

# Troubleshoot NSO Smart License Issues

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Problem](#)

[Troubleshoot Issue 1](#)

[Troubleshoot Issue 2](#)

[Troubleshoot Issue 3](#)

[Related Information](#)

## Introduction

This document describes how to troubleshoot license issues on Network Service Orchestrator (NSO) with Cisco Smart Software Manager (CSSM).

## Prerequisites

### Requirements

There are no specific requirements for this document.

### Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

## Problem

Errors encountered with NSO on CSSM are:

1. result Error: License Agent not ready. Please try again.
2. Failure Reason: Communication send error.
3. java.lang.module.FindException: Module java.xml.bind not found (From the ncs-smart-licensing.log)

You must check the next commands/outputs first.

- Verify that the smart-agent runs:

```
[root@nso-ha-1 home]# ps -ef | grep smart
root      63522  63419  0 06:56 ?        00:00:00
```

```

/opt/ncs/current/lib/ncs/lib/core/sls/priv/agentwrapper
java -Xmx64M -Xms16M -Djava.security.egd=file:/dev/./urandom -jar
/opt/ncs/current/lib/ncs/lib/core/sls/priv/webapp-runner.jar
/opt/ncs/current/lib/ncs/lib/core/sls/priv/smartagent --port 0 --path /smartagent --shutdown-
override root      63523  63522  4 06:56 ?          00:02:45 java -Xmx64M -Xms16M -
Djava.security.egd=file:/dev/./urandom -jar
/opt/ncs/current/lib/ncs/lib/core/sls/priv/webapp-runner.jar
/opt/ncs/current/lib/ncs/lib/core/sls/priv/smartagent
--port 0 --path /smartagent --shutdown-override [root@nso-ha-1 home]# systemctl status
ncs.service >... 63523 java -Xmx64M -Xms16M -Djava.security.egd=file:/dev/./urandom -jar
/opt/ncs/current/lib/ncs/lib/core/sls/priv/webapp-runner.jar
/opt/ncs/current/lib/ncs/lib/core/sls/priv/smartagent -
-port 0 --path /smartagent --shutdown-overr... ... Apr 13 06:56:15 nso-ha-1 ncs[63406]: -
Starting the NCS Smart Licensing Java VM root@ncs# show smart-license smart-agent start-status
smart-license smart-agent start-status started

```

- Confirm which method is used to register the token (direct/proxy/satellite). Once determined, verify if there is connectivity from the NSO server to the server:

```

wget https://www.cisco.com wget <IP of proxy or satellite server> curl -vvv <IP of proxy or
satellite server>

```

## Example

Proxy: [DDCEService](#)

Satellite: [DeviceRequestHandler](#)

- The next step is to verify where the smart-license related parameters have been configured. There are two ways to do this. One is to have the smart-license section under the ncs.conf file:

```

<smart-license>    <smart-agent>
  <java-executable>DEFAULT</java-executable>
  <java-options>DEFAULT</java-options>
  <production-url>DEFAULT</production-url><
  <alpha-url>DEFAULT</alpha-url>
  <override-url>
    <url>DEFAULT</url>
  </override-url>
  <proxy>
    <url>DEFAULT</url>
  </proxy>
</smart-agent>
</smart-license>

```

The other is to have the configurations made from cli.

```
admin@ncs# show running-config smart-license smart-agent
```

**Note:** On any system, you can only one of the two configurations. The configurations under ncs.conf always supersede the ones made from CLI.

- After that, check the license-related information with the next commands:

```
root@ncs# show license ?
```

Possible completions:

UDI	Show the device UDI.
all	Show all Smart Licensing information.
status	Show overall Smart Licensing status.
summary	Show Smart Licensing status summary.
tech-support	Show Smart Licensing tech support information.
trace	Show Smart Licensing trace status.
usage	Show Smart Licensing license usage.

The previous commands give an idea about all your current licenses and if it is InCompliance , OutofCompliance , Erroneous , Eval , and so on.

- Finally, the smart license logs can be enabled with the next configuration:

```
root@ncs# show running-config smart-license
smart-license smart-agent stdout-capture enabled
smart-license smart-agent stdout-capture file /var/log/ncs/ncs-smart-licensing.log
smart-license smart-agent stdout-capture stdout
```

## Troubleshoot Issue 1

With the error "License agent is not ready", check the smart-license that runs config and the ncs.conf file. If the smart-license section exists in the ncs.conf file, comment it out and configure the next syntax from the NSO CLI:

```
root@ncs# config
Entering configuration mode terminal
root@ncs(config)#smart-license smart-agent java-options "-Xmx64M -Xms16M -
Djava.security.egd=file:/dev/./urandom"
root@ncs(config)#smart-license smart-agent override-url url
```

The <URL> can be a proxy server as shown earlier or a satellite server. In the case of a satellite server, it is important to ensure that the port is "443" and HTTP is used. (8443 is used for GUI). Alternatively, if HTTP works, then the URL must be modified accordingly.

After you make these changes, the smart-agent restarts:

```
root@ncs(config)# smart-license smart-agent restart
result Started
```

There is a possibility that even after the previous changes, the license agent cannot start. Another configuration that can hinder the changes is related to 'HA'. In that situation, a quick check can be in the ncs.conf file to confirm if HA is enabled or disabled.

**Note:** On an HA setup, smart-licensing must be enabled (configured) only on the Primary node.

With these changes, the smart-license agent is ready for the next operation. Try to register the token to check.

## Troubleshoot Issue 2

Even when the smart-license agent works well, you can encounter the "Communication send error". Verify the override-url and ensure the ports work. As mentioned earlier, the connections must also be validated.

If there is nothing wrong with NSO, this error can be due to the connection between Smart Software Manager On-Prem Satellite (SSMS) and CSSM. There can be a change in the cName to match the FQDN that is not fully-synchronized with the CSSM, which is a mandatory step. If you do not do this step, there are cause problems while the token generated from CSSM is generated. This can cause this error:

```
Failure Reason: {"token": "The token '<Actual Token from CSSM>' is not valid."}]}
```

For information about how to register NSO with https, refer to [How to register your Device with HTTPS to Satellite Smart-licensing Server.](#)

## Troubleshoot Issue 3

The third issue happens when there is a problem with the java version that is used. The smart-agent is essentially a JAVA process (NCS Smart Licensing Java VM), it is imperative that the right java version is used.

This is the error seen:

```
Error occurred during initialization of boot layer
java.lang.module.FindException: Module java.xml.bind not found
child died (From the ncs-smart-licensing.log)
```

Java 11 removed the hints about a lost java module. If this issue is encountered, check the version and also if JAVA was recently upgraded first.

For details about how to configure Cisco Smart Licenses for NSO, refer to [Configure Cisco Smart Licenses with NSO.](#)

## Related Information

- [Technical Support & Downloads - Cisco Systems](#)