Understand NSO NED Trace Logs

Contents

Introduction

This document describes how to properly read and understand various phases of the NSO NED trace.

Background

The Cisco® Crosswork Network Service Orchestrator (NSO) can generate detailed traces of the communication between NSO and the devices managed by NSO's Network Element Drivers (NEDs). For Java-based NEDs, these trace files include are structured according to a shared framework. This document helps you understand this shared framework and the details in these logs.

Prerequisites

Components Used

- NSO 6.4
- NED cisco-iosxr-cli
- NED cisco-ios-cli
- NED cisco-nx-cli
- NED zte-zxros-cli
- NED alu-sr-cli
- NED juniper-junos_nc-gen

Disclaimer

This document assumes you are using a Java NED developed by Cisco. This includes CLI, Generic, and 3PY NEDs. The phases and framework explained in this document applies to Netconf NEDs, but the trace logs generated for Netconf NEDs do not label them in the same way as shown in this document.

While the phases described in this document are shared across all Java NEDs, specific operations run as part of that phase differ for each NED due to individual device needs.

Types of data

Data in a NED trace file can be viewed in 3 different categories.

Communication between Core NSO and the NED

NSO instructs the NED to start specific phases. Each operation in NSO results in the same sequences of phases but each NED runs unique instructions towards a network device. The NED trace file does not log the exact data flowing between NSO and the NED but it does log the instruction to start a phase, and the NED response indicating a phase has completed. Lines starting with >> indicate instructions to start a phase. Lines starting with << indicate the NED informing NSO the phase has finished.

```
>> 20-Mar-2025::23:23:17.277 user: admin/56 thandle 86091 hostname ncs device xr-netsim0 trace-id=1ee09
<< 20-Mar-2025::23:23:17.623 user: admin/56 thandle 86091 hostname ncs device xr-netsim0 trace-id=1ee09
>> 20-Mar-2025::23:24:41.703 user: admin/56 thandle 86213 hostname ncs device xr-netsim0 trace-id=1ee09
```

```
<< 20-Mar-2025::23:24:41.879 user: admin/56 thandle 86213 hostname ncs device xr-netsim0 trace-id=1ee09
```

Communication between the NED and a southbound device

Within each phase, the NED will run commands towards the network device to achieve the goals of each phase. Communication from the NED to the device is marked as *** output, communication the NED receives from the device is marked as *** input.

```
*** output 20-Mar-2025::13:08:31.955 user: admin/316551 thandle 18978916 hostname ncs device xr-netsim0
show running-config
*** input 20-Mar-2025::13:08:31.987 user: admin/316551 thandle 18978916 hostname ncs device xr-netsim0
show running-config

Thu Jan 5 13:08:37.274 BRT
Building configuration...
!! IOS XR Configuration 7.2.1
...
```

In CLI NEDs the input consists of all information which appears on the CLI of the device, including the commands sent by NSO.

Internal logging by the NED itself

The NED logs a certain amount of information which does not get relayed to either the device or to NSO. This includes ned-settings, the processing of data, and the expected changes in NSO database(CDB).

This type of information often starts with-- but that does not apply to all information of this type.

```
*** output 20-Mar-2025::13:08:31.955 user: admin/316551 thandle 18978916 hostname ncs device xr-netsim0
-- BEGIN SHOW
-- [20-Mar-2025::13:08:31.956] progress: show: reading config...
-- Reading running config...
show running-config
```

Sequences by Operation

This section goes through a list of common operations and documents the expected sequence of phases NSO initiates for each operation. More details for each phase can be found in the Phases section of this document.

Note: The IS_ALIVE, SET_TIMEOUT, and CLOSE phases are omitted from each sequence as they have little troubleshooting value

Connect

```
>> CLI CONNECT
<< CONNECTED
```

Or

```
>> GENERIC CONNECT
<< CONNECTED
```

While functionally, the CLI and GENERIC CONNECT phases are nearly identical, CLI and GENERIC type NEDs use different CONNECT phases.

Check-Sync

```
>> CLI CONNECT
<< CONNECTED
>> GET_TRANS_ID
<< TRANS_ID
```

Sync-From

```
>> CLI CONNECT
<< CONNECTED
>> SHOW
<< SHOW
>> GET_TRANS_ID
<< TRANS_ID
```

Or

```
>> CLI CONNECT
<< CONNECTED
>> SHOW
<< SHOW
<< PROVISIONAL TRANS_ID
```

Some NEDs have been optimized to use PROVISIONAL TRANS_ID instead of GET_TRANS_ID.

Compare-Config

```
>> CLI CONNECT
<< CONNECTED
>> SHOW
<< SHOW
```

The compare-config operation is very similar to sync-from, but it does not update the CDB. When compare-config detects a configuration difference, it does not invoke GET_TRANS_ID to update the checksum. When there is no configuration difference it does invoke GET_TRANS_ID and update the checksum.

Commit

```
>> CLI CONNECT
<< CONNECTED
>> INITIALIZE
<< INITIALIZED
>> PREPARE
<< PREPARE OK
>> COMMIT
<< COMMIT OK
>> PERSIST
<< PERSIST OK
>> GET_TRANS_ID
<< TRANS_ID
```

Commit dry-run and Commit no-networking

These operations do not engage the NED logic and do not cause any logs to be generated in the NED trace file.

Commit Dry-Run Outformat Native

This operation does not send any data to the network devices, but it does engage the NED logic.

```
>> CLI CONNECT
<< CONNECTED
>> PREPARE DRY
<< PREPARE DRY
```

Commit No-Overwrite

```
>> CLI CONNECT
<< CONNECTED
>> INITIALIZE
<< INITIALIZED
>> SHOW_PARTIAL
<< SHOW
>> PREPARE
<< PREPARE OK
>> COMMIT
<< COMMIT OK
>> PERSIST
<< PERSIST OK
>> GET_TRANS_ID
<< TRANS_ID
```

This sequence is misleading. It claims to include the INITIALIZE phase but the NED will not trigger any commands during this phase and effectively skips it. This is because the no-overwrite flag does not check the checksum but checks the configuration using SHOW_PARTIAL instead.

Commit with Error

```
>> CLI CONNECT
<< CONNECTED
>> INITIALIZE
<< INITIALIZED
>> PREPARE
>> CLOSE
<< CLOSED
>> CLI CONNECT
<< CONNECTED
>> SHOW_PARTIAL
<< SHOW
>> ABORT
<< ABORT OK
>> GET_TRANS_ID
<< TRANS_ID
```

When something goes wrong during a commit, NSO terminates the connection, reconnects, and attempts to restore the system. NSO does this by checking the configuration of the device using SHOW_PARTIAL and sending a sequence of commands to revert the current configuration to the configuration before the start of the commit during the ABORT phase. Devices with candidate configuration have alternative methods to recover which NSO can leverage, depending on when the error occurred.

Live-Status

```
>> CLI CONNECT
<< CONNECTED
>> COMMAND
<< COMMAND
```

Phases

All java-based NEDs use the same phases, but each NED adjusts the exact execution of the phase to the specific device it is handling.

CLI CONNECT, GENERIC CONNECT, and RECONNECT

During the CONNECT phase the NED prints information about itself, establishes a connection, disables pagination(for CLI NEDs), and collects device information. This includes the NSO and NED version, the NED settings, SSH algorithms, and the device model and version. It does not record the password exchange.

When encountering a failure for NSO to connect to a device, any part of this phase can be responsible. It is possible NSO was able to establish the SSH session but failed to retrieve the device model and version.

NEDs maintain a session with a device for several seconds after an operation has finished. If an other operation for the same device is required within that time frame, the NED will log a RECONNECT phase instead of CLI/GENERIC CONNECT and re-use the information.

GET_TRANS_ID

The GET_TRANS_ID phase collects information to calculate a checksum. This checksum can be verified to determine whether a device is out-of-sync during a commit or check-sync operation, or it can be stored for a future check. Cisco selects the most light-weight option available for each device. The cisco-ios-cli NED collects the full device configuration to generate a checksum. The cisco-iosxr NED uses the commit list and verifies whether the commit-id has changed since the last sync.

NEDs print the checksum they calculated at the end of the GET_TRANS_ID phase.

```
>> 15-Mar-2025::10:29:41.410 user: admin/205 thandle 1559 hostname ncs device alu0 GET_TRANS_ID
  *** output 15-Mar-2025::10:29:41.411 user: admin/205 thandle 1559  hostname ncs device alu0 ***
-- get config method: cli dump
admin display-config

  *** input 15-Mar-2025::10:29:41.415 user: admin/205 thandle 1559  hostname ncs device alu0 ***
admin display-config
...
<< 15-Mar-2025::10:29:42.045 user: admin/205 thandle 1559  hostname ncs device alu0 TRANS_ID 8f42fe893c
```

GET_TRANS_ID is invoked during check-sync, at the end of sync-from, at the end of commit, or at the end of compare-config if no diff was detected. Only during check-sync is the checksum not updated as part of GET_TRANS_ID. At the start of a commit operation, NSO also verifies the checksum but uses INITIALIZE instead of GET_TRANS_ID.

SHOW

During the SHOW phase a NED collects the current configuration on the device and parses it so it can be updated or compared to CDB. A SHOW phase can consist of one or more commands to collect the relevant data. Some NEDs prompt multiple SHOW phases in a row for different sections of the configuration.

```
<< 15-Mar-2025::14:17:07.190 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
>> 15-Mar-2025::14:17:07.210 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
<< 15-Mar-2025::14:17:07.211 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
>> 15-Mar-2025::14:17:07.211 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
<< 15-Mar-2025::14:17:07.212 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
>> 15-Mar-2025::14:17:07.212 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
<< 15-Mar-2025::14:17:07.212 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
>> 15-Mar-2025::14:17:07.213 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
<< 15-Mar-2025::14:17:07.213 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
>> 15-Mar-2025::14:17:07.214 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
<< 15-Mar-2025::14:17:07.214 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
>> 15-Mar-2025::14:17:07.214 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
<< 15-Mar-2025::14:17:07.215 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
>> 15-Mar-2025::14:17:07.215 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
<< 15-Mar-2025::14:17:08.672 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
<< 15-Mar-2025::14:17:08.672 user: admin/17279 thandle 7896374 hostname ncs device zte0 trace-id=438b8c
```

Once the data has been collected, the NED parses and prepares it for NSO. In CLI NEDs this process is marked by turbo-parser. Any data NSO does not understand and cannot map to the current yang model is logged as a skipped line.

```
-- turbo-mode parsing (setvalues) :: performance numbers :
-- ---------------------------------------------------------------------------------
--   number of lines parsed           :    469
--   number of lines skipped          :      2
--   time to parse config (ms)        :     20
--   time to transfer xml to nso (ms) :    531
-- ---------------------------------------------------------------------------------
--   skipped 2 lines in context '/' :
--    (line 16) : 'platform sslvpn use-pd'
--    (line 74) : 'pae'
-- ---------------------------------------------------------------------------------
-- [15-Mar-2025::17:00:07.906] progress: show: populating cdb ok [531 ms]
-- transformed >= sorted 8 'neighbor ' lines for hash checksum
-- show trans_id = 12b6f28a48520ca4b5c6ebdfe3d333ee
-- done show [5055 ms]
<< 15-Mar-2025::17:00:07.912 user: nsoadmin/23219 thandle 3068964 hostname ncs device ios0 SHOW
```

PROVISIONAL TRANS_ID

PROVISIONAL TRANS_ID is an optimization implemented in some NEDs to replace GET_TRANS_ID at the end of a sync-from operation. Without this optimization NEDs which are required to collect the full configuration of a device to calculate a checksum collect this data twice during sync-from. Once during the SHOW phase and again during the GET_TRANS_ID phase. PROVISIONAL TRANS_ID replaces GET_TRANS_ID in these situations to re-use the data from the SHOW operation.

A special implementation of this optimization exists in the cisco-iosxr-cli NED. This NED does not require the full configuration but configuration can be so big that parsing takes a long enough time for the SSH session to time out by the time GET_TRANS_ID starts. To avoid this, the NED collects the necessary information as part of SHOW and uses PROVISIONAL TRANS_ID instead.

INITIALIZE

The INITIALIZE phase is similar to GET_TRANS_ID. It collects the same data and calculates a checksum, but is only used at the start of a commit operation to verify whether a device is in-sync. If a device is found to not be in sync the INITIALIZE phase is followed by an UNINITIALIZE phase and an error is returned to NSO. INITIALIZE never updates the checksum.

Note: Commit no-overwrite does have an INITIALIZE phase but no commands are run during it since out-of-sync is not relevant.

PREPARE

The PREPARE phase is the most important phase of a commit operation. During the PREPARE phase the NED translates the intended changes in the NSO CDB to commands the device will understand. It then sends those commands to the device, including any commands to navigate the UI such as entering config mode.

For devices without candidate configuration, sending commands causes immediate impact to the running configuration and operation of the network.

COMMIT

During the COMMIT phase, the NED applies the configuration to the device. The COMMIT phase is empty for devices without candidate configuration such as devices managed by the cisco-ios-cli NED. If the device has commit-confirmed capabilities, NSO leverages those during this phase.

```
>> 8-Mar-2025::14:06:54.238 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace-id=c9c7a9

  *** output 8-Mar-2025::14:06:54.239 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace
-- BEGIN COMMIT
-- [08-Mar-2025::14:06:54.239] progress: commit: committing config...
-- Committing (confirmed) [num-commit 0 0a delayed=0]
commit confirmed 30 show-error
  *** input 8-Mar-2025::14:06:54.268 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace-
commit confirmed 30 show-error

Wed Mar 8 14:06:54.354 BRT
RP/0/RP0/CPU0:RNCOBSA0101(config)#
  *** output 8-Mar-2025::14:06:58.377 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace
commit show-error
  *** input 8-Mar-2025::14:06:58.404 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace-
commit show-error

Wed Mar 8 14:06:58.493 BRT
% Confirming commit for trial session.
RP/0/RP0/CPU0:RNCOBSA0101(config)#

  *** output 8-Mar-2025::14:06:58.734 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace
end
  *** input 8-Mar-2025::14:06:58.763 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace-
end

RP/0/RP0/CPU0:RNCOBSA0101#
  *** output 8-Mar-2025::14:06:58.832 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace
-- [08-Mar-2025::14:06:58.832] progress: commit: committing config ok [4593 ms]
-- DONE COMMIT [4594 ms]
<< 8-Mar-2025::14:06:58.832 user: admin/397910 thandle 18935883 hostname ncs device xr0 trace-id=c9c7a9
```

PERSIST

Some devices require additional instructions to ensure the configuration is maintained even when a device restarts. These commands are sent during the PERSIST phase.

PREPARE DRY

The PREPARE DRY phase is unique to `commit dry-run outformat native` operations. Similar to the PREPARE phase it translates the NSO intent to device commands, but does not send those commands to the device.

SHOW_PARTIAL

The SHOW_PARTIAL phase can be invoked by maapi instruction, or it is used during `commit no-overwite` and `rollback during commit error`.

This phase is similar to the SHOW phase in that it collects configuration data from the device and parses it. It collects a more specific set of data relevant to the current commit operation rather than the full configuration. Not all devices support collecting smaller sets of data.

ABORT

The ABORT phase is similar to the PREPARE phase but exclusive for rollback-recovery. NSO sends commands to revert the devices configuration to the way it was before the commit.

REVERT

The REVERT phase is used in situations where a commit encountered an error but NSO can simply tell a device to revert to a previous configuration. In this case, SHOW_PARTIAL and ABORT phases are not required.

COMMAND

The COMMAND phase is unique to live-status operations. During the COMMAND phase NSO passes instructions to the devices outside of the scope of typical commit operations.

IS_ALIVE

The IS_ALIVE phase is a healthcheck to verify whether the sessions between the NED, the device, and NSO are still healthy. If you encounter IS_ALIVE false it is likely you encountered a timeout in one of the sessions.

CLOSE

During the CLOSE phase the NED closes the SSH session to the end-device.

SET_TIMEOUT

The SET_TIMEOUT phase indicates a refresh of various timeouts managed by NSO and the NED.

Other Info in the Trace

Modifications

After a SHOW phase, the NED prints a list of expected changes to be made in the NSO CDB.

```
created    /ios:line/vty[first='5'][last='15']/login
created    /ios:line/vty[first='5'][last='15']/login/local
modified   /ios:interface/Loopback[name='2']
created    /ios:interface/Loopback[name='2']/shutdown
created    /ios:username[name='cisco']
value_set  /ios:username[name='cisco']/privilege 15
```