



NSO Installation Guide

Release: NSO 5.7.1

Published: April 9, 2014

Last Modified: January 26, 2022

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit <http://www.openssl.org/>.

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

U.S. Pat. No. 8,533,303 and 8,913,519

Copyright © 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Introduction 1

- Purpose 1
- Prerequisites 1
- NED Packages 2
- Man Pages 2
- Unpack NSO Installer 3
- Install Types 4

CHAPTER 2

NSO Local Install 5

- Introduction 5
- Installation 5
- NSO Upgrade 8
 - Upgrade from Local Install to System Install 8
- NSO UnInstallation 10

CHAPTER 3

NSO System Install 11

- Introduction 11
- Installation 11
- NSO Upgrade 15
- NSO Uninstallation 16

CHAPTER 4

Other Information 17

- References 17
- FAQs 17
- Support 18

CHAPTER I

Manual pages 19

- ncs-installer 21
- ncs-uninstall 25



CHAPTER

1

Introduction

Cisco Network Service Orchestrator (NSO) version 5.7.1 is an evolution of the Tail-f Network Control System (NCS). Tail-f was acquired by Cisco in 2014. The product has been enhanced and forms the base for Cisco NSO. Note that the terms 'ncs' and 'tail-f' are used extensively in file names, command-line command names, YANG models, application programming interfaces (API), etc. Throughout this document we will use NSO to mean the product, which consists of a number of tools and executables. These executable components will be referred to by their command line name, e.g. **ncs**, **ncs-netsim**, **ncs_cli**, etc.

- [Purpose, page 1](#)
- [Prerequisites, page 1](#)
- [NED Packages, page 2](#)
- [Man Pages, page 2](#)
- [Unpack NSO Installer, page 3](#)
- [Install Types, page 4](#)

Purpose

The purpose of the document is to describe how to Install NSO step-by-step and to explain how to perform basic operations based on Installation.

Target Audience

- The document is intended for evaluators, developers, system administrators, end users and others who would like to install NSO.

Installation can be performed in two ways.

- 1 Local Install for evaluation and development.
- 2 System Install for production deployment.

Prerequisites

NSO Supports Linux and OS X platforms. On Linux platforms NSO can be deployed in bare-metal as well as in virtualized (virtual machines) or containerized (containers) environments.

You need to know your system specifications (Operating System and CPU architecture) to choose the appropriate NSO Installer.

NSO is delivered as a self-extract archive which is OS/CPU specific. The archive file has the pattern `nso-VERSION.OS.ARCH.installer.bin`. The variables in the pattern refer to:

VERSION The NSO version to install.
OS The Operating System (`linux` for all Linux distributions and `darwin` for OS X).
ARCH The CPU architecture (`x86_64`).

Dependencies

- Before installing NSO, ensure that a Java JDK-8.0 or higher is installed. When JDK is properly installed the command **java -version** should indicate a java version of “1.8” or higher.
- The supported target NSO releases for a NSO CDM release migration are from 4.x.y releases with `x >= 5`. For earlier NSO target releases the migration needs to be performed in two steps. First upgrade to a NSO release higher than 4.5 and verify that this works and then as a second step do the NSO migration to the NSO 5.x release. For more information, follow NSO CDM Migration Guide
- After upgrading from a version below “4.7”, it is recommended to delete the default webui application meta data in `webui:webui/webui-one:applications/application`. This is to make sure the correct data is shown in the application hub.
- Python 3.7 or higher is required.
- For building NSO Examples, install:
 - Ant *1.9.3 or higher*
- For using netconf-console, install:
 - Python Paramiko *2.2 or higher*
- For NSO GUI capabilities, install one of the below browsers.
 - Safari *Version must be supported by Apple at the time of release*
 - Firefox *Version must be supported by Mozilla.org at the time of release*
 - Chrome *Version must be supported by Google at the time of release*
 - Edge *Version must be supported by Microsoft at the time of release*
- To allow for generation of SSH keys of currently recommended type, during NSO installation, install:
 - OpenSSH *6.5 or higher*

NED Packages

The NED Packages that are available with the NSO Installation are netsim based example NEDs. These NEDs are used for NSO examples only.

You should fetch latest production grade NEDs from `software.cisco.com` using the URLs provided on your NED license certificates.

Man Pages

The installation program will unpack the NSO manual pages from the documentation archive in `$NCS_DIR/man`. 'ncsrc' makes an addition to `$MANPATH`, allowing you to use the '**man**' command to view them.

Here follows few list of the manual pages installed:

`ncs(1)` Command to start and control the NSO daemon.

```

ncsc(1)          NSO Yang compiler.
ncs_cli(1)       Frontend to the NSO CLI engine.
ncs-netsim(1)    Command to create and manipulate a simulated network.
ncs-setup(1)     Command to create an initial NSO setup.
ncs.conf         NSO daemon configuration file format.

```

For example, to view the man page describing NSO configuration file you should type:

```
$ man ncs.conf
```

Apart from the man pages, extensive information about command line options can be obtained by running **ncs** and **ncsc** with the `--help` (abbreviated `-h`) flag.

```
$ ncs --help
```

```
$ ncsc --help
```

Unpack NSO Installer

The NSO installation is delivered with a cryptographic signature to make it possible to verify that the installation has not been altered or corrupted. This can be guaranteed since the signature was created by using a combination of cryptographic hash and public key encryption.

Unpack of the NSO Installer, automatically verifies the digital signature.

```
$ sh nso-VERSION.OS.ARCH.signed.bin
```

The variables in the command `VERSION` refers to the NSO version to install, `OS` refers to the Operating System (`linux` for any Linux distribution and `darwin` for OS X) and `ARCH` refers to the CPU architecture (`x86_64`). For Example:

```
$ sh nso-4.2.linux.x86_64.signed.bin
```

Once NSO is unpacked, all the files are unpacked in the current directory.

- 1 The NSO installer `nso-VERSION.OS.ARCH.installer.bin`
- 2 Signature generated for the NSO image `nso-VERSION.OS.ARCH.installer.bin.signature`
- 3 An enclosed Cisco signed `taif.cer` x.509 end-entity certificate containing public key that is used to verify the signature.
- 4 `README.signature` file which briefs you more details on the unpacked content and the steps on "How to run the signature verification program". If you would like to manually verify the signature, please refer to the steps in this file.
- 5 `cisco_x509_verify_release.py` python program that can be used to verify the 3-tier x.509 certificate chain and signature.



Note

If you observe network connectivity issues that might impact the signature verification, you can unpack the installer by using parameter `--skip-verification`. For example:

```
$ sh nso-4.2.linux.x86_64.signed.bin --skip-verification
```

Once the unpack is completed, the NSO `nso-VERSION.OS.ARCH.installer.bin` installation can be performed either by Local Install or System Install as described in the next section.

Install Types

Local Install

- Use Local Install `--local-install` option for Development, Evaluation, proof of concept and private lab purposes.
- All the NSO Examples and README steps provided with the installation are based on Local Install only. You should always use Local Install for evaluation and development purposes.
- Steps for Local Installation, please follow [Chapter 2, NSO Local Install](#).
- Local Install is possible on Linux OS and OS X.

System Install

- Use System Install `--system-install` option for production and system-wide deployment in a central location. You need root privileges for System Install procedure and administration of the installed NSO. As part of System install, the NSO daemon **ncs** is started at boot time.
System Install should be used only for production deployment. For all other purposes, use Local Install procedure.
- All the NSO examples and README steps are based on Local Install only. These cannot be run on System Install.
- Steps for System Installation, Please follow [Chapter 3, NSO System Install](#).
- NSO System Install is possible only on Linux OS.



CHAPTER 2

NSO Local Install

- [Introduction, page 5](#)
- [Installation, page 5](#)
- [NSO Upgrade, page 8](#)
- [NSO UnInstallation, page 10](#)

Introduction

Local Install should always be used for evaluation and development purposes. All the NSO examples and README steps provided with the installation are based on Local Install only. For production deployment, use System Install.

Before installing NSO:

- 1 Choose the correct Operating System. Local Install is supported on Linux and OS X for x86_64 architecture.
- 2 Before installing NSO, ensure that Java JDK-7.0 or higher is installed. For building and running NSO examples install applications mentioned in [Chapter 1, Introduction Prerequisites](#) section.

Installation

Procedure 2.1. Local Install Steps

Step 1 Local Install of NSO Software is performed in a single user specified directory, for example in your home directory \$HOME. It is always recommended to install NSO in a directory named as the version of the release.

```
$ sh nso-VERSION.OS.ARCH.installer.bin $HOME/ncs-VERSION --local-install
```

The variables in the command VERSION refers to the NSO version to install, OS refers to the Operating System (linux for any Linux distribution and darwin for OS X) and ARCH refers to the CPU architecture (x86_64). The --local-install parameter is an optional parameter. For Example:

```
$ sh nso-4.0.linux.x86_64.installer.bin $HOME/nso-4.0
```

Step 2 The installation program creates a shell script file named ncsrc in each NSO installation, which sets the environment variables. Source this file to get these settings in your shell. You may want to add this sourcing command to your login sequence, such as .bashrc.

For csh/tcsh users there is a ncsrc.tcsh file using csh/tcsh syntax. The example below assumes that you are using bash, other versions of /bin/sh may require that you use . instead of source.

Step 3

```
$ source $HOME/ncs-VERSION/ncsrc
```

Create a runtime directory for NSO to keep its database, state files, logs and other files. In these instructions we assume that this directory is `$HOME/ncs-run`.

Step 4

```
$ ncs-setup --dest $HOME/ncs-run
```

Start the NSO daemon `ncs`.

```
$ cd $HOME/ncs-run
```

```
$ ncs
```

When you start NSO, make sure that you are situated in the deployment directory since by default NSO reads its config files from its current working directory (i.e. `.`).

There are a set of examples available in the installation directory `$NCS_DIR/examples.ncs`

Please go through the examples for information on how to create run-time directories, start `ncs`, and other important NSO functionalities.

Step 5

NSO uses Cisco Smart Licensing, as described in Chapter 3, *Cisco Smart Licensing in NSO 5.7 Administration Guide*, to make it easy to deploy and manage NSO license entitlements. To conclude the NSO installation a license registration token must be created using a [Cisco Smart Software Manager \(CSSM\)](#) account. Login credentials to a CSSM account should have been provided by your Cisco contact and detailed instructions on how to create a registration token can be found in Chapter 3, *Cisco Smart Licensing in NSO 5.7 Administration Guide*.

**Note**

If you intend to use NSO instance for development purposes you should enable development mode using the command **license smart development enable**. When development mode is enabled the NSO instance will only consume a single development NSO license entitlement and nothing else.

When you have a token, start a Cisco CLI towards NSO and enter the token, i.e.

```
admin@ncs# license smart register idtoken YzIzMDM3MTgtZTRkNC00YjkxLTk2ODQtOGEzMTM3OTg5MG
Registration process in progress.
Use the 'show license status' command to check the progress and result.
```

The command **show license status** can be used to keep track of the registration status, e.g

```
admin@ncs# show license status
```

```
Smart Licensing is ENABLED
```

```
Registration:
```

```
Status: REGISTERED
Smart Account: Network Service Orchestrator
Virtual Account: Default
Export-Controlled Functionality: Allowed
Initial Registration: SUCCEEDED on Apr 21 09:29:11 2016 UTC
Last Renewal Attempt: SUCCEEDED on Apr 21 09:29:16 2016 UTC
Next Renewal Attempt: Oct 18 09:29:16 2016 UTC
Registration Expires: Apr 21 09:26:13 2017 UTC
Export-Controlled Functionality: Allowed
```

```
License Authorization:
```

```
License Authorization:
```

```
Status: IN COMPLIANCE on Apr 21 09:29:18 2016 UTC
Last Communication Attempt: SUCCEEDED on Apr 21 09:26:30 2016 UTC
Next Communication Attempt: Apr 21 21:29:32 2016 UTC
Communication Deadline: Apr 21 09:26:13 2017 UTC
```

**Note**

During Upgrades, If you find 'Communication Send Error' while license registration, please restart the Smart Agent.

Licensing activities are also logged in the NSO daemon log as described in the section called “Monitoring NSO” in *NSO 5.7 Administration Guide* . For example, a successful token registration results in the following log entry:

```
<INFO> 21-Apr-2016::11:29:18.022 miosaterm confd[8226]:
  Smart Licensing Global Notification:
    type = "notifyRegisterSuccess"
```

If no registration token is provided NSO enters a 90 days evaluation period and the remaining evaluation time is recorded hourly in the NSO daemon log:

```
...
<INFO> 13-Apr-2016::13:22:29.178 miosaterm confd[16260]:
  Starting the NCS Smart Licensing Java VM
<INFO> 13-Apr-2016::13:22:34.737 miosaterm confd[16260]:
  Smart Licensing evaluation time remaining: 90d 0h 0m 0s
...
<INFO> 13-Apr-2016::13:22:34.737 miosaterm confd[16260]:
  Smart Licensing evaluation time remaining: 89d 23h 0m 0s
...
```

Upon successful registration NSO automatically requests a license entitlement for its own instance and for the number of devices it orchestrates and their NED types. If development mode has been enabled only an development entitlement for the NSO instance itself is requested.

The requested entitlements can be inspected using the command **show license all** (or by inspecting the NSO daemon log):

```
admin@ncs# show license all
...
<INFO> 21-Apr-2016::11:29:18.022 miosaterm confd[8226]:
  Smart Licensing Global Notification:
    type = "notifyRegisterSuccess",
    agentID = "sal",
    enforceMode = "notApplicable",
    allowRestricted = false,
    failReasonCode = "success",
    failMessage = "Successful."
<INFO> 21-Apr-2016::11:29:23.029 miosaterm confd[8226]:
  Smart Licensing Entitlement Notification: type = "notifyEnforcementMode",
    agentID = "sal",
    notificationTime = "Apr 21 11:29:20 2016",
    version = "1.0",
    displayName = "regid.2015-10.com.cisco.NSO-network-element",
    requestedDate = "Apr 21 11:26:19 2016",
    tag = "regid.2015-10.com.cisco.NSO-network-element",
    enforceMode = "inCompliance",
    daysLeft = 90,
    expiryDate = "Jul 20 11:26:19 2016",
    requestedCount = 8
...
```

**Tip**

In a situation where the NSO instance has no direct access to the Cisco Smart Software Manager one option is a [Cisco Smart Software Manager satellite](#) can be installed to manage software licenses on the premises. Install the satellite and use the command **call-home destination address http <url:port>** to point to the satellite.

Another option when direct access is not wanted is to configure an HTTP or HTTPS proxy, e.g.: **smart-license smart-agent proxy url https://127.0.0.1:8080** If you plan to do this, take the note below regarding ignored CLI configurations into account.

**Tip**

When configuring NSO in High Availability (HA) mode the license registration token must be provided to the CLI running on the primary node. Read more about HA and node types in Chapter 7, *High Availability* in *NSO 5.7 Administration Guide*.

**Note**

Note that if ncs.conf contains configuration for any of java-executable, java-options, override-url/url or proxy/url under the configure path /ncs-config/smart-license/smart-agent/ any corresponding configuration done via the CLI is ignored.

NSO Upgrade

Upgrade from Local Install to System Install

This procedure is for the users that have a Local Install in production deployment and would like to change to System Install.

Assuming NSO is installed as mentioned in [Chapter 2, NSO Local Install](#), and you want to do a new installation as mentioned in [Chapter 3, NSO System Install](#) then the steps to upgrade NSO are:

**Note**

As NSO 5.3+ adds support for 256 bit AES encrypted strings, when upgrading from earlier versions, make sure you add a AES256CFB128 key to your encrypted-strings ncs.conf configuration. Optionally, if you use the external command feature, make sure that you add an AES256CFB128_KEY to the output.

One way of generating such a key is: **openssl rand -hex 32**.

Step 1 Install the new NSO version.

```
$ sudo sh nso-NEWVERSION.OS.ARCH.installer.bin --system-install
```

Step 2 Stop previously installed NSO.

```
$ ncs --stop
```

Step 3 Take a complete backup (for disaster recovery)

```
$ tar -czf $HOME/ncs-backup.tar.gz -C $HOME ncs-run
```

Step 4 Change to Super User privileges.

```
$ sudo -s
```

Step 5 Make sure the symbolic link points to the ncs-NEWVERSION.

```
# cd /opt/ncs
# rm -f current
# ln -s ncs-NEWVERSION current
```

Step 6 Make sure that the /var/opt/ncs/packages directory has packages that are appropriate for the new version. When upgrading from one version of NSO to another version in the same major branch i.e x.y are unchanged in nso-x.y.z (Ex: nso-5.7 to nso-5.7.1), it should be possible to continue to use the same packages. But when upgrading to a different major branch, the packages normally need to be rebuilt (or pre-built packages for the new major branch need to be used).

Step 7 Copy the CDB files from the local install to the central location.

```
# cp $HOME/ncs-run/ncs-cdb/*.cdb /var/opt/ncs/cdb
```

Step 8 Disable local-authentication in /etc/ncs/ncs.conf

```
<local-authentication>
  <enabled>false</enabled>
</local-authentication>
```

Step 9 Start NSO with package reload option.

```
# NCS_RELOAD_PACKAGES=true /etc/init.d/ncs start
```

Step 10 A license registration token must be created using a Cisco Smart Software Manager (CSSM) account. Follow the step 5 in System Install steps from [the section called “Installation”](#)

Step 11 Update log file paths for JAVA and Python VM by logging into ncs_cli.

```
$ ncs_cli -C -u admin
admin@ncs# config
Entering configuration mode terminal
admin@ncs(config)# unhide debug
admin@ncs(config)# show full-configuration java-vm stdout-capture file
java-vm stdout-capture file ./logs/ncs-java-vm.log
admin@ncs(config)# java-vm stdout-capture file /var/log/ncs/ncs-java-vm.log
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# show full-configuration java-vm stdout-capture file
java-vm stdout-capture file /var/log/ncs/ncs-java-vm.log
admin@ncs(config)# show full-configuration python-vm logging log-file-prefix
python-vm logging log-file-prefix ./logs/ncs-python-vm
admin@ncs(config)# python-vm logging log-file-prefix /var/log/ncs/ncs-python-vm
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# show full-configuration python-vm logging log-file-prefix
python-vm logging log-file-prefix /var/log/ncs/ncs-python-vm
admin@ncs(config)# exit
admin@ncs#
admin@ncs# exit
```

Step 12 Verify that everything is working.

Step 13 If you want to switch back to previous known/good state, start with stopping NSO.

```
# /etc/init.d/ncs stop
```

Exit from Super User privileges.

```
# exit
```

Remove runtime dir.

```
$ rm -rf $HOME/ncs-run
```

Restore runtime dir from backup.

```
$ tar -xzf $HOME/ncs-backup.tar.gz -C $HOME
```

Setup environment for the old version.

```
$ source $HOME/ncs-VERSION/ncsrc
```

Start NSO.

```
$ cd $HOME/ncs-run
$ ncs
```

NSO UnInstallation

Uninstallation of NSO is very easy. A single delete of the NSO installation directory and runtime directory is sufficient. Ensure to stop NSO before uninstall.



Note

To make sure that no license entitlements are consumed after you have uninstalled NSO be sure to perform the **deregister** command in the CLI:

```
admin@ncs# license smart deregister
```

```
$ rm -rf $HOME/ncs-VERSION
$ rm -rf $HOME/ncs-run
```



CHAPTER 3

NSO System Install

- [Introduction, page 11](#)
- [Installation, page 11](#)
- [NSO Upgrade, page 15](#)
- [NSO Uninstallation, page 16](#)

Introduction

NSO System Install results in system wide Installation and Deployment. Use this Install option only for Production deployment. For Development and Evaluation purposes, you should use only Local Install procedure.

Before installing NSO:

- 1 Ensure that the root permissions are enabled.
- 2 Choose the correct Operating System (Linux). Currently only Linux OS is supported.
- 3 Ensure that Java JDK-8.x or 11.x is installed.

Installation

Procedure 3.1. System Install Steps

Step 1

Use `--system-install` option to perform system installation. This option creates a system install of NSO, suitable for deployment. By default, Installation Directory is created in `/opt/ncs`, Configuration Directory is created in `/etc/ncs`, Running Directory is created in `/var/opt/ncs` and Log Directory is created in `/var/log/ncs`.

```
$ sudo sh nso-VERSION.OS.ARCH.installer.bin --system-install
```

The variables in the command `VERSION` refers to the NSO version to install. `OS` refers to the Operating System (`linux`). `ARCH` refers to the CPU architecture (`x86_64`). For example:

```
$ sudo sh nso-4.2.linux.x86_64.installer.bin --system-install
```

Here, as part of the system installation, in the Installation directory `/opt/ncs` “ncs-4.2” distribution is available. A symbolic link `/opt/ncs/current` is created, pointing to the newly installed version, in this example `/opt/ncs/ncs-4.2`

In the Configuration directory `/etc/ncs/ncs.conf` file, SSH keys, WebUI certificates are created. In Run directory `/var/opt/ncs` ncs run-time state files, CDB database, packages are created.

In Log directory `/var/log/ncs` ncs log files are populated. Also init scripts are created in `/etc/init.d/ncs` and system wide environment variables are created in `/etc/profile.d/ncs.sh`.

The installation has been configured for PAM authentication, with group assignment based on the OS group database (e.g. `/etc/group` file). Users that need access to NSO must belong to either the 'ncsadmin' group (for unlimited access rights) or the 'ncsoper' group (for minimal access rights).

To create the 'ncsadmin' group, use OS shell command:

```
# groupadd ncsadmin
```

To create the 'ncsoper' group, use OS shell command:

```
# groupadd ncsoper
```

To add an existing user to one of these groups, use OS shell command:

```
# usermod -a -G 'groupname' 'username'
```

For `--system-install` option, you can also choose user-defined (non-default) Installation Directory, Config Directory, Running Directory, Log Directory with `--install-dir`, `--config-dir`, `--run-dir` and `--log-dir` parameters, and specify that NSO should run as different user than `root` with the `--run-as-user` parameter.

For more information on ncs-installer see the [ncs-installer\(1\)](#) man page.

For an extensive guide to NSO deployment, please refer to Chapter 10, *NSO Deployment* in *NSO 5.7 Administration Guide*.



Note

If you choose a non-default installation directory by using `--install-dir`, you need to specify `--install-dir` for subsequent installs and also for backup and restore.



Note

If the Linux kernel allows overcommit of memory (this is the default), it introduces an unexpected and unreliable environment for NSO, since the “OOM killer” may terminate NSO in such a way that it is not restarted. Thus it is *strongly* recommended that overcommit is disabled.

To achieve this with immediate effect, give the command:

```
# echo 2 > /proc/sys/vm/overcommit_memory
```

To make sure that overcommit remains disabled across reboot, add this line to `/etc/sysctl.conf` file:

```
vm.overcommit_memory = 2
```

For details of this parameter, refer to the Linux *proc(5)* manual page.

Step 2

Change to Super User privileges.

```
$ sudo -s
```


Step 3 The installation program creates a shell script file in each NSO installation which sets environment variables needed to run NSO. With `--system-install` option, by default these settings are set on the shell. To explicitly set the variables, source `ncs.sh` or `ncs.csh` depending on your shell type.

Step 4 `# source /etc/profile.d/ncs.sh`
Start NSO.

`# /etc/init.d/ncs start`



Note Once you log on with the user that belongs to `ncsadmin` or `ncsoper` you can directly access the CLI.

`$ ncs_cli -C`

Step 5 NSO uses Cisco Smart Licensing, as described in Chapter 3, *Cisco Smart Licensing in NSO 5.7 Administration Guide*, to make it easy to deploy and manage NSO license entitlements. To conclude the NSO installation a license registration token must be created using a [Cisco Smart Software Manager \(CSSM\)](#) account. Login credentials to a CSSM account should have been provided by your Cisco contact and detailed instructions on how to create a registration token can be found in Chapter 3, *Cisco Smart Licensing in NSO 5.7 Administration Guide*.



Note If you intend to use NSO instance for development purposes you should enable development mode using the command **license smart development enable**. When development mode is enabled the NSO instance will only consume a single development NSO license entitlement and nothing else.

When you have a token, start a Cisco CLI towards NSO and enter the token, i.e.

```
admin@ncs# license smart register idtoken YzIzMMDM3MTgtZTRkNC00YjkxLTk2ODQ0tOGEzMTM3OTg5MG
Registration process in progress.
Use the 'show license status' command to check the progress and result.
```

The command **show license status** can be used to keep track of the registration status, e.g

```
admin@ncs# show license status

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Smart Account: Network Service Orchestrator
  Virtual Account: Default
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on Apr 21 09:29:11 2016 UTC
  Last Renewal Attempt: SUCCEEDED on Apr 21 09:29:16 2016 UTC
  Next Renewal Attempt: Oct 18 09:29:16 2016 UTC
  Registration Expires: Apr 21 09:26:13 2017 UTC
  Export-Controlled Functionality: Allowed

License Authorization:

License Authorization:
  Status: IN COMPLIANCE on Apr 21 09:29:18 2016 UTC
  Last Communication Attempt: SUCCEEDED on Apr 21 09:26:30 2016 UTC
  Next Communication Attempt: Apr 21 21:29:32 2016 UTC
  Communication Deadline: Apr 21 09:26:13 2017 UTC
```

**Note**

During Upgrades, If you find 'Communication Send Error' while license registration, please restart the Smart Agent.

Licensing activities are also logged in the NSO daemon log as described in the section called “Monitoring NSO” in *NSO 5.7 Administration Guide*. For example, a successful token registration results in the following log entry:

```
<INFO> 21-Apr-2016::11:29:18.022 miosaterm confd[8226]:
  Smart Licensing Global Notification:
    type = "notifyRegisterSuccess"
```

If no registration token is provided NSO enters a 90 days evaluation period and the remaining evaluation time is recorded hourly in the NSO daemon log:

```
...
<INFO> 13-Apr-2016::13:22:29.178 miosaterm confd[16260]:
  Starting the NCS Smart Licensing Java VM
<INFO> 13-Apr-2016::13:22:34.737 miosaterm confd[16260]:
  Smart Licensing evaluation time remaining: 90d 0h 0m 0s
...
<INFO> 13-Apr-2016::13:22:34.737 miosaterm confd[16260]:
  Smart Licensing evaluation time remaining: 89d 23h 0m 0s
...
```

Upon successful registration NSO automatically requests a license entitlement for its own instance and for the number of devices it orchestrates and their NED types. If development mode has been enabled only an development entitlement for the NSO instance itself is requested.

The requested entitlements can be inspected using the command **show license all** (or by inspecting the NSO daemon log):

```
admin@ncs# show license all
...
<INFO> 21-Apr-2016::11:29:18.022 miosaterm confd[8226]:
  Smart Licensing Global Notification:
    type = "notifyRegisterSuccess",
    agentID = "sal",
    enforceMode = "notApplicable",
    allowRestricted = false,
    failReasonCode = "success",
    failMessage = "Successful."
<INFO> 21-Apr-2016::11:29:23.029 miosaterm confd[8226]:
  Smart Licensing Entitlement Notification: type = "notifyEnforcementMode",
    agentID = "sal",
    notificationTime = "Apr 21 11:29:20 2016",
    version = "1.0",
    displayName = "regid.2015-10.com.cisco.NSO-network-element",
    requestedDate = "Apr 21 11:26:19 2016",
    tag = "regid.2015-10.com.cisco.NSO-network-element",
    enforceMode = "inCompliance",
    daysLeft = 90,
    expiryDate = "Jul 20 11:26:19 2016",
    requestedCount = 8
...
```

**Tip**

In a situation where the NSO instance has no direct access to the Cisco Smart Software Manager one option is a [Cisco Smart Software Manager satellite](#) can be installed to manage software licenses on the premises. Install the satellite and use the command **call-home destination address http <url:port>** to point to the satellite.

Another option when direct access is not wanted is to configure an HTTP or HTTPS proxy, e.g.: **smart-license smart-agent proxy url https://127.0.0.1:8080** If you plan to do this, take the note below regarding ignored CLI configurations into account.

**Tip**

When configuring NSO in High Availability (HA) mode the license registration token must be provided to the CLI running on the primary node. Read more about HA and node types in Chapter 7, *High Availability* in *NSO 5.7 Administration Guide*.

**Note**

Note that if ncs.conf contains configuration for any of java-executable, java-options, override-url/url or proxy/url under the configure path /ncs-config/smart-license/smart-agent/ any corresponding configuration done via the CLI is ignored.

NSO Upgrade

Assuming NSO is installed as mentioned in [Chapter 3, NSO System Install](#), then the steps to upgrade NSO to a new version are:

**Note**

As NSO 5.3+ supports 256 bit AES encrypted strings, when upgrading from earlier versions, make sure you add a AES256CFB128 key to your encrypted-strings ncs.conf configuration. Optionally, if you use the external command feature, make sure that you add an AES256CFB128_KEY to the output.

One way of generating such a key is: **openssl rand -hex 32**.

Step 1

Install the new version.

```
$ sudo sh nso-NEWVERSION.OS.ARCH.installer.bin --system-install
```

Step 2

Change to Super User privileges.

```
$ sudo -s
```

Step 3

Stop previously installed NSO.

```
# /etc/init.d/ncs stop
```

Step 4

Take a complete backup (for disaster recovery)

```
# ncs-backup
```

Step 5

Switch to new version by updating symbolic link.

```
# cd /opt/ncs
# rm -f current
# ln -s ncs-NEWVERSION current
```

- Step 6** Make sure that the `/var/opt/ncs/packages` directory has packages that are appropriate for the new version. When upgrading from one version of NSO to another version in the same major branch i.e x.y are unchanged in nso-x.y.z (Ex: nso-5.7 to nso-5.7.1), it should be possible to continue to use the same packages. But when upgrading to a different major branch, the packages normally need to be rebuilt (or pre-built packages for the new major branch need to be used).
- Step 7** Start NSO with package reload option.
- ```
NCS_RELOAD_PACKAGES=true /etc/init.d/ncs start
```
- Step 8** A license registration token must be created using a Cisco Smart Software Manager (CSSM) account. Follow the step 5 in System Install steps from [the section called “Installation”](#)
- Step 9** Verify that everything is working.
- Step 10** If you want to switch back to previous known/good state, start by stopping NSO.
- ```
# /etc/init.d/ncs stop
```
- Then restore from backup.
- ```
ncs-backup --restore
```
- Move back symbolic link
- ```
# cd /opt/ncs
# rm -f current
# ln -s ncs-VERSION current
```
- And restart NSO.
- ```
/etc/init.d/ncs start
```

## NSO Uninstallation

- NSO can be uninstalled using the [ncs-uninstall\(1\)](#) option only if NSO is installed with `--system-install` option. Either part of the static files or full installation can be removed using `ncs-uninstall` option. Ensure to stop NSO before Uninstall.



### Note

To make sure that no license entitlements are consumed after you have uninstalled NSO be sure to perform the **deregister** command in the CLI:

```
admin@ncs# license smart deregister
```

- `# ncs-uninstall --all`

Removes the Installation directory `/opt/ncs` including symbolic links, Configuration directory `/etc/ncs`, Run directory `/var/opt/ncs`, Log directory `/var/log/ncs`, init scripts from `/etc/init.d` and user profile scripts from `/etc/profile.d`

- For more information on ncs uninstall, see the [ncs-uninstall\(1\)](#) man page.



## CHAPTER 4

# Other Information

---

- [References, page 17](#)
- [FAQs, page 17](#)
- [Support, page 18](#)

## References

After successful installation, to get started with NSO refer to "NSO Getting Started Guide" available with NSO Installer. Also available in `$NCS_DIR/doc/ncs_getting_started.pdf`

Refer to "NSO Users Guide" for detailed information on NSO. Available in `$NCS_DIR/doc/ncs_user_guide.pdf`

In addition to the User Guide, we recommend you to browse through the variety of useful examples found under `$NCS_DIR/examples.ncs`. Each example has a README file explaining its purpose and usage.

For NSO System Administration details refer to "NSO Administration Guide" available in `$NCS_DIR/doc/ncs_admin_guide.pdf`

## FAQs

Below you can find few frequently asked questions or common mistakes by customers.

1. Is there a dependency between the NSO Installation Directory and Runtime Directory?  
No, there is no such dependency.
2. Do you need to source the `ncsrc` file before starting NSO?  
Local Install - Yes.  
System Install - No. (By default, the environment variables are configured and set on the shell while system install).
3. Can you start NSO from a directory, which is not a NSO runtime directory?  
Local Install - No (To start NSO, you need to point to the Run directory).  
System Install - Yes.
4. Can you stop NSO from a directory, which is not a NSO runtime directory?  
Local Install - Yes.  
System Install - Yes.

5. For evaluation and development purposes, instead of Local Install you made a System Install. Now you cannot build or run NSO examples as described in README files. How can you proceed further?

The easiest method is to Uninstall system installation using `ncs-uninstall --all` and do a Local Install from scratch.

6. Can we move NSO Installation from one folder to another ?

Local Install - Yes.

You can move the directory where you installed NSO to a new location in your directory tree. Simply move NSO's root directory to the new desired location, and update this file: `$NCS_DIR/ncsrc` (and `ncsrc.tcsh` if you want) This is a small and handy script that sets up some environment variables for you. Update the paths to the new location. The `$NCS_DIR/bin/ncs` and `$NCS_DIR/bin/ncsc` scripts will determine the location of NSO's root directory automatically.

System Install - No.

## Support

- If you are evaluating NSO, you should have a designated support contact.
- If you have a NSO support agreement, please use the support channels specified in the agreement.
- In either case, please do not hesitate to send us any questions or feedback you may come up with.

# Manual pages





## Name

ncs-installer — NCS installation script

## Synopsis

```
ncs-VSN.OS.ARCH.installer.bin [--local-install] LocalInstallDir
```

```
ncs-VSN.OS.ARCH.installer.bin --system-install [--install-dir InstallDir] [--config-dir ConfigDir] [--run-dir RunDir] [--log-dir LogDir] [--run-as-user User] [--keep-ncs-setup] [--non-interactive]
```

## DESCRIPTION

The NCS installation script can be invoked to do either a simple "local installation", which is convenient for test and development purposes, or a "system installation", suitable for deployment.

## LOCAL INSTALLATION

```
[--local-install]
LocalInstallDir
```

When the NCS installation script is invoked with this option, or is given only the *LocalInstallDir* argument, NCS will be installed in the *LocalInstallDir* directory only.

## SYSTEM INSTALLATION

```
--system-install
```

When the NCS installation script is invoked with this option, it will do a system installation that uses several different directories, in accordance with Unix/Linux application installation standards. The first time a system installation is done, the following actions are taken:

- The directories described below are created and populated.
- An init script for start of NCS at system boot is installed.
- User profile scripts that set up \$PATH and other environment variables appropriately for NCS users are installed.
- A symbolic link that makes the installed version the currently active one is created (see the --install-dir option).

```
[--install-dir
InstallDir]
```

This is the directory where static files, primarily the code and libraries for the NCS daemon, are installed. The actual directory used for a given invocation of the installation script is *InstallDir*/ncs-VSN, allowing for coexistence of multiple installed versions. The currently active version is identified by a symbolic link *InstallDir*/current pointing to one of the ncs-VSN directories. If the --install-dir option is omitted, /opt/ncs will be used for *InstallDir*.

```
[--config-dir ConfigDir
]
```

This directory is used for config files, e.g. ncs.conf. If the --config-dir option is omitted, /etc/ncs will be used for *ConfigDir*.

```
[--run-dir RunDir]
```

This directory is used for run-time state files, such as the CDB data base and currently used packages. If the --run-dir option is omitted, /var/opt/ncs will be used for *RunDir*.

[ --log-dir *LogDir* ]

This directory is used for the different log files written by NCS. If the --log-dir option is omitted, /var/log/ncs will be used for *LogDir*.

[ --run-as-user *User* ]

By default, the system installation will run NCS as the root user. If a different user is given via this option, NCS will instead be run as that user. The user will be created if it does not already exist. This mode is only supported on Linux systems that have the **setcap** command, since it is needed to give NCS components the required capabilities for some aspects of the NCS functionality.

When the option is used, the following executable files (assuming that the default /opt/ncs is used for --install-dir) will be installed with elevated privileges:

/opt/ncs/current/lib/  
ncs/lib/core/pam/priv/  
epam

Setuid to root. This is typically needed for PAM authentication to work with a local password file. If PAM authentication is not used, or if the local PAM configuration does not require root privileges, the setuid-root privilege can be removed by using **chmod u-s**.

/opt/ncs/current/lib/  
ncs/erts/bin/ncs /opt/  
ncs/current/lib/ncs/  
erts/bin/ncs.smp

Capability  
cap\_net\_bind\_service. One of these files (normally ncs.smp) will be used as the NCS daemon. The files have execute access restricted to the user given via --run-as-user. The capability is needed to allow the daemon to bind to ports below 1024 for northbound access, e.g. port 443 for HTTPS or port 830 for NETCONF over SSH. If this functionality is not needed, the capability can be removed by using **setcap -r**.

/opt/ncs/current/lib/  
ncs/bin/ip

Capability  
cap\_net\_admin. This is a copy of the OS **ip(8)** command, with execute access restricted to the user given via --run-as-user. The program is not used by the core NCS daemon, but provided for packages that need to configure IP addresses on interfaces (such as the **tailf-hcc** package).

/opt/ncs/current/lib/  
ncs/bin/arping

If no such packages are used, the file can be removed.

Capability `cap_net_raw`. This is a copy of the OS **arping(8)** command, with execute access restricted to the user given via `--run-as-user`. The program is not used by the core NCS daemon, but provided for packages that need to send gratuitous ARP requests (such as the `tailf-hcc` package). If no such packages are used, the file can be removed.



---

**Note**

When the `--run-as-user` option is used, all OS commands executed by NCS will also run as the given user, rather than as the user specified in e.g. [clispec\(5\)](#) definitions for custom CLI commands.

[ `--keep-ncs-setup` ]

---

The **ncs-setup** command is not usable in a "system installation", and is therefore by default excluded from such an installation to avoid confusion. This option instructs the installation script to include **ncs-setup** in the installation despite this.

[ `--non-interactive` ]

If this option is given, the installation script will proceed with potentially disruptive changes (e.g. modifying or removing existing files) without asking for confirmation.



## Name

ncs-uninstall — Command to remove NCS installation

## Synopsis

```
ncs-uninstall --ncs-version [Version] [--install-dir InstallDir] [--non-interactive]
```

```
ncs-uninstall --all [--install-dir InstallDir] [--non-interactive]
```

## DESCRIPTION

The **ncs-uninstall** command can be used to remove part or all of an NCS "system installation", i.e. one that was done with the `--system-install` option to the NCS installer (see [ncs-installer\(1\)](#)).

## OPTIONS

|                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>--ncs-version [ <i>Version</i> ]</code>    | Removes the installation of static files for NCS version <i>Version</i> . I.e. the directory tree rooted at <i>InstallDir/ncs-Version</i> will be removed. The <i>Version</i> argument may also be given as the filename or pathname of the installation directory, or, unless <code>--non-interactive</code> is given, omitted completely in which case the command will offer selection from the installed versions. |
| <code>--all</code>                               | Completely removes the NCS installation. I.e. the whole directory tree rooted at <i>InstallDir</i> , as well as the directories for config files (option <code>--config-dir</code> to the installer), run-time state files (option <code>--run-dir</code> to the installer), and log files (option <code>--log-dir</code> to the installer), and also the init script and user profile scripts.                        |
| <code>[ --install-dir <i>InstallDir</i> ]</code> | Specifies the directory for installation of NCS static files, like the <code>--install-dir</code> option to the installer. If this option is omitted, <code>/opt/ncs</code> will be used for <i>InstallDir</i> .                                                                                                                                                                                                       |
| <code>[ --non-interactive ]</code>               | If this option is used, removal will proceed without asking for confirmation.                                                                                                                                                                                                                                                                                                                                          |

