



## **SolCon User Guide**

### **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





CONTENTS

CHAPTER 1

**SolCon User Guide** 1

- Introduction 1
- Installing the SolCon Converter 1
  - Installing Compiled Binary 1
  - Installing and Running from Source 2
- Troubleshooting and Diagnostics 3
  - WARNINGS and ERRORS 3
  - Log-level DEBUG 3
  - Ensure types match 3
- Limitations 4

CHAPTER 2

**Appendix** 5

- Sources 5





## CHAPTER

# 1

# SolCon User Guide

---

- [Introduction, page 1](#)
- [Installing the SolCon Converter, page 1](#)
- [Troubleshooting and Diagnostics, page 3](#)
- [Limitations, page 4](#)

## Introduction

This document outlines how to use the SolCon tool to convert TOSCA YAML SOL001 files to JSON SOL006 files to be able to load merge into NCS in Rel3 with the SOL6 VNFD model.

It is important to note that the expectation for this tool is that it will do 80-90% of the work, and the last 10% be manually completed, as there are some things the converter cannot handle. See the [Limitations](#) section for more details.

## Installing the SolCon Converter

You can install the converter in any of the following ways:

- 1 Install the compiled binary on your system and run the converter.
- 2 Install and run the converter from the source code.

The following sections provide detailed descriptions. This documentation uses solcon-OSX-0.6 download version to provide examples. Be sure to replace `-OSX-0.6` in the commands listed in this documentation with your appropriate operating system and version for the converter tool.

## Installing Compiled Binary

- 1 Download the most recent version of SolCon for your supported operating system from [here](#).  
If you are not running OSX, Ubuntu, or CentOS see [Installing and Running from Source](#).
- 2 Make sure you are able to run the executable file to confirm installation.  

```
chmod +x solcon-OSX-0.6
```
- 3 Ensure the input and configuration files are present on your machine.
  - Input files are TOSCA specified YAML files, such as `altiostar_vCU.yaml`.
  - Configuration files are in TOML format.
    - TOSCA configuration for ESC VNFDs: [config-esc.toml](#)

- SOL6 output configuration: [config-sol6.toml](#)

**Note:** `config-esc.toml` determines how the YAML VNFD is read and `config-sol6.toml` specifies how the JSON SOL6 VNFD is output.

These are default files and do not need to be modified.

- 4 Run the compiled program from the terminal:

```
./solcon-OSX-0.6 -f altiostar_vCU.yaml -o output_altiostar.json -c
config-esc.toml
```

The argument `-o output_altiostar.json` determines the location and name of the JSON file.

If the program fails to run, download and run the package from the source code. See [Installing and Running from Source](#) for more information.

- 5 Load merge the JSON file into NCS.

Instead of entering NCS, run the following terminal command.

```
ncs_load -lm -F o altiostar_vCU.json
```

- `-lm` stands for load merge
- `-F` is the format
- `o` is the flag for JSON format

If the file to be load merged is not properly formatted, or is missing some required fields, the `ncs_load` program displays an error. See [Troubleshooting and Diagnostics](#) for more information.

## Installing and Running from Source

You can install and run SolCon from the source code if you are unable to install and run from the compiled binary, or if you wish to make changes on your own.

There are two options for downloading and installing the source code, the automated one is recommended.

### Automated Installation

- 1 Download [download-repo.sh](#)
- 2 To download, extract, and setup the required packages, run `sh download-repo.sh`

### Manual Installation

- 1 Download the source code from the repository. See [Sources](#).
- 2 Download and install Python3 based on your operating system.
- 3 Once python3 is installed, run the following command to install the PyYAML and toml packages  
`sh tools/setup-script.sh`

### Running from Source

- 1 Run the SolCon program.  
`python3 solcon.py -f altiostar_vCU.yaml -o output_altiostar.json -c config-esc.toml`
- 2 Load merge the JSON file into NCS.  
Instead of entering NCS, run the following terminal command.  
`ncs_load -lm -F o altiostar_vCU.json`

- -lm stands for load merge
- -F is the format
- o is the flag for JSON format

If the file to be load merged is not properly formatted, or is missing some required fields, the ncs\_load program displays an error. See [Troubleshooting and Diagnostics](#) for more information.

# Troubleshooting and Diagnostics

## WARNINGS and ERRORS

SolCon makes liberal use of WARNING log messages, for example:

```
NFO - Starting Cisco TOSCA -> SOL6 converter.
WARNING - parameters not found in topology_template;node_templates;vnf;interfaces;Vnflcm;inst
WARNING - type not found in topology_template;node_templates;vBBU1_vol;properties;virtual_blo
```

These kinds of warnings are semi-expected, and they do not preclude the converter from generating an output file. Specifically they occur when a field is expected in the TOSCA file, but is not found. Unless it is an extremely important field, the program just prints the warning message and continues converting what it can.

The converter will output an ERROR message when something critical has gone wrong. An output file will not be generated when an error occurs.

## Log-level DEBUG

Run the SolCon converter program with the log-level set to debug

```
./solcon-OSX-0.6 (...) --log-level DEBUG
```

This helps to capture more log information than running the program in the info mode. Use the log information stored in the logs/ folder to troubleshoot and diagnose any problems that you may encounter with the tool.

## Ensure types match

Ensure the types in config-esc.toml match the types in the TOSCA VNFD.

For example, given the following definitions in a TOSCA file:

```
c1:
  type: cisco.nodes.nfv.Vdu.Compute

vnfdl-deployment-control-function-1-cf-boot:
  type: cisco.nodes.nfv.Vdu.VirtualBlockStorage

c1_nic0:
  type: cisco.nodes.nfv.VduCp
```

The config-esc.toml configuration file must look like this:

```
[provider_identifiers.cisco]
vdu = ["type", "cisco.nodes.nfv.Vdu.Compute"]
int_cpd = ["type", "cisco.nodes.nfv.VduCp"]
int_cpd_mgmt = ["type", "cisco.nodes.nfv.VduCp"]
virtual_storage = ["type", "cisco.nodes.nfv.Vdu.VirtualBlockStorage"]
```

There are more type definitions in the default files. If any of the types are not set correctly in the configuration, they will not be found by the converter and thus will be skipped.

## Limitations

- Any data not present in the TOSCA file will not be able to be generated for the SOL6 model.
- Internal connection points and external connection points must have unique names.
- Type extension and default values are not supported at this time.

This is an incomplete list of entries that need to be manually added to the converted file

- NED
- External Networks





## CHAPTER 2

# Appendix

---

- [Sources, page 5](#)

## Sources

If you are installing from the source code, download it from [Here](#).

You can make modifications to the source code (check the license for specifics.)

