nShield Security World

nShield v13.6.5 HSM User Guide

08 January 2025

Table of Contents

1. Introduction . . . 1
    1.1. Read this guide if 1
        1.1.1. Terminology . . . 1
    1.2. Model numbers . . . 2
    1.3. Security World Software . . . 3
        1.3.1. Software architecture . . . 4
        1.3.2. Default directories . . . 5
        1.3.3. Utility help options . . . 7
    1.4. Setting the PATH for nShield utilities . . . 7
    1.5. Further information . . . 7
    1.6. Security advisories . . . 8
    1.7. Recycling and disposal information . . . 8
2. nShield Network-Attached HSMs . . . 9
    2.1. Physical security of the HSM . . . 9
        2.1.1. Tamper event . . . 9
        2.1.2. Physical security checks . . . 11
        2.1.3. Replacing the fan tray module and PSU . . . 13
    2.2. Front panel controls . . . 15
        2.2.1. Display screen and controls . . . 16
        2.2.2. Using the front panel controls . . . 18
        2.2.3. Using a keyboard to control the unit . . . 21
    2.3. Top-level menu . . . 22
    2.4. Basic HSM, RFS and client configuration . . . 23
        2.4.1. About nShield HSMs and client configuration . . . 24
        2.4.2. Basic HSM and RFS configuration . . . 25
        2.4.3. Basic configuration of the client to use the HSM . . . 47
        2.4.4. Basic configuration of an HSM to use a client . . . 50
        2.4.5. Restarting the hardserver . . . 53
        2.4.6. Zero touch configuration of an HSM . . . 54
        2.4.7. Checking the installation . . . 56
        2.4.8. Using a Security World . . . 57
    2.5. Client software and module configuration: network-attached HSMs . . . 57
        2.5.1. About user privileges . . . 59
        2.5.2. About client configuration . . . 59
        2.5.3. Basic HSM and remote file system (RFS) configuration . . . 61
        2.5.4. Configuring the nShield HSM to use the client . . . 69
        2.5.5. Changing the nShield HSM configuration from the Front Panel to use a

client . . . 75
2.5.6. Configuring client computers to use the nShield HSM . . . 75
2.5.7. Client Session Resumption (Impath Resilience) . . . 79
2.5.8. Client Licensing . . . 79
2.5.9. Configuring NTP in the nShield HSM . . . 81
2.5.10. Configuring Remote Syslog . . . 84
2.5.11. Set up client cooperation . . . 85
2.5.12. Routing . . . 87
2.5.13. Configuring an nShield HSM using the Serial Console . . . 91
2.5.14. Stopping and restarting the hardserver . . . 99
2.5.15. Resetting and testing the nShield HSM . . . 100
2.6. HSM and client configuration files . . . 102
2.6.1. Location of client configuration files . . . 102
2.6.2. Location of module configuration files . . . 102
2.6.3. Structure of configuration files . . . 103
2.6.4. Sections only in module configuration files . . . 105
2.6.5. Sections in both module and client configuration files . . . 112
2.6.6. Sections only in client configuration files . . . 123
2.7. Checking and changing the mode on a network-attached HSM . . . 127
2.7.1. Front panel controls . . . 127
2.7.2. Available modes . . . 127
2.7.3. Identifying the current mode . . . 128
2.7.4. Changing the mode . . . 130
2.8. Upgrading the image file and associated firmware: network-attached HSMs . . . 131
2.8.1. Version Security Number (VSN) . . . 131
2.8.2. Key data . . . 132
2.8.3. Upgrading the Connect image . . . 132
2.8.4. Upgrading the Connect image using the front panel . . . 132
2.8.5. Upgrading the nShield HSM from a privileged client . . . 133
2.8.6. After firmware installation . . . 136
2.9. Troubleshooting . . . 136
2.9.1. Checking operational status . . . 136
2.9.2. Module overheating . . . 142
2.9.3. Log messages for the module . . . 142
2.9.4. Utility error messages . . . 143
2.9.5. Storage error . . . 144
2.10. HSM maintenance . . . 144
2.10.1. Flash testing the module . . . 145
2.11. Approved accessories . . . 145

2.12. Resource Watchdog . . . 145
2.12.1. Enabling or disabling the Watchdog . . . 146
2.12.2. Understanding the default settings . . . 146
2.12.3. Reading or modifying the Watchdog Configuration File . . . 149
2.12.4. Troubleshooting . . . 151
2.13. Valid IPv6 Addresses . . . 152
2.14. Remote File System Volumes . . . 153
2.14.1. Allow custom RFS paths with an environment variable . . . 154
2.14.2. Allow custom RFS paths with a configuration file . . . 154
3. nShield PCIe HSMs . . . 156
3.1. Client software and module configuration: PCIe and USB HSMs . . . 156
3.1.1. About user privileges . . . 156
3.1.2. Set up client cooperation . . . 156
3.1.3. Configuring the hardserver . . . 158
3.1.4. Stopping and restarting the hardserver . . . 164
3.2. Hardserver configuration files . . . 165
3.2.1. Hardserver configuration files . . . 165
3.2.2. General hardserver configuration settings . . . 167
3.2.3. Sections only in client configuration files . . . 177
3.3. Checking and changing the mode on an nShield Solo module . . . 179
3.3.1. Back panel and jumper switches . . . 180
3.3.2. Physical mode switch . . . 180
3.3.3. Remote mode switch . . . 181
3.3.4. Override switches . . . 183
3.3.5. Changing the mode . . . 183
3.3.6. Status indications . . . 187
3.4. nShield 5s modes of operation . . . 187
3.4.1. Modes of operation . . . 187
3.4.2. Check and change the mode of operation . . . 188
3.4.3. Return to factory state . . . 189
3.4.4. Recovery mode . . . 191
3.5. Upgrade firmware: nShield 5s . . . 193
3.5.1. Primary, recovery and bootloader firmware . . . 193
3.5.2. Firmware version control . . . 193
3.5.3. Firmware on the installation media . . . 195
3.5.4. Firmware installation overview . . . 195
3.6. Upgrade firmware: nShield Solo, Solo XC, and Edge HSMs . . . 197
3.6.1. Version Security Number (VSN) . . . 197
3.6.2. Firmware on the installation media . . . 198

3.6.3. Using new firmware . . . 199
3.6.4. Firmware installation overview . . . 199
3.6.5. Upgrading both the monitor and firmware . . . 200
3.6.6. Upgrading firmware only . . . 202
3.6.7. After firmware installation . . . 204
3.7. Setting the system clock . . . 204
3.7.1. Setting the HSM system clock . . . 205
3.7.2. System interaction with the system clock . . . 206
3.7.3. Checking the system clock . . . 206
3.7.4. Adjusting the system clock . . . 207
3.8. Checking the installation . . . 208
3.8.1. Checking operational status . . . 208
3.8.2. Mode switch and jumper switches (nShield Solo and Solo XC only) . . . 211
3.8.3. Log message types . . . 211
3.8.4. BadTokenData error (Solo only) . . . 213
3.9. HSM status indicators and error codes (nShield 5s) . . . 213
3.9.1. LED status . . . 214
3.9.2. LED error states . . . 214
3.9.3. Error codes accessed remotely . . . 217
3.10. HSM Status indicators (nShield Solo and Solo XC) . . . 220
3.11. Regulatory notices . . . 221
3.11.1. Canadian certification - CAN ICES-3 (A)/NMB-3(A) . . . 222
3.11.2. Battery cautions . . . 222
3.11.3. Hazardous substance caution . . . 222
3.11.4. Recycling and disposal information . . . 222
3.12. Battery replacement . . . 222
3.12.1. Minimum requirements . . . 222
3.12.2. Replace a battery on an nShield Solo XC or nShield 5s HSM . . . 223
3.13. Replace the fan (Solo XC) . . . 224
3.14. Set up communication between host and module (nShield 5s HSMs) . . . 225
3.14.1. Overview of SSH keys . . . 225
3.14.2. Installation of SSH keys as part of software installation . . . 226
3.14.3. Installation of SSH keys independently of a software installation . . . 226
3.14.4. Viewing installed SSH keys . . . 227
3.14.5. Changing installed SSH keys . . . 227
3.14.6. Making a backup of installed SSH keys . . . 228
3.14.7. Restoring SSH keys from backup . . . 228
3.14.8. Preparing an HSM for use in another host . . . 228
3.15. SSH Client Key Protection (nShield 5s HSMs) . . . 229

3.15.1. SSH Services . . . 229
3.15.2. SSH Client Key Encryption . . . 230
3.15.3. Setting Protections on SSH keys . . . 231
3.15.4. Permissions on SSH keys . . . 231
3.16. Troubleshooting 5s . . . 232
3.16.1. nShield 5s running out of space due to signed HSM system logs . . . 232
3.17. Virtualization Remote Server . . . 233
3.17.1. Virtualization and Hyper-V . . . 234
3.17.2. Virtualization and XenServer/VMware vSphere hypervisor, ESXi . . . 234
3.17.3. ESXi environment . . . 234
3.17.4. XenServer environments . . . 236
3.17.5. Hyper-V environment . . . 239
4. nShield USB HSMs . . . 245
4.1. Using the nShield Edge . . . 245
4.1.1. Controls, card slot, and LEDs . . . 245
4.1.2. Mode LEDs . . . 246
4.1.3. Changing the mode . . . 246
4.1.4. Status LED . . . 246
4.2. Checking and changing the mode on an nShield Edge . . . 247
4.2.1. Mode LEDs . . . 247
4.2.2. Status LED . . . 248
4.3. nShield Edge Windows compatibility issues and considerations . . . 248
4.3.1. nShield Edge very slow in VMware virtual machine . . . 249
4.4. Troubleshooting . . . 250
4.4.1. None of the LEDs are lit . . . 250
4.4.2. The Mode LED is amber or red . . . 250
4.4.3. The Status LED is flashing irregularly and the nShield Edge is unresponsive for more than a few minutes . . . 250
4.4.4. The Security World Software does not detect the connected nShield Edge . . . 250
4.4.5. Upgrading the firmware . . . 250
4.5. Regulatory notices . . . 251
4.5.1. Canadian certification - CAN ICES-3 (A)/NMB-3(A) . . . 251
4.5.2. Recycling and disposal information . . . 251
5. HSM Management . . . 252
5.1. Optional features . . . 252
5.1.1. Persistence of features . . . 252
5.1.2. Enabling features . . . 252
5.1.3. Available optional features . . . 253

5.1.4. Ordering additional features . . . 258
5.1.5. Enable features on a network-attached HSM . . . 259
5.1.6. Enable features on PCIe and USB HSMs . . . 262
5.2. Administration of platform services (nShield 5 HSMs) . . . 263
5.2.1. hsmadmin . . . 264
5.3. Client cooperation . . . 281
5.3.1. Configure client cooperation . . . 282
5.3.2. Remove a cooperating client . . . 283
5.4. Morse code error messages . . . 284
5.4.1. Reading Morse code . . . 285
5.4.2. Runtime library errors . . . 286
5.4.3. Hardware driver errors . . . 286
5.4.4. Maintenance mode errors . . . 289
5.4.5. Operational mode errors . . . 290
5.4.6. Solo XC tamper event errors . . . 291
5.4.7. Other errors . . . 292
5.5. Working with CodeSafe . . . 292
5.5.1. CodeSafe applications . . . 292
5.5.2. Use a standalone application (nShield Connect) . . . 293
5.5.3. CodeSafe setup for the nShield 5c . . . 294
5.5.4. The csadmin utility tool (nShield 5 HSMs) . . . 298
5.6. Warrant Management . . . 317
5.6.1. Warrant management for nShield Solo and nShield Edge . . . 318
5.6.2. Warrant management for nShield Connect + and nShield Connect XC . . . 321
5.6.3. Warrant management for nShield 5s and nShield 5c . . . 321
5.7. Remote HSMs . . . 321
5.8. Remote Operator . . . 322
5.8.1. About Remote Operator . . . 322
5.8.2. Configuring Remote Operator . . . 323
5.8.3. Creating OCSs and keys for Remote Operator . . . 332
5.9. Using nShield commands from PowerShell . . . 335
5.9.1. Install and configure PowerShell . . . 336
5.9.2. Calling nShield commands at the PowerShell prompt . . . 337
5.9.3. PowerShell modes: interactive and batch . . . 337
5.9.4. Input pipelines . . . 338
5.9.5. Secure strings . . . 339
5.10. Preload Utility . . . 340
5.10.1. Overview . . . 340
5.10.2. Using Preload . . . 341

5.10.3. Preload File . . . 344
5.10.4. Softcard Support . . . 344
5.10.5. FIPS Auth . . . 345
5.10.6. Admin Keys . . . 346
5.10.7. High Availability . . . 346
5.10.8. Logging . . . 353
5.10.9. Using preloaded objects - Worked example . . . 354
5.11. Audit Logging . . . 355
5.11.1. Aims of audit logging . . . 355
5.11.2. Configuring audit logging . . . 356
5.11.3. Commands audited . . . 358
5.11.4. Audit log contents . . . 361
5.11.5. Audit log administration . . . 365
5.12. nShield Audit Log Service . . . 365
5.12.1. Introduction . . . 366
5.12.2. nShield Audit Log Service configuration . . . 366
5.12.3. Warrants . . . 370
5.12.4. Log databases . . . 371
5.12.5. Reading signed system logs . . . 373
5.12.6. Reading nCore audit logs . . . 376
5.12.7. Monitoring and managing audit data sources . . . 380
5.13. Configure the hardserver to export the module for guest VM usage . . . 384
5.14. Logging, debugging, and diagnostics . . . 386
5.14.1. Logging and debugging . . . 386
5.14.2. Diagnostics and system information . . . 395
5.14.3. How data is affected when a module loses power and restarts . . . 397
5.15. System logging (nShield 5 HSMs) . . . 397
5.15.1. Maximum log size . . . 398
5.15.2. Interaction with the system clock . . . 399
5.15.3. Init log . . . 399
5.15.4. System log . . . 400
5.16. Maintenance of nShield Hardware . . . 403
5.16.1. Voltage Monitoring for Battery Replacement (nShield Solo XC and nShield 5s) . . . 404
5.16.2. Temperature Monitoring for Airflow Validation . . . 405
5.17. Physical security of the HSM . . . 406
5.17.1. Tamper event . . . 407
5.17.2. Physical security checks . . . 408
5.17.3. Replacing the fan tray module and PSU . . . 410

5.18. Application Performance Tuning . . . 412
5.18.1. Job Count . . . 412
5.18.2. Client Configuration . . . 413
5.18.3. Highly Multi-threaded Client Applications . . . 413
5.18.4. File Descriptor Limits (Linux) . . . 414
5.19. Platform services and（nShield 5 HSMs) . . . 414
5.19.1. ncoreapi service . . . 414
5.19.2. Platform services . . . 414
5.19.3. Separation of services . . . 415
5.20. System upgrade . . . 416
5.20.1. Terminology . . . 416
5.20.2. Software and firmware compatibility . . . 416
5.20.3. System upgrade procedure . . . 416
5.21. Product returns . . . 418

1. Introduction

The nShield HSM User Guide provides useful information about your nShield HSMs. You should consult it before and while using a new HSM.

It contains a section for each type of HSM (network-attached, PCIe, and USB). There is also a section that contains information that applies to all types of HSM.

1.1. Read this guide if ...

Read this guide if you need to configure or manage an Entrust Hardware Security Module (HSM).

nShield hardware security modules use the Security World paradigm to provide a secure environment for all your HSM and key management operations. Guides are also available to help with the following tasks:

- Installing an nShield HSM
- Installing the nShield Security World software for an nShield HSM
- Using and administrating a Security World

nShield network-attached HSMs are connected to a network by an Ethernet connection. Each network-attached HSM is configured to communicate with one or more client computers on the network. You can also configure clients to make use of any other network-connected HSMs on the network, as well as locally connected HSMs.

All nShield HSMs support standard cryptography frameworks and integrate with many standards based products.

This guide assumes that:

- You are familiar with the basic concepts of cryptography and Public Key Infrastructure (PKI)
- You have read the relevant Security world and HSM documentation.
- You have installed your nShield HSM.

1.1.1. Terminology

Some information only applies to a specific HSM or HSM type. Where this is the case, the relevant HSM or HSM type is mentioned by name, otherwise the terms nShield "HSM", "hardware security module", or "module" are used interchangeably.

nShield HSMs by type:

nShield HSMs	nShield HSM type
Connect, Connect +, Connect XC, 5c	Network-attached HSMs
Solo, Solo +, Solo XC, 5s	PCIe HSMs
Edge	USB-attached HSMs

1.2. Model numbers

▼ *Details*

Model numbering conventions are used to distinguish different nShield hardware security devices.

Model number	Used for
NH2047	nShield Connect 6000
NH2040	nShield Connect 1500
NH2033	nShield Connect 500
NH2068	nShield Connect 6000+
NH2061	nShield Connect 1500+
NH2054	nShield Connect 500+
NH2075-B	nShield Connect XC Base
NH2075-M	nShield Connect XC Medium
NH2075-H	nShield Connect XC High
NH2082	nShield Connect XC SCAP
NH2089-B	nShield Connect XC Base - Serial Console
NH2089-M	nShield Connect XC Mid - Serial Console
NH2089-H	nShield Connect XC High - Serial Console
NH3003-B	nShield Connect CLX Base - Serial Console
NH3003-M	nShield Connect CLX Mid - Serial Console

Model number	Used for
NH3003-H	nShield Connect CLX High - Serial Console
NH2096-B	nShield 5c Base
NH2096-M	nShield 5c Medium
NH2096-H	nShield 5c High
nC3nnnE-nnn, nC4nnnE-nnn	nShield Solo PCIe
nC30n5E-nnn, nC40n5E-nnn	nShield Solo XC PCIe
NC5536E-B	nShield 5s Base
NC5536E-M	nShield 5s Medium
NC5536E-H	nShield 5s High
nC30nnU-10, nC40nnU-10	nShield Edge

1.3. Security World Software

PCIe and USB HSMS

The hardserver software controls communication between applications and Entrust nShield product line HSMs, which may be installed locally or remotely. It runs as a daemon (**Linux**) or a service (**Windows**) on the host computer.

The Security World for nShield is a collection of programs and utilities, including the hardserver, supplied by Entrust to install and maintain your nShield security system. The Security World Software includes the following:

- The appropriate installer for the client platform
- The client hardserver
- A set of utilities for configuring the nShield HSM
- A set of utilities and the KeySafe application for performing key management tasks on nShield HSMs.

Entrust provides the firmware that runs on the nShield HSM, and software to run on each client computer. The nShield HSM is supplied with the latest version of the HSM firmware installed. For more information about:

- Upgrading the firmware, see

    - Upgrade firmware: nShield Solo, Solo XC, and Edge HSMs (Solo and Edge models).
    - Upgrade firmware: nShield 5s nShield 5s.
    - Upgrading the image file and associated firmware: network-attached HSMs (network-attached HSMs).
- Installing and configuring the software on each client computer, see nShield Security World Software v13.6.5 Installation Guide and Client software and module configuration: PCIe and USB HSMs (PCIe and USB HSMs) or Client software and module configuration: network-attached HSMs (network-attached HSMs).
- The supplied utilities, see nShield v13.6.5 Utilities Reference.
- Maintenance of your nShield hardware, see Maintenance of nShield Hardware.

1.3.1. Software architecture

The software, firmware, and utilities have version numbers and there is also a version number for the World which refers to the World data that is stored in encrypted form on the client computer, typically in the `opt/nfast/kmdata` (**Linux**) or `C:\ProgramData\nCipher\Key Management Data` (**Windows**) directory or on the RFS. This data includes information concerning the World itself and also concerning each key that was created within that World. The World version created is determined by the version numbers of the software and firmware used when it was first created, see nShield Security World v13.6.5 Management Guide.

The latest World version is version 3. You can query the version of the World loaded on your system by using the command `kmfile-dump`.

1.3.1.1. Hardserver (network-attached HSMs)

The *hardserver* software controls communication between the internal security module and applications on the network.

Separate instances of the hardserver run on the unit and each client that is configured to work with the unit. There is a secure channel, known as the *impath*, between the two software instances, which forms a single secure entity for transferring data between the unit and the clients. See also Compatibility issues.

The unit's hardserver is configured using the front panel on the unit, or by means of uploaded configuration data. Configuration data is stored on the unit and in files in a specially configured file system on each client computer. For more information about using:

- The front panel to configure the unit, see Front panel controls

- The specially configured file system to configure the unit and the client, see Client software and module configuration: network-attached HSMs.

1.3.1.2. Remote file system (RFS) (network-attached HSMs)

Each unit uses a *remote file system* (RFS). You can configure the RFS on any computer, but it is normally located on the first client that is configured. The RFS contains:

- The master configuration information for the unit
- The Security World files
- The key data.

Do not copy the master configuration to file systems on other clients. You can copy Security World files and key data to other clients to allow you to manage the unit from more than one client. To make it available to the unit, copy to the RFS the data for Security Worlds, cards or keys that you create on a client that does not contain the RFS.

1.3.2. Default directories

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

Directory name	Default path (Linux)	Environment variable (Windows)	Default path (Windows)
nShield Installation	`/opt/nfast/`	`NFAST_HOME`	`C:\Program Files\nCipher\nfast`
Key Management Data	`/opt/nfast/kmdata/`	`NFAST_KMDATA`	`C:\ProgramData\nCipher\Key Management Data`
Dynamic Feature Certificates	`/opt/nfast/femcerts/`	`NFAST_CERTDIR`	`C:\ProgramData\nCipher\Feature Certificates`
Static Feature Certificates	`/opt/nfast/kmdata/hsm-ESN/features`		`%NFAST_KMDATA%\hsm-ESN\features` (**network-attached HSMs**) `%NFAST_KMDATA\features` (**PCIe and USB HSMs**)
Log Files	`/opt/nfast/log`	`NFAST_LOGDIR`	`C:\ProgramData\nCipher\Log Files`
User Log Files	`/home/<user>/nshieldlogs`	`NFAST_USER_LOGDIR`	`C:\Users\<user>\nshieldlogs`

Directory name	Default path (Linux)	Environment variable (Windows)	Default path (Windows)
Remote Static Feature Certificates	`opt/nfast/kmdata/hsm-ESN/features`		`%NFAST_KMDATA%\hsm-ESN\features` (**network-attached HSMs**) `%NFAST_KMDATA\features` (**PCIe and USB HSMs**)
Remote Dynamic Feature Certificates	`opt/nfast/kmdata/hsm-ESN/features`		`%NFAST_KMDATA%\hsm-ESN\features` (**network-attached HSMs**) `%NFAST_KMDATA\features` (**PCIe and USB HSMs**)

By default, the Windows `C:\ProgramData\` directory is a hidden directory. To see this directory and its contents, you must enable the display of hidden files and directories in the View settings of the Folder Options.

Dynamic feature certificates must be stored in the directory stated above. The directory shown for static feature certificates is an example location. You can store those certificates in any directory and provide the appropriate path when using the Feature Enable Tool. However, you must not store static feature certificates in the dynamic features certificates directory.

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

The instructions in this guide refer to the locations of the software installation and program data directories as follows:

- By name (for example, Key Management Data).
- **Linux**: By absolute path (for example, `/opt/nfast/kmdata`).
- **Windows**: By nShield environment variable names enclosed with percent signs (for example, `%NFAST_KMDATA%`).

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

- **Linux**: Create a symbolic link from `/opt/nfast/` to the directory where the software is actually installed.

- **Windows**: Ensure that the associated nShield environment variables are re-set with the correct paths for your installation. For more information about creating symbolic links, see your operating system's documentation.

1.3.3. Utility help options

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

- `-h|--help` displays help for the utility
- `-v|--version` displays the version number of the utility
- `-u|--usage` displays a brief usage summary for the utility.

1.4. Setting the PATH for nShield utilities

It is recommended that the PATH environment variable be changed to include `opt/nfast/bin` (**Linux**) or `<%NFAST_HOME%\bin>`, which is usually `C:\Program Files\nCipher\nfast\bin` (**Windows**).

This is the directory in the nShield installation that contains the nShield command-line utilities and some DLLs.

This will allow all the nShield command-line utilities to be run without the need to type the full path, for example running `enquiry` instead of `opt/nfast/bin/enquiry>` (**Linux**) or `<%NFAST_HOME%\bin\enquiry>` (**Windows**).

`opt/nfast/bin` (**Linux**) or `<%NFAST_HOME%\bin>` (**Windows**) must be set in the PATH in order to use the OpenSSL module in the Python that is bundled with nShield.

The Python bundled with nShield is located in `opt/nfast/python3/bin` (**Linux**) or `%NFAST_HOME\python3\bin`, which is usually `C:\Program Files\nCipher\nfast\python3\bin` (**Windows**). If using the nShield Python, you may additionally want to add this directory to the PATH environment variable so that you can run the nShield python as just the python command. You may not want to do this if you are also using other Python installations on the same machine.

1.5. Further information

This guide forms one part of the information and support provided by Entrust.

If you have installed the Java Developer component, the Java Generic Stub classes,

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

1.6. Security advisories

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

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

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

1.7. Recycling and disposal information

For recycling and disposal guidance, see the nShield product's Warnings and Cautions documentation.

2. nShield Network-Attached HSMs

2.1. 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.

2.1.1. 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.

2.1.1.1. 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.

2.1.1.2. 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

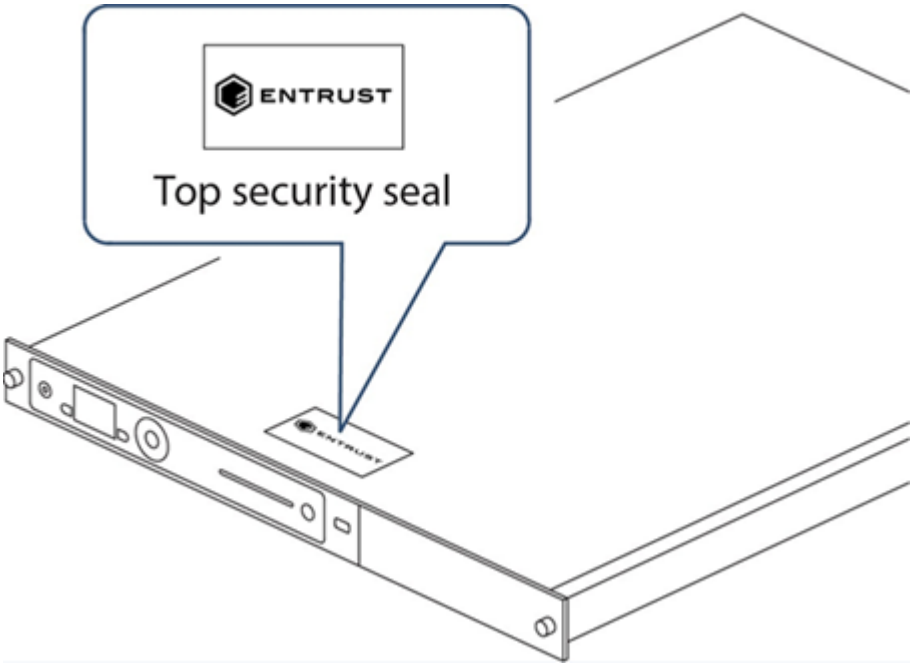
to a factory state. If the lid remains open, the above message will remain on the screen and all button presses are ignored.

2.1.2. 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.

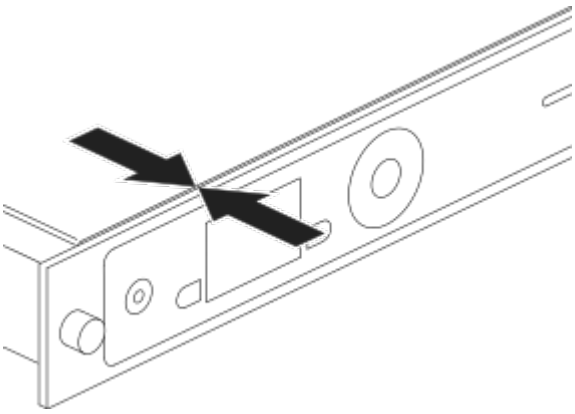
To determine if the security of the nShield HSM is compromised:

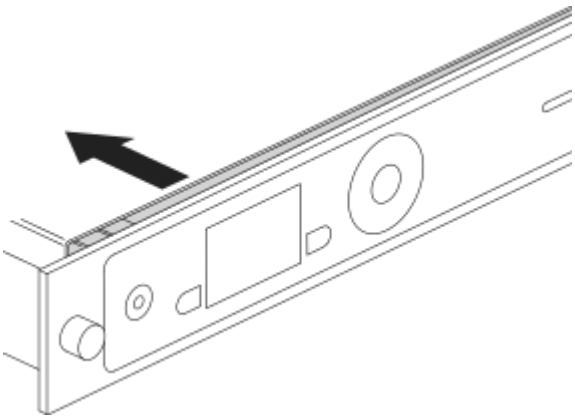
1. Check that the physical security seal is authentic and intact. Look for the holographic foil bearing the nCipher logo. Look for cuts, tears and voiding of the seal. The seal is located on the top of the nShield HSM chassis.



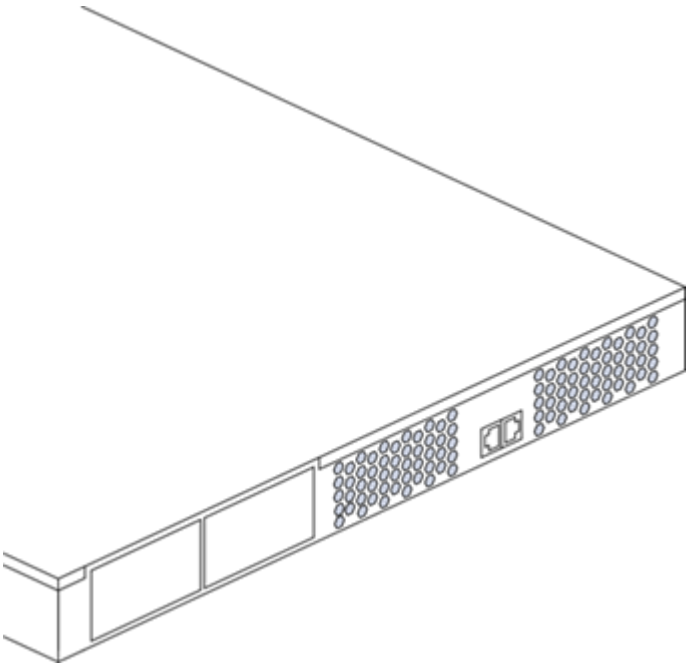
   For information about the appearance of intact and damaged security seals, see the *Physical Security Checklist*.

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.



2.1.3. 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.

2.1.3.1. 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*.

2.1.3.1.1. 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.

2.1.3.2. 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*.

2.1.3.2.1. 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.

2.1.3.3. Battery life when storing the nShield HSM

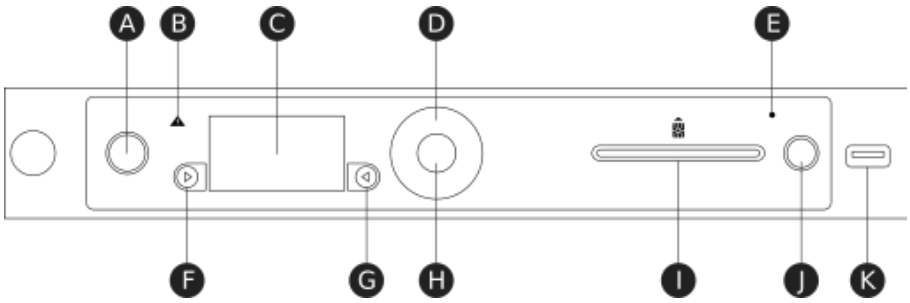
If a nShield HSM has been in storage for an extended period of time the fan tray module may need replacement.

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.

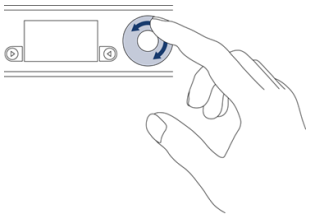
2.2. Front panel controls

This guide covers the following HSMs:

- nShield Connect
- nShield 5c

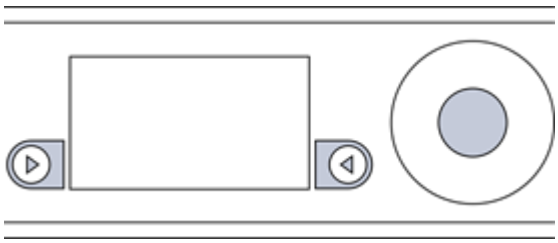


Key	Description
A	Power button
B	Warning LED (orange)
C	Display screen
D	Touch wheel
E	Status indicator LED (blue)
F	Display navigation button (left)
G	Display navigation button (right)
H	**Select** button
I	Slot for smart cards
J	Clear button This button is obsolete on the nShield 5c and is only retained for backward compatibility, it has no functionality.
K	USB connector



Use the touch wheel to change values or move the cursor on the display screen. To confirm a value, press the **Select** button.

2.2.1. Display screen and controls

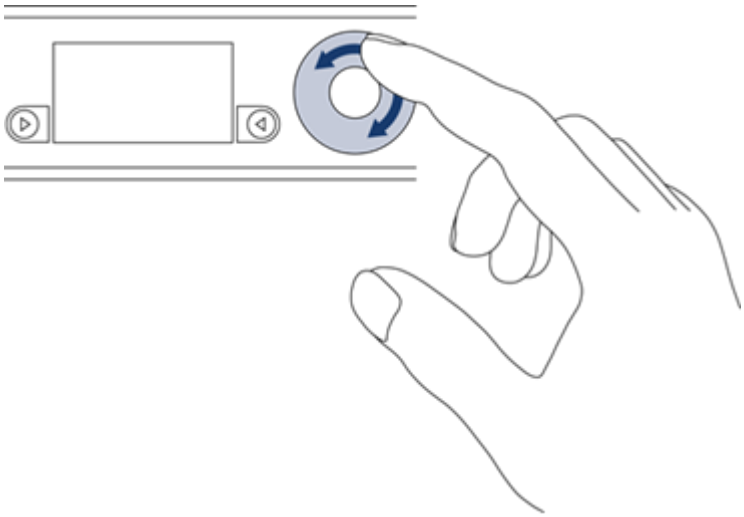


When the unit is powered, the display screen displays a menu or a dialog.

Each menu or dialog includes onscreen navigation labels that appear at the bottom of the display screen, on either side next to the display navigation buttons. Press the button next to the label to perform the action specified by the label.

To go back to the previous dialog or menu screen, use the navigation button to the left of the screen. To confirm a dialog value or select a menu option, either:

- Press the navigation button to the right of the screen.
- Touch the Select button.



Use the touch wheel to changes values or move the cursor on the display screen. To confirm a value, touch the Select button.

2.2.1.1. Menu screens

You can access menus from the display screen.

Menus are displayed as a list of selectable options. An onscreen arrow points to the currently selectable option. If the menu has more than four options, an arrow indicates the direction in which more options are available.

To select a menu option:

1. Move the indicator arrow up or down with the touch wheel.
2. When the indicator arrow points to the option you want to select, either:
    - Press the navigation button to the right of the screen (labeled onscreen as **SELECT**).
    - Touch the Select button.

At the top right of the display screen, a number sequence indicates the path to the current option. The last digit of the sequence shows the location of the menu you are currently viewing. The top level menu has no numbers, but when you select the System menu, the number **1** is shown.

The preceding digits in the sequence show the position of each option in turn that was selected in previous menu screens to reach the current menu. For example, the sequence **1-2** shows that the indicator is on the second option of the menu that was reached by selecting the first option on the top-level menu.

For a map of the menu screens, see Top-level menu.

2.2.1.2. Dialogs

For some tasks, a dialog is displayed onscreen. When the dialog opens, the cursor is in the first field. To change and then enter values:

1. Use the touch wheel to change the displayed value of the fields.
2. Touch the Select button to enter the displayed value and move to the next field in the dialog.

Repeat the procedure to enter all necessary values in the dialog.

2.2.1.3. Information display

When you use a dialog to request information (for example, a log or details of a key), there is often too much information to display onscreen. In such cases, only the first part of the information is displayed.

To view the rest of the information:

- Use the touch wheel to scroll the displayed information in the direction indicated by the onscreen arrows.
- When an **Options** label is displayed, press the right-hand navigation button to see a menu of navigation options. You can normally choose to go to the top, to the bottom, or to a specified line in the display.

The numbers of the lines currently being displayed onscreen are shown at the left of the screen. They are followed in parentheses (**( )**) by the total number of lines available for display.

2.2.2. Using the front panel controls

You can use the front panel controls to configure the unit and to perform other tasks described in this guide. When the unit is working over the network with another computer (a client computer), you can program and control the unit as if it were part of the client computer.

If the unit is powered down while you are logged in, you are logged out automatically.

2.2.2.1. Start-up information

When you turn on power to the unit and it has completed its initialization, the lower part of the display screen shows basic start-up information about the unit.

There is a series of start-up information topics available. By default, the first displayed topic is the current **System time**. Use the touch wheel to view the other start-up information topics.

2.2.2.2. Administrative control of the unit

You can view and control the status of the unit by using the front panel controls and menu options.

Tasks	Action
Understand and control the power status of the unit	Use the Power button to power up the unit. If the Power button is not illuminated, the unit is not powered. The Power button flashes intermittently as the unit powers up. It also flashes when the unit is in standby mode. For more information about the Power button, see Troubleshooting
Control access to the unit	You can control access to the menus on the unit and the Power button on the front panel by using **System > System configuration > Login settings**. When **UI Lockout with OCS** has been enabled, you must log in with an authorized Operator Card before you can access the menus. You can still view information about the unit on the start-up screen. When you are logged in, you can log out and leave the unit locked. When **UI Lockout without OCS** has been enabled, you cannot access the menus, but you can still view information about the nShield HSM on the start-up screen. The only way to disable this setting (apart from returning the HSM to factory state) is to push an updated configuration file to the nShield HSM. See About user privileges and *ui_lockout* for more information. **Power button lockout** can be enabled and disabled independently when UI Lockout allows access to the menus.
Unlock the unit	When **UI Lockout with OCS** has been enabled and you have logged out, the display screen displays the label **Login** next to the right-hand navigation button. Press the right-hand navigation button, then insert an Operator Card that has been authorized for login, and follow the onscreen instructions.

Tasks	Action
Log out of the unit	Select **Logout**. This option is not available if UI Lockout with OCS has not enabled.
Put the unit in standby mode	Press the Power button or select **System > Shutdown/Reboot > Shutdown**.
Restore the unit to its original configuration	Select **System > System configuration > Default config**.
Restore the unit to its factory state	Select **System > Factory state**.
Clear the memory of the internal hardware security module	Use the Clear button (**nShield Connect**), run `nopclearfail --clear --all` (**nShield 5c**), or select **HSM > HSM reset**.
View information about the current state of the internal hardware security module	Select **HSM > HSM information**.
View information about the current state of the system	See the next section.
Set the Real-Time Clock on the unit	Select **Security World mgmt > Admin operations > Set secure RTC**. After setting the RTC on an nShield 5c, you need to reboot the module.
Change the mode of the unit	Select **HSM > Set HSM mode**. • Select **Operational** mode to run the unit normally. • Select **initialization** mode to configure the unit with software utilities rather than the front panel.

2.2.2.3. Viewing the current status of the unit

To view information about the current state of the system, from the main menu select **System > System information**. Select an option to view the associated information as follows:

Option	Description
View system log	Displays the system log.
View hardserver log	Displays the module hardserver log.
Display tasks	Displays the tasks that the system is currently performing.
Component versions	Displays the version numbers of the various system software components.

Option	Description
View h/w diagnostics	Displays the following environmental information about the module: • The current temperature at the left and right sensors • The minimum and maximum previous temperature at each sensor • The voltage on each power rail • The speed of each fan.
View tamper log	Displays the tamper log.
View unit id	Displays the ID of the unit.

2.2.2.4. Viewing the mode of the unit

See Identifying the current mode.

2.2.3. Using a keyboard to control the unit

You can connect a keyboard to the USB connector on the front panel. You can connect either a US or a UK keyboard. To configure the unit for your keyboard type, select **System > System configuration > Keyboard layout** and then choose the keyboard type you require.

When you have connected a keyboard and configured the unit for its use, you can enter numbers and characters directly into the display. You can also control the unit by using the following keystrokes:

Keystroke	Use
F1	Same as pressing the left-hand navigation button on the front panel.
F2	Same as pressing the right-hand navigation button on the front panel.
F3	Same as touching the Select button.
Esc	Same as pressing the left-hand navigation button on the front panel.
Enter	Where the Select button is active, same as pressing Select: where Button B is active, same as pressing Button B.
Up arrow	Moves the indicator upwards in a menu.
Down arrow	Moves the indicator downwards in a menu.
Tab	Moves the cursor to the next field in a dialog.
Shift-Tab	Moves the cursor to the previous field in a dialog.
PgUp	Displays the previous screen.

Keystroke	Use
PgDn	Displays the next screen.

2.3. Top-level menu

This guide covers the following HSMs:

- nShield Connect
- nShield 5c

If you select an option, the module displays the menu options in the level below.

If you cancel a selected option, you return to level above.

Submenus for options marked with * depend on the settings of the module.

```
1 System
    1-1 System configuration
        1-1-1 Network config
            1-1-1-1 Set up interface #1
            1-1-1-2 Set up interface #2
            1-1-1-3 Set up bond
            1-1-1-4 Set default gateway
            1-1-1-5 Set up routing *
            1-1-1-6 Show routing table
            1-1-1-7 Ping remote host
            1-1-1-8 Trace route to host
        1-1-2 Hardserver config
        1-1-3 Remote file system
        1-1-4 Client config *
        1-1-5 Resilience config
        1-1-6 Config file options
            1-1-6-1 Fetch configuration
            1-1-6-2 Client config push
        1-1-7 Log config
        1-1-8 Date/time setting
        1-1-9 Keyboard layout
            1-1-9-1 UK keyboard
            1-1-9-2 US keyboard
        1-1-10 Default config
        1-1-11 Remote configuration options
            1-1-11-1 Remote mode change
    1-2 System information
        1-2-1 View system log
        1-2-2 View hardserver log
        1-2-3 View IPv6 addresses
        1-2-4 Display tasks
        1-2-5 Component versions
        1-2-6 View h/w diagnostics
            1-2-6-1 View power readings
            1-2-6-2 View other readings
            1-2-6-3 Critical Errors
        1-2-7 View tamper log
        1-2-8 View unit id
    1-3 Login settings
        1-3-1 Enable UI Lockout
            1-3-1-1 UI Lockout with OCS
```

```
            1-3-1-2 UI Lockout w/out OCS
        1-3-2 Power switch lockout
        1-3-3 Login control status
    1-4 Upgrade system
    1-5 Factory state
    1-6 Shutdown/Reboot
        1-6-1 Shutdown
        1-6-2 Reboot
2 HSM
    2-1 HSM information
        2-1-1 Display details
        2-1-2 Display secure RTC
        2-1-3 Speed test
        2-1-4 Display statistics
    2-2 HSM reset
    2-3 HSM feature enable
        2-3-1 Read FEM from card
        2-3-2 Read from a file
        2-3-3 View current state
        2-3-4 Write state to file
    2-4 Set HSM mode
        2-4-1 Operational
        2-4-2 Initialization
3 Security World mgmt
    3-1 Display World info
    3-2 Module initialization
        3-2-1 New Security World
        3-2-2 Load Security World
        3-2-3 Erase Security World
    3-3 RFS operations
        3-3-1 Update World files
        3-3-2 Remove RFS lock
    3-4 Admin operations
        3-4-1 Replace ACS
        3-4-2 Recover keys
        3-4-3 Recover PIN
        3-4-4 Set secure RTC
    3-5 Cardset operations
        3-5-1 Create OCS
        3-5-2 List cardsets
    3-6 Card operations
        3-6-1 Card details
        3-6-2 Check PIN
        3-6-3 Change PIN
        3-6-4 Erase card
    3-7 Keys
        3-7-1 List keys
        3-7-2 Verify key ACLs
    3-8 Set up remote slots *
    3-9 Set up dynamic slots
        3-9-1 Dynamic Slots
        3-9-2 Slot mapping
4 CodeSafe *
```

2.4. Basic HSM, RFS and client configuration

This guide covers the following HSMs:

- nShield Connect
- nShield 5c

The nShield nToken has been discontinued. If you already have one installed in your RFS, you can use the nToken options as described here. If you do not currently use an nToken, you must use software-based authentication.

This guide describes the initial nShield HSM, RFS and client computer configuration steps. For more about:

- Security World Software installation and options, see Install the Security World software.
- The menu options, see Top-level menu.
- Advanced HSM and client configuration options, see HSM and client configuration files.

An installation will have only one RFS, but may have one or more Clients. The RFS can also dual role as a Client. Before you can continue with the following configuration, the RFS and every Client must have the Security World software installed, see Install the Security World software.

2.4.1. About nShield HSMs and client configuration

An nShield HSM and a client communicate using their hardservers. These handle secure transactions between the HSM and applications that run on the client. To enable the secure transmission of data between an nShield HSM and each client, the hardserver uses an impath. You must configure:

- Each client hardserver to communicate with the hardserver of the HSM that it needs to use.
- The HSM hardserver to communicate with the hardserver of the clients that are allowed to use it.

Multiple nShield HSMs can be configured to communicate with one client, just as multiple clients can be configured to communicate with one HSM.

Any nShield HSM on the network can load a Security World securely over the network, and access its keys, wherever the HSM is installed.

Security World and key data is stored on the file system of the nShield HSM, where it is updated whenever card or key operations are performed on the HSM. The data is also automatically transferred to the *remote file system* (RFS). If required, you can also share the

data with client computers that use the Security World.

2.4.1.1. Remote file system (RFS)

Each HSM must have a remote file system (RFS) configured. This includes master copies of all the files that the HSM needs.

2.4.1.2. HSM configuration

The current configuration files for the hardserver of an HSM are stored in its local file system. These files are automatically:

- Updated when the HSM is configured.
- Exported to the appropriate RFS directory.

Each HSM in a Security World has separate configuration files on the RFS.

2.4.1.3. Client configuration

The current configuration files for the hardserver of a client are stored in its local file system.

The following steps assume that you have added the path `%NFAST_HOME%\bin` (Windows) or `/opt/nfast/bin/` (Linux) to the `PATH` system variable.

2.4.2. Basic HSM and RFS configuration

After installing the Security World Software and the HSM, you need to do the following:

- Configure the HSM Ethernet interfaces.
- Configure the RFS.

You should complete the RFS tasks before:

- Configuring the HSM and client to work together.
- Creating a Security World and an Operator Card Set (OCS).

2.4.2.1. Configuring the Ethernet interfaces - IPv4 and IPv6

An HSM communicates with one or more clients over an Ethernet network. You must

supply IP addresses for the HSM and the client. Contact your system administrator for this information if necessary.

There are two network interfaces on the HSM. Three configurations are supported:

- Single network interface.
- Two independent network interfaces.

  You must connect the interfaces to physically different networks.

- The two network interfaces combined as a bond interface.

  The bond interface can use:

  - Active backup mode.
  - 802.3ad mode (requires a switch that supports 802.3ad).

You can configure the HSM using the front panel **Network config** menu or by pushing a configuration file to the HSM over the network. The following can be configured:

- Interface addresses
- Bond
- Default gateway
- Network routes
- Network speed.

If the HSM is already configured, you can update the displayed values.

If you ever change any of the IP addresses on the HSM, you must update the configuration of all the clients that work with it to reflect the new IP addresses.

By default, the hardserver listens on all interfaces. However, you can choose to set specific network interfaces on which the hardserver listens. This may be useful in cases such as if one of the Ethernet interfaces is to be connected to external hosts. See Optionally configure hardserver interfaces for more information.

2.4.2.1.1. IPv4 and IPv6

Support for IPv6 is in addition to IPv4. Both Ethernet interfaces can be configured to support:

- IPv4 only
- IPv4 and IPv6

- IPv6 only.

Interface#1 is enabled by default and cannot be disabled. Interface #2 is disabled by default and can be enabled and disabled.

IPv6 addresses

An IPv4 address is 32 bits long and typically represented as 4 octets, for example 192.168.0.1. An IPv6 address is 128 bits long and is made up of a subnet prefix (n bits long) and an interface ID (128 - n bits long).

An IPv6 address and its associated subnet is typically represented by the notation *ipv6-address/prefix-length*, where:

- *ipv6-address* is an IPv6 address represented in any of the notations described below.
- *prefix-length* is a decimal value specifying how many of the leftmost contiguous bits of the address make up the prefix.

The IPv6 address notation mirrors the way subnets are represented in the IPv4 Classless Inter-Domain Routing (CIDR) notation.

IPv6 address notation

An HSM will accept an IPv6 address if it is entered in one of the forms shown below and if the address is valid for context in which it is used. There are two conventional forms for representing IPv6 addresses as text strings:

- The long representation is x:x:x:x:x:x:x:x, where each `x` is a field containing hexadecimal characters (`0` to `ffff`) for each 16 bits of the address.

  For example:

  `1234:2345:3456:4567:5678:6789:789a:89ab`

  `1234:5678:0:0:0:0:9abc:abcd/64`

- If one or more consecutive fields are 0 then they can be replaced by `::`.

  For example:

  `1234:5678:0:0:0:0:9abc:abcd/64` can be written as `1234:5678::9abc:abcd/64`

  `::` can only appear once in an IPv6 address.

Unless the address is a link-local address, the HSM front panel only allows lower-case letters in an IPv6 address.

IPv6 addresses keyed manually on the HSM front panel are validated on entry by the HSM. As well as checking that the format of the address is correct, the HSM also validates that the address entered is valid for the context in which it will be used, see Acceptable IPv6 address by use case.

If Stateless Address Auto Configuration (SLAAC) is enabled the HSM will automatically form IPv6 addresses from network prefixes contained in Router Advertisements (RAs). RAs are received directly by the HSM Operating System and automatically forms IPv6 addresses by combining the network prefixes contained in the RA with the MAC address of the receiving Ethernet interface. As they are created by the Operating System, SLAAC IPv6 addresses are not subject to the same validation rules as addresses entered via the HSM front panel. If SLAAC is to be used to configure HSM IPv6 addresses in preference to statically entered addresses, then network planners must take care to ensure that prefixes advertised to the HSM are of a suitable type, see Acceptable IPv6 address by use case.

(**nShield Connect only**) IPv6 compliance

The sub-menu (1-1-1-9 - **Set IPv6 compliance**) on the nShield Connect front panel menu permits the User to select an IPv6 compliance mode for an HSM. Compliance with **USGv6** or **IPv6 ready** can be selected.

Both these modes change the settings for the HSM firewall so that it will pass-through packets which are discarded in the normal **Default** mode. This behaviour is required for compliance testing but is not recommended for normal use since allowing packets with invalid fields or parameters through the firewall increases the attack surface. When either **USGv6** or **IPv6 ready** are selected, a confirmation message is displayed to reduce the likelihood that they are enabled by accident.

It is recommended that the IPv6 compliance mode is set to **Default** for all normal operations.

Acceptable IPv6 address by use case

The types of IPv6 which are acceptable as a static address are given in the table below For examples of valid IPv6 addresses, see Valid IPv6 Addresses.

Use Case	Acceptable Address Type
Static IPv6 Address Entry	• Global Unicast • Local Unicast

Use Case	Acceptable Address Type
IPv6 Default Gateway	• Global Unicast • Local Unicast • Link-local
IPv6 Route Entry - IP Range	• Unknown • Loopback • Global Unicast • Local Unicast • Link local • Teredo • Benchmarking • Orchid • 6to4 • Documentation • Multicast
IPv6 Route Entry - Gateway	• Global Unicast • Local Unicast • Link-local
RFS Address	• Global Unicast • Local Unicast
Client Address	• Global Unicast • Local Unicast
Push Client Address	• Global Unicast • Local Unicast
Ping	• Unknown • Loopback • Global Unicast • Local Unicast • Link-local • Teredo • Benchmarking • Orchid • 6to4 • Documentation • Multicast

Use Case	Acceptable Address Type
Traceroute	• Unknown • Loopback • Global Unicast • Local Unicast • Link-local • Teredo • Benchmarking • Orchid • 6to4 • Documentation • Multicast

2.4.2.1.2. Stateless address auto-configuration (IPv6 only)

Unlike IPv4, IPv6 is designed to be auto-configuring. SLAAC is an IPv6 mechanism by which IPv6 hosts can configure their IPv6 addresses automatically when connected to an IPv6 network using the Neighbour Discovery Protocol (NDP). Using NDP IPv6 hosts are able to solicit advertisements from on-link routers and use the network prefix(es) contained in the advertisements to generate IPv6 address(es).

SLAAC is disabled by default in the HSM, but can be selectively enabled for each Ethernet interface either using the front panel or by setting the appropriate configuration item and pushing a configuration file.

2.4.2.2. Configure Ethernet interface #1

To set up Ethernet interface #1 (default):

2.4.2.2.1. Enable/disable IPv4

To enable/disable IPv4:

1. From the front panel menu, select **System > System configuration > Network config > Set up interface #1 > Configure #1 IPv4 > IPv4 enable/disable**.

   The following screen displays:

   ```
   Network configuration

   IPv4 enable/disable:

   ENABLE
   ```

```
CANCEL    FINISH
```

2. Set the **ENABLE/DISABLE** field to the required option.
3. To accept, press the right-hand navigation button.

2.4.2.2.2. Set up IPv4 static address

To set up IPv4 static address:

1. From the front panel menu, select **System > System configuration > Network config > Set up interface #1 > Configure #1 IPv4 > Static IPv4 address**.

   The following screen displays:

```
Network configuration

Enter IPv4 address
for interface #1:
     0.  0.  0.  0
Enter netmask:
     0.  0.  0.  0
CANCEL           NEXT
```

2. Set each field of the IP address and netmask for the interface (press the **Select** button to move to the next field).
3. Once all fields have been set, press the right-hand navigation button to continue.
4. To accept the changes, press the right-hand navigation button and then **CONTINUE** to go back to the **Static IPv4 address** menu.

2.4.2.2.3. Enable/disable IPv6

To enable/disable IPv6:

1. From the front panel menu, select **System > System configuration > Network config > Set up interface #1 > Configure #1 IPv6 > Enable/Disable IPv6**.

   The following screen displays:

```
Network configuration

IPv6 enable/disable:
DISABLE

CANCEL    FINISH
```

2. Set the **ENABLE/DISABLE** field to the required option.
3. To accept, press the right-hand navigation button.

2.4.2.2.4. Set up IPv6 static address

To set up IPv6 static address:

1. From the front panel menu, select **System > System configuration > Network config > Set up interface #1 > Configure #1 IPv6 > Static addr/SLAAC > Select Static/SLAAC**.

   The following screen is displayed:

   ```
   Network configuration
   Do you want to use a
   static address or
   SLAAC?
   ```

   Select static and press the right-hand navigation button.

   Then, select **Static IPv6 address** and press the right-hand navigation button.

   The following screen displays:

   ```
   Network configuration
   Enter IPv6 address
   For interface #1:



   CANCEL    NEXT
   ```

2. Enter the required IPv6 address.
3. When the IPv6 address is correct, press the right-hand navigation button. The following screen displays:

   ```
   Network configuration
   IPv6 address
   xxxx:xxxx:xxxx:xxxx:
   xxxx:xxxx:xxxx:xxxx

   Enter prefix length:
   64

   BACK    NEXT
   ```

4. When the IPv6 address prefix details are correct, press the right-hand navigation button.

5. You are asked whether you wish to accept the new interface. To accept, press the right-hand navigation button.

Enabling static IPv6 addresses on HSM's network interface disables SLAAC on this interface. See Enable IPv6 SLAAC for SLAAC addresses.

2.4.2.2.5. Set the link speed for interface #1

To set up the link speed for interface #1:

1. From the front panel menu, select **System > System configuration > Network config > Set up interface #1 > Set link speed for #1**.
2. The following screen displays:

```
Network configuration

Select desired link
speed:
auto / 1Gb

CANCEL    NEXT
```

   You can choose from **auto / 1Gb**, **10BaseT**, **10BaseT-FDX**, **100BaseTX**, or **100BaseTX-FDX**.

   Entrust recommends that you configure your network speed for automatic negotiation, using the **auto / 1Gb** or **auto** option. You will be asked to confirm the changes if **auto / 1Gb** is not selected. Selecting **auto / 1Gb** is the only means of achieving 1Gb link speed.

3. Press the right-hand navigation button and you will be returned to the **Set up interface #1** screen and you can then continue with the configuration.

2.4.2.3. Configure Ethernet interface #2

To set up the Ethernet interface #2, if required:

1. From the front panel menu, select **System > System configuration > Network config > Set up interface #2**.
2. Enter the details for interface #2 in the same manner that you entered the details for interface #1.
3. Once the interface #2 details have been entered you need to explicitly enable interface #2. Select **System > System configuration > Network config > Set up interface #2 > Enable/Disable Int #2**.

4. The following screen displays:

```
Network configuration

Interface #2
DISABLE

CANCEL    FINISH
```

5. Select the **ENABLE** option.
6. Press the right-hand navigation button to accept. A screen similar to that used for interface #1 is displayed.

2.4.2.4. Configure an Ethernet bond interface

2.4.2.4.1. Enable or disable the use of a bond interface

1. From the front panel menu, select **System > System configuration > Network config > Set up bond > Enable/disable bond**.

   The following screen displays:

```
Network configuration

Bond Interface

DISABLE

CANCEL    FINISH
```

2. Set the **ENABLE/DISABLE** field to the required option.
3. To accept, press the right-hand navigation button.

2.4.2.4.2. Set up a bond interface

1. From the front panel menu, select **System > System configuration > Network config > Set up bond > Configure bond**.

   The following screen displays:

```
Bond interface config
will use the eth0
IPv4 and IPv6 config
if they are enabled

CANCEL    NEXT
```

2. Press the right-hand navigation button.

   The following screen displays:

   ```
   Bond interface config
   Update parameter

   mode: 802.3ad

   BACK    NEXT
   ```

3. Set the mode field to the required option, either `802.3ad` or `active-backup`.
4. To accept, press the right-hand navigation button.

   The following screen displays:

   ```
   Bond interface config
   Update parameter

   miimon: 100

   BACK    NEXT
   ```

5. Set the `miimon` field to the required value, the range is `0` - `10000` milliseconds.

   Setting the `miimon` value to `0` disables it. This can prevent the bonding resilience from functioning correctly in `active-backup` mode.

6. To accept, press the right-hand navigation button.

   The following screen displays:

   ```
   Bond interface config
   Update parameter

   lacp_rate: slow

   only valid for
   802.3ad (LACP) mode

   BACK    NEXT
   ```

7. Set the `lacp_rate` field to the required option, either `slow` or `fast`.

   This parameter is only valid for 802.3ad mode. This setting is ignored in other modes.

   slow request LACPDUs to be transmitted every 30 seconds

   fast request LACPDUs to be transmitted every 1 second

8. To accept, press the right-hand navigation button.

The following screen displays:

```
Bond interface config
Update parameter
xmit hash policy:
layer2

only valid for
802.3ad (LACP) mode

BACK    NEXT
```

9. Set the `xmit hash policy` field to the required option.

   This parameter is only valid for 802.3ad mode. This setting is ignored in other modes.

   Options:

   - layer2
   - encap2+3
   - layer2+3.

   For more information, see https://www.kernel.org/doc/Documentation/networking/bonding.txt

10. To accept, press the right-hand navigation button.

    The following screen displays:

```
Bond interface config
Update parameter

primary device: eth0

only valid for
active-backup mode

BACK    NEXT
```

11. Set the `primary device` field to the required option, either `eth0` or `eth1`.

    This parameter is only valid for `active backup` mode. This setting is ignored in other modes.

12. To accept, press the right-hand navigation button.

    The following screen displays:

```
Bond interface config
Update parameter

resend igmp: 1
```

```
only valid for
active-backup mode

BACK    NEXT
```

13. Set the `resend igmp` field to the required value. Range: `0` - `255`.

    This parameter is only valid for `active backup` mode. This setting is ignored in other modes.

14. To accept, press the right-hand navigation button.

    The following screen displays:

```
Bond interface config

Are you sure you wish
to change the config ?

CANCEL    CONFIRM
```

15. To accept and apply changes to the bond config, press the right-hand navigation button.

    The following confirmation screen displays:

```
Bond interface
config completed OK

CONFIRM
```

2.4.2.5. Default gateway

2.4.2.5.1. Set default gateway for IPv4

To set a default gateway for IPv4:

1. From the front panel menu, select **System > System configuration > Network config > Set default gateway > IPv4 Gateway**.

   The following screen is displayed:

```
Gateway configuration

Enter IPv4 address of
the default gateway:

0. 0. 0. 0
```

```
CANCEL    NEXT
```

2. Enter the IPv4 address of the default gateway.
3. Press the right-hand navigation button **NEXT** and then **FINISH** to accept.

2.4.2.5.2. Set default gateway for IPv6

To set a default gateway for IPv6:

1. From the front panel menu, select **System > System configuration > Network config > Set default gateway > IPv6 gateway**.

   The following screen is displayed:

   ```
   Gateway configuration

   Enter IPv6 address of
   the default gateway:



   CANCEL    NEXT
   ```

   Enter the address for the gateway. Press the right-hand navigation button. The following screen is displayed if the address entered was a link-local address:

   ```
   Gateway configuration

   Select an interface for link-local address:

   ::

   CANCEL    NEXT
   ```

   Select the interface for the IPv6 gateway. Press the right-hand navigation button to accept.

2.4.2.6. Set up Routing

2.4.2.6.1. Set up routing for IPv4

To set a new route entry for IPv4:

1. From the front panel menu, select **System > System configuration > Network config > Set up routing > New IPv4 route entry**.

   The following screen is displayed:

```
Edit route entry

Enter IP range
and mask length:
0.  0.  0.  0/ 0
Enter the gateway:
0.  0.  0.  0

CANCEL    FINISH
```

2. Enter the IPv4 address range details for the route. Press the right-hand navigation button to accept.

2.4.2.6.2. Set up routing for IPv6

To set a new route entry for IPv6:

1. From the front panel menu, select **System > System configuration > Network config > Set up routing > New IPv6 route entry**.

   The following screen is displayed:

```
Edit route entry

Enter the IP range
and prefix length:
::/64

CANCEL    NEXT
```

2. Enter the IPv6 address range details for the route. Press the right-hand navigation button to accept. The following screen is displayed:

```
Edit route entry
xxxx:xxxx:xxxx:xxxx:
 xxxx:xxxx:xxxx:xxxx
 /xxx

Enter the gateway:
::


BACK    NEXT
```

3. Enter the gateway address; if it is a link local address, the following screen is displayed.

```
Edit route entry

Select an interface
for link-local address:
fe80:xxxx:xxxx:xxxx:
xxxx:xxxx:xxxx:xxxx
  Interface #1
```

```
BACK    NEXT
```

4. Select the interface for the IPv6 gateway and press the right-hand navigation button to accept.
5. If the new route entry entered for IPv6 is incorrect an error message is displayed on the next screen, select **BACK** to go to the route entry screen. The new IPv6 route entry will need to be entered again.

2.4.2.7. Edit route entry

2.4.2.7.1. Edit IPv4 route entry

To edit a route entry for IPv4:

1. From the front panel menu, select **System > System configuration > Network config > Set up routing > Edit route entry**.

   The following screen is displayed:

```
►  1. 1. 1. 1/ 1
3. 3. 3. 3/ 3
1111:1111:1111:1111:
1111:1111:1111:1111
 /128

BACK    SELECT
```

2. Select the IPv4 route to be edited. Press the right-hand navigation button. The following screen is displayed:

```
Edit route entry

Enter the IP range
and mask length:
1. 1. 1. 1/ 1
Enter the gateway
2. 2. 2. 2
CANCEL    FINISH
```

3. Edit the IPv4 route entry. Press the right-hand navigation button to accept the changes.

2.4.2.7.2. Edit IPv6 route entry

To edit a route entry for IPv6:

1. From the front panel menu, select **System > System configuration > Network config > Set up routing > Edit route entry**.

   The following screen is displayed:

```
Edit route entry
►  1. 1. 1. 1/ 1
3. 3. 3. 3/ 3
1111:1111:1111:1111:
1111:1111:1111:1111
/128


BACK    SELECT
```

2. Select the IPv6 route to be edited. Press the right-hand navigation button. The following screen is displayed:

```
Edit route entry

Enter the IP range
and prefix length:
1111:1111::1111:1111:
 1111:1111:1111:1111/128

CANCEL    NEXT
```

3. Edit the IPv6 route entry. Press the right-hand navigation button.

```
Edit route entry
1111:1111:1111:1111:
 1111:1111:1111:1111/128

Enter the gateway
2222:2222:2222:2222

BACK  NEXT
```

4. Enter the IPv6 route gateway. If a link-local address is entered for the IPv6 route gateway the screen below will be displayed.

```
Edit route entry

Select an interface
for link-local address:
fe80:2222:2222:2222:
2222:2222:2222:2222
Interface #1
BACK    NEXT
```

5. Select the interface for the IPv6 gateway. Press the right-hand navigation button to accept.

2.4.2.8. Remove route entry

To remove a route entry:

1. From the front panel menu, select **System > System configuration > Network config > Set up routing > Remove route entry**.

   The following screen is displayed:

   ```
   ► 1. 1. 1. 1/ 1
   3. 3. 3. 3/ 3
   1111:1111:1111:1111:
   1111:1111:1111:1111
   /128


   BACK    SELECT
   ```

2. Select the IPv4/IPv6 route to be removed. Press the right-hand navigation button.
3. The selected route will be displayed. Press the right-hand navigation button to remove the route.

2.4.2.9. Enable IPv6 SLAAC

SLAAC can be enabled/disabled independently on each of the two interfaces.

To enable SLAAC:

1. From the front panel menu, select **System > System configuration > Network config > Set up interface #1 > Configure #1 IPv6 > Static addr/SLAAC > Select Static/SLAAC**.

   The following screen is displayed:

   ```
   Network configuration
   Do you want to use a
   static address or
   SLAAC?
   ```

2. Select **SLAAC** and press the right-hand navigation button.
3. The **IPv6 address config selected** screen is displayed. Press the right-hand navigation button to accept.
4. Select the required state and press the right-hand navigation button.
5. The **SLAAC configuration completed OK** screen is displayed. Press the right-hand navigation button to accept.

   Enabling SLAAC on the HSM's network interface disables the use of

> static IPv6 addresses on this interface.

2.4.2.10. Configuring the Remote File System (RFS)

The RFS contains the master copy of the Security World data for backup purposes. The RFS can be a standalone machine, and can also dual role as a client. If the RFS duals as a client, a common file structure serves both the RFS and the configuration files for the client.

The HSM must be able to connect to TCP port 9004 of the RFS. If necessary, modify the firewall configuration to allow this connection on either the RFS itself, or on a router between the RFS and the HSM, or both.

Obtain the following information about the HSM before you set up an RFS for the first time:

- The IP address.

  The following information can be obtained automatically (or manually):

- The electronic serial number (ESN).
- The hash of the K_{NETI} key (HK_{NETI}). The K_{NETI} key authenticates the HSM to clients. It is generated when the HSM is first initialized from factory state.

If your network is secure and you know the IP address of the HSM, you can use the anonkneti utility to obtain the ESN and hash of the K_{NETI} key.

Alternatively, you can find this information on the HSM startup screen. Use the touch wheel to scroll to the appropriate information.

When you have the necessary information, set up an RFS and HSM in the following order:

1. Prepare the RFS by running the following command on that computer:

   ```
   rfs-setup <Unit IP> A285-4F5A-7500 2418ec85c86027eb2d5959fef35edc5e1b3b698f
   ```

   In this command:

   - *<Unit IP>* is the IP address of the HSM.
   - `A285-4F5A-7500` is the ESN of the HSM.
   - `keyhash` is the hash of the K_{NETI} key.
2. On the HSM display screen, use the right-hand navigation button to select **System > System configuration > Remote file system**, and enter the IP address of the client computer on which you set up the RFS:

   ```
   Remote File System
   ```

```
Enter IP address:



CANCEL         CONTINUE
```

3. The next screen asks for the port number on which the RFS is listening. Enter the port number and press the right-hand navigation button to continue:

```
  Remote File System

Enter port number:
    9004

CANCEL         CONTINUE
```

Leave the port number at the default setting of 9004.

4. Select the config push mode and press the right-hand navigation button to continue:

```
  Remote File System

Set RFS config push
mode to:

        AUTO

CANCEL         CONTINUE
```

Three options are available:

- AUTO: The RFS is only allowed to push configuration files to the HSM if secure authentication is enabled. This is the default value.
- ON: The RFS is allowed to push configuration files to the HSM.
- OFF: The RFS is not allowed to push configuration files to the HSM.

5. You must then choose whether to enable or disable secure authentication when setting up the RFS. The following screen is displayed:

```
  Remote File System

Do you want secure
authentication enabled
on the RFS?

          YES
CANCEL         CONTINUE
```

a. Select No and press the right-hand navigation button to configure the RFS without secure authentication. The authentication of the RFS will be based on the IP

address only.

b. Select `Yes` and press the right-hand navigation button to configure the RFS with secure authentication.

6. Skip this step if you have not selected secure authentication.

   If an nToken is installed in the RFS, you will be asked to choose which authentication key to use. Select the desired option and press the right-hand navigation button:

   ```
   >0DA8-A5AE-BA0D
    Software Key


   BACK       SELECT
   ```

   a. The ESN of the nToken installed in the RFS.

   b. "Software Key" for software-based authentication.

   If no nToken is installed in the RFS, then software-based authentication is automatically selected.

7. Skip this step if you have not selected secure authentication.

   The next screen will ask you to verify that the key hash displayed by the HSM matches the RFS key hash:

   ```
   Remote 0DA8-A5AE-BA0D
   reported the key hash:
    9e0020264af732933574
    0cfe10446d33cea33f4a
   Is this EXACTLY right?

   CANCEL         CONFIRM
   ```

   The RFS key hash is obtained by running the commands described below. Take a copy of the returned key hash and compare it to the value reported on the HSM display.

   With software-based authentication

   Run the following command on the RFS:

   ```
   enquiry -m0
   ```

   This command returns the software key hash, tagged as `kneti hash`, as part of its output, for example:

   ```
   Server:
     enquiry reply flags   none
     enquiry reply level   Six
     ...
   ```

```
  kneti hash           d4c3d757a67416cb9ba31f33febd6ead688629e5
  ...
```

With nToken authentication

Run the following command on the RFS:

```
ntokenenroll -H
```

This command produces output of the form:

```
nToken module #1
nToken ESN:      0DA8-A5AE-BA0D
nToken key hash: 9e0020264af732933574
                 0cfe10446d33cea33f4a
```

Check that the ESN also matches the one reported on the HSM display.

If the RFS key hash matches the one reported on the HSM display, press the right-hand navigation button to continue the RFS configuration. Otherwise press the left-hand navigation button to cancel the operation.

8. The HSM displays "Transferring files..." followed by a message reporting that the RFS has been set up. Press the right-hand navigation button again to exit.

After you have defined the RFS, the HSM configuration files are exported automatically.

To modify the RFS at a later date, select **System > System configuration > Remote file system**, and then select the required action.

2.4.2.10.1. Systems configured for Remote Administration

Before using Remote Administration or configuring NTP, enable *config push* on the HSM for the RFS or client computer you intend to use for configuration. The RFS config push is preferred unless the config push client is not actually the same machine as the RFS. The RFS config push is recommended at least when securely bootstrapping the configuration of the system from the HSM front panel.

2.4.2.11. Enabling config push from the RFS

On the HSM display, use the right-hand navigation button to select **System > System configuration > Remote File System**, and follow the steps described in Configuring the Remote File System (RFS). To enable config push from the RFS, set the push mode to **AUTO** with RFS secure authentication enabled (recommended), or to **ON**.

The RFS config push supports specifying secure authentication from the HSM front panel, whereas the client config push only supports specifying authentication either from the HSM Serial Console `push` command, or from the config file itself.

2.4.2.12. Enabling config push from a client computer

To enable config push from a client computer, on the HSM display, use the right-hand navigation button to select **System > System configuration > Config file options > Client config push > Config push mode**, set **ON** or **OFF**, then select **CONFIRM**. A confirmation message will be displayed.

After enabling config push, specify the IP address of the client to push the configuration from. On the HSM display, use the right-hand navigation button to select **System > System configuration > Config file options > Client config push > Client address**. Enter the IP address and select **CONFIRM**. A message is displayed confirming your chosen IP address. Select **CONTINUE**.

Any remote computer is allowed to push configuration files if no IP address or the 0.0.0.0 address is specified.

After enabling config push, complete the configuration steps by editing the configuration files, rather than by using HSM front panel. See HSM and client configuration files for more information.

2.4.3. Basic configuration of the client to use the HSM

2.4.3.1. Client configuration utilities

Entrust provides the following utilities for client configuration:

Utility	Description
`nethsmenroll`	Used to configure the client to communicate with the HSM.
`config-serverstartup`	Used to configure the hardserver of the client to enable TCP sockets.

2.4.3.1.1. nethsmenroll

The `nethsmenroll` command-line utility edits the client hardserver's configuration file to add the specified HSM. If the HSM's `ESN` and `HKNETI` are not specified, `nethsmenroll` attempts to

contact the HSM to determine what they are, and requests confirmation.

Usage:

```
nethsmenroll [Options] --privileged <hsm-ip> <hsm-esn> <hsm-kneti-hash>
```

Options:

`-m|--module=MODULE`	Specifies the local module number that should be used (default is `0` for dynamic configuration by hardserver).
`-p|--privileged`	Makes the hardserver request a privileged connection to the HSM (default `unprivileged`).
`-<hsm-ip>`	The IP address of the HSM, which could be one of the following: • an IPv4 address • an IPv6 address, including a link-local IPv6 address • a hostname
`-r|--remove`	Removes the configuration of the specified HSM.
`-f|--force`	Forces reconfiguration of an HSM already known.
`--no-hkneti-confirmation`	Does not request confirmation when automatically determining the HSM's `ESN` and `HKNETI`. This option is potentially insecure and should only be used on secure networks where there is no possibility of a man-in-the-middle attack. For guidance on network security, see the *nShield Security Manual*.
`-V|--verify-nethsm-details`	When the `ESN` and `HKNETI` have been provided on the command line, verifies that the selected HSM is online, reachable and matches those details.
`-P|--port=PORT`	Specifies the port to use when connecting to the given HSM (default `9004`).
`-n|--ntoken-esn=ESN`	Specifies the `ESN` of the nToken to be used to authenticate this client. If the option is omitted, then software authentication will be used instead.

2.4.3.1.2. config-serverstartup

The `config-serverstartup` command-line utility automatically edits the `[server_startup]` section in the local hardserver configuration file in order to enable TCP ports for Java and KeySafe. Any fields for which values are not specified remain unchanged. After making any

changes you are prompted to restart the hardserver.

Run `config-serverstartup` using the following commands:

```
config-serverstartup [OPTIONS]
```

For more information about the options available to use with `config-serverstartup`, run the command:

```
config-serverstartup --help
```

2.4.3.2. Configuring a client to communicate through an nToken

You can configure a client to use its nToken to communicate with an HSM, if it has one installed. When this happens, the HSM:

- Examines the IP address of the client.
- Requires the client to identify itself using a signing key.

If an nToken is installed in a client, it can be used to both generate and protect a key that is then used for the impath communication between the HSM and the client. A strongly protected key is used at both ends of the impath as a result.

2.4.3.3. Enrolling the client from the command line

Complete the following steps to initially configure a client computer to communicate with and use an HSM. See Basic HSM, RFS and client configuration for more about the available options.

Do the following:

1. On the client, open a command line window, and run the command:

   ```
   nethsmenroll --help
   ```

2. Retrieve the `ESN` and `HKNETI` of the HSM:

   ```
   anonkneti <ip-address>
   ```

   If the `ESN` and `HKNETI` are not specified, `nethsmenroll` attempts to contact the HSM to determine what they are, and requests confirmation.

3. Do one of the following:

   If you are enrolling a client *with* an nToken installed, run the command:

   ```
   nethsmenroll --ntoken-esn <nToken ESN> [Options] --privileged <Unit IP> <Unit ESN> <Unit KNETI HASH>
   ```

   If you are enrolling a client *without* an nToken installed, run the command:

   ```
   nethsmenroll [Options] --privileged < Unit IP> < Unit ESN> < Unit KNETI HASH>
   ```

   The following is an example of the output:

   ```
   OK configuring hardserver's nethsm imports.
   ```

2.4.3.4. Configure the TCP sockets on the client for Java applications

To configure the TCP sockets on the client for Java applications (for example, KeySafe):

1. Run the command:

   ```
   config-serverstartup --enable-tcp --enable-privileged-tcp
   ```

2.4.4. Basic configuration of an HSM to use a client

Do the following:

1. On the HSM front panel, use the right-hand navigation button to select **System > System configuration > Client config > New client**.

   The following screen is displayed:

   ```
   Client configuration

   Please enter your
   client IP address:



   CANCEL          NEXT
   ```

   Enter the IP address of the client, and press the right-hand navigation button.

2. Use the touch wheel to confirm whether you want to save the IP or not, and press the right-hand navigation button.

```
Client configuration

Do you want to save
the IP in the config?
(No for dynamic client
IPs)
          No
BACK            NEXT
```

3. Use the touch wheel to select the connection type between the HSM and the client.

```
Client configuration

Please choose the
client permissions

    Unprivileged

BACK            NEXT
```

The following options are available:

Option	Description
Unprivileged	Privileged connections are never allowed.
Priv. on low ports	Privileged connections are allowed only from ports numbered less than 1024. These ports are reserved for use by root on Linux.
Priv. on any ports	Privileged connections are allowed on all ports.

A privileged connection is required to administer the HSM, for example to initialize a Security World. If privileged connections are allowed, the client can issue commands (such as clearing the HSM) which interfere with the normal operation of the HSM. Entrust recommends that you allow only unprivileged connections unless you are performing administrative tasks.

4. When you have selected a connection option, press the right-hand navigation button.

The following screen is displayed:

```
Client configuration

Do you want secure
authentication enabled
on this client?

         Yes
BACK             NEXT
```

a. Select No and press the right-hand navigation button to configure the client without secure authentication. The authentication of the client will be based on the IP address only.

b. Select Yes and press the right-hand navigation button to configure the client with secure authentication.

5. On the HSM, enter the number of the port on which the client is listening (the default is 9004), and press the right-hand navigation button. The following screen is displayed:

```
Client configuration

On what port is the
client listening?

        9004

CANCEL           NEXT
```

6. Skip this step if you have not selected secure authentication.

   If an nToken is installed in the client, you will be asked to choose which authentication key to use. Select the desired option and press the right-hand navigation button:

```
>3138-147F-2D64
 Software Key


BACK       SELECT
```

   a. The ESN of the nToken installed in the client.

   b. "Software Key" for software-based authentication.

   If no nToken is installed in the client, then software-based authentication is automatically selected.

   Software-based authentication is only supported from version 12.60.

7. Skip this step if you have not selected secure authentication.

   The next screen will ask you to verify that the key hash displayed by the HSM matches the client key hash:

```
Remote 3138-147F-2D64
reported the key hash:
 691be427bb125f387686
 38a18bfd2eab75623320
Is this EXACTLY right?

CANCEL         CONFIRM
```

The client key hash is obtained by running the commands described below. Take a copy of the returned key hash and compare it to the value reported on the HSM display.

With software-based authentication

Run the following command on the client:

```
enquiry -m0
```

This command returns the software key hash, tagged as `kneti hash`, as part of its output, for example:

```
Server:
  enquiry reply flags   none
  enquiry reply level   Six
  ...
  kneti hash            f8222fc007be38b78ebf442697e244dabded38a8
  ...
```

With nToken authentication

Run the following command on the client:

```
ntokenenroll -H
```

This command produces output of the form:

```
nToken module #1
nToken ESN:      3138-147F-2D64
nToken key hash: 691be427bb125f387686
                 38a18bfd2eab75623320
```

Check that the ESN also matches the one reported on the HSM display.

If the client key hash matches the one reported on the HSM display, press the right-hand navigation button to continue the RFS configuration. Otherwise press the left-hand navigation button to cancel the operation.

8. The HSM displays a message reporting that the client has been configured. Press the right-hand navigation button again.

2.4.5. Restarting the hardserver

In order to establish any configuration changes you may have entered, you must restart the hardserver (also called the nfast server).

1. Do one of the following to stop and restart the hardserver, according to your operating system:
   a. **Windows**:

```
net stop "nfast server"
net start "nfast server"
```

   b. **Linux**:

```
/opt/nfast/sbin/init.d-ncipher restart
```

2.4.6. Zero touch configuration of an HSM

On a serial-enabled HSM (see Model numbers) you can configure the HSM and set up the RFS by using the HSM Serial Console rather than the front panel. See Configuring an nShield HSM using the Serial Console for your HSM for more information on the Serial Console.

Once the HSM's power, Ethernet and serial cables have been connected, to allow zero touch configuration of the HSM (no further use of the front panel required), follow these steps:

2.4.6.1. Configuring the network interfaces via the Serial Console

1. Log in to the HSM Serial Console.
2. Configure networking on Ethernet Interface #1:
   a. Set the IP address and netmask of the interface:

```
(cli) netcfg iface=0 addr=0.0.0.0 netmask=0.0.0.0
```

   b. Set the IP address of the gateway for the HSM:

```
(cli) gateway 0.0.0.0
```

   If your network environment requires you to configure static routes you may also use the HSM Serial Console to configure static routes for the HSM at this stage.

2.4.6.2. Allowing configuration files to be pushed to the HSM via the Serial Console

To allow the Remote File System (RFS) to push configuration files to the HSM, configure

the RFS using the `rfsaddr` command. To allow other remote computers to push configuration files to the HSM, use the `push` command.

2.4.6.2.1. Configuring the Remote File System (RFS) via the Serial Console

1. Log in to the HSM Serial Console, and run the following commands to obtain the HSM ESN and KNETI hash, for example:

   ```
   (cli) esn
   ESN: 6B1D-03CE-2F9A
   (cli) kneti
   Kneti hash: 56304e3f752cd13d219fa47ad27d56bb6a6642aa
   ```

2. Run the `rfs-setup` command on the RFS with the IP address of the HSM and the values previously returned by the `esn` and `kneti` commands:

   ```
   rfs-setup <Unit IP address> <ESN> <KNETI hash>
   ```

   For information on running `rfs-setup`, see Configuring the Remote File System (RFS).

3. In the HSM Serial Console, configure the RFS using the `rfsaddr` command.

   ```
   (cli) rfsaddr address[:port] [keyhash [esn]] [push]
   ```

   In this command:

   - `address` is the RFS IP address.
   - `port` is the RFS port number (default is 9004).
   - `keyhash` is the RFS KNETI hash (default is 40 zeroes).
   - `esn` is the RFS nToken ESN (default is "", that is, no ESN).
   - `push` specifies if the RFS can push configuration files to the HSM:
     - `ON`: The RFS is allowed to push configuration files.
     - `OFF`: The RFS is not allowed to push configuration files.
     - `AUTO`: The RFS is allowed to push configuration files if RFS secure authentication is enabled. This is the default option.

   The `keyhash` and `esn` are optional, and can be used to enable the RFS secure authentication:

   a. No RFS secure authentication (not recommended): The `keyhash` and `esn` parameters are not specified.
   b. RFS software-based authentication: Only the `keyhash` parameter is specified. The

RFS software KNETI hash is obtained by running the `enquiry -m0` command on the RFS. The value is tagged as `kneti hash` in the command output.

c. RFS nToken authentication: The `keyhash` and `esn` parameters are specified. The RFS nToken KNETI hash and ESN are obtained by running the `ntokenenroll -H` command on the RFS.

2.4.6.2.2. Allowing configuration files to be pushed to the HSM from a remote computer via the Serial Console

In addition to the RFS, the `push` serial command can be used to allow a remote computer to push configuration files.

```
(cli) push ON [address] [keyhash]
```

In this command:

- `address` is the remote computer IP address. It defaults to 0.0.0.0 which allows any address to push. It is not recommended to leave the IP address unrestricted, unless `keyhash` is specified for authentication.
- `keyhash` is the hash of the key with which the authorized client is to authenticate itself (defaults to no key authentication required).

Enabling the push feature allows remote computers to change the HSM configuration file and make configuration changes that are normally only available through the HSM secure user interface.

After you enable the HSM for zero touch configuration, everything that can be configured using the front panel can be configured remotely using one of the following methods:

- The HSM Serial Console.
- The cfg-pushnethsm utility to push an updated configuration file to the HSM. From the configuration file you can configure the RFS, add clients, or change the network configuration.
- The nethsmadmin utility.

2.4.7. Checking the installation

To check that the module is installed and configured correctly on the client:

1. Log in as a user and open a command window.
2. Run the command:

```
enquiry
```

For an example of the output following a successful `enquiry` command. See Enquiry utility.

If you are configuring a client belonging to an HSM, the response to the `enquiry` command should be populated and the `hardware status` shown as `OK`.

If the `mode` is `operational` the HSM has been installed correctly.

If the `mode` is `initialization`, the HSM 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.

2.4.8. Using a Security World

You can update the Security World on the host using:

- The nShield HSM front panel controls
- The command-line utilities
- The Cryptographic Service Provider wizard
- KeySafe.

You can also use these tools to create keys or cards. If you perform such tasks on a client other than the computer on which the RFS is installed, you must transfer the updated files to the RFS before they are available to the HSM.

For more about using Security Worlds, refer to nShield Security World v13.6.5 Management Guide.

2.5. Client software and module configuration: network-attached HSMs

This chapter describes how to configure the internal security module of the nShield HSM and the client to communicate with each other, after you have installed the HSM and the Security World Software.

For more information about installing the HSM, see nShield v13.6.5 Hardware Install and Setup Guides.

The nShield HSM provides significant performance improvements, and can be deployed successfully with existing nShield products. Customers wishing to take advantage of these performance improvements **must** update their client machines with the latest Security World Software.

Linux

There are three levels of user:

- Superuser
- nfast group user
- normal

Typically, normal users can carry out operations involving Security Worlds, cardsets and keys, but not create Security Worlds, keys and cardsets. nfast group users have enhanced access, enabling them to create Security Worlds, cardsets and keys. For example, encrypted copies of keys are held in `kmdata` (`/opt/nfast/kmdata`). Normal users only have read access to the files, whereas nfast group users have read and write access, enabling them to create and use keys. nfast group users can also change the mode of an HSM remotely.

Superuser access (for example, `root`) is required for such tasks as software installation, starting and stopping the hardserver and SNMP.

Windows

There are two levels of user in Windows:

- Administrator access
- Normal

Administrator access (an Administrator on Windows) is required for such tasks as:

- Software installation, starting and stopping the hardserver and SNMP.
- Typically, any operation that requires write access, such as the creation of Security Worlds, card sets and keys.

Typically, normal users can only carry out read only operations involving Security Worlds, card sets and keys. For example, encrypted copies of keys are held in **Key Management Data** (`C:\ProgramData\nCipher\Key Management Data`). The default permissions allow all users to read these files, enabling them to use keys but not create them. File permissions can be altered to restrict access to specific keys to specific users/groups.

You should use the `nt_privpipe_users` option to define the name of

> the user who can carry out privileged operations, for example, using the `nopclearfail` utility. See *nt_privpipe_users* for more information.

2.5.1. About user privileges

Cryptographic security does not depend on controlling user privileges or access but maintaining the integrity of your system from both deliberate or accidental acts can be enhanced by appropriate use of (OS) user privileges.

2.5.2. About client configuration

You can add more HSMs to a client and more clients to an HSM at any time.

The HSM and a client communicate by means of the hardserver, which handles secure transactions between the HSM and applications that run on the client. You must configure:

- Each client hardserver to communicate with the HSM that the client needs to use
- The HSM to communicate with clients that are allowed to use the HSM.

Information about the current configuration of the HSM or a client is stored in configuration files that are stored in specified file systems on the clients and on the HSM. For more information about the contents of configuration files, see HSM and client configuration files.

For information about configuring the HSM by importing an edited configuration file, see About user privileges.

2.5.2.1. Remote file system (RFS)

Each HSM must have a remote file system (RFS) configured. The RFS contains master copies of all the files that the HSM needs:

- The HSM configuration file
- Feature-enabling certificates
- The encrypted Security World and key data for Security Worlds created on the HSM.

The RFS normally resides on a client computer, but it can be located on any computer that is accessible on the network.

For more information about setting up the remote file system, see Configuring the Remote

File System (RFS).

2.5.2.2. HSM configuration

The data that defines the configuration of the HSM hardserver is stored in a file on the HSM. This file is automatically:

- Updated when the HSM is configured from the front panel
- Exported to the remote file system (RFS) directory.

Each HSM has separate configuration files on the RFS, stored in the directories with names of the form `/opt/nfast/kmdata/hsm-ESN/config` (**Linux**) or `%NFAST_KMDATA%\hsm-ESN\config` (**Windows**) where `ESN` represents the electronic serial number of the HSM from which the files were exported. These directories can contain the following files:

Option	Description
`config`	The master configuration file. This contains the current configuration for the HSM. It is always present in the directory.
`config-name`	An alternative configuration file saved by the system.
`config.new`	A hand-edited configuration file that can be read by the HSM.

You normally configure the HSM using the front panel controls. However, in some cases (for example, if you need to configure an HSM remotely, or if you are importing a number of clients), you may prefer to edit the exported configuration file and then re-import the file into the HSM. For more information, see:

- About user privileges
- HSM and client configuration files.

2.5.2.3. Client configuration

The data that defines the configuration of the client hardserver is stored in a file on the client's file system.

You must load the configuration file for the configuration to take effect. For information about loading a client configuration remotely, see Remote configuration of additional clients.

You can configure a client to use multiple HSMs. All the HSMs configured for use by a client can fail over if the application that uses them is set up appropriately.

For more information about the contents of the client configuration file, see HSM and client configuration files.

You can also configure the client's hardserver by setting environment variables, as described in Setting environmental variables. Environment variable settings override settings in the client configuration files.

2.5.3. Basic HSM and remote file system (RFS) configuration

After installing the HSM hardware and software, there are several HSM and RFS configuration tasks you must perform. You perform these RFS tasks before:

- Creating the Security World and an Operator Card Set (OCS)
- Completing the process of configuring the HSM and client to work together.

2.5.3.1. Configuring the Ethernet interfaces

The HSM communicates with one or more clients. Each client is an Ethernet connected computer that has the Security World Software installed and configured. You must supply Internet Protocol (IP) addresses for the HSM and the client. Contact your system administrator for this information if necessary.

To configure the Ethernet interfaces (IPv4 and IPv6), see Basic HSM, RFS and client configuration.

2.5.3.2. Optionally configure hardserver interfaces

By default, the hardserver listens on all interfaces. However, you can alter the hardserver settings. Altering the hardserver settings would prove necessary, for example, if you wanted to connect one of the Ethernet interfaces to external hosts.

Ensure that you have configured the Ethernet interfaces on the HSM before attempting to configure the hardserver.

You can configure the following options to specify network interfaces on which the hardserver listens:

Option	Description
All interfaces	This option (which is the default) specifies that the hardserver listens on all interfaces.

Option	Description
IP address of interface #1	This option specifies that the hardserver listens only on interface 1. This option only appears if interface 1 has been configured.
IP address of interface #2	This option specifies that the hardserver listens only on interface 2. This option only appears if interface 2 has been configured.
Will not listen	This option specifies that the hardserver does not listen on any interfaces.

To define the interface and port on which the hardserver listens:

1. From the main menu, select **System > System configuration > Hardserver config**. The following screen appears:

```
Hardserver config
  Select network I/F
  hardserver listens on:
  All interfaces
  Select TCP port: 9004
  CANCEL           FINISH
```

2. Select the network interfaces on which the hardserver is to listen.

For security reasons, do not allow the hardserver to listen on any interface that is to connect to the public Internet.

3. Press the Select button to move to the TCP port field, and set the port on which the hardserver is to listen. The default is 9004.

Make sure that your firewall settings are consistent with your port settings. See Before you install the software for more about firewall settings.

4. When the network interface and port are correct, press the right-hand navigation button.
5. Press the right-hand navigation button again to continue.
6. You are asked if you wish to reboot the system now or later. Press the right-hand navigation button to reboot now.

2.5.3.3. Configuring the Remote File System (RFS)

The RFS contains the master copy of the Security World data for backup purposes. The RFS

can be located on either a client or another network-accessible computer where the Security World Software is installed. If the RFS is on a client, the same file structure also contains the configuration files for that client.

We recommend that you regularly back up the entire contents of the RFS. The `kmdata` (**Linux**) or `%NFAST_KMDATA%` (**Windows**) directory is required to restore the nShield HSM, or a replacement, to its current state, in case of failure.

You can specify a new remote file system, and modify or delete an existing remote file system configuration. To create or modify a remote file system configuration, specify the IP address of the computer on which the file system resides.

You must have created an RFS on the client computer before you specify the IP address of the client.

For more information about the RFS and its contents, see:

- Remote file system (RFS)
- Security World Files

The nShield HSM must be able to connect to TCP port 9004 of the RFS. If necessary, modify the firewall configuration to allow this connection on either the RFS itself or on a router between the RFS and the nShield HSM.

You can configure the connection to use secure authentication using software-based authentication or with an nToken (or local HSM) installed in the RFS. When enabled the nShield HSM not only examines the RFS's IP address, but also requires the RFS to identify itself using a signing key.

If an nToken is installed in the RFS, it can be used to both generate and protect a key that is used for the impath communication between the nShield HSM and the RFS. Thus a strongly protected key is used at both ends of the impath. A local nShield PCIe or USB-attached HSM can also be used to perform the role of the nToken.

Obtain the following information about the nShield HSM before you set up an RFS for the first time:

- The IP address
- The electronic serial number (ESN)
- The hash of the K_{NETI} key (HK_{NETI}). The K_{NETI} key authenticates the nShield HSM to clients. It is generated when the nShield HSM is first initialized from factory state.

If your network is secure and you know the IP address of the HSM, you can use the anonkneti utility to obtain the ESN and hash of the K_{NETI} key.

Alternatively, you can find this information on the nShield HSM startup screen. Use the touch wheel to scroll to the appropriate information.

When you have the necessary information, set up an RFS as follows:

1. Prepare the RFS on the client computer (or another appropriate computer) by running the following command on that computer:

   ```
   rfs-setup <Unit IP> <EEEE-SSSS-NNNN> <keyhash>
   ```

   In this command:

   - *<Unit IP>* is the IP address of the nShield HSM.
   - *<EEEE-SSSS-NNNN>* is the ESN of the nShield HSM.
   - *<keyhash>* is the hash of the K_{NETI} key.
2. On the nShield HSM display screen, use the right-hand navigation button to select **System > System configuration > Remote file system**, and enter the IP address of the client computer on which you set up the RFS:

   ```
     Remote File System

   Enter IP address:



   CANCEL         CONTINUE
   ```

3. The next screen asks for the port number on which the RFS is listening. Enter the port number and press the right-hand navigation button to continue:

   ```
     Remote File System

   Enter port number:
       9004

   CANCEL         CONTINUE
   ```

   Leave the port number at the default setting of 9004.

4. Select the config push mode and press the right-hand navigation button to continue:

   ```
     Remote File System

   Set RFS config push
   mode to:
   ```

```
       AUTO

CANCEL         CONTINUE
```

Three options are available:

- AUTO: The RFS is only allowed to push configuration files to the nShield HSM if secure authentication is enabled. This is the default value.
- ON: The RFS is allowed to push configuration files to the nShield HSM.
- OFF: The RFS is not allowed to push configuration files to the nShield HSM.

5. You must confirm whether to enable or disable secure authentication when setting up the RFS. The following screen is displayed:

```
  Remote File System

Do you want secure
authentication enabled
on the RFS?

          YES
CANCEL         CONTINUE
```

   a. Select No and press the right-hand navigation button to configure the RFS without secure authentication. The authentication of the RFS will be based on the IP address only.
   b. Select Yes and press the right-hand navigation button to configure the RFS with secure authentication.

6. Skip this step if you have not selected secure authentication.

   If an nToken is installed in the RFS, you will be asked to choose which authentication key to use. Select the desired option and press the right-hand navigation button:

```
>0DA8-A5AE-BA0D
 Software Key


BACK       SELECT
```

   a. The ESN of the nToken installed in the RFS.
   b. "Software Key" for software-based authentication.

   If no nToken is installed in the RFS, then software-based authentication is automatically selected.

7. Skip this step if you have not selected secure authentication.

At the next screen, verify that the key hash displayed by the nShield HSM matches the RFS key hash:

```
Remote 0DA8-A5AE-BA0D
reported the key hash:
 9e0020264af732933574
 0cfe10446d33cea33f4a
Is this EXACTLY right?

CANCEL         CONFIRM
```

The RFS key hash is obtained by running the commands described below. Take a copy of the returned key hash and compare it to the value reported on the nShield HSM display.

With software-based authentication

Run the following command on the RFS:

```
enquiry -m0
```

This command returns the software key hash, tagged as `kneti hash`, as part of its output, for example:

```
Server:
  enquiry reply flags   none
  enquiry reply level   Six
  ...
  kneti hash            d4c3d757a67416cb9ba31f33febd6ead688629e5
  ...
```

With nToken authentication

Run the following command on the RFS:

```
ntokenenroll -H
```

This command produces output of the form:

```
nToken module #1
nToken ESN:      0DA8-A5AE-BA0D
nToken key hash: 9e0020264af732933574
                 0cfe10446d33cea33f4a
```

Check that the ESN also matches the one reported on the nShield HSM display.

If the RFS key hash matches the one reported on the nShield HSM display, press the right-hand navigation button to continue the RFS configuration. Otherwise press the

left-hand navigation button to cancel the operation.

8. The nShield HSM displays "Transferring files..." followed by a message reporting that the RFS has been set up. Press the right-hand navigation button again to exit.

After you have defined the RFS, the nShield HSM configuration files are exported automatically. See HSM and client configuration files for more about configuration files.

To modify the RFS at a later date, select **System > System configuration > Remote file system**, and then select the required action.

You can allow other clients to access the remote file system and share Security World and key data that is stored in the `kmdata` (**Linux**) or `%NFAST_KMDATA%` (**Windows**) directory in the same way as the HSM. Clients that access data in this way are described as *cooperating clients*. To configure client cooperation, you need to know the details of each client including IP address and optionally their key hash and nToken ESN.

2.5.3.4. Configuring log file storage

You can choose to store log files on both the HSM and RFS or on the HSM only.

To configure log file storage, use the right-hand navigation button to select **System > System configuration > Log config**. Then select one of:

1. **Log** to store log files on the HSM only.
2. **Append** to store log files on both the HSM and remote file system.

We recommend selecting **Append** because if you select **Log** you can only view the log file from the nShield HSM front panel. Moreover, the log file stored on the HSM is cleared every time it is powered down.

You may also additionally configure the logs to be sent to a remote syslog server, see Configuring Remote Syslog.

2.5.3.5. Setting the time and date

If you do not intend to use NTP time synchronization, set the time and date as described in this section. If you configure the nShield HSM to use NTP time synchronization, then the time and date will be maintained by NTP.

To set the time and date on the HSM as UTC:

1. Use the right-hand navigation button to select and display the **System** menu:

```
1-1
System configuration
System information
Login settings
Upgrade system
Factory state
Shutdown/Reboot
BACK                SELECT
```

2. Select **System configuration** to display the **System configuration** menu:

```
1-1-1
Network config
Hardserver config
Remote file system
Client config
Resilience config
Config file options
BACK                SELECT
```

3. Use the touch wheel to move the arrow to **Date/time setting**, and press the right-hand navigation button to select it. The **Set system date** screen is displayed:

```
Set system date
Please enter the
current UTC date as
DD/MM/YYYY:
 27/ 5/2013
CANCEL              NEXT
```

4. For each date field, use the touch wheel to set the value and move the cursor to the next field.

   When you have completed all the fields, press the right-hand navigation button to confirm the date. The **Set system time** screen is displayed:

```
Set system time
Please enter the
current UTC time as
hour/mins/seconds:
 18:08:19
CANCEL              FINISH
```

Setting the time and date of the HSM as UTC does not reset the value of the Real Time Clock (RTC) on the HSM. The UTC date and time settings are used only in log messages.

2.5.3.6. Keyboard layout

You can connect a keyboard to the USB connector on the nShield HSM front panel. This enables you to control the nShield HSM using a special set of keystrokes instead of the standard front panel controls.

You can connect either a US or a UK keyboard. To configure the nShield HSM for your keyboard type, select **System > System configuration > Keyboard layout** and then choose the keyboard type you require.

2.5.4. Configuring the nShield HSM to use the client

You must inform the HSM hardserver of the location of the client computer.

You can configure the connection to use secure authentication using software-based authentication or with an nToken (or local PCIe HSM) installed in the client. When enabled the nShield HSM not only examines the client IP address, but also requires the client to identify itself using a signing key.

If an nToken is installed in a client, it can be used to both generate and protect a key that is used for the impath communication between the nShield HSM and the client. Thus a strongly protected key is used at both ends of the impath. A local PCIe HSM can also be used to perform the role of the nToken.

Software-based authentication is only supported from version 12.60. Previously enrolled clients using software-based authentication will need to be re-enrolled if an earlier version of Security World software is installed.

The client configuration process varies slightly depending on whether you are enrolling the client with or without secure client authentication:

1. On the nShield HSM front panel, use the right-hand navigation button to select **System > System configuration > Client config > New client**.

   The following screen is displayed:

   ```
   Client configuration

   Please enter your
   client IP address:



   CANCEL          NEXT
   ```

Enter the IP address of the client, and press the right-hand navigation button.

2. Use the touch wheel to confirm whether you want to save the IP or not, and press the right-hand navigation button.

```
Client configuration

Do you want to save
the IP in the config?
(No for dynamic client
IPs)
          No
BACK            NEXT
```

3. Use the touch wheel to select the connection type between the nShield HSM and the client.

```
Client configuration

Please choose the
client permissions

    Unprivileged

BACK            NEXT
```

The following options are available:

Option	Description
Unprivileged	Privileged connections are never allowed.
Priv. on low ports	Privileged connections are allowed only from ports numbered less than 1024. These ports are reserved for use by root on Linux.
Priv. on any ports	Privileged connections are allowed on all ports.

A privileged connection is required to administer the nShield HSM (for example, to initialize a Security World). If privileged connections are allowed, the client can issue commands that can interfere with the normal operation of the nShield HSM, such as clearing it. Entrust recommends that you allow only unprivileged connections unless you are performing administrative tasks.

4. When you have selected a connection option, press the right-hand navigation button.

The following screen is displayed:

```
Client configuration
```

```
Do you want secure
authentication enabled
on this client?

        Yes
BACK            NEXT
```

a. Select `No` and press the right-hand navigation button to configure the client without secure authentication. The authentication of the client will be based on the IP address only.

b. Select `Yes` and press the right-hand navigation button to configure the client with secure authentication.

5. On the nShield HSM, enter the number of the port on which the client is listening (the default is 9004), and press the right-hand navigation button. The following screen is displayed:

```
Client configuration

On what port is the
client listening?

        9004

CANCEL          NEXT
```

6. Skip this step if you have not selected secure authentication.

   If an nToken is installed in the client, you will be asked to choose which authentication key to use. Select the desired option and press the right-hand navigation button:

```
>3138-147F-2D64
 Software Key


BACK       SELECT
```

   a. The ESN of the nToken installed in the client.

   b. "Software Key" for software-based authentication.

   If no nToken is installed in the client, then software-based authentication is automatically selected.

   Software-based authentication is only supported from version 12.60.

7. Skip this step if you have not selected secure authentication.

   The next screen asks you to verify that the key hash displayed by the nShield HSM

matches the client key hash:

```
Remote 3138-147F-2D64
reported the key hash:
 691be427bb125f387686
 38a18bfd2eab75623320
Is this EXACTLY right?

CANCEL         CONFIRM
```

The client key hash is obtained by running the commands described below. Take a copy of the returned key hash and compare it to the value reported on the nShield HSM display.

With software-based authentication

Run the following command on the client:

```
enquiry -m0
```

This command returns the software key hash, tagged as `kneti hash`, as part of its output, for example:

```
Server:
  enquiry reply flags   none
  enquiry reply level   Six
  ...
  kneti hash            f8222fc007be38b78ebf442697e244dabded38a8
  ...
```

With nToken authentication

Run the following command on the client:

```
ntokenenroll -H
```

This command produces output of the form:

```
nToken module #1
nToken ESN:      3138-147F-2D64
nToken key hash: 691be427bb125f387686
                 38a18bfd2eab75623320
```

Check that the ESN also matches the one reported on the nShield HSM display.

If the client key hash matches the one reported on the nShield HSM display, press the right-hand navigation button to continue the RFS configuration. Otherwise press the left-hand navigation button to cancel the operation.

8. The nShield HSM displays a message reporting that the client has been configured. Press the right-hand navigation button again.

To modify or delete an existing client, select **System > System configuration > Client config** and perform the appropriate procedure.

If you want to use multiple clients with the nShield HSM, you must enable additional client licenses (see Optional features). When you have additional client licenses enabled, to configure more clients, repeat the appropriate steps of the procedure described in this section for each client.

2.5.4.1. Remote configuration of additional clients

Additional clients can be added remotely provided that config push is enabled. This can be done from the RFS or a client computer.

Before you can use multiple clients with the nShield HSM, you must enable the additional clients as described in Optional features.

To add clients remotely:

1. Copy the HSM configuration file, `NFAST_KMDATA/hsm-ESN/config/config` (**Linux**) or `NFAST_KMDATA\hsm-ESN\config\config` (**Windows**), to `config.new` in the same directory.

   The path to the new file will be `NFAST_KMDATA/hsm-ESN/config/config.new` (**Linux**) or `NFAST_KMDATA\hsm-ESN\config\config.new` (**Windows**).

2. Add a new entry to `config.new` in the `hs_clients` section to contain the details of the client to be added.

   The following two entries must exist in the configuration file:

   ```
   addr=<client_IP>
   clientperm=permission_type
   ```

   Where:

   `<client_IP>` can be either the IP address of the client or `0.0.0.0`, `::`, or blank if the HSM is to accept clients identified by their key hash instead of their IP address.

   `0.0.0.0` or `::`, and blank result in the same behavior. You can only use them in the configuration file, you cannot enter these values in the front-panel user interface.

   The default is blank.

If you set both the `<client_IP>` field (the client's IP address) and the key hash, the HSM must identify clients from both of these fields.

`permission_type` defines the type of commands the client can issue (`unpriv`, `priv` or `priv_lowport`).

a. If the client is using an nToken, two additional entries will need to be added to the configuration file:

```
esn=nToken_ESN
keyhash=nToken_keyhash
```

Where `nToken_ESN` is the ESN of the client's nToken and `nToken_keyhash` is the hash of the key that the client's nToken should authenticate itself with.

b. If the client is using software-based authentication, one additional entry will need to be added to the configuration file:

```
keyhash=software_keyhash
```

Where `software_keyhash` is the hash of the software generated key that the client should authenticate itself with. The `ESN` entry must be blank or omitted for software-based authentication.

Each client entry after the first must be introduced by a line consisting of one or more hyphens.

3. Load the updated configuration file on to the nShield HSM. To do this, run the following command:

```
cfg-pushnethsm --address=<module_IP_address> <new_config_file>
```

In this command, `<module_IP_address>` is the IP address of the nShield HSM and `<new_config_file>` is the location of the updated configuration file (`config.new`).

Alternatively, you can load the configuration file using the nShield HSM front panel without enabling config push. The configuration file (`config.new`) must be in the `/opt/nfast/kmdata/hsm-ESN/config` (**Linux**) or `%NFAST_KMDATA%\hsm-ESN\config` (**Windows**) directory on the remote file system. To do this, select **System > System configuration > Config file options > Fetch configuration**.

An SEE machine cannot be installed or configured using the fetch configuration option from the front panel, the config push feature

> must be used for this. See Remotely loading and updating SEE machines for more information.

2.5.5. Changing the nShield HSM configuration from the Front Panel to use a client

From the Front Panel, you can change the settings that your nShield HSM stores about its client.

> The Front Panel does not display all current properties for the client. Entrust recommends that you retrieve the current client settings before you start the update. See HSM and client configuration files. During the configuration update, check the current configuration file against the values displayed on the Front Panel. Check the values at each configuration step before proceeding to the next step and finally confirming the changes.

2.5.6. Configuring client computers to use the nShield HSM

Each client computer must be configured to use the internal security module of your nShield HSM. There are two methods for achieving this:

- Enrolling the client with the configuration file.
- Enrolling the client with command-line utilities.

2.5.6.1. Enrolling the client with the client configuration file

The client configuration files are in the `/opt/nfast/kmdata/config` (**Linux**) or `%NFAST_KMDATA%\config` (**Windows**) directory on the client computer's file system.

> For this release, you must generate a new client configuration file to take advantage of the new functionality. To generate a new client configuration file, back up your existing configuration file and run cfg-mkdefault. This generates a template for the configuration file into which you can copy the settings from your old configuration file.

The `nethsm_imports` section defines the network HSMs that the client imports (See nethsm_imports). It can also be set up by the `nethsmenroll` utility.

- Edit the following mandatory fields: `local_module`, `remote_ip`, `remote_esn`, `remote_keyhash` and `privileged`. The default value for `remote_keyhash` (40 zeros)

specifies that no authentication should occur.

- The ESN and hash of the HSM to import can be retrieved by running the command

```
anonkneti remote_ip
```

- If the client is to be enrolled with an nToken, open a command line window, and run the command: `ntokenenroll --H`. This command produces output of the form:

```
nToken module #1
nToken ESN: 3138-147F-2D64
nToken key hash: 691be427bb125f3876838a18bfd2eab75623320
```

- Enter the nToken's ESN in the field `ntoken_esn` in the config file.
- Each HSM entry after the first must be introduced by a line consisting of one or more hyphens, for example `---`.
- At the command line run cfg-reread to reload the hardserver's configuration.
- Verify that the client can use the nShield HSM by running `enquiry`, which reports the HSM's status.

If the client is to be enrolled with either software-based authentication or no authentication, the `ntoken_esn` field must be left **empty**.

For information about configuration file contents, see HSM and client configuration files.

2.5.6.2. Enrolling the client from the command line

The `nethsmenroll` command-line utility edits the client hardserver's configuration file to add the specified nShield HSM. For more information about the options available to use with `nethsmenroll`, read the following section Client configuration utilities, or run the command:

```
nethsmenroll --help
```

To retrieve the ESN and HKNETI of the HSM, run the command

```
anonkneti <ip-address>
```

This command produces output of the form:

```
3138-147F-2D64 691be427bb125f38768638a18bfd2eab75623320
```

If the nShield HSM's ESN and HKNETI are not specified, `nethsmenroll` attempts to contact

the HSM to determine what they are, and requests confirmation.

1. If you are enrolling the client *with* an nToken, run the command:

   ```
   nethsmenroll --ntoken-esn <nToken ESN> [Options] --privileged <Unit IP> <Unit ESN> <Unit KNETI HASH>
   ```

2. If you are enrolling the client *without* an nToken, that is, you are using software-based authentication or no authentication, run the command:

   ```
   nethsmenroll [Options] --privileged <Unit IP> <Unit ESN> <Unit KNETI HASH>
   ```

These commands produce output of the form:

```
OK configuring hardserver's nethsm imports.
```

2.5.6.3. Client configuration utilities

We provide the following utilities for client configuration:

Utility	Description
`nethsmenroll`	This utility is used to configure the client to communicate with the nShield HSM.
`config-serverstartup`	This utility is used to configure the client's hardserver to enable TCP sockets.

2.5.6.3.1. nethsmenroll

The `nethsmenroll` command-line utility edits the client hardserver's configuration file to add the specified nShield HSM. If the nShield HSM's `ESN` and `HKNETI` are not specified, `nethsmenroll` attempts to contact the nShield HSM to determine what they are, and requests confirmation.

Usage:

```
nethsmenroll [Options] --privileged <hsm-ip> <hsm-esn> <hsm-kneti-hash>
```

Options	Description
`-m|--module=MODULE`	Specifies the local HSM number to use (default is `0` for dynamic configuration by hardserver).
`-p|--privileged`	Causes the hardserver to request a privileged connection to the nShield HSM (default `unprivileged`).

Options	Description
-<hsm-ip>	The IP address of the nShield HSM, which could be one of the following: • an IPv4 address • an IPv6 address, including a link-local IPv6 address • a hostname
-r|--remove	Deconfigures the specified nShield HSM.
-f|--force	Forces reconfiguration of an nShield HSM already known.
--no-hkneti-confirmation	Does not request confirmation when automatically determining the nShield HSM's ESN and HKNETI This option is potentially insecure and should only be used on secure networks where there is no possibility of a man-in-the-middle attack. For guidance on network security, see the *nShield Security Manual*.
-V|--verify-nethsm-details	When the ESN and HKNETI have been provided on the command line, verifies that the selected HSM is alive, reachable and matches those details.
-P|--port=PORT	Specifies the port to use when connecting to the given nShield HSM (default 9004).
-n|--ntoken-esn=ESN	Specifies the ESN of the nToken to be used to authenticate this client. If the option is omitted, then software authentication will be used instead.

2.5.6.3.2. config-serverstartup

The `config-serverstartup` command-line utility automatically edits the `[server_startup]` section in the local hardserver configuration file in order to enable TCP ports for Java and KeySafe. Any fields for which values are not specified remain unchanged. After making any changes you are prompted to restart the hardserver.

Run `config-serverstartup` using commands of the form:

```
config-serverstartup [OPTIONS]
```

For more information about the options available to use with `config-serverstartup`, run the command:

```
config-serverstartup --help
```

2.5.7. Client Session Resumption (Impath Resilience)

Client Session Resumption (Impath Resilience) is a mechanism which mitigates brief network disruption between the nShield HSM and the client. When the hardserver on the nShield HSM detects that the client connection is lost, it will "park" the client session for a limited time enabling it to be resumed when the network recovers. The parked session (resumable session) holds state such as the loaded keys so that applications on the client machine can resume automatically when network recovers, without the need to re-load keys or other objects.

From v13.6 of the nShield HSM image, clients that share the same IP address, for example those behind NAT (Network Address Translation), or which share the same KNETI (network authentication key) will be treated as distinct sessions and will all benefit from Client Session Resumption. This supports scenarios such as dynamically spun-up clients running in VMs or containers. Prior to v13.6, only one client session for a given client "identity" was parked and benefited from session resumption.

2.5.8. Client Licensing

The license model, from v13.6 of the nShield HSM image, is based on the number of concurrent client connections rather than the number of clients configured in the `hs_clients` section in the nShield HSM configuration file. Prior to v13.6, licenses were enforced based on the number of clients configured; from v13.6 onwards, any number of clients may be configured in the nShield HSM configuration file, independently of the license limit.

The behavior in v13.6 onwards means that clients count towards the license limit only when they are actually running.

A licensable client (also referred to as a crypto client) is a client hardserver that creates a privileged or unprivileged nCore client connection to the nShield HSM. This is typically configured by running `nethsmenroll` on the client machine, or editing the `nethsm_imports` section of the client configuration file.

The license utilization can be queried using the `stattree` tool. The `MaxCryptoClients` field reports the number of crypto clients permitted by the license. The `CryptoClientCount` field reports the current usage, that is the number of currently live network connections from crypto clients plus any parked sessions. See CryptoClientCount.

```
+#RemoteServers:
   +#1:
      +#ServerGlobals:
         -Uptime               1805
         -CmdCount             1604
```

```
        -CmdBytes             65724
        -CmdMarshalErrors     0
        -ReplyCount           1604
        -ReplyBytes           81512
        -ReplyMarshalErrors   0
        -ClientCount          2
        -MaxClients           2
        -DeviceFails          0
        -DeviceRestarts       0
        -CryptoClientCount    2
        -MaxCryptoClients     3
```

When a client session has been parked as a result of Client Session Resumption (Impath Resilience), the parked session will hold the license until either the parked session expires or a new client requires a license. If there is otherwise no license slot available for the new client, the oldest parked session will be evicted and the recovered license is given to the new client.

2.5.8.1. Sharing KNETI (network authentication key)

Sharing the KNETI key may be useful in scenarios where multiple instances of the same client configuration are spun-up such as in containers or virtual environments. For security, it is recommended to always require KNETI authentication for clients even if the IP address is static, and even if multiple instances of the same client configuration are being spun-up.

Each crypto client that connects to the nShield HSM will occupy a license, even if it shares an IP address and/or KNETI with another crypto client.

A shared KNETI network authentication key is configured in the `hs_clients` section of the configuration by setting the `keyhash` field. If the client IP address is dynamic, the `addr` is configured as `0.0.0.0`, `::`, blank or omitted. If the clients share the same IP address, for example those behind NAT (Network Address Translation), then the `addr` is configured with the shared IP address.

For example, clients are in containers (or virtual machines), connected to a NAT virtual network on a host, the host's IP address is `192.30.100.100`. The `addr` is configured with the hosts IP address as it is shared with the clients. The `keyhash` field is added with the client's `kneti hash`, found in the response of the `enquiry -m0` command, executed on the client.

```
[hs_clients]
addr=192.30.100.100
clientperm=permission_type
keyhash=software_keyhash
```

In another case, clients have dynamic IP addresses on the same network as the nShield HSM. The `addr` field can be removed from the `hs_clients` section as it is not needed. The `keyhash` field is required and can be shared between multiple clients.

```
[hs_clients]
clientperm=permission_type
keyhash=software_keyhash
```

For multiple clients to share the same KNETI (specified via its keyhash), they must share the same `kneti-hardserver` key file. This key could be from an already enrolled client and can be copied from (and to) the `/opt/nfast/hardserver.d` directory (**Linux**) or `%NFAST_DATA_HOME%\hardserver.d` directory (**Windows**, `C:\ProgramData\nCipher\hardserver.d` by default). The existing permissions (ACLs on Windows) and file ownership must be preserved when copying the KNETI file. If updating an existing client hardserver to use a shared `kneti-hardserver` file it would have to be restated; details can be found in the Stopping and restarting the hardserver section. The clients can be enrolled using `nethsmenroll` or by editing the `nethsm_imports` section of the client `config` file as usual; if deploying a shared configuration to a container, this configuration is likely to be prepared in advance and then included as part of the container image or mounted from the host.

2.5.9. Configuring NTP in the nShield HSM

The nShield HSM has a standard NTP client that can be configured to support synchronization of system time on the HSM with one or more NTP servers. Network Time Protocol (NTP) is intended to synchronize all participating computers to within a few milliseconds of Coordinated Universal Time (UTC). System time on the nShield HSM is independent of the Real Time Clock (RTC) in the HSM and is used for log messages and front panel display.

Entrust recommends that the NTP Server(s) are trusted servers within your local network, not internet time servers.

After configuring NTP the HSM has to be re-started for the configuration to take effect. When starting up after being configured, the NTP client can make a step change to the system time to bring it into line with that of the NTP server(s). At all other times, the NTP client will only slew (gradually change) the system time. When using NTP there should be no need to set the system time by setting time and date from the front panel of the nShield HSM.

Before configuring NTP you must ensure the following:

- **config push** is enabled for the remote computer used to configure NTP. See Enable config push on nShield Connect for more information.
- The client computer enabled for **auto push** is configured for Privileged connections,

see Configuring the nShield HSM to use the client, so that the nShield HSM can be rebooted from the client computer.

2.5.9.1. Using the NTP configuration tool

NTP is configured using the cfg-pushntp utility on a client computer.

2.5.9.2. Restarting the nShield HSM

After configuring NTP, restart the nShield HSM, for example using `nethsmadmin --module=<MODULE> --reboot` (see nethsmadmin). Once the HSM has rebooted and the syslog output is available, check that there are no NTP failures reported in the syslog output.

2.5.9.3. Enable NTP for IPv6 (nShield 5c)

You can use HSM images that were built with buildroot from version 13 onwards with NTP over IPv6.

To run ntpd as a non-root user, start ntpd as root and use `switch -u` on the command line to change to the "ntp:ntp" non-root user.

To set NTP up for IPv6:

1. Reset the HSM to factory state using the UI (menu: 1-5). See
2. Assign an IPv6 address to the HSM on interface #1 (eth0).

   For example: `fd01:11::260:e0ff:fe81:4ac1`.

3. Assign an IPv6 address on the same subnet to the RFS host machine that will act as the NTP server.

   For example: `fd01:11::ae9f:98bc:e202:9941`.

4. Enroll the HSM on the RFS using the following commands:

   Ensure that the displayed kneti hash is not just a string of zeroes. If it is, reset the HSM to factory state.

   ```
   opt/nfast/bin/anonketi <HSM IP>

   opt/nfast/bin anonketi <HSM IPv6 address>

   sudo /opt/nfast/bin/rfs-setup --force <HSM IPv6 address> `/opt/nfast/bin/anonketi <HSM IPv6 address>`
   ```

```
sudo /opt/nfast/bin/nethsmenroll -fp <HSM IPv6 address>
```

5. Add the RFS IPv6 address to the HSM (menu: 1-1-3)
   - RFS IP address: `<RFS IPv6 address>`
   - Set RFS config push mode to: `AUTO`
   - Do you want secure authentication enabled on RFS? `YES`
6. Add the RFS IPv6 address to the HSM as a client (menu: 1-1-4)
   - Client IP address: `<RFS IPv6 address>`
   - Do you want to save IP in the config? `Yes`
   - Client permissions: `Priv on any port`
   - Do you want secure authentication enabled on this client? `No`
7. Check that the RFS can detect the HSM. If any errors appear, such as `AccessDenied`, wait for one minute and then run the command again.

   ```
   /opt/nfast/bin/enquiry
   ```

8. Ensure that auto-push is enabled on the HSM. If auto-push is not enabled, `cfg-pushntp` reports an `UnknownHostVolume` error.
   a. Toggle auto-push on:

      System > System configuration > Config file options > Setup auto push > Yes

   b. Add the RFS IPv6 address:

      System > System configuration > Config file options

9. Using the HSM front panel, set the date and time to be different from those on the RFS host machine. This will help you verify that HSM is updating from the RFS when completing this task.

   You can view the HSM's status, including the date and time, by scrolling through the front (top) menu on the front panel.

10. Install NTP on the RFS.

    For an Ubuntu 18.04 host:

    a. Run the following command:

       ```
       sudo apt install ntp
       ```

    b. Add the following lines to the `/etc/ntp.conf` file.

```
0.au.pool.ntp.org
1.au.pool.ntp.org
2.au.pool.ntp.org
3.au.pool.ntp.org
```

c. Restart the NTP service:

```
sudo service ntp restart
```

d. Verify that the service successfully restarted and is running:

```
sudo service ntp status
```

e. Check that the firewall is not blocking UDP and TCP ports 123.

11. Push the NTP configuration to the HSM from the RFS host machine:

```
/opt/nfast/bin/cfg-pushntp --address=<HSM IPv6 address> --ntp1=<RFS IPv6 address> enable
```

12. Reboot the HSM from the RFS host machine. The HSM updates its system time from the NTP server by running the `/sbin/setostime` script when it starts up from the `/etc/inittab` file.

```
sudo opt/nfast/bin/nethsmadmin --reboot --module 1
```

13. When the HSM has rebooted, check that the date and time in the front (top) menu match the date and time on the RFS host machine.

2.5.10. Configuring Remote Syslog

The nShield HSM can be configured to send logs directly to a remote syslog server, listening on a User Datagram Protocol (UDP) port, by editing the `remote_sys_log` section of the config file: `remote_syslog_ip` can be an IPv4 or IPv6 address.

This behavior can be configured in addition to storing the log files on the RFS. This means that you can configure the logs to be sent to a remote syslog server regardless of whether the nShield HSM logs are configured to be stored on the HSM or the RFS. For further information see Configuring log file storage.

There is no additional formatting of log messages (the logs sent are the same log messages that will appear on the unit or the RFS). It is the responsibility of the remote syslog server / SIEM application to format, sort and aggregate the incoming log messages.

To configure an HSM to send logs to a remote syslog server:

1. Configure the remote host to receive logs from the HSM.

   For instructions, see the documentation for the operating system of the remote host.

2. Make sure that the HSM can communicate with an RFS host so that you can push the new configuration file to the HSM. See Basic HSM, RFS and client configuration for more information.
3. If your HSM configuration file predates the functionality to send logs to a remote syslog server, manually create the `remote_sys_log` section in the config file for the remote module.

   See remote_sys_log.

4. In the `remote_sys_log` section, add the following settings:

   ```
   remote_sys_log_ip=REMOTE_SYSLOG_SERVER_IP
   remote_sys_log_port=REMOTE_SYSLOG_SERVER_PORT
   ```

5. Run the following command to push the new config file to the HSM:

   ```
   cfg-pushnethsm
   ```

   If you are using an older version of the Security World software with a Connect image that supports remote syslog, you will see an error message: **unrecognized section name: 'remote_sys_log'**. Use the following command to push the updated configuration file:

   ```
   cfg-pushnethsh --force
   ```

   If you are using a version of the Security World software that supports remote syslog with an image that does not support remote syslog, the configuration file will be pushed to the HSM but the HSM will reject it. You will see that the upgraded configuration file on the RFS is unchanged.

To turn off sending logs to a remote syslog server, remove the entries from the `remote_sys_log` section of the configuration file and push the updated configuration file.

2.5.11. Set up client cooperation

If you do not need to allow multiple clients access to your remote file system (RFS), you only need to follow the instructions provided in Configuring the Remote File System (RFS) to initialize your RFS. If you need to allow other clients to access your RFS (that is, able to access the RFS to share key data), see Client cooperation.

2.5.11.1. Useful utilities

- anonkneti
- rfs-sync

2.5.11.2. Setting environmental variables

This section describes how to set Security World Software-specific environment variables. You can find detailed information about the environment variables used by Security World Software in Environment variables.

Linux

You can set Security World Software-specific environment variables in the file `/etc/nfast.conf`. This file is not created by the installation process: you must create it yourself. `/etc/nfast.conf` is executed by the start-up scripts of nShield HSM services as the root user. This file should only contain shell commands that can safely be run in this context. `/etc/nfast.conf` should be created with access permissions that allow only the root user to write to the file.

Ensure that all variables are exported as well as set.

Windows

You can set Security World Software-specific environment variables as follows:

1. Open the **System** dialog by clicking **System** in the control panel menu.
2. Select the **Advanced** tab and click the **Environment Variables** button.
3. To add a variable, click **New**. Alternatively, to edit an existing variable select an entry in the **System Variables** list and click **Edit**.
4. In the **Variable Name** field, type or edit the name of the environment variable (for example, **NFAST_HOME**).
5. In the **Variable Value** field, type or edit the value to use.
6. Click the **OK** button to set the value, and then click the **OK** button to close the dialog.
7. Open the **Administrative Tools** dialog by clicking the **Administrative Tools** icon in the Control Panel

8. Open the **Services** console by clicking the **Services** icon.
9. From the displayed list of services, select the **nFast Server** icon, and select **Restart the service**.

2.5.11.3. Logging and debugging

Network-attached HSMs: You can view logs generated by the nShield HSM and applications that use it on the unit front panel. Application log messages are handled on the client.

The Security World Software generates logging information that is configured through a set of four environment variables:

- `NFLOG_FILE`
- `NFLOG_SEVERITY`
- `NFLOG_DETAIL`
- `NFLOG_CATEGORIES`

If none of these logging environment variables are set, the default behavior is to log nothing, unless this is overridden by any individual library. If any of the four logging variables are set, all unset variables are given default values.

Detailed information about controlling logging information by means of these environment variables is supplied in Logging, debugging, and diagnostics.

Some components of the Security World Software generate separate debugging information which you can manage differently. On network-attached HSMs, debugging information for applications is handled on the client.

If you are setting up the unit or the client to develop software that uses it, you should configure debugging before commencing software development.

2.5.11.4. Configuring Java support for KeySafe

To use KeySafe, follow the instructions in Using KeySafe.

2.5.12. Routing

If you have configured only one network interface, you do not need to configure a static route for the unit, although you can do so if you wish. If you have configured a second

network interface, you can choose to configure a static route.

If the unit is to connect to a remote host or network that is unreachable through the default gateway, you must set up extra static routes in the system routing table.

To set up the Ethernet interfaces (IPv4 and IPv6), see Basic HSM, RFS and client configuration.

After you have defined static routes, you should test them as described in Testing routes.

If you define, edit, or delete a route, you must reboot the unit before the route can be used and the routing table is updated.

2.5.12.1. Testing routes

When you have set up or edited a route, you should test the route.

2.5.12.1.1. Testing a route between the unit and the client

When you have installed the unit in its final location, you should test the connection between the unit and the client. You can do this from the client, as described in this section, or by using the **Ping remote host** option on the unit. To do this from the unit, select **System > System configuration > Network config > Ping remote host.**

You can also use the method described in this section to test the route between the client and a remote computer.

To test the route from the client to the unit, issue a `ping` command from the client for the IP address that you specified for the unit. (The format of the command and results may vary according to the platform that you are using.)

```
ping <xxx.xxx.xxx.xxx>
```

If the `ping` operation is successful, a message similar to the following is displayed:

```
Pinging xxx.xxx.xxx.xxx with 32 bytes of data
Reply from xxx.xxx.xxx.xxx: bytes=32 time=10ms TTL=125
Reply from xxx.xxx.xxx.xxx: bytes=32 time=10ms TTL=125
Reply from xxx.xxx.xxx.xxx10ms TTL=125
Reply from xxx.xxx.xxx.xxx10ms TTL=125
```

2.5.12.1.2. Testing a route between the unit and a remote host

When you have defined or edited a route from the unit to a remote computer, you should

test it. To do this you can issue a `ping` command from the unit to the IP address of the host.

You can also use this method to test the connection between the unit and the client.

To test a route from the unit to a host:

1. Select **System > System configuration > Network config > Ping remote host**. The following screen appears:

   ```
   Ping remote host
   Select IP address:
    0. 0. 0. 0
   RESET FINISH
   ```

2. Enter the IP address of the remote host.
3. Press **FINISH** to issue the ping. The following message appears:

   ```
   Please wait, running ping
   ```

4. After a short wait, a display similar to the following should appear, showing that the unit has managed to communicate with the host:

   ```
   Ping xxx.xxx.xxx.xxx:
   #0: rtt=0.0503 ms
   #1: rtt=0.0503 ms
   #2: rtt=0.0503 ms
   #3: rtt=0.0503 ms
   4 of 4 packets back.
    min avg max SD
    0.29 0.36 0.50 0.09 ms
   ```

   If not all of the information is visible onscreen, use the touch wheel to scroll up and down the page.

5. Press the left-hand navigation button to return to the **Network config** menu.

2.5.12.2. Tracing network routes

You can trace network routes from the unit and from clients.

2.5.12.2.1. Tracing the route from the unit

You can trace the route taken from the unit to a remote computer. You can also use this method to trace the route from the unit to the client.

1. Select **System > System configuration > Network config > Trace route to host**. The

following screen appears:

```
Trace route
Select IP address:
 0. 0. 0. 0
CANCEL FINISH
```

2. Press the right-hand navigation button to issue the `traceroute`. The following message appears:

```
Please wait, running traceroute.
```

3. After a short wait, a display similar to the following should appear, showing the IP addresses encountered along the route:

```
1: xxx.xx.xxx.x
 0.40 ms
2: *
3: xx.xxx.xx.xxx
 2.1 ms
4: xxx.xx.xxx.xxx
 2.4 ms
BACK
```

   If not all of the information is visible onscreen, use the touch wheel to scroll up and down the page.

4. Press the left-hand navigation button to return to the **Network config** menu.

2.5.12.2.2. Tracing the route from the client

You can trace the route from the client to the unit or (if the client is connected to the public Internet) to a remote computer.

To trace the route from the client to the unit, issue a `traceroute` command from the client for the IP address of the unit. (The format of the command, and results, may vary depending upon the platform that you are using.)

```
tracert <xxx.xxx.xxx.xxx>
```

After a short wait, a display similar to the following should appear, showing the IP addresses encountered along the route:

```
Tracing route to modulename (xxx.xxx.xxx.xxx)/ over a maximum of 30 hops:
1 xxx.xxx.xxx.xxx (xxx.xxx.xxx.xxx) 1.457 ms 0.513 ms 0.311 ms
2 xxx.xxx.xxx.xxx (xxx.xxx.xxx.xxx) 0.773 ms 0.523 ms 1.615 ms
```

2.5.12.2.3. Displaying the routing table

You can view details of all the IP addresses for which the internal security module has a route stored. The routing table includes entries for static routes (which are stored permanently) and local hosts to which the module has set up temporary routing entries.

To view the routing table:

1. Select **System > System configuration > Network config > Show routing table**. A display similar to the following appears:

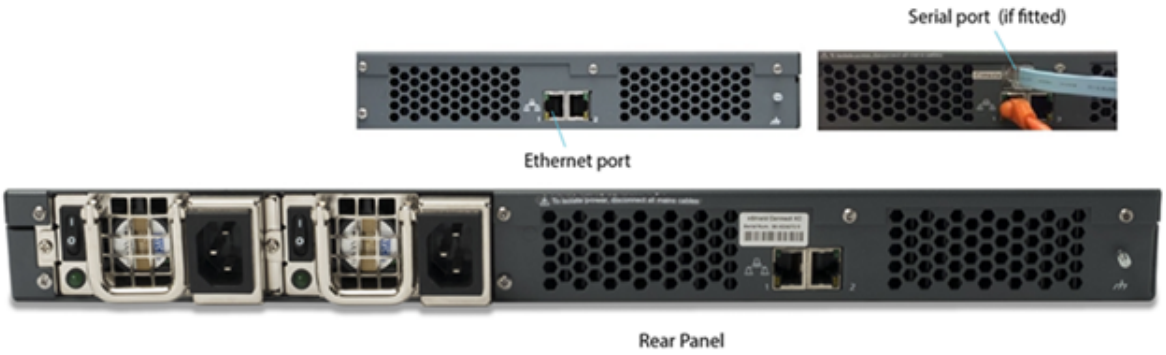
   ```
   Dest Gateway Flg
   1 xxx.xxx.xxx.xxx
    xxx.xxx.xxx.xxx UG
   2 xxx.xx.xx.x
    xxx.x.x.xxx UG
   BACK
   ```

   If not all of the information is visible on the unit display screen, use the touch wheel to scroll up and down the page.

2. Press the left-hand navigation button to return to the **Network config** menu.

2.5.13. Configuring an nShield HSM using the Serial Console

On supported hardware variants (see Model numbers) there is an RJ45 serial port connector at the rear of the nShield HSM marked **Console**. The serial port provides access to a serial console command line interface that enables remote configuration of the unit whilst also facilitating status monitoring. Tasks which would typically require a physical presence in front of the HSM, including setting IP addresses, can be done remotely using the serial console.



This functionality can help provide separation of duty between the data center technician installing the nShield HSM and the administrator configuring and using the HSM. The only required local activity is to connect the nShield HSM to power and to serial and Ethernet ports. Everything that can be configured using the front panel can then be configured remotely either over the serial interface, by using the `nethsmadmin` utility (see nethsmadmin) or by pushing an updated configuration file to the nShield HSM (see HSM and client configuration files for more information).

The Serial Console supports IPv4 and IPv6 addresses.

2.5.13.1. Serial port configuration

The RJ45 connector can be directly connected to your client machine or connected to a serial port aggregator for remote access.

To access the serial console command line interface, first determine the device name of the serial connection once it is connected to your client machine. Then configure the serial port connection in your serial port communication software with the following parameters:

- Line Speed (baud): 115200

This is the default baud rate. If you have manually changed the baud rate to 9600 (see here), enter this value instead.

- Data bits: 8
- Parity: None
- Stop bits: 1.

Once the connection is established, press **Return** until a login prompt is displayed. The login prompt should look like:

```
nethsm login:
```

2.5.13.2. Change the baud rate

To change the baud rate using the front panel, navigate to **System** > **System configuration** > **Remote config options** > **Serial Console** > **Serial baud rate** and select the required baud rate.

to change the baud rate remotely:

1. Copy the config file to `config.new`.
2. If the `config.new` file does not already contain the following section, add it to the file.

```
[cli]
# Start of the cli section
# The serial CLI baud rate configuration. Restart your CLI connection after
# changing.
# Each entry has the following fields:
#
# Set to "115200" or "9600" to select the relevant baud rate for the serial
# CLI connection. Note active CLI serial connections are broken upon the
# setting of a new baud rate.
baud_map=BAUDRATE
```

3. Edit the `baud_map` value.
4. Push `config.new` to the HSM using `cfg_pushnethsm`.

Changing the baud rate using the front panel creates the `[cli]` section in the config file if it does not already exist.

2.5.13.2.1. Troubleshooting

Error	Action
Nothing on the screen	Press **Enter** a few times. Make sure that the RJ45 connector is properly connected and that remote configuration is enabled on the nShield HSM, see Enabling and disabling the serial console.
The serial console appears frozen and it's not possible to log into it for up to five minutes	Frequent logging in and out of the serial console will temporarily disable it for five minutes.
Miscellaneous characters displayed on the screen	Make sure the serial port connection is configured correctly, see Serial port configuration.

2.5.13.3. Creating a serial console session

The username for accessing the serial console is `cli` and the default password is `admin`.

On first login you will be prompted to change the password for the `cli` user. The minimum length of the new password is 5 characters.

For guidance on selecting a password, see the *nShield Security Manual*.

nShield 5c: You must enter a new password. You cannot reuse `admin` the first time you change the password.

Once you are successfully logged in to a serial console session you will see the welcome message:

```
Welcome to the nShield HSM Serial Console. Type help or ? to list commands.

(cli)
```

The serial console session will automatically logout after 180 seconds of inactivity. To manually end a session, use the **logout** command.

2.5.13.4. Enabling and disabling the serial console

The serial console interface can be enabled and disabled using the nShield HSM front panel.

- To enable the serial console interface, navigate to **System** > **System configuration** > **Remote config options** > **Serial Console** and set to **On**.
- To disable the serial console interface, set **Serial Console** to **Off**.

The serial interface is enabled by default and will turn back on with the default password if

the unit is reset to its factory state. This means if you do not want the serial console enabled you will need to disable it after each factory state.

If you do not see the menu option **System** > **System configuration** > **Remote config options** > **Serial Console** on the front panel then this means that your nShield HSM does not support the serial console feature (the hardware does not support serial access).

The availability of the serial console feature can also be checked remotely from an enrolled client by running the enquiry utility.

Feature availability	Enquiry output
Serial console available	... level six flags SerialConsoleAvailable ...
Serial console not available	... level six flags none ...

2.5.13.5. Serial console commands

The serial console command line interface provides the following commands:

Command	Description
`cs5`	Configures CodeSafe functionality on the nShield 5c
`date`	Get/set the HSM system time
`enquiry`	Prints enquiry data from the HSM
`esn`	Show the Electronic Serial Number (ESN) of the HSM
`expose`	Get/set the expose NIC `nethsm_settings` configuration
`factorystate`	Reset unit to its original (factory) state This will reset the IP and serial console settings
`gateway`	Get/set the default IPv4 gateway address
`gateway6`	Get/set the default IPv6 gateway address
`getrtc`	Get the real time clock (RTC) of the nShield 5c Only available on the nShield 5c

Command	Description
`help` or `?`	List available commands with `help` or detailed help with `help cmd`
`hsmdiagnose`	Runs hsmdiagnose on the embedded HSM
`kneti`	Show the (hash of) Kneti (used for enrolling the HSM with clients)
`ks5agent`	Manage the KeySafe 5 agent Configure and enable a KeySafe 5 agent on this Connect to report to a central KeySafe 5 management platform
`logs`	Print nShield 5c diagnostic logs Only available on the nShield 5c
`logout`	Log out of the nShield HSM Serial Console
`maintmode`	Enable/disable maintenance mode Only available on the nShield 5c
`netcfg`	Get/set the IPv4 network interface configuration
`netcfg6`	Get/set the IPv6 network interface configuration
`netdiagnose`	Print network interface statistics
`netenable`	Enable IPv6
`netlink`	Get/set the network interface link, get the network interface MAC address
`bondcfg`	Get/set the HSM bond interface configuration
`bondlink`	Get/set the bond interface link
`passwd`	Set the serial console password
`ping`	Ping a remote host
`push`	Get/set the config push setting
`reboot`	Reboot the HSM
`rfsaddr`	Get/set the RFS IP address, port, optional secure authentication and push mode
`route`	Get/set IPv4 network routes
`route6`	Get/set IPv6 network routes
`routing`	Show the IPv4 routing table
`routing6`	Show the IPv6 routing table

Command	Description
`setrtc`	Set the RTC on the nShield 5c Only available on the nShield 5c Before you can use this command, you must put the nShield 5c into maintenance mode with `maintmode`
`stattree`	Run the `stattree` command
`sworldcheck`	Check for any Security World data on the HSM
`tamperlog`	Show the nShield HSM tamper log
`uptime`	Show how long the nShield HSM has been running (since last boot)
`version`	Show nShield HSM Serial Console version information
`watchdog`	Get/set the `nethsm_watchdog` configuration

For additional help on a command, run `help` command to see additional guidance on command usage, syntax and parameter documentation.

2.5.13.6. Using multiple modules

The hardserver can communicate with multiple modules connected to the host. By default, the server accepts requests from applications and submits each request to the first available module. The server can share load across buses, which includes the ability to share load across more than one module.

If your application is multi-threaded, you can add additional modules and expect performance to increase proportionally until you reach the point where cryptography no longer forms a bottleneck in the system.

2.5.13.6.1. Identifying modules

Modules are identified in two ways:

- By serial number
- By `ModuleID`.

You can obtain the `ModuleID` 's and serial numbers of all your modules by running the `enquiry` command-line utility.

2.5.13.6.2. Electronic Serial Number (ESN)

The serial number is a unique 12-digit number that is permanently encoded into each module. Quote this number in any correspondence with Support.

ModuleID

The `ModuleID` is an integer assigned to the module by the server when it starts. The first module it finds is given a `ModuleID` of 1, the next is given a `ModuleID` of 2, and this pattern of assigning `ModuleID` numbers continues for additional modules.

The order in which buses are searched and the order of modules on a bus depends on the exact configuration of the host. If you add or remove a module, this can change the allocation of ModuleIDs to all the modules on your system.

You can use the `enquiry` command-line utility to identify the PCI bus and slot number associated with a module.

All commands sent to nShield modules require a `ModuleID`. Many Security World Software commands, including all acceleration-only commands, can be called with a `ModuleID` of 0. Such a call causes the hardserver to send the command to the first available module. If you purchased a developer kit, you can refer to the developer documentation for information about the commands that are available on nShield modules.

In general, the hardserver determines which modules can perform a given command. If no module contains all the objects that are referred to in a given command, the server returns an error status.

However, some key-management operations must be performed together on the same module. In such cases, your application must specify the `ModuleID`.

To be able to share OCSs and keys between modules, the modules must be in the same Security World.

2.5.13.6.3. Adding a module

If you have a module installed, you can add further modules without reinstalling the server software.

However, we recommend that you always upgrade to the latest server software and upgrade the firmware in existing modules to the latest firmware.

1. Install the module hardware.
2. (**Linux**) Run the script `/opt/nfast/sbin/install`.
3. Add the module to the Security World. Refer to Adding or restoring an HSM to the

Security World.

2.5.13.6.4. Module fail-over

The Security World Software supports fail-over: if a module fails, its processing can be transferred automatically to another module provided the necessary keys have been loaded. Depending on the mode of failure, however, the underlying bus and operating system may not be able to recover and continue operating with the remaining devices.

To maximize uptime, we recommend that you fit any additional nShield modules for failover on a bus that is physically separate from that of the primary modules.

2.5.14. Stopping and restarting the hardserver

If necessary, you can stop the hardserver on the client, and where applicable the Remote Administration Service, by running the following command. On Windows, this must be a command window with administrative privileges.

Linux

```
/opt/nfast/sbin/init.d-ncipher stop
```

Windows

```
net stop "nfast server"
```

If the Remote Administration Service is running, you will be warned and given the option of continuing or not.

Similarly, you can start the hardserver on the client, and where applicable the Remote Administration Service, by running the following command. On Windows, this must be a command window with administrative privileges.

Linux

```
/opt/nfast/sbin/init.d-ncipher start
```

You can also restart the hardserver on the client, and where applicable the Remote Administration Service, by running the following command:

```
/opt/nfast/sbin/init.d-ncipher restart
```

Windows

```
net start "nfast server"
net start "nfast Remote Administration Service"
```

On Windows, you can also stop, start, or restart the hardserver, and where applicable the Remote Administration Service, from the Windows Control Panel:

1. From the Windows Start menu, open the Windows Control Panel.
2. Double-click Administrative Tools.
3. Double-click Services.
4. Locate **nFast Server** or **nFast Remote Administration Service** in the list of services, and from the **Action** menu, select **Stop**, **Start**, or **Restart** as required.

The **nFast Remote Administration Service**, where applicable, is dependent on the **nFast Server** so should be started or restarted after the **nFast Server**.

2.5.15. Resetting and testing the nShield HSM

2.5.15.1. Default configuration

To reset the unit to its default configuration, select **System > System configuration > Default config** and confirm that you want to set the default configuration.

This removes the configuration of the module but does not change its K_{NETI}.

2.5.15.2. Factory state

To reset the unit to its original (factory) state, select **Factory state** from the main menu and confirm that you want to return the unit to its factory state.

This gives a new K_{NETI} to the unit, which means that you must update the keyhash field of the unit's entry in the `nethsm_imports` section of the configuration file of all the clients that use it.

After a factory reset, ensure you re-enable any dynamic features. See Remotely enabling dynamic feature certificates including nShield HSM client licenses.

For more information about:

- The contents of the configuration files, see Module and client configuration files

- Configuring a new remote file system for the unit, see Configuring the Remote File System (RFS).

2.5.15.3. Testing the installation

To test the installation and configuration, follow these steps:

1. sign in to the client computer as a regular user, and open a command window.
2. Run the command:

   Linux

   ```
   opt/nfast/bin/enquiry
   ```

   Windows

   ```
   enquiry
   ```

   A successful `enquiry` command returns an output of the following form:

   ```
   Server:
    enquiry reply flags  none
    enquiry reply level  Six
    serial number        ####-####-####
    mode                 operational
    version              #-#-#
    speed index          ######
    rec. queue           ####..####
    ---
    version serial       #
    remote port (PV4)    ####

   Module ##:
    enquiry reply flags  none
    enquiry reply level  Six
    serial number        ####-####-####
    mode                 operational
    version              #-#-#
    speed index          #####
    rec. queue           ##..###
    ---
    rec. LongJobs queue  ##
    SEE machine type     PowerPCELF
    supported KML types  DSAp1024s160 DSAp3072s256
    hardware status      OK
   ```

   If the mode is `operational`, the unit is installed correctly.

   If the `enquiry` command says that the unit is not found:

   a. Restart the client computer.
   b. Re-run the `enquiry` command.

2.6. HSM and client configuration files

This chapter describes the configuration files that store the current configuration of an nShield HSM or client.

The *module configuration files* are stored on the internal file system of the nShield HSM. They are updated automatically when the nShield HSM is configured from the front panel. The configuration files are also exported automatically to the remote file system, where they can be edited manually and reloaded on the nShield HSM, if required. For more information, see Client software and module configuration: network-attached HSMs.

The *client configuration files* are stored in the client's file system. The client's hardserver can also be set up using environment variables. Environment variable settings override settings in the client configuration files. For more information, see Setting environment variables.

2.6.1. Location of client configuration files

The client configuration files are in `opt/nfast/kmdata/config/` (**Linux**) or `%NFAST_KMDATA%\config` (**Windows**) on the client computer's file system. This directory can contain the following files:

File	Description
`config`	The master configuration file. This contains the current configuration for the client. It is always present in the directory.
`config-default`	The default configuration file. This can be used to restore factory settings for the client. It is created with the cfg-mkdefault utility.

You can also save backup copies of configuration files in this directory.

2.6.2. Location of module configuration files

When you configure the nShield HSM from the front panel, the configuration files are exported automatically to the remote file system defined in the `rfs_client` section of the module configuration. The exported files are in `/opt/nfast/kmdata/hsm-ESN/config` (**Linux**) or `%NFAST_KMDATA%\hsm-ESN\config` (**Windows**).

In this path, *ESN* is the electronic serial number of the network nShield HSM from which the files were exported. This directory contains the master configuration file `config`, which specifies the current configuration for the nShield HSM. It is always present in the directory.

The remote file system information is also contained in the client configuration file section `remote_file_system`.

2.6.3. Structure of configuration files

The configuration files are text files. They must contain only characters with ASCII values between 32 and 127, and the tab, line break, and return characters.

Lines starting with one or more number sign characters (#) are comments and are ignored. Some comments that document the configuration options are generated by the configuration process. You can add your own comments, but in some cases they may later be overwritten.

A configuration file begins with a single line that specifies the version of the file syntax. This syntax-version line has the format:

```
syntax-version=n
```

In this syntax-version line example, *n* represents the version of the syntax in which the file is written. The system can process a file with a lower syntax version than the one it uses, but not one with a higher version.

After the syntax-version line, the rest of the configuration file consists of sections that can be edited to control different aspects of nShield HSM or client behavior. Each section begins with its name in square brackets, as in this example:

```
[slot_imports]
```

Module configuration files and client configuration files have some sections in common, and some specific to the type of file:

Both	Module file only	Client file only
`server_settings`	`nethsm_settings`	`module_settings`

`server_remotecomms`
 	
`server_remotecomms_ipv6`
	`nethsm_eth`
	
	`nethsm_eth_ipv6`
		`server_performance`

`server_startup`	`nethsm_gateway`
	
	`nethsm_gateway_ipv6`
		`nethsm_imports`

`load_seemachine`	`nethsm_bond`	`rfs_sync_client`

Both	Module file only	Client file only
slot_imports	nethsm_route nethsm_route_ipv6	remote_file_system
slot_exports	nethsm_eth1_enable	remote_administration_service_startup
dynamic_slots	nethsm_bond_enable	
slot_mapping	nethsm_enable	
dynamic_slot_timeouts	cosmod	
auditlog_settings	hs_clients	
	rfs_client	
	sys_log	
	remote_sys_log	
	config_op	
	ui_lockout	

You can update the parameters defined in most of these sections to configure the way that the hardserver handles secure transactions between the nShield HSM and applications that run on the client.

Some sections are updated automatically and should not be edited manually. For more information, see the descriptions of individual sections.

In each section, the bracketed name is followed by a specified set of fields. Each field is on a separate line. Each field begins with its name, followed by an equals sign (=) and a value of the appropriate type. White space can be included at either end of the line (for example, in order to indent lines as an aid to clarity).

Some types of field are grouped into entries. An entry is a set of fields of different types that define an instance of an object (for example, a particular client as distinct from other clients). Entries in the same section are separated by a line that contains one or more hyphens (-). Blank lines and comments are allowed between the fields in an entry.

Strings are case sensitive in the section names and field names.

Multiple clients can be added to one configuration file by separating each client entry from the next with a line consisting of one or more hyphens.

If a particular section is not present in the configuration file, it is assumed to have no entries.

2.6.4. Sections only in module configuration files

2.6.4.1. nethsm_settings

The `nethsm_settings` section defines settings specific to the nShield HSM. The section contains the following fields:

Field	Description
`enable_impath_resilience`	When set to the default `yes` value, this field enables impath resilience for the module. Setting this field to `no` disables impath resilience.
`impath_resilience_timeout`	This field specifies the interval of time that must pass for a resumable impath resilience session to expire. In the event of network errors, clients can reconnect and resume use of keys and other loaded objects until the specified interval has passed (after which all previously loaded objects must be reloaded). Specify this interval either in the form *N* `t`, where *N* is an integer and `t` is `s` for seconds, `m` for minutes, `h` for hours, `d` for days, or `w` for weeks, or as `never` (in which case sessions never expire). If you specify *N* but not `t`, the seconds are assumed. The default setting is `1800` for 1,800 seconds.
`impath_resilience_max_sessions`	This field specifies the maximum number of sessions that can be parked. A value of `0` means that there is no limit and all elligible sessions will be parked. The default is `0`.
`expose_nic_status`	When set to the `yes` value, this field enables NIC details in the `stattree` output. Setting this field to default `no` disables NIC details.
`expose_nic_addresses`	When set to the `yes` value, this field enables NIC addresses in the `stattree` output, only if `expose_nic_status` is enabled. Setting this field to default `no` disables NIC addresses.

2.6.4.2. nethsm_eth

The `nethsm_eth` section defines the Ethernet interfaces for IPv4 for the nShield HSM. Each interface is defined by an entry as follows:

Field	Description
`iface`	The identifier for the interface. This value must be `1` or `0`.

Field	Description
addr	The IP address of the nShield HSM. The default is 0.0.0.0.
netmask	The net mask for the nShield HSM. The default is 0.0.0.0.
gateway	This field is retained for backwards compatibility only. The IP address of the gateway is stored in the nethsm_gateway section and this field is set to 0.0.0.0.
linkspeed	This field specifies the link speed setting. It can be one of auto/1Gb (nShield 5s only), 10BaseT, 10BaseT-FDX, 100BaseTX, or 100BaseTX-FDX. We recommend that you accept the default auto/1Gb or auto setting. On the nShield HSM, this setting can auto-negotiate 1Gb Ethernet.

2.6.4.3. nethsm_eth_ipv6

The nethsm_eth_ipv6 section defines the Ethernet interfaces for IPv6 for the nShield HSM. Each interface is defined by an entry as follows:

Field	Description
iface	The identifier for the interface. This value must be 1 or 0.
addr	The static IPv6 address for this interface. The default is ::. If SLAAC is enabled, this address is ignored.
netmask	The subnet prefix length of the static IPv6 address for the interface. The default is 64.

2.6.4.4. nethsm_gateway

The nethsm_gateway section defines the default IPv4 Ethernet gateway for the nShield HSM. There is one field, as follows:

Field	Description
gateway	The IPv4 address of the gateway for the nShield HSM. The default is 0.0.0.0.

2.6.4.5. nethsm_gateway_ipv6

The nethsm_gateway_ipv6 section defines the default IPv6 Ethernet gateway for the nShield HSM. There is one field, as follows:

Field	Description
gateway	The IPv6 address of the gateway for the nShield HSM. The default is ::.

Field	Description
`linklocal_if`	The ethernet interface (`0` or `1`) to use if the IPv6 default gateway address is a link-local address. The information is not used if the IPv6 default gateway is not a link-local address (default `0`).

2.6.4.6. nethsm_bond

The `nethsm_bond` section defines the HSM bond interface, used for network bonding. The bond interface's address and netmask configuration are inherited from the `eth0` (`iface=0`) configuration. Each entry has the following fields:

Field	Description
`mode`	Possible values: • `802.3ad` • `active-backup` Default: `802.3ad`.
`miimon`	The MII link monitoring frequency in milliseconds. Range: `0` - `10000`. Default: `100`.
`primary`	Primary device. The specified device will always be the active slave while it is available. Possible values: • `eth0` • `eth1` Default: `eth0`. Only valid for `active-backup` mode.
`resend_igmp`	The number of IGMP membership reports to be issued after a failover event. Range: `0` - `255`. Default: `1`. A value of `0` prevents the IGMP membership report from being issued in response to the failover event. Only valid for `active-backup` mode.

Field	Description
`lacp_rate`	The rate in which the Ethernet interface asks the link partner to transmit LACPDU packets in `802.3ad` mode. Possible values: • `slow` • `fast` Default: `slow`. Only valid for `802.3ad` mode.
`xmit_hash_policy`	The transmit hash policy to use for slave selection in `802.3ad` mode. Possible values: • `layer2` • `layer2+3` • `encap2+3` Default: `layer2`.

The process of network bonding, including all the fields above, are described in https://www.kernel.org/doc/Documentation/networking/bonding.txt.

2.6.4.7. nethsm_route

The `nethsm_route` section defines the static IPv4 routes for the nShield HSM. Each route is defined by an entry as follows:

Field	Description
`addr`	The IPv4 address of the route destination. The default is `0.0.0.0`.
`masklen`	The length to mask for the route.
`gateway`	The IPv4 address of the gateway for the route. The default is `0.0.0.0`.

2.6.4.8. nethsm_route_ipv6

The `nethsm_route_ipv6` section defines the static IPv6 routes for the nShield HSM. Each route is defined by an entry as follows:

Field	Description
`addr`	Routable IPv6 address block. The default is `::`.

Field	Description
masklen	The number of bits to mask for the subnet address prefix, that is, the number in the range of 1 to 128) after the / of an address in CIDR format. The default is 64.
gateway	Route gateway. The default is ::.
linklocal_if	The ethernet interface (0 or 1) to use if the IPv6 default gateway address is a link-local address. The information is not used if the IPv6 default gateway is not a link-local address (default 0).

2.6.4.9. nethsm_eth1_enable

The nethsm_eth1_enable section defines if the eth1 interface is enabled.

Field	Description
enable	The indicator of whether the eth1 interface is enabled or disabled (default 'no').

2.6.4.10. nethsm_bond_enable

The nethsm_bond_enable section defines if the bond interface is enabled.

Field	Description
enable	The indicator of whether the bond interface is enabled or disabled (default 'no').

2.6.4.11. nethsm_enable

The nethsm_enable section defines whether IPv4 and or IPv6 are enabled or disabled for the interfaces of the unit. The enable fields are:

Field	Description
iface	The ethernet interface (0 or 1) to which the following fields apply.
enable_ipv4	Indicator of whether the IPv4 protocol on the interface is enabled (default: yes).
enable_ipv6	Indicator of whether the IPv6 protocol on the interface is enabled (default: no).
ipv6_conf_addr	Indicator of whether the interface uses IPv6 static addresses (static) or SLAAC (slaac). The default is static.

2.6.4.12. cosmod

The `cosmod` section defines the input configuration for the nShield HSM. The configuration is defined by an entry as follows:

Field	Description
`keymap`	The selected layout for the keyboard connected to the nShield HSM front panel. This value is either `UK` or `US`.

2.6.4.13. rfs_client

The `rfs_client` section defines the remote file system where the module configuration is backed up and the master copy of the Security World data is located, as follows:

Field	Description
`remote_ip`	The IP address of the RFS.
`remote_port`	The port number on which the RFS hardserver is listening.
`remote_keyhash`	Software or module KNETI hash used to authenticate the RFS, or 40 zeroes to indicate no authentication required (default is 40 zeroes).
`remote_esn`	ESN of the remote module used to authenticate the RFS, or empty when using software KNETI authentication or no authentication required (default is empty).
`push`	Whether to allow the RFS to push configuration files to the nShield HSM: • `ON`: This effectively allows the RFS to remotely configure the nShield HSM, and also to act as an additional privileged cryptographic client above the client licence limit. • `OFF`: This does not allow the RFS to remotely configure the nShield HSM. • `AUTO` (default): If the `remote_keyhash` field is set to a non-zero hash, behaviour shall be as though `ON` was set. If the `remote_keyhash` field is the all-zeroes hash, behaviour shall be as though `OFF` was set.

2.6.4.14. sys_log

The `sys_log` section defines how the nShield HSM logging works:

Field	Description
behaviour	The way the log is written. How the log is written is decided by setting either of the following: • log • push If log is set, the log is written only to the file system of the nShield HSM. It is lost if the nShield HSM is rebooted. Logging stops when the file system is full. If push is set, the log is automatically appended to the log file in the RFS or remote syslog server at the interval specified in push_interval.
push_interval	The number of minutes between the log being appended when behaviour is set to push. The default is 60. The minimum is 1.

2.6.4.15. remote_sys_log

The remote_sys_log section defines a remote syslog server to send the nShield HSM logging (system.log and hardserver.log) to. It can be an IPv4 or an IPv6 address.

Field	Description
remote_syslog_ip	The IP address of the remote syslog server to send logs to. The default is 0.0.0.0.
remote_sys_log_port	The UDP port of the remote syslog server to send logs to. The default is 514.

2.6.4.16. config_op

The config_op section defines whether a client computer is allowed to update the module configuration automatically (to push a configuration) from files on the client:

Field	Description
push	Whether the client is allowed to push configuration files to the nShield HSM. This is decided by setting one of the following: • ON • OFF If on, the client specified in the remote_ip section is allowed to update the configuration of the nShield HSM remotely.
remote_ip	The IP address of the client that is allowed to push config files. If no value is set, or if it is set to 0.0.0.0 or ::, any IP address can push on a new config file.

Field	Description
remote_keyhash	The hash of the key with which the authorized client is to authenticate itself, or (as the default) 40 zeros to indicate no key authentication required.

The default value for remote_keyhash (40 zeros) specifies that no authentication should occur. We recommend specifying a key hash in place of this default.

2.6.4.17. ui_lockout

The ui_lockout section defines whether you can lock the nShield HSM using login settings.

To be compatible with UI lockout, the OCS card(s):

- Must be persistent.
- Must not be remoteable.
- Do not need a passphrase, but if a passphrase is configured, it has to be used.

Field	Description
lockout_mode	Controls front panel access to the nShield HSM. Set to: • locked Enables UI lockout without requiring a logical token. • locked_lt Enables UI lockout with a logical token (OCS) (requires a valid ltui_hash to be set). • unlocked No UI lockout (default).
ltui_hash	The hash of the logical token (LTUI) required to authorize access to the nShield HSM menu structure when lockout_mode is set to locked_lt.
panel_poweroff	This controls the function of the Power button on the nShield HSM front panel when it is in operational mode. The default setting of yes enables the Power button. When set to no, the Power button is disabled.

2.6.5. Sections in both module and client configuration files

2.6.5.1. server_settings

The `server_settings` section defines the settings for the client hardserver you can modify while the hardserver is running.

These flags are used by the `NFLOG_DETAIL` environment variable (see Environment variables to control logging).

The section contains the following fields:

Field	Description
`loglevel`	This field specifies the level of logging performed by the hardserver. See hardserver loglevel and Logging, debugging, and diagnostics.
`logdetail`	This field specifies the level of detail logged by the hardserver. You can supply one or more flags in a space-separated list. For more information about the flags, see the table below.
`connect_retry`	This field specifies the number of seconds to wait before retrying a remote connection to a client hardserver. The default is 10.
`connect_maxqueue`	This field specifies the maximum number of jobs which can be queued on the hardserver. The default is 4096: this is also the maximum value. Setting `connect_maxqueue` to a high value allows high throughput, but may cause long latency if the hardserver goes down.
`connect_broken`	This field specifies the number of seconds of inactivity allowed before a connection to a client hardserver is declared broken. The default is 90.
`connect_keepalive`	This field specifies the number of seconds between `keepalive` packets for remote connections to a client hardserver. The default is 10.
`accept_keepidle`	This field specifies the number of seconds before the first keepalive packet for remote incoming connections. The default is 30. Ideally, `accept_keepalive` should be at least twice the value of the `connect_keepalive` setting on the unattended machines.
`accept_keepalive`	This field specifies the number of seconds between keepalive packets for remote incoming connections. The socket will be closed after up to ten consecutive probe failures. The default is 10. Ideally, `accept_keepalive` should be a value such that (10 * `accept_keepalive`) > `connect_broken` on the unattended machine. Using the default values for both these fields will fulfil this requirement.
`connect_command_block`	When the nShield HSM has failed, this field specifies the number of seconds the hardserver should wait before failing commands directed to that HSM with a `NetworkError` message. For commands to have a chance of succeeding after the nShield HSM has failed this value should be greater than that of `connect_retry`. If it is set to `0`, commands to an nShield HSM are failed with `NetworkError` immediately. The default is 35.

Field	Description
max_pci_if_vers	This field specifies the maximum PCI interface version number. If max_pci_if_vers is set to 0 (the default), there is no limit.
enable_remote_mode	If this field is set to yes (the default value) in the module configuration file, nShield HSM mode changing using the nopclearfail utility is enabled. If set to no, mode changing using the nopclearfail utility is disabled.  Do not set enable_remote_mode in the client configuration file.
enable_remote_reboot	If this field is set to yes (the default value) in the module configuration file, the nShield HSM remote reboot using the nethsmadmin utility is enabled. If set to no, remote reboot using the nethsmadmin utility is disabled. Run cfg-pushnethsm to push the new config file to the module.
enable_remote_upgrade	If this field is set to yes (the default value) in the module configuration file, the nShield HSM remote upgrade using the nethsmadmin utility is enabled. If set to no, remote upgrade using the nethsmadmin utility is disabled. Run cfg-pushnethsm to push the new config file to the module.

These flags are those used by the NFLOG_DETAIL environment variable (see Environment variables to control logging).

You can supply a number of flags with the logdetail field, which specifies the level of detail logged by the hardserver (see the table above). Supply the flags in a space separated list:

Flag	Description
external_time	This flag specifies the external time (that is, the time according to your machine's local clock) with the log entry.
external_date	This flag specifies the external date (that is, the date according to your machine's local clock) with the log entry.
external_tid	This flag specifies the external thread ID with the log entry.
external_time_t	This flag specifies the external time_ti (that is, the time in machine clock ticks rather than local time) with the log entry.
stack_backtrace	This flag specifies the stack backtrace with the log entry.
stack_file	This flag specifies the stack file with the log entry.
stack_line	This flag specifies the message line number in the original library. This flag is likely to be most useful in conjunction with example code we have supplied that has been written to take advantage of this flag.
msg_severity	This flag specifies the message severity (a severity level as used by the NFLOG_SEVERITY environment variable) with the log entry.

Flag	Description
msg_categories	This flag specifies the message category (a category as used by the NFLOG_CATEGORIES environment variable) with the log entry.
msg_writeable	This flag specifies message writeables, and extra information that can be written to the log entry, if any such exist.
msg_file	This flag specifies the message file in the original library. This flag is likely to be most useful in conjunction with example code we have supplied that has been written to take advantage of this flag.
msg_line	This flag specifies the message line number in the original library. This flag is likely to be most useful in conjunction with example code we have supplied that has been written to take advantage of this flag.
options_utc	This flag showing the date and time in UTC (Coordinated Universal Time) instead of local time.
unix_file_descriptor_max	This field sets the number of file descriptors the hardserver must be capable of having open concurrently on Linux. The value must be an integer. If unix_file_descriptor_max is set to **0** (the default), the value will be ignored by the hardserver. If it is set to a positive value, the hardserver will refuse to start if the file descriptor hard limit on the system is less than that value. This configuration entry can be used to make sure that the hardserver is capable of satisfying the maximum number of hardserver connections that applications may make use of.

2.6.5.2. hardserver loglevel

The table in this section describes the loglevels in increasing order of severity. If you set a custom [server_settings]/loglevel, you will get that level and all the more severe levels.

If the legacy NFAST_SERVERLOGLEVEL debug environment variable is set, it overrides any loglevel value set in the configuration file.

[server_settings]/loglevel **value**	**Appears in the hardserver log as**	**Description**
info	Information	Report about the hardserver start-up configuration, connections that have been established or closed. General information for nShield Support for debugging.
notice	Notice	Report about certain start-up events, some non-fatal and routine errors that the hardserver can handle internally.
client	Detected error in client behaviour	Malformed or invalid messages are received from a client, typically from a local client.

[server_settings]/log level value	Appears in the hardserver log as	Description
remoteserver	Remote server error	Malformed messages or protocol errors are received while communicating with remote peers over the nCipher Secure Transport/Impath protocol.
error	Nonfatal error	Unexpected but handled errors from system calls, for example for device or TCP I/O.
serious	Serious error, trying to continue	Unexpected errors from system calls, but they are more serious and likely to indicate an issue somewhere in the system.
internal	Serious internal error, trying to continue	Possible bug in the hardserver, but it might also be an issue in the environment the hardserver is running in.
startup	Fatal error during startup	The hardserver could not start, for example because of an invalid configuration file or because it cannot bind to a TCP socket which is already in use. The hardserver will abort.
fatal	Fatal runtime error	Fatal error usually referring to a non-ignorable error that has occurred after startup such as out of memory errors in certain contexts. Rarely used. The hardserver will abort.
fatalinternal	Fatal internal error	A non-recoverable failure occurred, for example certain internal self-consistency checks to detect program logic errors. The hardserver will abort.

2.6.5.3. server_remotecomms

The `server_remotecomms` section defines the remote IPv4 communication settings for the hardserver's impath port. These are read only at hardserver start-up.

Any changes made to these fields in the HSM config file should be followed by a reboot of the nShield HSM.

It is not recommended that the port number be changed. If it is changed, the port number must match in the configuration of peers. For example, if the port number is changed in the nShield HSM configuration, the remote_port field of the nethsm_imports section of the client-side hardserver config file must be updated to match. The port number can also be specified with the `-P` parameter when running `nethsmenroll`.

This section contains the following fields:

Field	Description
`impath_port`	This field specifies the port on which the hardserver listens for incoming impath connections. The default is 9004. Setting this field to 0 specifies that the hardserver does not listen for incoming connections (do not set to 0 on an nShield HSM). Make sure that firewall settings are consistent with port settings. If you change this field you will need to re-enroll your client with the new port value, see Enrolling the client from the command line.
`impath_addr`	This field specifies the IPv4 address at which the hardserver listens for incoming impath connections. If this field is set to `inaddr_any` (which is the default), the hardserver listens on all IP addresses.
`impath_interface`	This field specifies a string representing the name of the Ethernet interface on which the hardserver listens. This field is only examined if `impath_addr` is set to `inaddr_any`. By default, the `impath_interface` field is empty, which means that the hardserver listens on all interfaces. If the string is not recognized as the name of one of the interfaces of the nShield HSM, the hardserver does not listen.

2.6.5.4. server_remotecomms_ipv6

The `server_remotecomms_ipv6` section defines the remote IPv6 communication settings for the client hardserver. These are read only at hardserver start-up.

Any changes made to these fields in the HSM config file should be followed by a reboot of the nShield HSM.

It is not recommended that the port number be changed. If it is changed, the port number must match in the configuration of peers. For example, if the port number is changed in the nShield HSM configuration, the remote_port field of the nethsm_imports section of the client-side hardserver config file must be updated to match. The port number can also be specified with the `-P` parameter when running `nethsmenroll`.

This section contains the following fields:

Field	Description
`impath_port`	This field specifies the port on which the hardserver listens for incoming impath connections. The default is 9004. Setting this field to 0 specifies that the hardserver does not listen for incoming connections (do not set to 0 on an nShield HSM). Make sure that firewall settings are consistent with port settings. If you change this field you will need to re-enroll your client with the new port value, see Enrolling the client from the command line.
`impath_addr`	This field specifies the IPv6 address at which the hardserver listens for incoming impath connections. If this field is set to `::` (which is the default), the hardserver listens on all IP addresses.
`impath_interface`	This field specifies a string representing the name of the Ethernet interface on which the hardserver listens. This field is only examined if `impath_addr` is set to `::`. By default, the `impath_interface` field is empty, which means that the hardserver listens on all interfaces. If the string is not recognized as the name of one of the interfaces of the nShield HSM, the hardserver does not listen. Setting this field to `0` will disable IPv6 in the hardserver.

2.6.5.5. server_startup

The `server_startup` section defines the settings for the hardserver that are loaded at start-up. Any changes you make to the settings in this section do not take effect until after you restart the hardserver. For more information, see Stopping and restarting the client hardserver.

The section contains the following fields:

Field	Description
`unix_socket_name`	This field is not used on Windows. On Linux, this field specifies the name of the socket to use for non-privileged connections. The default is `/dev/nfast/nserver`. If the `NFAST_SERVER` environment variable is set, it overrides any value set for `unix_socket_name` in the hardserver configuration file. For more information about environment variables, see Environment variables.

Field	Description
`unix_privsocket_name`	This field is not used on Windows. On Linux, this field specifies the name of the socket to use for privileged connections. The default is `/dev/nfast/privnserver`. If the `NFAST_PRIVSERVER` environment variable is set, it overrides any value set for `unix_privsocket_name` in the hardserver configuration file.
`nt_pipe_name`	This field specifies the name of the pipe to use for non-privileged connections on Windows. An empty string specifies none. The default is `\\.\pipe\crypto`. If the `NFAST_SERVER` environment variable is set, it overrides any value set for `nt_pipe_name` in the hardserver configuration file. This field is not used on Linux.
`nt_pipe_users`	This field specifies the name of the user or group who is allowed to issue non-privileged connections on Windows. If this field is empty (which is the default), any user can make non-privileged connections. User or group names must be specified unqualified (for example, *bob* not *MYDOMAIN\bob* or *bob@MYDOMAIN*). This field is not used on Linux.
`nt_privpipe_name`	This field specifies the name of the pipe to use for privileged connections on Windows. An empty string specifies none. The default is `\\.\pipe\privcrypto`. If the `NFAST_PRIVSERVER` environment variable is set, it overrides any value set for `nt_privpipe_name` in the hardserver configuration file. This field is not used on Linux.
`nt_privpipe_users`	This field specifies the name of the user or group who is allowed to make privileged connections on Windows. If this field is empty (which is the default), only processes running with local administrator privilege can make privileged connections. User or group names must be specified unqualified (for example, *bob* not *MYDOMAIN\bob* or *bob@MYDOMAIN*). This field is not used on Linux.
`nonpriv_port`	This field specifies the port on which the hardserver listens for local non-privileged TCP connections. The value `0` (which is the default) specifie Make sure that your network firewall settings are correct. See Before you install the software for more about firewall settings. If the `NFAST_SERVER_PORT` environment variable is set, it overrides any value set for `nonpriv_port` in the hardserver configuration file.

Field	Description
priv_port	This field specifies the port on which the hardserver listens for local privileged TCP connections. The value 0 (which is the default) specifies none. Java clients default to connecting to port 9001. If the NFAST_SERVER_PRIVPORT environment variable is set, it overrides any value set for priv_port in the hardserver configuration file.

2.6.5.6. load_seemachine

The load_seemachine section of the hardserver configuration file defines SEE machines that the nShield HSMs connected to the client should load and, if required, start for use by other clients. Each SEE machine is defined by the following fields:

Field	Description
module	This field specifies the nShield HSM on to which to load the SEE machine. The value must be an integer. A nShield HSM with this ID must be configured on the client computer.
machine_file	This field specifies the file name of the SEE machine.
userdata	This field specifies the userdata file name to pass to the SEE machine on start-up. If this field is blank (" "), the SEE machine is loaded but not started. By default, this field is blank.
worldid_pubname	This field specifies the PublishedObject name to use for publishing the KeyID of the started SEE machine. If this field is blank (" "), the KeyID is not published. This field is ignored if the value of the userdata field is blank.
postload_prog	This field specifies the program to run after loading the SEE machine in order to perform any initialization required by the SEE machine or its clients. The specified program must accept an argument of the form -m module#. To run see-sock-serv directly on the nShield HSM, set this field to sockserv.
postload_args	This field specifies arguments to pass to the program specified by the postload_prog field. The argument -m module# is automatically passed as the first argument. The postload_args field is ignored if postload_prog is not specified or is blank. To run see-sock-serv directly on the nShield HSM, set this field to -p pubname.

Field	Description
pull_rfs	This field specifies whether the SEE machine name and userdata should be pulled from the RFS. The default is no: set to yes to pull the SEE machine and userdata from the RFS before loading on the remote nShield HSM. This field will be ignored if set on client machine configurations.  This field will not be added to existing configuration files if you are upgrading an image. If you require the new functionality enabled by this field, you can add the field to the load_seemachine section of your existing configuration file.

2.6.5.7. slot_imports

The slot_imports section defines slots from remote nShield HSMs that will be available to the client. Each slot is defined by the following fields:

Field	Description
local_esn	This field specifies the ESN of the local nShield HSM importing the slot.
local_slotid	This field specifies the SlotID to use to refer to the slot when it is imported on the local nShield HSM. The default is 2.
remote_ip	This field specifies the IP address of the machine that hosts the slot to import.
remote_port	This field specifies the port for connecting to the nShield HSM.
remote_esn	This field specifies the ESN of the remote nShield HSM from which to import the slot.
remote_slotid	This field specifies the SlotID of the slot to import on the remote nShield HSM. The value of this field must be an integer. The default is 0.

2.6.5.8. slot_exports

The slot_exports section defines the slots on the HSMs connected directly to the client that the client hardserver should export for other network HSMs to import. Each local slot has an entry for each network nShield HSM that can import it, consisting of the following fields:

Field	Description
local_esn	This field specifies the ESN of the local nShield HSM whose slot can be imported by a network nShield HSM.

Field	Description
`local_slotid`	This field specifies the `SlotID` of the slot that is allowed to be exported. The value must be an integer. The default is 0.
`remote_ip`	This field specifies the IP address of the nShield HSM that is allowed to import the slot. Keep this field blank to allow all machines. The default is blank.
`remote_esn`	This field specifies the ESN of the nShield HSM allowed to import the slot. Leave the value blank to allow all permitted nShield HSMs in the Security World. The default is blank.

2.6.5.9. dynamic_slots

The `dynamic_slots` section defines the number of Dynamic Slots that each HSM supports.

Field	Description
`esn`	ESN of the HSM to be configured with Dynamic Slots.
`slotcount`	The number of Dynamic Slots that the HSM is to support. If set to 0 (default) the HSM does not support Remote Administration.

2.6.5.10. slot_mapping

The `slot_mapping` section defines, for each specified HSM, a slot that is exchanged with slot 0 of the HSM. Slot 0 becomes a Dynamic Slot and the local slot becomes the specified slot number. This enables applications and utilities that only support slot 0 to use Remote Administration.

Field	Description
`esn`	ESN of the HSM to which the mapping is applied.
`slot`	The slot number to be swapped with slot 0, so that: • Slot 0 refers to a Dynamic Slot • The specified slot number refers to the local slot of the HSM. If `slot` is set to 0 (default) there is no slot mapping.

2.6.5.11. dynamic_slot_timeouts

The `dynamic_slot_timeouts` section defines timeout values that are used to specify expected smartcard responsiveness for all HSMs associated with the relevant host or client, when using the Remote Administration.

Field	Description
round_trip_time_limit	round_trip_time_limit > 5s + network latency time Round trip (HSM to smartcard and back) time limit in seconds. The card is regarded as removed, if no response has been received within the allowed time. Expected network delays need to be taken into account when setting this. The default is ten seconds.
card_remove_detect_time_limit	card_remove_detect_time_limit >= round_trip_time_limit *2 Maximum number of seconds that can pass without a response from the smartcard, before it is regarded as removed and all the keys that it protects are unloaded. Lower values increase network traffic. The default is 30 seconds.

2.6.6. Sections only in client configuration files

2.6.6.1. module_settings

The module_settings section defines the settings for the nShield HSM that can be changed while the hardserver is running. The section contains the following fields:

Field	Description
esn	This field specifies the electronic serial number of the nShield HSM.
priority	This field specifies the priority of the nShield HSM. The value for this field can be an integer from 1 (highest) to 100 (lowest). The default is 100.

2.6.6.2. server_performance

The server_performance section defines the performance settings for the client hardserver. These are read only at hardserver start-up. This section contains the following fields:

Field	Description
enable_scaling	This field determines whether multi-threaded performance scaling is enabled or not. If this field is set to **auto** (or not set), the hardserver automatically chooses the best option for the available hardware (enabled when using an nShield network-attached HSM, for which scaling is currently optimized, and disabled if using an nShield PCIe or USB-attached HSM). It can explicitly be enabled by setting to **yes**, and explicitly disabled by setting to **no**.

Field	Description
target_concurrency	This field allows the level of concurrency to be tuned. The value must be an integer and will only come into effect when multi-threaded performance scaling is enabled. If target_concurrency is set to **0** (the default), the value will be automatically configured by the hardserver based on the available number of physical CPU cores. The target concurrency configured is written to the hardserver log.

2.6.6.3. nethsm_imports

The `nethsm_imports` section defines the network nShield HSMs that the client imports. It can also be set up by the `nethsmenroll` utility. Each nShield HSM is defined by the following fields:

Field	Description
local_module	This field specifies the ModuleID to assign to the imported nShield HSM. The value must be an integer. An nShield HSM with this ID must not be already configured on the client computer.
remote_ip	This field specifies the IP address of the nShield HSM to import.
remote_port	This field specifies the port for connecting to the nShield HSM.
remote_esn	This field specifies the ESN of the imported nShield HSM.
keyhash	This field specifies the hash of the key that the nShield HSM should use to authenticate itself.
privileged	The value in this field specifies whether the client can make a privileged connection to the nShield HSM. The default is `0`, which specifies no privileged connections. Any other value specifies privileged connections.
ntoken_esn	This field specifies the ESN of this client's nToken, if an nToken is installed.

The default value for `remote_keyhash` (40 zeros) specifies that no authentication should occur. We recommend that you set a specific key hash in place of this default.

2.6.6.4. rfs_sync_client

This section defines which remote file system the client should use to synchronize its key management data:

Field	Description
remote_ip	The IP address of the RFS against which to synchronize.
remote_port	This field specifies the port for connecting to the RFS.
use_kneti	Setting this option to yes to use a local module KNETI instead of the default hardserver's software KNETI to authenticate this client to the RFS.
local_esn	This is only required if use_kneti is set to yes. It is the ESN of the local module used for authentication.
remote_keyhash	Software or module KNETI hash used to authenticate the RFS, or 40 zeroes to indicate no authentication required (default is 40 zeroes).
remote_esn	ESN of the remote module used to authenticate the RFS, or empty when using software KNETI authentication or no authentication required (default is empty).

2.6.6.5. remote_file_system

This section is updated automatically when the rfs-setup utility is run. Do not edit it manually.

The remote_file_system section defines a remote file system on the client by listing the nShield HSMs allowed to access the file system on this client. Each nShield HSM is defined by an entry consisting of the following fields:

Field	Description
remote_ip	This field specifies the IP address of the remote nShield HSM that is allowed to access the file system on this client.
remote_esn	This field specifies the ESN of the remote nShield HSM allowed to access the file system on this client.
keyhash	This field specifies the hash of the key with which the client must authenticate itself to the nShield HSM. The default is 40 zeros, which means that no key authentication is required.
native_path	This field specifies the local file name for the volume to which this entry corresponds.
volume	This field specifies the volume that the remote host would access to use this entry.
allow_read	If this field is set to yes, it means that a remote server is allowed to read the contents of the file. The default is no.
allow_write	If this field is set to yes, it means that a remote server is allowed to write to the file. The default is no.

Field	Description
allow_list	If this field is set to yes, it means that a remote server is allowed to list the contents of the file. The default is no.
is_directory	If this field is set to yes, it means that this entry represents a directory. The default is no.
is_text	If this field is set to yes, it means that line endings should be converted to and from the Linux convention for transfers.

If you upgrade from an earlier software version to v12 and are using Remote Administration, you need to manually add the following sections to your configuration file.

2.6.6.6. remote_administration_slot_server_startup

The remote_administration_slot_server_startup section defines the communication settings that are applied at start-up to the Remote Administration.

Field	Description
port	Which port to use to connect to the Remote Administration. The default is 9005.

2.6.6.7. hs_clients

The hs_clients section defines the clients that are allowed to connect to the nShield HSM. Each client is defined by an entry as follows:

Field	Description
addr	Either the IP address of the client or 0.0.0.0, ::, or blank if the HSM is to accept clients identified by their keyhash instead of their IP address. If no value is set (the field is blank), or if it is set to 0.0.0.0 or ::, only HKNETI identification is allowed. The default is blank. 0.0.0.0 or ::, and blank result in the same behavior. You cannot enter these values in the front-panel user interface. You can only use them in the configuration file and you will have to use the correct key hash for identification if no IP address is configured.

Field	Description
`clientperm`	The type of connection permitted from the client. This is one of the following: • `unpriv` (non-privileged connections) • `priv` (privileged connections) • `priv_lowport` (privileged connections on ports lower than 1024) The default is `unpriv`.
`keyhash`	The hash of the KNETI key that the client should present to authenticate itself. Both software based authentication and nToken authentication are supported, For nToken authentication, a value must also be provided for the `esn` field. The default is 40 zeros, which means that no key authentication is required.
`esn`	The ESN of the client's nToken. (Only applicable if nToken authentication is used.)

2.7. Checking and changing the mode on a network-attached HSM

This appendix tells you how to check and change the mode on the nShield HSM. You must change the mode to perform certain configuration tasks.

2.7.1. Front panel controls

See Front panel controls for a description of the nShield HSM user interface, including the front panel controls.

We recommend that you use a keyboard to manage the front panel menu options and enter text. See Using a keyboard to control the unit for more information.

2.7.2. Available modes

The following modes are available:

Operational The default setting for day-to-day use.

Initialization	Sets the nShield HSM to start in pre-initialization mode. This allows you to use the nShield HSM to create a Security World or add the module to an existing one.
Maintenance	You cannot select this mode manually. It is managed by the nShield HSM and cannot be set by a user.

2.7.3. Identifying the current mode

You can check the current mode of the nShield HSM:

- At the nShield HSM itself
- By using the `enquiry` command-line utility from a client computer
- By using KeySafe from a client computer

2.7.3.1. Checking the mode at the nShield HSM

2.7.3.1.1. The status LED

The nShield HSM Status LED indicates the operational status of the module.

Status LED	Description
On, occasionally blinks off.	**Status: Operational mode** The module is in Operational mode and accepting commands. The more frequently the Status LED blinks off, the greater the load on the module.
Flashes two short pulses, followed by a short pause.	**Status: Initialization mode** Existing Security World data on the module has been erased. The module is automatically placed in Initialization mode after a Security World is created.
Flashes two long pulses followed by a pause.	**Status: Maintenance mode** Used for reprogramming the module with new firmware. The module only goes into Maintenance mode during a software upgrade.

2.7.3.1.2. The front panel display screen

The nShield HSM screen shows a color-coded footer at the bottom of the display when it is not in Operational mode.

Footer color	Text in footer	Meaning
Yellow	Initialization	The system is rebooting or waiting for an Administrator Card to be inserted.
Blue	Maintenance	An administrative task is being performed. This mode is only entered during firmware upgrades.
Red	HSM Failed	The internal module has failed.

2.7.3.2. Checking the mode using enquiry

You can use the `enquiry` command-line utility to display information about the hardserver and the status of the nShield HSM. The `enquiry` utility is in the `bin` subdirectory of the `nCipher` directory. This is usually `/opt/nfast` (**Linux**) or `C:\Program Files\nCipher\nfast` (**Windows**)

To check the mode using **enquiry**:

1. Sign in to the client computer as a user, and open a command window.
2. Run the command:

   Linux

   ```
   opt/nfast/bin/enquiry
   ```

   Windows

   ```
   enquiry
   ```

   Example output:

   ```
   Server:
   enquiry reply flags     none
   enquiry reply level     Six
   serial number           ####-####-####-####
   mode                    operational
   version                 #.#.#
   speed index             ###
   rec. queue              ##..##
   ...
   version serial          #
   remote port (IPv4)      ####

   Module #1:
   enquiry reply flags     none
   enquiry reply level     Six
   serial number           ####-####-####-####
   mode                    operational
   version                 #.#.#
   speed index             ###
   rec. queue              ##..##
   ...
   ```

```
rec. LongJobs queue     ##
SEE machine type        PowerPCSXF
```

In this example, the `mode` line shows that the nShield HSM is in `operational` mode.

2.7.3.3. Checking the mode by using KeySafe

You can use the **Module Status tree** of the KeySafe GUI to identify the current mode of the nShield HSM.

To check the mode using KeySafe:

1. Start KeySafe on a client computer.
2. Locate the **Module Status tree** (part of the **Security World status** panel) positioned to the bottom left of the KeySafe window.
3. Expand the **Security World** and/or **Outside Security World** nodes as required.
4. Locate the appropriate nShield HSM (**Module**).
   The current mode of the module is displayed in the **State** field.

See Using KeySafe for more about using KeySafe. See Module information for more about checking the mode.

2.7.4. Changing the mode

You can change the mode using:

- The front panel controls of the nShield HSM
- The `nopclearfail` command-line utility from a client computer

2.7.4.1. Changing the mode using the front panel controls

To change the mode, use the front panel menu screens and dialogs to do the following:

1. Navigate to **HSM** > **Set HSM mode**.
2. Select **Initialisation** or **Operational** as required.

2.7.4.2. Changing the mode using remote mode and nopclearfail

You can enable or disable changing the mode remotely, see `enable_remote_mode` in the server_settings section or the *Top-level menu* chapter of the HSM *Install Guide*. Once you have enabled remote mode changes, you can change the mode of the nShield HSM from a

computer using the `nopclearfail` command, without accessing the unit itself.

2.7.4.2.1. Available commands

You can use the following commands to change the mode of a module:

Command	Resulting mode
`nopclearfail --operational | -O`	Operational
`nopclearfail --initialization | -I`	Pre-initialization

To change the mode, do the following:

1. Run either:
   a. The `nopclearfail --operational | -O` command.
   or:
   b. The `nopclearfail --initialization | -I` command.
   When finished, the system responds with `OK`.

   The system responds with `OK`, regardless of whether the mode of the nShield HSM has changed or not. To confirm that state of the module, do the following:

2. Run the `enquiry` command.
   The `mode` line of the `Module` section displays the current mode.

2.8. Upgrading the image file and associated firmware: network-attached HSMs

2.8.1. Version Security Number (VSN)

Each Connect image file has a Version Security Number (VSN). In addition, the internal module (HSM) firmware has its own individual VSN. This number is increased whenever we improve the security of the image file and/or firmware.

We supply several versions of the module firmware. You can always upgrade to firmware with an equal or higher VSN than that currently installed on your module.

The Version Security Number (VSN) stands as a safeguard to prevent earlier and potentially less secure images and accompanying firmware from being loaded onto the nShield HSM. It prevents the loading of an image file with a lower VSN than the existing VSN. The VSN is

not incremented with every release, but only in the event of a significant security enhancement to the nShield HSM and / or its internal module.

The Connect image VSN is only available from the nShield HSM Front Panel UI.

2.8.2. Key data

During an upgrade Security World and key data are preserved on the RFS host computer. Once you have upgraded the Connect image you must restore the unit to the Security World if you wish to continue using the key data.

Adding or restoring a module will require authorisation from a quorum of Administrator cards.

When upgrading the nShield HSM image file, client licenses and features activations on the HSM will persist. However, if you factory state the unit dynamic features are lost and must be re-enabled.

For more information, see Adding or restoring an HSM to the Security World.

Ensure you have a quorum of the ACS and that the ACS is available and operating correctly prior to commencing any firmware upgrade. If you do not, you will not be able to reload your Security World on your nShield HSM and you will not be able to use any of your keys.

2.8.3. Upgrading the Connect image

The Connect image (identified by the file extension .nff) is located on the .iso or DVD in the `nethsm-firmware` directory. The image file contains all necessary components required to upgrade the nShield HSM.

Before upgrading, copy the directory containing the `.nff` file from the .iso or DVD to the `nethsm-firmware` directory on the Remote File System of the nShield HSM.

The following sections describe how to load this firmware package onto your nShield HSM.

2.8.4. Upgrading the Connect image using the front panel

Before upgrading your Connect image ensure that you have a working quorum of Administrator Cards from the ACS. You need these together with the files in `/local` to restore your Security World on your nShield HSM after the upgrade.

To upgrade the Connect image:

1. Ensure that the Connect image file is named `nCx3N.nff` and located in the following directory:
   - Windows: `%NFAST_HOME%\nethsm-firmware\<version>\nCx3N.nff`.
   - Linux: `/opt/nfast/nethsm-firmware/<version>/nCx3N.nff`.

     Where `<version>` is a subfolder containing the firmware image to be used for the upgrade. There can be more than one `<version>` subfolder.

The directory `<version>` should match the image version string identified on the firmware ISO. If you are not sure on the details, contact Entrust nShield Support, https://nshieldsupport.entrust.com.

1. From the main menu on the unit, select **System** > **Upgrade system**.
2. Confirm that you want to upgrade the image file.
3. Select the directory that contains the image file or firmware that you require. If multiple Connect image directories are displayed, scroll to the relevant directory and select it.

   You are informed that the files are being transferred. The nShield HSM will disconnect from the network during the upgrade procedure and reconnect once the upgrade is complete.
4. Verify the image version, HSM (firmware) version, and image VSN that are displayed, and confirm the upgrade when prompted.

2.8.5. Upgrading the nShield HSM from a privileged client

The following description assumes the RFS and Client are separate machines which an nShield HSM has already been configured to use. If you are using a combined RFS/Client, then apply the following instructions to the same machine. The Client must have privileged access to the nShield HSM.

The image upgrade file may be supplied as a separate item that must be copied into the subfolder for its respective version. The default file name is `nCx3N.nff`.

1. Ensure that the new image file is in the following folder on the RFS:
   - Windows: `%NFAST_HOME%\nethsm-firmware\<version>`
   - Linux: `/opt/nfast/nethsm-firmware/<version>`

     Where `<version>` is a subfolder containing the image for the respective version. There can be more than one `<version>` subfolder. The string `<version>` should

match the name of the version folder in which the image is located on the version's firmware ISO.

If the <version> subfolder does not already exist on the RFS, it must be created by a user with the necessary privileges.

2. List the image file(s) available on the RFS, run the following command from the Client:

```
>nethsmadmin –m<n> -s <RFS_IP> -l
```

Where:

- <n> is the module number for the target nShield HSM
- <RFS_IP> is the IP address of the RFS.
- Additionally the --rfs-hkneti=<RFS_HKNETI> and --rfs-esn=<RFS_ESN> options can be set to enable secure authentication of the RFS. There are three possible cases:
  - **Without secure authentication**: The authentication of the RFS will be based on the IP address only if the --rfs-hkneti and --rfs-esn options are not specified.
  - **Software-based authentication**: The --rfs-hkneti option specifies the software KNETI hash of the RFS. The --rfs-esn option shall not be specified.

    <RFS_HKNETI> can be obtained by running anonkneti -m0 localhost on the RFS.

  - **nToken authentication**: Only if an nToken (or local HSM) is installed in the RFS. The --rfs-hkneti and --rfs-esn options specify the KNETI hash and ESN of the nToken.

    <RFS_HKNETI> and <RFS_ESN> can be obtained by running ntokenenroll -H on the RFS.

    For example, when the image file is located in the appropriately named <version> folder:

    ```
    >nethsmadmin -m1 -s 194.28.158.146 -l
    Initiating RFS nethsm image check on 194.28.158.146...

    Checking the nethsm-firmware directory on the RFS.
    nethsm-firmware/VersionName/nCx3N.nff
    nethsm-firmware/AnotherVersionName/nCx3N.nff

    Images were successfully found on the RFS (194.28.158.146).
    ```

    For example, if the version folder does not exist or its name is not correct, the

`nethsmadmin` command cannot find the image:

```
>nethsmadmin -m1 -s 194.28.158.146 -l
Initiating RFS nethsm image check on 194.28.158.146...

Checking the nethsm-firmware directory on the RFS.
No images found on the RFS (194.28.158.146).
```

3. In order to load (or upgrade) the Connect image run the following command from the Client:

```
>nethsmadmin –m<n> -s <RFS_IP> --upgrade-image=nethsm-firmware/<selected-image-version>/nCx3N.nff
```

Where:

- `<n>` is the module number for the target nShield HSM
- `<selected-image-version>` specifies the version subfolder on the RFS containing the firmware image you wish to load (upgrade) onto the nShield HSM.
- `<RFS_IP>` specifies the IP address of the RFS using the `-s` argument. For example

```
>nethsmadmin -m1 -s 194.28.158.14 --upgrade-image=nethsm-firmware/VersionName/nCx3N.nff
```

Copy the path to the required image file as provided by the available image list above. (Linux style path separators are used irrespective of whether the Client or RFS are Windows or Linux based).

For example:

```
>nethsmadmin -m1 -s 194.28.158.14 --upgrade-image=nethsm-firmware/VersionName/nCx3N.nff
Initiating appliance image upgrade using file nethsm-firmware/VersionName/nCx3N.nff...
Upgrade operation state changed to: Image Transfer Initiated
Upgrade operation state changed to: Image Transferred
Upgrade operation state changed to: Image Verified
Not able to contact appliance because of reason(23): CrossModule,#1-ExplicitRequest,#2-Mode
Upgrade operation final state: Image Verified
Image upgrade completed.
Please wait for appliance to reboot.
Please wait for approximately half an hour for the appliance to internally upgrade.
```

The following line is expected and requires no action:

```
Not able to contact appliance because of reason(23): CrossModule,#1-ExplicitRequest,#2-Mode
```

The notification appears because the RFS/client cannot contact the nShield HSM. Once the image is copied across, the nShield HSM will disconnect from the

network for the duration of the upgrade and reconnect once the upgrade is completed.

If the nShield HSM suffers a loss of power while you are upgrading the image file or internal module firmware, exit the `nethsmadmin` utility, wait until power is restored to the HSM, then try to restart the process as shown above.

4. After the image upgrade has completed, run the `enquiry` utility to check the image version of the target nShield HSM is as expected.

2.8.5.1. Enabling and disabling remote upgrade

You can enable or disable upgrading an nShield HSM remotely, see `enable_remote_mode` in the server_settings section or the *Top-level menu* chapter of the HSM *Install Guide*. Once you have enabled remote upgrade, you can upgrade an nShield HSM from a computer using the `nethsmadmin` command, without accessing the unit itself.

2.8.6. After firmware installation

After you have installed new firmware and initialized the HSM, you can create a new Security World with the HSM or reinitialize the HSM into an existing Security World.

If you are initializing the HSM into a new Security World, see Create a new Security World.

If you are re-initializing the HSM into an existing Security World, see Adding or restoring an HSM to the Security World.

2.9. Troubleshooting

This guide covers the following HSMs:

- nShield Connect
- nShield 5c

It describes what to do if you have an issue with your HSM, or your Security World Software.

2.9.1. Checking operational status

Use the following methods to check the operational status of the module.

2.9.1.1. Enquiry utility

Run the `enquiry` utility to check that your module is working correctly. The `enquiry` utility is 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 the message:

```
Server:
 enquiry reply flags  none
 enquiry reply level  Six
 serial number        ####-####-####
 mode                 operational
 version              #-#-#
 speed index          ######
 rec. queue           ####..####
 ...
 version serial       #
 remote port (IPv4)   ####

Module ##:
 enquiry reply flags  none
 enquiry reply level  Six
 serial number        ####-####-####
 mode                 operational
 version              #-#-#
 speed index          #####
 rec. queue           ##..###
 ...
 rec. LongJobs queue  ##
 SEE machine type     PowerPCELF
 supported KML types  DSAp1024s160 DSAp3072s256
 hardware status      OK
```

If the output from the `enquiry` utility does not show `mode operational`, you can use the Status LED to discover the status of the module.

2.9.1.2. Status LED

The blue Status LED indicates the operational status of the module.

Status LED	Description
Off.	**Status: Power off or Standby mode** There is either no power supply to the module or the module is in Standby mode. If you suspect that there is no power supply, check that the module is properly connected and switched on. If you believe the module's power supply unit has failed, contact Support.

Status LED	Description
On, occasionally blinks off.	**Status: Operational mode** The module is in Operational mode and accepting commands. The more frequently the Status LED blinks off, the greater the load on the module.
Flashes two short pulses, followed by a short pause.	**Status: Initialization mode** Existing Security World data on the module has been erased. The module is automatically placed in Initialization mode after a Security World is created.
On, occasionally blinks off.	**Status: Operational mode** The module is in Operational mode and accepting commands. The more frequently the Status LED blinks off, the greater the load on the module. **Status: Initialization mode** Existing Security World data on the module has been erased. The module is automatically placed in Initialization mode after a Security World is created.
Flashes two long pulses followed by a pause. (**nShield Connect only**)	**Status: Maintenance mode** Used for reprogramming the module with new firmware. The module only goes into Maintenance mode during a software upgrade.
Blue LED. (**nShield 5c only**)	**Status: Maintenance mode** Used for reprogramming the module with new firmware. The module only goes into Maintenance mode during a software upgrade.

Status LED	Description
Flashes SOS, the Morse code distress code (three short pulses, three long pulses, three short pulses). After flashing SOS, the Status LED flashes a Morse code letter which identifies the error. (**nShield Connect only**)	**Status: Error mode** If the module encounters an unrecoverable error, it enters Error mode. In Error mode, the module does not respond to commands and does not write data to the bus. For internal security modules running firmware 2.6.1.2 and above, the error code is also reported by the `enquiry` utility in the `hardware status field` of the `Module` and under `hardware errors` in the hardserver log. If a command does not complete successfully, the module normally writes an error message to the log file and continues to accept further commands. It does not enter Error mode.
Flashes BIOS code. (**nShield 5c only**)	**Status: Error mode** If the module encounters an unrecoverable error, it enters Error mode. In Error mode, the module does not respond to commands and does not write data to the bus. The error code is also reported by the `enquiry` utility in the `hardware status field` of the `Module` and under `hardware errors` in the hardserver log. If a command does not complete successfully, the module normally writes an error message to the log file and continues to accept further commands. It does not enter Error mode.

2.9.1.3. Audible warning

An audible warning sounds for some critical errors relating to the PSUs on the module. The orange warning LED (see Orange warning LED) accompanies the audible warning.

The warning sounds when only one of the two PSUs is powered and turned on. Check that:

- The rocker switch on both PSUs is in the `on` position.
- Both PSUs are connected to the mains supply.

If the audible warning continues, there might be a fault with one or both PSUs. Before investigating further, switch off the audible alarm by navigating to the **1-2-5-3 Critical Errors** screen. The orange warning LED remains on until you resolve the issue.

For more information about identifying and replacing a failed PSU, see the *nShield Power Supply Unit Installation Sheet* for your HSM.

2.9.1.4. Orange warning LED

If the orange warning LED is on, the module has encountered a critical error (for example, overheating or PSU failure) that may require immediate action. To find the cause of a critical error, navigate to **System information > View h/w diagnostics > Critical Errors**.

2.9.1.5. Checking the physical security of the module

The physical security measures implemented on the module include tamper detection. This warns you of tampering in an operational environment. For more information about tamper detection, including the tamper warning messages, see the *Physical Security Checklist* or the Physical security of the HSM.

2.9.1.6. Display screen

When the module is in Maintenance or Initialization mode, there is a color-coded footer at the bottom of the display screen. There is no footer when the module is in Operational mode.

Footer color	Text in footer	Meaning
Yellow	Initialization	The system is rebooting or waiting for an Administrator Card to be inserted.
Blue	Maintenance	An administrative task is being performed. This mode is only entered during firmware upgrades.
Red	HSM Failed	The internal module has failed. See Orange warning LED for more information.

Do not interrupt power to the module during a firmware upgrade.

The blue Status LED flashes to indicate the status of the internal security module.

2.9.1.7. Power button

The **Power** button, in combination with the display screen, indicates the general status of the module.

The display screen turns off automatically if the front panel buttons are inactive for more than three minutes. Use the touch wheel to turn the display screen back on.

Power button	Display screen	Status
On	On, displaying menus and dialogs	The module is operational.
On	On, displaying messages but not displaying labels for the navigation buttons	The module is running an upgrade. A color-coded footer indicates the specific status: yellow for initialization, red (maintenance) for upgrade.
On, flashes occasionally	On, displaying messages but not displaying labels for the navigation buttons	The module is performing start-up.
Mostly off, flashes occasionally	Off	The module is in Standby mode (that is, it has been powered down from the front panel using the **Power** button). Press the **Power** button to turn it on.
Flashes regularly	On, with "Critical Error" message	The module is unable to start-up or has failed. The error message describes the problem. If you can remedy the problem, do so, and press the **Power** button to restart the module. Otherwise, contact Support.
Flashes irregularly	Off	A low-level critical error has occurred.

2.9.1.8. Ethernet LEDs

There are four Ethernet LEDs, two for each of the two Ethernet ports on the module. The Ethernet LEDs indicate the status of the connection with other Ethernet devices.

Ethernet LEDs	Status
Flashes regularly	The status of the Ethernet link is currently unknown (the Ethernet LEDs flash when the module is powering up).
Off	There is no Ethernet link. The Ethernet cable is either not connected to the module or the cable is not connected to a functioning Ethernet device.
On, green only	Indicates a 10Mb or 100Mb Ethernet link.
On, green and orange	Indicates a 1Gb Ethernet link.

2.9.2. Module overheating

If the internal module of the HSM exceeds the safe operating temperature, the unit stops operating and displays the SOS-T error message on the Status LED. See Status LED for details of the SOS-T error message.

2.9.3. Log messages for the module

To view log messages from the main menu of the module:

1. Select **System > System information**.
2. Select either:
   - **View system log**.
   - **View hardserver log**.

The client can store logs, and can configure them to contain different types of message.

2.9.3.1. 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
```

2.9.3.2. Notice

This type of message is sent for information only:

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

2.9.3.3. 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
```

2.9.3.4. 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.

2.9.3.5. 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.

2.9.3.6. 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 Security World software, see Install the Security World software. If reinstallation does not solve the problem, contact Support.

2.9.3.7. 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.

2.9.4. Utility error messages

This type of message might indicate an error status when you run a command line utility.

2.9.4.1. BadTokenData error in nShield modules

Some nShield modules are equipped with a rechargeable backup battery for maintaining Real Time Clock (RTC) operation when the module is powered down. This battery normally lasts for up to two weeks if no power is supplied to the HSM.

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 this case is to leave the HSM powered up for at least 10 hours to recharge the battery, and then reset the clock. No other nonvolatile data is lost when this occurs.

2.9.5. Storage error

This type of message might 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. This type of error can be fixed using the nShield Log Service. This service runs in the background and will automatically retrieves and removes audit logs from the HSM. To learn more about this service, see nShield Audit Log Service

2.10. HSM maintenance

This guide covers the following HSMs:

- nShield Connect
- nShield 5c

The HSMs contain only two user-replaceable parts:

- The PSUs.
- The fan tray module.

Replacing a PSU or fan tray module does not affect FIPS 140 validations for the HSM, or result in a tamper event. However, in the very rare event that a PSU or fan tray module requires replacement, contact Support before carrying out the replacement procedure.

Do not allow a fan tray to be removed from the HSM for longer than 30 minutes, otherwise a tamper event will occur.

For more information about replacing either a PSU or the fan tray module, see the Installation Sheet that accompanies the replacement part.

Breaking the security seal or dismantling the HSM voids your warranty cover, and any existing maintenance and support agreements.

2.10.1. Flash testing the module

The module is designed to comply with IEC/EN/UL 62368-1 but should be tested only by trained safety professionals. Because the module is fitted with radio frequency interference suppressors, it is recommended that only a DC test be performed.

Repeated application of the flash test can damage safety insulation.

2.11. Approved accessories

This guide covers the following HSMs:

- nShield Connect
- nShield 5c

The following parts can be ordered with the HSM or separately.

Part	Part number	Comments
Slide rail assembly	AC2050	Optional slide rail assembly and fixing kit. For details of contents, see the *nShield Connect and nShield 5c Slide Rails Instructions*.
USB keyboard	M-030099-L	For more information about using a USB keyboard with the HSM, see Connecting the optional USB keyboard.
Replacement fan tray module	AC2064	Includes installation instructions.
Replacement PSU	AC2057	Includes installation instructions.

If you have an enquiry about any of the parts listed, contact Support.

2.12. Resource Watchdog

The Resource Watchdog monitors useful information such as CPU usage, ethernet interface states and addresses, and reports it in the syslog of the Connect. The Watchdog is entirely configurable with its own bespoke configuration file.

It is possible to redirect the syslog of the Connect to either the RFS or a client, for more information, see syslog.

2.12.1. Enabling or disabling the Watchdog

The Resource Watchdog is disabled by default and starts automatically at Connect boot. To disable it, or to re-enable it, go in the FPUI to Menu **System** > **System Configuration** > **Watchdog Config** > **Enable watchdog** and select the appropriate setting (**ENABLE** or **DISABLE**).

Connect needs to reboot before Enable or Disable setting takes place.

You can also modify the `[nethsm_watchdog]` section in the Connect Configuration File by setting **enable_watchdog** to **yes** or **no**. Then, push it to the Connect as usual using cfg-pushnethsm.

The watchdog automatically applies the new setting after a cycle (usually up to a minute).

2.12.2. Understanding the default settings

The default Watchdog Configuration File is as follows:

```
Network Devices:
- eth0
- eth1
- nshield0
Monitored Processes:
- hardserver
- netui
- cosmod
System Information Report:
  enabled: False
  interval: 1h
CPU Usage Monitoring:
  threshold: 180%
  interval: 60s
  frequency: 1h
```

2.12.2.1. Network Devices

The Watchdog monitors when the ethernet interfaces listed in its configuration file change state. By default, it monitors **eth0**, **eth1** and **nshield0** (only available on nShield 5c). You can add or remove ethernet interfaces from the configuration file.

A typical report looks like this:

```
Mar 13 10:47:50 nethsm nfwatchdog: Interface eth0 is now up.
```

```
Mar 13 10:47:50 nethsm nfwatchdog: Interface eth1 is now down.
```

The Watchdog also monitors address changes on these interfaces. A typical report looks like this:

```
Mar 13 10:47:50 nethsm nfwatchdog: The addresses of interface eth0 have been set to ["00:60:e0:87:a1:59",
"172.23.135.129"].
Mar 13 13:58:25 nethsm nfwatchdog: The addresses of interface eth0 have changed from ["00:60:e0:87:a1:59",
"172.23.135.8"] to ["00:60:e0:87:a1:59", "172.23.135.129"].
```

2.12.2.2. Monitored Processes

The Watchdog monitors when the processes listed in its configuration file start and stop. By default, it monitors **harderver**, **netui** and **cosmod**. You can add or remove processes from the configuration file. Please provide only the process name, do not include the path.

A typical report looks like this:

```
Mar 13 10:47:50 nethsm nfwatchdog: Process netui is not running.
Mar 13 10:47:50 nethsm nfwatchdog: Process hardserver is running: {"arguments": ["../sbin/hardserver", "-
Llogfile"], "children": [{"arguments": ["/opt/nfast/python3/bin/python3", "-uIBm", "nshield.entrypoint", "--",
"/opt/nfast/bin/hsc_servicehosts"], "children": [], "name": "python3", "pid": 607, "status": "running", "uptime":
"0:00:01"}], "name": "hardserver", "pid": 598, "status": "sleeping", "uptime": "0:00:01"}
```

If the process is running, the Watchdog reports, in JSON format, the process name, arguments, PID, uptime, status and children, if any.

2.12.2.3. System Information Report

The Watchdog can report System Information. By default, the report is disabled. To enable it, set **System Information Report** > **enabled** to **yes**.

By default, the System Information Report is printed every hour (if enabled). To change the interval, set **System Information Report** > **interval** to the desired interval. This field can be expressed in seconds, minutes, hours, or days. Examples: 8h, 2d8h5m20s, 2m4s. A unit is required.

A typical report looks like this:

```
Mar 13 14:05:32 nethsm nfwatchdog: System Report for the Connect is {"system_uptime": "3:20:13.930539",
"virtual_memory": {"total": 8253513728, "free": 7962390528, "available": 7996977152, "used": 173985792},
"load_avg": [0.09, 0.04, 0.01]}
```

It contains the following information in JSON format: * system up time * virtual memory in bytes (total, free and available, as defined in psutil) * system load over the last 1, 5 and 15

minutes, as defined in psutil. * The “load” represents the processes which are in a runnable state, either using the CPU or waiting to use the CPU (for example, waiting for disk I/O).

Field	Value	Comment
enabled	True or False, default: False.	Whether the System Information Report is enabled.
interval	An interval in seconds, minutes, hours or days, default: 1h.	The interval at which the System Information Report is printed, if enabled. A unit (s, m, h or d) is required.

2.12.2.4. CPU Usage Monitoring

The Watchdog monitors both the global CPU usage on the Connect and the CPU usage per process. Should the total CPU usage or the CPU usage of any process (as computed with psutil’s cpu_percent function) be above the threshold (default: 180%) for a sustained interval (default: 60s), then the Watchdog will report it. This check is performed every hour by default (**CPU Usage Monitoring** > **frequency**).

A typical report looks like this:

```
Mar 13 10:48:44 nethsm nfwatchdog: Global CPU usage for the Connect is 92.1%, higher than the threshold of 90%.
Mar 13 10:48:44 nethsm nfwatchdog: CPU usage for the following processes is higher than the threshold of 90%.
Mar 13 10:48:44 nethsm nfwatchdog: Process hardserver, 90.5%: {"arguments": ["../sbin/hardserver", "-Llogfile"],
"children": [{"arguments": ["../sbin/hardserver", "--spawn-svc"], "children": [], "name": "hardserver", "pid":
620, "status": "sleeping", "uptime": "0:00:54"} {"arguments": ["/opt/nfast/bin/ncssh", "-id",
"/opt/nfast/services/client/ncoreapi/ssh/id_ecdsa", "-known-hosts", "/opt/nfast/services/module/5CA8-5A32-
80F2/ncoreapi/known_hosts", "-hosts", "/opt/nfast/services/module/hosts.txt", "-hostname", "nshield-5CA8-5A32-
80F2.local", "-port", "2203", "-user", "ncoreapi", "ncoreapi", "operational"], "children": [], "name": "ncssh",
"pid": 626, "status": "sleeping", "uptime": "0:00:54"}], "name": "hardserver", "pid": 598, "status": "sleeping",
"uptime": "0:00:55"}
```

For each process, the Watchdog reports the process name, the CPU usage as reported by psutil’s cpu_percent, and in JSON format: the process name, arguments, PID, children, status and uptime.

The threshold must be a number between 0% and 100% x (number of CPUs on the Connect). The % sign is optional. The default is 180%.

The interval is the time during which the Watchdog samples the CPU usage. This interval is blocking, that is to say that the Watchdog is not able to perform any other monitoring while the sampling is happening. The default is 60s. This field can be expressed in seconds, minutes, hours, or days. Examples: 8h, 2d8h5m20s, 2m4s. A unit is required.

The frequency sets how often the Watchdog performs the sampling. By default, the Watchdog performs the CPU usage sampling for 60s every hour (frequency = 1h). This field can be expressed in seconds, minutes, hours, or days. Examples: 8h, 2d8h5m20s, 2m4s. A unit is required.

Field	Value	Comment
threshold	A percentage between 0% and 100% x number of CPU, default: 180%.	Threshold above which global CPU and process CPU usage is reported. The % sign is optional.
interval	An interval in seconds, minutes, hours or days, default: 60s.	The interval during which the Watchdog computed the CPU usage (both total and per process). A unit (s, m, h or d) is required.
frequency	A frequency in seconds, minutes, hours or days, default: 1h.	The frequency at which the Watchdog monitors the CPU usage. A unit (s, m, h or d) is required.

2.12.2.5. Other features

2.12.2.5.1. Zombie processes

On top of the aforementioned configurable features, the Watchdog also monitors zombie processes, or processes that are waiting for their parent.

A typical report looks like this:

```
Aug  7 03:47:46 nethsm nfwatchdog: Information: Process netui exited, awaiting parent.
Aug  7 03:47:51 nethsm nfwatchdog: Information: 3 processes (netui, cosmod, hardserver) exited, awaiting parent.
```

2.12.2.5.2. Report filtering

Should any Watchdog event occur several times in a row, the reports will be filtered out, so as not to flood the log. This will be reported by the Watchdog in the following way:

```
May  9 09:35:47 nethsm nfwatchdog: Global CPU usage for the Connect is 200.0%, higher than the threshold of
180.0%.
```

2.12.3. Reading or modifying the Watchdog Configuration File

1. Create a Watchdog Configuration file called `nfwatchdog.yml.new` on your client with your changes.
2. Prepare to push the new configuration file. Perform one of the following:
   - On the Connect UI:
     i. On Menu **System** > **System Configuration** > **Watchdog Config** > **Config push mode**, set **push** to **on**.
     ii. On Menu **System** > **System Configuration** > **Watchdog Config** > **Client address**, set the address of your client.

- Update the Connect Configuration File directly:
  i. Open the file and locate the `[nethsm_watchdog]` section.
  ii. Set **push** to **on**.
  iii. Set **remote_ip** to the IP of your client.
  iv. (Optionally) Set **remote_keyhash** as well. For more information, see config_op.

3. Push the Connect Configuration File as usual, for example using cfg-pushnethsm.
4. Use the following command to send your new Watchdog Configuration File to the Connect:

```
sudo nfcp ./nfwatchdog.yml.new <connect_ip>:cfg-nfwatchdog
```

The Connect will pick up the new configuration and the Watchdog will apply it after a cycle (usually up to 1 minute). The syslog should show:

```
Feb 10 13:22:36 nethsm ../sbin/config-update: found new pushed watchdog config, attempting to install
Feb 10 13:22:36 nethsm ../sbin/config-update: successfully installed new watchdog config
```

The Connect will then push the current configuration back to `/opt/nfast/kmdata/hsm-<esn>/config` on your client, alongside the Connect Configuration File, if you have enabled `[config_op]` push.

2.12.3.1. Configuring via hardserver configuration file

The Watchdog can be configured using the hardserver configuration file, in the following steps.

1. Set the Watchdog entries in the hardserver configuration file.

```
[nethsm_watchdog]
# Start of the nethsm_watchdog section
# Connect Watchdog configuration. This section allows you to enable or disable
# the Connect Watchdog and set up config file push.
# Each entry has the following fields:
#
# Enable the Connect watchdog. (default=no)
#  enable_watchdog=ENUM
#
# Whether to allow a client to push new watchdog config files to the netHSM.
# If "on" then this effectively allows a client to remotely configure the
# nethsm_watchdog. (default=off)
#  push=ENUM
#
# The IP address of the client allowed to push watchdog config files. If not
# set, or set to 0.0.0.0 or ::, allows ANY IP address to push on a new config
# file.
#  remote_ip=ADDR
```

```
#
# The hash of the key that the authorised client should use to authenticate
# itself, or 40 zeros to indicate no key authentication required. (Default is
# 40 zeros).
#  remote_keyhash=KEYHASH
```

`enable_watchdog`	This section allows you to enable or disable the Connect Watchdog and set up config file push. The default is 'no'.
`push`	Whether to allow a client to push new Watchdog config files to the netHSM. If this is "on", it allows a client to remotely configure the nethsm_watchdog. The default is 'off'.
`remote_ip`	The IP address of the client allowed to push Watchdog config files. If this is not set, or set to 0.0.0.0 or ::, it allows any IP address to push a new config file.
`remote_keyhash`	The hash of the key that the authorised client should use to authenticate itself, or 40 zeros to indicate no key authentication required. The default is 40 zeros.

2. Restart the hardserver to load the updated configuration file.
3. Reboot the connect if enabling or disabling the Watchdog.

2.12.4. Troubleshooting

The Watchdog is designed to fall back to its default values in case the Watchdog Configuration File is missing or malformed. If the Watchdog does not behave as you intended, examine the syslog for clues.

Factory stating the Connect makes the Watchdog Configuration File revert to its defaults.

Error message	Likely cause	Solution
Could not open Watchdog Configuration File. Watchdog Configuration File has been restored to its default state.	Watchdog Configuration File is missing or contains a YAML syntax error.	Push a new Watchdog Configuration File.
Watchdog Configuration File was unexpectedly empty. Watchdog Configuration File has been restored to its default state.	Watchdog Configuration File is missing or empty.	Push a new Watchdog Configuration File.
Could not read missing ... field. Using default ... instead.	The field is missing or there is a typo.	Update the Watchdog Configuration File and push it again.

Error message	Likely cause	Solution
Ignoring unrecognized category ... from Watchdog Configuration File. Valid categories are ...	There is a typo in the category or it is out-of-date.	Update the Watchdog Configuration File and push it again.
Ignoring unrecognized subcategory ... from Watchdog Configuration File. Valid subcategories are ...	There is a typo in the subcategory or it is out-of-date.	Update the Watchdog Configuration File and push it again.
Could not parse any time information from ... in the Watchdog Configuration File. Examples of valid strings: ... Could not use ... as .... Using default ... instead.	The string is not a valid time.	Update the Watchdog Configuration File and push it again.
Could not parse 'CPU Usage Monitoring/threshold': must be a number, not ...	The threshold provided is not a valid number.	Update the Watchdog Configuration File and push it again.
Could not parse 'CPU Usage Monitoring/threshold': must be between 0% and 200%, not ...	The threshold provided is not a number.	Update the Watchdog Configuration File and push it again.

The intervals and frequencies at which the Watchdog works cannot be enforced. That is, if the Watchdog is computing CPU usage and is due to compute the System Information Report, it will do so as soon as it is available.

2.13. Valid IPv6 Addresses

This guide covers the following HSMs:

- nShield Connect
- nShield 5c

The following table provides a list of valid IPv6 addresses for each of the types of addresses recognized by certain parts of the system. For information on setting up IPv6 addresses, see Configuring the Ethernet interfaces - IPv4 and IPv6 .

Address type	Address Range (inclusive) (From | To)		Example
Unspecified	::	::	::
Loopback	::1	::1	::1
Local Unicast	fc00::	fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff	fdf8:f53b:82e4::53

Address type	Address Range (inclusive) (From | To)		Example
Link-local	fe80::	febf:ffff:ffff:ffff:ffff:ffff:ffff:ffff	fe80::200:5aee:feaa:20a2
Site-local (depreciated)	fec0::	feff:ffff:ffff:ffff:ffff:ffff:ffff:ffff	fec0::100:abc:22
Teredo	2001::	2001:0:ffff:ffff:ffff:ffff:ffff:ffff	2001:0:4136:e378:8000:63bf:3fff:fdd2
Benchmarking	2001:2::	2001:2:0:ffff:ffff:ffff:ffff:ffff	2001:2:0:6c::430
Orchid	2001:10::	2001:1f:ffff:ffff:ffff:ffff:ffff:ffff	2001:10:240:ab::a
6to4	2002::	2002:ffff:ffff:ffff:ffff:ffff:ffff:ffff	2002:cb0a:3cdd:1::1
Documentation	2001:db8::	2001:db8:ffff:ffff:ffff:ffff:ffff:ffff	2001:db8:8:4::2
Global Unicast	2000::	3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff	20ab:45:fa::adb5
Multicast	ff00::	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff	ff01::2

The available addresses in the Global Unicast range are not contiguous.

2.14. Remote File System Volumes

The hardserver (**Linux**) or nFast Server (**Windows**) service restricts the paths that can be shared as RFS (Remote File System) volumes using the `remote_file_system` section of the `config` file or using the `rserverperm --accessfiles` command-line configuration.

By default, the following paths are permitted:

- `/opt/nfast/kmdata` (**Linux**).
- `%NFAST_KMDATA%`, typically `C:\ProgramData\nCipher\Key Management Data` (**Windows**).
- Any path that was created by the `rfs-setup` utility and associated with RFS volumes to prepare an RFS for an nShield HSM or for use with the `rfs-sync` utility.
- Subdirectories of permitted paths.

If you want to add custom paths not included in this list as RFS volumes, you must add them to the list of permitted paths before starting the hardserver (**Linux**) or nFast Server (**Windows**) service. If you make these changes after starting the service, you need to

restart it for the changes to take effect.

You can update the list of permitted paths by either setting the NFSERV_RFS_ALLOWED_PATHS environment variable (see Allow custom RFS paths with an environment variable) or by creating an additional `config.secure` configuration file (see Allow custom RFS paths with a configuration file.)

2.14.1. Allow custom RFS paths with an environment variable

Linux

1. If the `/etc/nfast.conf` file does not already exist, create it.

   This file must only be writable by root. This is enforced by nShield start-up scripts.

2. Add the `NFSERV_RFS_ALLOWED_PATHS` environment variable to the `nfast.conf` file with a colon-separated list of paths (`/<path>/share`).

   For example, to share `path1` and `path 2` (spaces are permitted):

   ```
   export NFSERV_RFS_ALLOWED_PATHS=/path1/share:/path 2/share
   ```

Windows

Create the `NFSERV_RFS_ALLOWED_PATHS` environment variable in the global system environment variables with a semicolon-separated list of paths (`\<path>\share`).

For example, to share `path1` and `path 2` (spaces are permitted):

```
C:\path1\share;D:\path 2\share
```

2.14.2. Allow custom RFS paths with a configuration file

1. Create the `config.secure` in `/opt/nfast/hardserver.d` (**Linux**) or the `%PROGRAMDATA%\nCipher\hardserver.d` directory, which is typically `C:\ProgramData\nCipher\hardserver.d\config.secure`.
2. Add the paths as values in an `rfs_allowed_paths` JSON array. The JSON must be valid.

   For example, to share `path1` and `path 2` (spaces are permitted):

   Linux

   ```
   {
     "rfs_allowed_paths" : ["/path1/share", "/path 2/share"]
   }
   ```

Windows

```
{
  "rfs_allowed_paths" : ["C:\\path1\\share", "D:\\path 2\\share"]
}
```

You must use a backslash (\) to escape the backslashes in the path.

3. nShield PCIe HSMs

3.1. Client software and module configuration: PCIe and USB HSMs

This chapter describes software and module configuration tasks that you can choose to perform after the initial installation of Security World Software and hardware.

You must determine whether particular configuration options are necessary or appropriate for your installation. The additional configuration options described in this chapter can be performed either before or after the creation of a Security World (as described in Create a new Security World) and an OCS (as described in Creating Operator Card Sets (OCSs)).

3.1.1. About user privileges

Cryptographic security does not depend on controlling user privileges or access but maintaining the integrity of your system from both deliberate or accidental acts can be enhanced by appropriate use of (OS) user privileges.

3.1.2. Set up client cooperation

You can allow an nShield HSM to automatically access the remote file system (RFS) belonging to another nShield HSM and share the Security World and key data stored in the Key Management Data directory. Client hardware security modules that access data in this way are described as *cooperating clients*.

To configure client cooperation for other clients or hardware security modules that are not nShield HSMs, see Client cooperation.

3.1.2.1. Useful utilities

- anonkneti
- rfs-sync

3.1.2.2. Setting environmental variables

This section describes how to set Security World Software-specific environment variables. You can find detailed information about the environment variables used by Security World Software in Environment variables.

Linux

You can set Security World Software-specific environment variables in the file `/etc/nfast.conf`. This file is not created by the installation process: you must create it yourself. `/etc/nfast.conf` is executed by the start-up scripts of nShield HSM services as the root user. This file should only contain shell commands that can safely be run in this context. `/etc/nfast.conf` should be created with access permissions that allow only the root user to write to the file.

Ensure that all variables are exported as well as set.

Windows

You can set Security World Software-specific environment variables as follows:

1. Open the **System** dialog by clicking **System** in the control panel menu.
2. Select the **Advanced** tab and click the **Environment Variables** button.
3. To add a variable, click **New**. Alternatively, to edit an existing variable select an entry in the **System Variables** list and click **Edit**.
4. In the **Variable Name** field, type or edit the name of the environment variable (for example, **NFAST_HOME**).
5. In the **Variable Value** field, type or edit the value to use.
6. Click the **OK** button to set the value, and then click the **OK** button to close the dialog.
7. Open the **Administrative Tools** dialog by clicking the **Administrative Tools** icon in the Control Panel
8. Open the **Services** console by clicking the **Services** icon.
9. From the displayed list of services, select the **nFast Server** icon, and select **Restart the service**.

3.1.2.3. Logging and debugging

Network-attached HSMs: You can view logs generated by the nShield HSM and applications that use it on the unit front panel. Application log messages are handled on the client.

The Security World Software generates logging information that is configured through a set of four environment variables:

- `NFLOG_FILE`
- `NFLOG_SEVERITY`

- `NFLOG_DETAIL`
- `NFLOG_CATEGORIES`

If none of these logging environment variables are set, the default behavior is to log nothing, unless this is overridden by any individual library. If any of the four logging variables are set, all unset variables are given default values.

Detailed information about controlling logging information by means of these environment variables is supplied in Logging, debugging, and diagnostics.

Some components of the Security World Software generate separate debugging information which you can manage differently. On network-attached HSMs, debugging information for applications is handled on the client.

If you are setting up the unit or the client to develop software that uses it, you should configure debugging before commencing software development.

3.1.2.4. Configuring Java support for KeySafe

To use KeySafe, follow the instructions in Using KeySafe.

3.1.3. Configuring the hardserver

The hardserver handles secure transactions between the HSMs connected to the host computer and applications that run on the host computer. In addition, the hardserver, for example:

- Controls any Remote Operator slots that the HSM uses
- Loads any SEE (Secure Execution Engine) machines that are to run on the HSM
- Enables Remote Administration and provides the communication channel between the Remote Administration Service and the HSM

The hardserver can handle transactions for multiple HSMs. This does not require configuration of the hardserver. For more information, see Using multiple modules.

The hardserver must be configured to control:

- The way the hardserver communicates with remote HSMs
- The way the hardserver communicates with local HSMs
- The import and export of Remote Operator slots

- The loading of SEE machines on to the HSM when the hardserver starts up
- The number of Dynamic Slots available on the HSM
- The port used to connect to the Remote Administration Service
- Whether a Dynamic Slot needs to be exchanged with slot 0 of an HSM
- Timeout values for nShield Remote Administration Card presence assurance
- Configuring the audit logging destination.

The hardserver configuration file defines the configuration of the hardserver. By default, it is stored in `/opt/nfast/kmdata/config/` (**Linux**) or `%NFAST_KMDATA%\config` (**Windows**), and a default version of this file is created when the Security World Software is installed. See Overview of hardserver configuration file sections for an overview of the hardserver configuration file, and see "Hardserver configuration files" for detailed information about the various options available through it.

In some previous releases of the Security World Software, hardserver configuration was controlled by environment variables. The use of these variables has been deprecated. If any of these environment variables are still set, they override the settings in the configuration file.

You must load the configuration file for the changes to the configuration to take effect.

To configure the hardserver, follow these steps:

1. Save a copy of the configuration file, `/opt/nfast/kmdata/config/` (**Linux**) or `%NFAST_KMDATA%\config\config` (**Windows**), so that the configuration can be restored if necessary.
2. Edit the configuration file to contain the required configuration. (See "Hardserver configuration files" for descriptions of the options in the configuration file.)
3. Run cfg-reread to load the new configuration.

   If you changed the server_startup section of the hardserver configuration file, you must restart the hardserver instead of running cfg-reread. For more information, see Stopping and restarting the hardserver.

4. Test that the hardserver is configured correctly by running the `enquiry` command-line utility.

   Check that an HSM with the correct characteristics appears in the output.

5. Test that the client has access to the Security World data by running the `nfkminfo` command-line utility.

Check that an HSM with the correct ESN appears in the output and has the state `0x2 Usable`.

3.1.3.1. Overview of hardserver configuration file sections

3.1.3.1.1. Configure remote HSM connections

You configure the hardserver's communications with remote HSMs in the `server_remotecomms` section of the hardserver configuration file. This section defines the port on which the hardserver listens for communications from remote HSMs. You need to edit this section only if the default port (9004) is not available.

For detailed descriptions of the options in this section, see server_remotecomms.

For information about configuring the Remote Operator feature (Remote Operator slots), as opposed to remote HSMs, see Remote Operator.

3.1.3.1.2. Hardserver settings

You configure the hardserver's settings in the `server_settings` section of the configuration file.

This section defines how connections and hardserver logging are handled. These settings can be changed while the hardserver is running.

For detailed descriptions of the options in this section, see server_settings.

3.1.3.1.3. Hardserver performance settings

You configure the hardserver performance settings in the server_performance section of the configuration file.

This section determines whether multi-threaded performance scaling is enabled or not. By default, scaling is not enabled. Any changes you make to the settings in this section do not take effect until after you restart the hardserver.

For detailed descriptions of the options in this section, see server_performance.

3.1.3.1.4. HSM settings

You configure the HSM's settings in the `module_settings` section of the configuration file.

This section defines the settings for the HSM that can be changed while the hardserver is

running.

For detailed descriptions of the options in this section, see module_settings.

3.1.3.1.5. Hardserver start-up settings

You configure the hardserver's start-up settings in the `server_startup` section of the configuration file.

This section defines the sockets and ports used by the hardserver. You need to change this section only if the default ports for privileged or unprivileged connections (9000 and 9001) are not available.

Windows only

You should use the `nt_privpipe_users` option to define the name of the user who is allowed to carry out privileged operations, for example, using the `nopclearfail` utility. See nt_privpipe_users for more information.

For detailed descriptions of the options in this section, see server_startup.

3.1.3.1.6. SEE machines

You configure the hardserver to load SEE machines on start-up in the `load_seemachine` section of the configuration file. The SEE Activation feature must be enabled on the HSM, as described in Optional features.

This section defines the SEE machines and optional user data to be loaded, as well any other applications to be run in order to initialize the machine after it is loaded.

For detailed descriptions of the options in this section, see load_seemachine.

For information about SEE machines, see CodeSafe applications.

3.1.3.1.7. Remote Operator slots

You configure Remote Operator slots in the `slot_imports` and `slot_exports` sections of the configuration file. These sections define the slots that are imported to or exported from the HSM. This applies to the Remote Operator feature only.

For detailed descriptions of the options in these sections, see slot_imports and slot_exports.

The Remote Operator feature must be enabled on the HSM, as described in Optional features.

3.1.3.1.8. Remote file system

Each client's remote file system is defined separately in the `remote_file_system` section of the configuration file with a list of HSMs that are allowed to access the file system on the given client. For information about setting up client cooperation, see Client cooperation.

The `remote_file_system` section is updated automatically when the `rfs-setup` utility is run. Do not edit the `remote_file_system` section manually.

As a reference, for detailed descriptions of the options in this section, see remote_file_system.

3.1.3.1.9. Audit logging

You configure the hardserver's audit logging in the `auditlog_settings` section of the configuration file.

This section defines the host IP address and port used as the destination for the syslog output of the audit logging capability. Optionally, the audit logging messages can be copied to the hardserver's log file.

For further details see Audit Logging.

The hardserver needs to be restarted for these settings to take effect.

3.1.3.2. Using multiple modules

The hardserver can communicate with multiple modules connected to the host. By default, the server accepts requests from applications and submits each request to the first available module. The server can share load across buses, which includes the ability to share load across more than one module.

If your application is multi-threaded, you can add additional modules and expect performance to increase proportionally until you reach the point where cryptography no longer forms a bottleneck in the system.

3.1.3.2.1. Identifying modules

Modules are identified in two ways:

- By serial number
- By `ModuleID`.

You can obtain the `ModuleID` 's and serial numbers of all your modules by running the `enquiry` command-line utility.

3.1.3.2.2. Electronic Serial Number (ESN)

The serial number is a unique 12-digit number that is permanently encoded into each module. Quote this number in any correspondence with Support.

ModuleID

The `ModuleID` is an integer assigned to the module by the server when it starts. The first module it finds is given a `ModuleID` of 1, the next is given a `ModuleID` of 2, and this pattern of assigning `ModuleID` numbers continues for additional modules.

The order in which buses are searched and the order of modules on a bus depends on the exact configuration of the host. If you add or remove a module, this can change the allocation of ModuleIDs to all the modules on your system.

You can use the `enquiry` command-line utility to identify the PCI bus and slot number associated with a module.

All commands sent to nShield modules require a `ModuleID`. Many Security World Software commands, including all acceleration-only commands, can be called with a `ModuleID` of 0. Such a call causes the hardserver to send the command to the first available module. If you purchased a developer kit, you can refer to the developer documentation for information about the commands that are available on nShield modules.

In general, the hardserver determines which modules can perform a given command. If no module contains all the objects that are referred to in a given command, the server returns an error status.

However, some key-management operations must be performed together on the same module. In such cases, your application must specify the `ModuleID`.

To be able to share OCSs and keys between modules, the modules must be in the same Security World.

3.1.3.2.3. Adding a module

If you have a module installed, you can add further modules without reinstalling the server software.

However, we recommend that you always upgrade to the latest server software and upgrade the firmware in existing modules to the latest firmware.

1. Install the module hardware.
2. (**Linux**) Run the script `/opt/nfast/sbin/install`.
3. Add the module to the Security World. Refer to Adding or restoring an HSM to the Security World.

3.1.3.2.4. Module fail-over

The Security World Software supports fail-over: if a module fails, its processing can be transferred automatically to another module provided the necessary keys have been loaded. Depending on the mode of failure, however, the underlying bus and operating system may not be able to recover and continue operating with the remaining devices.

To maximize uptime, we recommend that you fit any additional nShield modules for failover on a bus that is physically separate from that of the primary modules.

3.1.4. Stopping and restarting the hardserver

If necessary, you can stop the hardserver on the client, and where applicable the Remote Administration Service, by running the following command. On Windows, this must be a command window with administrative privileges.

Linux

```
/opt/nfast/sbin/init.d-ncipher stop
```

Windows

```
net stop "nfast server"
```

If the Remote Administration Service is running, you will be warned and given the option of continuing or not.

Similarly, you can start the hardserver on the client, and where applicable the Remote Administration Service, by running the following command. On Windows, this must be a command window with administrative privileges.

Linux

```
/opt/nfast/sbin/init.d-ncipher start
```

You can also restart the hardserver on the client, and where applicable the Remote Administration Service, by running the following command:

```
/opt/nfast/sbin/init.d-ncipher restart
```

Windows

```
net start "nfast server"
net start "nfast Remote Administration Service"
```

On Windows, you can also stop, start, or restart the hardserver, and where applicable the Remote Administration Service, from the Windows Control Panel:

1. From the Windows Start menu, open the Windows Control Panel.
2. Double-click Administrative Tools.
3. Double-click Services.
4. Locate **nFast Server** or **nFast Remote Administration Service** in the list of services, and from the **Action** menu, select **Stop**, **Start**, or **Restart** as required.

The **nFast Remote Administration Service**, where applicable, is dependent on the **nFast Server** so should be started or restarted after the **nFast Server**.

3.2. Hardserver configuration files

This guide applies to both PCIe and USB HSMs.

The default location of the hardserver configuration file is `/opt/nfast/kmdata/config/config` (**Linux**) or `%NFAST_KMDATA%\config\config` (**Windows**).

The hardserver configuration file has the following sections that you can update to configure the hardserver on an nShield module. If a section is not present, it is assumed to have no entries.

3.2.1. Hardserver configuration files

Hardserver configuration files are text files. They must contain only characters with ASCII values between 32 and 127, and the tab, line break, and return characters.

Lines starting with a # character are comments and are ignored. Some comments that document the configuration options are generated by the configuration process. You can add your own comments, but in some cases they may later be overwritten.

A hardserver configuration file begins with a single line that specifies the version of the file syntax. This syntax-version line has the format:

```
syntax-version=n
```

In this syntax-version line example, *n* represents the version of the syntax in which the file is written. The system can process a file with a lower syntax version than the one it uses, but not one with a higher version.

After the syntax-version line, the rest of the configuration file consists of sections that can be edited to control different aspects of hardserver behavior. Each section begins with its name in square brackets, as in this example:

```
[slot_imports]
```

You can update the parameters defined in most of these sections to configure the way that the hardserver handles secure transactions between modules connected to the host computer and applications that run on the host computer.

Some sections are updated automatically and should not be edited manually. For more information, see the descriptions of individual sections.

In each section, the bracketed name is followed by a specified set of fields. Each field is on a separate line. Each field begins with its name, followed by an equals sign (=) and a value of the appropriate type. White space can be included at either end of the line (for example, in order to indent lines as an aid to clarity).

Some types of field are grouped into entries. An entry is a set of fields of different types that define an instance of an object (for example, a particular client as distinct from other clients). Entries in the same section are separated by a line that contains one or more hyphens (-). Blank lines and comments are allowed between the fields in an entry.

Strings are case sensitive in the section names and field names.

If a particular section is not present in the configuration file, it is assumed to have no entries.

3.2.2. General hardserver configuration settings

3.2.2.1. server_settings

The `server_settings` section defines the settings for the client hardserver you can modify while the hardserver is running.

These flags are used by the `NFLOG_DETAIL` environment variable (see Environment variables to control logging).

The section contains the following fields:

Field	Description
`loglevel`	This field specifies the level of logging performed by the hardserver. See hardserver loglevel and Logging, debugging, and diagnostics.
`logdetail`	This field specifies the level of detail logged by the hardserver. You can supply one or more flags in a space-separated list. For more information about the flags, see the table below.
`connect_retry`	This field specifies the number of seconds to wait before retrying a remote connection to a client hardserver. The default is 10.
`connect_maxqueue`	This field specifies the maximum number of jobs which can be queued on the hardserver. The default is 4096: this is also the maximum value. Setting `connect_maxqueue` to a high value allows high throughput, but may cause long latency if the hardserver goes down.
`connect_broken`	This field specifies the number of seconds of inactivity allowed before a connection to a client hardserver is declared broken. The default is 90.
`connect_keepalive`	This field specifies the number of seconds between `keepalive` packets for remote connections to a client hardserver. The default is 10.
`accept_keepidle`	This field specifies the number of seconds before the first keepalive packet for remote incoming connections. The default is 30. Ideally, `accept_keepalive` should be at least twice the value of the `connect_keepalive` setting on the unattended machines.
`accept_keepalive`	This field specifies the number of seconds between keepalive packets for remote incoming connections. The socket will be closed after up to ten consecutive probe failures. The default is 10. Ideally, `accept_keepalive` should be a value such that (10 * `accept_keepalive`) > `connect_broken` on the unattended machine. Using the default values for both these fields will fulfil this requirement.

Field	Description
connect_command_block	When the module has failed, this field specifies the number of seconds the hardserver should wait before failing commands directed to that module with a NetworkError message. For commands to have a chance of succeeding after the module has failed this value should be greater than that of connect_retry. If it is set to 0, commands to a module are failed with NetworkError immediately, as soon as the module. The default is 35.
max_pci_if_vers	This field specifies the maximum PCI interface version number. If max_pci_if_vers is set to 0 (the default), there is no limit.
enable_remote_mode	If this field is set to yes (the default value) in the module configuration file, nShield HSM mode changing using the nopclearfail utility is enabled. If set to no, mode changing using nopclearfail is disabled.  Do not set enable_remote_mode in the client configuration file.
enable_remote_reboot	If this field is set to yes (the default value) in the module configuration file, the nShield HSM remote reboot using the nopclearfail is enabled. If set to no, remote reboot using nopclearfail is disabled. Run cfg-pushnethsm to push the new config file to the module.
enable_remote_upgrade	If this field is set to yes (the default value) in the module configuration file, the nShield HSM remote upgrade using the nopclearfail is enabled. If set to no, remote upgrade using nopclearfail is disabled. Run cfg-pushnethsm to push the new config file to the module.

These flags are those used by the NFLOG_DETAIL environment variable (see Environment variables to control logging).

You can supply a number of flags with the logdetail field, which specifies the level of detail logged by the hardserver (see the table above). Supply the flags in a space separated list:

Flag	Description
external_time	This flag specifies the external time (that is, the time according to your machine's local clock) with the log entry.
external_date	This flag specifies the external date (that is, the date according to your machine's local clock) with the log entry.
external_tid	This flag specifies the external thread ID with the log entry.
external_time_t	This flag specifies the external time_ti (that is, the time in machine clock ticks rather than local time) with the log entry.
stack_backtrace	This flag specifies the stack backtrace with the log entry.

Flag	Description
stack_file	This flag specifies the stack file with the log entry.
stack_line	This flag specifies the message line number in the original library. This flag is likely to be most useful in conjunction with example code we have supplied that has been written to take advantage of this flag.
msg_severity	This flag specifies the message severity (a severity level as used by the NFLOG_SEVERITY environment variable) with the log entry.
msg_categories	This flag specifies the message category (a category as used by the NFLOG_CATEGORIES environment variable) with the log entry.
msg_writeable	This flag specifies message writeables, and extra information that can be written to the log entry, if any such exist.
msg_file	This flag specifies the message file in the original library. This flag is likely to be most useful in conjunction with example code we have supplied that has been written to take advantage of this flag.
msg_line	This flag specifies the message line number in the original library. This flag is likely to be most useful in conjunction with example code we have supplied that has been written to take advantage of this flag.
options_utc	This flag showing the date and time in UTC (Coordinated Universal Time) instead of local time.
unix_file_descriptor_max	This field sets the number of file descriptors the hardserver must be capable of having open concurrently on Linux. The value must be an integer. If unix_file_descriptor_max is set to **0** (the default), the value will be ignored by the hardserver. If it is set to a positive value, the hardserver will refuse to start if the file descriptor hard limit on the system is less than that value. This configuration entry can be used to ensure that the hardserver is capable of satisfying the maximum number of hardserver connections that applications may make use of.

3.2.2.2. hardserver loglevel

The table in this section describes the loglevels in increasing order of severity. If you set a custom [server_settings]/loglevel, you will get that level and all the more severe levels.

If the legacy NFAST_SERVERLOGLEVEL debug environment variable is set, it overrides any loglevel value set in the configuration file.

[server_settings]/log level value	Appears in the hardserver log as	Description
info	Information	Report about the hardserver start-up configuration, connections that have been established or closed. General information for nShield Support for debugging.
notice	Notice	Report about certain start-up events, some non-fatal and routine errors that the hardserver can handle internally.
client	Detected error in client behaviour	Malformed or invalid messages are received from a client, typically from a local client.
remoteserver	Remote server error	Malformed messages or protocol errors are received while communicating with remote peers over the nCipher Secure Transport/Impath protocol.
error	Nonfatal error	Unexpected but handled errors from system calls, for example for device or TCP I/O.
serious	Serious error, trying to continue	Unexpected errors from system calls, but they are more serious and likely to indicate an issue somewhere in the system.
internal	Serious internal error, trying to continue	Possible bug in the hardserver, but it might also be an issue in the environment the hardserver is running in.
startup	Fatal error during startup	The hardserver could not start, for example because of an invalid configuration file or because it cannot bind to a TCP socket which is already in use. The hardserver will abort.
fatal	Fatal runtime error	Fatal error usually referring to a non-ignorable error that has occurred after startup such as out of memory errors in certain contexts. Rarely used. The hardserver will abort.
fatalinternal	Fatal internal error	A non-recoverable failure occurred, for example certain internal self-consistency checks to detect program logic errors. The hardserver will abort.

3.2.2.3. server_performance

The server_performance section defines the performance settings for the client hardserver. These are read only at hardserver start-up.

This section contains the following fields:

Field	Description
`enable_scaling`	This field determines whether multi-threaded performance scaling is enabled or not. If this field is set to **auto** (or not set), the hardserver automatically chooses the best option for the available hardware (enabled when using an nShield network-attached HSM, for which scaling is currently optimized, and disabled if using an nShield PCIe or USB-attached HSM). It can explicitly be enabled by setting to **yes**, and explicitly disabled by setting to **no**.
`target_concurrency`	This field allows the level of concurrency to be tuned. The value must be an integer and will only come into effect when multi-threaded performance scaling is enabled. If target_concurrency is set to **0** (the default), the value will be automatically configured by the hardserver based on the available number of physical CPU cores. The target concurrency configured is written to the hardserver log.

3.2.2.4. module_settings

The `module_settings` section defines the settings for the module that can be changed while the hardserver is running.

The section contains the following fields:

Field	Description
`esn`	This field specifies the electronic serial number of the module.
`priority`	This field specifies the priority of the module. The value for this field can be an integer from 1 (highest) to 100 (lowest). The default is 100.

3.2.2.5. server_remotecomms

The `server_remotecomms` section defines the remote communication settings for the client hardserver. These are read only at hardserver start-up.

This section contains the following fields:

Field	Description
`impath_port`	This field specifies the port on which the hardserver listens for incoming impath connections. The default is 9004. Setting this field to `0` specifies that the hardserver does not listen for incoming connections. Ensure that firewall settings are consistent with port settings. See Before you install the software for more information about firewall settings.

3.2.2.6. server_startup

The `server_startup` section defines the settings for the hardserver that are loaded at start-up. Any changes you make to the settings in this section do not take effect until after you restart the hardserver. For more information, see Stopping and restarting the hardserver.

The section contains the following fields:

Field	Description (Linux)	Description (Windows)
`unix_socket_name`	This field specifies the name of the socket to use for non-privileged connections on Linux. The default is `/dev/nfast/nserver`. If the `NFAST_SERVER` environment variable is set, it overrides any value set for `unix_socket_name` in the hardserver configuration file. For more information about environment variables, see Environment variables.	This field is not used on Windows.
`unix_privsocket_name`	This field specifies the name of the socket to use for privileged connections on Linux. The default is `/dev/nfast/privnserver`. If the `NFAST_PRIVSERVER` environment variable is set, it overrides any value set for `unix_privsocket_name` in the hardserver configuration file.	This field is not used on Windows.
`nt_pipe_name`	This field is not used on Linux.	This field specifies the name of the pipe to use for non-privileged connections on Windows. An empty string specifies none. The default is `\\.\pipe\crypto`. If the `NFAST_SERVER` environment variable is set, it overrides any value set for `nt_pipe_name` in the hardserver configuration file.
`nt_pipe_users`	This field is not used on Linux.	This field specifies the name of the user or group who is allowed to issue non-privileged connections on Windows. If this field is empty (which is the default), any user can make non-privileged connections. User or group names must be specified unqualified (for example, *bob* not *MYDOMAIN\bob* or *bob@MYDOMAIN*).

Field	Description (Linux)	Description (Windows)
nt_privpipe_name	This field is not used on Linux.	This field specifies the name of the pipe to use for privileged connections on Windows. An empty string specifies none. The default is \\.\pipe\privcrypto. If the NFAST_PRIVSERVER environment variable is set, it overrides any value set for nt_privpipe_name in the hardserver configuration file.
nt_privpipe_users	This field is not used on Linux.	This field specifies the name of the user or group who is allowed to make privileged connections on Windows. If this field is empty (which is the default), only processes running with local administrator privilege can make privileged connections. User or group names must be specified unqualified (for example, *bob* not *MYDOMAIN\bob* or *bob@MYDOMAIN*).
nonpriv_port	This field specifies the port on which the hardserver listens for local non-privileged TCP connections. The value 0 (which is the default) specifies none. Java clients default to connecting to port 9000. Ensure that your network firewall settings are correct. See Before you install the software for more information about firewall settings. If the NFAST_SERVER_PORT environment variable is set, it overrides any value set for nonpriv_port in the hardserver configuration file.	
priv_port	This field specifies the port on which the hardserver listens for local privileged TCP connections. The value 0 (which is the default) specifies none. Java clients default to connecting to port 9001. If the NFAST_SERVER_PRIVPORT environment variable is set, it overrides any value set for priv_port in the hardserver configuration file.	

3.2.2.7. load_seemachine

The load_seemachine section of the hardserver configuration file defines SEE machines that the module should load and, if required, start for use by other clients. Each SEE machine is defined by the following fields:

Field	Description
module	This field specifies the module on to which to load the SEE machine. The value must be an integer. A module with this ID must be configured on the client computer.
machine_file	This field specifies the file name of the SEE machine.

Field	Description
`userdata`	This field specifies the `userdata` file name to pass to the SEE machine on start-up. If this field is blank (`" "`), the SEE machine is loaded but not started. By default, this field is blank.
`worldid_pubname`	This field specifies the `PublishedObject` name to use for publishing the `KeyID` of the started SEE machine. If this field is blank (`" "`), the `KeyID` is not published. This field is ignored if the value of the `userdata` field is blank.
`postload_prog`	This field specifies the program to run after loading the SEE machine in order to perform any initialization required by the SEE machine or its clients. The specified program must accept an argument of the form `-m module#`. To run `see-sock-serv` directly on the nShield HSM, set this field to `sockserv`.
`postload_args`	This field specifies arguments to pass to the program specified by the `postload_prog` field. The argument `-m module#` is automatically passed as the first argument. The `postload_args` field is ignored if `postload_prog` is not specified or is blank. To run `see-sock-serv` directly on the nShield HSM, set this field to `-p *pubname*`.
`pull_rfs`	This field specifies whether the SEE machine name and userdata should be pulled from the RFS. The default is `0`: set to `1` to pull the SEE machine and userdata from the RFS before loading on the remote module. This field will be ignored if set on client machine configurations. This field will not be added to existing configuration files if you are upgrading an image. If you require the new functionality enabled by this field, you can add the field to the `load_seemachine` section of your existing configuration file.

3.2.2.8. slot_imports

The `slot_imports` section defines slots from remote modules that will be available to the local computer. Each slot is defined by the following fields:

Field	Description
`local_esn`	This field specifies the ESN of the local module importing the slot.
`local_slotid`	This field specifies the `SlotID` to use to refer to the slot when it is imported on the local module. The default is 0, and provides automatic assignment to the lowest available slotID after any configured dynamic slots.
`remote_ip`	This field specifies the IP address of the machine that hosts the slot to import.
`remote_port`	This field specifies the port for connecting to the nShield HSM.

Field	Description
remote_esn	This field specifies the ESN of the remote module from which to import the slot.
remote_slotid	This field specifies the SlotID of the slot to import on the remote module. The value of this field must be an integer. The default is 0.

3.2.2.9. slot_exports

The slot_exports section defines the slots on local modules that the local hardserver should allow network modules to import. Each local slot has an entry for each remote module that can import it, consisting of the following fields:

Field	Description
local_esn	This field specifies the ESN of the local module whose slot can be imported by a network module.
local_slotid	This field specifies the SlotID of the slot that is to be imported. The value must be an integer. The default is 0.
remote_ip	This field specifies the IP address of the module that is allowed to import the slot. Use 0.0.0.0 to allow all machines. The default is 0.0.0.0
remote_esn	This field specifies the ESN of the module allowed to import the slot. Leave the value blank to allow all permitted modules in the Security World. The default is blank.

3.2.2.10. dynamic_slots

The dynamic_slots section defines the number of Dynamic Slots that each HSM is to support for the Remote Administration Service.

Field	Description
esn	ESN of the HSM to be configured with Dynamic Slots.
slotcount	The number of Dynamic Slots that the HSM is to support. If set to 0 (default) the HSM does not support the Remote Administration Service.

3.2.2.11. slot_mapping

The slot_mapping section defines, for each specified HSM, a slot that is exchanged with slot 0 of the HSM. Slot 0 becomes a Dynamic and/or Remote Slot and the local slot becomes the specified slot number. This enables applications and utilities that only support

slot 0 to use Remote Administration and Remote Operator.

Field	Description
`esn`	ESN of the HSM to which the mapping is applied.
`slot`	The slot number to be swapped with slot 0, so that: • Slot 0 refers to a Dynamic and/or Remote Slot • The specified slot number refers to the local slot of the HSM. If `slot` is set to 0 (default) there is no slot mapping.

3.2.2.12. dynamic_slot_timeouts

The `dynamic_slot_timeouts` section defines timeout values that are used to specify expected smartcard responsiveness for all HSMs associated with the relevant host or client, when using the Remote Administration.

Field	Description
`round_trip_time_limit`	`round_trip_time_limit` > 5s + network latency time Round trip (HSM to smartcard and back) time limit in seconds. The card is regarded as removed, if no response has been received within the allowed time. Expected network delays need to be taken into account when setting this. The default is ten seconds.
`card_remove_detect_time_limit`	`card_remove_detect_time_limit` >= `round_trip_time_limit` *2 Maximum number of seconds that can pass without a response from the smartcard, before it is regarded as removed and all the keys that it protects are unloaded. Lower values increase network traffic. The default is 30 seconds.

3.2.2.13. audit_logging

The `audit_logging` section defines the settings for the syslog infrastructure used by the audit logging capability. These values require a restart of the hardserver to be recognized.

Field	Description
`auditlog_port`	This field specifies the UDP port to which audit log syslog messages should be delivered. The default is 514.
`auditlog_addr`	This field specifies the IP address of the machine that hosts the syslog server to which audit logging syslog messages shoud be sent.

Field	Description
auditlog_copy_hslog	This field specifies if audit logging sylog entries should be copied into the hardserver log as well as being transmitted to the syslog server. The default is off. It can be turned by setting to **yes**, **true**, **on** or **1**. Care should be taken when setting this field as this can cause the hardserver log to grow significantly. It should never be set on a nShield network-attached HSM.

3.2.2.14. auditdb_settings

The auditdb_settings section defines the settings for the nCore audit logging database used by the nShield Audit Log Service. to learn more about this service, see nShield Audit Log Service

Field	Description
free_space_min_mb	This field specifies the minimum available megabytes of free-space for audit DB to operate. The default is 0.
db_size_max_mb	This field specifies the maximum available megabytes of free-space for audit DB to operate. The default is 0 for no limit.
audit_indexes_max	This field specifies the maximum number of audit indexes in audit DB. The default is 0 for no limit.

3.2.3. Sections only in client configuration files

3.2.3.1. nethsm_imports

The nethsm_imports section defines the network modules that the client imports. It can also be set up by the nethsmenroll utility. Each module is defined by the following fields:

Field	Description
local_module	This field specifies the ModuleID to assign to the imported module. The value must be an integer. A module with this ID must not be already configured on the client computer.
remote_ip	This field specifies the IP address of the module to import.
remote_port	This field specifies the port for connecting to the nShield HSM.
remote_esn	This field specifies the ESN of the imported module.
keyhash	This field specifies the hash of the key that the module should use to authenticate itself.

Field	Description
privileged	The value in this field specifies whether the client can make a privileged connection to the module. The default is 0, which specifies no privileged connections. Any other value specifies privileged connections.
ntoken_esn	This field specifies the ESN of this client's nToken, if an nToken is installed.

The default value for remote_keyhash (40 zeros) specifies that no authentication should occur. We recommend that you set a specific key hash in place of this default.

3.2.3.2. rfs_sync_client

This section defines which remote file system the client should use to synchronize its key management data:

Field	Description
remote_ip	The IP address of the RFS against which to synchronize.
remote_port	This field specifies the port for connecting to the RFS.
use_kneti	Setting this option to yes to use a local module KNETI instead of the default hardserver's software KNETI to authenticate this client to the RFS.
local_esn	This is only required if use_kneti is set to yes. It is the ESN of the local module used for authentication.
remote_keyhash	Software or module KNETI hash used to authenticate the RFS, or 40 zeroes to indicate no authentication required (default is 40 zeroes).
remote_esn	ESN of the remote module used to authenticate the RFS, or empty when using software KNETI authentication or no authentication required (default is empty).

3.2.3.3. remote_file_system

This section is updated automatically when the rfs-setup utility is run. Do not edit it manually.

The remote_file_system section defines a remote file system on the client by listing the modules allowed to access the file system on this client. Each module is defined by an entry consisting of the following fields:

Field	Description
remote_ip	This field specifies the IP address of the remote module that is allowed to access the file system on this client.

Field	Description
remote_esn	This field specifies the ESN of the remote module allowed to access the file system on this client.
keyhash	This field specifies the hash of the key with which the client must authenticate itself to the module. The default is 40 zeros, which means that no key authentication is required.
native_path	This field specifies the local file name for the volume to which this entry corresponds.
volume	This field specifies the volume that the remote host would access to use this entry.
allow_read	If this field is set to yes, it means that a remote server is allowed to read the contents of the file. The default is no.
allow_write	If this field is set to yes, it means that a remote server is allowed to write to the file. The default is no.
allow_list	If this field is set to yes, it means that a remote server is allowed to list the contents of the file. The default is no.
is_directory	If this field is set to yes, it means that this entry represents a directory. The default is no.
is_text	If this field is set to yes, it means that line endings should be converted to and from the Linux convention for transfers.

If you upgrade from an earlier software version to v12 and are using Remote Administration, you need to manually add the following sections to your configuration file.

3.2.3.4. remote_administration_service_slot_server_startup

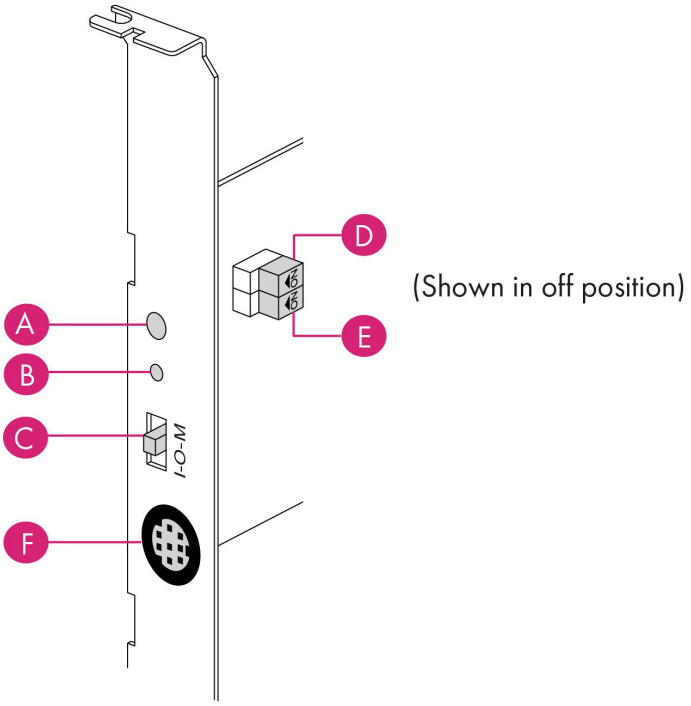
The remote_administration_service_slot_server_startup section defines the communication settings that are applied at start-up to the Remote Administration Service.

Field	Description
port	Which port to use to connect to the Dynamic Slot Server. The default is 9005.

3.3. Checking and changing the mode on an nShield Solo module

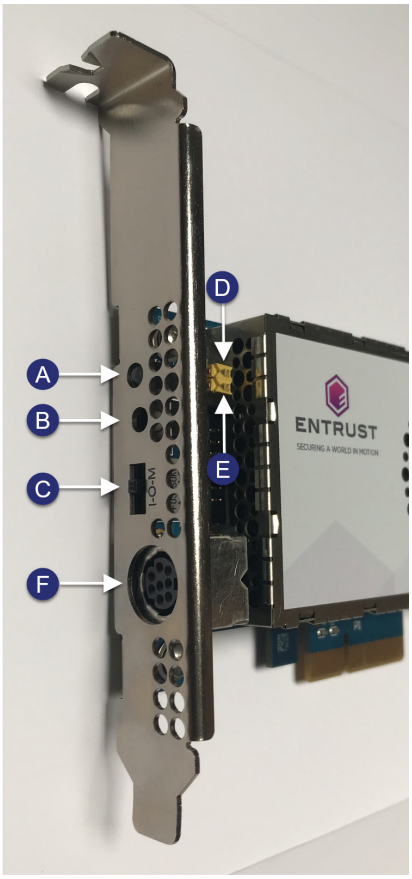
This appendix tells you how to check and change the mode on the nShield HSM. You must change the mode to perform certain maintenance and configuration tasks.

3.3.1. Back panel and jumper switches



(Shown in off position)

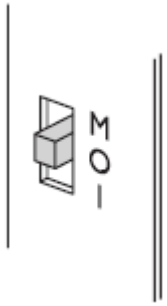
nShield Solo



nShield Solo XC

Label	Description
A	Status LED
B	Recessed reset button
C	Physical mode switch
D	Physical mode override jumper switch, in the **off** position. When set to **on**, the mode switch (C) is deactivated. See Override switches for further information.
E	Remote Administration override jumper switch, in the **off** position. When set to **on**, remote mode switching is disabled. See Override switches for further information.
F	Smart card connector, 8 Pole Female Mini-DIN connector on nShield PCIe HSM.

3.3.2. Physical mode switch



The physical mode switch on the back panel, as shown above and as 'C' in Back panel and jumper switches, enables you to select the mode on the module itself.

3.3.2.1. Available modes

The physical mode switch can be set to one of three positions:

Maintenance	Sets the module to start in pre-maintenance mode. Allows you to upgrade the firmware of the module
Operational	The default setting for day-to-day use.
Initialization	Sets the module to start in pre-initialization mode. This allows you to use the module to create a Security World or add the module to an existing one.

Once you have selected a mode, the module needs to be reset before the mode is actually changed. See Changing the mode for more about using the physical mode switch and resetting the module.

If the Physical mode override jumper switch ('D' in Back panel and jumper switches) is set to **on**, the mode is set to Operational (O) and you cannot change it using the physical mode switch. See Override switches for more about the Physical mode override jumper switch. You may, however, still be able to change the mode using the commanded mode switch. See Remote mode switch

3.3.3. Remote mode switch

The Remote mode switch enables you to change the mode from a computer using the `nopclearfail` command, without accessing the back panel of the module.

3.3.3.1. Available commands

You can use the following commands to change the mode of a module:

Command	Resulting mode
`nopclearfail --maintenance |-M`	Pre-maintenance
`nopclearfail --operational | -O`	Operational
`nopclearfail --initialization | -I`	Pre-initialization

3.3.3.2. Limitations

A privileged user can only change the mode using the remote mode switch according to the following:

- The physical mode switch must be set to Operational (O) to be able to use the remote mode switch to change the mode.
  - If the module is physically set to either Maintenance (M) or Initialization (I), the remote mode switch has no effect, once the module has been reset following the `nopclearfail` command.
- If the physical mode override jumper switch ('D' in Back panel and jumper switches) is set to **on**, the module behaves as if the physical mode switch is set to Operational (O) and the remote mode switch can be used to change the mode.
- If the remote mode override jumper switch ('E' in Back panel and jumper switches) is set to **on**, the remote mode switch cannot be used.

The following table summarizes the resulting module modes when using the remote mode switch, taking into account the physical mode switch and physical mode override jumper switch settings.

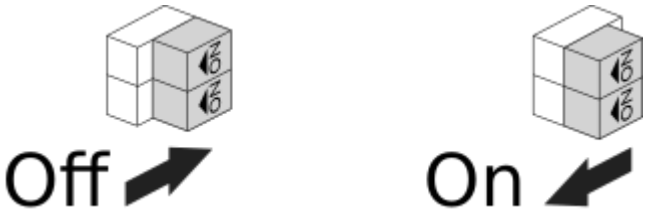
Command	Physical jumper off (D)			Physical jumper on (D)
	Physical mode switch position			
	M	O	I	
`nopclearfail --maintenance |-M`	M	M	I	M
`nopclearfail --operational | -O`	M	O	I	O
`nopclearfail --initialization | -I`	M	I	I	I

For you to be able to use the remote mode switch, the nShield HSM must be running 2.61.2 firmware or later. Otherwise the module responds with:

```
Module 1, command ClearUnitEx: HostDeviceDriverNotSupported
 -- device driver does not support software mode changes
```

See Changing the mode for more about using the remote mode switch. See Override switches for more about the remote mode override jumper switch.

3.3.4. Override switches



As shown in Back panel and jumper switches

- Switch 'D', the physical mode override jumper switch, deactivates the physical mode switch
- Switch 'E', the command mode override jumper switch, deactivates the commanded mode switch

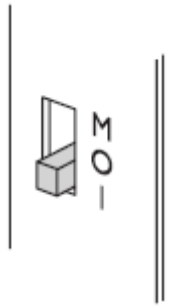
See Install a PCIe HSM for more about accessing and setting a mode override jumper switch to **off** or **on**.

3.3.5. Changing the mode

3.3.5.1. Putting a module into pre-initialization mode using the physical mode switch

Do the following:

1. Switch the physical mode switch on the back panel of the module to the initialization (I) position, as shown below:



2. Reset the module by doing one of the following:
   - Press the `Recessed reset button` ('B' in Back panel and jumper switches or:
   - Run the `nopclearfail --clear --all` command.

The module performs self-tests, during which the Status LED is lit continuously.

If the Status LED remains on continuously for more than a minute, the module self tests have resulted in a terminal failure. Contact Support.

When the self-tests are complete, the unit normally enters pre-initialization mode. In this mode, the Status LED flashes a series of single short pulses.

See Status indications for more about Status LED codes.

You can use the `enquiry` command-line utility to check that the module is in the pre-initialization mode.

After the module has been put into pre-initialization mode, it is ready to be initialized. It enters *initialization mode* when it receives an `initialization` command (for example, when you run the new-world command-line utility).

3.3.5.2. Putting a module into pre-initialization mode using the commanded mode switch

See Limitations for more about the conditions that are required to use the commanded mode switch. Do the following:

1. Run the `nopclearfail --initialization | -I` command.

   When finished, the system responds with `OK`.

   The system responds with `OK`, regardless of whether the module has been changed to the pre-initialization mode or not. To confirm that state of the module, do the following:

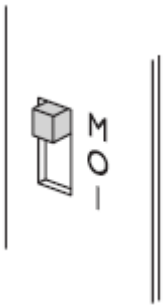
2. Run the `enquiry` command.
   The `mode` line of the `Module` section displays the current mode.

3.3.5.3. Putting a module into pre-maintenance mode using the physical mode switch

Only put a module into pre-maintenance mode if you need to upgrade module firmware. Do the following:

1. Switch the physical mode switch on the module's back panel to the maintenance (M)

position, as shown below:



2. Reset the module by doing one of the following:
   - Press the `Recessed reset button` ('B' in Back panel and jumper switches or:
   - Run the `nopclearfail --clear --all` command.

The module performs self-tests, during which the Status LED is lit continuously.

If the Status LED remains on continuously for more than a minute, the module self tests have resulted in a terminal failure. Contact Support.

When the self-tests are complete, the unit normally enters pre-maintenance mode. In this mode, the Status LED flashes a series of long pulses.

See Status indications for more about Status LED codes.

You can use the `enquiry` command-line utility to check that the module is in the pre-maintenance mode.

After the module has been put into pre-maintenance mode, it is ready for maintenance. It enters *maintenance mode* when it receives a `Maintenance` command (for example, when you run the `loadrom` command-line utility).

3.3.5.4. Putting a module into pre-maintenance mode using the commanded mode switch

See Limitations for more about the conditions that are required to use the commanded mode switch. Do the following:

1. Run the `nopclearfail --maintenance | -M` command.
   When finished, the system responds with `OK`.

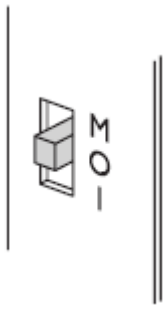
The system responds with `OK`, regardless of whether the module has been changed to the pre-maintenance mode or not. To confirm that state of the module, do the following:

2. Run the `enquiry` command.
   The `mode` line of the `Module` section displays the current mode.

3.3.5.5. Putting a module into operational mode using the physical mode switch

Do the following:

1. Switch the physical mode switch on the module's back panel to the operational (O) position, as shown below:



2. Reset the module by doing one of the following:
   - Press the `Recessed reset button` ('B' in Back panel and jumper switches or:
   - Run the `nopclearfail --clear --all` command. The module performs self-tests, during which the Status LED is lit continuously.

> If the Status LED remains on continuously for more than a minute, the module self tests have resulted in a terminal failure. Contact Support.

When the self-tests are complete, the unit normally enters operational mode and ready to accept commands.

In operational mode, the Status LED is mainly on, but blinks off briefly at regular intervals.

See Status indications for more about Status LED codes.

3.3.5.6. Putting a module into operational mode using the commanded mode switch

> See Limitations for more about the conditions that are required to use the commanded mode switch. Do the following:

1. Run the `nopclearfail --operational | -O` command.

   When finished, the system responds with `OK`.

> The system responds with `OK`, regardless of whether the module

> has been changed to the pre-maintenance mode or not. To confirm that state of the module, do the following:

2. Run the `enquiry` command.

   The `mode` line of the `Module` section displays the current mode.

3.3.6. Status indications

The following table explains the codes displayed by the Status LED.

LED	Mode	Reason
Mainly on but regularly blinks off (The exact timing depends on the nShield module. The longer the LED stays on the less the load. At 100% load the LED is off for as long as it is on.)	Operational	The Mode switch is in the operational position or the Mode override jumper switch is on. See Install a PCIe HSM for more about accessing the Mode override jumper switch and setting it to off.
Emits repeated short flashes	Pre-initialization	The Mode switch is in the initialization position.
Emits repeated long flashes	Pre-maintenance	The Mode switch is in the maintenance position.
Flashes the Morse SOS pattern followed by a code	Error	The module has encountered an unrecoverable error. See Morse code error messages for more about these errors.

3.4. nShield 5s modes of operation

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

- Modes of operation
- Check and change the mode of operation
- Return to factory state
- Recovery mode

3.4.1. Modes of operation

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

Status	Description
Starting up	The nShield 5s HSM is booting up and performing self tests. After all tests complete successfully, the HSM enters Operational mode.
Operational mode	The nShield 5s HSM is working and ready to perform cryptographic operations. An initialized HSM enters Operation mode automatically after it is powered up and all pre-tests are successfully completed. To enter Operational mode manually, see Check and change the mode of operation.
Emulated maintenance mode	The nShield 5s HSM is ready to receive maintenance commands, or is processing a maintenance command. The HSM remains in Emulated maintenance mode until you change mode manually, see Check and change the mode of operation.
Pre-initialization mode	The nShield 5s HSM is ready to receive initialization commands. For example, initialization commands to set the root-of-trust key (KNSO), to create a Security World, or load an existing Security World. To enter Pre-initialization mode, see Check and change the mode of operation.
Initialization mode	The nShield 5s HSM is processing an initialization command. After the command completes, the HSM will return to Pre-initialization mode.
Uninitialized mode	The nShield 5s HSM was booted with no root-of-trust key (KNSO) set. This typically happens after leaving a factory state, see Return to factory state. To resolve this, switch to Pre-initialization mode, set the KNSO and reboot the HSM.
Error	The nShield 5s HSM is in an error state, see HSM status indicators and error codes (nShield 5s). No cryptographic operations can be performed until this error has been cleared.
Recovery mode	The nShield 5s HSM is running on the recovery image instead on the primary image. See Recovery mode.
Factory state	The nShield 5s HSM is in a factory state. See Return to factory state.

3.4.2. Check and change the mode of operation

You must change the mode on the nShield 5s HSM to perform certain maintenance and configuration tasks. The nShield 5s HSM does not have a physical mode switch. Switch between modes using the `nopclearfail` utility.

Use the following commands to change the mode of an nShield 5s HSM:

Command	Resulting mode
`nopclearfail --maintenance |-M`	Emulated maintenance mode
`nopclearfail --operational | -O`	Operational
`nopclearfail --initialization | -I`	Pre-initialization

1. Run the `nopclearfail` command specifying the module number and the new mode.

   When finished, the system responds with `OK`. This message is not confirmation that the module has changed mode.

   ```
   nopclearfail --maintenance --module 1
   Module 1, command ClearUnitEx: OK
   ```

2. Confirm the new mode of the module by running the `enquiry` command.

   The `mode` line of the `Module` section displays the current mode.

   ```
   enquiry -m1
   Module #1:
   enquiry reply flags  none
   enquiry reply level  Five
   serial number        XXXX-XXXX-XXXX
   mode                 Emulated maintenance mode. hsmadmin may be used to perform module management whilst in
   this mode.
   module type code     14
   product name         NC5536E/NC5536N
   device name          #1 Secure Shell nshield-XXXX-XXXX-XXXX.local
   hardware status      OK
   ```

3.4.3. Return to factory state

nShield 5s HSMs that are delivered from the factory contain no data relating to the `ncoreapi` 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 possible 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.

3.4.3.1. 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

3.4.3.1.1. 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.

3.4.3.2. 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.

3.4.4. 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.

3.4.4.1. 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.

3.4.4.2. 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 rebooting. 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.

3.4.4.3. 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.

3.5. Upgrade firmware: nShield 5s

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

3.5.1. 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.

3.5.2. 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.

3.5.2.1. 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.

3.5.2.1.1. 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.

3.5.2.2. 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 Bootloader upgrade is reversible or not.

It is only possible to update the Bootloader when running firmware version 13.4 or later.

3.5.3. 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.

3.5.3.1. 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.

3.5.4. 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 installation.

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 upgrading 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.

3.6. Upgrade firmware: nShield Solo, Solo XC, and Edge HSMs

This appendix describes how to load an updated image file and associated firmware onto your nShield hardware security module.

3.6.1. Version Security Number (VSN)

The firmware includes a Version Security Number (VSN). This number is increased whenever we improve the security of the firmware.

We supply several versions of the module firmware. You can always upgrade to firmware with an equal or higher VSN than that currently installed on your module.

You can never load firmware with a lower VSN than the currently installed firmware.

Ensuring you use firmware with the highest available VSN allows you to benefit from security improvements and enhanced functionality. It also prevents future downgrades of the firmware that could potentially weaken security. However, you may choose to install an

associated firmware that does not have the highest available VSN. For example, if you have a regulatory requirement to use FIPS-approved firmware, you should install the latest available FIPS-validated firmware, which may not have the highest VSN. Similarly, if you want to install a version with enhanced features without committing yourself to the upgrade, you can do so providing you upgrade only to firmware with a VSN equal to that currently installed on your module.

3.6.2. Firmware on the installation media

Your nShield HSM and Firmware installation media contains several sets of firmware for each supplied product. These can include the latest available:

- FIPS-approved firmware with the base VSN
- FIPS-approved firmware with a higher VSN
- Firmware awaiting FIPS approval with the base VSN
- Firmware awaiting FIPS approval with a higher VSN.

You should ensure you are using the latest firmware, unless you have a regulatory requirement to use firmware that has been FIPS validated. In the latter case, you should ensure that you are using the latest available FIPS validated firmware.

3.6.2.1. Recognising firmware files

The firmware and monitor 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 and monitor files for hardware modules have a `.nff` filename suffix. Monitor filenames have a `solo-monitor` prefix and are in the `Solo Monitor` subdirectory. (Files that have a `.ftv` suffix are used for checking similarly named firmware files. They are not firmware files.)

Files for use with nShield Solo modules have `solo` in the filename and are in the `Solo` subdirectory. Files for use with nShield Solo XC modules have `soloxc` in the filename and are in the `SoloXC` subdirectory. Files for use with nShield Edge modules have `edge` in the filename and are in the `Edge` subdirectory.

The VSN of a 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

```
loadrom --view /disc-name/firmware/product/status/firmware_file.nff
```

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

Windows

```
loadrom --view E:\firmware\product\status\firmware_file.nff
```

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

3.6.3. Using new firmware

To use the new firmware, you must:

1. Install the latest software.
2. Install the latest firmware, as described below.

Windows-only This appendix assumes that you have installed the hardserver as a service. This is the default installation procedure.

3.6.4. Firmware installation overview

The process of installing or updating firmware on an nShield module depends on whether you need to upgrade the module's monitor.

The Solo XC module does not have a separate monitor program, see Upgrading firmware only.

Each module has a monitor, which allows you to load firmware onto the module.

To check the version number of the monitor on the module:

1. Log in to the host as a user in the group `nfast` (**Linux**) or as an Administrator (**Windows**).
2. Put the module in Maintenance mode and reset the module.
    - The HSM must be in pre-initialization mode.

3. Run the `enquiry` command-line utility and check that the module is in the pre-maintenance state.

   The `Version` number shown is for the monitor.

If you need to upgrade both the monitor and firmware, you must use the `nfloadmon` utility; see Upgrading both the monitor and firmware.

If you need to upgrade the firmware only, you must use the `loadrom` utility; see Upgrading firmware only.

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.

3.6.5. Upgrading both the monitor and firmware

You must only use this procedure if you need to upgrade the monitor and firmware on an nShield module, for example, for Remote Administration functionality. If you only need to upgrade the firmware, or have a Solo XC module, see Upgrading firmware only.

Follow this procedure carefully. Do not interrupt power to the module during this upgrade process.

To upgrade the monitor and firmware on a module:

1. Log in to the host as a user in the group `nfast` (**Linux**) or as an Administrator (**Windows**).
2. Run the command:

   Linux

   ```
   nfloadmon -m<module_number> --automode /disc_name/firmware/product/monitor/status/monitor_file.nff
   /disc-name/firmware/product/status/firmware_file.nff
   ```

   Windows

   ```
   nfloadmon -m<module_number> --automode E:\firmware\product\monitor\status\monitor_file.nff
   E:\firmware\product\status\firmware_file.nff
   ```

   In this command:

   - `<module_number>` is the module number (such as `-m2` for module 2).
   - `disc_name` (**Linux**) is the directory on which you mounted the installation media.

- `E` (**Windows**) is the drive letter of your installation media.
- `status` is the certification status.
- `monitor_file` is the monitor file name.
- `product` is the type of product.
- `firmware_file` is the firmware file name.

`--automode` enables automated mode switching for nShield PCIe HSMs, when supported in Remote Administration environments.

Monitor version 2.60.1 is required to enable remote mode switching. Remote mode switching is not supported on nShield USB-attached HSMs.

For example:

Linux

```
nfloadmon -m2 /mnt/cdromname/firmware/Solo/monitor/latest/solo-2-60-1-vsn26.nff
mnt/cdromname/firmware/Solo/latest/solo-13-3-1-vsn29.nff
```

Windows

```
nfloadmon -m2 --automode E:\firmware\Solo\monitor\latest\solo-2-60-1-vsn26.nff
E:\firmware\Solo\latest\solo-13-3-1-vsn29.nff
```

The firmware files are signed and encrypted; you can load only the correct version for your module.

Upgrading the nShield Solo XC to 13.3.x firmware also triggers additional reboots. These additional reboots are only triggered on the Solo XC and when upgrading to 13.3.x. They are not triggered on other nShield HSMs during firmware upgrade. On the Solo XC, the additional reboots increase the upgrade time by up to five minutes and require that you keep both the Solo XC and the host connected to the power.

3. Confirm the version of the monitor and firmware.
4. Put the module into the different modes if and when prompted to do so. When supported, the mode of the nShield PCIe HSM changes automatically. Changing mode on an nShield USB-attached HSM requires the **Clear** switch to be pressed.

   For information on changing the mode, see * The HSM must be in pre-initialization mode.

5. When the `nfloadmon` utility has completed, put the module into initialization mode (if prompted), and then initialize the module by running the command:

   ```
   initunit
   ```

6. Put the module in Maintenance mode and reset the module.
7. Run the `enquiry` command to verify the module is in maintenance state and has the correct monitor version.

   In Maintenance mode, the `enquiry` command shows the version number of the monitor.

8. Put the module in Operational mode and reset the module.
9. Run the `enquiry` command to verify the module is in operational state and has the correct firmware version.
10. Log in to the host as normal.

    In Operational mode, the `enquiry` command shows the version number of the firmware.

3.6.6. Upgrading firmware only

PCIe HSMs

The firmware is provided on a separate `.iso` and not on the Security World installation media. For the latest nShield firmware, request a DVD or `.iso` download link from Entrust Support at nshield.support@entrust.com.

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.

To upgrade the firmware on a module:

1. Log in to the host as a user in the group `nfast` (**Linux**) or as an Administrator (**Windows**).
2. Put the module in Maintenance mode and reset the module.
   - The HSM must be in pre-initialization mode.
3. Run the `enquiry` command-line utility to check that the module is in the pre-maintenance state.
4. Insert the firmware DVD or mount the firmware `.iso`, depending on the provided

upgrade media format.

5. Load the new firmware by running the command:

   Linux

   ```
   loadrom -m<module_number> /disc_name/firmware/product/status/firmware_file.nff
   ```

   Windows

   ```
   loadrom -m<module_number> E:\firmware\product\status\firmware_file.nff
   ```

   In this command:

   - `<module_number>` is the module number (such as `-m2` for module 2).
   - `disc_name` is the directory on which you mounted the installation media.
   - `E` is the drive letter of your installation media.
   - `product` is the type of product.
   - `status` is the certification status.
   - `firmware_file` is the firmware file name.

   For example:

   Linux

   ```
   loadrom -m2 /mnt/cdromname/firmware/Solo/latest/solo-13-3-1-vsn29.nff
   ```

   Windows

   ```
   loadrom -m2 E:\firmware\Solo\latest\solo-13-3-1-vsn29.nff
   ```

   The firmware files are signed and encrypted; you can load only the correct version for your module.

   Upgrading the nShield Solo XC to 13.3.x firmware also triggers additional reboots. These additional reboots are only triggered on the Solo XC and when upgrading to 13.3.x. They are not triggered on other nShield HSMs during firmware upgrade. On the Solo XC, the additional reboots increase the upgrade time by up to five minutes and require that you keep both the Solo XC and the host connected to the power.

6. **Solo XC only**

   Reboot the Solo XC for the firmware upgrade to take effect:

Linux bare metal environments

With the module in Maintenance mode, run the following command to reboot the Solo XC.

```
nopclearfail -S -m<module_number>
```

Linux virtual environment hosts

Reboot the Solo XC by rebooting the system that is hosting the Solo XC.

Windows

With the module in Maintenance mode, reboot the system that is hosting the Solo XC.

Wait for the Solo XC to reboot. This takes around 10 minutes on a host machine running Linux. The module has completed rebooting when running `enquiry` no longer shows the module as Offline.

7. Put the module in initialization mode and reset the module.
8. Initialize the module by running the command:

```
initunit
```

9. Put the module in Operational mode and reset the module.
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.

11. Log in to the host as normal.

3.6.7. After firmware installation

After you have installed new firmware and initialized the HSM, you can create a new Security World with the HSM or reinitialize the HSM into an existing Security World.

If you are initializing the HSM into a new Security World, see Create a new Security World.

If you are re-initializing the HSM into an existing Security World, see Adding or restoring an HSM to the Security World.

3.7. 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.

3.7.1. 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.

3.7.2. 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 administrator 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 information on setting the system clock.

3.7.3. 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.

3.7.4. 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.

3.7.4.1. Restrictions on setting the clock

To prevent malicious actors from tampering with the HSM's system clock by moving it backward, 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.

3.8. 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.

3.8.1. Checking operational status

3.8.1.1. 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 following:

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.

3.8.1.2. 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.

3.8.2. 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 prevent 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.

3.8.3. 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.

3.8.3.1. 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
```

3.8.3.2. Notice

This type of message is sent for information only:

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

3.8.3.3. 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
```

3.8.3.4. 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.

3.8.3.5. 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.

3.8.3.6. 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.

3.8.3.7. 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.

3.8.4. 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.

3.9. 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.

3.9.1. 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.

3.9.2. 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.

3.9.2.1. 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

Colour	Digits	Dots and dashes	Morse code	Meaning
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
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

Colour	Digits	Dots and dashes	Morse code	Meaning
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

3.9.3. 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 hardserver 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.

3.9.3.1. 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).

Code	Meaning
O L K	SIGOSERROR: runtime library internal error.
O L L	SIGUNKNOWN: invalid signal raised.

3.9.3.2. 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.
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.

Code	Meaning
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.

3.9.3.3. Operational mode 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.

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.

Code	Meaning	Action
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.

3.10. HSM Status indicators (nShield Solo and Solo XC)

This guide covers the following HSMs:

- nShield Solo
- nShield Solo XC

The blue Status LED indicates the operational status of the module.

Status LED	Description
Off.	`Status: Power off` There is no power supply to the module. Check that the module is correctly inserted in its PCIe slot, then restart the computer.
On, occasionally blinks off.	`Status: Operational mode` The HSM is accepting commands. The more frequently the Status LED blinks off, the greater the load on the module.
Flashes two short pulses, followed by a short pause.	`Status: Initialization mode` Used to create and load Security World data on the HSM and to erase Security World from the HSM and return it to factory state.
Flashes two long pulses followed by a pause.	`Status: Maintenance mode` Used for reprogramming the module with new firmware. The module only goes into Maintenance mode during a software upgrade.

Status LED	Description
Flashes SOS, the Morse code distress code (three short pulses, three long pulses, three short pulses). After flashing SOS, the Status LED flashes an error code in Morse code.	`Status: Error mode` If the module encounters an unrecoverable error, it enters Error mode. In Error mode, the module does not respond to commands and does not write data to the bus. For HSMs running firmware Remote Administration 2.61.2 and above, the error code is also reported by the `enquiry` utility in the `hardware status field` of the `Module`. If a command does not complete successfully, the module normally writes an error message to the log file and continues to accept further commands. It does not enter Error mode.

Use the mode switch to move between Maintenance, Operational, and Initialization modes. See Mode switch and jumper switches for more information.

3.11. 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.

3.11.1. Canadian certification - CAN ICES-3 (A)/NMB-3(A)

3.11.2. 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.

3.11.3. 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.

3.11.4. Recycling and disposal information

For recycling and disposal guidance, see the nShield product's *Warnings and Cautions* documentation.

3.12. 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.

3.12.1. 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

3.12.2. Replace a battery on an nShield Solo XC or nShield 5s HSM

Dispose of the used battery in accordance with local regulations.

Required tools

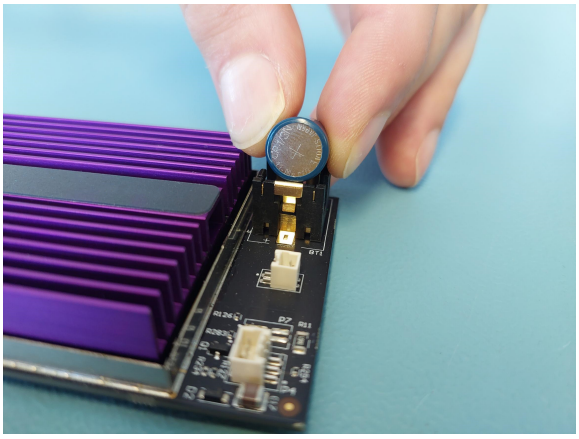
- Small non-conductive tweezers

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.

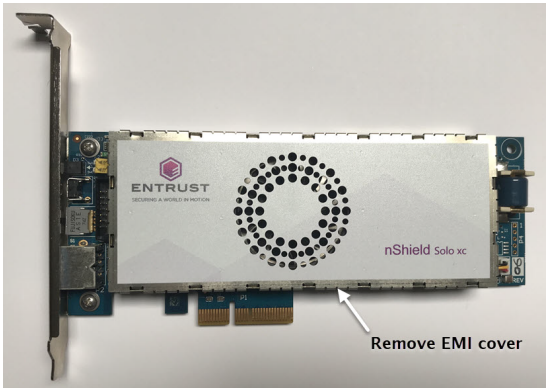
3.13. Replace the fan (Solo XC)

Required Tools

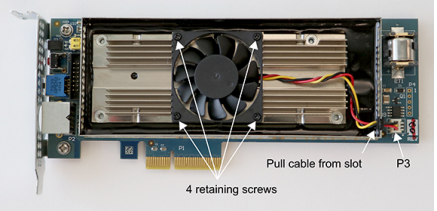
- Phillips screwdriver #0
- Phillips screwdriver #2
- Small needle nose pliers

Required Part

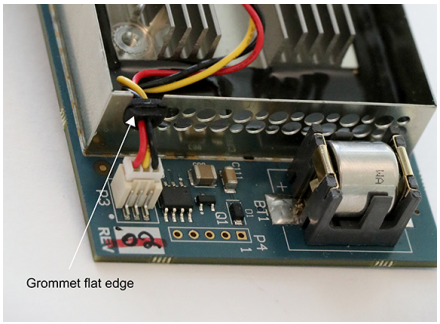
- Orderable part number SOLOXC-REP-FAN (Replacement fan assembly).
    1. Power off the system and while taking ESD precautions, remove the Solo XC card.
    2. Place the Solo XC on a flat surface.
    3. Remove the top EMI cover using a #2 screwdriver.



    4. Pull the fan power cable and grommet from the slot in the EMI fence.
    5. Using the needle nose pliers, gently remove the fan power cable from the P3 connector.



6. Using the #0 Phillips screwdriver, remove the four fan retaining screws.
7. Remove the defective fan from the Solo XC and install the replacement fan with the power cable positioned towards the P3 power connector. Ensure that the fan lays flat against the heatsink.
8. Replace the four fan retaining screws.
9. Install the power cable connector into the Solo XC P3 power connector.
10. Install the power cable grommet into the slot in the EMI fence, with the flat side towards the top of the fence.



11. Replace the top EMI cover.
12. Re-install the Solo XC into the PCIe slot.

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

3.14.1. 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 and (nShield 5 HSMs). You need to install the SSH keys for each service before you can use those services.

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 whenever SSH keys are installed or changed, if your security policy allows.

3.14.2. 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.

3.14.3. 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.

3.14.4. 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.

3.14.5. 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.

3.14.6. 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.

3.14.7. 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 services.

3.14.8. 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.

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

3.15.1. 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.

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.

Service	Service Description
`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.

3.15.2. 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.

3.15.2.1. 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 readable 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 readable 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 addition to any other protection options specified).

3.15.2.1.1. 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.

3.15.3. 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 service 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 variable 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`

3.15.4. 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`

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.

3.16. 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.

3.16.1. 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 ./`

3.17. 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 management 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.

3.17.1. 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.

3.17.2. 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.

3.17.3. 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.

3.17.3.1. 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.

3.17.3.2. 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.

3.17.3.3. 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 requirement 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**.

3.17.4. XenServer environments

Install the XenServer, follow the instructions in the *Citrix XenServer Quick Start Guide*, see https://docs.citrix.com/en-us/xenserver.

3.17.4.1. 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.

3.17.4.2. 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.

3.17.5. 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

> elevated mode.

3.17.5.1. Set up

3.17.5.1.1. 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/.

3.17.5.1.2. 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.

3.17.5.1.3. 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
```

3.17.5.1.4. 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
```

3.17.5.1.5. 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.

3.17.5.1.6. 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**.

3.17.5.1.7. 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
```

3.17.5.2. 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}
```

3.17.5.3. 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. nShield USB HSMs

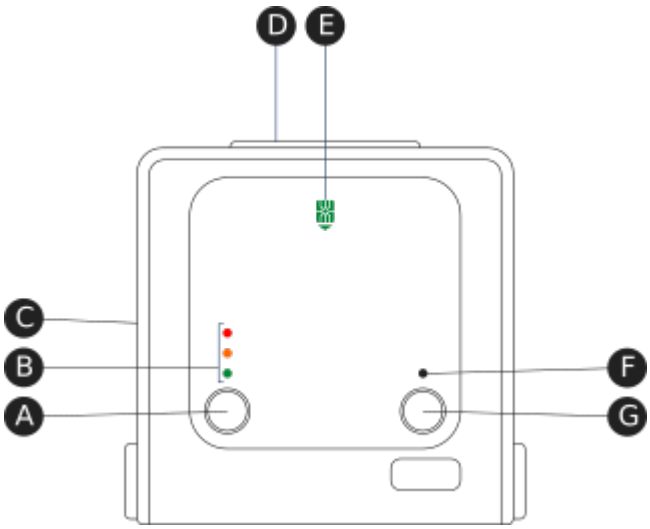
4.1. Using the nShield Edge

This guide covers the following HSMs:

- nShield Edge

For information on upgrading the firmware of an nShield Edge, see Upgrade firmware: nShield Solo, Solo XC, and Edge HSMs. For information on configuring the module, see Client software and module configuration: PCIe and USB HSMs.

4.1.1. Controls, card slot, and LEDs



Key:

A	Mode button	Selects a mode—the mode changes only when you press the **Clear** button.
B	Mode LEDs	Shows the current mode or selected mode.
C	B type USB port	For connecting the nShield Edge to the computer.
D	Card slot	For inserting the required smart card.
E	Card slot LED	Lights green when a smart card is inserted.
F	Status LED	Shows the status of the nShield Edge.

G	Clear button	Clears the memory of the nShield Edge and changes the selected mode. When using this button, press and hold it for a couple of seconds.

4.1.2. Mode LEDs

	Red	In Maintenance mode
	Red flashing	Maintenance mode selected
	Amber	In Initialization mode
	Amber flashing	Initialization mode selected
	Green	In Operational mode
	Green flashing	Operational mode selected

You generally use the nShield Edge in Operational (O) mode, but you must put it into Initialization (I) mode when creating the Security World.

4.1.3. Changing the mode

To change the mode:

1. Use the **Mode** button to highlight the required mode.
2. Within a few seconds, press and hold the **Clear** button for a couple of seconds.

If the mode changes, the new mode's LED stops flashing and remains lit. The Status LED might flash irregularly for a few seconds and then flashes regularly when the nShield Edge is ready.

Otherwise, the nShield Edge remains in the current mode, with the appropriate mode LED lit.

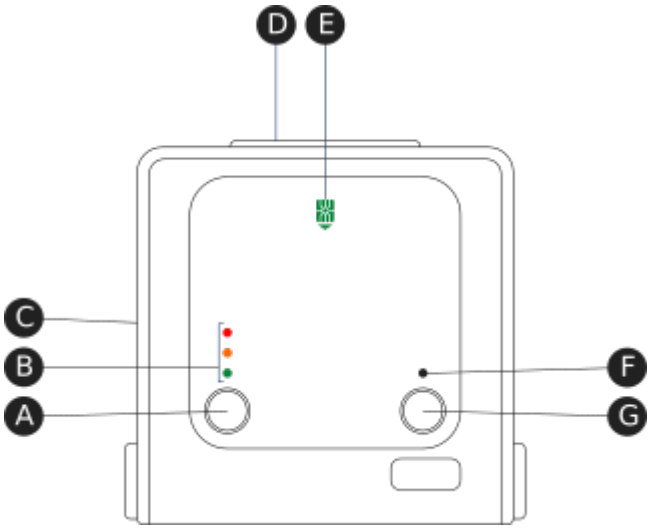
4.1.4. Status LED

	Long blue flash	In Operational mode
	Short blue flash	In Maintenance or Initialization mode
	Irregular flash	Changing mode or processing data
	Off	No power

If the Status LED flashes irregularly and the nShield Edge is unresponsive for more than a few minutes, see Troubleshooting.

4.2. Checking and changing the mode on an nShield Edge

This appendix tells you how to check and change the mode on your nShield HSM. You must change the mode to perform certain maintenance and configuration tasks. The Mode LEDs on the nShield HSM show the current or selected mode. The Status LED shows the status of the device.



A	Mode button	Selects a mode—the mode changes only when you press the `Clear` button.
B	Mode LEDs	Shows the current mode or selected mode.
C	B type USB port	For connecting the device to the computer.
D	Card slot	For inserting the required smart card.
E	Card slot LED	Lights green when a smart card is inserted.
F	Status LED	Shows the status of the device.
G	Clear button	When pressed and held for several seconds, clears the device's memory and changes the selected mode.

4.2.1. Mode LEDs

	Red	In Maintenance mode
	Red flashing	Maintenance mode selected
	Amber	In Initialization mode
	Amber flashing	Initialization mode selected
	Green	In Operational mode
	Green flashing	Operational mode selected

You mainly use the device in Operational (O) mode. You must put it into Initialization (I) mode when creating a Security World on the device, while Maintenance (M) mode allows you to upgrade the firmware. To change the mode:

1. Use the `Mode` button to highlight the required mode.
2. Within a few seconds, press and hold the `Clear` button for a couple of seconds.

   If the mode changes, the new mode's LED stops flashing and remains lit. The Status LED might flash irregularly for a few seconds and then flashes regularly when the device is ready.

   Otherwise, the device remains in the current mode, with the appropriate mode LED lit.

4.2.2. Status LED

	Long blue flash	In Operational mode
	Short blue flash	In Maintenance or Initialization mode
	Off	No power
	Irregular flash	Changing mode or processing data

If the Status LED flashes irregularly and the device is unresponsive for more than a few minutes, the device has encountered an error. Disconnect the device, wait a few seconds, and then reconnect it.

4.3. nShield Edge Windows compatibility issues and considerations

This guide covers the following HSMs:

- nShield Edge

For further information about compatible operating systems and virtual environments, see Compatibility in the release notes for the version of Security World you are using.

4.3.1. nShield Edge very slow in VMware virtual machine

In Windows installations, the nShield Edge can be very slow when used with a virtual machine under VMware (Workstation or Player). This can leading to the COM port timing out and errors in the Event log.

The problem does not happen in all installations and is not consistent on specific hardware platforms.

The work-around for the problem involves using the USB Serial driver on the Host rather than on the Guest, and mapping a serial port on the Guest to it (details below).

To apply the work-around to use the USB to serial driver on the Host rather than on the Guest, do the following:

1. With the Guest running, use the VMware Workstation/Player menu to disconnect the nShield Edge from the Guest and reconnect it to the Host. Now shut down the Guest.
2. Verify that the USB Serial Port now shows under Ports (COM & LPT) in Device Manager on the Host. On recent versions of Windows, the driver will be installed automatically or can be found via Window Update. If you are unable to find the drivers, you may need to install the Security World Software on the Host. If you do so, make sure to stop and disable the nFast Server and nFast Edge services on the Host, so they do not prevent the Guest from using of the unit. Make a note of the COM port number of the port.
3. Edit the settings of the Virtual machine in Workstation/Player. Disable the setting to automatically connect to new USB devices to make sure the Guest will not connect to the nShield Edge directly again. Add a serial port to the VM, specifying to use a physical serial port, on the host, and selecting the USB serial port from the previous step. Save the settings.
4. Start the Guest. Open the config file in a text editor. It is a plain text file named config (no extension), located in `%NFAST_KMDATA%\config`. In the section [server_startup] add a line: `serial_dtpp_devices=COM2`, specifying the COM port number of the new serial port in the VM. Make sure this is the only line with `serial_dtpp_devices` in the section. Save the file, and restart the nFast Server service to make the new configuration active.

You can now use the nShield Edge in the Guest without excessive time out errors.

4.4. Troubleshooting

This guide covers the following HSMs:

- nShield Edge

If the nShield Edge does not function as expected, check the symptoms against the following conditions and try the suggested action. If these actions do not solve your problem, contact https://nshieldsupport.entrust.com.

4.4.1. None of the LEDs are lit

The nShield Edge is not receiving power. Check that the USB cable is undamaged and connected to the nShield Edge and computer. Try another USB port on the computer.

4.4.2. The Mode LED is amber or red

The nShield Edge is not in the Operational (O) mode. Press the **Mode** button to select the Operational mode, and then press and hold the **Clear** button for a couple of seconds. Wait a few seconds before using the nShield Edge.

4.4.3. The Status LED is flashing irregularly and the nShield Edge is unresponsive for more than a few minutes

The nShield Edge has encountered an error. Disconnect the nShield Edge, wait a few seconds, and then reconnect it.

4.4.4. The Security World Software does not detect the connected nShield Edge

Disconnect the nShield Edge, wait a few seconds, and then reconnect it.

Run the `enquiry` command. If the command output says that the module is not found, restart the hardserver.

4.4.5. Upgrading the firmware

If you are instructed to upgrade firmware of the nShield Edge, see Upgrade firmware: nShield Solo, Solo XC, and Edge HSMs for instructions.

4.5. Regulatory notices

This page applies to the following HSMs:

- nShield Edge

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.

4.5.1. Canadian certification - CAN ICES-3 (A)/NMB-3(A)

4.5.2. Recycling and disposal information

For recycling and disposal guidance, see the nShield product's *Warnings and Cautions* documentation.

5. HSM Management

5.1. 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, 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

5.1.1. 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.

5.1.2. 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.

5.1.3. Available optional features

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

5.1.3.1. Elliptic Curve

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. 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.

5.1.3.2. Elliptic Curve activation

Prior to V13.5 firmware, all nShield HSMs require specific activation to utilize the elliptic curve features. HSMs use an activator smart card to enable this feature. Additionally it is possible to activate the elliptic curve feature without a physical smart card. In this case the certificate details can be provided by email and entered locally.

From firmware V13.5, elliptic curve support is always enabled.

Contact Sales if you require an EC activation.

nShield modules with elliptic curve activation support *MQV* (*Menezes-Qu-Vanstone*) modes.

5.1.3.3. 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 Brainpool 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
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.

5.1.3.4. 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.

5.1.3.5. 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	
	BrainpoolP224r1	NISTP521	
	BrainpoolP224t1	NISTB163	
	BrainpoolP256r1	NISTB233	
	BrainpoolP256t1	NISTB283	
	BrainpoolP320r1	NISTB409	
	BrainpoolP320t1	NISTB571	
	BrainpoolP384r1	NISTK163	
	BrainpoolP384t1	NISTK233	
	BrainpoolP512r1	NISTK283	
	BrainpoolP512t1	NISTK409	
		NISTK571	

5.1.3.6. 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.

5.1.3.7. 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.

5.1.3.8. Secure Execution Engine (SEE)

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.

5.1.3.9. 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.

5.1.3.10. 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.

5.1.3.11. 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.

5.1.3.12. 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).

5.1.3.12.1. 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.

5.1.4. Ordering additional features

When you have decided that you require a new feature, 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.

5.1.5. Enable features on a network-attached HSM

Feature enabling differs for static and dynamic features.

- You can enable static features from the front panel of the unit or from the client.
- Entrust recommends that you enable dynamic features from the client. If the dynamic feature applies directly to nShield HSM, for example client licenses, you can use a nethsmadmin option to apply them.

When enabling static feature(s) from the front panel, either using a card or a file, the module is cleared without warning. This will cause the HSM to drop or restart any SEE machine, and lose all the application keys that were loaded. In some cases, applications may need to be restarted.

5.1.5.1. View enabled features

To see which (if any) features have already been enabled on the nShield HSM, from the main menu select **HSM > HSM feature enable > View current state**.

To print this list to a file on the unit, select **HSM > HSM feature enable > Write state to file**. The resulting file is transferred when the unit configuration is pushed to the remote file system. You can find it in `/opt/nfast/kmdata/hsm-ESN/features/fet.txt` (**Linux**) or `%NFAST_KMDATA%\hsm-ESN\features\fet.txt`.

5.1.5.2. Enable features with a smart card

To enable a new feature with a Feature Enabling smart card from Entrust:

1. Insert the Feature Enabling smart card into the unit's slot.
2. From the front panel, select **HSM > HSM feature enable > Read FEM from card**.

A message is displayed if the features are enabled successfully. If you do not see this message confirming a successful upgrade, see Enabling features without a smart card.

5.1.5.2.1. Enabling features without a smart card

You can also provide the Feature Enabling Certificate information supplied by Entrust from a file.

To enable a feature without a smart card:

1. Put the file that contains the feature enabling certificate in `/opt/nfast/kmdata/hsm-ESN/features` (**Linux**) or `%NFAST_KMDATA%\hsm-ESN\features` (**Windows**) on the remote file system. In this path, `ESN` is the ESN of the module.

   You can give the file any name that you wish. You must enter the file name on the unit's front panel, so you may prefer to use a short name.

2. From the front panel, select **HSM > HSM feature enable > Read from a file**.
3. Specify the name of the file that contains the certificate.

If the feature is enabled successfully, a message confirms this.

5.1.5.3. Remotely enabling dynamic feature certificates including nShield HSM client licenses

Feature certificates contained on the remote file system (RFS) can be applied to the nShield HSM. The main use case for applying feature certificates is for enabling the client licenses dynamic feature which have been purchased after the initial nShield HSM purchase, although both static and other dynamic feature certificates can be applied.

If you have performed a factory reset of your HSM, ensure you re-enable any dynamic features.

To apply a dynamic feature certificate, such as an nShield HSM client license, do the following:

Feature certificates must be present on the RFS in the folder `$NFAST_KMDATA/hsm-ESN/features`.

1. Use the `nethsmadmin` utility to list the nShield HSM feature files on the RFS. Run the command:

   ```
   nethsmadmin --module=<MODULE> --rfs=<RFS_IP> --list-features
   ```

   In this command:

   - `<MODULE>` specifies the HSM to use, by its ModuleID (default = 1).

- <RFS_IP> specifies the IP address of the RFS.
- Additionally the --rfs-hkneti=<RFS_HKNETI> and --rfs-esn=<RFS_ESN> options can be set to enable secure authentication of the RFS. There are three possible cases:
  - **Without secure authentication**: The authentication of the RFS will be based on the IP address only if the --rfs-hkneti and --rfs-esn options are not specified.
  - **Software-based authentication**: The --rfs-hkneti option specifies the software KNETI hash of the RFS. The --rfs-esn option shall not be specified.

    <RFS_HKNETI> can be obtained by running anonkneti -m0 localhost on the RFS.
  - **nToken authentication**: Only if an nToken (or local HSM) is installed in the RFS. The --rfs-hkneti and --rfs-esn options specify the KNETI hash and ESN of the nToken.

    <RFS_HKNETI> and <RFS_ESN> can be obtained by running ntokenenroll -H on the RFS.

2. Use the nethsmadmin utility to make the nShield HSM use a specific feature file from the RFS. Run the command:

```
nethsmadmin --module=<MODULE> --rfs=<RFS_IP> --apply-feature=<feature_file>
```

In this command:

- <feature_file> must be the path to the feature file that is displayed when you run the nethsmadmin command with the --list-features option. Errors are reported if you use either just the feature name, or the full path. The file must be alphanumeric, and no longer than 150 characters.

  The following is an example of the output expected when applying a dynamic feature:

```
Applying feature <DYNAMIC_FEATURE> to module <MODULE_NO> ...
Feature <DYNAMIC_FEATURE> application process started on module <MODULE_NO>
*DYNAMIC_FEATURE DETECTED*
Please restart you clientside hardserver and check the enquiry output to ensure the dynamic feature
has been applied correctly!
For the client licences feature check the 'max exported modules' section in enquiry to see if the new
client number has been applied correctly.
```

  The following is an example of the output expected when applying a static feature:

```
Applying feature <STATIC_FEATURE> to module <MODULE_NO> ...
Feature <STATIC_FEATURE> application process started on module <MODULE_NO>
*STATIC_FEATURE DETECTED*
To be able to use the static feature please clear module MODULE_NO.
Use the fet utility to verify the feature was applied correctly.
```

In the output examples:

- `<DYNAMIC_FEATURE>` specifies the name of the dynamic feature file applied.
- `<STATIC_FEATURE>` specifies the name of the static feature file applied.
- `<MODULE_NO>` specifies the HSM that the feature was applied to.

5.1.6. Enable features on PCIe and USB HSMs

5.1.6.1. View enabled features

The **Feature Enable Tool** can be used to view the status of modules connected to the host or to confirm that a feature has been successfully enabled on all modules connected to the host. To view the status of features, run the tool without a smart card.

Some features do not appear in the default output from the **Feature Enable Tool** because they are no longer sold. To see the status of all features, run `fet --show-all`.

5.1.6.2. Enable features with a smart card

When it is launched, the **Feature Enable Tool** automatically scans the smart card readers of all modules attached to a host computer for any Feature Enabling smart cards present in the smart card readers, including imported Remote Operator slots and Dynamic Slots. However, feature enable smart cards do not work in Dynamic Slots.

To enable a new feature with a Feature Enabling smart card from Entrust:

1. Insert the Feature Enabling card from Entrust into a slot available to the module to be updated, excluding any Dynamic Slots.
2. Run the `fet` command-line utility to start the **Feature Enable Tool**.

A message is displayed if the features are enabled successfully. If you do not see this message confirming a successful upgrade, see Enable features without a smart card.

5.1.6.3. Enable features without a smart card

The **Feature Enable Tool** can also obtain the Feature Enabling Certificate information supplied by Entrust from a file or from the keyboard.

When you run the **Feature Enable Tool** without a Feature Enabling smart card in an HSM slot, a message similar to the following is displayed. There is a line for the features on each module, and a list of options.

In this example, only one module (`ESN 14BD-B089-E078`) is attached to the host.

```
                        Feature Enable Tool
                        ===================

                     ISO Smart Card Support
                     |  Remote Operator
                     |  |  Korean Algorithms
                     |  |  |  SEE Activation (EU+10)
                     |  |  |  |  SEE Activation (Restricted)
                     |  |  |  |  |  SEE Activation, CodeSafe 5
                     |  |  |  |  |  |  Elliptic Curve algorithms
                     |  |  |  |  |  |  |  Elliptic Curve MQV
                     |  |  |  |  |  |  |  |  Fast RNG for ECDSA
                     |  |  |  |  |  |  |  |  |  HSM Speed Rating
Mod   Electronic     |  |  |  |  |  |  |  |  |  |
No.  Serial Number
 1  14BD-B089-E078 -- Y  Y  Y  N  N  Y  Y  Y  Y  High Speed

0. Exit Feature Enable Tool.
1. Read FEM certificate(s) from a smart card or cards.
2. Read FEM certificate from a file.
3. Read FEM certificate from keyboard.
4. Write table to file.

Enter option :
```

When using the option to read the FEM certificate from a file you must either enter a fully qualified filename or alternatively run the `fet` command from the directory in which the FEM certificate is stored. You may also use `fet -c [FILENAME]` to specify the filename directly on the command line.

5.2. 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 of the platform services are administered by a unified utility called `hsmadmin`

5.2.1. hsmadmin

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

```
hsmadmin <subcommand>
```

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

5.2.1.1. 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.
`--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.
`--verbose`	Prints verbose logs.

5.2.1.2. 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.

Parameter	Description
`--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.

5.2.1.3. 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.

5.2.1.4. hsmadmin upgrade

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

```
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.

5.2.1.5. 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.

5.2.1.6. 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.

Linux-only

The install script calls this command automatically as `hsmadmin enroll --sshadmin-key /root/.ssh/id_nshield5_sshadmin`. This will generate SSH client keys and register them with the units if they have not been previously set up. If the `sshadmin` key is not found in its usual location under `/opt/nfast` then `/root/.ssh/id_nshield5_sshadmin` will be tried instead, so it is convenient to use `hsmadmin keys backup` to backup the key to this location.

If `hsmadmin enroll` is called after install to change the installed units, the hardserver will need to be restarted in order to pick up the configuration changes, for example, by running `/opt/nfast/sbin/init.d-ncipher restart`.

```
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.

5.2.1.7. 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

5.2.1.7.1. 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.

5.2.1.7.2. 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.

5.2.1.7.3. 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`.

5.2.1.7.4. 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.
`<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.

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.

5.2.1.7.5. 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`

5.2.1.7.6. 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

5.2.1.7.7. 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

5.2.1.8. hsmadmin logs

This command manages the system logs of connected HSMs. These logs are separate from the `ncoreapi` logs. See Platform services and (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

5.2.1.8.1. 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`

5.2.1.8.2. 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`

5.2.1.8.3. 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

5.2.1.8.4. 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.

5.2.1.8.5. 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.

5.2.1.9. 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.

5.2.1.10. 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.

5.2.1.11. 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.

5.2.1.12. 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.

5.2.1.13. hsmadmin getenvstats

This command returns the environmental monitoring statistics of the HSM.

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

hardware 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.

5.2.1.14. hsmadmin cs5

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

5.2.1.14.1. 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.

5.3. Client cooperation

You can allow an nShield HSM to automatically access the remote file system (RFS) belonging to another nShield HSM and share the Security World and key data stored in the Key Management Data directory. Client hardware security modules that access data in this way are described as *cooperating clients*.

5.3.1. Configure client cooperation

To configure client cooperation for other clients or hardware security modules that are not nShield HSMs:

1. Configure the RFS, or the RFS used by your nShield HSM, to accept access by cooperating clients:
   - For every authenticated client (with write access and K_{NETI} authorization) that needs to be a client of this remote file system, run the command:

     ```
     rfs-setup --gang-client <client_IP_address> <EEEE-SSSS-NNNN> <keyhash>
     ```

     In this command:

     - `<client_IP_address>` is the IP address of the client. This can be an IPv4 or IPv6 address.
     - `<EEEE-SSSS-NNNN>` is the ESN of the nToken used by the client when using a nToken K_{NETI} key to authenticate itself. When using software-based authentication, it must be empty (that, is "") or can be omitted altogether.
     - `<keyhash>` is the hash of the software or module K_{NETI} key used by the client.
   - For every unauthenticated client (with write access but without K_{NETI} authorization), run the command:

     Linux

     ```
     rfs-setup --gang-client --write-noauth <client_IP_address>
     ```

     Windows

     ```
     rfs-setup.exe --gang-client --write-noauth <client_IP_address>
     ```

     The `--write-noauth` option should be used only if you believe your network is secure. This option allows the client you are configuring to access the RFS without K_{NETI} authorization.

     To limit a gang-client to read-only, use the `--readonly` flag.

2. On each client that is to be a cooperating client, you must run the `rfs-sync` command-line utility with appropriate options:
   - for clients using a software K_{NETI} key to authenticate themselves to the RFS, run the command with the default options:

     ```
     rfs-sync --setup <RFS_IP_ADDRESS>
     ```

- for clients using a module K_{NETI} key to authenticate themselves to the RFS, run the command:

  ```
  rfs-sync --setup --authenticate --module=<MODULE> <RFS_IP_ADDRESS>
  ```

  In this command:

  - `<RFS_IP_ADDRESS>` is the IP address of the RFS.
  - `<MODULE>` is the local module to use for authentication.
- for clients to authenticate the RFS using software-based authentication, use the `--rfs-hkneti=HKNETI` option to specify the hash of the software K_{NETI} key of the RFS.
- for clients to authenticate the RFS using nToken authentication, use the `--rfs-esn=ESN` and `--rfs-hkneti=HKNETI` options to specify the `ESN` and hash of the K_{NETI} key of the nToken installed in the RFS.

The `rfs-sync` utility uses lock files to ensure that updates are made in a consistent fashion. If an `rfs-sync --commit` operation (the operation that writes data to the remote file system) fails due to a crash or other problem, it is possible for a lock file to be left behind. This would cause all subsequent operations to fail with a lock time-out error.

The `rfs-sync` utility has options for querying the current state of the lock file, and for deleting the lock file; however, we recommend that you do not use these options unless they are necessary to resolve this problem. Clients without write access cannot delete the lock file.

For more information about the `rfs-sync` utility, see rfs-sync

5.3.2. Remove a cooperating client

To remove a cooperating client so the RFS no longer recognizes it, you must:

- Know the IP address of the cooperating client that you want to remove
- Manually update the `remote_file_system` section of the hardserver configuration file by removing the following entries for that particular client:

  Linux

  ```
  remote_ip=<client_IP_address>
  remote_esn=keyhash=0000000000000000000000000000000000000000
  native_path=/opt/nfast/kmdata/local
  volume=kmdata-local
  allow_read=yes
  allow_write=yes
  allow_list=yes
  ```

```
is_directory=yes
is_text=no
```

and

```
remote_ip=<client_IP_address>
remote_esn=keyhash=0000000000000000000000000000000000000000
native_path=/opt/nfast/kmdata/local/sync-store/
volume=kmdata-backup
allow_read=yes
allow_write=yes
allow_list=yes
is_directory=yes
is_text=no
```

Windows

```
remote_ip=client_IP_address
remote_esn=keyhash=0000000000000000000000000000000000000000
native_path=%NFAST_KMDATA%\local
volume=kmdata-local
allow_read=yes
allow_write=yes
allow_list=yes
is_directory=yes
is_text=no
```

and

```
remote_ip=client_IP_address
remote_esn=keyhash=0000000000000000000000000000000000000000
native_path=%NFAST_KMDATA%\local\sync-store
volume=kmdata-backup
allow_read=yes
allow_write=yes
allow_list=yes
is_directory=yes
is_text=no
```

5.4. 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)

5.4.1. 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	...--
4	....-
5	.....
6	-....
7	--...
8	---..
9	----.
0	-----

5.4.2. 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.

5.4.3. 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
H L	. . . .	. - . .				M48T37 NVRAM (or battery) failed
H B	. . . .	- . . .				Debug serial port initialization failed.

Code						Meaning
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)
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.

Code						Meaning
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.
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.

Code						Meaning
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.

5.4.4. 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.

Code				Meaning	Action
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.

5.4.5. 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.
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.

Code				Meaning	Action
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.

5.4.6. 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
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

Code					Meaning	Action
WD2	. - -	- . .	. . - - -		Watchdog 2 failure	Contact Support
WD3	. - -	- . .	. . . - -		Watchdog 3 failure	Contact Support

5.4.7. 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 Entrust nShield Technical Support: nshield.support@entrust.com.

5.5. Working with CodeSafe

5.5.1. CodeSafe applications

To run CodeSafe applications on your system, you must have enabled the Secure Execution Engine (SEE) by purchasing and enabling an appropriate SEE activation licence as described in Optional features.

nShield 5c and 5s

You must also have loaded a valid certificate, known as a CodeSafe developer ID certificate, that can be used to verify the signature of each CodeSafe application that you want to run. These certificates will be supplied by the developer of the CodeSafe application and should be delivered to you together with the application.

CodeSafe developer ID certificates have a limited lifetime. If a certificate expires, applications that are already running will be unaffected but it will not be possible to start an application until a new certificate is loaded.

For this reason you should keep track of the expiry dates of your certificates and request a new certificate from the CodeSafe developer

> before your certificate expires.
>
> You can view the expiry date of certificates by using `csadmin ids list` and looking at the `notAfter` field for each certificate.
>
> New certificates can be loaded whilst an application is running without interrupting service.

If you want to develop your own CodeSafe applications, you must also purchase the CodeSafe developer kit.

An SEE application is typically a standalone SEE machine that is loaded automatically by the hardserver (for example, a CodeSafe C application).

Check the documentation that your CodeSafe application vendor supplies for information about how to set up and use the application, as well as for any other installation and configuration information.

CodeSafe applications are standalone applications, but each CodeSafe C application can consist of multiple parts, and its installation can include several configuration steps. For instructions on installing and configuring each application, see your application vendor's documentation.

nShield Solo HSMs: You may need to use the `hardserver`, `loadmache`, and `tct2` utilities when configuring and loading an application; see the *CodeSafe Developer Guide* for more information.

5.5.2. Use a standalone application (nShield Connect)

To use a standalone application:

1. Ensure that the SEE machine for the application is in the `/opt/nfast/custom-seemachines` (**Linux**) or `%NFAST_HOME%\custom-seemachines` (**Windows**) directory on the remote file system.

   If an SEE machine has previously been loaded on the HSM, press the Clear button on the front of the unit before proceeding to the next step. This clears the current SEE machine from memory.

2. From the main menu on the front panel of the HSM, select **CodeSafe**.
3. To enable the HSM to publish the SEE World for multiple clients, enter the following information when prompted:
   - The name of the SEE machine file.

- The name of the user data file, if required.
- The type of custom SEE machine you are using (select `SEElib` or `sockserv`).

This option is only available if you have provided a valid user data file in step 2. If `BSDlib sockserv` is selected, worldid_pubname, postload_prog, and postload_args will be passed to `load_seemachine`.

4. Log in to a host machine as a user in the `nfast` group and run the following command (*m1* is the Security World's module number for the HSM whose front panel you used in the previous steps):

```
sudo /opt/nfast/bin/nopclearfail -c -w -m1
```

For detailed descriptions of the options in this section, see load_seemachine.

- The ID of the SEE World to create.

This option is only available if you have selected the `SEElib` option in the previous step.

To use `see-sock-serv` directly, you must select `BSDlib sockserv`.

5.5.2.1. Remotely loading and updating SEE machines

The SEE remote push facility allows the remote deployment of CodeSafe SEE machines to an nShield HSM, negating the need to physically visit the HSM to load or update the SEE machine.

For instructions, refer to Remotely loading and updating SEE machines

5.5.3. CodeSafe setup for the nShield 5c

To use CodeSafe applications with the nShield 5c, you must exchange launcher service keys between the client and the nShield 5c by following the procedures in this section. You also have to enable port `2205` between the RFS and the network-attached HSM for `csadmin`.

To use multiple nShield 5c modules simultaneously from the same client machine, see Configure client to access multiple nShield 5c HSMs.

5.5.3.1. Initial Client Configuration: Retrieve or generate the client's public key

If the client's launcher key pair has already been generated, view the client's key type and public key data by issuing the following command:

```
hsmadmin cs5 clientinfo
```

The following is an example of the output expected for `clientinfo`:

```
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDmcFY2wxtNj+di478pvOK3czNeXJoqw+itu/Hw6KRcGjDO60Zqr8eaWJcDS7
9DHncvdUZv0ZPbLqTCnNjN0ECQ=
```

You must copy the key type public key. You will need these in a later step for setting the client info in the nShield HSM CLI.

If the client does not already have a launcher key pair, generate a new one on the client machine with the following command:

```
hsmadmin cs5 genclientinfo
```

And then view and copy the key type and public key as in the previous step.

5.5.3.2. Configure nShield5c for Client Authorization:

To configure the nShield HSM to deploy CodeSafe applications, you need access to its serial console. For more information, see Configuring an nShield HSM using the Serial Console.

1. Put the module into Maintenance mode (this will temporarily stop nCoreapi and launcher services):

   ```
   (cli)maintmode enable
   ```

   The module will not be able to perform any tasks and will appear unavailable until it is put back into Operational mode.

2. Remove any existing client information in the nShield HSM to ensure a clean setup:

   ```
   (cli)cs5 clearclientinfo
   ```

3. Set the client's public key for the launcher service by running the following command :

```
(cli)cs5 setclientinfo <key_type> <public_key>
```

Where <key_type> and <public_key> are the output from the hsmadmin cs5 clientinfo command that you copied earlier. For example:

```
(cli)cs5 setclientinfo ecdsa-sha2-nistp256
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ012345678
9ABCDEFGHIJKLMNOPQRSTUVWXY
```

4. Enable ports for Codesafe 5, using the nShield 5c's IP address:

```
(cli)cs5 ports enable <nshield-5c-ip-address>
```

Where <nshield-5c-ip-address> is the address on the HSM's network interface that CodeSafe 5 will access.

This prevents KeySafe 5 from being used until cs5 ports disable is run.

5. Retrieve all the information the client needs to connect to the remote launcher service (ESN, IP address, and launcher service public key):

```
(cli)cs5 getserverinfo
```

Copy the ESN, IP address, and launcher service public key for use in the final client configuration. The following is an example of the output expected for getserverinfo:

```
Codesafe 5 server info in format <ESN> <IP Addresss> <Port> <Key Type> <Public Key>
B0A7-B88C-D3F4 10.194.147.134 2205 ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBOEJMTeBR1546vmQ4YwT7WGlUv57PyCIShw66c9BAuD+WP5lrE8yjwJ
R/aBo7hrBxV95Vlgi6+4Up4fDI0uqO6s=
```

6. Put the module back into Operational mode:

```
(cli)maintmode disable
```

5.5.3.3. Finalize the Client Configuration:

On the client machine, enroll the nShield HSM for CodeSafe so that the client can connect to the remote launcher service with the following command:

```
hsmadmin cs5 enroll <ESN> <IP_Address> <Port> <Key_Type> <Public_Key>
```

Where ESN, IP_Address, Port, Key_Type and Public_key is the server information in the output for the cs5 getserverinfo command that you copied earlier.

You can verify the setup by running the following command:

```
csadmin list
```

The following is an example of the output expected when running csadmin list with no SEE machines installed:

```
B0A7-B88C-D3F4       No SEE machines currently installed
```

The nShield HSM is now ready for use with CodeSafe. For more information about building, signing, and running SEE machines with CodeSafe, see The csadmin utility tool (nShield 5 HSMs).

5.5.3.4. Configure client to access multiple nShield 5c HSMs

To use multiple nShield HSM HSMs from the same client you should configure them with the same launcher client key. To do this repeat the procedures in Configure nShield5c for Client Authorization: for each HSM using the same <key_type> and <public_key> obtained from cs5 clientinfo each time.

The information used in Finalize the Client Configuration: will be unique to each nShield HSM HSM and will be that returned by the cs5 getservinfo command for the individual HSM.

5.5.3.5. Removal of client configuration

When a client machine is finished with an nShield HSM, disconnect from it using:

```
hsmadmin cs5 unenroll <ESN>
```

Where <ESN> is the ESN of the HSM from which you wish to remove configuration.

Then disable ports for Codesafe 5:

```
(cli)cs5 ports disable
```

If you disable the ports and try to run csadmin list on the client you will notice that it will be unable to connect to the launcher.

The following is an example of the output expected when the ports are disabled:

```
% csadmin list
B0A7-B88C-D3F4        CONNECTION ERROR: Unable to connect to 'launcher': [Errno None] Unable to connect to port
2205 on 10.194.147.134
Failed to get SEE machines list for module(s): ['B0A7-B88C-D3F4']
```

5.5.4. The csadmin utility tool (nShield 5 HSMs)

The `csadmin` tool is used to manage CodeSafe 5 applications.

The `csadmin` tool covers CodeSafe application deployment from both the perspective of a CodeSafe application developer and a CodeSafe application user. The help text displays the complete set of commands available. This document lists the commands that are relevant to the CodeSafe application user.

You must be logged in as an Administrator or a user with local administrator rights to execute `csadmin` commands.

You must have `/opt/nfast/bin (Linux) or NFAST_HOME (Windows)` in your `PATH` environment variable to use `csadmin`.

To view the help text included here while using `csadmin`, run a command or sub-command with the `-h|--help` option.

The following examples use a Linux machine for the deployment of CodeSafe applications. The same commands can be applied to a Windows machine.

```
$ csadmin --help
usage: csadmin [-h] [-v] {image,load,start,stop,list,destroy,sshd,ids,log,config,stats} ...

Tool to manage nShield 5s CodeSafe.

positional arguments:
  {image,load,start,stop,list,destroy,sshd,ids,log,config,stats}
    image               Perform tasks related to loadable nShield 5s CodeSafe images
    load                Load a CodeSafe 5 image onto an nShield 5s HSM
    start               Start a previously loaded image on an nShield 5s HSM
    stop                Stop an SEE machine running on an nShield 5s HSM
    list                List the SEE machines loaded on an nShield 5s HSM
    destroy             Destroy an SEE machine loaded on an nShield 5s HSM
    sshd                Manage the SSH daemon dedicated to a specific SEE machine on an nShield 5s HSM
    ids                 Manage the CodeSafe Developer Authentication Certificates on an nShield 5s HSM
    log                 Manage the SEE machine logging state
    config              Manage the SEE machine configuration
    stats               Manage the SEE machine statistics

options:
  -h, --help            show this help message and exit
```

```
  -v, --version         Print csadmin version
```

5.5.4.1. Manage loadable CodeSafe images

CodeSafe 5 images have the extension `.cs5` and are often referred to as "CS5 files". Operations such as generating, signing, and inspecting loadable CodeSafe images are handled by `csadmin image`:

```
$ csadmin image --help
usage: csadmin image [-h] {generate,sign,inspect} ...

positional arguments:
  {generate,sign,inspect}
    generate            generate a loadable CodeSafe image.
    sign                sign a loadable CodeSafe image. Note: images are signed with an HSM.
    inspect             inspect a loadable CodeSafe image's metadata.

options:
  -h, --help            show this help message and exit
```

5.5.4.1.1. Generate loadable images

Only applicable to CodeSafe developers. See the *CodeSafe 5 Developer Guide*

5.5.4.1.2. Inspect images

The `csadmin` tool provides the `inspect` operation so developers can view the details of a `.cs5` file.

The `inspect` operation usage:

```
csadmin image inspect --help
usage: csadmin image inspect [-h] [--json] CS5FILE

positional arguments:
  CS5FILE     The cs5 file being inspected

optional arguments:
  -h, --help  show this help message and exit
  --json      Show details in json format
```

For example:

```
$ csadmin image inspect myapp.cs5
Type               codesafe-container
Platform           nShield5
PackageName        MyCodeSafeApp
Version            1.0
EntryPoint         /home/see/launch_see.sh
Format             1
```

5.5.4.1.3. Sign images

Only applicable to CodeSafe developers. See the *CodeSafe 5 Developer Guide*

5.5.4.2. Load a CodeSafe application

If you are using a third-party developed CodeSafe application, ensure that you trust the developer of the CodeSafe application, and verify that the image you are using has a genuine certificate issued by the trusted organization. You must have loaded a valid certificate prior to loading the application.

Loading a CodeSafe image onto the module creates the SEE container and the environment needed to run the application. This is done using `csadmin load`:

```
$ csadmin load --help

usage: csadmin load [-h] [--timeout TIMEOUT] [--esn ESN] [--verbose] [--json] CS5FILE

positional arguments:
  CS5FILE            The CodeSafe 5 image file to be loaded into the HSM(s).

optional arguments:
  -h, --help         show this help message and exit
  --timeout TIMEOUT  Time to wait for service response, in seconds. Default: 30s, must be between 3s and 120s
  --esn ESN          ESN of the HSM to load CodeSafe 5 image into
  --verbose          Print verbose logs
  --json             Show result output in json format
```

On success, the `load` sub-command returns the UUID of the new SEE container, for example:

```
$ csadmin load test_app_1.cs5

FEDC-BA09-8765        SUCCESS
UUID: c2e75658-04aa-4634-a7f8-fad3692dd9d7

$ csadmin load --json test_app_1.cs5
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "uuid": "fedcba09-8765-4321-1234-567890abcdef"
        }
    }
}
```

5.5.4.3. Start a CodeSafe application

Although multiple SEE machines can be loaded and started at the same time, only one can talk to the `ncoreapi` service.

A SEE machine will only start if a valid certificate is already loaded. Certificates have an expiry date and if the certificate has expired it will not be possible to start the SEE machine until a new certificate is loaded.

Starting an SEE machine previously loaded onto the HSM launches the container and executes the predetermined entry point. This is done using `csadmin start`:

```
$ csadmin start --help
usage: csadmin start [-h] [--timeout TIMEOUT] -u UUID [--esn ESN] [--verbose] [--json]

optional arguments:
  -h, --help            show this help message and exit
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 120s
  -u UUID, --uuid UUID  The UUID of the machine being started.
  --esn ESN             ESN of the HSM hosting the SEE machine to be started.
  --verbose             Print verbose logs
  --json                Show response in json format
```

The `start` command takes as main input the UUID of the machine to be started. On success, it returns the IP(v6) address of the container, for example:

```
$ csadmin start --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        SUCCESS
IP ADDRESS: ffff::fff:ffff:ffff:ffff

$ csadmin start --uuid fedcba09-8765-4321-1234-567890abcdef --json
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "ip": "ffff::fff:ffff:ffff:ffff"
        }
    }
}
```

To set the container to start automatically at startup and when the nShield 5 is reset, see Use set to enable autostart.

5.5.4.4. Stop a CodeSafe application

Stopping an SEE machine previously started on the HSM will gracefully terminate its programs execution and stop the container. This is done using `csadmin stop`. To display the sub-command usage, execute it with the `--help` option as shown below:

```
$ csadmin stop --help
usage: csadmin stop [-h] [--timeout TIMEOUT] -u UUID [--esn ESN] [--verbose] [--json]

optional arguments:
  -h, --help            show this help message and exit
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 120s
```

```
  -u UUID, --uuid UUID  The UUID of the machine being stopped.
  --esn ESN             ESN of the HSM hosting the SEE machine to be stopped.
  --verbose             Print verbose logs
  --json                Show response in json format
```

The `stop` command takes as main input the UUID of the machine to be stopped, for example:

```
$ csadmin stop --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        SUCCESS

$ csadmin stop --json --uuid fedcba09-8765-4321-1234-567890abcdef
{
    "FEDC-BA09-8765": {
        "succeeded": true
    }
}
```

5.5.4.5. Destroy a CodeSafe application

Destroying an SEE machine that is currently stopped on the HSM removes the application container and any user data created during the life of the machine. This is done using `csadmin destroy`:

```
$ csadmin destroy --help
usage: csadmin destroy [-h] [--timeout TIMEOUT] -u UUID [--esn ESN] [--verbose] [--json]

options:
  -h, --help            show this help message and exit
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 300s
  -u UUID, --uuid UUID  The UUID of the machine being destroyed.
  --esn ESN             ESN of the HSM hosting the SEE machine to be destroyed.
  --verbose             Print verbose logs
  --json                Show response in json format
```

5.5.4.6. List CodeSafe applications

`csadmin list` displays a list of the SEE machines currently loaded on the HSM:

```
$ csadmin list --help
usage: csadmin list [-h] [--timeout TIMEOUT] [--esn ESN] [--verbose] [--json]

optional arguments:
  -h, --help         show this help message and exit
  --timeout TIMEOUT  Time to wait for service response, in seconds. Default: 30s, must be between 3s and 120s
  --esn ESN          ESN of the HSM to retrieve SEE machines list for. Can be specified multiple times for
multiple HSMs.
  --verbose          Print verbose logs
  --json             Show response in json format
```

For example:

```
$ csadmin list
FEDC-BA09-8765
UUID                                  State     Name
------------------------------------------------------------------------
12345678-90ab-cdef-fedc-ba0987654321    STOPPED   nShield Docs Server
fedcba09-8765-4321-1234-567890abcdef    RUNNING   CodeSafe App 1
09876543-21fe-dcba-abcd-ef1234567890    RUNNING   CodeSafe App 2
```

5.5.4.7. Configure CodeSafe 5

`csadmin config` allows the user to set and query the state of CodeSafe's auto-start and logging features:

```
$ csadmin config --help
usage: csadmin config [-h] {list,get,set} ...

positional arguments:
  {list,get,set}
    list          List CodeSafe machine configuration.
    get           Get CodeSafe machine configuration.
    set           Set CodeSafe machine configuration.

options:
  -h, --help      show this help message and exit

configuration keys:
  "autostart": "enabled" or "disabled". Automatically start the SEE machine when the module resets. Disabled by
default.
  "log": "enabled" or "disabled". Capture a container's SEE logs. Disabled by default.
```

You must use a positional argument (`list`, `get`, `set`) with the `config` command.

5.5.4.7.1. List machine configuration

`csadmin config list` lists the current configuration keys and values of a given machine:

```
$ csadmin config list --help
usage: csadmin config list [-h] -u UUID [--esn ESN] [--verbose] [--json] [--timeout TIMEOUT]

options:
  -h, --help            show this help message and exit
  -u UUID, --uuid UUID  The UUID of the machine.
  --esn ESN             ESN of the HSM hosting the SEE Machine.
  --verbose             Print verbose logs
  --json                Show response in json format
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 300s
```

For example:

```
$ csadmin config list --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765
        autostart             : disabled
        log                   : disabled
```

```
$ csadmin config list --json --uuid fedcba09-8765-4321-1234-567890abcdef
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "autostart": "disabled",
            "log": "disabled"
        }
    }
}
```

5.5.4.7.2. Get machine configuration

`csadmin config get` queries the current value of a given machine's specific configuration key:

```
$ csadmin config get --help
usage: csadmin config get [-h] -u UUID [--esn ESN] [--verbose] [--json] [--timeout TIMEOUT] key

positional arguments:
  key                   Configuration key

options:
  -h, --help            show this help message and exit
  -u UUID, --uuid UUID  The UUID of the machine.
  --esn ESN             ESN of the HSM hosting the SEE Machine.
  --verbose             Print verbose logs
  --json                Show response in json format
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 300s
```

For example:

```
$ csadmin config get log --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        disabled

$ csadmin config get log --json --uuid fedcba09-8765-4321-1234-567890abcdef
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "log": "disabled"
        }
    }
}
```

5.5.4.7.3. Set machine configuration

`csadmin config set` sets a given machine's specified configuration key to the specified value:

```
$ csadmin config set --help
usage: csadmin config set [-h] -u UUID [--esn ESN] [--verbose] [--json] [--timeout TIMEOUT] key value

positional arguments:
  key                   Configuration key
```

```
  value                 Configuration value

options:
  -h, --help            show this help message and exit
  -u UUID, --uuid UUID  The UUID of the machine.
  --esn ESN             ESN of the HSM hosting the SEE Machine.
  --verbose             Print verbose logs
  --json                Show response in json format
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 300s
```

Use set to enable logging

```
$ csadmin config set log enabled --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765
        state                 : enabled

$ csadmin config set log enabled --json --uuid fedcba09-8765-4321-1234-567890abcdef
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "state": "enabled"
        }
    }
}
```

Use set to enable autostart

If Autostart is enabled the SEE machine will automatically start after a reset of the HSM.

Before starting, the system will check that a valid certificate matching the application has been loaded. If the certificate has expired the SEE machine will not start.

For this reason it is important that you check the expiry dates on your certificates and request new certificates from the CodeSafe developer for any certificates that are near their expiry date. You can check the expiry date on certificates by using `csadmin ids list` and looking for the `notAfter` field.

```
$ csadmin config set autostart enabled --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765
        state                 : enabled

$ csadmin config set autostart enabled --json --uuid fedcba09-8765-4321-1234-567890abcdef
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "state": "enabled"
        }
    }
}
```

5.5.4.8. Manage CodeSafe 5 logging

When logging is enabled (see Use set to enable logging), the standard output and standard error of all programs running on the container is saved into a log file that can be pulled and cleared by `csadmin log`:

```
$ csadmin log --help
usage: csadmin log [-h] {get,clear} ...

positional arguments:
  {get,clear}  manage a container's SEE logging operations (get/clear)
    get        Get a container's SEE log
    clear      Clear SEE logs on modules

options:
  -h, --help   show this help message and exit
```

You must use a positional argument (`get`, `clear`) with the `csadmin log` command.

5.5.4.8.1. Get CodeSafe 5 logs

To obtain CodeSafe 5 logs, use `csadmin log get`:

```
$ csadmin log get --help
usage: csadmin log get [-h] --uuid UUID [--json] [--timeout TIMEOUT] [--esn ESN] [--verbose]

options:
  -h, --help            show this help message and exit
  --uuid UUID, -u UUID  The UUID of the machine to get the log from
  --json                Show response in json format
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 300s
  --esn ESN             ESN of the HSM to manage SEE logging for.
  --verbose             Print verbose logs
```

For example:

```
$ csadmin log get --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        SUCCESS
Success: Started ipcdaemon

$ csadmin log get --json --uuid fedcba09-8765-4321-1234-567890abcdef
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "contents": "Success: Started ipcdaemon\n"
        }
    }
}
```

The internal log file has a maximum size of 128KB. When this size is reached, the log is cleared and a new log is started. This deletes previous data from the log. There is no rotation of the log.

5.5.4.8.2. Clear CodeSafe 5 logs

To clear CodeSafe 5 logs, use `csadmin log clear`:

```
$ csadmin log clear --help
usage: csadmin log clear [-h] --uuid UUID [--json] [--timeout TIMEOUT] [--esn ESN] [--verbose]

options:
  -h, --help            show this help message and exit
  --uuid UUID, -u UUID  The UUID of the machine to clear/delete the log from
  --json                Show response in json format
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 300s
  --esn ESN             ESN of the HSM to manage SEE logging for.
  --verbose             Print verbose logs
```

For example:

```
$ csadmin log clear --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        SUCCESS
log: log cleared

$ csadmin log clear --json --uuid fedcba09-8765-4321-1234-567890abcdef
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "state": "log cleared"
        }
    }
}
```

5.5.4.9. Obtain machine-specific statistics

`csadmin stat` provides a mechanism for obtaining runtime data for CodeSafe machines on the system. It can only obtain data from machines in the `RUNNING` state:

```
$ csadmin stats --help
usage: csadmin stats [-h] [--timeout TIMEOUT] [-u UUID] [--esn ESN] [--verbose] [--json]

options:
  -h, --help            show this help message and exit
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 300s
  -u UUID, --uuid UUID  The UUID of the machine to query stats from.
  --esn ESN             ESN of the HSM hosting the SEE machine to query stats from.
  --verbose             Print verbose logs
  --json                Show response in json format
```

When executed without arguments, the stats command displays the runtime data of every machine currently running on the system:

```
$ csadmin stats
FEDC-BA09-8765
UUID                                   State     Name                      IP Address                 PID
CPU           Memory          KMem             Link        Tx bytes     Rx bytes    Total bytes
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
12345678-90ab-cdef-fedc-ba0987654321    RUNNING   hello               ffff::fff:ffff:ffff:eeee           1393
0.49 seconds    1.20 MiB       848.00 KiB        vethM19WNi    1.12 KiB      8.04 KiB      9.15 KiB
fedcba09-8765-4321-1234-567890abcdef    RUNNING   hello               ffff::fff:ffff:ffff:ffff            1610
0.42 seconds    1.26 MiB       836.00 KiB        vethsE1Pv4    1.12 KiB      4.98 KiB      6.10 KiB

$ csadmin stats --json
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "cs_stats": {
                "12345678-90ab-cdef-fedc-ba0987654321": {
                    "Name": "12345678-90ab-cdef-fedc-ba0987654321",
                    "State": "RUNNING",
                    "PID": "1393",
                    "CPU use": "0.51 seconds",
                    "Memory use": "1.24 MiB",
                    "KMem use": "888.00 KiB",
                    "Friendly Name": "hello",
                    "IP": "ffff::fff:ffff:ffff:eeee",
                    "Logging enabled": "False",
                    "Link": {
                        "name": "vethM19WNi",
                        "TX bytes": "1.12 KiB",
                        "RX bytes": "8.30 KiB",
                        "Total bytes": "9.42 KiB"
                    }
                },
                "fedcba09-8765-4321-1234-567890abcdef": {
                    "Name": "fedcba09-8765-4321-1234-567890abcdef",
                    "State": "RUNNING",
                    "PID": "1610",
                    "CPU use": "0.44 seconds",
                    "Memory use": "1.24 MiB",
                    "KMem use": "864.00 KiB",
                    "Friendly Name": "hello",
                    "IP": "ffff::fff:ffff:ffff:ffff",
                    "Logging enabled": "False",
                    "Link": {
                        "name": "vethsE1Pv4",
                        "TX bytes": "1.12 KiB",
                        "RX bytes": "5.24 KiB",
                        "Total bytes": "6.36 KiB"
                    }
                }
            }
        }
    }
}
```

When executed with the `--uuid|-u` argument, the command only displays the data of the specified machine:

```
$ csadmin stats --uuid fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765
UUID                                    State     Name                      IP Address                 PID
CPU           Memory          KMem              Link        Tx bytes     Rx bytes    Total bytes
-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
fedcba09-8765-4321-1234-567890abcdef    RUNNING   hello               ffff::fff:ffff:ffff:ffff           1610
0.48 seconds    1.21 MiB       716.00 KiB        vethsE1Pv4    1.12 KiB      5.76 KiB      6.88 KiB

$ csadmin stats --uuid fedcba09-8765-4321-1234-567890abcdef --json
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "cs_stats": {
                "fedcba09-8765-4321-1234-567890abcdef": {
                    "Name": "fedcba09-8765-4321-1234-567890abcdef",
                    "State": "RUNNING",
                    "PID": "1610",
                    "CPU use": "0.49 seconds",
                    "Memory use": "1.39 MiB",
                    "KMem use": "840.00 KiB",
                    "Friendly Name": "hello",
                    "IP": "ffff::fff:ffff:ffff:ffff",
                    "Logging enabled": "False",
                    "Link": {
                        "name": "vethsE1Pv4",
                        "TX bytes": "1.12 KiB",
                        "RX bytes": "6.03 KiB",
                        "Total bytes": "7.15 KiB"
                    }
                }
            }
        }
    }
}
```

5.5.4.10. Manage container SSHD

The container SSHD allows connections to be forwarded from the client machine to the container. Its keys and state can be queried and managed with `csadmin sshd`:

```
$ csadmin sshd --help
usage: csadmin sshd [-h] [--timeout TIMEOUT] [--esn ESN] {keys,state} ...

positional arguments:
  {keys,state}
    keys             manage a container's SSH daemon keys
    state            manage a container's SSH daemon state

optional arguments:
  -h, --help         show this help message and exit
  --timeout TIMEOUT  Time to wait for service response, in seconds. Default: 30s, must
                     be between 3s and 300s
  --esn ESN          ESN of the HSM to manage SSHD daemon for.
```

5.5.4.10.1. Manage SSHD keys

The public keys used to connect to the container SSHD can be managed with `csadmin sshd keys`:

```
$ csadmin sshd keys --help
usage: csadmin sshd keys [-h] {getserver,getclient,setclient} ...

positional arguments:
  {getserver,getclient,setclient}
    getserver           Obtain a container's SSH daemon keys
    getclient           Obtain a container's SSH daemon client keys
    setclient           Set a container's SSH daemon client keys

optional arguments:
  -h, --help            show this help message and exit
```

Retrieve keys

The SSHD's public key and the client keys set by the user can be retrieved by the `getserver` and `getclient` sub-commands respectively:

```
$ csadmin sshd keys getserver -h
usage: csadmin sshd keys getserver [-h] --uuid UUID [--json]

optional arguments:
  -h, --help            show this help message and exit
  --uuid UUID, -u UUID  The UUID of the machine to retrieve SSH daemon keys for
  --json                Show response in json format
```

`getserver` returns the server public keys for the container specified by the given UUID. `getclient` returns the client public key set by the user.

For example, `getserver` produces the following output, shown in plain and json format:

```
$ csadmin sshd keys getserver -u fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        SUCCESS
KEY_TYPE: ecdsa-sha2-nistp256
KEY_BODY: AAAA.....................w4wtS4=

$ csadmin sshd keys getserver -u fedcba09-8765-4321-1234-567890abcdef --json
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "server_key": {
                "key_type": "ecdsa-sha2-nistp256",
                "key_body": "AAAA.....................w4wtS4="
            }
        }
    }
}
```

Set client keys

The client public key that is used to connect to the SSHD server on the HSM is set by the user with `csadmin sshd keys setclient`:

```
$ csadmin sshd keys setclient --help
usage: csadmin sshd keys setclient [-h] --uuid UUID
                                   (--keyfile KEYFILE | --keytype KEYTYPE KEYBODY)

optional arguments:
  -h, --help            show this help message and exit
  --uuid UUID, -u UUID  The UUID of the machine to set the SSH daemon keys for
  --keyfile KEYFILE, -f KEYFILE
                        File containing the public key body and type (a .pub file in
                        PEM format)
  --keytype KEYTYPE KEYBODY, -t KEYTYPE KEYBODY
                        Keytype of the public key (ex. ecdsa-sha2-nistp256) followed by
                        keybody
```

The key can be set using a keyfile (`--keyfile`) or by manual text entry (`--keytype`).

- Keyfile:

  When using the `--keyfile` option, the user passes in the path to the keyfile to be used. This is the public (`.pub`) key.

  Accepted public key types are `ecdsa-sha2-nistp521` and `ecdsa-sha2-nistp256`.

  For example:

  ```
  $ csadmin sshd keys setclient -u fedcba09-8765-4321-1234-567890abcdef --keyfile ~/test_key.pub
  ```

  Where `test_key.pub` is a key in OpenSSH format containing the key type and key body, for example:

  ```
  ecdsa-sha2-nistp256 AAAAE.....................5s8=
  ```

- Manual text entry:

  When using the `--keytype option`, the user manually enters the key type followed by the key body.

  For example:

  ```
  $ csadmin sshd keys setclient -u fedcba09-8765-4321-1234-567890abcdef --keytype ecdsa-sha2-nistp256
  TestKeyBody
  ```

5.5.4.10.2. Manage the SSHD state

The user can query and manage the container SSHD state using `csadmin sshd state`:

```
$ csadmin sshd state --help
usage: csadmin sshd state [-h] {enable,disable,get} ...

positional arguments:
  {enable,disable,get}
    enable              Enable a container's SSH daemon
    disable             Disable a container's SSH daemon
    get                 Get a container's SSH daemon state

optional arguments:
  -h, --help            show this help message and exit
```

Enable the container SSHD

The container SSHD will only start when the container has been enabled. By default, it is disabled. It can be enabled with `csadmin sshd state enable`:

```
$ csadmin sshd state enable --help
usage: csadmin sshd state enable [-h] --uuid UUID [--json]

optional arguments:
  -h, --help            show this help message and exit
  --uuid UUID, -u UUID  The UUID of the machine to enable SSH daemon for
  --json                Show response in json format
```

If you enable the SSHD while the container is not running, it will start when the container starts, otherwise it starts immediately.

The `enable` subcommand also returns the port and address on which the SSHD is listening, for example (output in plain text and in JSON format):

```
$ csadmin sshd state enable -u fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        SUCCESS
SSHD PORT: 3006
LISTENING ADDRESS: aaaa::aa:aaaa:aaaa:aaaa

$ csadmin sshd state enable -u fedcba09-8765-4321-1234-567890abcdef --json
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "sshd_port": "3006",
            "listening_address": "aaaa::aa:aaaa:aaaa:aaaa"
        }
    }
}
```

Disable the container SSHD

When the container SSHD state is disabled, the container SSHD will not start. This is the default state. It can also be disabled with `csadmin sshd state disable`:

```
$ csadmin sshd state disable --help
usage: csadmin sshd state disable [-h] --uuid UUID

optional arguments:
  -h, --help            show this help message and exit
  --uuid UUID, -u UUID  The UUID of the machine to disable SSH daemon for
  --json                Show response in json format
```

If the container sshd is running, the disable command will also stop it. It will not start again until is has been enabled.

Retrieve the container SSHD state

The user can also query the SSHD state with `csadmin sshd state get`. This displays the current SSHD state (enabled or disabled):

```
$ csadmin sshd state get --help
usage: csadmin sshd state get [-h] --uuid UUID [--json]

optional arguments:
  -h, --help            show this help message and exit
  --uuid UUID, -u UUID  The UUID of the machine to get the SSH daemon state for
  --json                Show response in json format
```

If the SSHD is disabled, the output only returns the state:

```
$ csadmin sshd state get -u fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        SUCCESS
STATE: disabled

$ csadmin sshd state get -u fedcba09-8765-4321-1234-567890abcdef --json
{
    "FEDC-BA09-8765": {
        "succeeded": true,
        "data": {
            "state": "disabled"
        }
    }
}
```

If the SSHD is enabled, the output returns the port and address that the SSH server on the HSM is listening on, as well as the port on the container where connections should be forwarded to:

```
$ csadmin sshd state get -u fedcba09-8765-4321-1234-567890abcdef
FEDC-BA09-8765        SUCCESS
STATE: enabled
SSHD_LISTEN: 3001
CONTAINER_FORWARD: 9000
LISTENING ADDRESS: aaaa::aa:aaaa:aaaa:aaaa

$ csadmin sshd state get -u fedcba09-8765-4321-1234-567890abcdef --json
{
    "FEDC-BA09-8765": {
```

```
        "succeeded": true,
        "data": {
            "state": "enabled",
            "ports": {
                "sshd": "3001",
                "container": "9000"
            },
            "listening_address": "aaaa::aa:aaaa:aaaa:aaaa"
        }
    }
}
```

The container port is supplied by the user in `network-conf.json`.

5.5.4.10.3. Forward to the container SSHD

To establish port forwarding on the host machine, use the SSH local port forward command. Do this after the container SSHD is enabled, started, and its keys are set. Use of a secure tunnel over an open TCP connection to communicate with the container is recommended, so use local port forwarding instead dynamic port forwarding to enable SSH tunneling:

```
ssh -L local_port:destination_server_ip:remote_port ssh_server_hostname
```

The full details of the command:

```
ssh -i [PATH_TO_PRIV_KEY] -L [::1]:[LOCAL_PORT]:[CONTAINER_IP_ADDR%lxcbr0]:[CONTAINER_PORT] -f -N -p [SSHD_PORT]
launcher@[LISTENING_ADDRESS]%nshield0
```

Where:

- `PRIV_KEY` is the private key to the public client key that was set with `setclient`.
- `LOCAL_PORT` is the port on the local client where traffic to be forwarded will be sent.
- `CONTAINER_IP_ADDR` is the IP address of the container. This is the address returned when container is started.
- `CONTAINER_PORT` is the port on the container that is listening for forwarded traffic. This is set in `network-conf.json` as "ssh_tunnel" (valid range: 1024-65535).
- `SSHD_PORT` is the port the SSH daemon is listening on. This port is returned when SSHD is enabled and can also be found with `csadmin sshd state get`.
- `LISTENING_ADDRESS` is the address the host clients use to communicate with the HSM (on the HSM side). This address is returned when SSHD is enabled and can also be found with `csadmin sshd state get`.

For example:

```
ssh -i ~/test_ssh_keys/test_key -L [::1]:8888:[ffff::fff:ffff:ffff:ffff%lxcbr0]:8888 -f -C -N -p 6789
launcher@aaaa::aa:aaaa:aaaa:aaaa%nshield0
```

In preparation, there must also be some endpoint in the container that listens on the specified container port, for example, a socket. An example simple python socket script is shown below:

```
import socket
import sys

HOST = ''
PORT = 8000 #the container port set in the network conf

soc = socket.socket(socket.AF_INET6, socket.SOCK_STREAM)

try:
    soc.bind((HOST, PORT))
except socket.error as message:
    print('Bind failed. Error code: '
        + str(message[0]) + ' Message '
        + message[1])
    sys.exit()

print('Socket binding complete')

soc.listen(9)

conn, address = soc.accept()

with conn:
    print('Connected by', address)
    while True:
        data = conn.recv(1024)
        if not data: break
        conn.send(data)

soc.close()
```

After the SSH instance is established, any traffic sent to the local port on the local client should be forwarded to the container on the container port. Because of the port forwarding restrictions set for the container SSHD in the `authorized_keys` file, attempts to forward to any port on the container other than the one specified in `network_conf.json` will be rejected.

5.5.4.11. Manage identities

Identities are needed in order to load and run CodeSafe applications. Developer identities (certificates) are managed with `csadmin ids`:

```
$ csadmin ids --help
usage: csadmin ids [-h] [--timeout TIMEOUT] [--esn ESN] {add,remove,get,validate,list,create} ...

positional arguments:
  {add,remove,get,validate,list,create}
```

```
    add                 Add a CodeSafe developer identity certificate to the nShield5s HSM
    remove              Remove a CodeSafe developer identity certificate from the nShield5s HSM
    get                 Get a CodeSafe developer identity certificate from the nShield5s HSM
    validate            Validate a CodeSafe developer identity certificate on the nShield5s HSM
    list                List all of the CodeSafe developer identity certificates on the nShield5s HSM
    create              Create a CSR for a named developer id key. Also create the developer id key if it does
not already exist

options:
  -h, --help            show this help message and exit
  --timeout TIMEOUT     Time to wait for service response, in seconds. Default: 30s, must be between 3s and 300s
  --esn ESN             ESN of the HSM to manage developer identities certificates for.
```

The command allows end users to add new identities onto the HSM, remove existing identities from the HSM, retrieve (`get`) identities from the HSM, list identities presently installed on the HSM, and validate a specific identity present on the HSM.

5.5.4.11.1. Create a developer ID certificate

Only applicable to CodeSafe developers. See the *CodeSafe 5 Developer Guide*

5.5.4.11.2. List identity certificates

List identity certificates using `csadmin ids list`:

Listing certificates allows you to check their expiry dates by looking for the `notAfter` field for each certificate. You should request new certificates from the CodeSafe developer for any certificates that have expired or are near their expiry date. Entrust recommend that you remove any expired certificates using `csadmin ids remove`.

```
$ csadmin ids list --help
usage: csadmin ids list [-h] [--json] [--verbose]

options:
  -h, --help  show this help message and exit
  --json      Show response in json format
  --verbose   Print verbose logs
```

5.5.4.11.3. Add identities

Add identity certificates using `csadmin ids add`:

```
$ csadmin ids add --help
usage: csadmin ids add [-h] [--verbose] CERTFILE

positional arguments:
  CERTFILE    The developer identity certificate file to be added into the HSM(s).

options:
```

```
  -h, --help  show this help message and exit
  --verbose   Print verbose logs
```

5.5.4.11.4. Remove identities

Remove identity certificates using `csadmin ids remove`:

Entrust recommend that you remove any expired certificates so that only valid certificates remain in the system.

```
$ csadmin ids remove --help
usage: csadmin ids remove [-h] -s SERIALNO [--verbose]

options:
  -h, --help            show this help message and exit
  -s SERIALNO, --serialno SERIALNO
                        Serial number of the certificate in base ten
  --verbose             Print verbose logs
```

5.5.4.11.5. Retrieve identities

Retrieve identity certificates using `csadmin ids get`:

```
$ csadmin ids get --help
usage: csadmin ids get [-h] -s SERIALNO [--json] [--verbose]

options:
  -h, --help            show this help message and exit
  -s SERIALNO, --serialno SERIALNO
                        Serial number of the certificate in base ten
  --json                Show response in json format
  --verbose             Print verbose logs
```

5.5.4.11.6. Validate identities

Validate identity certificates using `csadmin ids validate`:

```
$ csadmin ids validate --help
usage: csadmin ids validate [-h] -s SERIALNO [--verbose]

options:
  -h, --help            show this help message and exit
  -s SERIALNO, --serialno SERIALNO
                        Serial number of the certificate in base ten
  --verbose             Print verbose logs
```

5.6. Warrant Management

5.6.1. Warrant management for nShield Solo and nShield Edge

You must use a Windows machine to manage warrants for nShield Edge HSMs.

This appendix describes how you can ensure that a suitable warrant is available to allow an nShield Remote Administration Card to be used with nShield Solo and Edge HSMs. To be able to use an nShield Remote Administration Card you need to ensure that:

- The appropriate firmware is installed on the nShield Solo or Edge HSM. (Firmware 2.61.2 or later)
- The nShield Solo or Edge HSM has a KLF2 warrant installed in the appropriate place.

5.6.1.1. Warranting steps for nShield Solo and nShield Edge

You need an appropriate support contract to obtain a KLF2 warrant from Entrust.

Ensure v12.xx Security World Software has been installed on your host computer (to access the `nfwarrant` tool) and the nShield Solo or Edge HSM has Firmware 2.61.2 firmware or later installed.

You then need to carry out the following steps to ensure a suitable warrant is available

1. Check if the relevant module has the appropriate firmware.
2. Check if a warrant upgrade is required, if so, follow steps 3-6.
3. Generate a Certificate Signing Request (CSR) for the warrant.
4. Send the CSR to Entrust.

Ensure that the ESN contained in the upgrade request is the one that belongs to the relevant module, for example, by running the nfkminfo command-line utility.

5. Validate the warrant that you receive from Entrust to ensure that it matches the sent request.
6. Install the warrant.

5.6.1.2. nfwarrant command-line utility

The `nfwarrant` command-line utility enables you to carry out all of the relevant warrant steps. It is used to:

- Identify modules that have the appropriate firmware and KLF2 key

- Identify modules that need their KLF2 key to be warranted by Entrust
- Generate a warrant upgrade request for a specific module, as required
- Install an upgraded warrant
- List KLF2 warrants

Usage:

```
nfwarrant [--help] [--list] [--check] [--warrant] [--csr] [--details= FILE]
[--install= FILE] [--req= MODULE] [--out= FILE] [--verbose] [--version]
```

Options:

Option	Description
-h|--help	Displays the options you can use with the utility.
--list	List ESNs of installed warrants
--check	List ESNs of known modules and their warrant state
--warrant	Perform warrant operations
--csr	Perform CSR operations
--details=<FILE>	Display the module ESN found in the CSR/warrant *<file>*
--install=<FILE>	Install the warrant from *<file>*
--req=<MODULE>	Request a warrant CSR for the given module number/ESN
--out=<FILE>	Save the new requested CSR to *<file>*
--verbose	Print extra information about CSR and warrant files
--version	Print the version number of the nfwarrant tool

5.6.1.2.1. Check the available hardware

```
$ nfwarrant --check
```

The following is an example output:

```
1 XXXX-XXXX-E0D2 Local, Warrant installed
2 XXXX-XXXX-CF11 Local, Warrant upgrade request possible
3 XXXX-XXXX-F1F2 Local, Warrant upgrade not supported
4 XXXX-XXXX-213B Remote, Warrant upgrade not required
```

In this example:

- (1) already has a relevant warrant installed.
- (2) is available for a warrant upgrade.
- (3) cannot be upgraded. For example, the appropriate firmware is not installed.
- (4) no warrant upgrade is required. The module is an nShield Connect.

5.6.1.2.2. Generate a warrant upgrade request for nShield Solo

Run the following command:

```
$ nfwarrant --csr --req <module>
```

The following is an example output, displaying the location of the resultant upgrade request for a module with ESN XXXX-XXXX-CF11:

```
CSR written to 'C:\ProgramData\nCipher\Key Management Data\warrants\csr_XXXX-XXXX-CF11'
```

Ensure that the ESN in this request file is the correct one and send the file to Entrust to be signed.

5.6.1.2.3. Validate the warrant you receive from Entrust

1. Run the following command:

   ```
   $ nfwarrant --warrant --details <file>
   ```

   The following is an example output:

   ```
   Warrant details: Filename: XXXX-XXXX-CF11 ESN: XXXX-XXXX-CF11 Keytype: ECDSAPublic Curve: NISTP521
   ```

2. Compare the ESN in the file received from Entrust with the one in the original request, by running the following command:

   ```
   $ nfwarrant --csr --details <file>
   ```

   The following is an example output:

   ```
   XXXX-XXXX-CF11
   ```

5.6.1.2.4. Install a warrant for nShield Solo

Run the following command:

```
$ nfwarrant --warrant --install <file>
```

<file> is the signed warrant provided by Entrust.

5.6.2. Warrant management for nShield Connect + and nShield Connect XC

You do not need to manage the warrants for nShield Connect HSMs.

When you start or reboot the HSM, the warrant is copied to the appropriate location on the host or RFS:

- **Linux:** `/opt/nfast/kmdata/hsm-<ESN>/warrants`
- **Windows:** `C:\ProgramData\nCipher\Key Management Data\hsm-<ESN>\warrants`

Where `<ESN>` is the ESN of the relevant module.

When you configure the RFS for an nShield Connect, rfs-setup creates the required directory structure. If you cannot find a warrant at this location, re-run `rfs-setup`` to ensure that the RFS is configured correctly, and then reboot the HSM.

5.6.3. Warrant management for nShield 5s and nShield 5c

You do not need to manage the warrants for nShield 5s. Entrust supplies these HSMs with the required warrants pre-installed and stored within the module. The Security World software fetches warrants from the module when they are needed.

This includes a KLF2 and a KLF3 warrant. The KLF3 warrant is currently unused and is installed in preparation for multi-tenant systems.

To view the warrants installed on a module, run `retrievewarrants`. This stores a copy of the warrants in the host file system.

5.7. Remote HSMs

A *remote HSM* is an HSM that is not connected directly to the host computer but with which the hardserver can communicate.

There is no specific limit to the total number of remote HSMs enrolled to a security world. However:

- The number of PCIe and USB-attached remote HSMs in a security world is limited by the PCIe slots and USB connections the client machine can support.
- There can be a maximum of 500 network-attached remote HSMs in a security world.

A remote HSM can be one of the following:

- A network-connected nShield HSM that is configured to use the host computer as a client computer
- An HSM to which an attended Remote Operator slot is imported for the hardserver's unattended local HSM

  (Remote Operator feature only).

For configuration instructions, see Configure remote HSM connections.

5.8. 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.

5.8.1. 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.

5.8.2. Configuring Remote Operator

This section explains how to configure Remote Operator.

5.8.2.1. 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.

5.8.2.2. 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
```

5.8.2.3. Configuring slot import and export

For information about the parameters controlled by the hardserver configuration file, see HSM and client configuration files (network-attached HSMs) or Hardserver configuration files (PCIe and USB HSMs).

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.

5.8.2.3.1. 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.

5.8.2.3.2. 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-BBBB-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.

5.8.2.4. 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 module 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.
  See the *HSM and client configuration files* chapter in the *User guide* for your HSM for more about module configuration file.
- 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.

5.8.2.5. 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.

5.8.2.6. 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
```

5.8.2.7. 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 unattended 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 cannot remap any of its imported slots.

5.8.3. 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.

5.8.3.1. Creating OCSs for use with Remote Operator

You can generate Remote OCSs by using KeySafe or 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 by using KeySafe or the `replaceocs` command-line utility.

5.8.3.2. 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.

5.8.3.3. Generating keys for use with Remote Operator

After you have created a Remote OCS, to generate keys protected by it you can run KeySafe or the `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.

KeySafe can list imported slots, but cannot use them.

If you already have an OCS-protected key that you want to use, but the protecting OCS is not a Remote OCS, you can use KeySafe to protect the key under a new Remote OCS if the key was originally generated with the key recovery option enabled.

However, if the key was not generated with key recovery enabled, you cannot protect it under a different OCS. In such a case, you must generate a new key to be protected by a Remote OCS.

5.8.3.4. 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.

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

5.8.3.5.1. 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.

5.8.3.5.2. 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.

5.9. Using nShield commands from PowerShell

PowerShell is a powerful console tool for scripting operations on Windows. nShield applications can be run from PowerShell, locally or remotely, interactively or non-

interactively (batch mode). nShield library code implements a set of commands for reading text and passphrases.

5.9.1. Install and configure PowerShell

1. Install PowerShell, see https://docs.microsoft.com/en-us/powershell/.
2. Ensure that executing PowerShell scripts is enabled in the system.

   Script execution is not enabled by default on Windows clients. The default permissions usually allow script execution on Windows Server operating systems, but it may be necessary to enable this in a custom Windows Server configuration.

   Open PowerShell and set the signing property and scope of script execution.

   The support files for running nShield commands from PowerShell are Authenticode-signed, so the execution can be restricted to only signed scripts:

   ```
   Set-ExecutionPolicy -ExecutionPolicy AllSigned
   ```

   If unsigned PowerShell scripts are to be executed, you may want to relax this so that locally created scripts can be run without signing:

   ```
   Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
   ```

   The above commands can be run with the additional parameter `-Scope CurrentUser` to restrict the changes to the currently logged-in user. For example to permit the current user to run locally-created scripts, run:

   ```
   Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
   ```

3. By default, PowerShell commands are created for all executables in `$env:NFAST_HOME\bin`. If there are additional directories containing nShield executables that you wish to include in the nShield PowerShell module support, you can specify those directories with a semicolon separated list of paths in the `NC_PS_ADDITIONAL_DIRECTORIES` environment variable.

   ```
   $env:NC_PS_ADDITIONAL_DIRECTORIES="C:\Path1;C:\Path2"
   ```

4. To load support for nShield commands in PowerShell, import the `nShieldTools.psd1` module located in `$env:NFAST_HOME\bin`.

```
Import-Module 'C:\Program Files\nCipher\nfast\bin\nShieldTools.psd1'
```

The support module is installed in the active shell.

5. To load the `nShieldTools.psd1` module automatically every time PowerShell is opened, you can add it to your PowerShell profile with the `Add-nShieldToProfile` command, included in the `nShieldTools.psd1` module.

Parameter	Description
`AllHosts`	(Recommended) Profile is available for all PowerShell hosts, for example both ConsoleHost and ISE, rather just the currently running host.
`AllUsers`	Profile is available for all users on the local machine rather than just the current user.

Example:

```
Add-nShieldToProfile -AllHosts
```

5.9.2. Calling nShield commands at the PowerShell prompt

nShield commands can be called with their usual names from PowerShell, for example `enquiry`, `cardpp`, `generatekey`, or `new-world`. Aliases for the `.exe` variants, for example `new-world.exe`, are also registered so that the nShield executables in the PATH are only called with PowerShell support.

Do not call the executables directly (for example, do not call & 'C:\Program Files\nCipher\nfast\bin\new-world.exe') because this will not enable the PowerShell support.

Command-line parameters to nShield PowerShell commands can be provided in the same way as the corresponding command under regular Windows consoles. If you can run `cardpp --check` in a regular Windows console, you can run `cardpp --check` in PowerShell.

The global variable `$LASTEXITCODE` of PowerShell contains the exit code (0 for success) immediately after execution of the nShield command.

5.9.3. PowerShell modes: interactive and batch

nShield commands can be run in either interactive or batch (non-interactive) mode.

In the default **interactive mode**, the output is displayed incrementally on the screen in the

PowerShell host user interface. UI prompts are displayed when the command attempts to read user input, for example passphrases or confirmations. Output is not written to a PowerShell pipeline in this mode.

Batch mode is intended for automation. Commands can be run from a script and an output can be redirected to a file. Batch mode does not prompt for input in the host UI. Input can be supplied programmatically with a PowerShell input pipeline. If input is needed but no suitable pipeline was supplied, the command fails rather than stall program execution to wait for user interaction. Standard output and standard error text printed by the underlying nShield program is written to output and error pipelines that can be redirected or piped. If the program fails, that is, it returns a non-zero exit code, the error code is also thrown as an exception that appears in the error pipeline. In batch mode, output and error texts are visible only at the very end of execution, because such texts are objects that the command returns to the pipeline instead of writing them incrementally to the host UI.

If the mode is not explicitly set, PowerShell normally defaults to interactive mode. However, if any input pipeline objects are supplied, PowerShell defaults to batch mode. You can therefore switch to batch mode without setting it explicitly simply by piping `$null`:

```
$null | enquiry > enquiry.txt
```

You can change the default mode to batch mode by setting the `NC_PS_INTERACTIVE` environment variable:

```
$env:NC_PS_INTERACTIVE=0
```

Mode change commands provided in the `nShieldTools.psd1` module:

Command	Notes
`Set-nShieldBatchMode`	
`Set-nShieldInteractiveMode`	
`Reset-nShieldCommandMode`	Can be used to restore the default PowerShell behavior based on presence or absence of pipeline input.

To restrict a setting to a particular PowerShell scope, you can use the PowerShell variable `$nShieldInteractiveCommandMode`, which can be set to `$True` or `$False`.

5.9.4. Input pipelines

In both interactive and batch mode, nShield commands support input pipelines with the

PowerShell pipe ("|") syntax. The input pipeline can be used to automate the execution of nShield commands that would otherwise have to prompt for user input. For example, a passphrase check on an OCS card can be performed automatically by executing the following command:

```
Set-nShieldBatchMode
$passphrase | cardpp --check
```

`$passphrase` is a variable in the command or script, and contains the card's passphrase.

Multiple values can be supplied to provide the input to successive prompts. For example, the `generatekey` command can be automated to provide passphrases for operator cards, softcards, or administrator cards. In the following example, the passphrase variables are passed to the input pipeline, and the remaining key generation parameters are passed on the command-line:

```
PS C:\temp\test> $acs_passphrase, $softcard_passphrase | generatekey --batch pkcs11 protect=softcard
softcard=mysoftcard plainname=mykeyname nvram=yes
key generation parameters:
operation    Operation to perform              generate
application  Application                       pkcs11
module       Module to use                     1
protect      Protected by                      softcard
slot         Slot to read cards from           0
softcard     Soft card to protect key          mysoftcard
recovery     Key recovery                      yes
verify       Verify security of key            yes
type         Key type                          RSA
size         Key size                          2048
pubexp       Public exponent for RSA key (hex)
logkeyusage  Log key usage                     no
plainname    Key name                          mykeyname
nvram        Blob in NVRAM (needs ACS)         yes

Load Admin Card (for KNV):
Module 1 slot 0: Admin Card #1
Module 1 slot 0: Enter passphrase:
Module 1 slot 0:- passphrase supplied - reading card
Module #1 Slot #0: Processing ...
Card reading complete.

Please enter the passphrase for softcard 'mysoftcard':
Please wait........
Key successfully generated.
Path to key: C:\ProgramData\nCipher\Key Management Data\local\key_pkcs11_uce586891...
```

5.9.5. Secure strings

A passphrase or other sensitive string can be read into a variable in PowerShell using `$passphrase = Read-Host -AsSecureString`. `$passphrase` in this case is an instance of `System.Security.SecureString` and not the `System.String` type used for normal strings. The contents of a `SecureString` cannot be read directly. If print the value of `$passphrase` in

PowerShell, you only see the type name displayed and not the value that was entered to the `Read-Host -AsSecureString` prompt.

nShield commands support using `SecureString` instances both for the input pipeline and as parameters to the nShield command. This helps reduce the visibility of plaintext passphrases or other sensitive values in scripts or in the shell. This is useful when using the input pipeline to automate the presentation of passphrases to the prompts in card-loading commands. It also means that nShield commands that take a passphrase as a command-line parameter can be presented that string without the string becoming directly visible.

Example:

```
ppmk --new --newpp $passphrase newsoftcard
```

5.10. Preload Utility

5.10.1. Overview

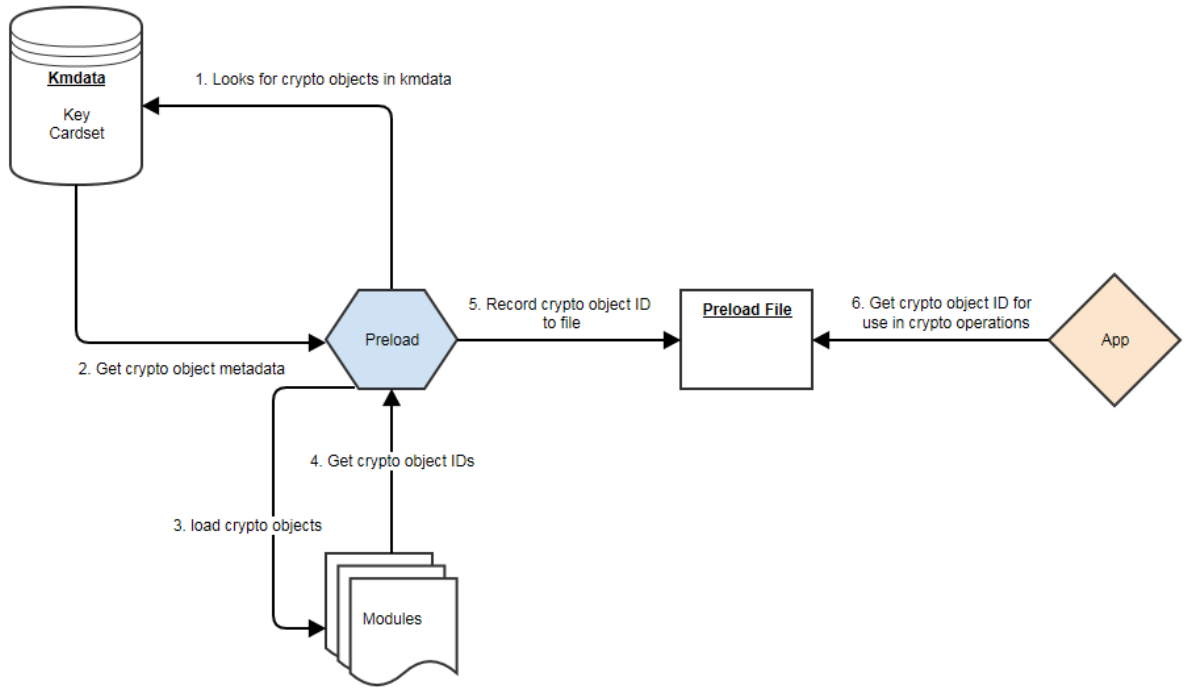
The preload utility loads persistent cryptographic objects (keys/OCS/softcards) onto a chosen set of modules, then makes those objects available for use by applications. This removes the need for applications to load keys/cards themselves, and allows for easy sharing of keys/cards between multiple applications. Additionally, `preload` can manage keys, such that they are reloaded/maintained on modules to provide high availability.

Preloading is achieved via keys/cardsets being loaded, then once loaded the IDs of these objects are recorded persistently to a file (the preload file), which can be read via another application sharing the same session, and subsequently used.

Keys/cardsets must have previously been created before they can be preloaded, and all modules participating in a preload session must be in the same security world.

The preload binary can be found in `/opt/nfast/bin` (**Linux**) or `%NFAST_HOME%\bin` (**Windows**). This binary calls the `preload.py` script found in `/opt/nfast/python/scripts` (**Linux**) or `%NFAST_HOME%\python\scripts` (**Windows**).

The image below shows the relationship between preload, modules and applications:



5.10.2. Using Preload

5.10.2.1. Preload Commands

A command is needed in order to run preload. This command needs to be specified after the preload arguments.

The purpose of this command is to decide what needs to be done after preload has found and loaded all its crypto objects (OCS/softcards/keys).

```
> preload [arguments] command
```

Preload has a choice of 3 commands:

1. `pause` - continue to run the preload process forever. This is useful to load keys in one session and use them in another.
2. `exit` - exit preload gracefully. This is useful to add keys to the preload session. Not available in combination with high availability mode.
3. `subprocess` - execute this subprocess and exit once the subprocess has finished

The only exception to this is the `--list-admin` option that does not require a command.

The preload session remains open, and thus the preloaded keys remain loaded, as long as at

least one instance of preload continues to run. If/when the final preload instance terminates, all loaded objects will be cleaned up.

Example showing a single key, of type simple, being loaded and then an application being launched:

```
> preload -A simple -K key1 myapplication.py
```

5.10.2.2. Preload file location

The environment variable `NFAST_NFKM_TOKENSFILE` holds the path to the preload file. If it is not set, then the default location is used. A non-default location can also be set via the --preload-file option when invoking preload.

5.10.2.3. Preload Command Line Arguments

Argument	Effect
`--version`	show program's version number and exit
`-h|--help`	show help message and exit
`-m MODULE_NUMBER| --module=MODULE_NUMBER`	Load on specified module (may be repeated; default = all).
`-c IDENT|--cardset=IDENT`	Load all cardsets matching `IDENT`. If `IDENT` looks like a hash it will be interpreted as that, otherwise it will be interpreted as a name. If it is definitely a name, use `--cardset-name`.
`--cardset-name=NAME`	Load cardset(s) named `NAME`.
`-s IDENT|--softcard=IDENT`	Load all softcards matching `IDENT`. If `IDENT` looks like a hash it will be interpreted as that, otherwise it will be interpreted as a name.
`--softcard-name=NAME`	Load softcard(s) named `NAME`.
`-o|--any-one`	Load a single cardset.
`-i|--interactive`	Load cardsets interactively until told to stop.
`-A APP|--appname=APP`	Choose the `appname` for subsequent `-K` options.
`-K IDENT|--key-ident=IDENT`	Load keys with ident matching `IDENT`.
`-n PATTERN|--name-pattern=PATTERN`	Load keys with name matching `PATTERN`. Use * for wildcard.
`--name-exact=NAME`	Load keys with name `NAME`.
`-M|--module-prot`	Load all module protected keys, in addition to any others requested.

Argument	Effect
`--no-cardset-keys`	Do not automatically load keys protected by requested cardsets. Deprecated.
`--no-token-keys`	Do not automatically load keys protected by requested tokens.
`--admin=KEYS`	Load admin keys (separate with commas, or use `all`).
`--list-admin`	List available admin key names (for `--admin`).
`-F|--require-fips`	Require FIPS-auth to be loaded.
`-N|--no-fips`	Do not record FIPS auth, even if available. (overrides `-F`).
`-H|--high-availability`	High availability mode.
`--polling-interval=POLLING_INTERVAL`	Interval (s) between polls for changes to the module list (default=60). High availability mode only.
`-f PRELOAD_FILE|--preload -file=PRELOAD_FILE`	Use specified preloaded objects file, instead of the default.
`-R|--reload-everything`	Reload keys and tokens that are already loaded.
`--show-key-info`	Display key information for keys as they are loaded.
`-l|--file-logging`	Log to file.
`-S|--no-stderr-logging`	Do not log to `stderr`, this is independent of file logging.
`--log-file=LOG_FILE`	The file destination for the log, defaults to `preload_%pid.log` in the `nfast` log directory.
`--log-level=LOG_LEVEL`	The log level to log, options: `DEBUG`, `INFO`, `WARNING`, `ERROR`. Default is `INFO`, if unrecognized option it will fall back to default.

5.10.2.4. Pattern Matching

Options to preload that use pattern matching, namely `--name-exact` and `--key-ident`, can accept the following wildcards:

Wildcard	Definition
`*`	matches everything
`?`	matches any single character
`[seq]`	matches any character in seq
`[!seq]`	matches any character not in seq

It is advised that all arguments that using wildcards are surrounded by quotations to ensure

that they are passed to preload as intended. For example, to load all keys whose names start with `keyname`, the following pattern could be used:

```
> preload --name-pattern 'keyname*' exit
```

5.10.3. Preload File

The IDs of preloaded crypto objects are persistently stored in a preload file.

Each entry has the following format:

Element	Description
`Hash`	The `sha1 hash` of the crypto object.
`module`	The module which this object is present.
`objectid`	The id reference as a `M_KeyID`.
`generation`	This element is reserved for internal use.

Example `nfkminfo` output with preloaded crypto objects:

```
Pre-Loaded Objects ( 10):  objecthash    module     objectid   generation
c29da3ac0d99a7c01477831ac31a4bebe283c4f8      1   0xac57be2e            1
c29da3ac0d99a7c01477831ac31a4bebe283c4f8      2   0xac57be2d            1
1080cca2be9588e6e47bcd870ebcbb133ea0561b      1   0xac57be2c            1
1080cca2be9588e6e47bcd870ebcbb133ea0561b      2   0xac57be13            1
```

By default the preload file location is `/tmp` (**Linux**) or the current user's temporary folder (**Windows**):

Linux

```
/tmp/nfpriv_<username>/default
```

Windows

```
<current user's temporary folder>\nfpriv_<username>\default
```

This location can be changed by using the command line option `-f PRELOAD_FILE|--preload-file=PRELOAD_FILE`.

5.10.4. Softcard Support

Softcards are now supported in preload, along with module protected keys and OCS

cardsets.

In order to preload a softcard and the corresponding keys being protected by said softcard the `-s` or `--softcard-name` arguments can be used.

The `-s` option can be used with the softcard name or the hash of the softcard:

```
> nfkminfo -s
...
Operator logical token hash               name
3768b8efb7c7324dd8a1edbe2650c2015281c877  test
```

```
> nfkminfo -k simple aes128simplesoftcard1
...
name                  "aes128simplesoftcard1"
hash                  07c8110498dc0315455457f25564fc288c7da304
...
softcard              3768b8efb7c7324dd8a1edbe2650c2015281c877
```

```
> preload -s test nfkminfo
...
Pre-Loaded Objects ( 4):  objecthash   module objectid  generation
07c8110498dc0315455457f25564fc288c7da304   1 0xa411c0ab 1
07c8110498dc0315455457f25564fc288c7da304   2 0xa411c09e 1
3768b8efb7c7324dd8a1edbe2650c2015281c877   1 0xa411c09d 1
3768b8efb7c7324dd8a1edbe2650c2015281c877   2 0xa411c0a0 1
```

This shows the softcard is loaded on modules 1 and 2. It additionally shows that the key protected by the softcard has been loaded on both modules.

5.10.4.1. No Cardset Keys

The `--no-cardset-keys` command line option can also be used for softcards.

This command line option will ensure that only the softcard is preloaded, and no keys protected by that cardset:

```
> preload -s test --no-cardset-keys
...
Pre-Loaded Objects (  2):  objecthash   module objectid   generation
3768b8efb7c7324dd8a1edbe2650c2015281c877     2 0xa9ba32a9 1
3768b8efb7c7324dd8a1edbe2650c2015281c877     1 0xa9ba32aa 1
```

5.10.5. FIPS Auth

FIPS Auth can be made available via preload.

The command line `-F` will ensure FIPS auth is preloaded everywhere.

The command line `-N` will ensure FIPS auth is not recorded, and will negate `-F`.

FIPS auth is also an admin key, see Admin Key section for more information.

5.10.6. Admin Keys

5.10.6.1. Listing

Admin keys can be listed using the `--list-admin` command line option.

This should be run without a command:

```
> preload --list-admin
```

Available admin keys are `NSO`, `M`, `RA`, `P`, `NV`, `RTC`, `FIPS`, `MC`, `RE`, `DSEE`, `FTO`.

5.10.6.2. Loading

Admin keys can be loaded using the `--admin=KEYS` command line option, supplying the value `--admin=ALL` to load all available admin keys. Note that admin key loading will require an ACS card being present in a slot of each module that is to be used.

Also note that the logical token of the admin key is preloaded alongside the key itself, for example, kfips and ltfips.

5.10.7. High Availability

Preload provides a high availability mode. When this mode is invoked Preload will load all requested keys, and will then periodically check for modules added or removed from the security world, or for keys becoming unloaded on existing modules. Should old or new modules be found to not have the specified keys/cardsets loaded, then preload will attempt to load them. This ensures that all available/usable modules have the requested keys loaded at all times, available for use by applications. Merged keyIDs are used to ensure applications can continually use these keys without interruption or changing key IDs. Preloaded keys are not only available to one application, but to any/all applications that share the preload session.

When preload is invoked with the `--high-availability` or `-H` option, it does the following differently:

1. Whenever preload loads a key onto the HSMs, it creates a Merged Key to represent the

set of (HSM, key ID) pairs. Applications will then use these merged IDs to address the keys.
    - As discussed below, this in itself provides failback, resilience and increased availability: the Merged Key ID remains usable even if some HSMs fail or are removed from the security world.
2. For as long as preload is running, it does the following repeatedly, once per polling interval:
    - Consult the hardserver to get a list of operational HSMs which are in the relevant security world.
    - For each Merged Key that was loaded by this instance of preload:
    - Ensure there is a valid current entry for each usable HSM.
    - To achieve this, check HSMs and load (or re-load) keys onto them as necessary, and update Merged Key contents.
    - Ensure that the individual key IDs within each Merged Key are valid: Remove any that are no longer valid and usable (such as those for a removed HSM).
    - Update the preload file to reflect changes, if any.
    - When finished, sleep for an interval of time, then repeat.

In summary, this mode attempts to keep preloaded crypto objects present on all usable modules in a security world (or a set of modules if requested via the -m argument) for as long as preload is running, with a keyID that remains constant, so that keys are available for use by any applications sharing the preload session.

5.10.7.1. Prerequisites for high availability mode

Users should not mix and match instances of preload with and without the `-H` high availability option, if those instances are sharing a session.

Managing OCS cardset-protected keys requires the following:

- the OCS protecting the key(s) be a 1/N quorum
- the passphrase for each card of the OCS set be identical
- one card of the OCS set be left inserted in a slot (local or remote) for each module
- if the card is non-persistent, it must be left in a local slot.

5.10.7.2. Differences from legacy behaviour

When running in high availability mode, certain behaviours of may differ from those outside of high availability mode. This includes the prompts for PIN entry and error messages. This

is due to a necessary difference in implementation between the two modes, and is expected.

5.10.7.3. Conditions for Management/Reloading

As mentioned above, preload in high availability mode will (re)load keys onto modules when a module is usable. A module will be considered usable if that module is in operational mode and in the correct world (and in the case of OCS protected keys, if a card from the OCS set is inserted into the module, locally or remotely). Preload will not attempt to perform actions that involve world administration, such as world loading or client enrolment. Users are responsible for managing worlds and client enrolment, and thus for bringing modules into a usable state.

The automatic loading/reloading of keys onto usable modules is not to be confused with forced reloading of keys provided by the -R option.

5.10.7.4. Merged Keys in the Preload File

When high availability mode is activated, all keys are represented in the preload file as Merged keys; cardsets and softcards are represented in the same way as non-high-availability mode.

Due to the fact that in high availability mode keys are represented as MergedKeys, which do not correspond to any one particular module, the module element of the preload file is no longer relevant for keys. However, for cardsets, the module field is still utilized.

For symmetric and private halves of asymmetric keys the module number is represented as a `-1` and for public halves of asymmetric keys the module number is represented as a `-2`.

This is evident in the output from `nfkminfo`. (Note that `nfkminfo` ignores the 32-bit two's-complement representation, thus displaying -1 and -2 as (2^{32} - 1) and (2^{32} - 2) respectively: 4294967295 and 4294967294):

```
Pre-Loaded Objects (  4):  objecthash   module objectid  generation
84749a62d0f71db7f80c5df6469c11685f7f1b78   1 0xb5c0c7fa 1
84749a62d0f71db7f80c5df6469c11685f7f1b78   2 0xb5c0c7fd 1
28dcee51dfc53387f4dc4d55538d8b5253ee85d1 4294967295 0xb5c0c7f7 1
c2afe833ae6e823a37777c633a5b3a18a9e5dfbd 4294967294 0xb5c0c7f8 1
```

As shown above, cardsets/softcards are still module specific.

To make `nfkminfo` show the preloaded objects, run it as a `subprocess` as part of the `preload` command. See the section above on using preload.

Merged Key IDs (just like single-key IDs) are shared between multiple instances of preload that are invoked by the same client, that is, using the same `ClientID`. As such, applications must ensure that they perform no operations that delete or replace the merged key ID, or alter the keys that are part of that merged key ID.

5.10.7.5. Polling Interval

Preload manages its crypto objects by polling available modules, based on a polling interval.

Once per interval, if preload detects modules (new or existing) without the relevant crypto objects (keys/cards) present, it will attempt to load those missing objects.

This polling interval is configurable via the command line option `--polling-interval=SECONDS`

By default the polling interval is 60 seconds.

5.10.7.6. Key timeouts and use limits

It is advised to not use OCSs or keys with timeouts in high availability mode, as preload will be unable to reload objects once their timeouts have expired.

In high availability mode, there are situations where OCS/keys that have previously timed out, or reached maximum use limits, may be reloaded (and thus their limits reset) without user interaction. In general within high availability mode keys that have timed out or reached their use limits will be left in place, unusable, respecting the limits. However if the module containing those keys reboots or resets then, upon the module's return, preload will notice that the keys are not loaded and will load them. This reloading of keys will necessarily reset timeouts and use limits. If the timeout on an OCS has reached its limit, any keys protected by that OCS will not be reloaded on newly-indoctrinated modules in the security world.

5.10.7.7. Multiple Preload instances in high availability mode

As described above, keys will be maintained by the preload instance that first introduces them, and will cease being maintained when that instance ends. (Here maintained means reloaded automatically onto relevant HSMs that lack them.)

Therefore when `preload` is invoked with `exit` (or a short-lived `subprocess` command) it will load the specified keys but then exit, leaving those keys unmaintained.

If a `preload` process is already running under high availability mode, any new preload process (with the same preload file) will gain access to the preloaded keys. As such that later instance must also be run in high availability mode (and preload will reject an attempt to run it in plain mode in this situation).

The `pause` command may be useful for setting up availability of keys for subsequent use by multiple applications:

First, a long-running preload instance to load keys and maintain them indefinitely:

```
$ preload --high-availability [...other options...] pause
```

Then run applications (possibly short-lived) that use those keys:

```
$ preload --high-availability [...other options...] app --args --for --app
```

5.10.7.7.1. Managing Keys

Given multiple preload processes run under high availability, the process that will manage the keys is the first process to find them, based on command line options.

For example, Security World crypto objects:

crypto object	name	protected by
Softcard	softcard1	N/a
Key	simple_softcard1	softcard1
Key	simple_module1	module

First preload process started:

```
> preload -H -s softcard1 pause
```

This would load the softcard `softcard1` on all modules as well as the key `simple_softcard1`:

```
> preload -H nfkminfo
...
Pre-Loaded Objects (  3):  objecthash   module objectid  generation
29235f2a0b77fc1e18641b0820fe3c93e030a02e 4294967295 0x44313d41 1
5bccb6f540802ef1da3828f6b8b0f3fc985272e6   2 0x44313d47 1
5bccb6f540802ef1da3828f6b8b0f3fc985272e6   1 0x44313d46 1
...
> nfkminfo -k simple simplesoftcard1
...
name                 "simple_softcard1"
hash                 29235f2a0b77fc1e18641b0820fe3c93e030a02e
```

```
...
> nfkminfo -s
...
Operator logical token hash               name
5bccb6f540802ef1da3828f6b8b0f3fc985272e6  softcard1
```

Second preload process started:

```
> preload -H -n simple pause
```

This would load the key `simple_module` on all modules:

```
> preload -H nfkminfo
...
Pre-Loaded Objects (  4):  objecthash   module objectid  generation
600bcc26336c13f2371bdbb54b1cde293ded9a15 4294967295 0x44313d29 1
29235f2a0b77fc1e18641b0820fe3c93e030a02e 4294967295 0x44313d41 1
5bccb6f540802ef1da3828f6b8b0f3fc985272e6   2 0x44313d47 1
5bccb6f540802ef1da3828f6b8b0f3fc985272e6   1 0x44313d46 1
...
> nfkminfo -k simple simplemodule1
...
name                 "simple_module1"
hash                 600bcc26336c13f2371bdbb54b1cde293ded9a15
```

The evidence that the first preload process is still managing the key `simple_softcard1`, even though the second preload process could have loaded it, is in the `objectid`.

The object id for key `simple_softcard1` has not changed (**0x44313d41**).

5.10.7.8. FIPS Auth in High Availability mode

Fips auth can be preloaded when running preload in high availability mode. In this scenario fips auth will be loaded as a high availability key (ie, reloaded/maintained on modules, as with other preloaded keys).

To enable FIPS auth use the command line option `-F`.

However, note that fips auth is represented differently, in comparison to other high availability mode keys, within the preload file.

The FIPS auth key is represented in the preload file multiple times: once for each module it is loaded on, and one extra time with a negative module ID as with other merged IDs. However the `objectid` is still a Mergedkey so will remain the same across those entries. This duplication of entries is to maintain compatibility with legacy behaviour/applications.

The following shows the pre-loaded FIPS auth objects on an estate of 3 modules - note there are 4 entries, each with the same `objectid`:

```
Pre-Loaded Objects (  4):  objecthash   module objectid  generation
aa462d0dd9dfeaa80968aadda2610ac0f6f94352   3 0xa824b9ab 1
aa462d0dd9dfeaa80968aadda2610ac0f6f94352   2 0xa824b9ab 1
aa462d0dd9dfeaa80968aadda2610ac0f6f94352   1 0xa824b9ab 1
aa462d0dd9dfeaa80968aadda2610ac0f6f94352 4294967295 0xa824b9ab 1
...
hkfips      aa462d0dd9dfeaa80968aadda2610ac0f6f94352
```

5.10.7.9. PKCS #11 and JCE

Both PKCS #11 and JCE applications are compatible with the high availability mode of preload, provided the PKCS #11 or JCE library that the application uses is from the 12.60 release or later. Flags or environment variables only need to be set to enable this when PKCS #11 is used for key reloading.

5.10.7.9.1. Use PKCS #11 for key reloading

PKCS #11 key reloading requires preload to be run in high availability mode, with the following options enabled:

- `--high-availability`.
- `--no-token-keys`.
- `--preload-file=PRELOAD_FILE`, where PRELOAD_FILE must match the location given to PKCS #11 with the NFAST_NFKM_TOKENSFILE environment variable.
- Either `--cardset=<IDENT>` or `--softcard=<IDENT>` (depending on whether using card set or softcard protected keys), where `<IDENT>` is the identifier of the card set or softcard, respectively.

  PKCS #11 key reloading is also supported for module-protected keys, but the PKCS #11 application must still be run under a preload application that is reloading tokens for another key.

  Using preload in high availability mode with Operator Card Sets has a set of restrictions, see Overview.

- Additionally, the following option is not required, but recommended:

  `--polling-interval=<POLLING_INTERVAL>`, where `<POLLING_INTERVAL>` also determines how often PKCS #11 will attempt to reload keys. The default is 60 seconds.

For more information, see PKCS#11 with key reloading.

5.10.7.10. Unsupported options

The `-H --high-availability` option may not be used in conjunction with any of the following options:

- `-o --any-one`
- `-i --interactive`
- `exit`
- `--admin`
- `--reload-everything`

5.10.8. Logging

By default `preload` logs to `stderr`.

Logs follow the format: `yyyy-mm-dd hh:mm:ss: [pid]: LogLevel: message`

For example:

```
2019-03-27 09:45:50: [439]: INFO: loading objects
```

Preload can also log to a file, this behaviour is separate from `stderr` logging. Therefore we can disable logging or log to `stderr` and/or a file.

To disable `stderr` logging, use the command line option `-S`. To enable file logging use the command line option `-l`.

The default file location for logs is `/opt/nfast/logs/preload_log_pid.log` (**Linux**) or `%NFAST_HOME%\logs\preload_log_pid.log` (**Windows**).

To change the file location, use the command line option `--log-file=FILE`.

As standard, preload has different log levels. These are:

- `DEBUG`
- `INFO`
- `WARNING`
- `ERROR`
- `CRITICAL`

The log level is by default: `INFO` and can be changed via the command line option `--log-level=LEVEL`.

5.10.9. Using preloaded objects - Worked example

In order to use preloaded objects, an application needs to create a connection that reads in the preload file:

Python:

```
import nfkm

conn = nfkm.connection(existingobjects="") # Reads file from default location

# If no existingobjects parameter is specified,
# the connection will not attempt to read any preload file:
conn_no_preload = nfkm.connection()
```

If the `existingobjects` argument is the empty string, the connection will use the file from the default location.

Any other string should be a path to different preload file. It can then call `NFKM_GetInfo` to get the security world info:

Python:

```
world_info = nfkm.getinfo(conn)
```

This results in a data structure with all the preloaded objects (this list is static and created at the time of connection creation):

Python:

```
import nfkm

conn = nfkm.connection(existingobjects="")

world_info = nfkm.getinfo(conn)

print world_info.existingobjects
```

Result:

```
[
ExistingObjectInfo
.module= 2
.hash= 84749a62 d0f71db7 f80c5df6 469c1168 5f7f1b78
.change= 1
.id= 0xffffffff88afd208,
ExistingObjectInfo
.module= 1
.hash= 84749a62 d0f71db7 f80c5df6 469c1168 5f7f1b78
.change= 1
.id= 0xffffffff88afd20b
```

```
]
```

Once an application has the M_KeyID references, it can use those cryptographic objects:

```
objid = world_info.existingobjects[0].id

cmd = nfkm.Command(["GetLogicalTokenInfo", 0, objid])

print conn.transact(cmd)
```

Result:

```
Reply.cmd= GetLogicalTokenInfo
.status= OK
.flags= 0x0
.reply.state= Present
.hkt= 84749a62 d0f71db7 f80c5df6 469c1168 5f7f1b78
.shares= empty
.sharesneeded= 0
```

5.11. Audit Logging

5.11.1. Aims of audit logging

Audit Logging on nShield HSMs provides the means to log administrative operations and key usage events across your estate of HSMs.

Some events are always logged and some events are logged only if certain conditions are met. Details of which events are logged, the conditions for logging the event, and the main information in those logs is described in Commands audited.

Logs are accurately time-stamped to allow correlation of these logs with other systems that may form part of your security or network environments.

Logs are signed and can be verified by the tools provided so that any attempts to alter the contents of the logs can be detected.

Audit logging is a compliance requirement of some certification schemes such as Common Criteria but can optionally be used with any Security World. The option must be specified when the World is created and cannot be added later.

5.11.1.1. Audit logging and CEF audit logging

There are two different audit log formats which differ depending on the firmware loaded on the HSM.

HSMs loaded with firmware versions prior to 13.5 will produce audit logs in CEF format and this guide refers to the production of logs in that format as 'CEF audit logging'.

HSMs loaded with firmware versions of 13.5 or later will produce audit logs in a new format and this guide refers to the production of logs in this format as 'audit logging'.

For a description of CEF audit logging and how to manage it, see the following v13.4 guides:

- Audit Logging (nShield Connect v13.4.5 User guide)
- Audit Logging (nShield Solo v13.4.5 User guide)
- Audit Logging (nShield Edge v13.4.5 User guide)

Since CEF was the only format of audit logs available prior to the release of Security World 13.5 older versions of the User Guides refer to CEF audit logging simply as 'audit logging'.

If your Security World uses a mix of HSMs with firmware before and after 13.5 you will receive logs in both formats. These logs are essentially independent and you must manage them separately.

5.11.2. Configuring audit logging

5.11.2.1. Enabling audit logging

Audit Logging is enabled on an HSM when it is added to a Security World that is configured for audit logging.

Since accurate time-stamps are important for audit logging Entrust recommends that you ensure that the system clock has been accurately set before enabling audit logging. See Manage the system clock of an nShield 5s for help managing the system clock on nShield 5s HSMs.

The Security World is configured for audit logging when it is created. This can be done in one of two ways:

- Specifying the `--audit-logging` or `-G` option in the `new-world` command
- Specifying the `--mode=common-criteria-cmts` option in the `new-world` command

For the overall procedure, see Creating a Security World using new-world.

The `--mode=common-criteria-cmts` option will create a Security World supporting Common Criteria PP 419 221-5 which imposes other restrictions not related to audit logging. Only use this option if you

> require compliance to Common Criteria

There are some differences in the conditions that must be met for particular events to be audited depending on which of the two options above has been used to create the Security World. These differences are explained in Commands audited.

5.11.2.2. Disabling audit logging

Audit Logging is set for the lifetime of the Security World.

To disable Audit Logging on an HSM:

1. Reinitialize the HSM using `initunit`. See Erasing a module with initunit

> Before removing the HSM from the Security World you should check that you have exported and verified all the logs that you require. Once the HSM has been erased it may not be possible to obtain and verify any logs that had not yet been exported.

5.11.2.3. Key usage logging

By default Audit Logging does not log usage of keys for cryptographic operations such as sign, verify, encrypt and decrypt or their usage in channels for these purposes. The capability to log these operations is determined on a per-key basis by the `LogKeyUsage` permission group flag on the ACL group authorizing the operation for which logging is desired.

See the *nCore Developer Tutorial* for further information on ACLs.

The `generatekey` utility (see Key generation options and parameters) provides the ability to set this permission group flag when a key is generated by either:

- Specifying `logkeyusage=yes` as an option on the command line
- Answering **yes** to the `logkeyusage` question if the command is being used interactively.

When `generatekey` is used this flag is applied to all permission groups but is only checked by the HSM on the group authorizing the desired action.

The following example shows this set on permission group 0 of a key's ACL.

```
groups[    0].flags= LogKeyUsage
                     .n_limits= 0
                     .n_actions= 2
                     .actions[    0].type= OpPermissions
                                           .details.oppermissions.perms= DuplicateHandle
ExportAsPlain GetAppData SetAppData
```

```
ReduceACL ExpandACL Encrypt Verify UseAsBlobKey GetACL
```

The table in Commands audited shows which commands implement conditional key usage logging.

5.11.3. Commands audited

The table below shows which nCore commands may produce audit logs. The column titled 'Logged?' specifies if the logging is conditional or not. An entry in that column of 'ALWAYS' means that the command is unconditionally logged. An entry in that column of 'CONDITIONAL' means that logging is conditional according to the criteria below:

Conditional items:

(a) means the command is logged if an ACL check on a supplied key involves the `LogKeyUsage` bit, see Key usage logging

(b) means the command may need NSO permissions or (for files/shares with their own ACLs) some other certifier. If the command is certified using a `CertType_SigningKey` entry, a log record may be generated as a result of an ACL check (for `UseAsCertificate` permission) on that key.

(c) means the command is logged if the object involved is "loggable". Note that keys are always loggable if the Security World was created with `--mode=common-criteria-cmts`, see Enabling audit logging

(d) means this command becomes 'Always Logged' if the Security World was created with `--mode=common-criteria-cmts`, see Enabling audit logging

(e) means this command is logged if it accesses a (local or remote) smartcard

Commands marked "(a),(c)" will therefore generate audit-log records if either one of the "input" keys requires logging as part of an ACL check, or one of the "output" keys is a loggable object.

The other two columns in the table refer to the information that is logged for that command. See Audit log contents for more information.

Command	Logged?	CmdAuditInfo contents	Extra AuditInfo entries
Cmd_ChangeShareGroupPIN	ALWAYS	SlotID slot table: ShareInfo shares	ObjectUse(cert) TokenID
Cmd_ChangeSharePIN	ALWAYS	SlotID slot ShareInfo share	ObjectUse(cert) TokenID

Command	Logged?	CmdAuditInfo contents	Extra AuditInfo entries
Cmd_ChannelOpen	CONDITIONAL (a)	ChannelMode mode	ObjectUse(key,cert)
Cmd_CheckUserAction	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_CreateSEEConnection	ALWAYS	UUID containerid	-
Cmd_Decrypt	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_DeriveKey	CONDITIONAL (a),(c),(d)	-	ObjectUse(key,cert) ObjectNew
Cmd_Destroy	CONDITIONAL (c)	AuditObjectID objid Word refcount	-
Cmd_Duplicate	CONDITIONAL (a),(c), (d)	-	ObjectUse(key,cert) ObjectNew
Cmd_DynamicSlotCreateAssociation	ALWAYS	SlotID slot	-
Cmd_DynamicSlotsConfigure	ALWAYS	Word slot	-
Cmd_Encrypt	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_EraseFile	CONDITIONAL (b),(e)	-	ObjectUse(cert) TokenID
Cmd_EraseShare	ALWAYS	SlotID slot Word i ShortHash hkt	ObjectUse(cert) TokenID
Cmd_Export	CONDITIONAL (a),(d)	-	ObjectUse(key,cert)
Cmd_FileCopy	CONDITIONAL (b),(e)	-	ObjectUse(cert) TokenID
Cmd_FileCreate	CONDITIONAL (b),(e)	-	ObjectUse(cert) TokenID
Cmd_FileErase	CONDITIONAL (b),(e)	-	ObjectUse(cert) TokenID
Cmd_FileOp	CONDITIONAL (b),(e)	-	ObjectUse(cert) TokenID
Cmd_FormatToken	ALWAYS	SlotID slot optional: KeyHashEx auth_key	ObjectUse(cert) TokenID
Cmd_GenerateKey	CONDITIONAL (c),(d)	-	ObjectUse(cert) ObjectNew
Cmd_GenerateKeyPair	CONDITIONAL (c),(d)	-	ObjectUse(cert) ObjectNew (x 2)
Cmd_GenerateLogicalToken	ALWAYS	-	ObjectUse(key,cert) ObjectNew
Cmd_GetACL	CONDITIONAL (a)	-	ObjectUse(key,cert)

Command	Logged?	CmdAuditInfo contents	Extra AuditInfo entries
Cmd_GetAppData	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_GetRTC	CONDITIONAL (b)	-	ObjectUse(cert)
Cmd_GetTicket	CONDITIONAL (c)	AuditObjectID objid	-
Cmd_ImpathKXBegin	ALWAYS	optional: ASCIIString esn	ObjectNew
Cmd_ImpathKXFinish	ALWAYS	-	ObjectUse(imp)
Cmd_Import	CONDITIONAL (c),(d)	-	ObjectNew
Cmd_InitialiseUnit	ALWAYS	=InitialiseUnitEx	-
Cmd_InitialiseUnitEx	ALWAYS	InitModeFlags initmodeflags KMLType kmltype	Startup
Cmd_InsertSoftToken	ALWAYS	SlotID slot	-
Cmd_LoadBlob	CONDITIONAL (a),(c),(d)	-	ObjectUse(lt,key,cert) ObjectNew
Cmd_LoadLogicalToken	ALWAYS	-	ObjectUse(lt,cert) ObjectNew
Cmd_MakeBlob	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_NVMemAlloc	CONDITIONAL (b)	-	ObjectUse(cert)
Cmd_NVMemFree	CONDITIONAL (b)	-	ObjectUse(cert)
Cmd_NVMemOp	CONDITIONAL (b)	-	ObjectUse(cert)
Cmd_ReadFile	CONDITIONAL (b),(e)	-	ObjectUse(cert) TokenID
Cmd_ReadShare	ALWAYS	SlotID slot Word i Word sharesleft	ObjectUse(cert,lt) TokenID
Cmd_ReceiveKey	ALWAYS	-	ObjectNew ObjectUse(imp)
Cmd_ReceiveShare	ALWAYS	Word i Word sharesleft	ObjectUse(imp,lt)
Cmd_RedeemTicket	CONDITIONAL (c)	AuditObjectID objid Word refcount	-
Cmd_RemoveKM	ALWAYS	=RemoveKMEx	ObjectUse(cert)
Cmd_RemoveKMEx	ALWAYS	KeyHashEx hkm	ObjectUse(cert)
Cmd_RemoveSoftToken	ALWAYS	SlotID slot	-
Cmd_SendKey	ALWAYS	-	ObjectUse(imp,key,cert)

Command	Logged?	CmdAuditInfo contents	Extra AuditInfo entries
Cmd_SendShare	ALWAYS	SlotId slot Word i KeyHashEx hkm TokenHash hkt	ObjectUse(imp,cert) TokenID
Cmd_SetACL	CONDITIONAL (a),(d)	-	ObjectUse(key,cert)
Cmd_SetAppData	CONDITIONAL (a),(d)	-	ObjectUse(key,cert)
Cmd_SetKM	CONDITIONAL (a),(d)	-	ObjectUse(key,cert)
Cmd_SetNSOPerms	ALWAYS	=SetNSOPermsEx	-
Cmd_SetNSOPermsEx	ALWAYS	NSOPermsModeFlags nsoflags KeyHashEx hknso NSOPerms publicperms	-
Cmd_SetRTC	ALWAYS	RTCTime oldtime RTCTime newtime	-
Cmd_Sign	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_SignModuleState	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_StartAuditLogging	ALWAYS	ByteBlock nonce	Startup
Cmd_StaticFeatureEnable	ALWAYS	FeatureInfo info	-
Cmd_StopAuditLogging	ALWAYS	-	Shutdown
Cmd_Verify	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_WriteFile	CONDITIONAL (b),(e)	-	ObjectUse(cert) TokenID
Cmd_WriteShare	ALWAYS	SlotID slot Word i	ObjectUse(lt,cert) TokenID
Cmd_CreateBuffer (Solo-XC only)	CONDITIONAL (a)	-	ObjectUse(key,cert)
Cmd_CreateSEEWorld (Solo-XC only)	ALWAYS	-	-
Cmd_ForeignTokenOpen (Solo-XC only)	ALWAYS	SlotID slot	-
Cmd_SetSEEMachine (Solo-XC only)	ALWAYS	-	-

5.11.4. Audit log contents

Audit log information is recorded in audit records. Audit records are sent from the HSM within a data structure called an audit segment. Each audit segment starts with an

AuditSegmentHeader and ends with a signature block signed by a key KML. See Audit log verification for further information on this signature and how to verify it.

The number of audit records contained within an audit segment is variable and depends upon factors such as the processing load of the HSM and the number of auditable events being triggered.

Audit logs exported by the system are not in human readable format unless converted using a suitable tool such as nshieldaudit. For more information, see Read the audit log databases. The examples shown below have been converted to JSON format. A text format is also available.

5.11.4.1. Audit segment header contents

The AuditSegmentHeader contains the following information:

- The ESN of the HSM that produced the audit segment
- The logID (KML hash) of the HSM that produced the audit segment
- The runID. This changes each time the system boots
- The start and end audit indices
- A timestamp of when the audit segment was started in milliseconds since January 1 1970
- A timestamp of when the audit segment was signed in milliseconds since January 1 1970
- A hash of the audit logs contained within the segment

An example AuditSegmentHeader is shown below:

```
"header": {
            "v": 0,
            "flags": "0x0",
            "esn": "14BD-B089-E078",
            "logID": "33133940 fb686dc5 4184a726 b670b7c1 508a8ea1",
            "runID": 4,
            "startIndex": 3458,
            "endIndex": 3458,
            "starttime": 1690801099591,
            "signtime": 1690801099591,
            "datahash": {
                "mech": "SHA256Hash",
                "data": {
                    "h": "2dcd7912 b9f0edd3 82df65f7 9ccdf289 4470a569 078638bf 20158eb1 e4c6aa6f"
                }
            }
```

5.11.4.1.1. Audit index

Each audit log record has an audit log index which increments by one for each log created since the start of audit logging. The `AuditSegmentHeader` shows the index for the first and last audit log contained in the segment.

In normal operation no audit records are lost and there will be no gaps in the audit index sequence.

If an abnormal shutdown occurs there is a possibility that some records may have been lost. This will be indicated by a gap in the audit index sequence and also by an `AbnormalShutdown` entry associated with the `StartAuditLogging` command.

It is not possible to tell exactly how many audit records were lost due to an abnormal shutdown. The gap in the audit index sequence indicates the maximum number that could have been lost based on the load of the HSM at the time of the abnormal shutdown but the actual number lost could be fewer than this and could even be zero.

5.11.4.2. Audit log contents

Each audit log produced will contain information as shown in the `CmdAuditInfo contents` column of the table in Commands audited. All audit logs contain basic information to indicate the name of the command that triggered the audit log and a status that indicates the outcome of that command, as in the extract below from an audit log produced by a `GenerateKeyPair` command.

```
{
      "type": "Command",
      "body": {
            "flags": "0x0",
            "cmd": "GenerateKeyPair",
            "info": {},
            "status": "OK"
      }
}
```

If the `CmdAuditInfo contents` column shows other information that will appear in the `info` section.

If the `Extra AuditInfo entries` column of the table in Commands audited contains entries these will appear as separate log entries, as in the extract below.

```
{
     "type": "ObjectUse",
    "body": {
         "v": 0,
        "objid": 137,
```

```
        "action": {
            "type": "MakeBlob",
            "details": {
                "flags": "0x0"
            }
        }
    }
}
```

5.11.4.2.1. Object identification

Objects are identified by an `Objid`. A new ID is assigned when the object is created and will appear in the `ObjectNew` record as shown in the example below.

An object identifier is unique whilst the system runs uninterrupted but values may be reused if the system reboots. The combination of `RunID` from the audit segment header and `ObjID` is always unique.

```
 {
     "type": "ObjectNew",
    "body": {
        "v": 0,
        "objid": 135,
        "type": "Key",
        "details": {
            "type": "RSAPrivate",
            "hash": {
                "mech": "SHA1Hash",
                "data": {
                    "hash": "bb6b1e71 1cc06123 853af0d6 ff781cb0 b7a4b515"
                }
            }
        }
    }
}
```

The same ID will appear in `ObjectUse` records and other command records relating to the same object allowing you to trace the object through the audit log. If an object is copied the new object will have its own ID so that you can keep track of each copy. When the copy is created you can identify the object being copied by its hash.

5.11.4.3. Audit log verification

As described in Audit log contents audit log records are protected by a signature across the audit segment in which they are delivered. The signature is made using `KML` which is a secret key unique to the HSM and protected by the HSM.

Each time the HSM is re-initialized (e.g. after upgrading firmware and re-loading the Security World) the previous `KML` is destroyed and a new `KML` is created. The hash of `KML` is used as the `logID` for the audit log.

You can print the `KML` hash of the current Security World using `nfkminfo` and looking for `hkml`. If you have more than one HSM connected `nfkminfo` will show the `hkml` for each HSM.

5.11.5. Audit log administration

5.11.5.1. Reading the audit logs

Audit logs are read as described in the nShield Audit Log Service.

5.11.5.2. Audit logging and firmware upgrade

The audit log should be finalized before upgrading the firmware on an HSM enrolled in an audit logging Security World. To finalize the audit log, see Disabling audit logging

If you do not finalize the audit log before attempting to upgrade the firmware of a PCIe or USB HSM, you will receive an error message. You can either finalize the audit log and repeat the command, or use one of the following commands to override the warning:

- **nShield Solo, Solo XC, and Edge HSMs:** The `loadrom` command with the `--noauditcheck` option.
- **nShield 5s HSMs:** The `hsmadmin upgrade` command with the `--force` option.

Upgrading the firmware without finalizing the audit log will result in there being no final audit log record. This means it will not be possible to prove completeness of the audit log.

5.11.5.3. Audit logging and the system clock

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

If the system clock is lost, for instance due to the HSM running on battery power for an extended period of time, ncoreapi commands that would result in an audit log being generated will be inhibited until the system clock is restored:

See Setting the system clock for more information on setting the system clock on nShield 5s HSMs.

5.12. nShield Audit Log Service

5.12.1. Introduction

The nShield Audit Log Service (`nshieldauditd`) retrieves and removes audit logs from HSMs that are available to the local hardserver and are listed in the nShield audit log service's configuration file. There are two types of audit logs: nCore audit logs and signed system logs.

It retrieves nCore audit logs, see Audit Logging, from PCIe HSMs running firmware version 13.5 or later and from network-attached HSMs running firmware version 13.6 or later.

nShield Audit Log Service and the new audit databases support the new audit logs for 13.5 firmware onwards only. It does not retrieve the CEF-format audit logs from earlier firmware versions (the legacy audit configuration settings in `[auditlog_settings]` continue to be supported for those logs, and verification thereof can be done with the separate `cef-audit-verify` tool).

It also retrieves signed system logs from nShield 5s HSMs running firmware version 13.5 or later and nShield 5c HSMs running firmware version 13.6 or later. See System logging (nShield 5 HSMs).

Both types of logs are saved in separate local database files stored by default in a subdirectory of `NFAST_KMDATA`.

The service must be configured to enable log fetching from available HSMs. Once configured, the service runs automatically in the background without user intervention.

Connect XC and 5c HSMs enrolled in a Security World with audit-logging enabled (including a Common-Criteria (CMTS) compatible Security World) will continuously generate nCore audit-logs and, if these audit-logs are not transferred and removed by the nshieldauditd service, the HSMs may run out of disk space and will not process certain commands until their audit-logs are transferred and removed.

5s and 5c HSMs generate signed system logs internally in v13.5 onwards. These logs also need to be transferred and removed by the nShield Audit Log Service. This is true even when the Security World does not have audit-logging enabled.

5.12.2. nShield Audit Log Service configuration

The nShield Audit Log Service is configured by editing a YAML-format configuration file. Configuration changes can be applied by restarting the service.

5.12.2.1. Restarting nshieldauditd on Linux

The nShield Audit Log Service can be restarted on Linux by running the following command as `root`:

```
/opt/nfast/scripts/init.d/nshieldauditd restart
```

Instead of `restart` use `stop` followed by `start` parameters to explicitly stop and start the service respectively.

5.12.2.2. Restarting nshieldauditd on Windows

The service can be stopped and started from an elevated (Administrator) Windows command prompt using these commands:

```
net stop "nShield Audit Log Service"
net start "nShield Audit Log Service"
```

To restart the service as a single operation, run the following PowerShell command from an elevated (Administrator) PowerShell shell:

```
Restart-Service "nShield Audit Log Service"
```

5.12.2.3. Editing the configuration file on Linux

The configuration file is located at `/opt/nfast/kmdata/auditlogs/nshieldauditd.conf`. Editing the file requires membership of the `nfast` group.

5.12.2.4. Editing the configuration file on Windows

The configuration file is located at `C:\ProgramData\nCipher\Key Management Data\auditlogs\nshieldauditd.conf` by default. Editing the file requires the privileges of the built-in local Administrators group.

5.12.2.5. Default configuration file

The default configuration is created when the service first starts, and looks as follows:

```
# nShield Audit Log Service config file
#
# This configuration file is in yaml format.

#---------------------------------------------------------------------
```

```
# modules is the list of ESNs of HSMs from which to fetch nCore and
# (if applicable) syslog audit.
# By default, no modules are included in the list.
# Set modules: ["AAAA-AAAA-AAAA", "BBBB-BBBB-BBBB"] to enable fetching
# of audit from modules with ESNs AAAA-AAAA-AAAA and BBBB-BBBB-BBBB
# Comment out or delete the modules: [] line to fetch audit from
# all modules.

modules: []

#------------------------------------------------------------------
# syslog_fetch_interval_hours specifies the time in hours to wait
# between fetching the nShield 5 HSM syslog audit.
# It is always fetched when the service first starts.
# Default value is 12.

#syslog_fetch_interval_hours: 12

#------------------------------------------------------------------
# suppress_repeat_logs_interval specifies the time
# to wait for repeated log messages to be logged again.
# Set interval values in 's'(seconds), 'm'(minutes), or 'h'(hours),
# for example: '0s', '5m', '10h' etc.
# If set to '0s' then no logs will be suppressed.
# Recommended to set it to '0s' while in debug mode.
# Default value is '5m'.

#suppress_repeat_logs_interval: '5m'

#------------------------------------------------------------------
# logging_level controls the verbosity of log messages from the nShield Audit Log Service's own operations (not
the audit logs themselves)
# Available levels in increasing order of severity are DEBUG, INFO, WARNING, ERROR, or CRITICAL (default is INFO)

#logging_level: INFO
```

5.12.2.6. Configurable options

5.12.2.6.1. modules

The nShield Audit Log Service only fetches logs from the modules that are specified in the configuration file. If the `modules` field is commented out or absent then all HSMs available to the local hardserver will be monitored by the nShield Audit log service; this provides a shorthand if the goal is to fetch audit from all enrolled modules.

By default, the `modules` field is an empty list, so this must be configured to enable the fetching of logs. Only one client system of the modules should enable log fetching, otherwise the audit log services will conflict with each other; that is why the user must explicitly opt-in to enable the fetching, to prevent any conflicts from default configuration. You can add or remove a module by editing the configuration file and restarting the nShield Audit Log Service. The `ESN` of each module to be enrolled should be specified within quotes and separated with a comma.

To fetch nCore or signed system logs from a network-attached HSM

(Connect XC or 5c) the client machine where the nShield Audit Log Service has been configured must have been enrolled as a privileged client of that HSM.

Security World HSM estates with multiple client machines must configure just one client's nShield Audit Log Service to monitor any given HSM, although different nShield Audit Log Services may be configured to monitor different subsets of HSMs. If multiple services fetch logs from the same HSM, then each client's databases will only contain some of the log records.

5.12.2.6.2. syslog_fetch_interval_hours

This option configures how often the service should fetch the signed system logs from the HSM.

By default, `syslog_fetch_interval_hours` is commented out and the default value of 12 hours will be used.

The interval in the configuration file is in hours. The interval must be set in between 1 and 168 hours.

Signed system logs are produced at a relatively low rate, and fetching and removing the logs itself produces an audit record. This is why fetching of the logs is infrequent, unlike the continuous streaming of records that is done for nCore audit.

Signed system logs are not produced by nShield Solo XC and nShield Connect XC HSMs.

5.12.2.6.3. suppress_repeat_logs_interval

This option configures how often the service should show repeated log messages.

By default, `suppress_repeat_logs_interval` is commented out and the default value of 5 minutes will be used,

The interval for log suppression can be set in seconds, minutes or hours.

It is recommended to set the `suppress_repeat_logs_interval` to 0 when debugging.

5.12.2.6.4. logging_level

This option configures how much debug information is recorded in the service's application log. This is for monitoring the nShield Audit Log Service's operation in fetching nCore and signed system logs, and does not control the verbosity of that actual audit log data.

By default the service logging level is set to `INFO`, which will also report any higher severity messages. This is the recommended setting, although logging can be reduced by setting `WARNING`, `ERROR`, or `CRITICAL` to restrict only to messages at that log level or above.

If additional logging is desired, the `DEBUG` setting can be set, but note that this will write considerably more information to the service log, and will also slow down service operation as it will also include a human-readable representation of the nCore audit logs as they arrive. This is only recommended when investigating issues or if advised to do so by nShield Support.

On Linux, the service debug log is `/opt/nfast/log/nshieldauditd.log`. On Windows, the service debug log can be viewed in the Windows Event Viewer (`nCipherAuditSvc` event source) or can be queried using `nshieldeventlog -s nCipherAuditSvc -c 10` (for example) to view the most recent 10 lines of the service Event Log (adjust `-c` parameter for a larger number of lines, or use the `-f` parameter to write to a file).

When first configuring the system, it is recommended to check the service debug log to ensure that it reports that it has detected the intended modules and enabled log fetching for them.

5.12.3. Warrants

The nShield Audit Log Service attempts to retrieve the KLF2 warrant for each configured module automatically. It stores the warrants in the nCore audit log database for their module.

In nShield 5 HSMs, this warrant is embedded in the module and is always available.

In nShield XC HSMs, the KLF2 warrant must be present in one of the following locations in the filesystem of the machine running the nShield Audit Log Service, depending on how you are interacting with the HSM:

- `/opt/nfast/kmdata/warrants` (**Linux**) or `C:\ProgramData\nCipher\Key Management Data\warrants` (**Windows**)
  This is the same location as where a warrant should be placed if using nShield Remote Administration smartcards.
- `/opt/nfast/kmdata/hsm-<ESN>/warrants` (**Linux**) or `C:\ProgramData\nCipher\Key Management Data\hsm-<ESN>\warrants` (**Windows**)
  Where `<ESN>` is the ESN of the relevant module.

This is where nShield Connect XC warrants are automatically placed on the RFS. If the warrant is not here, you will need to reconfigure (see rfs-setup) and reboot the HSM to copy the warrant to the correct location.

nShield Connect XC

To run the nShield Audit Log Service on a different client machine to the RFS, you must first copy the warrant file to `/opt/nfast/kmdata/warrants` or `C:\ProgramData\nCipher\Key Management Data\warrants`.

If there was no warrant present when the nShield Audit Log Service was started, you can still verify the chain up to the root of trust post-hoc with `nshieldaudit`. Copy the warrant to `/opt/nfast/kmdata/warrants` or `C:\ProgramData\nCipher\Key Management Data\warrants` on the machine where you are running `nshieldaudit`. When it displays the information, for example in `nshieldaudit ncore query`, it always re-runs the verification of the chain and will pick up the warrant from the aforementioned location.

5.12.4. Log databases

The nShield Audit Log Service saves the nCore audit logs and signed system logs in local database files. Logs for each module are saved in their own database in a subdirectory of the `auditlogs` directory named after the ESN of that module.

To back-up the log databases, first stop the nShield Audit Log Service, recursively copy the `auditlogs` directory, then start the service again. Stopping the service during backup ensures that a consistent state of the database files is copied, with no temporary transaction files being present. If restoring from backup, the original permissions and ownership of files and directories should be restored.

5.12.4.1. Linux

The databases are located in `/opt/nfast/kmdata/auditlogs`.

The directory and its contents have `read` and `write` permissions for `nfast` group by default. To restrict access to a different group, run the `/opt/nfast/sbin/install` script with `NSHIELDAUDITD_GROUP` environment variable set to the name of the alternate group. This must be set every time `install` is run in order to retain this setting. The `nshieldauditd` service user will be added to the alternate group automatically, and the group will be

created during install if it does not already exist.

5.12.4.2. Windows

The databases are located in `C:\ProgramData\nCipher\Key Management Data\auditlogs`.

Access is restricted to members of the built-in Administrators group (with elevated privilege) and the nShield Audit Log Service itself. To allow access to other users or groups without elevated privilege, the DACL of the directory may be modified, but note that it will be reset to defaults when the nShield software is re-installed.

5.12.4.3. Alternative log database directory

The above log database locations can be overridden by setting the environment variable `NFAST_AUDIT_LOGDIR` to the alternative location. This can be another local directory, or a network share.

If `NFAST_AUDIT_LOGDIR` is set, that also changes the service configuration file location to be inside that directory, that is, `NFAST_AUDIT_LOGDIR/nshieldauditd.conf`. Set the environment variable `NFAST_AUDIT_SERVICE_CONF` if a different location is desired for the configuration file.

These environment variables must be set in the environment of the service, and of any tools which read the logs. On Windows, set the environment variables in the Windows system variables, and on Linux in the `/etc/nfast.conf` file, in order to make them visible to the service. The service must be restarted for the change to take effect.

If you already have audit logs stored in the default location, it is recommended that the nShield Audit Log Service be stopped, the existing contents moved to the new location, and then the service started after setting the environment variables, so that the existing portion of the logs are not lost.

5.12.4.4. Read the audit log databases

The nShield Audit Log Service databases are accessed with the command line tool `nshieldaudit` which manages logs using different subcommands as explained in the following sections. By default, running the `nshieldaudit` subcommands requires membership of `nfast` group on Linux and an elevated (Administrator) command prompt on Windows.

```
nshieldaudit <subcommand>
```

The audit logs in the databases are not in a human-readable format. Therefore, to be made human-readable, they must be exported to a file using the `nshieldaudit` tool.

Timestamps reported by `nshieldaudit` and in the audit logs themselves are UTC.

`nshieldaudit` supports the following subcommands:

- syslogs
- ncore
- serveradmin

5.12.5. Reading signed system logs

The `nshieldaudit syslogs` subcommand allows the user to access the signed system logs stored in the database.

```
nshieldaudit syslogs [-h] {query,export,info}
```

This subcommand takes the following parameters:

Parameter	Description
--help or -h	Show the help message and exit
query	Show the contents of the signed system logs database
export	Export the audit data to a file in JSON format
info	An overview of one or more log entries

5.12.5.1. Querying available signed system logs

Use the `nshieldaudit syslogs query` subcommand to quickly see what signed system logs databases are available, an overview of the date and sequence ranges of their content, and a "Log Status" to summarize any detected verification errors or missing data.

```
nshieldaudit syslogs query [-h] [-e ESN(s) [ESN(s) ...]]
```

The subcommand `nshieldaudit syslog query` takes the following parameters:

Parameter	Description
--help or -h	Show the help message and exit
-e ESN(s) [ESN(s) ...] or --esn ESN(s) [ESN(s) ...]	One or more ESN(s) of the HSM(s) that generated the audits

A typical output looks like the following:

```
$ nshieldaudit syslogs query
--esn AAA1-BBB2-CCC3
    Sequence numbers:          --start 1 --end 7775
    Time:                      --from 2024-01-13_12:11:03 --to 2024-05-28_08:31:40
    Malformed Logs:            0
    Unverified Logs:           0
    Unattested Logs:           0
    Missing Sequences:         None
    Log Status:                OK
--esn 1ABC-1DEF-1GHI
    Sequence numbers:          --start 13 --end 125
    Time:                      --from 2024-05-23_13:05:48 --to 2024-05-28_07:48:33
    Malformed Logs:            0
    Unverified Logs:           0
    Unattested Logs:           0
    Missing Sequences:         1-12
    Log Status:                Issues found
```

5.12.5.1.1. nshieldaudit syslogs info

This subcommand allows the user to see a more detailed summary of the available signed syslogs, for a given ESN, than the `nshieldaudit syslogs query` subcommand. It allows the time ranges and number of log lines in each of the sequence numbers in the signed syslogs to be listed, in support of constraining the range of interest when retrieving the log contents using the `export` subcommand.

```
nshieldaudit syslogs info [-h] -e ESN(s) [ESN(s) ...] [--start START_INDEX] [--end END_INDEX] [--from FROM_DATE]
[--to TO_DATE]
```

The subcommand `nshieldaudit syslog info` takes the following parameters:

Parameter	Description
--help or -h	Show the help message and exit
-e ESN(s) [ESN(s) ...] or --esn ESN(s) [ESN(s) ...]	One or more ESN(s) of the HSM(s) that generated the audits

The subcommand `nshieldaudit syslog info` allows the user to add the following filters:

Parameter	Description
--start START_INDEX	Start index of the audits desired
--end END_INDEX	End index of the audits desired
--from FROM_DATE	Start date of the audits desired (UTC)
--to TO_DATE	End date of the audits desired (UTC)

5.12.5.2. Exporting signed system logs to JSON format or text format

Use the subcommand `nshieldaudit syslogs export` to export the signed system logs data to a file. The output file is written in JSON format (default), or in the human-readable text format if the `-t` or `--text` parameter is added.

```
nshieldaudit syslogs export [-h] -e ESN(s) [ESN(s) ...] [-v] -f FILE [--overwrite] [--malformed] [-t] [--start
START_INDEX] [--end END_INDEX] [--from FROM_DATE] [--to TO_DATE]
```

The subcommand `nshieldaudit syslog export` takes the following parameters:

Parameter	Description
--help or -h	Shows the help message and exits
-e ESN --esn ESN	Required: the ESN of the module whose signed system logs is to be exported
-v, --verify	Re-verify signatures whilst exporting; this significantly increases export time. (default is to report cached verification status)
-f FILE, --file FILE	Required: Store output in this file (in JSON format)
--overwrite	Overwrite the output file if it exists
--malformed	Export malformed logs
-t, --text	Export of human-readable summary of logs in the text format rather than full JSON

The subcommand `nshieldaudit syslog export` allows the user to add the following filters:

Parameter	Description
--start START_INDEX	Start index of the audits desired
--end END_INDEX	End index of the audits desired
--from FROM_DATE	Start date of the audits desired (UTC)
--to TO_DATE	End date of the audits desired (UTC)

Run `nshieldaudit syslogs query` first in order to get parameters that can be copied to produce the `nshieldaudit syslogs export` command-line.

By default, the output format is JSON format (it is formatted with whitespace indentation to also be human-readable). To emit in a more compact human-readable text format, add the `--text` parameter. In many cases `--text` will be the more convenient format to use for browsing the data, and it is faster to export and produces a smaller output file.

Example command (text output):

```
nshieldaudit syslogs export --esn 5323-D01E-6DC0 -t
```

The output file will contain content like the following.

```
2024-11-12 06:51:19 HDR to=2024-11-12 07:51:33 esn=5323-D01E-6DC0 seq_no=1231
2024-11-12 06:51:19 nshield5 monitor[3987]: New log file: current seq-no=1231
2024-11-12 06:51:19 nshield5 monitor[3987]: Last log: {'signature': {'key': 'AAAA...', 'sig': 'AAAA...', 'mech':
'ecds...', 'hash': 'a684...'}, 'seq-no': 1230}
2024-11-12 06:51:19 nshield5 monitor[3987]: Command response: {'log-data': '2024-11-12 05:51:06 nshield5
moni...', 'signature': {'key': 'AAAA...', 'sig': 'AAAA...', 'mech': 'ecds...', 'hash': 'a684...'}, 'seq-no':
1230}
2024-11-12 06:51:28 nshield5 monitor[4002]: Command args: {'subcommand': 'expirelog', 'seq_no': 1230}
2024-11-12 06:51:28 nshield5 monitor[4002]: Removed saved log file messages.saved.1230
2024-11-12 06:51:28 nshield5 monitor[4002]: Command response: None
2024-11-12 07:51:33 nshield5 monitor[4017]: Command args: {'subcommand': 'exportlog', 'saved_log': False}
```

Run `nshieldaudit syslogs export --malformed` to export malformed logs into a file. By default the export file will be in JSON format unless `-t` or `--text` parameter is added.

```
nshieldaudit syslogs export --malformed -f ./log_export -e 5323-D01E-6DC0 --text --overwrite
```

5.12.6. Reading nCore audit logs

The `nshieldaudit ncore` subcommand allows the user to read nCore audit logs from the database.

```
nshieldaudit ncore [-h] {query,export}
```

This subcommand takes the following parameters:

Parameter	Description
--help or -h	Show the help message and exit
query	Shows the contents of the audit logs database

Parameter	Description
export	Export the audit data to a file in JSON (default) or text format

5.12.6.1. Querying available nCore audit logs

Use the `nshieldaudit ncore query` subcommand to quickly see what nCore audit log databases are available, an overview of the date and index ranges of their content, and a "Log Status" to summarize any detected verification errors or missing data.

```
nshieldaudit ncore query [-h] [-e ESN(s) [ESN(s) ...]] [-l LOGID [LOGID ...]]
```

This subcommand takes the following parameters:

Parameter	Description
--help or -h	Show the help message and exit
-e ESN(s) [ESN(s) ...] or --esn ESN(s) [ESN(s) ...]	One or more ESN(s) of the HSM(s) that generated the audits
-l LOGID [LOGID ...] or --logid LOGID [LOGID ...]	One or more LogID(s) of the nCore audit-log(s)

A typical output looks like this:

```
$ nshieldaudit ncore query
--esn A1A1-B2B2-C3C3 --logid f515236b5a4970f0e0ac7a3c2852c6b53fee28e3
    Indexes:                --start 1 --end 6546650
    Time:                   --from 2024-05-30_12:21:56 --to 2024-06-04_13:16:25
    KML Status:             Verification chain OK
    Finalized Status:       Finalized
    Malformed Segments:     0
    Unverified Segments:    0
    Missing indexes:        None
    Log Status:             OK
```

A brief description of each field:

Parameter	Description
Indexes	First and last Audit Index present in the local database for this log.
Time	UTC timestamps of the first and last audit records for this log.
KML Status	Status of verification chain of the KML public key (used to sign audit records) up to the nShield HSM root of trust.
KML Error	This field only appears if there is an error in KML Status, to give more details on KML verification error.

Parameter	Description
KML Warning	This field only appears if it was not possible to find a warrant to verify the chain of trust.
Finalized Status	`Finalized` if the module has been re-initialized (with `initunit` or by loading a Security World again) and `Unfinalized` otherwise.
Malformed Segments	Count of the number of audit log segments that could not be processed.
Unverified Segments	Count of the number of audit log segments that failed to verify with the KML key.
Missing indexes	List of the ranges of Audit Indexes that are missing in the database. This is inclusive counting, for example `1-2` means both Audit Index `1` and `2` are missing.
Log Status	`OK` if there are no detected issues with this log, and `Issues Found` if there are any errors or warnings reported in the other fields.

5.12.6.2. Exporting nCore audit to JSON or text format

Use the `nshieldaudit ncore export` subcommand to export the audit data to a file. The output file is written in JSON format (default), or in the human-readable text format if the `-t` or `--text` parameter is added.

```
nshieldaudit ncore export [-h] -e ESN(s) [ESN(s) ...] -l LOGID [LOGID ...] [-v] -f FILE [--overwrite] [-t] [--
start START_INDEX] [--end END_INDEX] [--from FROM_DATE] [--to TO_DATE]
```

The `nshieldaudit ncore export` command can take some time if there is a very large amount of data to export.

The subcommand `nshieldaudit ncore export` takes the following parameters:

Parameter	Description
--help or -h	Show the help message and exit
-e ESN or --esn ESN	Required: ESN of the module whose logs to export
-l LOGID or --logid LOGID	Required: The specific LogID from this module to export
-t, --text	Export of human-readable summary of logs in text format rather than full JSON
-v, --verify	Re-verify signatures whilst exporting; this significantly increases export time. (default is to report cached verification status)
-f FILE, --file FILE	Required: Store output in this file

Parameter	Description
--overwrite	Overwrite the output file if it exists
--malformed	Export malformed logs

The subcommand `nshieldaudit ncore export` allows the user to add the following filters:

Parameter	Description
--start START_INDEX	Start index of the audits desired
--end END_INDEX	End index of the audits desired
--from FROM_DATE	Start date of the audits desired (UTC)
--to TO_DATE	End date of the audits desired (UTC)

Run `nshieldaudit ncore query` first in order to get parameters that can be copied to produce the `nshieldaudit ncore export` command-line.

The `--esn` and `--logid` options must be supplied to specify the log whose data to export (the `nshieldaudit ncore query` command can be used to get these parameters), along with the `--file` parameter to specify the output file path.

By default, the output format is JSON format (it is formatted with whitespace indentation to also be human-readable). To emit in a more compact human-readable text format, add the `--text` parameter. In many cases `--text` will be the more convenient format to use for browsing the data, and it is faster to export and produces a smaller output file.

Example command (text output):

```
$ nshieldaudit ncore export --esn 4210-02E0-D947 --logid 9455993eca529189f44a4861d2c23ef359f549ff --text --file
ncore.audit.4210-02E0-D947.txt
100%
```

The output file will contain content like the following. Note that identical repeated operations within each "segment" of log like this are collated with the "REPEAT=+17" field which indicates that this command was repeated a further 17 times, that is, 18 times total. This reduces the verbosity of data when the system is under load.

```
2024-06-07 22:43:59.876 HDR logid=9455993eca529189f44a4861d2c23ef359f549ff runid=2 start=36398476 end=36398546
signtime=2024-06-07_22:43:59.883
2024-06-07 22:43:59.876 idx=36398476 src=Host cmd=Sign rc=OK obj=162 REPEAT=+17
2024-06-07 22:43:59.877 idx=36398477 src=Host cmd=Sign rc=OK obj=161 REPEAT=+19
2024-06-07 22:43:59.877 idx=36398480 src=Host cmd=Sign rc=OK obj=163 REPEAT=+15
2024-06-07 22:43:59.878 idx=36398484 src=Host cmd=Sign rc=OK obj=160 REPEAT=+16
```

Run `nshieldaudit ncore export --malformed` to export malformed logs into a file. By default the export file will be in JSON format unless `-t` or `--text` parameter is added.

```
nshieldaudit ncore export --malformed -f ./log_export_ncore -e 5323-D01E-6DC0 --text --overwrite -l
0032bf17c0aeb84aefa31c29f846886d39710eff
```

Audit records are verified automatically as they arrive by the nShield Audit Log Service. To re-verify records when exporting, pass the `--verify` option, otherwise the cached verification status is reported in the output file. Re-verification will significantly increase the time to export the logs.

5.12.7. Monitoring and managing audit data sources

5.12.7.1. Hardserver audit database storage limit

The hardserver has its own audit database as the temporary staging area for nCore audit records from modules that are local to it. This ensures that the audit records are not lost whilst waiting to be handed over to the nShield Audit Log Service.

- For Linux it is located at `/opt/nfast/hardserver.d/audit.db`.
- For Windows it is located at `C:\ProgramData\nCipher\hardserver.d\audit.db`.
- For network-attached HSMs (Connect XC and 5c), it is stored internally on the device.

This is not the same database as the nShield Audit Log Service database which is located as described in Log databases.

There is a configuration section `auditdb_settings`, which is present on both client-side and in network-attached HSMs configuration. If the section is missing, then the defaults will be applied. To learn more about the configuration section, see Audit Database setting.

A default `auditdb_settings` configuration section is as follows:

```
[auditdb_settings]
# Start of the auditdb_settings section
# Hardserver settings for (new format v13.5 and later) nCore audit logging
# database.
# Each entry has the following fields:
#
# Minimum available megabytes of free-space for audit DB to operate.
# (default=0 for auto-configured)
#  free_space_min_mb=MB
#
# Maximum total size of audit DB in megabytes. (default=0 for no limit)
#  db_size_max_mb=MB
#
# Maximum number of audit indexes in audit DB. (default=0 for no limit)
```

```
#  audit_indexes_max=COUNT
```

This is not the same configuration file as the nShield Audit Log Service configuration file (which is described in nShield Audit Log Service configuration).

By default network-attached HSMs (5c or Connect XC) pause HSM auditing when 500 MB of free space is left in the appliance, and client-side hardservers pause auditing of local HSMs when 50 MB of free space is left in the local storage. When HSM auditing is paused, any nCore commands which require the creation of an audit record will be denied. The paused status of an HSM will also be visible in the output of the `enquiry` tool in the `hardware status` field.

The configuration section changes can be applied dynamically by pushing configuration to a network-attached HSM. For local modules run the `cfg-reread` command to apply a change.

5.12.7.2. Checking nCore audit temporary storage usage

In normal usage, when the nShield Audit Log Service is operating correctly, the amount of temporary storage consumed for audit on HSMs should be small, as the data is being offloaded to the long-term databases. The following commands can be used to monitor to help detect issues.

Checking temporary storage for nCore audit usage for on a network-attached HSM (Connect XC or 5s) can be checked using the command `stattree RemoteServers 1 ServerGlobals` (replacing `1` with whatever the module number is).

Local temporary storage for nCore audit usage for Solo XC or 5s can be checked using the command `stattree ServerGlobals`.

The `AuditDBUsedSpaceMB` field states the number of megabytes of space consumed by the temporary database (where data is staged before being fetched by the nShield Audit Log Service). The `AuditDBFreeSpaceMB` field states the number of megabytes of available space on the same partition as that temporary database.

Example output (for local storage usage):

```
stattree ServerGlobals | grep AuditDB
   -AuditDBFreeSpaceMB   102694
   -AuditDBUsedSpaceMB   45
```

5.12.7.3. Checking or repairing hardserver audit database

The `nshieldaudit serveradmin` subcommand enables the system administrator to check or repair, if needed, the temporary database of nCore audit logs not yet processed by the nShield Audit Log Service. It requires a privileged connection to the hardserver (membership of `nfast` group by default on Linux, and elevated Administrator command prompt by default on Windows). If managing a network-attached HSM, that must also have been enrolled as privileged.

This command only interacts with the hardserver to query its internal staging database that temporarily stores nCore audit data, and does not relate to the long-term databases that the `nshieldaudit ncore` and `nshieldaudit syslogs` subcommands provide access to.

To learn more about the hardserver internal `audit.db`, see hardserver audit database.

```
nshieldaudit serveradmin [-h] [-m MODULE] [-q] [-e ESN] [-l LOGID] [--delete] [--end END] [--force] [--delete-status] [--recreate-db] [--no-confirm]
```

This subcommand takes the following parameters:

Parameter	Description
-m or --module	Module ID of source data server. Use 0 to refer to local hardserver.
-e or --esn	ESN of the log. Required when deleting.
-q or --query	query status of logs
-l or --logid	Log id of audit logs. Required when deleting.
--delete	Manually delete logs. Log deletion is normally automatic. This option should only be used in a recovery situation.
--end	When deleting audit logs the index of the last logs to be deleted.
--force	When deleting override check on whether logs have already been exported.
--delete-status	Delete the audit export status information. Not recommended unless the LogID is reported as finalized.
--recreate-db	Deletes entire database and creates a new empty database. Not recommended except in a recovery situation.
--no-confirm	Omits the interactive confirmation for --delete and --recreate-db options.

5.12.7.3.1. Querying hardserver database contents

Run `nshieldaudit serveradmin --query` to query the contents of the hardserver's

temporary staging database.

The required hardware capabilities of the host machine running the nShield Audit Log Service depend on the amount of audit-log data that it needs to export, verify and expire. By running `nshieldaudit serveradmin --query` the user can monitor that the nShield Audit Log Service is successfully keeping up with offloading nCore audit records.

`nshieldaudit serveradmin --query` reports on the local system's hardserver by default (this will show information for audit from Solo XC and 5s modules). To query the temporary staging database of a network-attached HSM (Connect XC or 5c), add the `-m` or `--module` parameter to specify the module number of that HSM.

Example output (showing both the completed offload of a "Finalized" log as well as the in-progress offload of the log for the currently loaded Security World):

```
nshieldaudit serveradmin --query -m1
--esn=4210-02E0-D947 --logid=1330c21f241db5b2d86f5ddb99bccaacf12453df
    Indexes still on source data server (inclusive): --start=0 --end=0
    Total remaining index count: 0
    Exported to index: 218871
    Log creation time: 2024-04-08 10:12:56.145
    Log status: Finalized
--esn=4210-02E0-D947 --logid=9455993eca529189f44a4861d2c23ef359f549ff
    Indexes still on source data server (inclusive): --start=749879020 --end=749880109
    Total remaining index count: 1090
    Exported to index: 749879019
    Log creation time: 2024-06-07 20:57:03.480
    Log status: Unfinalized
```

Placeholder records of previously encountered logids seen by the system are kept by default (as in the "Finalized" log example above). If you wish to delete these placeholders for a "Finalized" log, you may issue the command `nshieldaudit serveradmin --delete --delete-status --esn=4210-02E0-D947 --logid=1330c21f241db5b2d86f5ddb99bccaacf12453df -m1` (replacing `1` with the module number in `-m1` - this may be omitted if it is a local module, and passing the actual `--esn` and `--logid` values for the log status information to be deleted).

5.12.7.3.2. Forcibly deleting records or re-creating hardserver database

The `nshieldaudit serveradmin --delete` command can be used to delete records up to a specified `--end` value.

The `nshieldaudit serveradmin --recreate-db` command can be used to delete the entire database, that is, all status tracking information and any nCore audit records not yet offloaded will be lost.

In both cases, if interacting with a remote module, add the `-m1` (or appropriate module

number) parameter to specify which hardserver the command should be sent to.

These commands will prompt for confirmation by default. They are not intended to be issued except in non-production test systems or as emergency recovery options if advised to do so by nShield Support.

5.12.7.4. Checking signed system logs HSM storage

The percentage free space for signed system logs temporary storage for a 5s or 5c can be queried using the command `appliance-cli -m1 gethsmenvstats`, replacing `1` with the module number. This is not applicable to XC modules.

Example output showing 90% free space, that is, only 10% used:

```
appliance-cli -m1 gethsmenvstats | grep log_free_space
           "log_free_space": 90,
```

Large consumption of this storage space is not expected in normal usage as signed system logs are only written occasionally, and the HSM stores the logs in a compressed form.

5.13. Configure the hardserver to export the module for guest VM usage

This feature is not available on nToken models, but works with all other local HSM models. If you previously configured this feature using rserverperm, you may wish to update to using these instructions using the config file to specify the permissions for guest VMs to access the HSMs in a persistent manner.

1. Configure the host hardserver to permit guest VM hardservers to share access to the module:
   a. Edit the host hardserver config file `NFAST_KMDATA/config/config` (**Linux**) or `NFAST_KMDATA\config\config` (**Windows**).
   b. Add a new entry in the `hs_clients` section to contain the details of the client to be added.

If your config file does not already contain a `hs_clients` section you may add it yourself with a line containing only `[hs_clients]`.

The `addr` and `clientperm` fields are required for each client, and `keyhash` is

recommended for authentication: :

```
[hs_clients]
addr=<client_IP>
clientperm=permission_type
keyhash=software_keyhash
```

Where:

`<client_IP>` can be either the IP address of the guest VM or any of `0.0.0.0`, `::`, or blank if the host hardserver is to accept clients identified by their key hash instead of their IP address.

If you set both the `<client_IP>` field (the guest VM's IP address) and the key hash, client connections will be restricted based on both values.

`permission_type` defines the type of commands the client can issue (`unpriv` for unprivileged only, `priv` for privileged or `priv_lowport` for privileged connections restricted to low port numbers).

`software_keyhash` is the hash of the software-generated authentication key that the client should authenticate itself with.

If there is more than one client being configured, the fields for each client must be separated by line consisting of one or more hyphens (e.g. `----`).

It is recommended that the firewall on the host be configured so that only connections from intended network interfaces can be made to the host hardserver on its Impath port (port 9004 by default).

c. Load the updated configuration file in the host hardserver. To do this, run the following command:

```
hsc_nethsmexports
```

This command only needs to be run when the config is added or modified. The permissions for guest VMs will be re-applied automatically when the host hardserver is restarted.

2. Configure the hardserver in the guest VM to enroll to the host hardserver with an IP address using the virtual switch. Enter the following command for each guest hardserver that should have unprivileged access:

```
nethsmenroll <host-hardserver-ip>
```

Run the following command if the guest hardserver should have privileged access for mode change and administration:

Not all administration operations will be permitted from a privileged guest VM, such as firmware updates, which must be carried out from the host.

```
nethsmenroll -p <host-hardserver-ip>
```

You will be asked to confirm your entries. You should then see the following message:

```
OK configuring hardserver's nethsm imports
```

3. Confirm the connection from the guest VMs by running `enquiry`.

5.14. Logging, debugging, and diagnostics

5.14.1. Logging and debugging

When using a network-atttached HSM, you can view log files using the front panel. Application log messages are handled on the client.

The current release of Security World Software uses controls for logging and debugging that differ from those used in previous releases. However, settings you made in previous releases to control logging and debugging are still generally supported in the current release, although in some situations the output is now formatted differently.

Some text editors, such as Notepad, can cause `NFLOG` to stop working if the `NFLOG` file is open at the same time as the hardserver is writing the logs.

Debug logs may include sensitive data.

`CKNFAST_DEBUG`	All severities from `DL_Call` to `DL_DetailMutex` may result in sensitive data being logged
`JCECSP_DEBUG`	No further guidance
`NFJAVA_DEBUG`	No further guidance

`NFLOG_DETAIL`	The `0x00010000` flag may result in sensitive data being logged
`NFLOG_SEVERITY`	All the `DEBUGn` settings may result in sensitive data being logged

5.14.1.1. Environment variables to control logging

The Security World for nShield generates logging information that is configured through a set of four environment variables:

`NFLOG_FILE`

This environment variable specifies the name of a file (or a file descriptor, if prefixed with the `&` character) to which logging information is written. The default is `stderr` (the equivalent of `&2`).

Ensure that you have permissions to write to the file specified by `NFLOG_FILE`.

`NFLOG_SEVERITY`

This environment variable specifies a minimum severity level for logging messages to be written (all log messages less severe than the specified level are ignored). The level can be one of (in order of greatest to least severity):

1. `FATAL`
2. `SEVERE`
3. `ERROR`
4. `WARNING`
5. `NOTIFICATION`
6. `DEBUG`*N*, where *N* can be an integer from 1 to 10 inclusive that specifies increasing levels of debugging detail, with 10 representing the greatest level of detail, although the type of output is depends on the application being debugged.

The increasingly detailed information provided by different levels of `DEBUG`*N* is only likely to be useful during debugging, and we recommend not setting the severity level to `DEBUG`*N* unless you are directed to do so by Support.

The default severity level is `WARNING`.

`NFLOG_DETAIL`

This environment variable takes a hexadecimal value from a bitmask of detail flags as described in the following table (the `logdetail` flags are also used in the hardserver

configuration file to control hardserver logging; see *server_settings* in the *Hardserver configuration file* (**PCIe and USB HSMs**) or *HSM and client configuration files* (**network-attached HSMs**) chapter.

Hexadecimal flag	Function	logdetail flags
0x00000001	This flag shows the external time (that is, the time according to your machine's local clock) with the log entry. It is on by default.	external_time
0x00000002	This flag shows the external date (that is, the date according to your machine's local clock) with the log entry.	external_date
0x00000004	This flag shows the external process ID with the log entry.	external_pid
0x00000008	This flag shows the external thread ID with the log entry.	external_tid
0x00000010	This flag shows the external `time_t` (that is, the time in machine clock ticks rather than local time) with the log entry.	external_time_t
0x00000020	This flag shows the stack backtrace with the log entry.	stack_backtrace
0x00000040	This flag shows the stack file with the log entry.	stack_file
0x00000080	This flag shows the stack line number with the log entry.	stack_line
0x00000100	This flag shows the message severity (a severity level as used by the `NFLOG_SEVERITY` environment variable) with the log entry. It is on by default.	msg_severity
0x00000200	This flag shows the message category (a category as used by the `NFLOG_CATEGORIES` environment variable) with the log entry.	msg_categories
0x00000400	This flag shows message writeables, extra information that can be written to the log entry, if any such exist. It is on by default.	msg_writeable
0x00000800	This flag shows the message file in the original library. This flag is likely to be most useful in conjunction with Security World Software-supplied example code that has been written to take advantage of this flag.	msg_file

Hexadecimal flag	Function	logdetail flags
0x00001000	This flag shows the message line number in the original library. This flag is likely to be most useful in conjunction with example code we have supplied that has been written to take advantage of this flag.	msg_line
0x00002000	This flag shows the date and time in UTC (Coordinated Universal Time) instead of local time.	options_utc
0x00004000	This flag shows the full path to the file that issued the log messages.	options_fullpath
0x00008000	This flag includes the number of microseconds in the timestamp.	options_time_us
0x00010000	This flag enables logging of potentially secret values in generic stub log output.	msg_secrets

NFLOG_CATEGORIES

This environment variable takes a colon-separated list of categories on which to filter log messages (categories may contain the wild-card characters * and ?). If you do not supply any values, then all categories of messages are logged. This table lists the available categories:

Category	Description
nflog	Logs all general messages relating to nflog.
nflog-stack	Logs messages from `StackPush` and `StackPop` functions.
memory-host	Logs messages concerning host memory.
memory-module	Logs messages concerning module memory.
gs-stub	Logs general generic stub messages. (Setting this category works like using the `dbg_stub` flag with the logging functionality found in previous Security World Software releases.)
gs-stubbignum	Logs bignum printing messages. (Setting this category works like using the `dbg_stubbignum` flag with the logging functionality found in previous Security World Software releases.)
gs-stubinit	Logs generic stub initialization routines. (Setting this category works like using the `dbg_stubinit` flag with the logging functionality found in previous Security World Software releases.)

Category	Description
gs-dumpenv	Logs environment variable dumps. (Setting this category works like using the `dbg_dumpenv` flag with the logging functionality found in previous Security World Software releases.)
nfkm-getinfo	Logs `nfkm-getinfo` messages.
nfkm-newworld	Logs messages about world generation.
nfkm-admin	Logs operations using the Administrator Card Set.
nfkm-kmdata	Logs file operations in the `kmdata` (**Linux**) or `%NFAST_KMDATA%` (**Windows**) directory.
nfkm-general	Logs general NFKM library messages.
nfkm-keys	Logs key loading operations.
nfkm-preload	Logs `preload` operations.
nfkm-ppmk	Logs softcard operations.
serv-general	Logs general messages about the local hardserver.
serv-client	Logs messages relating to clients or remote hardservers.
serv-internal	Logs severe or fatal internal errors.
serv-startup	Logs fatal startup errors.
servdbg-stub	Logs all generic stub debugging messages.
servdbg-env	Logs generic stub environment variable messages.
servdbg-underlay	Logs messages from the OS-specific device driver interface
servdbg-statemach	Logs information about the server's internal state machine.
servdbg-perf	Logs messages about the server's internal queuing.
servdbg-client	Logs external messages generated by the client.
servdbg-messages	Logs server command dumps.
servdbg-sys	Logs OS-specific messages.
pkcs11-sam	Logs all security assurance messages from the PKCS #11 library.
pkcs11	Logs all other messages from the PKCS #11 library.
rqcard-core	Logs all card-loading library operations that involve standard message passing (including slot polling).
rqcard-ui	Logs all card-loading library messages from the current user interface.
rqcard-logic	Logs all card-loading library messages from specific logics.

You can set a minimum level of hardserver logging by supplying one of the values for the `NFLOG_SEVERITY` environment variable in the hardserver configuration file, and you can likewise specify one or more values for the `NFLOG_CATEGORIES` environment variable. For detailed information about the hardserver configuration file settings that control logging, see *server_settings* in the *Hardserver configuration file* (**PCIe and USB HSMs**) or *HSM and client configuration files* (**network-attached HSMs**) chapter.

If none of the four environment variables are set, the default behavior is to log nothing, unless this is overridden by any individual library. If any of the four variables are set, all unset variables are given default values.

5.14.1.2. Logging from the nShield CSP and CNG (Windows)

By default, logging is disabled for the nShield CSP and CNG.

To enable logging, use a suitable registry editor such as `regedit`.

Depending on whether the program you wish to debug is 64-bit or 32-bit based, you will have to create respective registry keys if they do not already exist.

For a 64-bit program on a 64-bit OS, create the following registry key if it does not already exist:

`HKEY_LOCAL_MACHINE\SOFTWARE\nCipher\Cryptography`

For a 32-bit program on a 64-bit OS, create the following registry key if it does not already exist:

`HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\nCipher\Cryptography`

Open the registry at the required `Cryptography` key as described above, and under the key create the following variables.

1. Create a new `string` variable named `PathName`.
2. Open the `PathName` variable and provide a value which is a suitable path to where you want the log file(s) to be placed (for example, `C:\Users\MyName\Documents`.) Do not give a log file name. The log file name(s) will be created automatically when logging starts.

   It must be possible for the provider to write to the specified path.

3. Create a new `DWORD (32 bit)` variable named `LogLevel`.
4. Open the `LogLevel` variable and provide a suitable value (for example, `2`).

Permitted values for `LogLevel` are:

Value	Output
0	Messages are sent to the event log.
1	Error messages are sent to the log file.
2	Function calls and error messages are sent to the log file.
3	All information, including debugging information, is sent to the log file.

Do not set this value to 3 unless either you are asked to do so by Support or you are debugging your own code. At this level of logging, a single SSL connection generates approximately 30 kilobytes of log messages. In addition, sensitive information may appear in the log file.

If `LogLevel` is not set, then the provider only logs messages of warning severity or greater (equivalent to setting `NFLOG_SEVERITY=warning`).

If neither `PathName` nor `LogLevel` are set for the CSP or CNG, logging remains disabled.

If logging is successfully enabled, the log file(s) should appear at the location specified in `PathName`, and will have names similar to:

- `nfdebug.txt`
- `ncspdddebug.txt`
- `nckspswdebug.txt`

5.14.1.3. Logging and debugging information for PKCS #11

In order to get PKCS #11 logging and debugging output, you must set the `CKNFAST_DEBUG` variable. A debug output file (with path) can also be set using the `CKNFAST_DEBUGFILE` variable. These variables can be set in the environment or the `/opt/nfast/cknfastrc` (**Linux**) or `%NFAST_HOME%\cknfastrc` (**Windows**) file. Normally settings in the environment should override the same settings (if any) in the `cknfastrc` file. You can specify any appropriate PKCS #11 categories using the `NFLOG_CATEGORIES` environment variable.

To produce PKCS #11 debug output, the `CKNFAST_DEBUG` variable can be a given value from 1 through to 11, where the greater the value the more detailed debug information is provided. A value of 7 is a reasonable compromise between too little and too much debug information. A value of 0 switches the debug output off.

This environment variable takes a colon-separated list of categories on which to filter log messages (categories may contain the wildcards characters * and ?).

The following table maps PKCS #11 debug level numbers to the corresponding NFLOG_SEVERITY value:

PKCS #11 debug level	PKCS #11 debug meaning	NFLOG_SEVERITY value	Output in log
0	DL_None	NONE	
1	DL_EFatal	FATAL	"Fatal error:"
2	DL_EError	ERROR	"Error:"
3	DL_Fixup	WARNING	"Fixup:"
4	DL_Warning	WARNING	"Warning:"
5	DL_EApplic	ERROR	"Application error:"
6	DL_Assumption	NOTIFICATION	"Unsafe assumption:"
7	DL_Call	DEBUG2	">> "
8	DL_Result	DEBUG3	"< "
9	DL_Arg	DEBUG4	"> "
10	DL_Detail	DEBUG5	"D "
11	DL_DetailMutex	DEBUG6	"DM "

5.14.1.4. Hardserver debugging (PCIe and USB HSMs)

Hardserver debugging is controlled by specifying one or more servdbg-* categories (from the NFLOG_CATEGORIES environment variable) in the hardserver configuration file; *server_settings* in the *Hardserver configuration file* chapter. However, unless you also set the NFAST_DEBUG environment variable to a value in the range 1 – 7, no debugging is produced (regardless of whether or not you specify servdbg-* categories in the hardserver configuration file). This behavior helps guard against the additional load debugging places on the CPU usage. You can set the desired servdbg-* categories in the hardserver configuration file, and then enable or disable debugging by setting the NFAST_DEBUG environment variable.

The NFAST_DEBUG environment variable controls debugging for the general stub or hardserver. The value is an octal number, in the range 1 – 7. It refers bitwise to a number of flags:

Flag	Result
1	Generic stub debugging value.
2	Show bignum values.
4	Show initial `NewClient` or `ExistingClient` command and response.

For example, if the `NFAST_DEBUG` environment variable is set to 6, flags 2 and 4 are used.

If the `NFAST_DEBUG` environment variable value includes flag 1 (Generic stub debugging value), the `logdetail` value in the hardserver configuration file (one of the values for the `NFLOG_DETAIL` environment variable) controls the level of detail printed.

Do not set the `NFAST_DEBUG` environment variable to a value outside the range 1 – 7. If you set it to any other value, the hardserver does not start.

5.14.1.5. Debugging information for Java

This section describes how you can specify the debugging information generated by Java.

5.14.1.5.1. Setting the Java debugging information level

In order to make the Java generic stub output debugging information, set the Java property `NFJAVA_DEBUG`. The debugging information for `NFJAVA`, `NFAST`, and other libraries (for example, `KMJAVA`) can all use the same log file and have their entries interleaved.

You set the debugging level as a decimal number. To determine this number:

1. Select the debugging information that you want from the following list:

```
NONE = 0x00000000 (debugging off)
MESS_NOTIFICATIONS = 0x00000001 (occasional messages including important errors)
MESS_VERBOSE = 0x00000002 (all messages)
MESS_RESOURCES = 0x00000004 (resource allocations)
FUNC_TRACE = 0x00000008 (function calls)
FUNC_VERBOSE = 0x00000010 (function calls + arguments)
REPORT_CONTEXT = 0x00000020 (calling context e.g ThreadID and time)
FUNC_TIMINGS = 0x00000040 (function timings)
NFJAVA_DEBUGGING = 0x00000080 (Output NFJAVA debugging info)
```

2. Add together the hexadecimal value associated with each type of debugging information.

   For example, to set `NFJAVA_DEBUGGING` and `MESS_NOTIFICATIONS`, add 0x00000080 and

0x00000001 to make 0x00000081.

3. Convert the total to a decimal and specify this as the value for the variable.

   For example, to set NFJAVA_DEBUGGING and MESS_NOTIFICATIONS, include the line:

   ```
   NFJAVA_DEBUG=129
   ```

   For NFJAVA to produce output, NFJAVA_DEBUG must be set to at least NFJAVA_DEBUGGING + MESS_NOTIFICATIONS. Other typical values are:

   - 255: All output
   - 130: nfjava debugging and all messages (NFJAVA_DEBUGGING and MESS_VERBOSE)
   - 20: function calls and arguments and resource allocations (FUNC_VERBOSE and MESS_RESOURCES)

5.14.1.5.2. Setting Java debugging with the command line

You can set the Java debug options by immediately preceding them with a -D. Use the NJAVA_DEBUGFILE property to direct output to a given file name, for example:

```
java -DNFJAVA_DEBUGFILE=myfile -DNFJAVA_DEBUG=129 -classpath .... classname
```

Do not set NFJAVA_DEBUG or NFJAVA_DEBUGFILE in the environment because Java does not pick up variables from the environment.

If NFJAVA_DEBUGFILE is not set, the standard error stream System.err is used.

Set these variables only when developing code or at the request of Support.

Debug output contains all commands and replies sent to the hardserver in their entirety, including all plain texts and the corresponding cipher texts as applicable.

5.14.1.6. appliance-cli: system logs utility

See appliance-cli

5.14.2. Diagnostics and system information

Besides the diagnostic tools described in this section, we also supply a performance tool that you can use to test Web server performance both with and without an nShield HSM. This tool is supplied separately. If you require a copy, contact your Sales representative.

5.14.2.1. NIC Reporting (network-attached HSMs)

Networking information regarding NIC (Network Interface Card) status and address details can be reported via the `stattree` output.

NIC status and address details are disabled by default. To enable or disable NIC details in the `stattree` output is done via the configuration menu, **System** > **System Configuration** > **NIC exposure config** > **Device Status** and select the appropriate setting (**Yes** or **No**). Similarly for enabling or disabling the NIC addressing which is done via the **Address Reporting** option.

Reporting NIC addressing is only performed when device status is enabled.

A typical `stattree` report with NIC status and address details enabled looks like:

```
+#HostSysInfo:
   +SystemFans:
      +#1:
         -CurrentFanRPM        6240
      +#2:
         -CurrentFanRPM        6240
      +#3:
         -CurrentFanRPM        6240
      +#4:
         -CurrentFanRPM        6300
   +NetworkInterfaces:
      +#1:
         -InterfaceName        eth0
         -InterfaceLinkState   up
         +NetworkAddresses:
            +#1:
               -InterfaceNetworkAddress 172.23.135.127
      +#2:
         -InterfaceName        eth1
         -InterfaceLinkState   down
         +NetworkAddresses:
```

5.14.2.2. nShield HSM tamper log (network-attached HSMs)

The nShield HSM tamper log contains:

- The date and time of any tamper events
- Whether the tamper detection functionality was ever disabled or re-enabled.

It is no longer possible to disable tamper detection.

You view the tamper log using the nShield HSM front panel, by selecting **System** > **System information** > **View tamper log**. You cannot erase the tamper log.

5.14.3. How data is affected when a module loses power and restarts

nShield modules use standard RAM to store many kinds of data, and data stored in such RAM is lost in the event that a module loses power (either intentionally, because you turned off power to it, or accidentally because of a power failure).

Therefore, after restoring power to a module, you must reload any keys that had been loaded onto it before it lost power. After reloading, the `KeyID`s are different.

Likewise, after restoring power to a module, you must reload any cards that were loaded onto it before it lost power.

However, data stored in NVRAM is unaffected when a module loses power.

If you are using multiple nShield modules in the same Security World, and have the same key (or keys) loaded onto each module as part of a load-sharing configuration, loss of power to one module does not affect key availability (as long as at least one other module onto which the keys are loaded remains operational). However, in such a multiple-module system, after restoring power to a module, you must still reload any keys to that module before they can be available from that module.

5.15. 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 and (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.

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.

5.15.1. 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.

5.15.2. 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.

5.15.3. Init log

The `init` log records information that is produced each time the system boots.

The amount of information recorded depends on the firmware version loaded on the HSM.

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 Recovery Mode.

5.15.3.1. 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.

5.15.3.2. 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 13.5 onwards it is not necessary to clear the `init` log because it is cleared automatically each time the system successfully boots.

5.15.4. 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.

5.15.4.1. 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.

5.15.4.1.1. 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.

5.15.4.2. Retrieving the system log

5.15.4.2.1. 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.

5.15.4.2.2. 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.

5.15.4.3. 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.

5.15.4.3.1. 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.

5.15.4.3.2. 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.

5.15.4.4. 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://nshieldsupport.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.

5.16. 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.

5.16.1. 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 battery 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 information regarding error reporting.

Contact Support to request information regarding a replacement battery if `stattree` reports the battery voltage to be below 2.70V.

See Battery replacement for instructions on replacing the battery in your module.

5.16.2. 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. Calculated `stattree` statistics such as minima and maxima are reset on module reboot.

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 v13.6.5 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

`stattree` Statistic	Description	Minimum expected in optimum environment	Maximum expected in optimum environment
`MaxTempC` (**PCIe HSMs**)	Maximum temperature measured on either processor	-	78°C
`MinTempC` (**PCIe HSMs**)	Minimum temperature measured on either processor	10°C	-
`CurrentTempC` (**network-attached HSMs**)	Internal temperature 1	10°C	45°C
`CurrentTemp2C` (**network-attached HSMs**)	Internal temperature 2	10°C	45°C
`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.

5.17. 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.

5.17.1. 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.

5.17.1.1. 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.

5.17.1.2. 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.

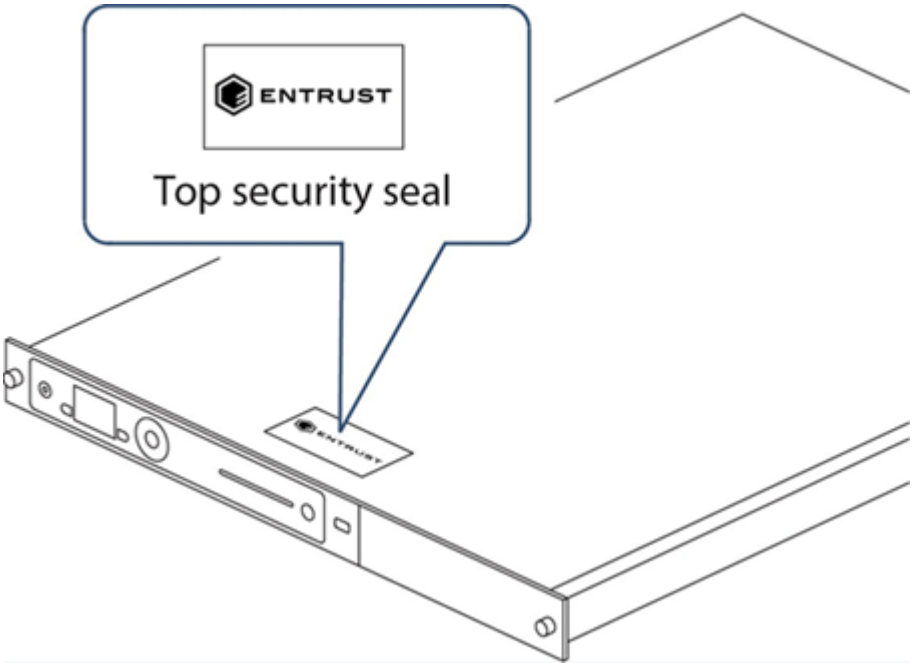
After closing the lid you must reboot the nShield HSM. The unit will then automatically reset to a factory state. If the lid remains open, the above message will remain on the screen and all button presses are ignored.

5.17.2. 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.

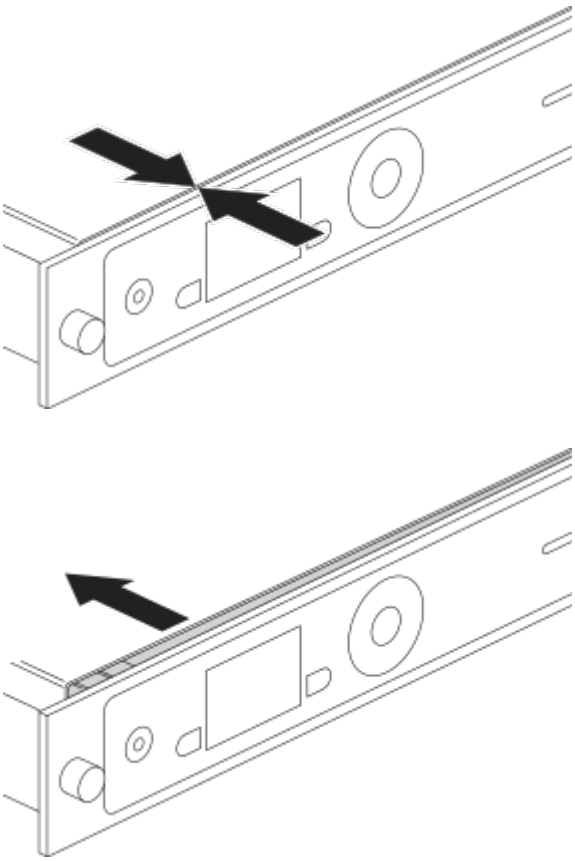
To determine if the security of the nShield HSM is compromised:

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*.

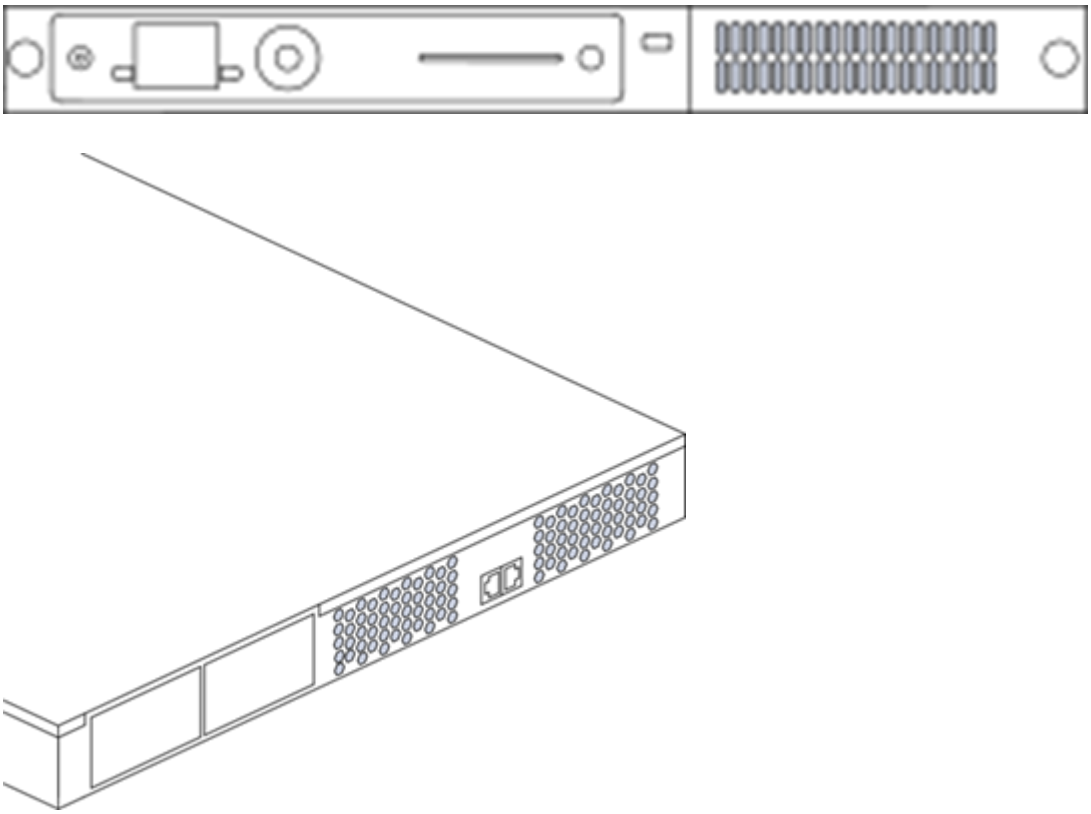
2. Check that the metal lid remains flush with the nShield HSM chassis.



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.



5.17.3. 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.

5.17.3.1. 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*.

5.17.3.1.1. 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.

5.17.3.2. 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*.

5.17.3.2.1. 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.

5.17.3.3. 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.

5.18. Application Performance Tuning

5.18.1. Job Count

To achieve the best throughput of cryptographic jobs (such as Sign or Decrypt) in your application, arrange for multiple jobs to be on the go at the same time, rather than doing them one at a time. This is true even when using only a single HSM in your system.

When using an nShield HSM, Entrust recommend that you set the number of outstanding jobs within the `rec. queue` (recommended queue) range specified by the `enquiry` output for the module.

If you are sending single jobs synchronously from each thread of your client application, try to keep the number of threads within this `rec. queue` range for best throughput.

When using higher-level APIs, such as PKCS#11, your application could benefit from increasing the thread count above the `rec. queue` range or the number that gives the best throughput when using nCore directly.

If you are load-balancing across multiple HSMs and want to maximize throughput across all of them, then use the sum of all `rec. queue` ranges for each of the modules to set the target for the outstanding jobs.

The `ncperftest` utility supports performance measurements of a range of cryptographic operations with different job counts and client thread counts. You may find this useful to inform tuning of your application. Run `ncperftest --help` to see the available options.

5.18.2. Client Configuration

If your application is coded directly against nCore, you have a choice of sending multiple jobs asynchronously from a single client connection to the hardserver, or having multiple threads each with their own client connection to the hardserver with a single job sent synchronously in each. You can use the `--threads` parameter to the `ncperftest` utility to experiment with the performance impact of having more threads/connections with fewer jobs outstanding in each, or having fewer or just one thread/connection with more jobs outstanding in that connection.

When using higher-level APIs such as PKCS#11, all cryptographic operations are synchronous, so larger numbers of threads must be used to increase the job count and make full use of HSM resources. These APIs automatically create a hardserver connection for each thread. If many HSMs are being used, a great many threads may be required to achieve best throughput. You can adjust the thread counts in the performance test tools for these APIs (for example, `cksigtest` for PKCS#11) to gauge how much concurrency is required for best throughput in your application.

5.18.3. Highly Multi-threaded Client Applications

If your application is highly multi-threaded, operating system defaults may not be optimal for best performance:

You may benefit from using a scalable memory allocator that is designed to be efficient in multi-threaded applications, examples include `tcmalloc`.

On some systems the default operating system scheduling algorithm is also not optimized for highly multi-threaded applications. A real-time scheduling algorithm such as the POSIX round-robin scheduler may yield noticeable performance improvements for your application.

5.18.4. File Descriptor Limits (Linux)

On Linux systems, large numbers of threads each with their own hardserver connection will require your application to make use of large numbers of file descriptors. It may be necessary to increase the file descriptor limit for your application. This can be done using `ulimit -n NewLimit` on most systems, but you may need to increase system-wide hard limits first.

5.19. Platform services and (nShield 5 HSMs)

The nShield HSM firmware provides multiple services. These are divided into platform services and the `ncoreapi` service.

5.19.1. 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.

5.19.1.1. Multi-tenancy ready

The system has been prepared for use in multi-tenant systems. In the current firmware version, only one instance of the `ncoreapi` service is allowed to run at any one time. Future versions of firmware will allow multiple instances of the `ncoreapi` service to run concurrently.

5.19.2. Platform services

Several platform services are provided which perform the tasks associated with the installation, commissioning, and maintenance of the HSM firmware and hardware. These run

independently of the `ncoreapi` service.

The platform services are

Service name	Function
`updater`	This services provides functions to upgrade the HSM firmware
`setup`	This service provides functions to view the HSM 'lifetime' data installed in the factory and to return the HSM to factory settings
`monitor`	This service provides functions to retrieve and clear logs stored within the HSM
`sshadmin`	This service provides functions to manage the SSH keys used by the platform services and the `ncoreapi` service
`launcher`	Launcher service. On versions with CodeSafe 5 support, this is used for starting CodeSafe 5 applications on the HSM.

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

An interlock mechanism prevents most platform services from being accessed when 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
```

5.19.3. Separation of services

Each of the platform services and the `ncoreapi` 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 is described in Set up communication between host and module (nShield 5s HSMs).

5.20. System upgrade

5.20.1. 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.

5.20.2. 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 behaviour of v13.5 software interacting with v13.5 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.

5.20.3. 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.

5.20.3.1. 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.

5.20.3.1.1. 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.

5.20.3.1.2. 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.

5.20.3.1.3. 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.

5.21. Product returns

If you need to return your nShield HSM, contact Entrust nShield Support for instructions: https://nshieldsupport.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