nShield Security World (Multi-Tenancy)

nShield v14.1.1 Service Provider User Guide

11 June 2026

Table of Contents

1. Introduction

The nShield Service Provider User Guide provides useful information about how to use your nShield HSM to provision VCMs for your tenants.

1.1. Read this guide if ...

Read this guide if you need to configure or manage an Entrust Hardware Security Module (HSM) for use in multi-tenancy.

Before using this guide you should have:

- Installed your nShield HSM as described in Installing an nShield HSM
- Installed the software as described in Installing the nShield Security World software for an nShield HSM

1.1.1. Terminology

Term	Description
Service Provider	A person or organisation that manages VCMs for use by tenants. The service provider has physical access to the HSM.
Tenant	A person or organisation that makes use of a VCM to provide cryptographic services.
VCM	A cryptographic module implemented as a share of a physical HSM that provides all the cryptographic services of an HSM and is securely separated from any other VCMs implemented on the same physical HSM.

2. After installation

If your HSM was a new installation or, if your HSM was upgraded from a firmware version earlier than v14.1.1 then, once the hardware and software have been successfully installed, the HSM will be running a number of platform services but there will be no VCMs running and therefore no tenant services, see Platform services (nShield 5 HSMs).

If you have not purchased a multi-tenancy license you will be restricted to creating and run ning a single VCM. In this case you can automatically create, start and enroll a single VCM by use of the command hsmadmin vcm single-setup.

Entrust only recommend use of `hsmadmin vcm single-setup` in situations where you intend to have only a single VCM. This is because the command automatically sets a number of options and makes networking choices that may not be appropriate when creating subsequent VCMs.

In order to create more than one VCM you must purchase a licence from Entrust Sales. See Maximum number of concurrently active VCMs feature. Then create VCMs for your tenants by following the instructions at providing VCMs for your tenants.

3. HSM management for multi-tenancy

3.1. Management of VCMs

3.1.1. Terminology

Term	Description
Active VCM	A VCM that is currently running and available for use by its tenant. An active VCM will consume a portion of the memory, disk and CPU resources of the HSM. The maximum number of active VCMs is a licensable feature.
Inactive VCM	A VCM that is not currently running and is not contactable or usable by its tenant. An inactive VCM consumes negligible memory, disk and CPU resources. If a Security World had previously been loaded on the inactive VCM, this is retained and will be available for use as soon as the VCM is made active without requiring the Security World to be re-loaded and without requiring a quorum of ACS cards to be presented.

3.1.2. Providing VCMs for your tenants

To provide VCMs for your tenants, you must first configure the VCM networking, which is common to all VCMs, and then create as many VCMs as required by following the guidance below.

3.1.2.1. Set VCM networking

Before you create your first VCM you must decide the network addressing for the VCMs which must be consistent for all VCMs created on the HSM.

You may choose from:

- IPv4 addresses
- IPv6 addresses
- IPv6 link local addresses

Note that IPv6 link local addresses are only useful if all tenants are hosted on the same physical machine in which the nShield HSM is installed. This may be the case if you have only a single VCM created or if you are using a solution in which the tenant host machines are containerized.

Once you have chosen the networking you wish to use, configure the HSM using the com-

mand hsmadmin setnetwork

After changing the network settings you must reset the HSM using the command hsmadmin reset in order for them to take effect.

3.1.2.2. Create a VCM

VCMs are requested by tenants and in order to create a VCM you must have received a tenant request message. The tenant request message will include:

- A public key to be used for communication with the `sshadmin` service in the VCM
- A public key to be used to verify subsequent requests from the tenant
- Any properties the tenant has requested for the VCM; see VCM properties.

You can view the contents of the tenant request with the command hsmadmin vcm inspect-request.

You may choose to create the VCM with a different set of properties from those requested by the tenant. If you do this you should communicate with the tenant and in some cases it may be necessary to ask the tenant to send an updated request message.

Once you have decided on the VCM properties, create the VCM by using vcmadmin create.

The creation of the VCM will automatically create a configuration file that you must send to the tenant.

A tenant can enroll only one VCM hosted on a given HSM at a time. If you create more than one VCM on the same HSM for the same tenant, they must unenroll the existing VCM before enrolling a new one. In general, create only one VCM per HSM per tenant unless otherwise agreed with the tenant.

3.1.2.2.1. Limits on creating VCMs

If you have not purchased a multi-tenant licence, see Maximum number of concurrently active VCMs feature you will only be able to create a single VCM.

If you have purchased a multi-tenant licence, the number of VCMs that you can create will depend on the speed rating of the HSM. Base speed HSMs are limited to a maximum of 5 VCMs. Higher speed HSMs can create up to 1000 VCMs. If you wish to upgrade the speed rating of your HSM you can do this by purchasing a speed upgrade licence. See Speed ratings.

If you try to create a VCM in excess of the limit the command will fail. You must delete an existing VCM using vcmadmin delete.

You can see the maximum number of VCMs that you can create by using the command hsmadmin info and looking at the 'max_creatable_vcms'.

3.1.2.3. VCM properties

A VCM can be configured with different properties. These properties can be specified when the VCM is created or modified later, as described in Modify a VCM. The VCM must be stopped before its properties can be modified.

3.1.2.3.1. autostart

This property determines whether the VCM will automatically start after a reboot of the HSM on which it is hosted.

This property is mutually exclusive with the `restrict-startup` property.

3.1.2.3.2. restrict-startup

This property determines whether the service provider is able to start the VCM without authorization from the tenant. If this property is selected, the tenant must generate a start request message and send it to the service provider to authorize startup of the VCM.

This property is mutually exclusive with the `autostart` property.

3.1.2.3.3. serial-cardreader

This property determines whether the VCM has access to the serial cardreader port that is part of the physical HSM on which the VCM is hosted. Since the HSM is owned by the service provider and located within the service provider's premises, this property would normally be used only when the tenant and service provider belong to the same organization.

Only one VCM can have access to the serial cardreader port on a given HSM at a time. Therefore, this property is normally granted on a temporary basis, except when an HSM is expected to host only one tenant.

3.1.2.3.4. features

This property determines the optional VCM features listed in Available optional features. For more information about managing VCM features, see VCM features.

3.1.2.3.5. name

This property sets the name of the VCM as seen by the service provider in some command outputs. It is never used as an input parameter but acts as a 'friendly name' in command out puts that support it.

3.1.2.3.6. IP address

This property sets the IP address that the tenant will use to contact their VCM. It must be within the range that was specified in Set VCM networking.

3.1.2.3.7. k_tenant_req

This property sets the public key used by the tenant to sign tenant request messages. It is automatically set when the VCM is created and should be changed only if the tenant informs you that the request signing key has changed. A tenant request to change the request signing key must be signed by the current request signing key.

3.1.3. Managing existing VCMs

Once VCMs have been created, you can manage them using the guidance below.

3.1.3.1. Start a VCM

The tenant will not be able to enroll or use the VCM if it is not started. You can start the VCM using vcmadmin start.

In some cases the tenant may have specified that they wish to authorize the starting of their VCM. If this is the case you must receive a valid request message from the tenant before you can start the VCM. The request message may have a limited validity time and you must start the VCM before this expires. If the request has expired you must ask the ten ant to generate a new request.

It is important that the clock on the tenant's machine is synchronized with the clock on your machine. You will need to contact the tenant to confirm this. It is also important that the module clock is synchronized with your host machine. Refer to Setting the system clock for how to set the system clock, as any error messages received will refer to the module clock. If this is not the first time the VCM is being started since it was created, as would be the case if the VCM had been stopped by following the steps in Stop a VCM then starting a VCM will put the ten-

> ant back in the same position they were before the VCM was stopped. Therefore if the VCM was previously enrolled there is no need for the tenant to repeat the enrollment and, if a Security World was previously loaded there is no need to reload the Security World, and thus no need for a quorum of ACS cards.

3.1.3.1.1. Limits on starting VCMs

The maximum number of VCMs that can be concurrently active is a licensable feature, see Maximum number of concurrently active VCMs feature. If you try to start VCMs in excess of your current license the command will fail. Either stop another VCM that is currently running using vcmadmin stop or contact Entrust Sales to purchase an updated license.

You can see the maximum number of VCMs that you can start by using the command hsmadmin info and looking at the 'max_active_vcms'.

3.1.3.1.2. Autostarting VCMs

When a VCM is created it can be configured to autostart after a reboot of the HSM. This property must be included in the tenant request message. VCMs configured with this property will automatically start after a reboot of the HSM and for this reason this property is mutually exclusive with the property for the tenant to authorize starting of the VCM.

If more VCMs are configured with the autostart property than the maximum number of con currently active VCMs, see Maximum number of concurrently active VCMs feature then not all of them will be started after the reboot. For this reason you should not create more VCMs with this property than your licence allows. If a tenant requests the autostart property and you choose not to configure it, you should contact the tenant to let them know.

3.1.3.2. Stop a VCM

When a VCM is temporarily no longer needed it may be stopped using the command vcmad min stop. This puts the VCM in an inactive state where the tenant can no longer use or contact the VCM, but does not erase any Security World data.

You should not stop a VCM that is currently in use by a tenant. You should first contact the tenant and ask them to put their VCM in maintenance mode by following the instructions at Check and change the mode of operation.

If you intend to permanently delete the VCM after stopping it you should also ask the tenant if the VCM was using audit logging, see Audit Logging and if so ask them to finalize the audit log.

If you need to stop the VCM and cannot contact the tenant then you can use the `--force` option to force the VCM to stop.

An inactive VCM can be made active by following the steps in Start a VCM. This will allow the tenant to contact and use the VCM again and puts them back in the same position they were prior to the VCM being stopped. However if you used the `--force` option when stopping the VCM then the tenant will need to restart their hardserver in order to resume communication with it.

3.1.3.3. Delete a VCM

When a VCM is permanently no longer needed it may be deleted using the command vcmadmin delete. This permanently deletes the VCM and destroys any secrets associated with it, including any Security World data. This action is irreversible.

You can only delete an inactive VCM. If the VCM is currently active you must stop it by following the steps at Stop a VCM before you can delete it.

If the VCM has an unfinalized audit log then you will not be able to delete the VCM unless you use the `--force` option. See Disabling audit logging for information on how to finalize the audit log.

You cannot delete a VCM that is still running even with the `--force` option. You must follow the procedures in Stop a VCM first. The `--force` option for `vcmadmin delete` is only for the case in which the VCM has an unfinalized audit log.

3.1.3.4. Modify a VCM

You can view the current properties of a VCM with the command vcmadmin show.

In order to modify the properties of a VCM you must have received a request message from the tenant. The request message may have a limited validity time and you must modify the VCM before this expires. If the request has expired you must ask the tenant to generate a new request.

It is important that the clock on the tenant's machine is synchronized with the clock on your machine. You will need to contact the tenant to confirm this. It is also important that the module clock is synchronized with your host machine. Refer to Setting the system clock for information about setting the system clock, as any error messages received will refer to the module clock.

You can view the contents of the tenant request with the command hsmadmin vcm inspect-request. You can only modify a property that is contained in the tenant request.

A tenant request message may include requests to change more than one property, but you can modify only one property at a time with this command. To change multiple properties, issue the command multiple times and reuse the same tenant request file each time.

VCM properties are modified using the command vcmadmin properties set.

The properties `autostart`, `restrict_startup`, and `serial_cardreader` are set to either `true` or `false`. You can also use `True` and `False`. Other properties are supplied as a new value for the property.

For example:

```
vcmadmin properties set --request AUTOSTART_REQUEST 4f9d05b8-95f4-41ab-8b0a-0c9c7ce14209 autostart "true"
vcmadmin properties set --request NAME_REQUEST 4f9d05b8-95f4-41ab-8b0a-0c9c7ce14209 name "My_new_VCM1"
```

You can only modify a property to the exact value included in the tenant request. If you attempt to use a different value, you will receive an error similar to the example below:

```
vcmadmin properties set --request REQUEST_FILE 4f9d05b8-95f4-41ab-8b0a-0c9c7ce14209 restrict_startup "False"
ERROR: Failed to set VCM properties: orchestrator setproperties: Invalid property value: restrict_startup
mismatch
```

The `restrict_startup` and `autostart` properties are mutually exclusive and it is not possible to modify them so that both are set to "true".

3.1.3.5. Show VCM statistics

You can retrieve information about the resource usage of all the VCMs that have been created.

VCM statistics are collected by a background process that periodically polls all VCMs, both active and inactive, and stores the information within the module. Once the information has been collected there is a fixed 30s delay before information is polled again. The command vcmadmin stats is used to retrieve the information from the module. Depending on which point in the polling cycle this command is issued, the retrieved information can be stale by up to a maximum of 30s plus the time taken to poll all VCMs.

The time taken to poll the VCMs depends on how many VCMs that are running and the loading of the HSM at the time. In the worst case, with 21 running VCMs and the HSM at maximum load, information returned may be up to 90s out of date.

3.1.3.5.1. VCM resource usage

You can see the memory used by each VCM by using the command vcmadmin stats and looking at the `Memory use`. You can see the total memory used by the HSM by use of the command hsmadmin getenvstats and looking at `mem_used`.

The memory used by the HSM will be the total of the memory used by all the VCMs hosted on it plus the memory being consumed by the platform. You can get an approximate figure for the memory used by the platform by seeing the `mem_used` figure when no VCMs are created.

The amount of persistent data stored by a VCM can be seen by using the command vcmadmin stats and looking at the `Disk use`. This memory is consumed even when the VCM is inactive.

To prevent a single VCM from consuming too much memory due to persistent storage a VCM will be automatically stopped if its disk usage exceeds 16 MB. Thus you should periodically monitor the disk usage of each VCM and contact the tenant if they are approaching the limit.

Once a VCM has been stopped due to exceeding the limit it must be deleted by using the command vcmadmin delete.

3.2. Platform features

In a multi-tenant system features exist at both service provider (platform) level and tenant (VCM) level.

Features at service provider level are purchased from Entrust and supplied as licence files. These licence files are then applied to the system using hsmadmin fet.

You can examine a license file with any suitable text editor as the start of the licence file is human readable and gives information about the license type and the ESN to which the license applies.

For handling of features at VCM level see VCM features

3.2.1. Platform specific features

Some features are only relevant at platform level and cannot be set as a VCM feature. They are listed below.

3.2.1.1. Maximum number of concurrently active VCMs feature

The number of VCMs that can be active at the same time on a given HSM is controlled by a feature licence. If a licence has not been purchased the HSM will be limited to creating and starting only one VCM.

To increase the number of active VCMs you must purchase a license from Entrust and apply it with hsmadmin fet.

Once you purchase a license, regardless of how many active VCMs it is for, you will be able to create up to 1000 VCMs on a High or Mid speed HSM and up to 5 VCMs on a Base speed HSM, the license limit applies only to the number of VCMs that can be simultaneously active.

If the license applied is for a lower number of active VCMs than currently set on the system, the license will still be applied successfully. Any VCMs currently active will still run but you will not be able to start any new VCMs until the number of currently active VCMs is reduced below the maximum. For this reason you should inspect your license to see how many VCMs it is for before applying it. License files can be examined with any text editor and include the value in human readable text at the start of the file.

3.3. VCM features

3.3.1. Handling of VCM features

Features applicable to a VCM may be specified when the VCM is created by using the `--features` option of vcmadmin-create. If the `--features` property is omitted when creating the VCM then the VCM features will default to be the same as those at platform level.

VCM features can be altered by first stopping the VCM with command vcmadmin stop and then altering the features with vcmadmin properties set. You will require a request file signed by the tenant to alter the properties.

It is important that the clock on the tenant's machine is synchronized with the clock on your machine. You will need to contact the tenant to confirm this. It is also important that the module clock is synchronized with your host machine. Refer to Setting the system clock for how to set the system clock, as any error messages received will refer to the module clock.

In all cases the features set at VCM level must be equal to, or a subset of, the features at platform level. It is not possible to set a feature for a VCM that does not exist at platform level. If you wish to set a feature for a VCM that does not exist at platform level you must first purchase a license for the HSM and apply that license at platform level.

3.3.2. Built-in features

A number of features that were optional on earlier versions of firmware are now built in to the firmware and are always available for use regardless of whether they are specified in any commands or not. These are:

- StandardKM
- EllipticCurve
- ECCMQV
- AcceleratedECC
- PostQuantum

Specifying, or not specifying, these features in any commands will have no effect on the tenant's ability to use these features.

3.3.3. Specification of VCM features

The features option can be specified in one of two ways as described below.

3.3.3.1. Specification of VCM features as feature names

Features may be specified as a list of feature names, as specified in the table below, separated by the | character. To protect the | character from the command line interpreter it must be enclosed in double quotes. For example if the required features are:

- FTO commands available
- Korean Algorithms available
- ECC mechanisms available

This would be written as

```
"ForeignTokenpen|KISAAlgorithms|EllipticCurve"
```

3.3.3.2. Specification of VCM features as a 32-bit word

Features may be specified as a 32-bit hexadecimal word, calculated by adding the hexadecimal values of the features listed in the table below.

For example if the required features are:

- FTO commands available
- ReceiveShare command available
- Any SEE machine may be loaded
- Korean Algorithms available
- Standard key management commands available
- ECC mechanisms available
- ECC-MQV key agreement protocol is available
- ECC point multiplications are accelerated where available
- High Speed HSM

The calculation would be:

```
00000001 + 00000002 + 00000004 + 00000010 + 00000020 + 00000200 + 00000800 + 00001000 + 00008000 = 9A37
```

3.3.3.3. VCM feature values and feature names

Value (Hexadecimal)	Feature	Feature name
00000001	FTO commands available	ForeignTokenOpen
00000002	ReceiveShare command available	RemoteShare
00000004	Any SEE machine may be loaded (Not applicable to nShield 5)	GeneralSEE
00000008	CodeSafe 5 activation	ExportCGEA
00000010	Korean algorithms available	KISAAlgorithms
00000020	Standard key management commands available	StandardKM
00000200	ECC mechanisms available	Elliptic curve
00000400	Base Speed HSM	HSMSpeed0
00000800	ECC-MQV key agreement protocol is available	ECCMQV
00001000	ECC point multiplications are accelerated where available	AcceleratedECC
00004000	Mid Speed HSM	HSMSpeed1

Value (Hexadecimal)	Feature	Feature name
00008000	High Speed HSM	HSMSpeed2
00100000	Enable high-speed but non-certified random number source for ECDSA operations	FastRandom

Any bits not appearing in the table above are reserved or obsolete and will have no effect on operation.

3.3.3.4. Output of VCM features

The features set on a VCM can be displayed with vcmadmin show and are always displayed as a list of feature names regardless of which option was used to set the features.

An example `vcmadmin show` output would be:

```
FEATURES               :  ForeignTokenOpen|RemoteShare|StandardKM|EllipticCurve|ECCMQV|AcceleratedECC|HSMSpeed2
```

The 32 bit word representing the features can be seen by using the `--json` option with either vcmadmin list or vcmadmin show.

An example `vcmadmin show --json` output would be:

```
          "features": 39459,
```

Due to the internal format used, this JSON formatted output will appear as a decimal number and must be converted to hexadecimal if you wish to use this as an input parameter. For example, 39459 in decimal is 9A23 in hexadecimal.

The tenant will have access to all built-in features regardless of these settings. See Built-in features

3.3.4. VCM speed rating

You can control the relative proportion of available processing capacity that is allocated to an individual HSM using the following speed bits:

- `HSMSpeed0`
- `HSMSpeed1`
- `HSMSpeed2`

This does not provide total control, because the exact allocation of resources depends

upon a number of factors, including the number of active VCMs, the total load of the HSM, and the individual cryptographic operations being performed by each VCM. This means that the VCM speed rating is only an approximation of actual performance.

You cannot set the speed rating of a VCM higher than the speed rating of the HSM on which it is hosted. This means that you cannot use `HSMSpeed3` because it is never set on an HSM.

When hosting VCMs on an nShield 5s Mid-speed HSM, you cannot create a VCM with the `HSMSpeed1` bit set. Requests for VCMs with an `HSMSpeed1` speed bit are automatically converted to `HSMSpeed0` when hosted on a Mid-speed HSM.

3.4. Administration of VCMs

nShield VCMs are administered through the unified utility `vcmadmin`, which directs the command to the service that implements the command.

Some commands require elevated privileges by default because both the permissions and the protection settings have an impact on the usability of the keys by non-administrative users. Commands that create keys or modify configuration always require elevated privileges. Elevated privileges mean `root` on Linux, and the built-in local Administrators group (running in an elevated shell) on Windows. If a command requires elevated privileges, this is indicated in the command description.

You can modify the permissions and protection options on service keys to allow particular groups of users to execute commands that require the private key for a given service. See Permissions on SSH keys and Setting protection on SSH keys.

3.4.1. vcmadmin

The `vcmadmin` utility manages the administration of VCMs using different subcommands.

```
vcmadmin <subcommand>
```

You can use one of the following subcommands each time you run `vcmadmin`:

- create
- start
- stop
- list

- show
- delete
- stats
- properties

3.4.1.1. vcmadmin create

This command creates a VCM and sets the initial sshadmin public key to the one provided by the tenant. An IP address and UUID will be returned which should be sent to the tenant who requested the VCM.

```
vcmadmin create --request REQUESTFILE [-h] [--timeout <TIMEOUT>] [--esn <ESN>] [--verbose] [--json] [--serial-cardreader] [--autostart] [--ip-address IP_ADDDRESS] [--features FEATURES] [--outdir DIR] NAME
```

This command takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	ESN of module on which to create the VCM
`--verbose`	Prints verbose logs.
`--request`	File containing the signed tenant request
`--features`	Features set for the VCM
`--serial-cardreader`	This VCM has access to the HSM card-reader
`--autostart`	This VCM will automatically start after a reboot of the HSM
`--ip-address`	IP address to be allocated to the VCM
`--outdir`	Directory into which the VCM configuration file will be written

3.4.1.2. vcmadmin start

This command starts the specified VCM.

```
vcmadmin start [-h] [--esn <ESN>] [--timeout <TIMEOUT>] [--verbose] [--json] [--request <REQUESTFILE>] UUID
```

If the VCM was created with the `restrict-startup` option, a valid tenant request file will be needed to use this command.

This command takes the following parameters:

Parameter	Description
`--verbose`	Prints verbose logs.
`--json`	Prints the HSM firmware version and image version in JSON format.
`--esn`	ESN of the HSM on which to start the VCM
`--request`	File containing the signed tenant request
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`UUID`	Identifier of the VCM to be started.

3.4.1.3. vcmadmin stop

This command stops the specified VCM.

It is not possible to stop a VCM in the following situations:

- It has an `ncoreapi` service running

In this case you should contact the tenant that is using the VCM and ask them to put the module in maintenance mode by following the instructions at Check and change the mode of operation.

Alternatively you can force the VCM to stop by using the command with the `--force` option.

Forcing a VCM to stop may cause problems for the tenant. You should only do this if you have authority to do so.

```
vcmadmin stop [-h] [--esn <ESN>] [--timeout <TIMEOUT>] [--force] [--verbose] [--json] UUID
```

This command takes the following parameters:

Parameter	Description
`--verbose`	Prints verbose logs.
`--json`	Prints the HSM firmware version and image version in JSON format.
`--esn`	ESN of the HSM on which to start the VCM.
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--force`	Ignore warnings and force VCM to stop.

Parameter	Description
UUID	Identifier of the VCM to be stopped.

3.4.1.4. vcmadmin list

This command lists VCMs that have been created.

```
vcmadmin list [-h] --esn <ESN> [--timeout <TIMEOUT>] [--verbose] [--json] [--active]
```

This command takes the following parameters:

Parameter	Description
--verbose	Prints verbose logs.
--json	Prints metadata in JSON format.
--esn	Lists only VCMs created on the ESN specified. If omitted all VCMs on all ESNs will be listed.
--timeout	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
--active	Lists only those VCMs that are currently running.

3.4.1.5. vcmadmin show

This command shows the properties of VCMs

```
vcmadmin show [-h] --esn <ESN> [--timeout <TIMEOUT>] [--verbose] [--json] uuid <UUID>
```

This command takes the following parameters:

Parameter	Description
--verbose	Prints verbose logs.
--json	Prints metadata in JSON format.
--esn	Shows only VCMs created on the ESN specified. If omitted all VCMs on all ESNs will be shown.
UUID	Identifier of the VCM to be shown.
--timeout	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.

3.4.1.6. vcmadmin delete

This command deletes the VCM specified.

It is not possible to delete a VCM that is currently running. You must first stop the VCM using vcmadmin stop.

It is not possible to delete a VCM in the following situations:

- It has an unfinalized audit log running

In this case you should contact the tenant that is using the VCM and ask them to finalize the audit log as described at disabling audit logging.

Alternatively you can force the VCM to delete by using the command with the `--force` option.

```
vcmadmin delete [-h] [--esn <ESN>] [--timeout <TIMEOUT>] [--verbose] [--json] [--force] UUID
```

This command takes the following parameters:

Parameter	Description
`--verbose`	Prints verbose logs.
`--json`	Prints metadata in JSON format.
`--force`	Forces the VCM to delete even with an unfinalized audit log.
`--esn`	Deletes only VCMs from the ESN specified. If omitted the VCMs will be deleted regardless of ESN.
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`UUID`	Identifier of the VCM to be deleted.

3.4.1.7. vcmadmin stats

This command prints statistics about VCM resource usage.

VCM statistics are gathered by a background process and may reflect the usage up to 90s ago.

```
vcmadmin stats [--timeout <TIMEOUT>] [--esn <ESN>] [--uuid <UUID>] [--json]
```

This command takes the following parameter:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--json`	Prints metadata in JSON format.
`--esn`	Prints statistics only for VCMs from the ESN specified. If omitted statistics for all VCMs will be printed.
`UUID`	Prints statistics only for the VCM specified by this identifier.

3.4.1.8. vcmadmin properties

This command manages the properties of VCMs.

```
vcmadmin properties <subcommand>
```

You can use one of the following subcommands with this command:

- set
- list

3.4.1.8.1. vcmadmin properties set

This subcommand sets properties for a VCM.

```
vcmadmin properties set [-h] --request <REQUESTFILE> UUID PROPERTY VALUE
```

This subcommand takes the following parameters:

Parameter	Description
`request`	File containing the tenant request
`UUID`	Identifier of the VCM.
`PROPERTY`	VCM property to be set
`VALUE`	Value to be set for PROPERTY

3.4.1.8.2. vcmadmin properties list

This subcommand lists the VCM properties that can be set by the `set` command. To see values currently set on a given VCM use the show command or the list command using the `--json` option.

```
vcmadmin properties list [-h]
```

4. Management of nShield 5s HSMs

4.1. Upgrade firmware: nShield 5s

This section describes how to upgrade firmware on your nShield HSM hardware security module.

4.1.1. Primary, recovery and bootloader firmware

HSM firmware consists of three major components:

- Primary image firmware
- Recovery image firmware
- Bootloader firmware

Upgrade packages may contain updates for any of these components. The same upgrade method is used in all cases. The system will automatically detect which components are included in the update package and will load the firmware to the correct location.

If upgrade packages are available for both Primary and Recovery firmware it is not recommended to upgrade them both at the same time. The recommended procedure is to always upgrade the Primary firmware first. Test that the system performs as expected and then upgrade the Recovery firmware at a later date.

4.1.2. Firmware version control

The version of Primary and Recovery image firmware that can be installed on an HSM is controlled by the Version Security Number, see Version Security Number.

Bootloader firmware version control is described in Bootloader version.

4.1.2.1. Version Security Number (VSN)

Entrust supply several versions of the module firmware. Primary and Recovery image firmware includes a Version Security Number (VSN). This number is increased whenever Entrust improve the security of the firmware. Ensuring you use firmware with the highest available VSN allows you to benefit from these security improvements.

However, if you have a regulatory requirement to use certified firmware such as that approved by FIPS or Common Criteria, you should only install the latest available firmware

that has been certified by the relevant certification authority. This firmware may not have the highest VSN available.

Every HSM records the minimum firmware VSN that it will accept. You can always upgrade to firmware with an equal or higher VSN than the minimum VSN set on your module, even if the firmware currently installed on the module has a higher VSN than the firmware to which you are upgrading.

You can upgrade to a firmware version with a higher VSN than the HSM's current firmware, without committing yourself to the upgrade, by installing the newer firmware without altering the HSM's minimum VSN requirement. The older firmware can be reinstalled at any time provided the HSM's minimum VSN has not been altered.

You can never load firmware with a lower VSN than the target HSM's minimum VSN requirement. For example, if the HSM has a minimum VSN requirement of 3 and the currently installed firmware has a VSN of 4, you can install firmware with a VSN of 3 or above to the HSM. You cannot install firmware with a VSN of 1 or 2 to this HSM.

4.1.2.1.1. Configuring the minimum VSN

To increase the HSM's minimum VSN requirement, use the command hsmadmin setminvsn. The new VSN must be greater than or equal to the HSM's current minimum required VSN, and cannot be greater than the VSN of the firmware currently installed on the HSM.

It is recommended that the `hsmadmin setminvsn` command always be used as soon as the decision has been made not to return to the older version of the firmware. This prevents future downgrades of the firmware that could potentially weaken security.

4.1.2.2. Bootloader version

Bootloader firmware does not have a VSN. The Bootloader version number is included in the filename of the upgrade package. Entrust recommend that you always install the latest version available.

For security reasons some Bootloader firmware upgrades are irreversible. These Bootloader upgrades revoke the signing key used to sign previous Bootloader firmware and thus it is not possible to revert back to previous Bootloader firmware after such an upgrade.

Refer to the release notes accompanying the firmware release to identify whether the Boot loader upgrade is reversible or not.

It is only possible to update the Bootloader when running firmware ver-

> sion 13.4 or later.

4.1.3. Firmware on the installation media

Your Firmware installation media may contain several sets of firmware for each supplied product. These can include the:

- latest FIPS approved firmware
- latest Common Criteria approved firmware
- latest firmware available

You should ensure you are using the latest firmware available, unless you have a regulatory requirement to use firmware that has been certified by a specific certification authority.

4.1.3.1. Recognising firmware files

The firmware files are stored in subdirectories within the `firmware` directory on the installation media. The subdirectories are named by product and then certification status, which can be `latest`, `fips-pending`, `fips`, or `cc`.

Firmware files for nShield HSM modules have a `.npkg` filename suffix.

The VSN of a Primary or Recovery image firmware file is incorporated into its filename and is denoted by a dash and the letters "vsn" followed by the digits of the VSN. For example, `-vsn24` means the VSN is 24.

To display information about a firmware file on the installation media, enter the following command:

Linux

```
hsmadmin npkginfo /disc-name/firmware/nShield5s/status/firmware_file.npkg
```

In this command, *disc-name* is the directory on which you mounted the installation media, *status* is the certification status, and *firmware_file* is the file name.

Windows

```
hsmadmin npkginfo E:\firmware\nShield5s\status\firmware_file.npkg
```

In this command, *E* is the drive letter of your installation media, *status* is the certification status, and *firmware_file* is the file name.

4.1.4. Firmware installation overview

Normal procedure is to install firmware when the HSM is running in primary mode. If the HSM is running in recovery mode, as described in Recovery Mode the procedure is identical except that the reboot caused by `hsmadmin upgrade` will cause the module to factory state and it will be necessary to run `hsmadmin enroll` before continuing with the rest of the instal lation.

If you are upgrading a module which has SEE program data or NVRAM-stored keys in its nonvolatile memory, use the `nvram-backup` utility to backup your data first.

If the HSM to be upgraded is part of an audit logging Security World you will need to finalize the audit log before starting the upgrade. See audit logging and firmware upgrade for information on how to do this.

You should always check that the system clock is correct before upgrad ing the firmware and adjust it if necessary, see Setting the system clock.

1. Put the module in Maintenance mode.

   See Check and change the mode of operation.

2. Check the version of the firmware currently loaded, see hsmadmin status.
3. View information about the firmware in the upgrade file including the version and the VSN, see hsmadmin npkginfo.
4. Optionally make a dry-run of the upgrade by using hsmadmin upgrade with the `--dry -run` option. This will check that everything is in place for the upgrade to succeed but will not upgrade the firmware. If any errors are reported fix these before continuing to the next step.
5. Upgrade the firmware, see hsmadmin upgrade.

   The current firmware version and the firmware version being loaded will be displayed automatically.

   The module will be programmed with the new firmware and will be automatically rebooted.

   If the installation is being run from recovery mode this reboot will factory state the HSM and `hsmadmin enroll` must be run before continuing.

   The module will report which internal components of the firmware

> have been updated. These components are pre-determined by the individual upgrade file and the internal names are intended for use by Entrust support staff only.

6. Check the version of the firmware now loaded, see hsmadmin status.
7. Put the module in initialization mode.

   See Check and change the mode of operation.

8. Restore the HSM to the Security World, see Adding or restoring an HSM to the Security World

   > If the HSM is not part of a Security World you can use the command `initunit` instead of this step.

9. Put the module in Operational mode.

   See Check and change the mode of operation.

10. Run the `enquiry` command to verify the module is in operational state and has the correct firmware version.

    In Operational mode, the `enquiry` command shows the version number of the firmware loaded. This is the `version` field listed per module.

4.2. Setting the system clock

This guide covers the following HSMs:

- nShield 5s

Entrust recommends that you set the HSM system clock before performing any other actions. This is because the HSM clock may have drifted from real time whilst the HSM was running on battery power in storage.

> The HSM system clock is set by fetching the current time and date from the host machine in which the HSM is fitted. Therefore it is important to check that the time and date is set correctly on the host machine.

> Initial setting of the HSM system clock should be performed with the HSM in maintenance mode. If your HSM is not in maintenance, you must put it into maintenance mode. For instructions, see nShield 5s modes of operation.

4.2.1. Setting the HSM system clock

1. Make sure that the date and time on the host machine are set correctly according to the documentation for the operating system on the host machine.
2. Run the following command as a user with `root` privileges on Linux or the privileges of the built-in local Administrators group on Windows:

```
/opt/nfast/bin/hsmadmin settime
```

The settime command uses UTC date and time format.

When you are setting time at the very first time on an nShield 5s HSM, it is recommended to avoid the optional `--adjust` parameter. This parameter is intended to be used when the HSM is already in operational mode. It can be used on a periodic basis to gradually reconcile any discrepancies between the host's and the HSM's clocks. Gradual reconciliation prevents sudden time discrepancies and ensures smooth operation.

The HSM's date and time are used to validate security certificate expiry dates and to provide accurate timestamps for system and audit logs. Thus ensuring that an HSM's system clock is closely synchronized with an external time source is critical for maintaining robust security and audit capabilities.

The external time source used is the clock of the host PC in which the HSM is installed.

Make sure that the date and time on the host machine are set correctly according to the documentation for the operating system on the host machine.

The system clock should have been set as part of the installation process, see Setting the system clock but the clocks on the HSM and the host PC can drift apart over time due to hardware and environmental differences and must be periodically adjusted to keep them in synchronization.

The system clock may also be incorrect if the HSM has been running on battery power for some time.

4.2.2. System interaction with the system clock

It is important that the system clock is accurate so that timestamps in the system logs can be correlated across the whole network in which the HSM is operating.

For HSMs running firmware version 13.5 or later, if the system clock is lost, for instance due to the HSM running on battery power for an extended period of time, the following adminis trator actions will be prohibited until the system clock is restored:

- Factory state, see Return to Factory State
- Firmware upgrade, see Firmware upgrade
- Setting the minimum VSN, see Version Security Number
- Setting SSH keys (unless running in recovery mode), see Set up communication between host and module (nShield 5s HSMs)
- Log expiry, see Expiring signed logs
- Log export, see Retrieving signed logs
- Adjusting system time, see Adjusting the system clock
- Adminstration of CodeSafe, see The csadmin tool

If you cannot communicate with the HSM because the system clock is lost, you must be in Recovery mode to reset the system time. See Setting the system clock for more informa- tion on setting the system clock.

4.2.3. Checking the system clock

The system clock can be checked using the command gettime.

If the system clock is incorrect it can be set by using the command settime.

For small errors in the system clock it is recommended to adjust the clock as described in Adjusting the system clock. For larger errors in the system clock it may be necessary to use the procedures described in Setting the system clock since use of the `--adjust` option would result in the clock being incorrect for the long period of time required to converge the clocks.

Following the procedures in Setting the system clock will require the HSM to be taken out of operational mode.

4.2.4. Adjusting the system clock

The system clock is adjusted by using the `settime` command with the `--adjust` command line option. This will gradually reduce any difference in time between the host and the HSM's clocks, preventing large jumps or discontinuities in time.

Use of the `--adjust` parameter allows the system time to be adjusted whilst the HSM is in operational mode.

The procedure gradually converges the clocks at the rate of a few seconds per day and thus it may take a long time to correct a large error.

When you execute `hsmadmin settime adjust`, the command immediately returns `HSM system time calibration in progress` to acknowledge that the calibration process has started. There is no notification when the calibration is complete.

The frequency at which the `hsmadmin settime --adjust` command should be used depends on multiple factors, for example the precision of the internal clock of the HSM host PC and the extent of drift between the host's clock and the HSM's clock.

The recommended starting point for most systems would be to issue the command once per day. Experimentation is required to find the optimal frequency.

4.2.4.1. Restrictions on setting the clock

To prevent malicious actors from tampering with the HSM's system clock by moving it back ward, the HSM is designed to prevent the setting of a time and date that is earlier than a previously set time and date. This ensures that the HSM's system clock remains secure and accurate and helps prevent unauthorized access that could occur if the system clock were tampered with.

When the `settime` command is issued without the `--adjust` parameter, the new time is saved within the HSM. Subsequent `settime` commands are prohibited from setting a time earlier than this saved time.

It is only possible to set the HSM system clock to an earlier time than the saved time by returning the HSM to factory state first, see Return to Factory State. This will erase the saved time within the HSM and allow you to issue the `settime` command without restriction.

As a security measure it is never possible to set the HSM clock to a time earlier than its manufacturing time. The manufacturing time can be obtained with the command `hsmadmin info` and is shown as the `mfgtime` entry.

Setting the system date and time automatically resets the HSM.

4.3. Checking the installation

This guide covers the following HSMs:

- nShield 5s
- nShield Solo
- nShield Solo XC

This guide describes what to do if you have an issue with the module or the software.

The facilities described below are only available if the software has been installed successfully. If the software has not installed correctly see, Problems during installation and commissioning.

4.3.1. Checking operational status

4.3.1.1. Enquiry utility

Run the `enquiry` utility to check that the module is working correctly. You can find the `enquiry` utility in the `bin` subdirectory of the `nCipher` directory. This is usually:

- `C:\Program Files\nCipher\nfast` for Windows
- `/opt/nfast` for Linux

If the module is working correctly, the `enquiry` utility returns a message similar to the follow ing:

nShield 5s

```
Server:
enquiry reply flags  none
enquiry reply level  Six
serial number        #############-####
mode                 operational
version              #.#.#
speed index          ###
rec. queue           ##..##
...
module type code     0
product name         nFast server
...
Module ##:
enquiry reply flags  none
enquiry reply level  Six
serial number        #############-####
mode                 operational
version              #.#.#
speed index          ###
rec. queue           ##..##
...
module type code     14
product name         #######/#######
...
rec. LongJobs queue  ##
SEE machine type     None
```

```
supported KML types  DSAp1024s160 DSAp3072s256
active modes         none
physical serial      48-U50104
hardware part no     PCA10005-01 revision 03
hardware status      OK
```

nShield Solo

```
Server:
enquiry reply flags    none
enquiry reply level    Six
serial number          #############-####
mode                   operational
version                #.#.#
speed index            ###
rec. queue             ##..##
...
version serial         #
remote server port     ####
...
module type code       0
product name           nFast server
...
Module ##:
enquiry reply flags    none
enquiry reply level    Six
serial number          #############-####
mode                   operational
version                #.#.#
speed index            ###
rec. queue             ##..##
...
module type code       7
product name           #######/#######/#######
...
rec. LongJobs queue    ##
SEE machine type       Power PCSXF
supported KML types    DSAp1024s160 DSAp3072s256
hardware status        OK
```

nShield Solo XC

```
Server:
enquiry reply flags    none
enquiry reply level    Six
serial number          #############-####
mode                   operational
version                #.#.#
speed index            ###
rec. queue             ##..##
...
module type code       0
product name           nFast server
...
version serial         #
remote server port     ####

Module ##:
enquiry reply flags    none
enquiry reply level    Six
serial number          #############-####
mode                   operational
version                #.#.#
```

```
speed index             ###
rec. queue              ##..##
...
module type code        12
product name            #######/#######/#######
...
rec. LongJobs queue     ##
SEE machine type        Power PCELF
supported KML types     DSAp1024s160 DSAp3072s256
hardware status         OK
```

If the mode is operational the module has been installed correctly.

If the mode is initialization or maintenance, the module has been installed correctly, but you must change the mode to operational.

If the output from the `enquiry` command says that the module is not found, first restart your computer, then re-run the `enquiry` command.

If the operating system supports power saving, disable power saving. See Install a PCIe HSM for more information. Otherwise, if your system enters Sleep mode, the HSM may not be found when running `enquiry`. If this happens, you need to reboot your system.

4.3.1.2. nFast server (hardserver)

Communication can only be established with a module if the nFast server is running. If the server is not running, the `enquiry` utility returns the message:

```
NFast_App_Connect failed: ServerNotRunning
```

Restart the nFast server, and run the `enquiry` utility again. See Stopping and restarting the hardserver for more about how to restart the nFast server.

4.3.2. Mode switch and jumper switches (nShield Solo and Solo XC only)

The mode switch on the back panel controls the mode of the module. See Checking and changing the mode on an nShield Solo module for more about checking and changing the mode of an HSM. You can set the physical mode override jumper switch on the circuit board of the nShield Solo to the **On** position, to prevent accidental operation of the mode switch. If this override jumper switch is on, the nShield Solo and nShield XC will ignore the position of the mode switch (see Back panel and jumper switches).

You can set the remote mode override jumper switch on the circuit board of the nShield Solo and nShield Solo XC to the **On** position to pre vent mode change using the `nopclearfail` command. This should be done if, for example, the security policies of your organization require the physical mode switch to be used to authorize mode changes.

4.3.3. Log message types

By default, the hardserver writes log messages to:

- The in Windows Operating System event log.
- `log/logfile` in the `nCipher` directory (normally `opt/nfast/log` directory) on Linux. The environment variable `NFAST_SERVERLOGLEVEL` determines what types of message you see in your log. The default is to display all types of message.

  `NFAST_SERVERLOGLEVEL` is a legacy debug variable.

4.3.3.1. Information

This type of message indicates routine events:

```
nFast Server service: about to start
nFast Server service version starting
nFast server: Information: New client clientid connected
nFast server: Information: New client clientid connected - privileged
nFast server: Information: Client clientid disconnected
nFast Server service stopping
```

4.3.3.2. Notice

This type of message is sent for information only:

```
nFast server: Notice: message
```

4.3.3.3. Client

This type of message indicates that the server has detected an error in the data sent by the client (but other clients are unaffected):

```
nFast server: Detected error in client behaviour: message
```

4.3.3.4. Serious error

This type of message indicates a serious error, such as a communications or memory failure:

```
nFast server: Serious error, trying to continue: message
```

If you receive a serious error, even if you are able to recover, contact Support.

4.3.3.5. Serious internal error

This type of message indicates that the server has detected a serious error in the reply from the module. These messages indicate a failure of either the module or the server:

```
nFast server: Serious internal error, trying to continue: message
```

If you receive a serious internal error, contact Support.

4.3.3.6. Start-up errors

This type of message indicates that the server was unable to start:

```
nFast server: Fatal error during startup: message nFast Server service version failed init.
nFast Server service version failed to read registry
```

Reinstall the server. If this does not solve the problem, contact Support.

4.3.3.7. Fatal errors

This type of message indicates a fatal error for which no further reporting is available:

```
nFast server: Fatal internal error
```

or

```
nFast server: Fatal runtime error
```

If you receive either of these errors, contact Support.

4.3.4. BadTokenData error (Solo only)

The PCIe module (not the Solo XC module) is equipped with a rechargeable backup battery

for maintaining Real-Time Clock (RTC) operation when the module is powered down. This battery typically lasts for two weeks. If the module is without power for an extended period, the RTC time is lost. When this happens, attempts to read the clock (for example, using the `ncdate` or `rtc` utilities) return a `BadTokenData` error status.

The correct procedure in these cases is to reset the clock and leave the module powered up for at least ten hours to allow the battery to recharge. No other nonvolatile data is lost when this occurs. See rtc for more about resetting the clock.

The Solo XC module is equipped with a battery with a ten year life for maintaining RTC operation when the module is powered down. The RTC will not require resetting after the module has been shut down for extended periods. The battery is not rechargeable.

4.4. HSM status indicators and error codes (nShield 5s)

This guide covers the following HSMs:

- nShield 5s

The Entrust nShield 5s HSM is fitted with a tri-color LED on the back panel. This LED will typically indicate the operational state of the HSM, see LED status. However, the LED can also indicate if the HSM is in an unrecoverable error state, see LED error states. Unrecoverable error state codes can also be retrieved remotely using the `enquiry` utility, see Error codes accessed remotely.

4.4.1. LED status

The Entrust nShield 5s HSM is fitted with a tri-color LED on the back panel. This LED typically indicates the status of the HSM.

The following states indicate a normal operational state:

Colour	Pattern	Meaning
N/A	Blank	No power or processors not working.
Green	Solid	Power is good. Main processor has not started booting.
Cyan	Solid	Main processor is booting.
Cyan/Blue	Slow flash	Security processor firmware upgrade in progress.
Blue	Solid	System has booted, now idle.
Blue	Flickering	System is active - normal operation.

The following states indicate an error within the HSM:

Colour	Pattern	Meaning
Blue	Morse code	Error state for when the HSM is in an unrecoverable state, see LED error states for more information.
Red	Morse code	Error state for when the HSM is in an unrecoverable state see LED error states for more information.
Red	Fast flash	Security processor bootloader failure.
Blue/Red	Flash	Security processor detected a tamper condition.
Other	Any	Contact Entrust Support.

4.4.2. LED error states

If the Entrust nShield 5s HSM encounters an unrecoverable error, it enters an error state. In an error state, the HSM does not respond to commands and does not write data to the bus. The LED displays a Morse code pattern to indicate a specific error state, see Error codes shown on the LED.

In some cases you can reset an HSM in an error state by powering down the HSM and then reapplying power, or with `hsmadmin reset`. Not all errors can be reset in this way.

If any HSM goes into an error state, except as a result of you issuing the `nopclearfail --fail` command, contact Entrust Support, and give full details of your HSM set-up and the error code.

Entrust recommends that you contact Entrust Support even if you successfully recover from the error.

For troubleshooting information, see Troubleshooting 5s.

4.4.2.1. Error codes shown on the LED

If an HSM enters an error state, the LED flashes with a Morse code pattern corresponding to an error code.

Error codes can also be retrieved remotely using the `enquiry` utility, see Error codes accessed remotely.

All the LED error codes have three digits:

- The first digit is indicated by a number of dots.

- The second digit is then indicated by a number of dashes.
- The third digit is then indicated by a number of dots.

There is then a longer gap and the error code repeats.

The following guidelines are useful when reading LED code messages from the HSM:

- The duration of a dash (-) is three times the duration of a dot (.).
- The gap between components of a letter has the same duration as a dot.
- The gap between digits has the same duration as a dash.
- The duration of the gap between repeating codes is seven times the duration of a dot.

The numbers of dots/dashes and the Morse code equivalent is shown in the table below.

Colour	Digits	Dots and dashes	Morse code	Meaning
Red	1-1-1	. - .	E T E	Battery voltage out of spec
Red	1-2-1	. - - .	E M E	Crypto SerDes core voltage out of spec
Red	1-2-2	. - - . .	E M I	Main processor SerDes core voltage out of spec
Red	1-2-3	. - - . . .	E M S	Main processor core voltage out of spec
Red	1-2-4	. - -	E M H	Main processor SerDes core IO voltage out of spec
Red	1-2-5	. - -	E M 5	Crypto SerDes IO voltage out of spec
Red	1-3-1	. - - - .	E O E	Main processor IFC IO voltage out of spec
Red	1-3-2	. - - - . .	E O I	DDR access voltage out of spec
Red	1-3-3	. - - - . . .	E O S	DDR IO voltage out of spec
Red	1-3-4	. - - -	E O H	V12 voltage out of spec
Red	1-3-5	. - - -	E O 5	Security processor voltage out of spec
Red	1-5-1	. - - - - - .	E 0 E	Security processor temperature out of spec
Red	1-5-2	. - - - - - . .	E 0 I	Main processor temperature out of spec
Red	1-5-3	. - - - - - . . .	E 0 S	Crypto temperature out of spec
Red	1-5-4	. - - - - -	E 0 H	Security processor app blank
Red	1-5-5	. - - - - -	E 0 5	Security processor app invalid
Red	2-1-1	. . - .	I T E	Security processor secure state corrupted
Red	2-1-2	. . - . .	I T I	No bootloader heartbeat

Colour	Digits	Dots and dashes	Morse code	Meaning
Red	2-1-3	. . - . . .	I T S	Board-ID PROM failed
Blue	2-1-5	. . -	I T 5	Firmware signature auth failure
Red	2-2-2	. . - - . .	I M I	Crypto known-answer tests failed
Red	2-2-3	. . - - . . .	I M S	RNG driver failed
Red	2-2-4	. . - -	I M H	FIPS DRBG failed
Red	2-2-5	. . - -	I M 5	OpenSSL failed
Red	2-3-1	. . - - - .	I O E	OpenSSH failed
Red	2-3-2	. . - - - . .	I O I	Library signature verification failed
Red	2-3-3	. . - - - . . .	I O S	FPGA initialisation failed
Red	2-3-4	. . - - -	I O H	Init script failed
Red	2-3-5	. . - - -	I O 5	An unknown error occurred
Red	2-5-1	. . - - - - - .	I 5 E	Error playing LED sequence
Red	2-5-2	. . - - - - - . .	I 5 I	runleveld crashed
Red	2-5-3	. . - - - - - . . .	I 5 O	SPI interface failed consecutively 15 times
Red	2-5-4	. . - - - - -	I 5 H	envmon crashed
Red	2-5-5	. . - - - - -	I 5 5	Log volume has reached critical threshold
Red	3-1-1	. . . - .	O E E	Uboot PCIe PLL lock failed
Red	3-1-2	. . . - . .	O E I	Uboot DRAM init failed

4.4.3. Error codes accessed remotely

If an HSM enters an error state, you can retrieve error codes using the `enquiry` utility. These codes appear in the `hardware status field` of the `Module` and are included in the hard-server log.

There are three error categories:

- Runtime library errors.
- Hardware driver errors.
- Operational mode errors.

Error codes are also indicated by the LED on the back of the HSM, see LED error states.

4.4.3.1. Runtime library errors

The runtime library error codes described in the following table indicate one of the following:

- There is a bug in the firmware.
- There is a hardware fault.

If any of these errors occur, reset the HSM.

Code	Meaning
O L C	SIGABRT: assertion failure and/or `abort()` called.
O L D	Interrupt occurred when disabled. This is more likely to indicate a hardware problem than a firmware problem.
O L E	SIGSEGV: access violation. This is more likely to indicate a hardware problem than a firmware problem.
O L J	SIGFPE: unsupported arithmetic exception (such as division by 0).
O L K	SIGOSERROR: runtime library internal error.
O L L	SIGUNKNOWN: invalid signal raised.

4.4.3.2. Hardware driver errors

The hardware driver error codes described in the following table indicate one of the following:

- Some form of automatic hardware detection has failed.
- There is a bug in the firmware.
- The wrong firmware has been loaded.

If any of these errors is indicated, contact Entrust Support.

Code	Meaning
H L	M48T37 NVRAM (or battery) failed.
H C V	CPLD wrong version for PCI policing firmware.
H C X	No crypto offload hardware detected.

Code	Meaning
H P P	PCI Interface Policing failure.
H V	Environment sensors failed. For example, the temperature sensor.
H D	Failure reading unique serial number.
H R	Random number generator failed.
H R F O	FIPS continuous RNG failed.
H R A O	Periodic RNG test failed.
H R S	RNG startup failed.
H R T	RNG selftest failed.
H R T P	Periodic (scheduled daily) RNG selftest failed.
H R M	RNG data matched.
H R Z	Impossible RNG Failure (match after PRNG).
H S S	Security processor internal semaphore error.
H O	Token interface initialization failed.
H E	EEPROM failed on initialization.
H C	Processing thread initialization failed.
H C P	Card poll thread initialization failed.
H F	Starting up crypto offload.
H C V	CPLD version number incorrect.
H J V	IPC-watcher failed.
H J U	IPC-EPD failed.
H J R	Module reset notification failed.
K R	RSA selftest failed.
H H D	Unique serial number detection failed.
H H P	PCI bus hardware detection failed.
H H R	RTC hardware detection failed or random number generator detection failed.
H S C	Error writing correct SOS message.

4.4.3.3. Operational mode errors

The runtime library error codes described in the following table indicate one of the follow-

ing:

- There is a bug in the firmware.
- There is a hardware fault.

Code	Meaning	Action
T	Temperature of the HSM has exceeded the maximum allowable.	Restart your host computer, and improve HSM cooling. For the cooling requirements for your HSM, see Prerequisites and product information.
D	Fail command received.	Reset HSM by turning it off and then on again.
G G G	Failure when performing `ClearUnit` or `Fail` command.	Contact Entrust Support.
I J A	Audit logging: failed to send audit log message. This can occur for any type of log message. That is, a log message, signature block or certifier block.	Contact Entrust Support.
I J B	Audit logging: no module memory (therefore failed to send audit log message).	Contact Entrust Support.
I J C	Audit logging: key problem or FIPS incompatibility (therefore failed to sign audit log message).	Contact Entrust Support.
I J D	Audit logging: NVRAM problem (therefore failed to configure or send audit log message).	Contact Entrust Support.

4.5. nShield 5s platform modes

This chapter describes the nShield 5s platform modes of operation:

- Normal operation
- Return to factory state
- Recovery mode

4.5.1. Modes of operation

The status of the nShield 5s HSM can only be one of the following:

Status	Description
Primary mode	The nShield 5s HSM is running on the primary firmware image. This is the normal operational mode.
Recovery mode	The nShield 5s HSM is running on the recovery image instead of the primary image. See Recovery mode.
Factory state	The nShield 5s HSM is in a factory state. See Return to factory state.

4.5.2. Normal operation

In normal operation the nShield 5s HSM will be running the primary firmware image. In a multi-tenant system there will be `ncoeapi` service running at platform level.

The `ncoreapi` service runs only inside a VCM and it will be necessary to create at least one VCM in order to have access to `ncoreapi` services.

4.5.3. Return to factory state

nShield 5s HSMs that are delivered from the factory contain no data relating to the `ncore-api` service. A small amount of 'lifetime' data, which is used by the platform services, is pre-installed. This data is for personalisation and identification of the individual HSM, such as its ESN.

You can perform a reset operation that returns the data stored in an HSM to the state it was in when it left the factory. This erases user credentials and information, leaving only the 'lifetime' data.

When an HSM is in this state it will not support any user commands other than hsmadmin enroll and it will be necessary to follow the process described in Installation of SSH keys before any further actions can be taken.

Returning to factory state will erase any optional features that were not installed at the factory. See, Optional features.

Returning to factory state will change the key used to sign system logs. You should make a record of the new log verification key as soon as pos sible after returning an HSM to factory state. See Verifying Signed Logs for more information. Signed system logs are only available from firmware version 13.5 onwards so this is not necessary for HSMs running older firmware.

4.5.3.1. Purpose of factory state

The main reason for returning an nShield 5s HSM to factory state is to securely erase all user secrets. This is important when, for example:

- The HSM is being taken out of service.
- The HSM is being moved from one domain to another, where it is important to ensure that there is no possibility of secrets being leaked between domains.
- The HSM is being returned to Entrust for servicing or warranty.
- You have lost the SSH keys used to communicate with the HSM, see Recovery from loss of SSH keys

4.5.3.1.1. Recovery from loss of SSH keys

Returning a unit to factory state will be necessary if you have lost possession of the SSH keys used to communicate with the HSM and you have not previously made a backup of those keys with hsmadmin keys backup (or `hsmadmin keys backup --passphrase` if the HSM is being re-installed in a different machine). If this happens, returning the HSM to factory state will allow hsmadmin enroll to successfully create new keys and re-establish communication with the HSM.

4.5.3.2. Enter and exit the factory state

The nShield 5s HSM can be returned to factory state in one of two ways. Either by use of hsmadmin factorystate or by placing the HSM in Recovery mode.

If the SSH keys used to communicate with the HSM have been lost, only the Recovery mode option is possible. Both of the above methods include a reboot of the HSM.

The command hsmadmin factorystate is prohibited if the system logs have exceeded a maximum size, see maximum log size or if the system clock is invalid, see System interaction with the system clock. In these situations you can only return to factory state by placing the HSM in Recovery mode.

The HSM is taken out of factory state by use of hsmadmin enroll.

4.5.4. Recovery mode

nShield 5s HSMs are loaded with two different firmware images:

- The Primary image.
- The Recovery image.

During normal operation, the HSM is running firmware that is loaded from the Primary image.

If required, the HSM can be forced into recovery mode to run firmware loaded from the Recovery image. Entry into recovery mode performs the same actions as `hsmadmin factorystate`

Recovery mode is useful in the following cases:

- To return the HSM to a known good state for disaster recovery.
- To retrieve the `init` log if the HSM fails to boot into primary mode, see Retrieving the init log
- To clear the system log if the HSM is prohibiting actions because it has exceeded the maximum log size, see Maximum log size
- To restore communication with the HSM if the SSH keys have been lost and no backup is available, see Set up communication between host and module (nShield 5s HSMs).
- To restore communication with the HSM if an invalid system clock is preventing you from modifying the SSH keys in primary mode. See System interaction with the system clock.

4.5.4.1. Restrictions in recovery mode

The main purpose of recovery mode is to allow essential maintenance activities that are not possible in when the nShield 5s is running the primary image firmware.

The `ncoreapi` and `launcher` services don't run when the nShield 5s is in recovery mode. Only the platform services are available, meaning that only the commands described in Administration of platform services (nShield 5 HSMs) are available.

If you run `hsmadmin enroll` in recovery mode, a warning will appear. This is because the certificates for the SSH keys described in Set up communication between host and module (nShield 5s HSMs) are not created in recovery mode. You can ignore this warning.

Commands that use `ncoreapi` or `launcher` service do not run and may show error messages.

4.5.4.2. Entry into recovery mode

Boot the nShield 5s HSM into recovery mode by holding down the recovery mode button on the back panel of the HSM and then rebooting the HSM. You must continue holding

down the button for 60 seconds after initiating the reboot. The button is non-latching.

You must hold down the recovery mode button while the HSM is reboot ing. If you reboot the HSM and then press and hold down the button, you will miss the part of the reboot process in which you can change the mode of the HSM.

See Install a PCIe HSM for the location of the recovery mode button. You can trigger a reboot with `hsmadmin reset` or by power cycling the host machine containing the HSM.

If you cannot reach the recovery mode button and enter the reboot command simultaneously, you might need to connect a keyboard, mouse, and monitor to the back of the server hosting the HSM. If this is not possible, you need a second person to pass the command to the HSM while you hold down the button, or to hold down the button while you pass the command.

Entering and exiting recovery mode return the HSM to factory state. You must run `hsmadmin enroll` after the boot has completed before any further actions can be performed.

Run `hsmadmin status` to verify that the HSM is in recovery mode. If you are still in primary mode, try the process again, making sure that the recovery mode button is pressed down before or as soon as the reboot command is passed, and that it is held for the allotted time.

4.5.4.3. Exit from recovery mode

Exit recovery mode by booting the nShield 5s HSM without the recovery mode button held down. If the firmware is changed whilst in recovery mode using `hsmadmin upgrade`, the unit automatically reboots.

When the unit next boots into primary mode it will be in factory state. You must run `hsmadmin enroll` again before any further actions can be performed.

If you exited recovery mode using `hsmadmin reset`, or as part of a firmware upgrade, you must restart the hardserver/nFast server after running `hsmadmin enroll`.

Run `hsmadmin status` to verify that the HSM is in the correct mode.

4.6. Platform services (nShield 5 HSMs)

The nShield HSM firmware provides multiple services which manage different parts of the system. Each service has its own SSH keys that allow communication with the service, see separation of services.

This allows you to partition the users of the system into different groups and restrict certain user groups to the use of certain services by restricting who has access to the relevant keys.

There are two major groups of services:

- Platform services
- End-user services

Platform services are used to perform the tasks associated with the installation, commissioning, and maintenance of the HSM firmware and hardware.

In a multi-tenant system this would be the responsibility of the Service Provider.

There will only ever be one instance of each platform service running at any one time.

End-user services are used to provide cryptographic services to the end-user. If your firmware supports multi-tenancy then there could be multiple instances of end-user services running concurrently.

In a multi-tenant system the end-user would be the tenant. The tenant will also have access to other services that are needed to manage their tenancy.

4.6.1. End-user services

4.6.1.1. ncoreapi service

The `ncoreapi` service provides cryptographic services to the end user. This can either be via custom applications created by the end user accessing services using the `ncoreapi` service, as described in *nCore API Documentation* and *Cryptographic API*, or by using the utilities provided on the installation media.

4.6.1.2. monitor service

This service provides functions to retrieve and clear logs stored within a VCM.

4.6.1.3. sshadmin service

This service provides functions to manage the SSH keys used by the end-user services within a VCM.

4.6.2. Platform services

4.6.2.1. updater service

This services provides functions to upgrade the HSM firmware.

4.6.2.2. setup service

This service provides functions to view information about the HSM, to configure the HSM and to return the HSM to factory settings.

4.6.2.3. monitor service

This service provides functions to retrieve and clear logs stored within the HSM.

4.6.2.4. sshadmin service

This service provides functions to manage the SSH keys used by the platform services. If your system has not been configured for multi-tenancy the `sshadmin` service also manages the keys for the `ncoreapi` service.

4.6.2.5. launcher service

On versions with CodeSafe 5 support, this is used for starting CodeSafe 5 applications on the HSM.

4.6.2.6. orchestrator service

This service is used to manage VCMs.

4.6.3. Administration of platform services

The administration of platform services is described in Administration of platform services (nShield 5 HSMs)

4.6.4. Service interlock

An interlock mechanism prevents most platform services from being accessed whilst the `ncoreapi` service is in operational mode:

- Non-invasive services that only access information, such as log retrieval or a firmware version check, can be used while `ncoreapi` is running.
- Invasive services that would change the platform's state, such as log clearing or firmware updates, cannot be used while `ncoreapi` is running.

To access invasive platform services the `ncoreapi` service must be put into maintenance mode using `nopclearfail -M -m <MODULEID> -w`.

For example:

```
>nopclearfail -M -m 1
Module 1, command ClearUnitEx: OK
```

In a multi-tenant system a similar interlock mechanism exists preventing some platform operations whilst an `ncoreapi` service is running in a VCM. If you receive such an error message you should ask the tenant to put the VCM into maintenance mode using `nopclearfail -M -m <MODULEID> -w`.

4.6.5. Separation of services

Each service has its own communication channel with the host PC that is protected by use of SSH encryption. The procedure for installing the necessary SSH keys for platform services is described in Set up communication between host and module (nShield 5s HSMs). If your system has not been configured for multi-tenancy this procedure will also install the SSH keys for the end-user services.

In a multi-tenant system the procedure for installing the necessary SSH keys for end-user services is described in hsmadmin vcm enroll.

4.7. Set up communication between host and module (nShield 5s HSMs)

4.7.1. Overview of SSH keys

Communications between the host and the HSM are protected by use of SSH secure channels. To allow mutual authentication of the endpoints, the SSH protocol uses separate key pairs in the host and the HSM. The functionality within the HSM is divided into different services that use separate SSH channels. See Platform services (nShield 5 HSMs). You need to install the SSH keys for each service before you can use those services.

You should ensure that the SSH keys stored on your host machine are adequately pro-

tected. See SSH Client Key Protection (nShield 5s HSMs).

From firmware versions 13.5 onwards the SSH keys are further protected by an internally generated certificate. This certificate binds an SSH key to the ESN of the module which generated the key. Existing warrants validate that the HSM is a genuine Entrust module with that same ESN. The combination of the certificate and warrant provides a way to validate that the SSH keys have not been tampered with after being generated.

The internal certificates are generated each time the HSM is factory-stated. If your HSM has been upgraded from a firmware version earlier than 13.5 you must factory state your HSM to generate the certificates. See factory state for more information.

From firmware versions 13.5 onwards, you will not be able to change the SSH keys while in Primary mode if the system clock is invalid. See System interaction with the system clock. In this situation you must be in Recovery mode to reset the system time.

Entrust recommends that you back up the `sshadmin` key as described in Making a backup of installed SSH keys whenever SSH keys are installed or changed, if your security policy allows.

4.7.2. Installation of SSH keys as part of software installation

The `hsmadmin enroll` command automates the installation of SSH keys.

On Linux, this command is run automatically as part of the software installation script. On Windows, this command must be run immediately after installation of the software.

See hsmadmin enroll for further details.

Also see SSH Client Key Protection (nShield 5s HSMs) for information on protecting your SSH keys.

After a successful installation you should backup your SSH keys. See Making a backup of installed SSH keys.

Linux-only

If the installation successfully enrolls the HSM and the `/root/.ssh` directory exists, the install script will automatically create a local backup of the `sshadmin` key in `/root/.ssh/id_nshield5_sshadmin`.

> In subsequent installations, `/root/.ssh/id_nshield5_sshadmin` will be tried if the `sshadmin` key is not found in its usual location under `/opt/nfast/services/client`, for example, if that directory has been accidentally deleted.

4.7.3. Installation of SSH keys independently of a software installation

If the HSM has been returned to factory state, either with the `hsmadmin factorystate` command or by booting the HSM in recovery mode, as described in factory state, you must install the SSH keys with the `hsmadmin enroll` command before any other actions can be performed.

The `hsmadmin enroll` command can be run on a module in which SSH keys have already been installed. In such a system, the command detects that valid keys already exist and takes no action.

If you are installing SSH keys due to their accidental loss or erasure, and you have previously made a backup of the `sshadmin` key using `hsmadmin keys backup`, then you can install them without returning your HSM to factory state by passing the path to the backed-up `sshadmin` key to `hsmadmin keys restore`.

The `hsmadmin enroll` command automatically validates certificates as part of the enrollment process and produces a warning if it fails to find a certificate for any service. This warning is expected if the HSM:

- is in recovery mode
- is running a firmware version prior to 13.5
- has been upgraded to a firmware version of 13.5 or later but has not performed a factory state operation since the upgrade.

If you receive this warning in any other circumstance you should contact Entrust support.

4.7.4. Viewing installed SSH keys

The SSH keys installed on the host and each connected HSM can be viewed using the command `hsmadmin keys show`. All the keys shown are public keys. Private keys are not viewable with this command.

The command also shows the date and time at which the client (host) keys were installed.

4.7.5. Changing installed SSH keys

If your security policy requires you to change the client (host) SSH keys, you can achieve this with the following method.

1. Print the currently installed keys with the command `hsmadmin keys show`
2. Generate and install new client keys with the command `hsmadmin keys roll`. See SSH Client Key Protection (nShield 5s HSMs) for information about protection options that can be set on keys during generation.

Linux-only

The hardserver must be restarted in order to be able to use the new `ncoreapi` SSH client key after performing this operation, for example, with `/opt/nfast/sbin/init.d-ncipher restart`.

3. Verify that the new keys have been installed with the command `hsmadmin keys show`

It is not possible to change the server (HSM) keys with this method. Should you be required to change the server keys, this can only be achieved by returning the unit to factory state with the `hsmadmin factorystate` command or by booting the HSM in recovery mode, see nShield 5s modes of operation.

4.7.6. Making a backup of installed SSH keys

If your security policy allows it, make a backup of your private client key for the `sshadmin` service so that communication with the HSM can be re-established if the installed keys are erased or otherwise lost.

Do this with command hsmadmin keys backup for verbatim copy of the `sshadmin` key with its existing protections (by default, it is tied to the host machine). Use `hsmadmin keys backup --passphrase` to backup the `sshadmin` key with a user-supplied passphrase so that it can be restored on another machine or after a re-installlation of the OS if necessary.

The backup key should be protected from unauthorized access. Refer to your security procedures for information on how to store the backup file.

4.7.7. Restoring SSH keys from backup

If you erase or lose your SSH keys, communication with the HSM will be lost. If you have previously made a backup of those keys using the command hsmadmin keys backup you can restore that backup with the command hsmadmin keys restore. This command will restore the private client key for the `sshadmin` service and then create keys for all other ser-

vices.

4.7.8. What to do if you have deleted your SSH keys and have no backup

If you erase or lose your SSH keys, communication with the HSM will be lost. If you do not have a backup copy of the keys then you will need to return the unit to factory state in order to restore communication. See Recovery from loss of SSH keys.

On Linux systems a previous successful installation will have attempted to automatically write a backup key to `/root/.ssh/id_nshield5_sshadmin`. If this key exists, then you can attempt to use it by running the install script. If the installation is successful, then you will not need to factorystate your HSM.

Without SSH keys it will not be possible to use the `hsmadmin factorystate` command and you should use the following procedure:

- Enter recovery mode by booting the unit whilst holding down the recovery mode button
- Wait for the boot sequence to complete
- Return to primary mode by booting the unit without holding down the recovery mode button
- Wait for the boot sequence to complete
- Install SSH keys as described at Installation of SSH keys independently of a software installation

Returning a unit to factory state erases all user credentials and information as described at Return to factory state

4.7.9. Preparing an HSM for use in another host

The client (host) SSH keys must be the same for every HSM connected to the same host. This will happen automatically if the HSMs are all installed together and are all in factory state. The `hsmadmin enroll` command installs the same client keys in each HSM.

Additional HSMs can be installed in a host at any time and, provided that the new modules are in factory state, the `hsmadmin enroll` command installs the same client keys in the new modules as are currently installed in any existing modules.

If it is necessary to be able to transfer a module from one host to another without returning

it to factory state this can be achieved with the following method.

In the method below, the term 'source' refers to the host from which the module will be transferred and the term 'destination' refers to the host to which the module will be transferred.

1. Backup the private `sshadmin` client key from the destination host to a location that can be accessed by the source host, such as a shared drive or a USB stick, with the following command:

   ```
   hsmadmin keys backup --passphrase <FILE>
   ```

   Where `<FILE>` specifies the location of the shared drive or USB stick. You will be prompted to enter and confirm a passphrase to use to protect the key.

2. Make sure that the HSM is in Maintenance mode, then install the destination host private `sshadmin` key on the source host with the following command:

   ```
   hsmadmin keys migrate --privkeyfile <FILE>
   ```

   Where `<FILE>` specifies the location of the file written in the previous step. You will be prompted to enter the passphrase of the key.

3. Remove the module from the source host and install in the destination host.

   If the keys are not changed on the destination host, this step may be left indefinitely or until needed. For example, the module could be kept in storage as a cold standby unit.

4. Run `hsmadmin enroll` on the source host to refresh the list it has of installed nShield HSM HSMs.
5. Run `hsmadmin enroll` on the destination host.

4.8. SSH Client Key Protection (nShield 5s HSMs)

4.8.1. SSH Services

This table explains what the different services are used for, to help inform what protection settings are appropriate for the client keys for those services in a deployment.

The `tenantreq` key is not used to access a corresponding HSM service, it is used by tenant tools to sign requests that are sent to the service

> provider. It is included here as the key should be protected in a similar way to the SSH keys.

Service	Service Description
`sshadmin`	Main authority for administration of the SSH services on the HSM. This key should have the strongest protection.
`ncoreapi`	nCore API service. Used by the hardserver for routine communication with the HSM.
`setup`	Setup service. Used for some administration options such as factory state.
`updater`	Updater service. Used for installing signed firmware upgrade packages.
`monitor`	Monitor service. Used for diagnostic operations such as retrieving logs with `hsmadmin logs`.
`launcher`	Launcher service. On versions with CodeSafe 5 support, this is used for starting CodeSafe 5 applications on the HSM.
`orchestrator`	Orchestrator service. On versions with multi-tenancy support, this is used for managing virtual crypto modules (VCM) on the HSM.
`tenantreq`	`Tenant requests`. On versions with multi-tenancy support, this is used for signing tenant request messages sent between the tenant and service provider.

4.8.2. SSH Client Key Encryption

SSH client keys are protected by a passphrase derived from one or more inputs including machine IDs and user-supplied passphrases.

These passphrases are derived automatically by applications which use the SSH keys, and with the exception of the option of user-supplied passphrases do not prompt the user.

4.8.2.1. Available SSH Key Protection Options

The following are the supported protection options for SSH keys. Multiple options can be combined in any order.

On Linux, although the hardserver runs as `nfast` user, it starts as root and then drops privileges. The hardserver derives any SSH key passphrases before dropping privileges, and so can use protections that are available only to root.

Option	Description
K	Per-key nonce. Ensures that the derived passphrase is unique to the key.
F	Fixed nonce present in the nShield client software.
S	System UUID. Ties the key to the local machine. On Linux, this is only read able by root. This is available on most systems.
B	Baseboard UUID. Ties the key to the baseboard (motherboard) of the local machine. On Linux, this is only readable by root. This may not be available on all systems.
G	Global nonce. Ties the key to the local OS install. The global nonce is read able only by root or Administrators and is generated the first time the option is used to protect a key.
P	User passphrase. The key will require a user-supplied passphrase (in addi-tion to any other protection options specified).

4.8.2.1.1. User-supplied SSH Key passphrases

If a key is generated with protection by a user-supplied SSH key passphrase, there will be an interactive prompt on the console to enter and confirm the passphrase. When a key is loaded with passphrase protection, there will be an interactive prompt on the console to enter the passphrase.

To avoid interactive prompts for automation purposes, the user passphrase can be supplied using the environment variable `NFAST_KEYPROT_PASS`. If a user passphrase is specified for the `ncoreapi` service SSH key, then the environment variable is the only way to supply the passphrase as interactive prompts are disabled for the hardserver service.

4.8.3. Setting Protections on SSH keys

Protections are set on SSH keys when they are generated, either during `hsmadmin enroll` (if the keys are not already present) or during `hsmadmin keys roll` (if switching to a freshly generated set of SSH client keys).

Protections can be overridden using environment variables that are set in the environment of the above commands when the keys are generated. The protections for all SSH service keys can be overridden using the `NFAST_KEYPROT` environment variable. Individual SSH ser-vice keys can have their protections set directly using per-service environment variables as specified in the table below. If both `NFAST_KEYPROT` and the per-service environment vari-able are set, the per-service environment variable takes precedence.

Service	Default Protection	Environment Variable
sshadmin	KFSG	NFAST_SSHADMIN_KEYPROT
ncoreapi	KFSG	NFAST_NCOREAPI_KEYPROT
setup	KFSG	NFAST_SETUP_KEYPROT
updater	KF	NFAST_UPDATER_KEYPROT
monitor	KF	NFAST_MONITOR_KEYPROT
launcher	KF	NFAST_LAUNCHER_KEYPROT
orchestrator	KF	NFAST_ORCHESTRATOR_KEYPROT
tenantreq	KFSG	NFAST_TENANTREQ_KEYPROT

4.8.4. Permissions on SSH keys

Access to SSH keys is controlled by permissions on the directories they are created in. The key directories are created with permissions during installation of the nShield software.

Linux

All keys are owned by user root, except for ncoreapi which is owned by user nfast. The table below shows what group can read each of the service keys by default. The group that can read each key can also be overridden with the environment variables listed below if set in the environment when running the install script /opt/nfast/sbin/install. The install script will always set the owner and group on the key, so if custom groups are used, they must be specified every time the install script is run. If the group specified in the environment variable does not exist, it will be created automatically by the install script.

Service	Default Group	Environment Variable
sshadmin	root	NFAST_SSHADMIN_GROUP
ncoreapi	root	NFAST_NCOREAPI_GROUP
setup	root	NFAST_SETUP_GROUP
updater	nfastadmin	NFAST_UPDATER_GROUP
monitor	nfastadmin	NFAST_MONITOR_GROUP
launcher	nfastadmin	NFAST_LAUNCHER_GROUP
orchestrator	nfastadmin	NFAST_ORCHESTRATOR_GROUP
tenantreq	root	NFAST_TENANTREQ_GROUP

Windows

SSH keys are created under the directory `%NFAST_SERVICES_HOME%\client` (`C:\ProgramData\nCipher\services\client` by default). The installer sets permissions on this directory to be read/write by the built-in local Administrators group only. If different permissions are wanted on particular keys to enable particular users or groups to perform certain operations, then these must be set manually on the keys and parent directories to permit the required access.

4.9. Optional features

nShield HSMs support a range of optional features that provide additional functionality which must be enabled before the HSM can perform certain actions and use particular mechanisms.

Some features, such as speed ratings, see Speed ratings, can be ordered when you purchase a unit and will have been enabled in the factory.

All features can be enabled after purchase by means of a feature certificate that is supplied by Entrust, obtainable from your Entrust account manager. Feature certificates are supplied as a file made available for download or requested as a smart (Activator) card, to be delivered by post.

See Ordering additional features

4.9.1. Built-in features

A number of features that were optional on earlier versions of firmware are now built-in to the firmware and are always available for use regardless of whether a license has been purchased or not. These are:

- StandardKM (all firmware)
- EllipticCurve (from v13.5 onwards)
- ECCMQV (from v13.7 onwards)
- AcceleratedECC (from v13.5 onwards)
- PostQuantum (from v13.7 onwards, nShield 5 only)

4.9.2. Persistence of features

Most features are static and remain enabled even if the HSM is initialized.

On a network-attached HSM, client licenses are dynamic and need to be reapplied if the HSM is initialized.

For nShield Connect and Solo HSMs the Feature SEE Activation (Restricted) is a dynamic feature and must be reapplied if the HSM is initialized.

All other features are static.

nShield 5s

Features that have been set in the factory will persist if the module is returned to factory state as described in return to factory state. Features that have been set with a feature certificate will be lost if the unit is returned to factory state and must be enabled again by re-applying the certificate when the HSM is returned to service.

4.9.3. Enabling features

See Enable features on a network-attached HSM and Enable features on PCIe and USB HSMs for help enabling features.

After you have enabled features on a PCIe or USB HSM, you must clear the module to make them available. Clear the module by running the command `nopclearfail --clear --all`, or by pressing the module's **Clear** button on pre-nShield 5s models.

If you are enabling the Remote Operator feature, you must enable it on the HSM that is to be used as the unattended HSM.

For information about Remote Operator, see Remote Operator.

4.9.4. Available optional features

This section lists the features that can be added to the HSM. For details of all available features, contact Sales.

4.9.4.1. Elliptic Curve Cryptography

Cryptography based on elliptic curves relies on the mathematics of random elliptic curve elements.

It offers better performance for an equivalent key length than either RSA or Diffie-Hellman public key systems. For example, using RSA or Diffie-Hellman to protect 128-bit AES keys

requires a key of at least 3072 bits. The equivalent key size for elliptic curves is only 256 bits. Using a smaller key reduces storage and transmission requirements.

Elliptic curve cryptography is endorsed by the US National Security Agency and NIST (the National Institute of Standards and Technology), and by standardization bodies including ANSI, IEEE and ISO.

nShield modules incorporate hardware that supports elliptic curve operations for ECDH (Elliptic curve Diffie-Hellman) and ECDSA (Elliptic Curve Digital Signature Algorithm) keys.

4.9.4.1.1. Elliptic Curve activation

Prior to v13.5 firmware, all nShield HSMs require specific activation to utilize the elliptic curve features, see Ordering additional features

From firmware v13.5, elliptic curve support is always enabled.

4.9.4.1.2. Elliptic Curve MQV

Prior to v13.7 firmware, elliptic curve support for *MQV* (*Menezes-Qu-Vanstone*) modes requires specific activation, see Ordering additional features

From firmware v13.7, MQV support is always enabled.

4.9.4.1.3. Elliptic Curve support on the nShield product line

The following table details the range of nShield HSMs and the level of elliptic curve support that they offer.

HSM module type	**Elliptic Curve support**		**Elliptic Curve offload acceleration**[3]	
	Named curves[2]	Custom curves[1], [5]	Named curves[2]	Custom curves[1], [5]
nShield Edge	Yes	Yes	No	No
nShield Solo 500 and 6000 nShield 500, 1500, and 6000	Yes	Yes	No	No
nShield Solo 500+, 6000+ nShield 6000+	Yes	Yes	Yes, Prime curves and twisted Brain pool curves are accelerated[4].	Yes
nShield Solo XC	Yes	Yes	Yes, Prime curves and both twisted and non-twisted Brainpool curves are accelerated[4].	Yes

HSM module type	Elliptic Curve support		Elliptic Curve offload acceleration[3]	
nShield 5s	Yes	Yes	Yes, Prime curves and both twisted and non-twisted Brainpool curves are accelerated.	Yes

[1]Accessed via nCore, PKCS#11 and JCE APIs.

[2]Both Prime and Binary named curves are supported. Refer to Named Curves, below, which lists the most commonly supported elliptic curves.

[3]Offload acceleration refers to offloading the elliptic curve operation from the main CPU for dedicated EC hardware acceleration.

[4]Binary curves are supported, but are not hardware offload accelerated.

[5]Brainpool curves are supported as named curves via nCore, PKCS#11 and JCE only.

4.9.4.1.4. nShield software / API support required to use elliptic curve functions

	Security World Software for nShield	CodeSafe
Elliptic curve supported / API	Microsoft CNG, PKCS#11, Java Cryptographic Engine (JCE)[1].	Microsoft CNG, PKCS#11, Java Cryptographic Engine (JCE)[1].

[1]Java elliptic curve functionality is fully supported by the nShield security provider, nCipherKM. There is also the option to use the Sun/IBM PKCS #11 Provider with nCipherKM configured to use the nShield PKCS#11 library.

PCIe and USB HSMs: To demonstrate the accelerated performance of elliptic signing and verify operations, run the perfcheck utility.

4.9.4.1.5. Named Curves

This table lists the supported named curves that are pre-coded in nShield module firmware.

Supported named curves			
ANSIB163v1	BrainpoolP160r1	NISTP192	SECP160r1
ANSIB191v1	BrainpoolP160t1	NISTP224	SECP256k1
	BrainpoolP192r1	NISTP256	
	BrainpoolP192t1	NISTP384	

Supported named curves			
	BrainpoolP224r1	NISTP521	
	BrainpoolP224t1	NISTB163	
	BrainpoolP256r1	NISTB233	
	BrainpoolP256t1	NISTB283	
	BrainpoolP320r1	NISTB409	
	BrainpoolP320t1	NISTB571	
	BrainpoolP384r1	NISTK163	
	BrainpoolP384t1	NISTK233	
	BrainpoolP512r1	NISTK283	
	BrainpoolP512t1	NISTK409	
		NISTK571	

4.9.4.1.6. Custom curves

nShield modules also allow the entry of custom elliptic curves which are not pre-coded in firmware. If the curve is Prime, it may benefit from hardware acceleration if supported by the nShield HSM (see nShield software / API support required to use elliptic curve functions, above).

Custom curves are supported by nCore and PKCS #11 APIs.

4.9.4.1.7. Further information on using elliptic curves

For more information on how to use elliptic curves, see the following sections:

- PKCS #11:
  - Mechanisms supported by PKCS #11: Mechanisms.
- CNG (**Windows**):
  - Supported algorithms, including key exchange, for CNG: Supported Algorithms.
- Symmetric and asymmetric algorithms: Cryptographic algorithms
- Using `generatekey` options and parameters to generate ECDH and ECDSA keys: Key generation options and parameters

Java elliptic curve functionality is fully supported by the nShield security provider, nCipherKM. There is also the option to use the Sun/IBM

> PKCS #11 Provider with nCipherKM configured to use the PKCS #11 library.

4.9.4.2. Secure Execution Engine (SEE)

> This feature is not supported in multi-tenant systems (v14.1 firmware).

The SEE is a unique secure execution environment. The SEE features available to you are:

nShield 5 HSMs: SEE Activation (CodeSafe 5)	This feature enables the ability to run signed SEE applications within your HSM. To develop your own SEE applications, you must also purchase the CodeSafe SDK and obtain a CodeSafe developer id certificate from Entrust. For more information about how to develop SEE applications, see the *CodeSafe 5 Developer Guide*.
nShield Connect and Solo HSMs: SEE Activation (EU+10)	This SEE feature is provided with the CodeSafe developer product to enable you to develop and run SEE applications. The CodeSafe developer product is only available to customers in the Community General Export Area (CGEA, also known as EU+10). Contact Entrust to find out whether your country is currently within the CGEA. For more information about the SEE, see the *CodeSafe Developer Guide*.
nShield Connect and Solo HSMs: SEE Activation (Restricted)	This SEE feature is provided with specific products that include an SEE application. This feature enables you to run your specific SEE application and is available to customers in any part of the world.

4.9.4.3. Remote Operator support

Many Entrust customers keep critical servers in a physically secure and remote location. The Security World infrastructure, however, often requires the physical presence of an operator to perform tasks such as inserting cards. Remote Operator enables these customers to remotely manage servers running Security World Software using a secure nShield communications protocol over IP networks.

The Remote Operator feature must be enabled on the module installed in the remote server. Remote Operator cannot be enabled remotely on an unattended module.

For more information about using Remote Operator, see Remote Operator.

For v12 and later, Entrust recommends that you use Remote Administration, which is more flexible than the Remote Operator functionality.

4.9.4.4. ISO smart card Support (ISS)

ISS, also called Foreign Token Open (FTO) allows data to be read to and written from ISO 7816 compliant smart cards in a manner prescribed by ISO7816-4. ISS allows you to develop and deploy a security system that can make full use of ISO 7816 compliant smart cards from any manufacturer.

4.9.4.5. Korean algorithms

This feature enables the following mechanisms:

- Korean Certificate-based Digital Signature Algorithm (KCDSA), which is a signature mechanism.

  KCDSA is used extensively in Korea as part of compliance with local regulations specified by the Korean government. For more information about the KCDSA, see the *nCore API Documentation*.

- SEED, which is a block cipher.
- ARIA, which is a block cipher.
- HAS160, which is a hash function.

4.9.4.6. Fast RNG for ECDSA

Utilise a faster alternative for Random Number Generation (RNG) for Elliptic Curve Digital Signature Algorithm (ECDSA). This feature is applicable only for nShield Solo XC, nShield Connect XC, nShield 5s, and nShield 5c.

The faster performance, comparable with v12.40 performance, is achieved by the RNG part of ECDSA being done on the NXP C291 Crypto Coprocessor.

This implementation of ECDSA uses an RNG that is not within scope for the nShield HSM certifications and for this reason it will not be used when the HSM is in a fips-140-level-3 or common-criteria-cmts Security World (regardless of the feature bit setting).

4.9.4.7. Post-Quantum algorithms

Currently the Post-Quantum algorithms supported are based upon the Module Lattice Digi-

tal Signature Algorithm (MLDSA) FIPS-204 post-quantum algorithm.

This feature enables the following mechanisms:

- MLDSA for randomized signatures
- MLDSAdet for deterministic signatures

The following MLDSA variants are supported:

- MLDSA-44
- MLDSA-65
- MLDSA-87

Post-quantum algorithms are always enabled on nShield 5 HSMs. The feature licence is only required for nShield XC.

4.9.4.8. Client licenses (network-attached HSMs)

You can purchase additional client licenses that allow you to run multiple clients for the unit. Three clients are always enabled on each unit.

4.9.5. Speed ratings

Entrust HSMs are available in a number of different speed ratings which determine the max imum number of cryptographic transactions that can be processed per second.

The speed rating of an HSM is set during manufacturing and Entrust will deliver an HSM with the speed rating configured to match the model number that was ordered.

If you have purchased a lower speed model it is possible to upgrade the speed of the unit by purchasing an upgrade licence. This license is applied in the same way as other optional features.

4.9.6. Ordering additional features

When you have decided that you require a new feature or a speed upgrade you can order it from Entrust. Before you call Entrust, collect information about your HSM as follows:

- Make a note of the Electronic Serial Number:
  - **Network-attached HSMs:** Go to **HSM > HSM information > Display details** on the front panel.
  - **PCIe and USB HSMs:** Run the `enquiry` command.

You must provide the ESN number to order a new feature.

- If possible, make a note of the serial number.
  - **Network-attached HSMs:** This is the unit serial number and is on the base of the unit.
  - **PCIe HSMs:** This is on the circuit board of the nShield module.

    nShield 5s: You can also get the serial number of the nShield HSM with `hsmadmin info`:

    ```
    dbid                : PLEXUS-01
    psn                 : 48-U50071
    mfgtime             : 2022-02-17 12:14:26 GMT Standard Time
    ```

The serial number is the `psn` in the extract of the printout above.

When your order has been processed, you will receive a Feature Enabling Certificate in one of the following ways:

- Entrust e-mails you the Feature Enabling Certificate.
- Entrust sends you a smart card that contains the Feature Enabling Certificate.

The Feature Enabling Certificate contains the information that you need to enable the features you have ordered.

For more information, including pricing of features, telephone or email your nearest Sales representative using the contact details from this guide, or contact Entrust nShield Support, https://nshieldsupport.entrust.com.

4.10. Administration of platform services (nShield 5 HSMs)

nShield 5s platform services are administered through the unified utility `hsmadmin`, which directs the command to the service that implements the command.

Some commands require elevated privileges by default because both the permissions and the protection settings have an impact on the usability of the keys by non-administrative users. Commands that create keys or modify configuration always require elevated privileges. Elevated privileges mean `root` on Linux, and the built-in local Administrators group (running in an elevated shell) on Windows. If a command requires elevated privileges, this is indicated in the command description.

You can modify the permissions and protection options on service keys to allow particular groups of users to execute commands that require the private key for a given service. See Permissions on SSH keys and Setting protection on SSH keys.

All platform services are administered through a unified utility called `hsmadmin`.

4.10.1. hsmadmin

The `hsmadmin` utility manages the administration of nShield HSMs using different subcommands.

```
hsmadmin <subcommand> [-h] [-v] [--no-reset]
```

You can use one of the following subcommands each time you run `hsmadmin`:

- factorystate
- status
- npkginfo
- upgrade
- reset
- enroll
- keys
- logs
- info
- settime
- gettime
- setminvsn
- getenvstats
- cs5
- select
- vcm
- fet
- setnetwork

You can use the following options with `hsmadmin`:

- Reset options:
  - `--no-reset`: Does not trigger an immediate reset of any HSM. If a reset is required you must reboot the host system.
- Help options:
  - `-h`, `--help`: Displays help for `hsmadmin`.

- `-v`, `--version`: Displays the version number of the Security World Software

4.10.1.1. hsmadmin factorystate

This command requires `root` privileges on Linux and the privileges of the built-in local Administrators group on Windows.

Before running this command, place the unit in maintenance mode using `nopclearfail -M -m <MODULEID> -w`.

This command returns an HSM to the state it was in when it left the factory. This securely erases all user credentials and information. It resets the `sshadmin` SSH credential to the default.

```
hsmadmin factorystate [-h] [--timeout <TIMEOUT>] [--esn <ESN>] [--verbose]
```

This command takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--all`	Resets all modules without prompting for confirmation.
`--esn`	Resets specific modules to factory state. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin factorystate --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321` If no ESNs are specified, the command resets all connected modules. The command will prompt you to confirm this action unless the --all parameter is specified.
`--verbose`	Prints verbose logs.

4.10.1.2. hsmadmin status

This command displays the ESN and currently loaded firmware version for discovered HSMs. It also displays whether the current image is a primary or a recovery image. When used with the `--json` option it displays primary firmware version, recovery firmware version, and uboot version.

```
hsmadmin status [-h] [--esn <ESN>] [--timeout <TIMEOUT>] [--verbose] [--json]
```

This command takes the following parameters:

Parameter	Description
`--verbose`	Prints verbose logs.
`--json`	Prints the HSM firmware version and image version in JSON format.
`--esn`	Displays information for specified HSMs. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin status --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321` If you do not specify any ESNs, the command displays information for all connected HSMs.
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.

4.10.1.3. hsmadmin npkginfo

This command inspects the `npkg` file and displays the metadata.

```
hsmadmin npkginfo [--json] <NPKGFILE>
```

This command takes the following parameters:

Parameter	Description
`--json`	Prints metadata in JSON format.
`<NPKGFILE>`	Specifies the NPKG-format file to inspect.

4.10.1.4. hsmadmin upgrade

This command installs firmware packages in npkg format. The command can install both primary and recovery firmware.

The command requires `root` privileges on Linux and the privileges of the built-in local Administrators group on Windows.

Before running this command, place the module to be upgraded in maintenance mode

using `nopclearfail -M -m <MODULEID> -w`. This requirement can be overridden by using the `--force` option.

In a multi-tenant system all VCMs running on the HSM to be upgraded must be stopped for this command to succeed. This cannot be overridden using the `--force` option.

```
hsmadmin upgrade [-h] --esn <ESN> [--timeout <TIMEOUT>] [--dry-run] [--force] [--verbose] [--json] <NPKGFILE>
```

This command takes the following parameters:

Parameter	Description
`--verbose`	Prints verbose logs.
`--json`	Prints metadata in JSON format.
`--dry-run`	Don't load the package, just validate it.
`--force`	Ignore warnings and force upgrade to proceed.
`--esn`	Specifies the HSMs in which to load the NPKG file. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin upgrade --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321 <NPKGFILE>`
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`<NPKGFILE>`	Specifies the npkg file to load in to the HSMs.

4.10.1.5. hsmadmin reset

Before running this command, place the unit in maintenance mode using `nopclearfail -M -m <MODULEID> -w`. If you run the command while in operational mode, it creates a failed state and you will need to run `nopclearfail -r -m <MODULEID>` to correct it.

This command resets the nShield HSM.

```
hsmadmin reset [-h] [--esn <ESN>]
```

This command takes the following parameters:

Parameter	Description
`--esn`	Specifies the HSMs to reset. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin reset --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321` If you do not specify any ESNs, all connected HSMs will be reset.

4.10.1.6. hsmadmin enroll

This command requires `root` privileges on Linux and the privileges of the built-in local Administrators group on Windows.

Before running this command, place the unit in maintenance mode using `nopclearfail -M -m <MODULEID> -w`.

This command configures the SSH keys for the nShield HSM.

```
hsmadmin enroll [--timeout <TIMEOUT>] [--verbose] [--sshadmin-key <SSHADMIN_KEY>]
```

This command takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--verbose`	Prints verbose logs.
`--sshadmin-key`	Path to backup of `sshadmin` key to use if not present in the standard location.

4.10.1.7. hsmadmin keys

This command requires `root` privileges on Linux and the privileges of the built-in local Administrators group on Windows.

Before running this command, place the unit in maintenance mode using `nopclearfail -M -m <MODULEID> -w`.

This command is used to manage the SSH keys currently loaded on a module.

```
hsmadmin keys [--timeout <TIMEOUT>] <subcommand>
```

This command takes the following parameter:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.

You can use one of the following subcommands with this command:

- show
- migrate
- roll
- backup
- restore
- remote-set
- remote-remove

4.10.1.7.1. hsmadmin keys show

This subcommand displays the public client and server keys used to communicate with the HSMs. For client keys, it also displays the time stamp held on the associated key file in the host file system.

```
hsmadmin keys show [--json] [--verbose]
```

This subcommand takes the following parameters:

Parameter	Description
`--json`	Prints output in JSON format.
`--verbose`	Prints verbose logs.

4.10.1.7.2. hsmadmin keys migrate

This subcommand changes the SSHAdmin client key on all connected modules to match a public key. The public key is derived from the private key specified in the subcommand.

```
hsmadmin keys migrate --privkeyfile <PRIVKEYFILE> [--json] [--verbose]
```

This subcommand takes the following parameters:

Parameter	Description
`--json`	Prints output in JSON format.
`--verbose`	Prints verbose logs.
`--privkeyfile`	Specifies the file containing the private key to be migrated to.

4.10.1.7.3. hsmadmin keys roll

This subcommand changes the client keys for all services.

See SSH Client Key Protection (nShield 5s HSMs) for information about protection options that can be set on keys during generation.

```
hsmadmin keys roll [--json] [--verbose]
```

This subcommand takes the following parameters:

Parameter	Description
`--json`	Prints output in JSON format.
`--verbose`	Prints verbose logs.

On Linux, the hardserver must be restarted in order to be able to use the new `ncoreapi` SSH client key after performing this operation, for example, with `/opt/nfast/sbin/init.d-ncipher restart`.

4.10.1.7.4. hsmadmin keys backup

This subcommand makes a backup of the private client key for the `sshadmin` service.

The backup key should be protected against unauthorized access. Refer to your security procedures for information on how to store the backup file.

```
hsmadmin keys backup [--passphrase] <FILE>
```

This subcommand takes the following parameters:

Parameter	Description
`--passphrase`, `-p`	Replace host key protection with passphrase protection.

Parameter	Description
<FILE>	Path to file in which to store backup.

If the --passphrase option is not supplied, then the existing sshadmin key file will be copied verbatim with whatever existing protections it has. By default, the sshadmin key is tied to the host machine and OS install, and will not be usable on another machine. A warning about this restriction to the local machine will be printed if the --passphrase option is omitted (add --local to indicate that this is the explicit choice in order to prevent the warning).

If the --passphrase option is used, then the sshadmin key will be loaded and re-encrypted using a user passphrase that must be supplied at the prompt. If the existing sshadmin key was also protected with a user passphrase (this is not the case by default), then there will be a prompt for that key's passphrase too. The backup key will not be tied to the host machine in this case, and can be used to re-install the HSM on another machine.

On Linux, the backup file will be generated with owner and group matching the directory in which it is created, and readable by owner only.

4.10.1.7.5. hsmadmin keys restore

This subcommand restores the private client key for the sshadmin service from a backup file that has previously been created with the hsmadmin keys backup command.

Once the private client key for the sshadmin service has been successfully restored, this command will automatically configure all other SSH keys for the HSM.

```
hsmadmin keys restore <FILE>
```

This subcommand takes the following parameter:

Parameter	Description
<FILE>	Path to file previously created by hsmadmin keys backup

4.10.1.7.6. hsmadmin keys remote-set

This subcommand installs a specific SSH public key for remote access to one HSM service.

```
hsmadmin keys remote-set <SERVICE> <KEYTYPE> <KEYDATA>
```

This subcommand takes the following parameters:

Parameter	Description
<SERVICE>	HSM service to be accessed remotely
<KEYTYPE>	SSH public key type
<KEYDATA>	SSH public key

4.10.1.7.7. hsmadmin keys remote-remove

This subcommand removes a specific SSH public key that had previously been set for remote access and restores the local client key.

```
hsmadmin keys remote-remove <SERVICE>
```

This subcommand takes the following parameter:

Parameter	Description
<SERVICE>	HSM service from which to remove remote access

4.10.1.8. hsmadmin logs

This command manages the system logs of connected HSMs. These logs are separate from the `ncoreapi` logs. See Platform services (nShield 5 HSMs) for more information about platform services and `ncoreapi`.

For more information about system logs, see System logging (nShield 5 HSMs).

For more information about managing `ncoreapi` logs, see Audit Logging.

```
hsmadmin logs <subcommand>
```

You can use one of the following subcommands with this command:

- get
- clear
- export
- expire
- getkey

4.10.1.8.1. hsmadmin logs get

This subcommand retrieves logs from a connected HSM.

HSMs running firmware version 13.5 or later can produce logs in either a signed or unsigned format. This subcommand will retrieve unsigned logs. To retrieve logs in a signed format, use the `export` subcommand.

```
hsmadmin logs get [-h] [--verbose] [--timeout <TIMEOUT>] --esn <ESN> --log <LOG> [--json | --out <OUTFILE>]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Specifies the HSM from which to retrieve logs. Only one ESN can be used in the command to retrieve the logs of one specific HSM.
`--json`	Prints output in JSON format.
`--out`	Write logs to file specified by OUTFILE
`--verbose`	Prints verbose logs.
`--log`	Selects log to be retrieved. Options are `system`, `init`

4.10.1.8.2. hsmadmin logs clear

This subcommand clears logs from connected HSMs.

The `system` log can only be cleared using this command on firmware versions earlier than 13.5. The `system` log on HSMs running firmware version 13.5 or later is cleared using the `expire` command. See Logging, debugging, and diagnostics for more information. The `init` log can be cleared on all firmware versions using this command.

Before running this command, place the unit in maintenance mode using `nopclearfail -M -m <MODULEID> -w`.

```
hsmadmin logs clear [-h] [--verbose] [--timeout <TIMEOUT>] [--esn <ESN] --log <LOG> [--json]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Specifies the HSMs from which to clear logs. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin logs clear --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321 --log <LOG>` If you do not specify any ESNs, logs will be cleared from all connected HSMs.
`--json`	Prints output in JSON format.
`--verbose`	Prints verbose logs.
`--log`	Selects log to be cleared. Options are `system`, `init`

4.10.1.8.3. hsmadmin logs export

This subcommand retrieves and validates signed logs from a connected HSM.

The directory used for storing the log files must exist before running this command.

```
hsmadmin logs export [-h] [--verbose] [--timeout <TIMEOUT>] [--esn <ESN>] [--saved] [--expire] [--json | --outdir <OUTDIR>]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Specifies the HSM from which to export logs.
`--json`	Prints metadata in JSON format.
`--outdir`	Write logs to directory specified by OUTDIR
`--verbose`	Prints verbose logs.
`--expire`	Expire the log after exporting it
`--saved`	If not expired, re-export a previously saved log

4.10.1.8.4. hsmadmin logs expire

This subcommand expires saved system logs from a connected HSM.

```
hsmadmin logs expire [-h] [--verbose] [--timeout <TIMEOUT>] [--esn <ESN>] --seq <SEQ_NO> [--json]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Specifies the HSM from which to expire logs.
`--json`	Prints output in JSON format.
`--seq`	expire the log identified by `<SEQ_NO>`
`--verbose`	Prints verbose logs.

4.10.1.8.5. hsmadmin logs getkey

This subcommand retrieves the system log signing key from a connected HSM.

```
hsmadmin logs getkey [-h] [--verbose] [--timeout <TIMEOUT>] [--esn <ESN>] [--json | --out <OUTFILE>]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Specifies the HSM from which to retrieve the log signing key
`--json`	Prints output in JSON format.
`--out`	Write key to file specified by `<OUTFILE>`.
`--verbose`	Prints verbose logs.

4.10.1.9. hsmadmin info

This command requires `root` privileges on Linux and the privileges of the built-in local Administrators group on Windows.

This command returns information that was loaded in the HSM during manufacturing. This

information is persistent even after returning the HSM to factory state.

```
hsmadmin info [-h] [--timeout <TIMEOUT>] [--esn <ESN>] [--verbose] [--json]
```

This command takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Returns information for the HSM identified by <ESN>. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin info --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321` If no ESNs are specified, the command returns information for all connected modules.
`--verbose`	Prints verbose logs.
`--json`	Prints output in JSON format.

4.10.1.10. hsmadmin settime

This command is used to synchronize the HSM system clock with the clock in the host PC.

See Setting the system clock for more information on managing the system clock.

This command requires `root` privileges on Linux and the privileges of the built-in local Administrators group on Windows.

To use this command without the `--adjust` parameter, the HSM must be in maintenance mode.

Setting the system date and time without the `--adjust` parameter automatically resets the HSM.

```
hsmadmin settime [-h] [--adjust <adjust>] [--timeout <TIMEOUT>] [--esn <ESN>] [--verbose]
```

This command takes the following parameters:

Parameter	Description
`--adjust`	Optional parameter. If specified an HSM System clock drift calibration is executed.
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Sets the system date and time of specific modules. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin settime --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321` If no ESNs are specified, the command resets all connected modules. If the adjust parameter is specified, a module reset is not required.
`--verbose`	Prints verbose logs.

4.10.1.11. hsmadmin gettime

This command requires `root` privileges on Linux and the privileges of the built-in local Administrators group on Windows.

It returns the system date and time of the HSM.

```
hsmadmin gettime [-h] [--timeout <TIMEOUT>] [--esn <ESN>] [--verbose] [--json]
```

This command takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Returns information for the HSM identified by <ESN>. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin gettime --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321` If no ESNs are specified, the command returns the HSM system date and time for all connected modules.
`--verbose`	Prints verbose logs.
`--json`	Prints output in JSON format.

4.10.1.12. hsmadmin setminvsn

This command requires `root` privileges on Linux and the privileges of the built-in local Administrators group on Windows.

Before running this command, place the unit in maintenance mode using `nopclearfail -M -m <MODULEID> -w`.

This command sets the minimum VSN number of the firmware which the HSM will in the future accept as an upgrade.

```
hsmadmin setminvsn [-h] [--timeout <TIMEOUT>] [--esn <ESN>] [--verbose] [--json] <VSN>
```

This command takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Sets the minimum VSN on the HSM identified by <ESN>. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin setminvsn --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321 2` If no ESNs are specified, the command sets the minimum VSN on all connected HSMs.
`--verbose`	Prints verbose logs.
`--json`	Prints output in JSON format.
`<VSN>`	The minimum VSN to set. Once this command is executed, the HSM will no longer accept a command to upgrade to a firmware with a VSN lower than <VSN>. The new minimum VSN cannot be lower than the HSM's current VSN, and cannot be higher than the VSN of the firmware currently installed on the HSM.

4.10.1.13. hsmadmin getenvstats

This command returns the environmental monitoring statistics of the HSM.

Environmental monitoring statistics available depend on the model of the HSM, the hard-

ware revision and the version of the firmware installed on the HSM.

For the nShield5 with firmware version 13.3 the available statistics are:

uptime	The time since the HSM was last rebooted, in seconds.
current_time	The current system time of the HSM.
mem_total	Total amount of physical RAM, in kilobytes.
msp_temp	Temperature recorded by the MSP sensor, in degrees C.
cpu_temp	Temperature recorded by the CPU sensor, in degrees C.
crypto_co_proc_temp	Temperature recorded by the cryptographic co-processor sensor, in degrees C.
voltage_t1022_core	Voltage drawn by the T1022 core chip.
voltage_t1022_ifc_io	Voltage drawn by the T1022 IFC I/O chip.
voltage_t1022_serdes	Voltage drawn by the T1022 SERDES chip.
voltage_t1022_serdes_io	Voltage drawn by the T1022 SERDES I/O chip.
voltage_c292_serdes	Voltage drawn by the C292 SERDES chip.
voltage_fpga_serdes	Voltage drawn by the FPGA SERDES chip.
voltage_c292_serdes_io	Voltage drawn by the C292 SERDES I/O chip.
voltage_fpga_serdes_io	Voltage drawn by the FPGA SERDES I/O chip.
voltage_msp_avcc	MSP Analogue Vcc.
voltage_ddr4_io_access	Voltage drawn by the DDR4 I/O access chip.
voltage_ddr4_io	Voltage drawn by the DDR4 I/O chip.
voltage_battery	Voltage supplied by the on-board battery.
voltage_pci_bus	Voltage drawn by the PCI bus.
max_temp	Highest temperature recorded by any temperature sensor since statistics were reset.
min_temp	Lowest temperature recorded by any temperature sensor since statistics were reset.
ais31_preliminary_alarm_count	AIS31 (RNG) preliminary alarm count.
spi_retries	SPI protocol failure count.
sp_i2c_total_failures	MSP430 I2C total failures.
sp_i2c_slave_failures	MSP430 I2C slave failures.
sp_temp_failures	MSP430 temperature failures.

sp_voltage_failures	MSP430 voltage failures.
host_bus_exceptions	PCI0 (Host) NPE and PE error count.
crypto_bus_exceptions	PCI1 (Crypto) NPE error count.
sp_sensor_cmd_failures	Read security processor handshake line failure count.
nvm_free_space	Free space on user NVRAM.
nvm_wear_level	Wear level on user NVRAM.
nvm_worn_blocks	Worn block count on user NVRAM.
bios_code	Not used; always reports 'None'
dfs_throttling	Whether CPU performance is currently degraded due to excessive heat.

```
hsmadmin getenvstats [-h] [--timeout <TIMEOUT>] [--esn <ESN>] [--verbose] [--json]
```

This command takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Returns information for the HSM identified by <ESN>. You need to add `--esn` before each ESN you include in the command, for example: `hsmadmin getenvstats --esn 1A23-BC45-6789 --esn 9Z87-YX65-4321` If no ESNs are specified, the command returns the environmental monitoring statistics of all connected modules.
`--verbose`	Prints verbose logs.
`--json`	Prints output in JSON format.

4.10.1.14. hsmadmin cs5

CodeSafe 5 is not supported in multi-tenant systems (v14.1 firmware).

This command is used to manage some aspects of CodeSafe SEE machines running on the HSM.

See also csadmin for additional commands related to managing CodeSafe SEE machines.

```
hsmadmin cs5 <subcommand>
```

You can use the following subcommand with this command:

- stats

The following subcommands are only relevant to the nShield 5c. See CodeSafe setup for the nShield 5c for more detail about the subcommands.

- clientinfo
- genclientinfo
- enroll
- unenroll
- list

4.10.1.14.1. hsmadmin cs5 stats

This subcommand gets statistics from active SEE machines.

```
hsmadmin cs5 stats [--timeout TIMEOUT] [-u UUID] [--esn ESN] [--json]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--u`	UUID of SEE machine from which to obtain statistics. If no UUID is specified, statistics will be retrieved for all running SEE machines.
`--esn`	Returns statistics for the HSM identified by <ESN>. If no ESNs are specified, the command returns statistics for all connected modules.
`--json`	Prints output in JSON format.

4.10.1.15. hsmadmin select

This command is used to select configuration options that are not controlled by licenses.

This command can only be used when the unit is in maintenance mode. It requires `root` priv

ileges on Linux and the privileges of the built-in local Administrators group on Windows.

```
hsmadmin select <option> [--esn ESN] [--json] [--timeout]
```

All select <option> commands support the following parameters:

Parameter	Description
`--esn`	Select this option on the HSM identified by <ESN>. If no ESNs are specified, the selection is applied to all connected modules.
`--json`	Prints output in JSON format. You can use the following options with this command:
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.

4.10.1.15.1. hsmadmin select acceleration

This option selects which algorithms will be accelerated.

```
hsmadmin select acceleration [--set ID | --show]
```

This option takes the following parameters:

Parameter	Description
`--set`	Set the accelerator to use. The module must be reset for the change to take effect.
`--show`	Show the available, and the currently selected accelerators.

4.10.1.16. hsmadmin vcm

This command allows the tenants to manage their VCMs.

```
hsmadmin vcm <subcommand>
```

You can use one of the following subcommands with this command:

- create

- start
- enroll
- unenroll
- setproperties
- inspect-request
- info
- getenvstats
- single-setup
- logs

4.10.1.16.1. hsmadmin vcm create

This subcommand generates a request to create a VCM. The request will be written to the file specified in the command. This file should be sent to your service provider.

The file is signed by a signing key unique to you. This key is automatically generated when you create your first request and will then be used for all subsequent requests.

```
hsmadmin vcm create [-h] [--verbose] --request-file <REQUEST_FILE> [--request-validity <REQUEST_VALIDITY>] [--restrict-startup | --autostart] [--serial-cardreader] [--features] [--name <NAME>]
```

This subcommand takes the following parameters:

Parameter	Description
`--request-file`	Specifies the name of the file to which the creation request will be written
`--verbose`	Prints verbose logs.
`--request-validity`	The time period in seconds for which the request is valid
`--restrict-startup`	If this option is chosen, the service provider will be unable to start the VCM without a valid start request
`--autostart`	A VCM with this option will automatically start after a reboot of the HSM on which it is hosted. The service provider may choose to ignore this request when creating the VCM if, for example, more tenants request this option than the service provider has a license for.
`--serial-cardreader`	A VCM with this option will have access to the serial card reader attached to the HSM. Only one VCM can have access to the card reader at any one time so the service provider may choose to ignore this request.

Parameter	Description
`--features`	The ncoreapi features you would like your VCM to have, expressed as a 32 bit hexadecimal word. The service provider may choose to provision a different set of features
`--name`	The name you would like the VCM to have. This name is only visible to the service provider so this option is mainly for use where the service provider is part of your own organisation

4.10.1.16.2. hsmadmin vcm start

This subcommand generates an authorization to start a VCM. This is needed if the VCM was created with the `--restrict-startup` option. The subcommand will write the authorization to a file specified in the command. This file must be sent to the service provider. There is an option to specify the validity period for the authorization and if this is selected the service provider must action the request before the end of the validity period and if not a new authorization must be generated.

```
hsmadmin vcm start [-h] --request-file <REQUEST_FILE> [--request-validity <REQUEST_VALIDITY>] [--uuid <UUID>]
```

This subcommand takes the following parameters:

Parameter	Description
`--request-file`	Specifies the name of the file to which the start authorization will be written
`--request-validity`	The time period in seconds for which the authorization is valid
`--uuid`	The UUID of the VCM for which the request is valid. A VCMs UUID is contained in the configuration file that was received from the service provider when the VCM was created and can also be seen in the `enquiry` command as part of the `device name`.

4.10.1.16.3. hsmadmin vcm enroll

This subcommand enrolls a VCM using the configuration file received from the service provider.

```
hsmadmin vcm enroll [-h] [--verbose] --config <CONFIG_FILE> [--no-service-restart]
```

This subcommand takes the following parameters:

Parameter	Description
`--config`	Specifies the configuration file that was received from the service provider
`--no-service-restart`	If this option is specified the hardserver and other services will not be restarted during enrollment
`--verbose`	Prints verbose logs.

4.10.1.16.4. hsmadmin vcm unenroll

This subcommand removes details of an enrolled VCM from the host file system.

```
hsmadmin vcm unenroll [-h] [--no-service-restart] [--verbose] uuid
```

This subcommand takes the following parameters:

Parameter	Description
`uuid`	Specifies the uuid of the VCM to unenroll
`--no-service-restart`	If this option is specified the hardserver and other services will not be restarted during enrollment
`--verbose`	Prints verbose logs.

4.10.1.16.5. hsmadmin vcm setproperties

This subcommand generates a request to change the properties of a VCM. The request is written to a file specified in the command and should be sent to the service provider. There is an option to specify the validity period for the request and if this is selected the service provider must action the request before the end of the validity period and if not a new request must be generated.

```
hsmadmin vcm setproperties [-h] --request-file <REQUEST_FILE> [--request-validity <REQUEST_VALIDITY>] [--uuid
<UUID>] [--restrict-startup | --no-restrict-startup] [--autostart | --no-autostart] [--serial-cardreader | --no
-serial-cardreader] [--features <FEATURES>] [--name <NAME>]
```

This subcommand takes the following parameters:

Parameter	Description
`--request-file`	Specifies the name of the file to which the request will be written
`--request-validity`	The time period in seconds for which the request is valid

Parameter	Description
`--uuid`	The UUID of the VCM for which the request is valid. A VCMs UUID is contained in the configuration file that was received from the service provider when the VCM was created and can also be seen in the `enquiry` command as part of the `device name`.
`--restrict-startup`	Enable the restrict-startup property
`--no-restrict-startup`	Disable the restrict-startup property
`--autostart`	Enable the autostart property
`--no-autostart`	Disable the autostart property
`--serial-cardreader`	Enable the serial-cardreader property
`--no-serial-cardreader`	Disable the serial-cardreader property
`--features`	Modify the VCM features. The new value is expressed as a 32 bit hexadecimal word.
`--name`	Modify the VCM name

4.10.1.16.6. hsmadmin vcm inspect-request

This subcommand displays the contents of a request or authorization file.

```
hsmadmin vcm inspect-request [--verify] [--key-file <KEY_FILE>] <REQUEST_FILE>
```

This subcommand takes the following parameters:

Parameter	Description
`--verify`	Verify the request signature
`--key-file`	The public key file for verifying the request signature

4.10.1.16.7. hsmadmin vcm info

This subcommand retrieves information about the HSM on which the VCM is hosted.

```
hsmadmin vcm info [-h] [--timeout <TIMEOUT>] [--uuid <UUID>] [--json]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--uuid`	Specifies the UUID of the VCM to retrieve info for. If omitted, information will be retrieved for all enrolled VCMs.
`--json`	Prints output in JSON format.

4.10.1.16.8. hsmadmin vcm getenvstats

This subcommand retrieves some statistics for a VCM, including the system time.

Times are reported in seconds using 'Unix time', also known as 'Epoch time'. There are readily available converters that can convert Unix time to other time formats.

```
hsmadmin vcm getenvstats [-h] [--verbose] [--timeout <TIMEOUT>] [--uuid <UUID>] [--json]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--uuid`	Specifies the UUID of the VCM from which to retrieve statistics.
`--json`	Prints output in JSON format.
`--verbose`	Prints verbose logs.

4.10.1.16.9. hsmadmin vcm single-setup

This subcommand will automatically create, start and enroll a single VCM hosted on the same machine as the HSM host.

```
hsmadmin vcm single-setup [-h] [--verbose] [--timeout <TIMEOUT>] [--esn <ESN>] [--json] [--name <NAME>]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	Specifies the HSM on which to create the VCM

Parameter	Description
`--json`	Prints output in JSON format.
`--name`	The name of the VCM which will be created
`--verbose`	Prints verbose logs.

4.10.1.16.10. hsmadmin vcm logs

This command manages the VCM system logs.

```
hsmadmin vcm logs <subcommand>
```

You can use one of the following subcommands with this command:

- export
- expire
- getkey

4.10.1.16.11. hsmadmin vcm logs export

This subcommand retrieves and validates signed logs from a connected VCM.

The directory used for storing the log files must exist before running this command.

```
hsmadmin vcm logs export [-h] [--verbose] [--timeout <TIMEOUT>] --uuid <UUID> [--saved] [--expire] [--json |
--outdir <OUTDIR>]
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--uuid`	Specifies the uuid of the VCM from which to export logs.
`--json`	Prints metadata in JSON format.
`--outdir`	Write logs to directory specified by OUTDIR
`--verbose`	Prints verbose logs.
`--expire`	Expire the log after exporting it

Parameter	Description
--saved	If not expired, re-export a previously saved log

4.10.1.16.12. hsmadmin vcm logs expire

This subcommand expires saved system logs from a connected VCM.

```
hsmadmin vcm logs expire [-h] [--verbose] [--timeout <TIMEOUT>] --uuid <UUID> --seq <SEQ_NO> [--json]
```

This subcommand takes the following parameters:

Parameter	Description
--timeout	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
--esn	Specifies the uuid of the VCM from which to expire logs.
--json	Prints output in JSON format.
--seq	Expire the log identified by <SEQ_NO>
--verbose	Prints verbose logs.

4.10.1.16.13. hsmadmin vcm logs getkey

This subcommand retrieves the system log signing key from a connected VCM.

```
hsmadmin vcm logs getkey [-h] [--verbose] [--timeout <TIMEOUT>] --uuid <UUID> [--json | --out <OUTFILE>]
```

This subcommand takes the following parameters:

Parameter	Description
--timeout	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
--esn	Specifies the uuid of the VCM from which to retrieve the log signing key
--json	Prints output in JSON format.
--out	Write key to file specified by <OUTFILE>.
--verbose	Prints verbose logs.

4.10.1.17. hsmadmin fet

This subcommand applies a feature licence to the HSM.

```
hsmadmin fet [--timeout TIMEOUT] [--verbose] [--esn ESN] [--json] <FILENAME>
```

This subcommand takes the following parameters:

Parameter	Description
`--timeout`	Time to wait for service response, in seconds. Default 30 seconds, minimum 3s, maximum 120s.
`--esn`	ESN of the HSM on which to apply the license This must match the ESN contained in the license.
`--json`	Prints output in JSON format.
`FILENAME`	Certificate file containing licence to be applied

4.10.1.18. hsmadmin setnetwork

This command allows you to specify the subnetwork for this HSM on which VCMs will be created.

After applying this command, you must reset your HSM using hsadmin reset for the settings to take effect.

```
hsmadmin setnetwork <subcommand>
```

You can use one of the following subcommands with this command:

- default
- ipv4static
- ipv6static

4.10.1.18.1. hsmadmin setnetwork default

This subcommand sets the network to use IPv6 link local addresses, which is the default setting. The main use of this is where all tenants are on the same machine as the HSM. For example when the HSM is used for single tenant operation or if multiple tenants are hosted in containers on the same machine.

```
hsmadmin setnetwork default [-h]
```

This subcommand takes no parameters.

4.10.1.18.2. hsmadmin vcm setnetwork ipv4static

This subcommand sets the IPv4 subnetwork on which VCMs will be created.

```
hsmadmin setnetwork ipv4static --address <ADDRESS> --gateway <GATEWAY>
```

This subcommand takes the following parameters:

Parameter	Description
address	IPv4 address and network mask in CIDR format
gateway	IPv4 address of network gateway

4.10.1.18.3. hsmadmin vcm setnetwork ipv6static

This subcommand sets the IPv6 subnetwork on which VCMs will be created.

```
hsmadmin setnetwork ipv6static --address <ADDRESS> --gateway <GATEWAY>
```

This subcommand takes the following parameters:

Parameter	Description
address	IPv6 address and network mask in CIDR format
gateway	IPv6 address of network gateway

4.11. Remote Operator

This chapter explains:

- The concept of Remote Operator
- How to configure Remote Operator.

If you wish to use the Remote Operator feature, you must have enabled it as described in Optional features. The Remote Operator feature must have been ordered for, and enabled on, the nShield module that you intend to use as the remote, unattended module.

4.11.1. About Remote Operator

The Remote Operator feature enables the contents of a smart card inserted into the slot of one module (the *attended module*) to be securely transmitted and loaded onto another module (an *unattended module*). This is useful when you need to load an OCS-protected key onto a machine to which you do not have physical access (because, for example, it is in a secure area).

For Remote Operator to work, the modules must be in the same Security World. You insert the required cards from the OCS into a slot in the attended module. From this module, the contents of the OCS are transmitted over secure channels to the unattended module, which then loads them. You do not need physical access to the unattended module in order to load the OCS onto it.

The following limitations apply to Remote Operator:

- You cannot access non-persistent card sets remotely
- You cannot use createocs to write new cards or card sets remotely.

You can export a slot from an attended module and import a slot to any (unattended) module in the Security World. Before you can import a slot to one module, you must first export it from another module.

4.11.2. Configuring Remote Operator

This section explains how to configure Remote Operator.

4.11.2.1. Overview of configuring Remote Operator

Before you can use Remote Operator, you must perform the following initial configuration tasks:

1. Configure the HSMs for Remote Operator.

   The HSMs must be in the same Security World, and must have been initialized with remote card set reading enabled.

   Both the attended and the unattended HSM must be in operational mode before they can import or export slots. For more about changing the mode, see:

   - Network-attached HSMs: Checking and changing the mode on a network-attached HSM
   - nShield Solo and Solo XC: Checking and changing the mode on an nShield Solo

module

- nShield 5s: nShield 5s modes of operation
- USB HSMs: Checking and changing the mode on an nShield Edge

2. Configure the HSMs (**network-attached HSMs**) or the HSM hardservers on their respective host machines (**PCIe and USB HSMs**) for slot import and export, as appropriate.

Starting from 12.81, you can export and import dynamic slots as Remote Operator slots.

After the initial configuration is complete, to use Remote Operator you must:

1. Create a Remote OCS (that is, an OCS with the correct permissions for Remote Operator).
2. Generate keys that are protected by the Remote OCS.
3. Ensure your application is configured to use keys protected by the Remote OCS.

4.11.2.2. Configuring HSMs for Remote Operator

1. Ensure both HSMs are initialized into the same Security World; see Adding or restoring an HSM to the Security World.

By default, HSMs are initialized with remote card-set reading enabled. If you do not want an HSM to be able to read remote card sets, you can initialize it by running the `new-world` with the `-S` *MODULE* (where *MODULE* is the HSM's ID number).

2. For the unattended HSM:
   a. Check whether the Remote Operator feature is enabled by running the `enquiry` command-line utility. The output for the HSM must include `Remote Share` in its list of Features.
   b. **Network-attached HSMs:** Check whether the HSM has permission to allow loading of Remote OCSs by selecting **Security World mgmt** > **Display World info**.
   c. Check whether the correct software, with permission to receive remote shares, is present by running the `nfkminfo` command-line utility.

      The output from this selection must show that `flags` are set to include **ShareTarget**, as in the following example:

```
Module #1
generation 2
state 0x2 Usable
flags 0x10000 ShareTarget
n_slots 3
```

```
esn 8851-43DF-3795
hkml 391eb12cf98c112094c1d3ca06c54bfe3c07a103
```

4.11.2.3. Configuring slot import and export

For information about the parameters controlled by the hardserver configuration file, see nShield HSM configuration files.

Before you can configure hardservers for Remote Operator, ensure that:

- You have configured the attended and unattended HSMs for Remote Operator as described in Configuring HSMs for Remote Operator.
- Your network firewall settings are correct. See Before you install the software for more information about firewall settings.

When the HSMs have been configured, use one of the following methods to configure slot import and export:

- **Network-attached HSMs:** Use the nShield HSM front panel, see Configuring slot import and export using the nShield HSM front panel (**network-attached HSMs**).
- **PCIe and USB HSMs:** Use the cfg-remoteslots utility.
- Update the HSM configuration file, see Configuring hardservers for Remote Operator using the HSM configuration file.

4.11.2.3.1. Configuring slot import and export using the nShield HSM front panel (**network-attached HSMs**)

1. Configure the attended HSM to export a slot by following these steps:
   a. From the main menu, select **Security World mgmt** > **Set up remote slots** > **Export slot**.

      Use this option for exporting slot #0 only.

      If you need to configure the export of slots other than 0, see Configuring hardservers for Remote Operator using the HSM configuration file.

   b. Specify the HSM to which the slot is being export by supplying values for:
      - The IP address of the unattended HSM
      - The ESN of the unattended HSM.
2. Configure the unattended HSM to import the slot that you are exporting from the attended HSM by following these steps:
   a. From the main menu, select **Security World mgmt** > **Set up remote slots** > **Import slot**.

b. Specify the details of the Remote Operator slot by supplying values for:
   - The IP address of the HSM from which the slot is being exported
   - The ESN of the HSM from which the slot is being exported
   - The ID of the slot on the importing HSM
   - The port to use to connect to the hardserver hosting the attended HSM.

You can check that the slot was imported successfully by, on the unattended machine, running the command:

```
slotinfo -m 1
```

If slot importation was successful, the output from this command includes the line:

```
Slot Type           Token   IC    Flags   Details
#0   Smartcard      present 3     A
#1   Software Tkn   -       0
#2   Smartcard      -       0     AR
```

The `R` in the `Flags` column indicates that slot `2` is a Remote Operator slot.

Applications running on the unattended machine can now use slot `2` to load OCSs that are presented to slot `0` on the attended machine. If any of the cards require a passphrase, the application must pass this to the unattended HSM in the usual way.

For the application to be able to load the OCS onto the unattended HSM, it must be able to read the card set files associated with the OCS from the local Key Management Data directory. If the OCS was created on a different machine, you must copy the card set files in the Key Management Data directory onto the unattended machine (either manually or by using client cooperation; for more information, see Client cooperation).

The same applies for any keys that an application on an unattended HSM needs to load but that were not generated on that machine.

4.11.2.3.2. Configuring hardservers for Remote Operator using the HSM configuration file

1. On the attended HSM's host machine, configure the hardserver to allow slot 0 of the local HSM (with ESN AAAA-AAAA-AAAA) to be exported to a remote HSM (with ESN BBBB-BBBB-BBBB, hosted by the machine with the IP address 222.222.222.222):

   ▼ *PCIe and USB HSMs*

   a. Edit the `slot_exports` section of the hardserver configuration file by adding lines of the form:

```
local_esn=AAAA-AAAA-AAAA
local_slotid=0
remote_ip=222.222.222.222
remote_esn=BBBB-BBBB-BBBB
```

b. Run cfg-reread to prompt the hardserver to read the configuration changes.

▼ *Network-attached HSMs*

a. Create a copy of the configuration file as `config.new` in the `/opt/nfast/kmdata/hsm-ESN/config` (**Linux**) or `C:\ProgramData\nCipher\nfast\kmdata\hsm-ESN\config` (**Windows**) directory.

b. Edit the sections related to slot export in `config.new`:

```
[slot_exports]
# Start of the slot_exports section
# Local slots that the hardserver should allow remote modules to import. Note
# that if a slot which has been remapped in the slot_mapping section is to be
# exported, it must be referred to in this section by its original
# (pre-mapping) local_slotid.
# Each entry has the following fields:
#
# ESN of the local module whose slot is allowed to be exported.
#  local_esn=ESN
#
# SlotID of the slot which is allowed to be exported. (default=0)
#  local_slotid=INT
#
# IP address of the machine allowed to import the slot or empty to allow all
# machines. (which is the default)
#  remote_ip=ADDR
#
# ESN of the module allowed to import the slot or "" to allow all modules
# which are permitted in the security world. (default ="")
#  remote_esn=ESN
```

```
[slot_mapping]
# Start of the slot_mapping section
# Slot remapping configuration. Notes for Remote Operator users: If a slot
# which is remapped in this section is also exported in the slot_exports
# section, the local_slotid field in the slot_exports section must be set to
# the original (pre-mapping) local_slotid. When importing that slot in another
# module, the slot_imports section must refer instead to the new
# (post-mapping) remote_slotid.
# Each entry has the following fields:
#
# ESN of the module on which slot 0 will be remapped with another.
#  esn=ESN
#
# Slot to exchange with slot 0. Setting this value to 0 means do
# nothing.(default=0)
#  slot=INT
```

c. Run cfg-pushnethsm on the updated configuration file, specifying the updated file and the network address of the nShield HSM to load the new configuration.

```
cfg-pushnethsm --address=<module_address> <path_to_config_file>
```

d. Check that the configuration file has been updated. This can be confirmed using the timestamp on the updated config file.

e. Clear the HSM for the changes to take effect, run the `nopclearfail` command:

2. On the unattended module's host machine, configure the hardserver to import slot 0 from the remote attended module (with ESN *AAAA-AAAA-AAAA*, hosted by the machine with the IP address *111.111.111.111*) to the local module (with ESN *BBBB-BBB-BBBB*).

▼ *PCIe and USB HSMs*

a. Edit the `slot_imports` section of the hardserver configuration file by adding lines of the form:

```
local_esn=BBBB-BBBB-BBBB
local_slotid=2
remote_ip=111.111.111.111
remote_esn=AAAA-AAAA-AAAA
remote_slotid=0
```

This example assigns the imported slot to ID 2.

▼ *Network-attached HSMs*

a. Edit the sections related to slot import in `config.new`:

```
[slot_imports]
# Start of the slot_imports section
# Remote slots that the hardserver should import to modules on this machine.
# Note that if a remote slot which has been remapped in the slot_mapping
# section on the remote system is to be imported, it must be referred to in
# this section by its new (post-mapping) remote_slotid.
# Each entry has the following fields:
#
# ESN of the local module to import the slot to
#  local_esn=ESN
#
# SlotID to use to refer to the slot when it is imported on the local module.
# Setting this value to 0 means it will be automatically assigned to the
# lowest available value. (default=0)
#  local_slotid=INT
#
# IP address of the machine hosting the slot to import
#  remote_ip=ADDR
#
# Port to connect to on the remote machine
#  remote_port=PORT
#
# ESN of the remote module to import the slot from
#  remote_esn=ESN
#
# SlotID of the slot to import on the remote module (default=0)
```

```
#  remote_slotid=INT
```

b. Run cfg-pushnethsm on the updated configuration file, specifying the updated file and the network address of the nShield HSM to load the new configuration.

```
cfg-pushnethsm --address=<module_address> <path_to_config_file>
```

c. Check that the configuration file has been updated. This can be confirmed using the timestamp on the updated config file.

d. Clear the HSM for the changes to take effect, run the `nopclearfail` command:

3. Check the Remote Operator slot configuration:

```
slotinfo -m 1
```

If slot import was successful, the output from this command includes the line:

```
Slot Type           Token   IC    Flags   Details
#0   Smartcard      present 3     A
#1   Software Tkn   -       0
#2   Smartcard      -       0     AR
```

The `R` in the `Flags` column indicates that slot `2` is a Remote Operator slot.

Applications running on the unattended machine can now use slot `2` to load OCSs that are presented to slot `0` on the attended machine. If any of the cards require a passphrase, the application must pass this to the unattended HSM in the usual way.

For the application to be able to load the OCS onto the unattended HSM, it must be able to read the card set files associated with the OCS from the local Key Management Data directory. If the OCS was created on a different machine, you must copy the card set files in the Key Management Data directory onto the unattended machine (either manually or by using client cooperation; for more information, see Client cooperation).

The same applies for any keys that an application on an unattended HSM needs to load but that were not generated on that machine.

4.11.2.4. Using Remote Operator with applications requiring cards in slot 0

If you want to use Remote Operator, but have an application that expects cards to be presented in slot 0, you must configure a slot mapping for each affected HSM.

PCIe HSMs

- Use the `slot_imports` section in the hardserver configuration file to import remote slots from HSMs in the same Security World for each relevant HSM.
- Use the slot_mapping section in the hardserver configuration file to define the remote slot which is to be swapped with slot #0 for each relevant HSM.

Network-attached HSMs

Use one of the following methods:

- Use the slot_mapping section in the Connect configuration file to define a Dynamic Slot to exchange with slot 0 for an HSM and push the updated configuration file to the nShield HSM.
- Use the front panel controls to navigate to **Security World mgmt > Set up dynamic slots > Slot mapping** and follow the instructions on the screen.

You can check the mapping by:

- Running the command:

```
slotinfo -m 1
```

  For example, if remote slot #2 has been mapped to slot #0, the output from this command includes the lines:

```
Slot Type Token IC Flags Details
#0 Smartcard - 1 AR
#1 Software Tkn - 0
#2 Smartcard - 0 A
```

- The `R` in the `Flags` column indicates that slot `#0` is now a Remote Slot

Slot mapping can also be configured for a dynamic remote slot, that is, a dynamic slot in a different HSM which has been imported to the relevant HSM. The **Flags** column will contain the flags ARD.

- **Network-attached HSMs:** Using the front panel controls to navigate to **Security World mgmt > Display World**.

When dynamic slots are added to an HSM after the initial configuration was done with only remote slots, the dynamic slots will take precedence over the remote slots. The slot numbers of the remote slots will therefore change. You will have to revise the slot mapping and specify the new slot number of the remote slot.

4.11.2.5. Using Remote Operator on Remapped Slots

If a slot has been mapped to slot #0 on the attended HSM, it is still possible to export the local slot to an unattended HSM. Further, if the mapped slot is a dynamic slot, it is possible to export it as well. To do this, do the following:

1. On the attended HSM's host machine, configure the hardserver to allow the export of the relevant slot by referring to it by its original slotID.
   a. To export the local slot, local_slotid=0.
   b. To export a dynamic slot, local_slotid=2 (or higher if the HSM is configured with multiple dynamic slots).
2. On the unattended HSM's host machine, configure the hardserver to import the relevant slot by referring to it by its new slotID.
   a. To import the exported local slot, remote_slotid=2 (or higher, same as the slotID specified in the mapping section of the attended HSM's configuration file).
   b. To import the exported dynamic slot, remote_slotid=0.

4.11.2.6. Configuration Example for Using Remote Administration and Remote Operator Concurrently

Below is an example of the relevant portions of a hardserver config file to achieve concurrent usage of Remote Administration and Remote Operator. It is broken up and explained per config file section.

The `dynamic_slots` section allocates exactly 1 dynamic slot to each of modules 1 and 2.

```
[dynamic_slots]
esn=BBBB-BBBB-BBBB
slotcount=1
-----
esn=AAAA-AAAA-AAAA
slotcount=1
```

The `slot_imports` section first imports module 1 slot #0 to module 2 slot #3 and then imports module 1 slot #2 to module 2 slot #4.

```
[slot_imports]
local_esn=AAAA-AAAA-AAAA
remote_ip=127.0.0.1
remote_port=9004
remote_esn=BBBB-BBBB-BBBB
remote_slotid=0
-----
local_esn=AAAA-AAAA-AAAA
remote_ip=127.0.0.1
remote_port=9004
remote_esn=BBBB-BBBB-BBBB
```

```
remote_slotid=2
```

The `slot_exports` section allows module 1 slot #0 and module 1 slot #2 to be exported by that module.

```
[slot_exports]
local_esn=BBBB-BBBB-BBBB
local_slotid=0
-----
local_esn=BBBB-BBBB-BBBB
local_slotid=2
```

The `slot_mapping` section swaps module 2 slot #0 and module 2 slot #2.

```
[slot_mapping]
esn=AAAA-AAAA-AAAA
slot=2
```

After making the changes above to the hardserver configuration file:

1. **Network-attached HSMs:** Push the hardserver configuration file to the nShield HSM by running cfg-pushnethsm.
2. **PCIe or USB HSMs:** Re-read the hardserver configuration file by running `cfg-reread`.
3. Clear the modules by running `nopclearfail`.

This is the expected system configuration output for the relevant modules:

```
slotinfo -m1
Slot Type             Token    IC     Flags    Details
#0   Smartcard        -        0      A
#1   Software Tkn     -        0
#2   Smartcard        -        0      AD

slotinfo -m2
Slot Type             Token    IC     Flags    Details
#0   Smartcard        -        0      AD
#1   Software Tkn     -        0
#2   Smartcard        -        0      A
#3   Smartcard        -        0      AR
#4   Smartcard        -        0      ARD
```

4.11.2.7. Using Remote Operator with Remote Administration with Older Versions of the Software

Versions of Remote Operator older than 12.81 do not support its concurrent use with the Remote Administrator feature. In such a case, the following features are not supported:

- Exporting and importing dynamic slots

- Mapping remote slots to slot #0
- Automatic assignment of slotID when importing slots

It is possible to use some of the features when the attended HSM (exporting end) has the new version of the software (12.81+) and the unattended HSM (importing end) has an older version (pre-12.81).

A dynamic slot which has been exported by the attended HSM can be imported to the unat tended HSM. Its local slotID will need to be manually specified if the unattended HSM has any dynamic slots configured. This is due to the default import slot (slot #2) being occupied by the dynamic slot. The unattended HSM can remap its dynamic slots to slot #0, but can-not remap any of its imported slots.

4.11.3. Creating OCSs and keys for Remote Operator

When you have configured the HSMs and either slot import and export (**network-attached HSMs**) or hardservers for Remote Operator (**PCIe and USB HSMs**), you can create Remote OCSs and generate keys protected by them. These Remote OCSs and keys can be used by applications running on the unattended HSM.

For the most part, card sets and keys intended to be used with Remote Operator are similar to their ordinary, non-Remote counterparts.

4.11.3.1. Creating OCSs for use with Remote Operator

You can generate Remote OCSs by running the `createocs` command-line utility with the `-q|--remotely_readable` option specified. The cards in a Remote OCS must be created as persistent; see Persistent Operator Card Sets.

To check whether the card in a slot is from a Remote OCS, run the `nfkminfo` command-line utility or, on network-attached HSMs, select **Security World mgmt** > **Display World info** from the front panel main menu.

The output displays slot section information similar to the following:

```
Module #1 Slot #0 IC 1
generation         1
phystype           SmartCard
slotlistflags      0x2
state              0x5
Operator flags     0x20000 RemoteEnabled
shareno            1
shares             LTU(Remote)
error              OK
```

In this example output, the `RemoteEnabled` flag indicates the card in the slot is from a Remote OCS.

If you create a Remote OCS on the attended machine, then you must copy the Key Management Data files on the attended machine to the unattended machine.

Both the attended and unattended HSMs must be in the same Security World before you generate a Remote OCS. If you are not using client cooperation, the Key Management Data directories must be manually synchronized after you generate the Remote OCS.

If you already have recoverable keys protected by a non-Remote OCS, you can transfer them to a new Remote OCS with the `replaceocs` command-line utility.

4.11.3.2. Loading Remote Operator Card Sets

Once configured, the Remote Operator slots can be used by all the standard nShield libraries. A Remote Operator slot can be used to load any OCSs that have been created to allow remote loading. For more information about the applications to use with remote cards, see Application interfaces. For more information about Remote Operator slots, see Remote Operator.

After an OCS has been inserted into a Remote Operator slot, for each time a given card is inserted, the module only allows each share on that card to be read one time. If there is a second attempt to read shares from that card before the card is reinserted, the operation fails with a `UseLimitsUnavailable` error.

4.11.3.3. Generating keys for use with Remote Operator

After you have created a Remote OCS, to generate keys protected by it you can run `generatekey` and `preload` command-line utilities on the unattended module, inserting cards to the slot attached to the attended module. For more information about generating and working with keys, see Working with keys.

If you generate keys protected by a Remote OCS on the attended module, then you must copy the files in the Key Management Data directory on the attended machine to the unattended module.

If the key was not generated with key recovery enabled, you cannot protect it under a differ

ent OCS. In this, you must generate a new key to be protected by a Remote OCS.

4.11.3.4. Configuring the application

After you have configured the HSMs and either slot import and export (**network-attached HSMs**) or hardservers (**PCIe and USB HSMs**) for Remote Operator, created a Remote OCS, and generated keys protected by the Remote OCS, configure the application with which you want to use these keys as appropriate for the particular application.

After you have configured the application, start it remotely from the attended machine. Insert cards from the OCS into the attended machine's exported slot as prompted.

4.11.3.5. Managing Remote Operator slots using the unit front panel (network-attached HSMs)

4.11.3.5.1. Editing Remote Operator slots

You can change the details of a Remote Operator slot. You must always update the details of both the exported slot on the local module and the imported slot on the remote module.

To update an exported a slot on the module:

1. From the main menu, select **Security World mgmt** > **Set up remote slots** > **Edit exported slot**.
2. Select the exported slot that you want to update. Slots are identified by the IP address of the remote module.
3. Update the details of the slot.

To update an imported slot on the unit:

1. From the main menu, select **Security World mgmt** > **Set up remote slots** > **Edit imported slot**.
2. Select the imported slot that you want to update. Slots are identified by the IP address of the remote module.
3. Update the details of the slot.

4.11.3.5.2. Deleting Remote Operator slots

You can delete Remote Operator slots.

To delete an exported slot, from the main menu, select **Security World mgmt** > **Set up remote slots** > **Delete exported slot** and select the slot you want to delete.

To delete an imported slot, from the main menu, select **Security World mgmt** > **Set up remote slots** > **Delete imported slot** and select the slot you want to delete.

4.12. System logging (nShield 5 HSMs)

This section describes how you can access the logging information generated by the platform. This information is separate from that produced by `ncoreapi`. See Platform services (nShield 5 HSMs) for more information about platform services and `ncoreapi`.

For information about the logging and debugging information generated by `ncoreapi`, see Logging, debugging, and diagnostics and Audit Logging.

Two system logs are automatically created by the system. These are the `init` log and the `system` log.

The `init` log records information created by the system when it boots. Its primary purpose is for use by Entrust Support personnel.

The `system` log continuously records system level information whenever the system is running.

Both logs record information automatically and there is no user configuration required. The information recorded is determined by the system and there is no user configuration of the level of information recorded.

The commands used to retrieve and clear logs are described in hsmadmin logs.

4.12.1. Maximum log size

The `init` and `system` log share a common non-volatile storage area within the HSM. This storage area has the capacity to store the logs for a long period of normal usage but, if the logs are not periodically retrieved and cleared, it could eventually become full.

The `system` log is normally much larger than the `init` log but Entrust recommend that you clear both logs at the same time. This is because there are no specific warnings produced by actions that write to the `init` log. From firmware versions 13.5 onwards the `init` log sized is fixed and it is not necessary to clear the `init` log.

For HSMs running firmware version 13.5 or later, the system will detect when the logs have exceeded a safe working capacity and will prohibit the following actions:

- Factory state, see Return to Factory State
- Firmware upgrade, see Firmware upgrade

- Setting the minimum VSN, see Version Security Number
- Setting SSH keys, see Set up communication between host and module (nShield 5s HSMs)
- Setting or adjusting system time, see Setting the system clock and Adjusting the system clock
- Adminstration of CodeSafe, see The csadmin tool

These actions will continue to result in errors until the log volume is reduced back to a safe working level.

If the logs are not cleared promptly when the safe working capacity is exceeded the log volume may reach a critical capacity and at this point the system will enter an error state and display an error code on the LED. If this happens all actions are prevented other than retrieving and clearing the logs and it will be necessary to reboot the HSM to clear the error state.

It is not normally possible to clear the `system` log from HSMs running v13.5 or later firmware without first exporting the entries. However it is possible to do this in recovery mode. See Recovery Mode.

4.12.2. Interaction with the system clock

It is important that the timestamps in the system logs are accurate so that events can be correlated across the whole network in which the HSM is operating.

For HSMs running firmware version 13.5 or later, if the system clock is lost, for instance due to the HSM running on battery power for an extended period of time, a number of administrator actions will be prohibited until the system clock is restored, including:

- Log expiry, see Expiring signed logs
- Log export, see Retrieving signed logs

See System interaction with the system clock for more information.

4.12.3. Init log

The `init` log records information that is produced each time the system boots.

The information is intended for use by Entrust Support to diagnose any issues that cause an HSM to fail to boot and thus the log is normally retrieved from recovery mode, see Recov-

ery Mode.

The amount of information recorded depends on the firmware version loaded on the HSM. From firmware versions v13.5 onwards, after a successful boot, the `init` log will contain only a message indicating that the boot was successful.

4.12.3.1. Retrieving the init log

The `init` log can be retrieved with the following command:

```
hsmadmin logs get --esn <ESN> --log init [--json | --out <OUTFILE>]
```

Entrust recommend that you always direct the output to a file using the `--out` parameter and save the file for forwarding to Entrust Support.

4.12.3.2. Clearing the init log

The `init` log can be cleared with the following command:

```
hsmadmin logs clear [--esn <ESN>] --log init
```

Before running this command, place the unit in maintenance mode using `nopclearfail -M -m <MODULEID> -w`.

From firmware versions v13.5 onwards it is not necessary to clear the `init` log because it is cleared automatically each time the system successfully boots.

From firmware version v13.7 onwards this command will fail unless the unit is in recovery mode, see Recovery Mode.

4.12.4. System log

The `system` log records system level events and warnings.

For HSMs running firmware version 13.5 or later these logs are produced in a signed format. HSMs running firmware earlier than 13.5 produce logs in an unsigned format.

The procedures for managing logs differ depending on whether the logs are signed or unsigned.

4.12.4.1. Signed system logs

Signed system logs are produced by firmware version 13.5 or later and have a number of benefits.

Signed logs can be verified to ensure that they have not been tampered with. See Verifying Signed Logs.

Signed logs contain a sequence number that prevents unintentional deletion of logs and can be used to help order stored logs and identify any gaps. See Log Sequence Number.

4.12.4.1.1. Log sequence number

Signed system logs use a sequence number to prevent the unintentional loss of logs and to aid in sorting logs.

The HSM holds an internal sequence number for the system log currently being written. This sequence number is persistent over both reboots and factory state operations.

When the system log is exported see Retrieving Signed Logs the sequence number is incremented and a new log is started with the new sequence number. The previous log is not deleted by the `export` command but remains stored internally within the HSM.

To prevent unintentional duplication of logs, signed logs can normally only be exported once. If it is required to export the same log for a second time the `--saved` option must be used with the `export` command.

Once a log has been successfully exported it can be cleared from the HSM as described in Expiring Signed Logs. The use of the sequence number in this command ensures that no logs are deleted that have not been exported.

It is possible to export and expire logs in a single command but Entrust do not recommend doing this because if the command fails for any reason there is a risk that logs may be lost. The recommended procedure is to export the logs first and then expire the logs as a separate procedure only once the export has completed successfully.

Each exported log begins with one entry showing its own sequence number and another entry showing the sequence number of the preceeding log. This allows logs to be chained together to identify any missing gaps.

4.12.4.2. Retrieving the system log

4.12.4.2.1. Retrieving unsigned logs

The `system` log can be retrieved with the following command:

```
hsmadmin logs get --esn <ESN> --log system [--json | --out <OUTFILE>]
```

This command will retrieve the logs in unsigned format and will work for HSMs running any firmware.

4.12.4.2.2. Retrieving signed logs

For HSMs running firmware version 13.5 or later Entrust recommend that system logs are automatically retrieved by configuring the nShield Audit Log service as described in nShield Audit Log Service. If you wish to retrieve the logs manually you can do so with the following command:

```
hsmadmin logs export --esn <ESN> [--json | --outdir <OUTDIR>] [--saved] [--expire]
```

This command automatically verifies the logs as part of the export process

Logs should normally be exported only once. If an attempt is made to export a log that has already been exported the command will fail with a warning. If you wish to export a log that has previously been exported you should use the `--saved` option with the above command.

When upgrading to firmware version 13.5 or later from a firmware version lower than 13.5, there may initially be a saved log in the system created by the previous firmware. You should export this log using the `--saved` option and then expire it.

It is possible to automatically expire logs whilst exporting them by use of the `--expire` option but this is not recommended as it may result in loss of logs should the command fail for any reason.

4.12.4.3. Clearing the system log

The procedures for clearing the system log differ depending on whether the logs are produced in signed or unsigned format. If your HSM is running firmware version 13.5 or later your logs are produced in signed format. If your HSM is running a firmware version earlier than 13.5 your logs are produced in unsigned format.

The process of clearing a signed log is referred to as expiring.

4.12.4.3.1. Clearing unsigned logs

For HSMs running firmware versions earlier than 13.5 the log can be cleared with the following command:

```
hsmadmin logs clear [--esn <ESN>] --log system
```

For HSMs running firmware versions later than 13.5 this command will fail. You must follow the procedures described in Expiring Signed Logs.

It is possible to use this command to clear signed logs if the HSM is in recovery mode. See Recovery Mode. Entrust do not recommend this procedure unless instructed to do do by Entrust Support.

4.12.4.3.2. Expiring signed logs

If your HSM is running a firmware version of 13.5 or later the logs will be in signed format. To prevent unintentional loss of logs, signed logs must be exported before they can be cleared. You can export signed logs by following the procedures at Retrieving Signed Logs.

Logs that have previously been exported can be expired with the following command:

```
hsmadmin logs expire --esn <ESN> --seq SEQ_NO
```

This will expire the log with the sequence number `SEQ_NO`. The sequence number is included as the first line of any exported log.

It is possible to automatically expire logs whilst exporting them by use of the `--expire` option on the export command but this is not recommended as it may result in loss of logs should the command fail for any reason.

4.12.4.4. Verifying signed logs

Signed logs are automatically verified as part of the export process.

It is also possible to verify exported logs at any time in the future should you wish to do so, provided that you have access to the verification key. This verification can be conducted without access to the HSM.

The verification key is persistent over reboots but will be changed by a factory state operation. Therefore it is recommended that you record the verification key as soon as possible after any factory state operation. See Return to Factory State for information about factory

state.

The system log will contain an entry for the factory state event. This log entry will contain the value of the verification key so it will always be possible to obtain the verification key if you forget to record it.

The current log verification key can be obtained with the following command:

```
hsmadmin logs getkey --esn <ESN> [--json | --out <OUTFILE>]
```

This command automatically validates the certificate protecting the verification key and produces a warning if it fails to find a certificate for the key. This warning is expected if the HSM:

- is in recovery mode
- has been upgraded to a firmware version of 13.5 or later but has not performed a factory state operation since the upgrade.

If you receive this warning in any other circumstance, contact https://trustedcare.entrust.com/.

If you have upgraded to firmware version 13.5 or later, but have not performed a factory state operation since the upgrade, perform a factory state as soon as possible. This generates an internal certificate for the log signing key, which allows validation of the key all the way back to the root of trust.

4.13. Maintenance of nShield Hardware

This chapter describes maintenance steps for your nShield hardware installation.

This guidance is not applicable to nShield Solo+ products.

After installing your nShield HSM, Entrust recommend that you use some of the provided software utilities to monitor your installation. Specifically, the `stattree` command allows reporting of voltages and temperatures from your module.

For more information regarding `stattree`, see stattree.

4.13.1. Voltage Monitoring for Battery Replacement (nShield Solo XC and nShield 5s)

All of the voltage rails in the nShield HSM are monitored to protect against potential over- or under-voltage attacks. You can view the most recent measurement of the voltages using the `stattree` command.

These modules also contain a user-replaceable battery. The battery powers security functions on the module when the main module power is removed, for example when the host is turned off, so it is expected that the battery voltage will drop over time as the battery drains. To avoid module downtime due to battery replacement we recommend that the bat tery voltage is monitored regularly, especially if a module has had its main power removed for considerable time.

`CPUVoltage10` reported by `stattree` under the `ModuleEnvStats` node tag displays the current battery voltage:

```
+PerModule:
   +#1:
      +ModuleEnvStats:
         ...
         -CPUVoltage10         3.16
         ...
```

The battery supplied with the nShield HSM has a nominal voltage of 3.0V. In the above example the battery is fully charged and has been measured at 3.16V, which is within the acceptable range of 2.46V - 3.55V. If the battery voltage is measured to be lower than 2.46V, the module will report an `SOS-B1` error. See HSM status indicators and error codes (nShield 5s) (5s) and HSM status indicators and error codes (nShield 5s) (XC) for more infor mation regarding error reporting.

Contact Support to request information regarding a replacement bat- tery if `stattree` reports the battery voltage to be below 2.70V.

See Battery replacement for instructions on replacing the battery in your module.

4.13.2. Temperature Monitoring for Airflow Validation

Temperatures within a module are monitored to protect against potential attacks, and to prevent overheating:

- **Network-attached HSMs:** The temperature of the internal ambient air of an nShield HSM is reported under the `HostEnvStats` node tag of `stattree`as:
  - `CurrentTempC`
  - `CurrentTemp2C`
- **PCIe HSMs:** The temperatures of the processors within a PCIe HSM are reported under

the `ModuleEnvStats` node tag of `stattree` as:

- `CurrentCPUTemp1`
- `CurrentCPUTemp2`

As an nShield 5s has a passively-cooled heatsink, care must be taken to install it in an environment with forced airflow. See Prerequisites and product information for airflow guidance.

The table below documents the expected normal operating ranges for the temperatures of your module. Module temperatures would be expected to be within these values when installed with sufficient cooling in an approximately 20-30°C ambient air temperature environment. The minimum and maximum temperatures recorded by the HSM's temperature sensor are stored in non-volatile memory and are only cleared when the HSM is initialized.

The temperatures in this table do not cover operation of the product across the full temperature range specified in the *Warnings & Cautions* documentation and the nShield v14.1.1 Hardware Install and Setup Guides prerequisites pages. This is because these values are recommendations to ensure a long product lifetime, thus are specified for 20-30°C ambient air operation.

`stattree` **Statistic**	**Description**	**Minimum expected in optimum environment**	**Maximum expected in optimum environment**
`CurrentCPUTemp1` (**PCIe HSMs**)	First processor temperature	10°C	75°C
`CurrentCPUTemp2` (**PCIe HSMs**)	Second processor temperature	10°C	78°C
`MaxTempC` (**PCIe HSMs**)	Maximum temperature, measured on either processor, since the last time the HSM was initialized.	-	78°C
`MinTempC` (**PCIe HSMs**)	Minimum temperature, measured on either processor, since the last time the HSM was initialized.	10°C	-
`CurrentTempC` (**network-attached HSMs**)	Internal temperature 1	10°C	45°C
`CurrentTemp2C` (**network-attached HSMs**)	Internal temperature 2	10°C	45°C

`stattree` Statistic	Description	Minimum expected in optimum environment	Maximum expected in optimum environment
`MaxTempC` (**network-attached HSMs**)	Maximum of internal temperature 1	-	45°C
`MaxTemp2C` (**network-attached HSMs**)	Maximum of internal temperature 2	-	45°C

If any of the above temperatures are reporting higher than their specified maximum it is likely your nShield hardware does not have sufficient cooling.

4.14. Physical security of the HSM

This chapter provides a brief overview of the physical security measures that have been implemented to protect your nShield HSM. You are also shown how to check the physical security of your nShield HSM.

The tamper detection functionality on the nShield HSM provides additional physical security, over and above that provided by the holographic security seal, and alerts you to tampering in an operational environment. There is a removable lid on top of the nShield HSM, protected by the security seal and tamper switches. To prevent the insertion of objects into the nShield HSM, baffles are placed behind vents.

To optimize their effectiveness, use the physical security measures implemented on the nShield HSM in association with your security policies and procedures. For more information about creating and managing security policies, see the *Security Policy Guide* on the NIST CMVP website.

Currently, the FIPS 140 Level 3 boundary is at the internal module. Future software releases may move the FIPS boundary so that it includes the entire nShield HSM chassis.

For more information about FIPS 140, see http://csrc.nist.gov/publications/fips/fips140- 2/fips1402.pdf.

4.14.1. Tamper event

The nShield HSM offers several layers of tamper protection. The outer boundary of the box is tamper-responsive. When tampered, the unit ceases to provide cryptographic functionality, alerts the operator of the event, and ultimately forces the operator to reset the unit to

factory defaults. Movements/vibrations, or replacing the fan tray module or a PSU, does not activate the tamper detection functionality.

If a tamper event does occur, you can use the Security World data stored on the RFS and the Administrator Card Set to recover the keys and cryptographic data.

4.14.1.1. nShield HSM lid is closed

If the nShield HSM is powered, a tamper event has occurred, and the lid is closed, the unit will automatically reset to a factory state.

Should this happen, examine your unit for physical signs of tampering (see Physical security checks).

If you discover signs of tampering do *not* attempt to put the unit back into operation. The date and time of the tamper event are recorded in the log (see Logging, debugging, and diagnostics).

The tamper-responsiveness circuitry has a Real Time Clock that is synchronised to the system time of the nShield HSM, however the times associated with events in the tamper log may still have slight offsets to times recorded in other log files.

If there are signs of tampering, and the tamper event occurred:

- During transit from Entrust, contact Support.
- After installation, refer to your security policies and procedures.

For more information about creating and managing security policies, see the *Security Policy Guide*.

You require a quorum of the Administrator Card Set (ACS) to restore the key data and reconnect the nShield HSM to the network.

4.14.1.2. nShield HSM lid is open

If the nShield HSM is powered, a tamper event has occurred, and the lid is open, the following message is displayed onscreen:

```
Unit lid is open
```

An open lid indicates that the physical security of the unit is compromised. You may want to examine your unit for other physical signs of tampering (see Physical security checks).

Do *not* attempt to put the unit back into operation.

The date and time of the tamper event are recorded in the log files (see Logging, debugging, and diagnostics). If the tamper event occurred:

- During transit from Entrust, contact Support.
- After installation, refer to your security policies and procedures. For more information about creating and managing security policies, see the *Security Policy Guide* on the NIST CMVP website.

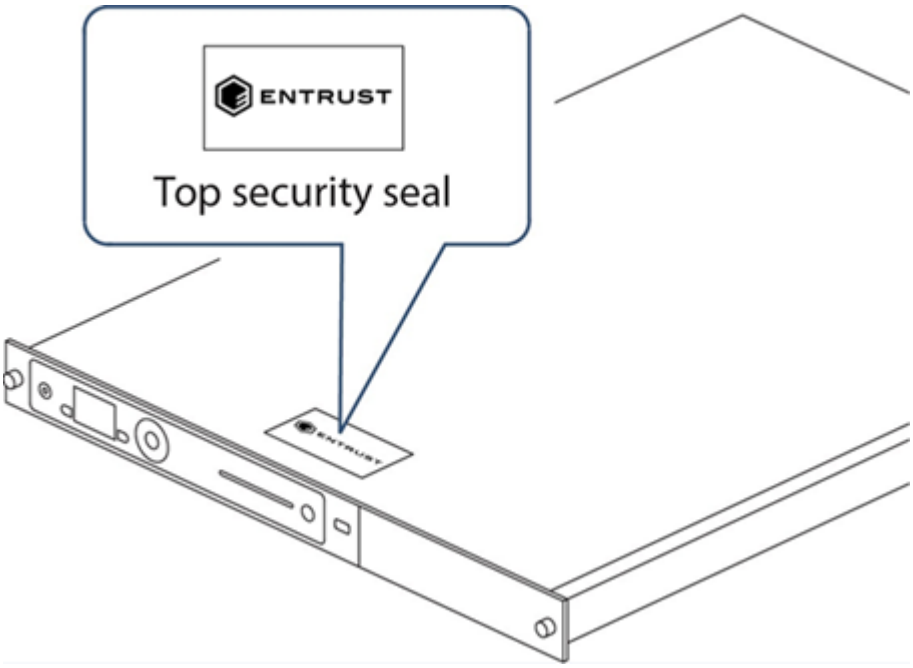
After closing the lid you must reboot the nShield HSM. The unit will then automatically reset to a factory state. If the lid remains open, the above message will remain on the screen and all button presses are ignored.

4.14.2. Physical security checks

Check the physical security of your nShield HSM before installation and at regular intervals afterwards. For an alternative presentation of the physical security checks described here, see the *Physical Security Checklist*. For more information about tamper events, and what actions to take if you discover signs of tampering, see Tamper event.

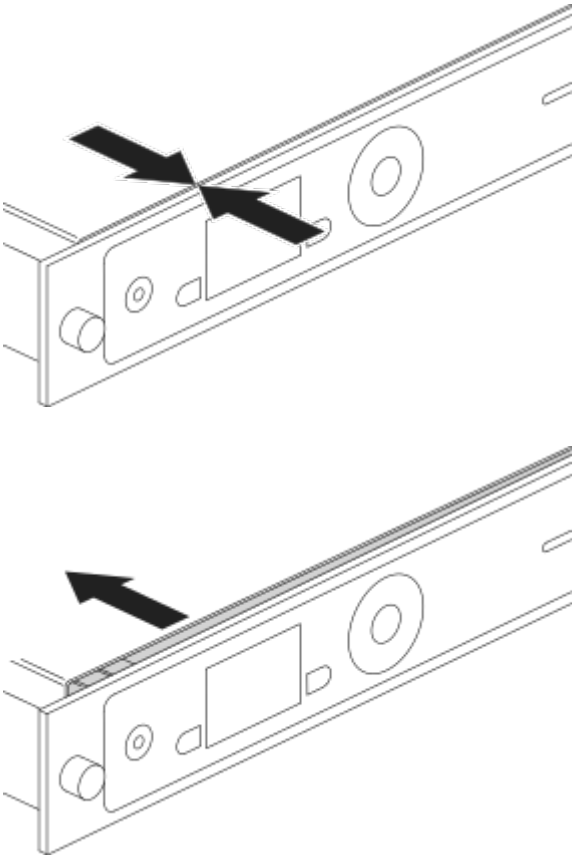
To determine if the security of the nShield HSM is compromised:

1. Check that the physical security seal is authentic and intact. Look for the holographic foil bearing the nCipher logo. Look for cuts, tears and voiding of the seal. The seal is located on the top of the nShield HSM chassis.

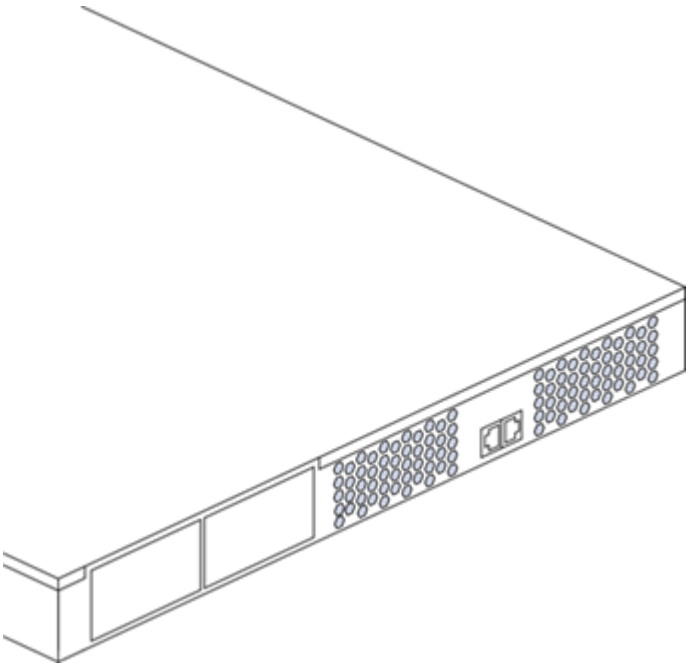


For information about the appearance of intact and damaged security seals, see the *Physical Security Checklist*.

2. Check that the metal lid remains flush with the nShield HSM chassis.



3. Check all surfaces — the top, bottom and sides of the nShield HSM — for signs of physical damage.
4. Check that there are no signs of physical damage to the vents, including attempts to insert objects into the vents.



4.14.3. Replacing the fan tray module and PSU

You can replace the fan tray module or a power supply unit (PSU) **without** activating a tamper event as both are outside the security boundary. You can access:

- The PSU(s) from the rear of the nShield HSM.
- The fan tray module through the removable front vent.

Should a problem occur with the fan tray module or a PSU, contact Support **before** taking further action. For more information about replacing the fan tray module or a PSU, see the *Fan Tray Module Installation Sheet* or the *Power Supply Unit Installation Sheet*.

The fan tray module contains back-up batteries providing reserve capacity (a guaranteed minimum of 3 years) for tamper detection functionality even when the nShield HSM is in an unpowered state.

The tamper protection circuitry remains fully operational if the nShield HSM is placed on standby while a replacement operation is performed (whether you are replacing the fan tray module or one of the two PSUs, in the case of dual PSU units).

Provided that the nShield HSM is connected to the mains power supply, it displays an onscreen error message when back-up battery power is low.

4.14.3.1. Replacing the fan tray module

It is not necessary to remove mains power to replace a fan tray module (we recommend that you power down the unit into standby state using the front panel power button). However, if mains power is removed then a replacement fan tray module **must be installed within an hour** to ensure that a tamper event is not activated. If put in standby state the time required to change fan tray module is unlimited. For more information about replacing the fan tray module, see the *Fan Tray Module Installation Sheet*.

4.14.3.1.1. Fan tray module error messages

If you receive any of the following error messages on the nShield HSM display, accompanied by the orange warning LED, follow the related action in the table below:

Error message	Action
Single fan fail	Contact Support
Many fans fail	Replace fan tray
Battery power low	Consider replacing fan tray during the next scheduled service/maintenance period.
System Shutdown Both fans in a pair had failed	Replace fan tray

If the error message is **Single fan fail**, the nShield HSM can continue operating under the specified operating environment. Although you are advised to contact Support, the limited nature of such a failure means you can replace the fan tray module at your convenience.

If the error message is **Many fans fail**, you must replace the fan tray module immediately.

If the error message is **Battery Power low**, this indicates that one or both of the backup batteries located on the fan tray module (required only when the nShield HSM is removed from mains power) is running low.

The **Battery Power low** indication has no detrimental affect on the nShield HSM performance whilst the unit remains powered. Entrust recommend customers should consider replacing the fan tray module during the next service/maintenance.

If two fans fail from a redundant pair, the nShield HSM will display the error message **Many fans have failed** for a few seconds and it will then shutdown. On reboot, the nShield HSM will then display the error messages **System Shutdown** and **Both fans in a pair had failed**. In this situation the fan tray module must be replaced immediately.

4.14.3.2. Replacing the PSU

If you have a dual PSU nShield HSM, you do not have to remove power to the functioning PSU while replacing a faulty PSU. Tamper detection functionality will operate normally throughout the PSU replacement process. If you decide to remove power from both PSUs, tamper detection functionality will continue to operate normally for at least 3 years, as the fan tray module provides back-up capacity for this circuitry. For more information about replacing the PSU, see the *Power Supply Unit Installation Sheet*.

4.14.3.2.1. PSU error messages

If a PSU fails, an orange warning LED comes on and an error message is displayed on the nShield HSM display. Although you are advised to contact Support, the unit can continue to operate normally and you can replace the failed PSU at your convenience. There is no need to power down the unit when you replace the failed PSU.

In addition to the orange warning LED, an audible warning is given when a PSU fails on an nShield HSM. The audible warning is turned off when you navigate to the Critical errors screen.

4.14.3.3. Battery life when storing the nShield HSM

If a nShield HSM has been in storage for an extended period of time the fan tray module may need replacement.

Entrust guarantees a *minimum* battery life of three years, even if the nShield HSM remains in storage and is not connected to the mains power supply during this time.

4.15. System upgrade

4.15.1. Terminology

Within this section the term 'software' is used to mean Security World software running on the PC in which the HSM is installed and the term 'firmware' is used to mean the Security World firmware running on the HSM.

The software and firmware can be upgraded independently.

4.15.2. Software and firmware compatibility

In general, Entrust recommends that you use the software and firmware from the same version of Security World. The system is designed to be backwards compatible so that it will

still operate with differing versions of software and firmware but some functionality may not be available and you may receive warnings during operation.

This user guide describes the behavior of v14.1.1 software interacting with v14.1.1 firmware. Some areas where functionality differs depending on the version of firmware loaded are also described in this guide but it is not possible to describe all possible combinations of software and firmware.

Release notes and user guides for each Security World release are available from the Entrust website and these together with Entrust Support will help you should you experience any problems when operating with differing versions of software and firmware.

4.15.3. System upgrade procedure

When upgrading the whole system, Entrust recommends that you always upgrade the host software before upgrading the HSM firmware, See Upgrading the image file and associated firmware: network-attached HSMs (network-attached HSMs) or Upgrade firmware: nShield Solo, Solo XC, and Edge HSMs (PCIe and USB HSMs) for more information.

Always read the release notes accompanying the Security World release before upgrading any part of the system as these may include additional upgrade steps.

If the Version Security Number (VSN) of the firmware has been increased, it may not be possible to roll-back the firmware to the previous version after upgrade.

4.15.3.1. Software upgrade procedure

For Security World software upgrades, you do not need to delete key data or any existing Security World. If you do delete Security World data, it cannot be restored unless you have an up-to-date backup and a quorum of the Administrator Card Set (ACS) available.

4.15.3.1.1. Before upgrading software

You must perform these steps if you are planning to re-install the Security World software, for example to re-install it on the same machine after an operating system update, or to install a newer Security World software version as part of an upgrade.

Performing these steps is useful even if you are not planning a re-install because it preserves data that you would otherwise irretrievably lose when you uninstall the Security World software.

1. For Linux installations make a backup of your Security World and nShield configuration files stored in `/opt/nfast/kmdata/` and `/opt/nfast/hardserver.d` by copying them to external media or to a location not within `/opt/nfast`.

   When you are upgrading the Security World, you will also restore the backup to preserve your PKCS #11 and Soft KNETI authentication settings and any customizations. If you delete the `/opt/nfast` or `$NFAST_HOME` directory without making a copy of it, you will lose these configuration settings. When you are restoring a Security World from a backup, you will need to maintain permissions.

2. **nShield 5s:** Back up your SSH keys, see Making a backup of installed SSH keys.

   - If you are planning a clean reinstallation of the Security World software on the **same machine and same operating system**, back up your SSH keys in `/opt/nfast/services` using `hsmadmin keys backup`.
   - If you are planning to re-create the Security World on a **different machine or after re-installing the operating system**, use `hsmadmin keys backup --passphrase`. `hsmadmin keys backup` alone is only suitable for a local backup followed by a local restore on the same machine and same operating system.

   If you erase your SSH keys without making a backup you will need to use recovery mode, see Recovery mode to restore communication with the HSM. This will return the HSM to factory state, see Factory state.

4.15.3.1.2. Upgrading software

Software upgrade is performed by uninstalling the old software as described in Uninstalling Security World Software and then installing the new software as described in Install the Security World software.

4.15.3.1.3. After upgrading software

1. Copy back any data that was manually backed-up as part of the procedures in Before upgrading software to the locations from which it was copied.
2. **nShield 5s:** Restore communication with the HSM by following the procedures at restoring SSH keys from backup.

4.16. Morse code error messages

For nShield 5 errors, see HSM status indicators and error codes (nShield

> 5s)

If a Hardware Security Module (HSM) encounters an unrecoverable error, it enters the error state. In the error state, the module does not respond to commands and does not write data to the bus.

The blue Status LED flashes the Morse distress code (SOS: three short pulses, followed by three long pulses, followed by three short pulses). The Morse distress code is followed by one of the error codes listed in the tables shown in this guide.

For nShield HSMs running firmware 2.61.2 and above, the error code listed in this chapter is also reported by the `enquiry` utility in the `hardware status field` of the `Module`. You can also find it under `hardware errors` in the hardserver log (**network-attached HSMs**).

Errors are a rare occurrence. If any module goes into the error state, except as a result of you issuing the `Fail` command, contact Support, and give full details of your set up and the error code.

Contact Support even if you successfully recover from the error by taking the recommended action. The following troubleshooting pages might also be useful:

- Troubleshooting (network-attached HSMs)
- Troubleshooting 5s (nShield 5s HSMs)
- Troubleshooting (USB HSMs)

4.16.1. Reading Morse code

The following guidelines are useful when reading Morse code messages from the module:

- The duration of a dash (`-`) is 3 times the duration of a dot (`.`).
- The gap between components of a letter has the same duration as a dot.
- The gap between letters has the same duration as a dash.
- The duration of the gap between repeated series of letters (a Morse code word gap) is 7 times the duration of a dot.

The following table shows the error codes corresponding to numerals.

Numeral	Morse
1	. - - - -
2	. . - - -
3	. . . - -

Numeral	Morse
4	....-
5	.....
6	-....
7	--...
8	---..
9	----.
0	-----

4.16.2. Runtime library errors

Memory failures can occur if the module is exposed to excessive heat. If you experience these errors, check the ventilation around the module. The module generates considerable heat and, if not well ventilated, may be operating at too high a temperature, even if the rest of your server room is at an appropriate temperature.

The runtime library error codes could be caused by firmware bugs or by faulty hardware. If any of these errors is indicated, reset the module.

Code				Meaning
O L C	---	.-..	-.-.	SIGABRT: assertion failure and/or `abort()` called.
O L D	---	.-..	-..	Interrupt occurred when disabled.
O L E	---	.-..	.	SIGSEGV: access violation.
O L I	---	.-..	..	SIGSTAK: out of stack space.
O L J	---	.-..	.---	SIGFPE: unsupported arithmetic exception (such as division by 0).
O L K	---	.-..	-.-	SIGOSERROR: runtime library internal error.
O L N	---	.-..	-.	SIGFATALPANIC: error in error handling code.

Codes `OLD`, and `OLE` are more likely to indicate a hardware problem than a firmware problem.

To reset a unit that is in an error state, turn off the unit and then turn it on again.

4.16.3. Hardware driver errors

In general, the hardware driver error codes described in the following table indicate that

some form of automatic hardware detection has failed. As well as indicating simple hardware failure, one of these error codes could indicate that there is a bug in the firmware or that the wrong firmware has been loaded.

In the following table, the symbol "#" stands for a given numeral's Morse code representation.

If any of these errors is indicated, contact support.

Code						Meaning
HL	. . . .	. - . .				M48T37 NVRAM (or battery) failed
H B	. . . .	- . . .				Debug serial port initialization failed.
H C	. . . .	- . - .				Processing thread initialization failed.
HCP	. . . .	- . - .	. - - .			Card poll thread initialization failed.
H D	. . . .	- . .				Failure reading unique serial number.
H E	. . . .	.				EEPROM failed on initialization.
HF	. . . .	. . - .				Starting up crypto offload.
H I	. . . .	. .				Interrupt controller initialization failed.
H M	. . . .	- -				System hardware initialization failed.
H O	. . . .	- - -				Token interface initialization failed.
H R	. . . .	. - .				Random number generator failed.  This code may also be generated if an attempt is made to downgrade firmware on an nShield Solo+ to version 2.50.x or older.
HRS	. . . .	. - .	. . .			RNG startup failed.
HRTP	. . . .	. - .	-	. - - .		Periodic (scheduled daily) RNG selftest failed.
HRM	. . . .	. - .	- -			RNG data matched.
HS	. . . .	. . .				Unexpected error from SCSI controller or host interface initialization failed.
HV	. . . .	. . . -				Environment sensors failed (for example, temperature sensor)

Code						Meaning
HCV	. . . .	- . - .	. . . -			CPLD wrong version for PCI policing firmware.
H P P	. . . .	. - - .	. - - .			PCI Interface Policing failure.
HST	. . . .	. . .	-			Speed test failed.
H H R	. . . .	. . . .	. - .			RTC hardware detection failed or random number generator detection failed.
H R H	. . . .	. - .	. . . .			RNG hardware failed during operation
KR	- . -	. - .				RSA selftest failed.
H M *n*	. . . .	- -	#			DSP *n* failed self-test at start up.
H C *n* C A	. . . .	- . - .	#	- . - .	. -	CPU *n* failed self-test; no memory for cached RAM test.
H C *n* C C	. . . .	- . - .	#	- . - .	- . - .	CPU *n* failed self-test; CPU ID check failed.
H C *n* C F	. . . .	- . - .	#	- . - .	. . - .	CPU *n* failed self-test; freeing memory for cached RAM test.
H C *n* C G	. . . .	- . - .	#	- . - .	- - .	CPU *n* failed self-test; setting up cached RAM test.
H C *n* C R	. . . .	- . - .	#	- . - .	. - .	CPU *n* failed self-test; read error during cached RAM test.
H C *n* C V	. . . .	- . - .	#	- . - .	. . . -	CPLD version number incorrect (**PCIe HSMs**).
H C *n* C W	. . . .	- . - .	#	- . - .	. - -	CPU *n* failed self-test; write error during cached RAM test.
H C *n* H D	. . . .	- . - .	#	. . . .	- . .	DRBG *n* failed self-test.
H C *n* K A	. . . .	- . - .	#	- . -	. -	CPU n failed selftest - AES known-answer test.
H C *n* K B	. . . .	- . - .	#	- . -	- . . .	CPU n failed selftest - AES CMAC known-answer test.
H C *n* K C	. . . .	- . - .	#	- . -	- . - .	CPU n failed selftest - ECDSA known-answer test
H C *n* K E	. . . .	- . - .	#	- . -	.	CPU *n* failed self-test; DES known-answer test.
H C *n* K F	. . . .	- . - .	#	- . -	. . - .	CPU *n* failed self-test; Triple-DES known-answer test.

Code						Meaning
H C *n* K H	. . . .	- . - .	#	- . -	. . . .	CPU *n* failed self-test; SHA-1 known-answer test.
H C *n* K I	. . . .	- . - .	#	- . -	. .	CPU n failed selftest - HMAC-SHA512 known-answer test.
H C *n* K J	. . . .	- . - .	#	- . -	. - - -	CPU n failed selftest - HMAC-SHA256 known-answer test.
H C *n* K M	. . . .	- . - .	#	- . -	- -	CPU *n* failed self-test; HMAC-SHA1 known-answer test.
H C *n* K N	. . . .	- . - .	#	- . -	- .	CPU n failed selftest - HMAC-SHA224 known-answer test.
H C *n* K P	. . . .	- . - .	#	- . -	. - - .	CPU n failed selftest - HMAC-SHA384 known-answer test.
H C *n* K R	. . . .	- . - .	#	- . -	. - .	CPU n failed selftest - RSA known-answer test
H C *n* K S	. . . .	- . - .	#	- . -	. . .	CPU *n* failed self-test; DSA known-answer test.
H C *n* L C	. . . .	- . - .	#	. - . .	- . - .	CPU *n* failed self-test; locking check.
H C *n* P S	. . . .	- . - .	#	. - - .	. . .	CPU *n* failed self-test; test terminated at start.
H C *n* RT	. . . .	- . - .	#	. - .	-	CPU n failed selftest - RTC check.
H C *n* S A	. . . .	- . - .	#	. . .	. - - .	CPU *n* failed self-test; no memory for uncached RAM test.
H C *n* S F	. . . .	- . - .	#	. . .	. . - .	CPU *n* failed self-test; freeing memory for uncached RAM test.
H C *n* S R	. . . .	- . - .	#	. . .	. - .	CPU *n* failed self-test; read error during uncached RAM test.
H C *n* S W	. . . .	- . - .	#	. . .	. - -	CPU *n* failed self-test; write error during uncached RAM test.
H C *n* T S	. . . .	- . - .	#	-	. . .	CPU *n* failed self-test; could not start test.

4.16.4. Maintenance mode errors

The following error codes indicate faults encountered when a module is in the maintenance mode.

Code				Meaning	Action
I D	. .	- . .		Copies of metadata do not match when trying to run image.	Contact Support.
I H	. .	. . . .		Bad metadata: hash mismatch.	Repeat firmware upgrade.
I I	. .	. .		Execution image does not match metadata.	Contact Support.
I L	. .	. - . .		Bad metadata: either bad length or bad metadata when running loadboot application.	Repeat firmware upgrade.
I M	. .	- -		Bad metadata: malformed ImageMetaData.	Repeat firmware upgrade.
I P	. .	. - - .		Bad metadata: bad padding.	Repeat firmware upgrade.
I R	. .	. - .		Bad metadata: extra bytes at end.	Repeat firmware upgrade.
I S	. .	. . .		Image entry point not found.	Contact Support.
I U	. .	. . -		Bad metadata: ROM blank.	Repeat firmware upgrade.
I X	. .	- . . -		Bad metadata: malformed header.	Repeat firmware upgrade.
J H	. - - -	. . . .		Both copies of metadata invalid.	Contact Support.
H Z E	. . . .	- - . .	.	Monitor checksum failed.	Contact Support.
K F E	- . -	. . - .	.	Flash sector erase failed.	Repeat firmware upgrade.
K F P	- . -	. . - .	. - - .	Flash sector program failed.	Repeat firmware upgrade.
M M D	- -	- -	- . .	No memory for download buffer.	Contact Support.

4.16.5. Operational mode errors

The following runtime library error codes could be caused by either bugs in the firmware or faulty hardware.

Code				Meaning	Action
D	- . .			Fail command received.	Reset module by turning it off and then on again.
T	-			Temperature of the module has exceeded the maximum allowable.	Restart your host computer, and improve module cooling.

Code				Meaning	Action
G G G	- - .	- - .	- - .	Failure when performing `ClearUnit` or `Fail` command.	Contact Support.
I J A	..	. - - -	. -	Audit logging: failed to send audit log message.	Contact Support.
I J B	..	. - - -	- ...	Audit logging: no module memory (therefore failed to send audit log message).	Contact Support.
I J C	..	. - - -	-.-.	Audit logging: key problem or FIPS incompatibility (therefore failed to sign audit log message).	Contact Support.
I J D	..	. - - -	-..	Audit logging: NVRAM problem (therefore failed to configure or send audit log message).	Contact Support.

SOS IJA can occur for any type of log message:

- log message
- signature block
- certifier block

To improve the cooling of your PCIe module, increase the distance between PCIe cards, and increase the airflow through your host computer.

4.16.6. Solo XC tamper event errors

The following error codes indicate a hard tamper event has occurred on a Solo XC module. The Solo XC will become non-operational if tamper event error is indicated.

If a tamper event error occurs the Solo XC module must be destroyed or returned to Entrust.

Code					Meaning	Action
TT	-	-			Hard temperature tamper	Contact Support
VV	. . . -	. . . -			Hard voltage tamper	Contact Support
T	.				Soft temperature tamper	Contact Support
V	. . . -				Soft voltage tamper	Contact Support

Code					Meaning	Action
B	- . . .				Low battery voltage, <2.5V	Contact Support
HI2C	. . . .	. .	. . - - -	- . - .	I2C Failure	Contact Support
WD0	. - -	- . .	- - - - -		Watchdog 0 failure	Contact Support
WD1	. - -	- . .	. - - - -		Watchdog 1 failure	Contact Support
WD2	. - -	- . .	. . - - -		Watchdog 2 failure	Contact Support
WD3	. - -	- . .	. . . - -		Watchdog 3 failure	Contact Support

4.16.7. Other errors

Code	Meaning	Action
SFP	The Security Fuse Processor (SFP) has failed and is unable to handle further requests sent from the client's hardserver.	Restart the HSM. This resets the SFP. If this does not resolve the issue, or the SFP fails again, contact Entrust Support.

For information on error codes not listed on this page, contact nShield Support: nshield.support@entrust.com.

4.17. Troubleshooting 5s

This guide covers the following HSMs:

- nShield 5s

It describes what to do if there is an issue with the HSM, or the Security World Software.

If you encounter any errors that are not listed in the following table, contact Support.

4.17.1. nShield 5s running out of space due to signed HSM system logs

This type of error indicate that the HSM is running out of disc space. It is possible that the logs generated by the HSM are taking up disc space.

Error	Explanation	Action	Example
hardware status: ncoreapi service terminated unexpectedly; last log entry was: <date and time> Process exited with status 137 from signal KILL	The HSM has reached it's critical storage limit and has now entered a monitor only state.	Use `hsmadmin` command to export and expire the signed system logs from the nShield 5s. Reboot the nShield 5s and restart the hardserver to bring the HSM out of its failed state. Make sure that the nShield 5s is enrolled in nShield Audit Log service to avoid the storage limit reaching a critical level. To learn more about this service, see nShield Audit Log Service.	Example command to export and expire signed system logs: `hsmadmin logs export --expire --esn AAAA-BBBB-CCCC --outdir ./`

4.18. Virtualization Remote Server

Virtualization provides an environment where multiple operating systems can run at the same time on one physical computer. Each virtual machine is an isolated, virtualized computer system that can run its own operating system.

The nShield Solo XC is compatible with the leading server virtualization and hypervisor man agement platforms, including:

- **Microsoft Hyper-V**, a role in Windows Server used to create and manage a virtualized server computing environment
- **VMware vSphere / ESXi**, a robust, bare-metal hypervisor that installs directly onto your physical server.

  All vSphere management functions are performed through remote management tools.

- **Citrix XenServer** - includes the XenCenter management console.

  PCI passthrough is configured using the XenCenter software with command line tools and utilities. PCI passthrough allows a VM client direct access to the nShield Solo XC.

The operating system that runs within a virtual machine is referred to as a *guest operating system*.

nShield software includes the nShield hardserver applications. These applications enable applications running on multiple virtual guest operating systems to all share nShield Solo XC

hardware.

Hardserver processing services can be shared among multiple virtual operating system instances as long as each instance has hardserver installed. Inside of the operating system, hardservers can communicate with other hardservers.

4.18.1. Virtualization and Hyper-V

The host hardserver is configured to run on the Parent/Dom0 operating system. The Parent/Dom0 operating system has privileged access to the Solo XC hardware over the PCI bus.

Instead of using a physical network for communication between the VM guest instances running on the same physical system, most hypervisors provide the capability to instantiate some form of virtual switch which allows the network communication to take place between the VMs entirely within the hypervisor software. This means that nCore data does not need to be routed outside of the server hardware.

4.18.2. Virtualization and XenServer/VMware vSphere hypervisor, ESXi

ESXi and XenServer do not use the concept of a Parent/Dom0 VM. Instead, an additional VM is defined in the system as the host with passthrough permissions to enable access to the nShield Solo XC.

4.18.3. ESXi environment

After installing VMware ESXI, the VM guest can be remotely managed and the PCI passthrough of the Solo module configured using vSphere. PCI passthrough allows a VM guest direct access to the nShield Solo XC.

4.18.3.1. Set up a basic single-node vCenter server instance

Follow the steps below to use the vCenter Simple Install to set up a basic single-node vCenter Server instance. You will install the vSphere Web Client and use its in-browser interface to add ESXi hosts to your vSphere inventory.

1. Log on the system as administrator and start at least one ESXi host.
2. Install ESXi using the vCenter Simple Install option using the instructions provided in the VMware vSphere documentation (https://docs.vmware.com/en/VMware-vSphere/index.html).

3. Install the vSphere Web Client using the instructions provided in the VMware vSphere documentation.

4.18.3.2. Configure passthrough devices on a host

Follow the steps below to add ESXi hosts to the vCenter Server inventory, in order to create a vSphere environment and use vSphere features.

1. Enter the IP address, username and root password of your host created when you installed ESXi.
2. Select **Login**, the **Getting Started** page will be displayed.
3. Select the **Configuration** tab.
4. Select **Advanced Settings**.
5. Select **Configure Passthrough**. The **Passthrough Configuration** page is displayed listing all available passthrough devices.
6. Select **Edit**.
7. Select the check box to mark the endpoint for passthrough.

   For example, the check mark box for 02:00.0 will be **Freescale Semiconductor Inc <class> Power PC**.

8. Select **OK**.

ESXi will now be successfully installed and the Solo PCIe module has been configured for passthrough.

4.18.3.3. Create the VM guest instance

VMware ESXi provides the capability of PCI passthrough and it is a bare metal Hypervisor. This requires the creation of two or more guests which communicate via Vswitch. One of the guest will act as the primary guest and will be configured as described below utilizing the PCI card connected via passthrough. The second and subsequent guests can be composed of the identical configuration with the exception of the PCI passthrough connection.

To create the VM guest instance:

1. Navigate to **File > New > Virtual Machine** in the vSphere Client. A wizard will prompt you through each of the settings displayed in the working pane.
2. Select **Typical Configuration** and then select **Next**.
3. Enter a name and select **Next**.
4. Select a storage device for the VM files.

5. Select a Guest Operating System (OS) and an OS version from the drop down menu.
6. Select **Next**.
7. Configure the network connections as follows:
    a. How many NICs do you want to connect? **1**.
    b. Network: **VM Network**.
    c. Adapter: **VMXNET 3**.
    d. Connect at Power On: ✓.
8. Select **Next**.
9. Configure the virtual disk size for the guest VM as follows:

It is important to select the same network configuration for both the guest primary VM and the guest secondary VM, as it is a require ment for IP communication between the two.

    a. Datastore: **<datastore1>**.
    b. Available space (GB): **<357.3>**.
    c. Virtual disk size: **50 GB**.
    d. Select **Thick Provisioned Lazy Zeroed**.
10. Select **Next**.
11. Select **Edit the virtual machine settings before completion**.
12. Select **Continue**.
13. Select **Add**.
14. Select **PCID**.
15. Select **Next**.
16. Select the configured PCI passthrough device.

    For example, 02:00.0 will be **Freescale Semiconductor Inc <class> Power PC**.

17. Select **Next**.
18. Select **Finish**.

4.18.4. XenServer environments

Install the XenServer, follow the instructions in the *Citrix XenServer Quick Start Guide*, see https://docs.citrix.com/en-us/xenserver.

4.18.4.1. Configure the XenCenter client

To remotely manage VM guests and configure PCI passthrough of the nShield Solo XC:

1. Enter the XenServer web client IP address.
2. Select **XenCenter installer**. The XenCenter software will auto install.
3. Select the XenServer that you want to connect to and manage from the Resources pane. A connection is established providing access to all the VMs installed on the server.
4. Select the **Console** tab from the **Properties** tabs pane.

Dom0 is the initial domain started by the Xen hypervisor on boot. Dom0 runs the Xen management toolstack and is has direct access to the hardware. Dom0 provides Xen virtual disks and network access for VM guests, each VM guest is referred to as a DomU (that is, an unprivileged domain).

5. Run the command `lspci`.

   A detailed list of all the PCI buses and devices in the system is displayed, for example:

   ```
   02:00.0 Power PC:  Freescale Semiconductor Inc Device 082c (rev11)02:00:0
   ```

   represents the nShield Solo XC card endpoint.

6. Open the file `/boot/extlinux.conf` and scroll to the dom0 Linux kernel append section. Add the PCI slot as shown below with the following command:

   ```
   pciback.hide=(02:00.0)
   ```

Newer versions of Citrix XenServer utilize:

```
xen-pciback.hide=(xx:xx:x)
```

7. Scroll to the end of the file.
8. Run the command:

   ```
   pciback.hide=<NG solo card endpoint>
   ```

   This command enters the PCI slot, for example:

   ```
   pciback.hide=(02:00.0) --- /boot/initrd-fallback.img
   ```

9. Save and close the file.
10. Run the command:

```
extlinux -I /boot
```

11. Run the command:

```
reboot
```

12. Run the command:

```
xe vm-list
```

13. Locate the **uuid** using the VI Editor for the VM that you want to assign the PCI passthrough to.
14. Run the command:

```
xe vm-param-set other-config:pci=0/0000:<endpoint of the NG solo card> uuid: <uuid>
```

This command adds the PCI device to the selected VM, for example:

```
xe vm-param-set other-config:pci=0/0000:02:00.0 uuid: 4a4ab965-a91d-70e7-2ec-a4c0004e1e8d
```

If a PCI passthrough needs to be removed from a specific guest VM, run the command:

```
xe vm-param-clear param-name=other-config uuid=<vm uuid>
```

When the installation of XenCenter has completed, you can access `[https://( XENSERVER-IP)]` to acquire the corresponding XenCenter Client Remote management interface.

4.18.4.2. Create a XenServer guest instance and hardserver configuration

The XenServer is a bare metal Hypervisor that provides the PCI passthrough capability. As part of this process, you must create two **Dom U** guests that communicate through the Vswitch. One guest acts as the primary guest and is configured as described below utilizing the PCI card connected via passthrough. The second guest can be composed of the identical configuration with the exception of the PCI passthrough connection.

To create the first DomU guest VM:

1. Select the server from the **Resources** pane, right-click and select **New VM** from the

dropdown menu.

2. Select a **Template**.
3. Select an **Operating system** for the first DomU guest VM.
4. Select **Next**.
5. Select **Name**.
6. Enter a name and select **Next**.

The DomU guest VM name will also be displayed in the XenCenter's Resources pane. You can change the name at any time.

7. Select **Installation Media**.
8. Select **Install from ISO library or DVD drive** and then select the appropriate media from the drop down menu.
9. Select **Next**.
10. Select **Home Server**.
11. Select **Place the VM on this server** and then select a home server from the drop down list of available servers.
12. Select **Next**.
13. Select **CPU & Memory** and enter the number of CPUs, chose your topology and enter an amount for memory.
14. Select **Next**.
15. Select **Storage**.
16. Select **Use these virtual disks:** and select a virtual disk from the display.
17. Select **Next**.
18. Select **Networking** and select the virtual network interface.
19. Select **Finish**.

If the guest VM is configured to have a PCI module via passthrough and the module is not connected to the VM instance, the guest VM instance will fail to power on. Verify that the Solo XC card is located on the same slot that was selected for the passthrough to the guest VM.

4.18.5. Hyper-V environment

The instructions assume there is a single nShield Solo XC module in the system.

The commands starting with `PS C:\>` should be run in PowerShell in ele-

> vated mode.

4.18.5.1. Set up

4.18.5.1.1. Install Hyper-V on the server

Follow the instructions in the Windows documentation for Hyper-V, see https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/.

4.18.5.1.2. Add the Hyper-V role to the server

To add the Hyper-V role in Windows server:

1. Log in as Administrator.
2. Open **Server Manager**.
3. Select **Manage**.
4. Select **Add Roles and Features**.
5. Select **Next**.
6. Select **Role-based or feature-based installation**.
7. Select **Next**.
8. Select **Select a server from the server pool**.
9. Select a server that has Windows 2016 installed. You will be adding Hyper-V to this server.
10. Select **Next**.
11. Select **Hyper-V**.
12. Select **Next**.
13. Reboot the system.

Once rebooted, Hyper-V will be supported by the Server 2016 instance.

4.18.5.1.3. Prepare the server

1. Enable the Input Output Memory Management Unit (IOMMU) policy on the server. This policy controls whether the Hyper-V server uses an IOMMU. To enable it, run the command:

```
bcdedit /set hypervisoriommupolicy enable
```

2. Check no devices are already set up for VM. Run the command:

```
PS C:\> Get-VMHostAssignableDevice
```

4.18.5.1.4. Prepare the device

1. Display the device address. Run the command:

```
PS C:\> (Get-PnpDevice -PresentOnly).Where{ $_.InstanceId -like '*VEN_1957*' } | Format-Table -autosize
```

2. Disable the device. Run the command:

```
PS C:\> Disable-PnpDevice -Verbose -InstanceId $instanceId -Confrm:$false
```

To find the `$instanceId` run the command:

```
PS C:\> $instanceId = (Get-PnpDevice -PresentOnly).Where{ $_.InstanceId -like
'*VEN_1957*' } | select -expand InstanceId
```

3. Dismount the device. Run the command:

```
PS C:\> $locationPath = Dismount-VmHostAssignableDevice -LocationPath $locationPath -Force -Verbose
```

To find the `$locationPath` run the command:

```
PS C:\> $locationPath = (Get-PnpDeviceProperty -KeyName
DEVPKEY_Device_LocationPaths -InstanceId $instanceId).Data[0]
```

4. Verify that the device is disabled and dismounted. Run the command:

```
PS C:\> Get-VMHostAssignableDevice
```

4.18.5.1.5. Install the Security World software

Install the Security World software suite into the operating system of the guest VM. Once the suite is installed, you can initialize the hardserver and then configure the guest VMs.

1. Insert the DVD-ROM containing the Security World software. The Security World software will auto install.
2. Run the `enquiry` utility to check that the module is working correctly. See Checking the installation.

4.18.5.1.6. Create the VM guest instance on the server

1. Open the Hyper-V Manager within your Windows 2016 server.
2. Log in as Administrator.
3. Navigate to **Action > New > Virtual Machine**.
4. Select **Next** (to create a virtual machine with a custom configuration).
5. Enter a name for the new guest VM instance.

   Use the default location setting.

6. Select **Next**.
7. Select the OS generation to be installed on the new guest VM instance.

   For example, **Generation 2** is selected. **Generation 2** is valid for products such as Windows 8 and beyond and with Windows Server 2016.

8. Select **Next**.
9. Select an amount of memory for allocation to this guest VM instance.
10. Select **Next**.
11. Select **Next**.
12. Select **Create a virtual hard disk**.
13. Enter **Name**, **Location** and **Size**.
14. Select **Next**.
15. Select one of the following options:
    - **Install an operating system later**, if you have a disk.
    - **Install an operating system from a bootable image file**, if you have the ISO path.
16. Select **Next**.
17. Select **Finish**.

4.18.5.1.7. Configure the VM guest instance on the server

1. Stop and select the VM guest instance. Run the commands:

```
PS C:\> $vmName = 'ws2016'

PS C:\> Stop-VM -VMName $vmName
```

2. Turn off the Automatic Stop Action. Run the command:

```
PS C:\> Set-VM -VMName $vm Name -AutomaticStopAction TurnOff
```

3. Make sure the memory minimum bytes match the memory startup bytes. Run the command:

```
PS C:\> Set-VM -VM $vm -DynamicMemory -MemoryMinimumBytes 4096MB -MemoryMaximumBytes 16384MB
-MemoryStartupBytes 4096MB
```

4. Assign a device to the VM guest instance. Run the commands:

```
PS C:\> Add-VMAssignableDevice -VM $vmName -LocationPath $locationPath -Verbose

PS C:\> Start-VM -VMName $vmName
```

To find the $locationPath run the command:

```
PS C:\> $locationPath = (Get-PnpDeviceProperty -KeyName
DEVPKEY_Device_LocationPaths -InstanceId $instanceId).Data[0]
```

It is possible to assign the same device to a single VM guest instance multiple times. In this case the VM will not start. To check currently assigned devices, run the command below. To remove an assigned device see Remove a device from the VM guest instance.

```
PS C:\> Get-VMAssignableDevice -VMName $vmName
```

4.18.5.2. Remove a device from the VM guest instance

1. Remove a device from the VM. Run the commands:

```
PS C:\> $vmName = “ws2016”

PS C:\> Remove-VMAssignableDevice -Verbose -VMName $vmName}
```

4.18.5.3. Undo passthrough

1. Mount a single device. Run the command:

```
Mount-VMHostAssignableDevice -Verbose -LocationPath $locationPath
```

To find the `$locationPath` run the command:

```
PS C:\> $locationPath = (Get-PnpDeviceProperty -KeyName
DEVPKEY_Device_LocationPaths -InstanceId $instanceId).Data[0]
```

2. Enable a single device in device manager. Run the command:

```
Enable-PnpDevice -Confirm:$false -Verbose -InstanceId $instanceId
```

To find the `$locationPath` run the command:

```
PS C:\> $locationPath = (Get-PnpDeviceProperty -KeyName
DEVPKEY_Device_LocationPaths -InstanceId $instanceId).Data[0]
```

4.19. Product returns

If you need to return your nShield HSM, contact Entrust nShield Support for instructions: https://trustedcare.entrust.com/

Before sending a nShield HSM back to Entrust, you should return it to factory state. If the HSM is non-functional, you can send it back to Entrust without returning it to factory state.

See Remove modules and delete Security Worlds for more information about returning a module to factory state.

The following links might also be helpful:

- **nShield 5 HSMs:** Return to factory state
  See also the documentation for the hsmadmin factorystate command.
- **Network-attached HSMs with a front panel:** Resetting and testing the nShield HSM: factory state
- factorystate

4.20. Regulatory notices

This page applies to the following HSMs:

- nShield 5s
- nShield Solo
- nShield Solo XC

The HSMs listed on this page comply with Part 15 of the FCC rules. Operation is subject to the following two conditions:

1. The device may not cause harmful interference, and
2. The device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the users will be required to correct the interference at their own expense.

4.20.1. Canadian certification - CAN ICES-3 (A)/NMB-3(A)

4.20.2. Battery cautions

Danger of explosion if the battery is incorrectly replaced. The battery may only be replaced with the same or equivalent type. Dispose of the used battery in accordance with your local disposal instructions.

The battery cautions apply to the nShield 5s and Solo XC.

4.20.3. Hazardous substance caution

This product contains a lithium battery and other electronic components and materials which may contain hazardous substances. However, this product is not hazardous providing it is used in the manner in which it is intended to be used.

The hazardous substance caution applies to the nShield 5s and Solo XC.

4.20.4. Recycling and disposal information

For recycling and disposal guidance, see the nShield product's *Warnings and Cautions* documentation.

4.21. Battery replacement

Entrust can provide replacement batteries for nShield Solo XC and nShield 5s HSMs if

required. If you prefer to source your own third-party batteries, you must ensure they meet the minimum requirements.

4.21.1. Minimum requirements

The following specification documents the key parameters of the backup battery used on nShield Solo XC and nShield 5s HSMs, as supplied by the Original Equipment Manufacturer (OEM).

If you replace the battery in one of these products in the field, ensure that the component sourced meets the following requirements as a minimum.

Adherence to these parameters is critical to the safety certification of the overall product. Entrust cannot accept responsibility for damage or loss of performance due to incorrect battery replacement.

nShield Solo XC and nShield 5c Backup Lithium Cell

Size	CR 1/3N
Nominal Voltage	3V
Typical capacity	170mAh
Chemistry	Li-MnO2 (lithium manganese dioxide)
Temperature characteristics	≤-25°C to ≥70°C
Max. safe reverse current	2mA
UL approved and marked	UL 1642

4.21.2. Replace a battery on an nShield Solo XC or nShield 5s HSM

Dispose of the used battery in accordance with local regulations.

Required tools

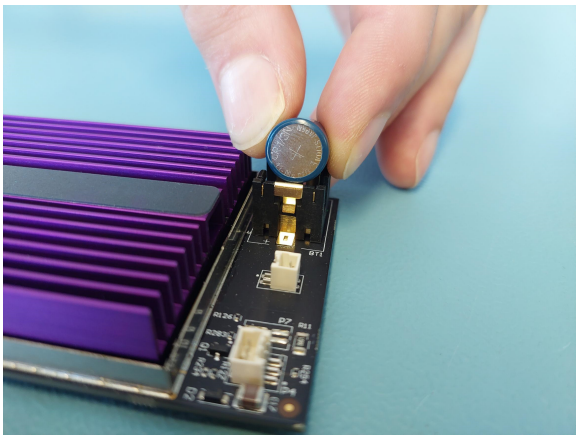
- Small non-conductive tweezers

Required part

- Orderable part number: SOLOXC-REP-BATT (Replacement battery)
  See Minimum requirements for battery requirements if you are providing your own.

To remove and replace the battery:

1. Power off the system and while taking ESD precautions, remove the module.
2. Place the module on a flat surface.
3. Gently remove the battery from the BT1 connector. If you cannot remove the battery with your fingers, use a pair of non-conductive tweezers.



4. Observing the polarity, install the replacement battery in the BT1 connector.
5. Re-install the module into the PCIe slot.