From_nCipher_CAPI
Found unnamed key
Importing NFKM key.. done
```

Run `cnglist` with the `--list-keys` option to confirm that the key has been successfully imported:

```
cnglist --list-keys
Signature_Key_Imported_From_nCipher_CAPI: RSA
cngsoak: ECDH_P256
```

Follow the same procedure for importing the key exchange key from the nShield CAPI container.

7. Key Authorization

When an application needs keys that are protected by an Operator Card Set or a Softcard, a user interface is invoked to prompt the application user to insert the smart card and/or enter appropriate passphrases.

The user interface prompt is not provided if your application is working in silent mode. The nShield CNG providers attempt to load the required authorization (for example, from an Operator Card that has already been inserted) but fail if no authorization can be found. For more information about silent mode, refer to the documentation of the CNG Key Storage Functions at: http://msdn2.microsoft.com/en-us/library/aa376208.aspx.

When the CNG application is running in Session 0 (that is, loaded by a Windows service), the user interface is provided by an agent process **nShield Service Agent** that is started when the user logs in. This agent, when running, is shown in the Windows System Tray. All user interaction requests from a CNG application running in Session 0 cause dialogs to be raised by the agent allowing the user to select cardsets, modules and enter passphrases. The interaction with the user is functionally identical to that described in this section.

There can only be one instance of the agent running (indicated by a blue globe in the Tray Notification area in the toolbar). Attempts to start a second instance will fail with a **CreateNamedPipe** error. If the agent is not running, attempts to invoke dialogs through it will fail and this is logged in the Windows Event Log. It can be restarted by logging off and on or by explicitly executing either `%NFAST_HOME%\bin\nShield_service_agent64.exe` or `%NFAST_HOME%\bin\nShield_service_agent.exe`. On 64 bit platforms either of these can be used irrespective of the bit size of the underlying application.

For more information about autoloadable Card Sets and the considerations of silent mode, see the authorisation requests diagram towards the end of this section.

You define key protection and authorization settings with the CNG Configuration Wizard on the **Key Protection Setup** screen. For more information about the CNG Configuration Wizard, see Register the nShield CNG CSP.

The options on this screen that are relevant to key protection and authorization are:

- **Module protection**

  Select this option to make keys module protected by default.

- **Softcard Protection**

  Select this option to generate new keys with a particular Softcard by default.

- **Operator Card Set protection**

  Select this option to generate new keys with a particular Operator Card Set by default.

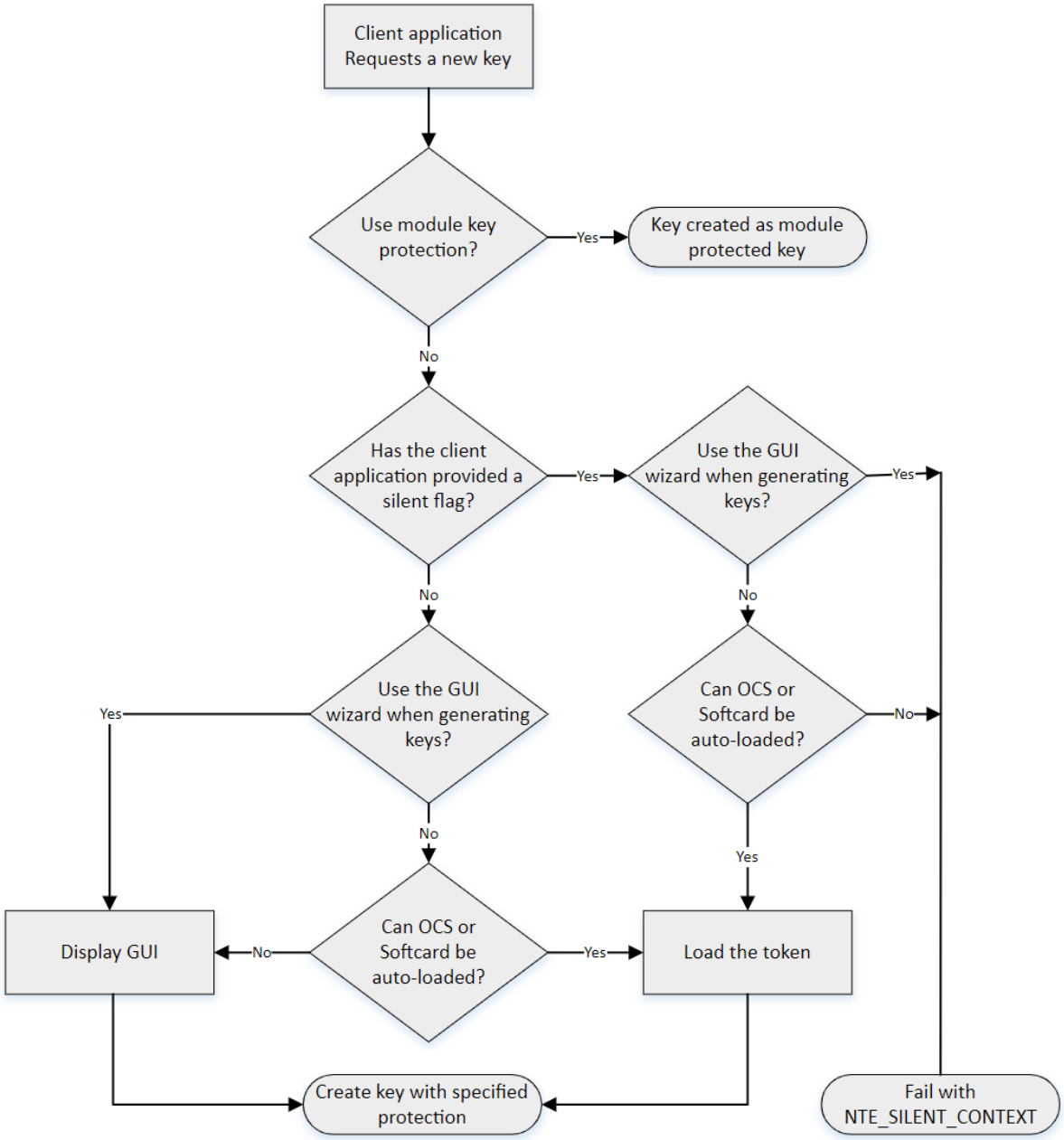
- **Allow any protection method to be selected in the GUI when generating**

  Select this option to defer selection of the key protection until the key is generated. When generating a key, the choice between Module protection, or protection with an existing Softcard or Operator Card Set, will be offered.

If you select Softcard or Operator Card Set protection, you will be offered the choice between selecting an existing protection token and creating a new one on the next page.

The CNG Configuration Wizard can be re-run to change the default protection. Existing keys that were generated with a different protection can still be loaded even if they don't match the protection that was selected in the wizard.

The nShield GUI is never enabled for calls with a valid **Silent** option. If the **Use the GUI wizard..** option is selected, and the providers have been passed the **Silent** option, key generation will always fail. For Softcard and Operator Card Set protection, **Silent** mode will work only if the Softcard or Operator Card Set can be autoloaded without prompting for user interaction or passphrase entry.



FIPS 140 Level 3 environments always require card authorization for key creation. When using the CNG Primitive Functions the user is not prompted to provide card authorization, but the request fails if no card is provided.

The key storage providers always respect calls made with the `Silent` option. Primitive providers never display a user interface.

Applications may have a mechanism to disable silent mode operation, thereby allowing appropriate passphrases to be entered. Ensure that you configure applications to use an appropriate level of key protection. For example, in Microsoft Certificate Services, you must select the **Use strong private key protection features provided by the CSP** option to dis-

able silent mode operation.

8. Key Use Counting

You can configure the CNG provider to count the number of times a key is used. Use this functionality, for example, to retire a key after a set number of uses, or for auditing purposes.

Key counting is not supported in HSM Pool mode.

To enable key use counting in the Security World Key Storage Provider, call `NCryptSetProperty` with `NCRYPT_USE_COUNT_ENABLED_PROPERTY` on the provider handle. Alternatively, to override the behavior of third-party software that would not otherwise provide the user with the option to enable key use counting, use one of the following methods:

- Set the environment variable `NCCNG_USE_COUNT_ENABLED` to `1`.
- Set the registry key `Software\nCipher\CryptoNG\UseCountEnabled` to `1`.

Keys created while the provider has key use counting enabled continue to have their use counts incremented, regardless of the state of the provider's handle. Key use counts are not recorded for keys created while the `NCRYPT_USE_COUNT_ENABLED_PROPERTY` is disabled on the provider handle.

Because the key counter is a 64-bit area in a specific module's NVRAM, the counted keys are specific to a single module. When a key is created you are prompted to specify which module to use, unless there is only one module in the Security World, or `preload` was used to preload authorization from an ACS on only one module.

The key counter is incremented each time a private key is used to:

- sign
- decrypt
- negotiate a secret agreement.

To test the performance of keys with counters, run the `cngsoak` command with the `-C` option:

```
cngsoak -C --sign --length=1024
```

To view the current key use count for keys, run the `cnglist` command with the `--list-keys` and `--verbose` options:

```
cnglist --list-keys --verbose
```

9. Using CAPI Keys

We now provide the capability to use keys generated by CAPI in CNG applications. This is provided through the standard `NCryptOpenKey` CNG API call. Passing either `AT_SIGNATURE` or `AT_KEYEXCHANGE` as the `dwLegacyKeySpec` parameter and the CAPI container name as the `pszKeyName` parameter will invoke this mode of operation. The CAPI key will be loaded into the CNG provider and will behave as if it was a CNG key. Any key authorization required will be handled with a user interface being invoked to prompt the application user to insert the smart card or enter appropriate passphrases. There is support for Key Usage and Key Counting properties.

The CNG application has to be written such that it calls `NCryptOpenKey` to open a CAPI key explicitly.

10. Utilities

Use the `nfkmverify` command-line utility to check the security of all stored keys in the Security World. Use `nfkminfo`, `nfkmcheck`, and other command-line utilities to assist in this process.

On 64-bit versions of Windows, both the 32-bit and 64-bit versions of the listed utilities are installed. When working on an 64-bit version of Windows, always ensure that you use the 64-bit version of the utility (if one is available).

nShield CNG CSP utilities:

- cngimport
- cnginstall
- cnglist
- ncsvcdep
- configure-csp-poolmode
- cngsoak

10.1. cngimport

x86	x64
`cngimport32.exe`	`cngimport.exe`

This utility is used to migrate Security World, CAPI, and CNG keys to the Security World Key Storage Provider. For more information, see Migrate keys for CNG.

10.2. cnginstall

x86	x64
`cnginstall32.exe`	`cnginstall.exe`

This utility is the nShield CNG CSP installer. Only use this utility to remove or reinstall the provider DLLs and associated registry entries manually.

To uninstall the nShield CNG DLL files, run the command:

```
cnginstall -U
```

This command removes the provider DLL files from your system. It produces output of the form:

```
ncksppt.dll removed.
nckspsw.dll removed.
ncpp.dll removed.
```

Before you uninstall the nShield CNG DLL files, ensure that you unregister the CNG CSP. For more information, see:

- cngregister
- UnregisterRegisterCNGCSP

After unregistering the nShield CNG CSP, you can reregister it at any time as long as the files have not been uninstallted from your system. To reregister the nShield CNG CSP on your system, run the command:

```
cngregister
```

10.3. cnglist

x86	x64
cnglist32.exe	cnglist.exe

This utility displays details of CNG providers, keys, and algorithms.

To list details of the CNG providers, run the `cnglist` command with the `--list-providers` option:

```
cnglist --list-providers
```

Output from this command is of the form:

```
Microsoft Primitive Provider
Microsoft Smart Card Key Storage Provider
Microsoft Software Key Storage Provider
Microsoft SSL Protocol Provider
nCipher Primitive Provider
nCipher Security World Key Storage Provider
```

To list details of the algorithms, run the `cnglist` command with the `--list-algorithms`

option:

```
cnglist --list-algorithms
```

Output from this command has the form:

```
BCryptEnumAlgorithms(BCRYPT_CIPHER_OPERATION):
  Name                          Class      Flags
  AES                           0x00000001 0x0
  RC4                           0x00000001 0x0
  DES                           0x00000001 0x0
  DESX                          0x00000001 0x0
  3DES                          0x00000001 0x0
  3DES_112                      0x00000001 0x0
BCryptEnumAlgorithms(BCRYPT_HASH_OPERATION):
  Name                          Class      Flags
  SHA1                          0x00000002 0x0
  MD2                           0x00000002 0x0
  MD4                           0x00000002 0x0
  MD5                           0x00000002 0x0
  SHA256                        0x00000002 0x0
  SHA384                        0x00000002 0x0
  SHA512                        0x00000002 0x0
  AES-GMAC                      0x00000002 0x0
  SHA224                        0x00000002 0x0
BCryptEnumAlgorithms(BCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION):
  Name                          Class      Flags
  RSA                           0x00000003 0x0
```

To list details of the algorithms for the Security World Key Storage Provider, run the `cnglist` command with the `--list-algorithms`, `--keystorage`, and `--nc` options:

```
cnglist --list-algorithms --keystorage --nc
```

Output from this command has the form:

```
NCryptEnumAlgorithms(NCRYPT_CIPHER_OPERATION) no supported algorithms
NCryptEnumAlgorithms(NCRYPT_HASH_OPERATION) no supported algorithms
NCryptEnumAlgorithms(NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION):
  Name                          Class      Operations Flags
  RSA                           0x00000003 0x00000014 0x0
NCryptEnumAlgorithms(NCRYPT_SECRET_AGREEMENT_OPERATION):
  Name                          Class      Operations Flags
  DH                            0x00000004 0x00000008 0x0
  ECDH_P224                     0x00000004 0x00000008 0x0
  ECDH_P256                     0x00000004 0x00000008 0x0
  ECDH_P384                     0x00000004 0x00000008 0x0
  ECDH_P521                     0x00000004 0x00000008 0x0
NCryptEnumAlgorithms(NCRYPT_SIGNATURE_OPERATION):
  Name                          Class      Operations Flags
  RSA                           0x00000003 0x00000014 0x0
  DSA                           0x00000005 0x00000010 0x0
  ECDSA_P224                    0x00000005 0x00000010 0x0
  ECDSA_P256                    0x00000005 0x00000010 0x0
  ECDSA_P384                    0x00000005 0x00000010 0x0
  ECDSA_P521                    0x00000005 0x00000010 0x0
```

To list details of the algorithms for a specific named key storage provider, run the `cnglist` command with the `--list-algorithms` and `--provider`="*ProviderName*" options:

```
cnglist --list-algorithms --provider="Microsoft Software Key Storage Provider"
```

Output from this command has the form:

```
Microsoft Software Key Storage Provider
NCryptEnumAlgorithms(NCRYPT_CIPHER_OPERATION) no supported algorithms
NCryptEnumAlgorithms(NCRYPT_HASH_OPERATION) no supported algorithms
NCryptEnumAlgorithms(NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION):
  Name                          Class      Operations Flags
  RSA                           0x00000003 0x00000014 0x0
NCryptEnumAlgorithms(NCRYPT_SECRET_AGREEMENT_OPERATION):
  Name                          Class      Operations Flags
  DH                            0x00000004 0x00000008 0x0
  ECDH_P256                     0x00000004 0x00000018 0x0
  ECDH_P384                     0x00000004 0x00000018 0x0
  ECDH_P521                     0x00000004 0x00000018 0x0
NCryptEnumAlgorithms(NCRYPT_SIGNATURE_OPERATION):
  Name                          Class      Operations Flags
  RSA                           0x00000003 0x00000014 0x0
  DSA                           0x00000005 0x00000010 0x0
  ECDSA_P256                    0x00000005 0x00000010 0x0
  ECDSA_P384                    0x00000005 0x00000010 0x0
  ECDSA_P521                    0x00000005 0x00000010 0x0
```

10.4. cngregister

x86	x64
`cngregister32.exe`	`cngregister.exe`

This is the nShield CNG CSP registration utility. You can use it to unregister and re-register the nShield providers manually.

To configure user key namespacing, run the command and set the `--user-namespacing` option to `disabled` or `enabled` as required:

```
cngregister --user-namespacing=<disabled or enabled>
```

By default, user key namespacing is enabled.

To configure access control for user and machine keys when they are generated, the options are:

- `STANDARD`: Provides read and write access for the creating user (built-in Administrators group for machine keys) only (default).

- `CLASSIC`: Uses the `NFAST_KMLOCAL` default ACLs (classic behaviour).
- A valid SID starting with `S-1-`: Provides read and write access for the creating user (built-in Administrators group for machine keys) and read access for a specified user or group SID.
- A valid SDDL, either SACL or DACL, to be applied to the key blob by default.

Examples of user key access control configuration:

```
cngregister --user-access-control=STANDARD

cngregister --user-access-control=CLASSIC

cngregister --user-access-control=S-1-5-32-569

cngregister --user-access-control=D:P(A;;GA;;;SY)(A;;GA;;;BA)(A;;GR;;;S-1-5-32-569)
```

Examples of machine key access control configuration:

```
cngregister --machine-access-control=STANDARD

cngregister --machine-access-control=CLASSIC

cngregister --machine-access-control=S-1-5-32-569

cngregister --machine-access-control=D:P(A;;GA;;;SY)(A;;GA;;;BA)(A;;GR;;;S-1-5-32-569)
```

To configure the key protection type to one of the following:

- `wizard`: Prompt the user when the key is generated.
- `module`: Use module protection (default).
- `softcard`: Use softcard protection.
- `cardset`: Use OCS cardset for protection.

For `softcard` and `cardset` protection, the additional `token-hash` must be specified also:

```
cngregister --protection-type=wizard

cngregister --protection-type=module

cngregister --protection-type=softcard --token-hash=34c257d1485aa433f80a5b23b30cb4caac177959

cngregister --protection-type=cardset --token-hash=8389d4031e3e235a18c80fa93a3cc893c5ea55ba
```

To unregister the nShield CNG CSP, run the command:

```
cngregister -U
```

This command produces output for the form:

```
Unregistered provider 'nCipher Primitive Provider'
Unregistered provider 'nCipher Security World Key Storage Provider'
```

This command unregisters the CNG CSP, but does not remove the provider DLL files from your system. For information about removing these files, see:

- cnginstall
- UninstallingReinstallingCNG.

If any applications or services are using the nShield CNG CSP for key storage or cryptography, unregistering it can cause system instability.

After unregistering the nShield CNG CSP, you can reregister it at any time as long as the files have not been uninstalled from your system. To reregister the nShield CNG CSP on your system, run the command:

```
cngregister
```

You cannot use the `cngregister` command-line utility to configure the nShield CNG providers for use as defaults. We recommend that you always use the nShield CNG providers by selecting them directly with the application that is using CNG.

10.5. ncsvcdep

x86	x64
`ncsvcdep32.exe`	`ncsvcdep.exe`

This utility is the service dependency tool. You can configure some service based applications, such as Microsoft Certificate Services and IIS, to use the nShield CNG CSP. The nShield Service dependency tool enables you to add the nFast Server to the dependency list of such services.

Use the `ncsvcdep` utility to ensure that the nShield `nFast Server` service is running before certain services are enabled. For example, Active Directory Certificate Services or Internet Information Services require that the hardserver is running in order to use the nShield CNG CSP. Failure to set this dependency can lead to system instability.

To list installed services, run the `ncsvcdep` command with the `-l` option:

```
ncsvcdep -l
```

Output from this command has the form:

```
Installed Services (Count - "Display Name" - "Service Name")
0 - "Application Experience" - "AeLookupSvc"
1 - "Application Layer Gateway Service" - "ALG"
2 - "Application Information" - "Appinfo"
3 - "Application Management" - "AppMgmt"
4 - "Windows Audio Endpoint Builder" - "AudioEndpointBuilder"
.
.
108 - "nFast Server" - "nFast Server"
109 - "Active Directory Certificate Services" - "CertSvc"
```

Always run `ncsvcdep` as a user with full administrative privileges.

To set a dependency, run the command:

```
ncsvcdep -a "DependentService"
```

In this command, *DependentService* is the service that has the dependency. The following example shows how to make the Active Directory Certificate Services dependent on the nFast Server:

```
ncsvcdep -a "CertSvc"
Dependency change succeeded.
```

To remove a specific dependency relationship, run `ncsvcdep` with the `-r` option, for example:

```
ncsvcdep -r "CertSvc"
Dependency change succeeded.
```

To remove all dependencies, run `ncsvcdep` with the `-x` option:

```
ncsvcdep -x
```

Microsoft Certificate Services require that the `certsvc` service is made dependent on the hardserver.

Microsoft Internet Information Services require that the `http` service is made dependent on the hardserver.

10.6. configure-csp-poolmode

x86	x64
configure-csp-poolmode32.exe	configure-csp-poolmode.exe

This utility enables you to configure HSM Pool mode for the nShield CNG CSP without using the CNG wizard.

To enable HSM Pool mode for CNG run the command:

```
configure-csp-poolmode --cng --enable
```

To disable HSM Pool mode for CNG run the command:

```
configure-csp-poolmode --cng --disable
```

To remove HSM Pool mode setting for CNG from the registry, use the command:

```
configure-csp-poolmode --cng --remove
```

10.7. cngsoak

x86	x64
cngsoak32.exe	cngsoak.exe

This utility provides statistics about the performance of the nShield CNG CSP. Specifically, use `cngsoak` to determine the speed of:

- Signing a hash (`cngsoak --sign`)
- encryption (`cngsoak --encrypt`)
- key exchange (`cngsoak --keyx`)
- key generation (`cngsoak --generate`).

The output from `cngsoak` displays information as columns of information. From left to right, these columns display:

- The time in second that `cngsoak` has been running
- the total number of operations completed
- the number of operations completed in last second
- the average number of operations completed each second.

11. Environment Variables for CNG Protection

A set of environment variables are supported for controlling CNG protection options on a per-application basis. These variables are documented here to facilitate more complicated deployments, but it should be noted that they are liable to change between releases.

Environment Variable	Description
`NCCNG_PIN`	Passphrase for Softcard. This enables the passphrase to be specified programmatically rather than through the GUI passphrase prompt. Note: This can expose your passphrase.  It is recommended that this be set in a context where the passphrase will be visible only to the user or service that should have access to this passphrase. It should **not** be set as a machine-wide environment variable.
`NCCNG_USE_MODULE_KEYS`	• If set to 1, module protection will be used for new keys that are generated. • If set to 0, the `NCCNG_PROTECTION_TOKEN` environment variable controls the protection option used.
`NCCNG_PROTECTION_TOKEN`	If `NCCNG_USE_MODULE_KEYS` is set to 0 (or a protection option other than module key protection or HSM pool mode was selected in the wizard) this environment variable enables the protection token to be specified for new keys that are generated. • If set to `softcard:HASH` the Softcard with the specified hash will be used. • If set to `cardset:HASH` the OCS with the specified hash will be used. • If set to anything else (for example, wizard), the GUI key protection wizard will be used. The HASH for Softcard or OCS protections refers to its Security World hash in hexadecimal, which can be identified using `nfkminfo -s` for softcards or `nfkminfo -c` for OCS.
`NCCNG_ALWAYS_USE_AGENT`	By default, if a CNG provider must display GUI, it will display it in the calling application if not in Session 0, and in the nShield Service Agent if running in Session 0 (for example, running as a service). Setting `NCCNG_ALWAYS_USE_AGENT` to 1 forces CNG GUI prompts to always be displayed in the nShield Service Agent regardless of whether it is running in Session 0. (If setting this value to 1 ensure that the nShield Service Agent is running).

Environment Variable	Description
NCCNG_USER_NAMESPACING	Namespacing allows multiple users to generate user keys of the same name. Namespacing is enabled by default. • If set to 1, namespacing will be used for new keys that are generated. The key blobs are prefixed with the creating user's SID. • If set to 0, the NCCNG_PROTECTION_TOKEN environment variable controls the protection option used. The key blobs are prefixed with "user".
NCCNG_MACHINE_ACCESS_CONTROL	Configures the default permissions for the key blob files when machine keys are generated if no security descriptor is provided by the application. By default, the machine key access control is configured as STANDARD. • If set to STANDARD, then generated keys have read and write permissions for the built-in Administrators group only. • If set to CLASSIC, then generated keys use the NFAST_KMLOCAL default permissions. • If set to a valid SID starting with S-1-, then generated keys have read and write permissions for the built-in Administrators group and read permissions for the user or group specified by the SID. • If set to a valid SDDL, either SACL or DACL, then generated keys will have the file permissions of the SDDL applied to the key blob by default.
NCCNG_USER_ACCESS_CONTROL	Configures the default permissions for the key blob files when user keys are generated if no security descriptor is provided by the application. By default, the user key access control is configured as STANDARD. • If set to STANDARD, then generated keys have read and write permissions for the creating user only. • If set to CLASSIC, then generated keys use the NFAST_KMLOCAL default permissions. • If set to a valid SID starting with S-1-, then generated keys have read and write permissions for the creating user and read permissions for the user or group specified by the SID. • If set to a valid SDDL, either SACL or DACL, then generated keys will have the file permissions of the SDDL applied to the key blob by default.
NCCNG_IGNORE_SECURITY_DESCRIPTOR	By default, this is used if an application, for example certreq.exe, provides a security descriptor over the CNG API when keys are generated. This option allows CNG to ignore the provided security descriptor and use the default instead. • If set to 1, any supplied security descriptor will be discarded when a new key is generated and the default access control will be applied to the new key. • If set to 0, when a new key is generated the supplied security descriptor will be applied to the new key (default).