ndCONF Agent Development Studio User Guide

for Linux installations

NuDesign Technologies Inc.

2026

Table of Contents

1 Introduction....11
2 Installation Files....13
2.1 <SDK installdir>/ndconf....13
2.2 <SDK installdir>/ndconf/CodeGenerator/ndt-template....13
2.3 <SDK installdir>/ndconf/doc....13
2.4 <SDK installdir>/ndconf/mib2xml....14
2.4.1 <SDK installdir>/ndconf/mib2xml/MIB....14
2.4.2 <SDK installdir>/ndconf/mib2xml/out....14
2.5 <SDK installdir>/ndconf/nccliout....14
2.6 <SDK installdir>/ndconf/ncclient....14
2.6.1 <SDK installdir>/ndconf/ncclient/config....14
2.6.2 <SDK installdir>/ndconf/ncclient/config/edit-config....15
2.6.3 <SDK installdir>/ndconf/ncclient/config/edit-data....15
2.6.4 <SDK installdir>/ndconf/ncclient/config/rpc....16
2.6.5 <SDK installdir>/ndconf/ncclient/config/ssh....16
2.6.6 <SDK installdir>/ndconf/ncclient/config/stream-filter....16
2.6.7 <SDK installdir>/ndconf/ncclient/config/tls....17
2.6.8 <SDK installdir>/ndconf/ncclient/config/yangpush....17
2.7 <SDK installdir>/ndconf/ndcliclient....17
2.8 <SDK installdir>/ndconf/ndconfsvr....17
2.8.1 <SDK installdir>/ndconf/ndconfsvr/config....18
2.8.2 <SDK installdir>/ndconf/schema....18
2.9 <SDK installdir>/ndconf/ndGarageMib....18
2.9.1 <SDK installdir>/ndconf/ndGarageMib/Linux....18
2.10 <SDK installdir>/ndconf/rcvlan....18
2.10.1 <SDK installdir>/ndconf/rcvlan/Linux....19
2.10.2 <SDK installdir>/ndconf/testagent....19
2.10.3 <SDK installdir>/ndconf/testagent/config....19
2.10.4 <SDK installdir>/ndconf/testagent/ssh....19
2.10.5 <SDK installdir>/ndconf/testagent/tls....20
2.11 <SDK installdir>/ndconf/sshd....20
2.12 <SDK installdir>/ndconf/include....20
2.13 <SDK installdir>/ndconf/lib64....20
2.14 <SDK installdir>/ndconf/restconf....20
3 Installation....21
3.1 SDK Requirements....21
3.2 Supported Distributions....21
3.3 Notes....21
3.4 Installing the RESTCONF Access Agent....23
3.4.1 Lighttpd....23
3.4.2 NGINX....26
3.4.3 Apache (httpd)....27
4 Working with the included Source Projects....29
4.1 Building Projects....29
4.2 Configuring testagent for the Produced Shared Objects Libraries....30
4.2.1 ndGarage <option name="co-level-sampling-interval " 30
4.3 Startup Data....31
5 Working with testagent or ndconfsvr....32
5.1 testagent and ndconfsvr Module Compliance....32
5.2 testagent Command Line Options....34
5.3 ndconfsvr Command Line Options....35
5.4 Agent Configuration Parameter File....36

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

5.5 Agent Configuration Options in ndconf.xml....37
5.6 Using testagent....38
5.7 The “Interfaces” DataDlls....39
5.8 The Syslog DataDll....39
6 Working with the CLI....40
6.1 ndGarageMib Example....40
6.2 rcvlan Example....42
6.3 ndcliclient....44
6.3.1 Command Line Options....44
7 Working with ncclient....45
7.1 ncclient Command Line Options....45
7.2 Using ncclient....45
7.2.1 connect....45
7.2.1.1 Connecting using SSH....46
7.2.1.2 Connecting using TLS....47
7.2.2 When a netconf server “calls-home”....47
7.2.3 nc – “netconf connection”....47
7.2.4 ndGarageMib Example....49
7.2.5 rcvlan based example....50
7.2.6 Notifications in ncclient....52
7.3 nccliout....55
8 Working with SNMP....56
8.1 Configuring ndtaasnmp.so.0....56
8.1.1 <option name="postCommitInterval" 56
8.1.2 <option name="postCommitMaxInterval" 56
8.1.3 <option name="postCommitSaveToStartup" ….56
8.2 Default SNMP Users....57
8.2.1 SNMP v1/v2c....57
8.2.2 SNMP v3....57
8.3 Working with MIBrowser....57
8.3.1 Notifications in MIBrowser....58
8.4 Putting it All together....60
9 Configuring DataServer Options....63
9.1 <option name="nacmtrace" ….63
9.2 Configuring to Retrieve User groups Using an External Application....63
9.2.1 ndtrdsclient....63
9.2.2 RADIUS configuration requirements....64
9.3 <option name="grpAuthExe" ….64
9.4 <option name="authAddr" ….64
9.5 <option name="authPswd" ….65
10 Configuring NETCONF....66
10.1 <option name="numthreads" ….66
10.2 <option name="enableSSH" ….66
10.3 <option name="enableTLS" ….66
10.4 Configuring TLS certificates....66
10.5 <option name="call-home-reconnect-interval" ….67
10.6 <option name="max-incoming-connections" ….67
10.7 <option name="call-home-sshd-debug" ….68
10.8 <option name="call-home-sshd-exe" ….68
10.9 <option name="call-home-sshd-cfg" ….68
10.10 <option name="call-home-sshd-ndncsub" ….68
11 Working with RESTCONF....69
11.1 Configuring libndtaarestconf.so.0....69
11.1.1 <option name="socket" 69

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

11.1.2 <option name="HttpsOnly" 70
11.1.3 <option name="ClientCertRequired" 70
11.1.4 <option name="AuthRequired" 70
11.1.5 <option name="PassWebUsertoNACM" 71
11.1.6 <option name="UseClientCertCN" 71
11.1.7 <option name="DisableFlushToStartup" 71
11.1.8 <option name="FlushtoStartupPeriod" 71
11.1.9 <option name="UseCandidateAndCommit" 72
11.1.10 <option name="NotifSocket" 72
11.1.11 <option name="CalcEtagForNonConfig" 72
11.1.12 <option name="AllowPartialReadKeys" 72
11.2 Using ‘restconf-submit-test.html’....73
11.2.1 The ‘Input’ Section....73
11.2.2 The ‘Results’ Section....75
11.2.3 Examples of Using NETCONF RPCs....76
11.2.4 RESTCONF Notifications....78
11.3 Using curl....80
11.4 Current Implementation Status & Limitations....81
12 CLI....82
12.1 Command completion....82
12.2 Commands and parameters....82
12.2.1 ?....83
12.2.2 about....83
12.2.3 accessagent....83
12.2.4 cls....84
12.2.5 cmpcfg....84
12.2.5.1 configure....84
12.2.6 datadll....84
12.2.7 exit....85
12.2.8 help....85
12.2.9 quit....86
12.2.10 script....86
12.2.11 session....86
12.2.12 show....86
12.2.12.1 show agents....86
12.2.12.2 show data....87
12.2.12.3 show datadll....87
12.2.12.4 show history....87
12.2.12.5 show namespaces....87
12.2.12.6 show remaining....88
12.2.12.7 show running....88
12.2.13 sleep....88
12.3 Configuration Mode....89
12.3.1 commit....90
12.3.2 create....90
12.3.3 delete....90
12.3.4 diff....91
12.3.5 edit....91
12.3.6 exit....92
12.3.7 load....92
12.3.8 quit....92
12.3.9 run....93
12.3.10 save....93
12.3.11 set....93

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

12.3.12 show....94
12.3.13 top....95
12.3.14 up....95
12.4 Configuring libndtaacli.so.0....95
13 ncclient - NETCONF client....96
13.1 ncclient Commands and parameters....97
13.1.1 ?....97
13.1.2 about....97
13.1.3 access-agent....97
13.1.4 cls....98
13.1.5 connect....98
13.1.6 cancel-commit....100
13.1.7 close-session....101
13.1.8 commit....101
13.1.9 copy-config....101
13.1.10 create-subscription....102
13.1.11 datadll....103
13.1.12 delete-config....103
13.1.13 delete-subscription....104
13.1.14 discard-changes....104
13.1.15 disconnect....104
13.1.16 edit-config....104
13.1.17 edit-data....105
13.1.18 establish-subscription....106
13.1.19 exit....108
13.1.20 get....108
13.1.21 get-access-agents....109
13.1.22 get-config....109
13.1.23 get-data....109
13.1.24 get-data-dlls....110
13.1.25 get-namespaces....111
13.1.26 get-schema....111
13.1.27 help....111
13.1.28 kill-session....111
13.1.29 kill-subscription....112
13.1.30 list....112
13.1.31 lock....112
13.1.32 modify-subscription....113
13.1.33 nc....114
13.1.34 now....115
13.1.35 partial-lock....115
13.1.36 partial-unlock....115
13.1.37 quit....115
13.1.38 redirectqueue....116
13.1.39 rpc....116
13.1.40 show....117
13.1.41 sleep....117
13.1.42 unlock....117
13.1.43 validate....117
14 NACM - Access Control....118
14.1 Incoming RPC Message Validation....119
14.2 Outgoing <notification> Authorization....119
14.3 Data Node Access Validation....121
14.4 Debugging Access Control....123

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

15 NETCONF call-home....124
15.1 The truststore and keystore....124
15.1.1 truststore....124
15.1.2 keystore....125
15.2 Server....126
15.2.1 TLS endpoint....126
15.2.2 SSH endpoint....129
15.3 client....131
15.3.1 TLS endpoint....131
15.3.2 SSH endpoint....133
15.4 A call-home example....134
15.5 Keys and Certificates....136
15.5.1 TLS....136
15.5.1.1 Client Identity Mapping....138
15.5.2 SSH....139
SSH keys....140
15.5.2.1 Configuring sshd and ncclient to use the include (NDT) SSH keys....141
15.5.2.2 Call-home over SSH using keys....143
15.5.3 SSH Certificates....145
15.5.3.1 Configuring sshd and ncclient to use the "ndt" SSH certificates....147
15.5.3.2 Call-home over SSH using certificates....148
16 Datastores....150
17 Subscribed Notifications....152
17.1 Filters....153
17.2 Dynamic Subscriptions....153
17.3 Configured Subscriptions....154
17.4 Subscription monitoring....156
18 Yang push....158
18.1 Filters....158
18.2 System capabilities....158
18.3 Dynamic Subscriptions....159
18.3.1 Periodic subscriptions....159
18.3.2 On change subscriptions....160
18.4 Configured Subscriptions....162
19 SNMP....165
20 Code Generation....166
20.1 Code Generator....166
20.2 ndt3.pyc Plugin....166
20.2.1 Plugin Command Line Options....166
20.3 Generating Code with the plugin....167
20.4 ndtGen....167
20.4.1 ndtGen Synopsis....167
20.4.2 ndtGen Command Line Options....167
20.4.3 Generating code with ndtGen....168
20.5 The Generated Project....168
20.6 Building the Project....168
20.7 Testing the Project....169
20.8 Building with the Source SDK....169
20.9 Configuring testagent to use the Project’s shared object output....169
20.10 Enabling SNMP access to a Project's shared object output....170
20.11 Other Resources....171
20.11.1 Project Design Notes....171
20.11.2 Class Reference....171
20.11.3 man pages....171

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

20.11.4 HTML formatted “man” pages....171
20.11.5 Default TCP Ports....172
21 Products for ndCONF Development....173
21.1 ndCONF Builder....173
21.2 ndCONF SDK....173
21.3 NETCONF, RESTCONF, CLI and SNMP Access Agents....173
21.4 Visual MIBuilder....173
22 About NuDesign....174
23 Appendix....175
23.1 'ND-GARAGE-MIB' Yang module....175
23.2 'ND-GARAGE-MIB' MIB module....179
23.3 ‘RAPID-CITY-MIB’ (rcvlan) Yang Module....184
23.4 ‘RAPID-CITY-MIB’ (rcvlan) MIB Module....189
23.5 Capabilities....202

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

DOCUMENT HISTORY

Date	Revision	Remarks
2017-12-10	.9	Imported from v1.12 Yang DEMO Quick Start document.
2018-01-03	1.0	Update and add SDK specific content.
2018-01-15	1.1	Add appendix sections for RAPID-CITY YANG and MIB modules. Miscellaneous edits
2018-01-23	1.2	Add ‘Other Resources’ and ‘Startup Data’ sections. Other miscellaneous updates.
2018-01-31	1.3	- change references to /usr/local/NuDesign/ to the SDK install directory. - change references to ‘NcClient’ and ‘YANGDemoAgent’ - add sub sections for **CLI** ndGarageMib and rcvlan examples. - add sub sections for **ncclient** ndGarageMib and rcvlan examples. - text regarding the derivation of **rcvlan.yang** and the related project.
2018-02-01	1.4	- minor edits
2018-02-09	1.5	- documentation changes relating to renaming and redefining to ndtCONF - revert lib64 and include references to fixed install location.
2018-02-14	1.6	- change footer. - add documentation about included design reference. - add documentation about included design notes for the two datadll projects.
2018-02-23	1.7	Add section to describe NuDesign Products for customizing ndCONF. - rename references of ngsettings.xml to ndconf.xml - rename reference of ngsettings_cli .xml to ndconf_cli.xml - miscellaneous updates
2018-02-28	1.8	- change image of directory structure. - correct section 2 sub headings. - minor revisions.
2018-08-01	1.9	- added - confirmed commit - writable running - partial locking
2018-08-20	2.0	- content updates
2018-09-06	2.1	- update partial-lock, partial-unlock
2018-09-18	2.2	- correct section numbering
2019-02-01	2.3	- change paths for updated layout of the SDK - add CodeGen section for non-eval SDK - restructure some of the headings. - edits and corrections.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

		- update copyright notices
2019-02-01	2.4	- change demo-data-cfg.xml to project specific name - several minor edits.
2019-03-15	2.5	- Add sections and content relating to **ndconfsrv** and **ndcliclient**.
2019-08-07	2.6	- update for conversion to libxml2 implementation. - add openconfig-interfaces implementation.
2019-08-19	2.7	- add “namespace” documentation.
2019-08-23	2.8	- document configuration differences to change to alternate interfaces datadlls.
2019-09-06	2.9	- correct “get path” example.
2019-10-24	3.0	- fix errors, minor editorial changes
2019-10-30	3.1	- add RESRCONF updates
2019-12-27	3.2	- updated lighttpd configuration info. - add nginx configuration info. - add apache configuration info. - add Using RESTCONF section. - miscellaneous minor corrections.
2020-01-06	3.3	- miscellaneous minor edits.
2020-01-23	3.4	- add section describing option to change the CLI background service port. - add namespace to partial-lock documentation. - remove load <file> “running” from documentation.
2020-02-01	3.5	- updated commit & cancel-commit
2020-02-10	3.6	- change pyang information. - add section about building generated projects with the source SDK.
2020-03-02	3.7	- remove ‘script’ from internal ncclient cli. - minor CLI changes - additional notes relating to ncclient wait interval. - remove reference to two separate libietfInterface data dlls. - add documentation on use of redirection in the CLI.
2020-04-15	3.8	- update **testagent** and **ndconfsvr** command line options. - add section describing the agent configuration parameter file.
2020-04-22	3.9	- update CLI with new “show” commands: “data”, “namespaces” and “remaining”. - add information about specifying multiple keys in “edit”. - more details on “cmpcfg” command.
2020-05-01	3.10	- add a section on testing the generated code.
2020-07-08	3.11	- update RESTCONF limitations. - add curl examples for POST, PUT and DELETE. - name change ndtaanetconf to ndtaanc.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

		- add section on Configuring Netconf access agent.
2020-07-28	3.12	- add documentation on RESTCONF option **flushtoStartupPeriod**.
2020-09-08	3.13	- update RESTCONF limitations section.
2020-09-14	3.14	- cli: accessagent - ncclient - get-access-agents, get-data-dlls, access-agent, datadll
2020-10-14	3.15	- add notification stream documentation. - update restconf-submit-test.html documentation. - add to and update RESTCONF options section.
2021-01-12	3.16	- add new SNMP AA options. - update restconf-submit-test.html section. - RESTCONF implementation/limitations section updated. - ndtGen section updated.
2021-01-19	3.17	- update ndtGen documentation.
2021-02-11	3.18	- add section for server configuration items in ndconf.xml.
2021-05-20	3.19	- add description for apache CGI timeout variable CGIDScriptTimeout - add description for lighttpd WebApp timeout variables server.max-write-idle and server.max-read-idle. - add document for the SYSLOG datadll. - update RESTCONF stream documentation. - update restconf-submit-test.html documentation.
2021-11-25	4.0	- update RESTCONF web server requirement. - update Apache/httpd documentation. - update restconf-submit-test.html documentation. - add RADIUS integration documentation. - add document entry for main-man.html - add run time requirements for nettle and gnutls libraries. - update “confirmed commit” example. - multiple updates for tls. - update **ncclient** with tls/ssh choice - update capabilities
2022-12-21	4.1	- add client certificate updates. - add CalcEtagForNonConfig RESTCONF option. - add AllowPartialReadKeys RESTCONF option. - update RESTCONF compliance statement. - update capabilities list - minor edits
2023-04-19	5.0	- updates for NETCONF call-home.
2023-04-21	5.1	- update requests and supported distributions
2023-08-03	5.2	- remove references to test.html. - numerous minor updates. - add ‘now’, ‘nc’, ‘nccliout’ documentation.
2025-04-02	5.3	- add documentation for “co-level-sampling-interval”

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

		- update documentation relating to nettle library requirements. - update configuration parameters elating to RADIUS support. - update relating to using SSH with keys and call-home. - update copyright notices
2025-04-07	5.3a	- further update to using SSH with keys relating to the included files known_hosts and authorized_keys
2025-08-16	5.4	- NMDA - subscribed-notifications - yang-push
2025-08-21	5.4a	- correct for some missing section 2 entries. - minor editorial changes. - add -o option doc to testagent and ndconfsvr.
2026-02-26	5.4b	- update lighttpd and httpd configuration sections.
2026-04-20	5.5	- add section describing default SNMP user information.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

1 Introduction

This document describes the NuDesign ndCONF Agent Development Studio product, a complete management agent development environment containing an extensible **YANG Datastore Server**, **NETCONF / YANG Datastore SDK** with include and library files, API documentation for the YANG Datastore object providers and for the NETCONF, SNMP, RESTCONF and CLI protocol **Access Agents**. It also includes source code sample projects for demonstrating dynamically loadable Datastore data extensions, created using our generator tools.

The product's basic b agent is for operational and configuration aspects of the system. Multiple, dynamically loadable data extensions are supported by the **testagent** and **ndconfsvr**[1]. The evaluation version is available for download from the NuDesign site at www.ndt-inc.com, as a Linux executable, installation package.

The evaluation version includes: a time limited **testagent** and **ndconfsvr**[2], an implementation for the standard YANG-module ietf-interfaces, an implementation for the standard YANG-module openconfig-interfaces plus two NDT code generated C++ source code projects[3]. The first one is for a data model defined from the "ND-GARAGE-MIB" YANG module and the other one is defined in the "**rcvlan**" YANG module. Both YANG documents were created from MIB modules, using NuDesign's Visual MIBuilder. These are the ND-GARAGE-MIB and (a) simplified RAPID-CITY-MIB (rcvlan) SNMP MIB Modules. In the case of the **rcvlan** example, the default MIB derived code has been modified to demonstrate how to transform the code into a more CLI centric data model. All modules are listed in the document Appendix.

Access to the ndCONF Agent is provided via an "Access Agent" component (AA). An 'AA' is a Dynamic-Link Library loaded by the **testagent** at startup. Each "handles" a specific protocol (i.e. NETCONF/SNMP/CLI/RESTCONF...). Protocol operations that manipulate data (read/write) are handled commonly by the data store in the **testagent**. In other words, all access to data is common, irrespective of the protocol used to access it. The evaluation version includes three additional Access Agents (beside basic CLI AA), one each for NETCONF, RESTCONF and SNMP.

The **testagent** is a console application; the CLI AA provides command line handling for it.

Note: before you start the **testagent**, you need to stop any SNMP agent you may have running on your computer, as there may be a UDP port conflict with the testagent, to provide SNMP access. Typically **SNMP** operates on **UDP port 161**, which usually requires root privilege to open, so testagent should also be run "as root".

ncclient is a simple, multi-connection NETCONF client console application and call-home end-point; it may be used to access the **testagent** using the NETCONF protocol over **SSH** or **TLS**. NuDesign's Visual MIBrowser Pro is the SNMP client used within this document to present SNMP operations, though any SNMP manager could be used. You can download the evaluation version of Visual MIBrowser Pro at NuDesign web site at https://www.ndt-inc.com/ndt/eval-download/

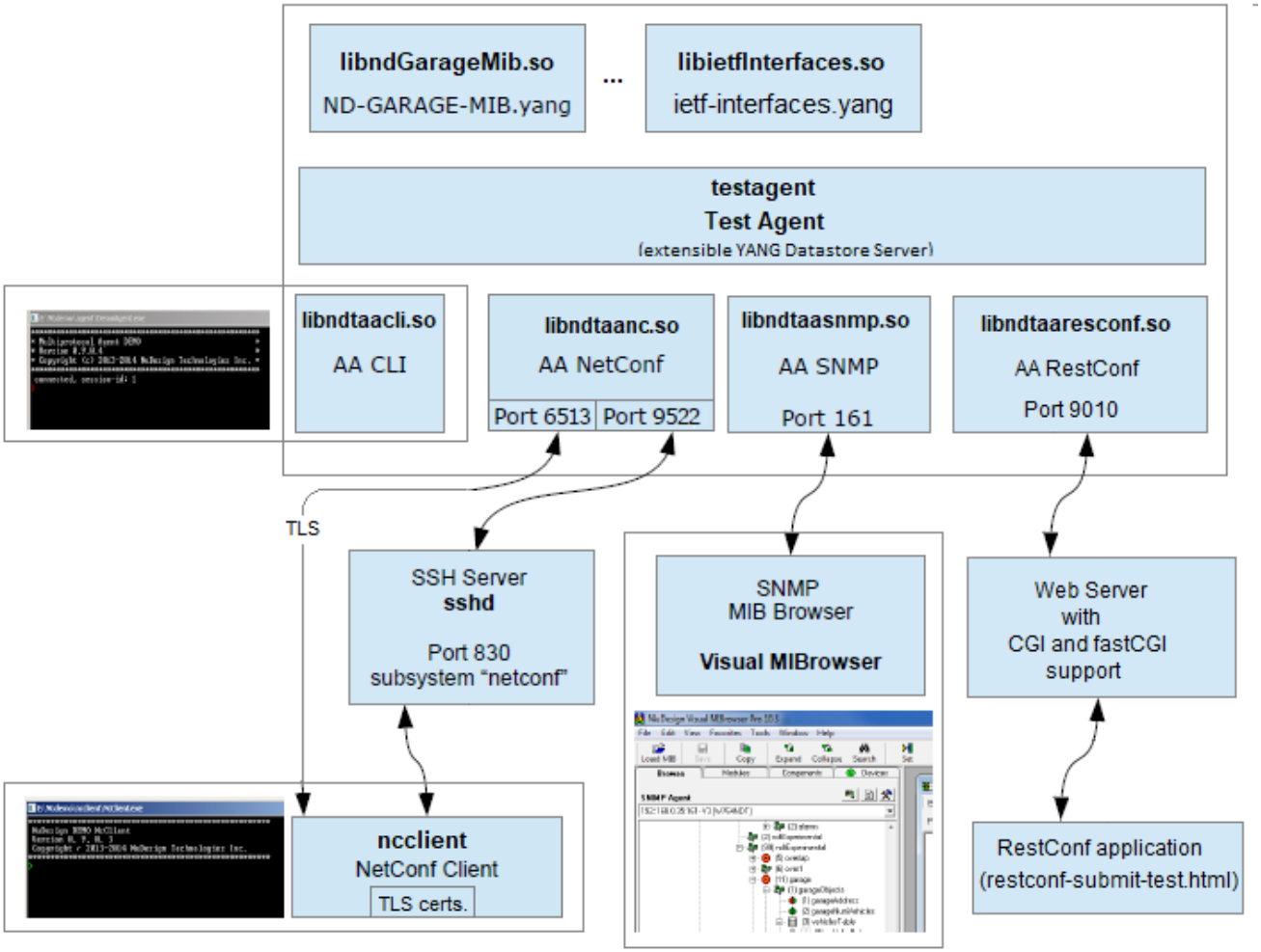
Also Note: in the diagram that follows, two data models are depicted as integrated with the testagent. These are provided for by '**libndGarageMib.so**' and '**libietfInterfaces.so.0**'. Several others are provided. One, **rcvlan**, is discussed later in the document. The next, l**ibietdYangLibrary**, provides the ietf-yang-library module implementation. The third is a group of data dlls of an alterative implement of interfaces, openconfig-interfaces. The last is **libsyslogMsgMib.so** which is an implementation of SYSLOG-MSG-MIB (RFC5676) that provides access to syslog notifications.

Lastly, the **evaluation** version of the product expires **30 days** after installation. It also does not include any of the code generation facility available in the licensed version.

---

1 **`ndconfsvr`** is a version of the **`testagent`** that behaves the same as **`testagent`**, but can be placed into the background. When in the background, **`ndcliclient`** is used to access the **`CLI`**. The documentation generally pertains to both, except for this particular aspect.
2 **`testagent`** and **`ndconfsvr`** are time limited only on the evaluation version of the product.
3 See ndCONF Builder

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

2 Installation Files

By default, the installer will place 'ndconf' under the directory '**/opt/ndconfBuilder-x.xx-buildyyy**' (and '**/opt/ndconfEval-x.xx-buildyyy**' for the evaluation version). **x.xx** is the version number being installed and **YYY** is the build designation. As a convention, that location is identified below by '**<SDK installdir>**'[4].

2.1 <SDK installdir>/ndconf

`uninstall`	Uninstall executable
`uninstall.dat`	Data used by the uninstaller
`install.log`	A log of operations performed by the post installation script.

2.2 <SDK installdir>/ndconf/CodeGenerator/ndt-template

The directories below this point contain platform specific code generation template files.

`linux`	Linux specific template files.

2.3 <SDK installdir>/ndconf/doc

`Errors.htm`	HTML page documenting system error interpretation.
`Main-pdf.html`	Simple HTML front end to class library PDFs.
`Main-man.html`	Simple HTML front end to man html pages.
`ndCONF Builder – CodeGen Tech Reference.pdf`	Design notes for datadlls, based strictly on generated code and specifically the **`ndGarageMIB`** project.
`ndCONF Builder – UsingExistingImpl Tech Reference.pdf`	Design notes for datadlls where there is pre-existing implementation code and specifically the **rcvlan** project.
`NuDesign ndCONF Agent Development Studio Evaluation Package Quick Start.pdf`	Quick start guide,
`ndtbase.pdf`	PDF document providing design documentation for the **ndtbase** classes
`ndtcli.pdf`	PDF document providing design documentation for the **ndtcli** classes
`ndtmibh.pdf`	PDF document providing design documentation for the **ndtmibh** classes
`ndtsnmp.pdf`	PDF document providing design documentation for the **ndtsnmp** classes
`ndtsocket.pdf`	PDF document providing design documentation for the **ndtsocket** classes
`ndtsys.pdf`	PDF document providing design documentation for the **ndtsys** classes

4 The **CodeGenerator** directory and files below exist only in the release version of the SDK.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

`ndtcert.pdf`	PDF document providing design documentation for the **ndtcert** classes
`ndtxml.pdf`	PDF document providing design documentation for the **ndtxml** classes
`NuDesign ndCONF User Guide.PDF`	This document.
`NuDesign ndCONF Studio Eval License Agreement.pdf`	Evaluation licensing information. This file exists only when the installation is for evaluations.
`NuDesign ndCONF Studio License Reference.pdf`	Licensing information. This file exists only when the installation is a release version..
`ReadMe – ndCONF SDK.txt`	Text file containing SDK specific release information.
`...`	Other component specific text files containing component specific release information, plus html files of the installed man pages.

2.4 <SDK installdir>/ndconf/mib2xml

`mib2xml`	Executable to convert a MIB into a MIB info document
`mib2xml.cfg`	Configuration file for the mib2xml executable.

2.4.1 <SDK installdir>/ndconf/mib2xml/MIB

`*.mib`	MIB files

2.4.2 <SDK installdir>/ndconf/mib2xml/out

`*.xml`	Output XML files

2.5 <SDK installdir>/ndconf/nccliout

`nccliout`	(optional) ncclient output client

2.6 <SDK installdir>/ndconf/ncclient

`ncclient`	NETCONF client (command line)
`ncclientstart`	Script to start ncclient with typical parameters

2.6.1 <SDK installdir>/ndconf/ncclient/config

`NcClient.xml`	Message configuration file for **ncclient**
`namespaces.xml`	An XML document of namespaces
`ndconf-err.xml`	error message configuration file
`ndconf.xml`	**ncclient** configuration file
`startup-cfg.xml`	**ncclient** start up configuration file
`cfgparams.xml`	operational configuration file for **ncclient**.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

2.6.2 <SDK installdir>/ndconf/ncclient/config/edit-config

Repository of files that contain portions of xml encodings for an **edit-config** rpc.

For example

```
1:T admin@127.0.0.1> edit-config candidate c1
```

would generate the edit-config request for target=candidate, and the rest of the xml would be taken from the file "c1.xml" that resides in ...\config\edit-config folder. Content of the files in the edit-config folder could contain <default-operation>, <test-option>, <error-option> and|or <config> elements for valid edit-config request.

This is an example of the edit-config generated:

```
<rpc message-id="3" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <target>
      <candidate />
    </target>
    <default-operation>merge</default-operation>
    <test-option>set</test-option>
    <config>
      <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
        <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create" >
          <t:vehicleIndex>4</t:vehicleIndex>
          <t:vehicleLicencePlate>777 SWE</t:vehicleLicencePlate>
          <t:vehicleModel>Koenigsegg One:1</t:vehicleModel>
        </t:vehiclesEntry>
      </t:garage>
    </config>
  </edit-config>
</rpc>
```

Note that the red portion is copied directly from the file "c1.xml".

2.6.3 <SDK installdir>/ndconf/ncclient/config/edit-data

Contains a file that is a portion of an xml encoding for an **edit-config** rpc that is used by the **edits-data** command.

For example

```
1:T admin@127.0.0.1> edit-data candidate c1
```

would generate the edit-data request for target=candidate, and the rest of the xml would be taken from the file "d1.xml" that resides in ...\config\edit-data folder. Content of the files in edit-data folder could contain <default-operation>.

This is an example of the edit-config generated:

```
<rpc message-id="3" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <target>
      <candidate />
    </target>
    <default-operation>merge</default-operation>
    <config>
      <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
        <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create" >
          <t:vehicleIndex>4</t:vehicleIndex>
          <t:vehicleLicencePlate>777 SWE</t:vehicleLicencePlate>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
          <t:vehicleModel>Koenigsegg One:1</t:vehicleModel>
        </t:vehiclesEntry>
      </t:garage>
    </config>
  </edit-config>
</rpc>
```

Note that the red portion is copied directly from the file "c1.xml".

2.6.4 <SDK installdir>/ndconf/ncclient/config/rpc

`ncclient` implements the 'rpc' command. The directory contains a variety of rpc xmls that demonstrate various rpc operations. E.g.

```
1:T admin@127.0.0.1> rpc f1
```

This command would search for file "f1.xml" in "rpc" folder, and if found, read it and pass the content of this file to **`testagent`** for processing. An example of the content:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <garage></garage>
    </filter>
  </get>
</rpc>
```

2.6.5 <SDK installdir>/ndconf/ncclient/config/ssh

Repository of SSH related authentication files.

2.6.6 <SDK installdir>/ndconf/ncclient/config/stream-filter

This folder contains files with the ‘stream-filter’ portion of an **establish-subscription** or **modify-subscription** RPC which is used by the "establish-subscription stream" or “modify-subscription stream” commands to generate a compete RPC.

E.g. if the file “f1.xml” contains:

```
<stream-xpath-filter>
  //ndg:coLevelFallingAlarm
</stream-xpath-filter>
```

then the command

```
1:T admin@127.0.0.1> establish-subscription stream NETCONF filter f2
```

will generate the following RPC:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="35">
  <n2:establish-subscription>
    <n2:stream>NETCONF</n2:stream>
    <n2:stream-xpath-filter>//ndg:coLevelFallingAlarm</n2:stream-xpath-filter>
  </n2:establish-subscription>
</rpc>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

2.6.7 <SDK installdir>/ndconf/ncclient/config/tls

A hierarchical repository of TLS related authentication files.

2.6.8 <SDK installdir>/ndconf/ncclient/config/yangpush

This folder contains files with ‘datastore-filter’ and ‘trigger’ portions of an RPC which are used by "establish-subscription datastore” or “modify-subscription datastore” commands to generate a complete RPC.

If the file “filt2.xml” contains:

```
<datastore-xpath-filter>
  /ndg:garage/ndg:garageObjects
</datastore-xpath-filter>
```

and trig2.xml:

```
<on-change xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <dampening-period>2000</dampening-period>
    <sync-on-start>true</sync-on-start>
    <excluded-change>create</excluded-change>
</on-change>
```

then the command

```
1:T admin@127.0.0.1> establish-subscription datastore operational filter filt2 trigger trig2
```

will generate the following RPC:

```
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
      xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="37">
    <n2:establish-subscription xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
      <yp:datastore>operational</yp:datastore>
      <yp:datastore-xpath-filter>/ndg:garage/ndg:garageObjects</yp:datastore-xpath-filter>
      <yp:on-change xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <yp:dampening-period>2000</yp:dampening-period>
        <yp:sync-on-start>true</yp:sync-on-start>
        <yp:excluded-change>create</yp:excluded-change>
      </yp:on-change>
    </n2:establish-subscription>
  </rpc>
```

2.7 <SDK installdir>/ndconf/ndcliclient

`ndcliclient`	A remote cli client for ndconfsrv.[5]

2.8 <SDK installdir>/ndconf/ndconfsvr

`ndconfsvr`	The NDT YANG Datastore Agent Server[6][7]
`ndconfsvrstart`	A script to simplify starting **ndconfsvr**.

5 The directory and file only exists on release versions of the SDK.
6 The directory and file only exists on release versions of the SDK.
7 A valid license file must be present in /etc to use the release version of this application.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

2.8.1 <SDK installdir>/ndconf/ndconfsvr/config

`ssh`	repository of SSH related authentication files
`tls`	repository of TLS related authentication files
`cfgparams.xml`	operational configuration file for **ndconfsvr**.
`ndconf.xml`	configuration file for **ndconfsvr**.
`ndconf_cli.xml`	configuration file for CLI Access agent
`startup-cfg.xml`	startup configuration data for the **ndconfsvr**.
`factory-cfg.xml`	factory configuration data for the **ndconfsvr**.
`m2y_*.xml`	MIB-to-YANG mapping files
`mib_*.xml`	MIB info files (generated by mib2xml tool)
`ndconf-err.xml`	error message configuration file
`yp-capabilities.xml`	configuration of ietf-system-capabilities and ietf-notification-capabilities

2.8.2 <SDK installdir>/ndconf/schema

Repository of YANG modules on which the included projects are dependent. The **get-schema** rpc method also looks here for YANG files.

2.9 <SDK installdir>/ndconf/ndGarageMib

`ndGarageMib-startup-cfg-demo.xml`	Sample initialization data. This data may be copied to the **testagent**'s startup-cfg.xml file to auto initialize running data for this project. See the section 'Startup Data' below.
`m2y_ndGarageMib.xml`	MIB-to-YANG mapping file.
`mib_ND_GARAGE-MIB.xml`	MIB info file (generated by mib2xml tool)
`NdGarageMib*.cpp`	Project c++ source files.
`NdGarageMib*.h`	Project source include files.
`NDGarageV2.mib`	Mib file on which the project is based.

2.9.1 <SDK installdir>/ndconf/ndGarageMib/Linux

`makefile`	Makefile for project
`StdAfx.h`	Platform specific project include file.

2.10 <SDK installdir>/ndconf/rcvlan

`rcvlan-startup-cfg-demo.xml`	Sample initialization data. This data may be copied to the **testagent**'s startup-cfg.xml file to auto initialize running data for this project. See the section 'Startup Data' below.
`m2y_rcvlan.xml`	MIB-to-YANG mapping file.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

`mib_RCVLAN.xml`	MIB info file (generated by mib2xml tool)
`rcvlan*.cpp`	Project c++ source files.
`rcvlan*.h`	Project source include files.
`rcvlan.mib`	Mib file on which the project is based.
`rcvlan.yang`	YANG file (derived from RAPID-CITY.YANG) on which the project is based.

2.10.1 <SDK installdir>/ndconf/rcvlan/Linux

`makefile`	Makefile for project
`StdAfx.h`	Platform specific project include file.

2.10.2 <SDK installdir>/ndconf/testagent[8]

`testagent`	test agent
`Main.cpp`	Source file
`AppInfo.cpp`	Source file
`teststart`	Script to start **testagent** with the default parameters.

2.10.3 <SDK installdir>/ndconf/testagent/config

`cfgparams.xml`	operational configuration file for **testagent**
`ndconf.xml`	configuration file for test agent.
`ndconf_cli.xml`	configuration file for CLI Access agent
`ndconf-err.xml`	XML document of errors.
`startup-cfg.xml`	startup configuration data for the **`testagent`**.
`factory-cfg.xml`	Factory configuration data for the **`testagent`**.
`m2y_IF-MIB.xml.bak`[9]	MIB-to-YANG mapping file for the ietf-interfaces datadll.
`m2y_openconfigInterfaces.xml`	MIB-to-YANG mapping file for the openconfig-interfaces datadll[10].
`m2y_rcvlan.xml`	MIB-to-YANG mapping file for the rcvlan datadll.
`m2y_ND-GARAGE-MIB.xml`	MIB-to-YANG mapping file for the ND-GARAGE-MIB datadll.
`mib_*.xml`	MIB info files (generated by mib2xml tool)
`yp-capabilities.xml`	configuration of ietf-system-capabilities and ietf-notification-capabilities

2.10.4 <SDK installdir>/ndconf/testagent/ssh

Repository of SSH authentication files.

8 This documentation applies to the ndconfsvr directory hierarchy as well.
9 When using the ietfInterfaces configuration, you need to rename this file to remove the “,.bak” and add “.bak” to the “m2y_openconfigInterfaces.xml”. This will allow the ietfInterfaces.so data dll to respond to SNMP requests.
10 Only one of the m2y modules for openconfig-interfaces and ietf-interfaces should exist in this directory at one time, otherwise overlap in SNMP definitions will occur, leading to unexpected results. By default, the openconfig-interfaces configuration is installed, hence m2y_IF_MIB.xml has the extension .bak to prevent it from being loaded by the server. When changing to the ietf-interfaces, remove the .bak extension from this file and add one to m2y_openconfInterfaces.xml.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

2.10.5 <SDK installdir>/ndconf/testagent/tls

Repository of TLS authentication files.

2.11 <SDK installdir>/ndconf/sshd

`ndncsub`	Pipe process executable.

2.12 <SDK installdir>/ndconf/include

This directory contains a hierarchy of include files for building the included projects.

Note: The **makefile** for each project expects the includes be in this directory. If the includes are moved for any reason, then the **makefile** for each project will need to be **modified**.

2.13 <SDK installdir>/ndconf/lib64

This directory contains 64 bit shared object library files for building and executing the included programs.

Note: All SDK executables are built expecting the libraries to be in this directory. This installation sets up this library load path on most distributions, however sometimes other installers may disrupt the load path later or you may wish to move the libraries for some reason, then you may need to restore the load path to these libraries. This can be accomplished a number of ways on Linux, Including the following:

- copy, move or soft link them to one of the conventional library repositories, such as **/usr/lib** or **/usr/lib64**.
- Use **ldconfig**. i.e. `sudo /sbin/ldconfig <the installation directory>/lib64`
- export **LD_LIBRARY_PATH** with a value that includes a path to the new location of the **lib64** directory. This may be done from the command line, or in the user's **.bashrc**.

2.14 <SDK installdir>/ndconf/restconf

This directory contains an example html file for accessing RESTCONF.

`restconf-submit-test.html`	Example RESTCONF application.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

3 Installation

3.1 SDK Requirements

This SDK has a number of requirements for use. These are:

- a GNU C/C++ development environment, version 5.3 or greater, to build the projects.
- Open SSH server
- OpenSSL 1.1.x/3.x
- libssh2
- libxml2 + development includes files
- A FastCGI compliant web server (for RESTCONF) that passes the HTTP_AUTHORIZATION environment variable[11].
- Nettle and gnutls libraries[12].

These should all be available from your Linux distribution.

3.2 Supported Distributions

This Linux version of YANG Datastore Generator SDK has been verified to build the projects and operate on a variety of recent generation 64 bit Linux distributions, including:

- Fedora 33 - Current
- Debian 9.3 – Current
- Ubuntu 19.04 – Current
- OpenSUSE Leap 15.2 – Current
- RHEL 8.0/CentOS8/Oracle Linux 8/Rocky 8.0 - Current

3.3 Notes

Note 1: Some of these notes may be Fedora specific.

Note 2: most of the following steps require root level privilege.

Note 3: the server needs to be run as root (or sudo) to access privileged ports: (SNMP uses UDP port 161 by default, which is privileged on most Linux distributions.)

11 This may require a configuration modification to achieve this requirement. See the Apache/httpd documentation.
12 These are only required if using the provided ndtrdsclient for radius authentication. Specifically libnettle.so.8 is required. If you platform does not support this version, you will either have to use a distribution that offers it or manually install it.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

1. **chmod +x** the install **.run** file you downloaded to make sure it is still executable after downloading. You will need to have root privileges to execute the **.run** file to install the SDK. Select a default or your preferred location to install the SDK.
2. Make sure **sshd** is installed on your distribution. How to do this varies with the distribution. Once you have, you need to stop it to modify the configuration. You can use the following (as root):

```
# systemctl stop sshd
```

3. Add entries for '**ndncsub**' to **/etc/ssh/sshd_config** directly below appropriate lines (commented or not)

```
Port 830
Subsystem netconf <SDK install directory>/ndconf/sshd/ndncsub
```

4. Ensure sshd starts at startup (it may be disabled by default, depending on distribution)

   Note: systemctl may not be available on all distributions, if not you can likely skip this step)

```
# systemctl enable sshd
```

5. Enable **sshd** to use port 830 (for NETCONF). This is probably not necessary on many distributions, but is required on SELinux distributions, such as Fedora & RedHat.

```
# semanage port -a -t ssh_port_t -p tcp 830
```

6. Make sure sshd is restarted with the new configuration items. How depends on the distribution.

```
# systemctl restart sshd
```

   or

```
# service sshd restart
```

   or

```
# /etc/init.d/ssh restart
```

   If all else fails, kill and start '**sshd -D**' manually.

   Do a '**netstat -a -t**' to ensure port 830 is open.

7. If the firewall is running, open firewall ports for remote access (may not be necessary on all distributions)
   - Install "Firewall configuration tool" (optional)
   - Open "Firewall configuration tool" (optional)
   - add port to firewall exceptions (make it "permanent", if you want it to power up with these)

SNMP	udp	161-162
NETCONF	ssh	tcp 830

   Alternatively, for testing, you can disable the firewall. This can be done (on Fedora) with

```
# systemctl stop firewalld
```

   To have it not start again on startup, you can do:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
# systemctl disable firewalld
```

Now you should be ready to run the **testagent** and the NETCONF Client, ncclient.

8. To run the **testagent** using **teststart**. (See the **man** pages for and teststart.) Change the directory to **<SDK installdir>/ndconf/testagent**, then type

```
# ./teststart
```

This starts testagent with the default parameters to run it from this directory.

9. run the demo **NETCONF** client. (See the man pages for **ncclient** noting in particular the library requirements.) Change the directory to **<SDK installdir>/ndconf/ncclient**, then type:

```
# ./ncclientstart
```

3.4 Installing the RESTCONF Access Agent

The **RESTCONF** implementation in **libndtaarestconf.so** uses a **fastCGI** interface to any Web server that supports it. The first step then is to identify a compliant webserver. This document provides configuration details for using three common Linux distributed webservers that support fastcgi: **lighttpd**, **NGINX** and **Apache**.

Provided in the SDK is a javascript based test HTML file, “restconf-submit-test.html”[13], to provide access to RESTCONF operations.

Also provided in the SDK is the CGI executable ‘**ndrcstream**’. This needs to be copied under the web server’s file space. See below for more information.

A second step may be required, which is to install libfcgi.so.0 on your system. The file sometimes is installed when you install an fcgi compliant webserver.on your system, sometimes not. The library will likely be in one of the system library directories if it is installed. Below is a list of package names and a command line sequence that should install libfcgi.so.0 on a number of common distributions, if it isn’t.

- Fedora, fcgi, sudo dnf install fcgi
- Debian, libfcgi-bin, sudo apt-get install libfcgi-bin
- Ubuntu, libfcgi-bin, sudo apt-get install libfcgi-bin
- openSUSE, FastCGI sudo zypper install FastCGI

3.4.1 Lighttpd

The **lighttpd** configuration file is usually located under **/etc/lighttpd** and is named **lighttpd.conf**.

To enable fastCGI there needs to be a directive that looks like:

```
server.modules += ( "mod_fastcgi" )
```

This line needs to exist to enable fastcgi processing. In addition to this module, three other modules need to be loaded. These are:

13 “restconf-submit-test.html” may also be used from the file system directly.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
server.modules += ( "mod_alias" )
server.modules += ( "mod_setenv" )
server.modules += ( "mod_cgi" )
```

There may be a section in your configuration file already pertaining to **fastCGI**. (There may also be a separate file for fastcgi configuration, if that's the case, then there's a line something like:

```
include "conf.d/fastcgi.conf"
```

If you find a line like this, go to that specific file. The rest of the documentation that follows should be the same.

Now look for a line that looks like

```
fastcgi.server = (
```

above that add the following:

```
alias.url += ( "/restconf/streams/" => server_root + "/cgi-bin/ndrcstream/" )

#to enable flushing as output is sent to/from CGI process
server.stream-response-body=1

$HTTP["url"] =~ "/restconf/streams/" {
    setenv.set-environment = ( "RCSPORT" => "9524" ) # specify the service port for ndrdstream
                                                     # to use.
    cgi.assign = ( "" => "" )                        # tell server there's no file extension
                                                     # to ndrcstream cgi executable.
}

$HTTP["url"] !~ "^/restconfsubmit-test.html" {       # exclude the requesting page name
    $HTTP["url"] !~ "^/restconf/streams/" {
```

To this, add the section below…

```
            "/.well-known/" =>
            (( "host" => "127.0.0.1",
                 "port" => 9010,
                 "check-local" => "disable",
             )),
             "/restconf" =>
             (( "host" => "127.0.0.1",
                 "port" => 9010,
                 "check-local" => "disable",
             ))
            )
        } #note this closes the section
    }
```

The entire section without any other fastcgi clients should look as follows:

```
$HTTP["url"] !~ "^/restconfsubmit-test.html" {    # exclude the requesting page name
  $HTTP["url"] !~ "^/restconf/streams/" {

        "/.well-known/" =>
            (( "host" => "127.0.0.1",
                "port" => 9010,
                "check-local" => "disable",
        )),
        "/restconf/" =>
            (( "host" => "127.0.0.1",
                "port" => 9010,
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
                "check-local" => "disable",
            ))
        )
        } #note this closes the section
}
```

This causes lighttpd to pass all URLs starting with "**/.well-known/**" or "**/restconf/**" to the RESTCONF server at 127.0.0.1:9010, for processing. The last line (**check-local**) suppresses lighttpd from checking that the URL exists before handing a request off to the fastCGI processor in the master agent.

It also causes any URL starting with "**/restconf/stream/**" to be passed the CGI process "ndrcstream" to manage one oe more streams.

Typically, the root directory for web pages on **lighttpd** is '**/var/www/lighttpd**'. Copy the html file from the installation's restconf directory to this directory.

You also need to copy the CGI executable "**ndrcstream**" to the the CGI bin directory. In the example above this is given by the 'alias.url=' specification. This would usually be '**/var/www/cgi-bin**'.

Depending on the frequency of events being sent, **lighttpd** may time out '**ndrcstream**'. Normally the server will keep the client alive by sending "idle" (empty) messages to the client to avoid timeouts. However it is possible to control the timeout interrval that **lighttpd** uses with the 'server.max-write-idle' and 'server.max-read-idle' variables. E.g.

```
server.max-write-idle = 3600
server.max-read-idle = 3600
```

would set the timeout to 1 hour.

When SSL has been enabled, to have the client send its certificate, in the SSL configuration, ensure that you have set the following option:

```
ssl.verifyclient.activate = "enable"
```

This will have **lighttpd** request that the client send a certificate. You should also have **ssl.ca-file** set to the PEM file containing the authority certificate used by the client (and server).

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

3.4.2 NGINX

The **NGINX**[14] configuration file is usually located under **/etc/nginx** and is named **nginx.conf**.

NGINX typically enables fastCGI, so all that needs to be done is provide configuration components to have NGINX redirect requests to the RESTCONF module. To do this, within the "**server**" block in **nginx.conf**, define the following:

```
# /restconf/streams configuration
   location /restconf/streams/ {
          fastcgi_param NO_BUFFERING 1;
          include fastcgi_params;
          fastcgi_param SCRIPT_FILENAME /var/www/cgi-bin/ndrcstream;
          fastcgi_param RCSPORT 9524;
          fastcgi_pass 127.0.0.1:9030;
          fastcgi_buffering off;
          auth_basic "Password Required";
          auth_basic_user_file /etc/nginx/.htpasswd;
    }

location /restconf/ {
    fastcgi_pass   127.0.0.1:9010;
    include    fastcgi_params;
}
location /.well-known/ {
    fastcgi_pass   127.0.0.1:9010;
    include    fastcgi_params;
}
```

This causes NGINX to pass all URLs starting with "**/.well-known/**" or "**/restconf/**" to the restconf server at 127.0.0.1:9010 for processing.

Typically, the root directory for web pages using NGINX is '**/usr/share/nginx/html**'. Copy the html file from the installation's restconf directory to this directory.

The above example assumes that the CGI executable '**ndrcstream**' was copied to the '/var/www/cgi-bin' directory. If you're using a different directory for, alter the above as necessary.

If you have enabled SSL and you wish to enable processing client certificates, then you also need to add the three lines similar to the following, to the "server" block:

```
ssl_trusted_certificate httpscert/ca.pem;
ssl_client_certificate httpscert/client.pem;
ssl_verify_client optional;
```

Note: the path and file names will be specific to your installation. (See NGINX documentation for more information.)

Additionally you'll need to add the following two lines to **both** the '/restconf' and '/restconf/streams' location blocks:

```
fastcgi_param SSL_CLIENT_VERIFY ${ssl_client_verify};
fastcgi_param SSL_CLIENT_S_DN ${ssl_client_s_dn};
```

14 NGINX doesn't support CGI directly, so it is not as well suited for managing streams with this implementation. However, you can use a third party application such as 'fcgiwrap' to add this capability. The NGINX documentation examples assumes that it is in use and also that it supports the "NO_BUFFERING" options. Some distributions include this option. If yours doesn't, more information about how to add it is available on the internet.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

3.4.3 Apache (httpd[15])

The distribution may (usually) or may not include the fastcgi modules, **mod_proxy** and **mod_proxy_fcgi**, when **apache** is installed. If you cannot find them under (typically) **/etc/httpd/modules**, then you'll first need to install these modules. In addition, you will also need to ensure the alias module **mod_alias** and the CGI module **mod_cgi** have been loaded.

The **httpd** configuration file is usually located under **/etc/httpd/conf** and is named **httpd.conf**.

In the <IfModule alias_module> section, assuming your using the standard CGI bin directory, add these two lines:

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
ScriptAlias /restconf/streams/ "/var/www/cgi-bin/ndrcstream/"
```

Add a 'Directory' section as follows:

```
<Directory "/var/www/cgi-bin">
   AllowOverride None
   ### only needed to support symbolic links in the cgi-bin directory ####
   Options SymLinksIfOwnerMatch
   Require all granted
</Directory>
```

Toward the end of the configuration module, add the following:

```
ProxyPass "/restconf-submit-test.html" !
ProxyPass "/restconf/streams/" !
ProxyPass "/restconf/" "fcgi://127.0.0.1:9010/"
ProxyPass "/.well-known/" "fcgi://127.0.0.1:9010/"
```

Typically, the root directory for web pages using httpd is '/var/www/html'. Copy the html file from the installation's restconf directory to this directory.

The above example assumes that the CGI executable 'ndrcstream' was copied to the '/var/www/cgi-bin' directory. If you're using a different directory for cgi executables, alter the above as necessary.

Depending on the frequency of events being sent, apache may time out 'ndrcstream'. Normally the server will keep the client alive by sending "idle" (empty) messages to the client to avoid timeouts. However it is possible to control the timeout interval that Apache uses with the 'CGIDScriptTimeout' variable.

E.g.

```
IfModule cgid_module>
    IDScriptTimeout 1800s
/IfModule>
```

Would set the timeout interval to 1800 seconds (half an hour).

In addition to this, on SELinux systems[16] you have to enable **httpd** to have access other network processes. This can be done with the following command:

```
# setsebool httpd_can_network_connect 1 -P
```

When '**authrequired**' is set to '**true**', you usually must add to the configuration the line

```
SetEnvIF "(.*)" HTTP_AUTHORIZATION=$1
```

to enable Apache to forward the **HTTP_AUTHORIZATION** environment variable to **RESTCONF**. Alternatively

15 The apache web server 'httpd' may be referred to as 'apache2' on some distributions. File names and directories may be slightly different than discussed here. Consult distribution specific documentation for details on these systems.
16 Red Hat, Fedora and SUSE are examples of systems that install **SELinux** by default.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

you may also accomplish this by setting **CGIPassAuth** appropriately.

E.g.

```
CGIPassAuth “On”
```

When **SSL** has been enabled, to have the client send its certificate, in the SSL configuration, ensure that you have set the following option:

```
SSLVerifyClient optional
```

The variable can also be set to “required”. This will have **httpd** request that the client send a certificate.

You should also have **SSLCACertificateFile** set to the **CRT** file containing the authority certificate used by the client (and server).

Lastly, you need to add “**+StdEnvVars**” to the **SSLOptions** variable. E.g.

```
SSLOptions +StdEnvVars
```

You might consider setting this option only within a “Location” block specific to “/restconf/”, “/.well-known/” and as well “/var/www/cgi-bin”.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

4 Working with the included Source Projects

This install contains three C++ projects. The first two projects were generated with NuDesign's ndCONF Builder product[17]. The latter is a simple console, data server application, implementing NETCONF, RESTCONF, SNMP and CLI from the resources provided by this installation.

ndGarageMib	A YANG data provider library project, implementing the **ND-GARAGE-MIB.yang** module, which is a direct MIB-to-YANG conversion from the **ND-GARAGE-MIB.mib** module.
rcvlan	A YANG data provider library project, implementing **rcvlan.yang.** **rcvlan.yang** is a subset derived from the **RAPID-CITY.yang** module, which is a direct MIB-to-YANG conversion of a subset of the **RAPID-CITY** enterprise MIB. This subset MIB contains just two related MIB tables. This project demonstrates the consolidation of those two MIB tables into nested YANG lists.
testagent	The source project for the **testagent**.

Each is located in the SDK directory of the same name.

4.1 Building Projects

To build these projects, change directory to the **Linux** sub-directory in each project and typing make.

There are a number of targets and build options for these projects. Targets are provided on the make command line. Options may be provided via the environment or on the make command line. These are:

clean This target indicates that the project should be "cleaned". This deletes all object files and libraries or the executable, depending on the project.

MODE This variable selects whether the build is a 'debug' or 'release' build. The default value is 'debug'. To create a release version of the project you would set the value to 'release'. E.g.

```
# make clean
# make MODE=release
```

The first line "cleans" the project and the second builds a release version of the project.

NUDESIGNDIR This variable specifies where the SDK include and library files are located. The default is <SDK installdir>/ndconf . If the files under this directory have been move to another directory, then this variable will need to specified in the makefile.

17 For the purposes of your evaluation, you may contact us about generating a project for you, from your YANG document.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

4.2 Configuring testagent for the Produced Shared Objects Libraries

Two of the source projects produce a “data dll”. These are the

- **`ndGarageMib`**
- **`rcvlan`**

projects. Both of these produce a “shared object” run-time loadable library. On Linux, the name of the library file will be ‘**lib**’ + project name + “**.so**” and will be located in the **Linux** sub-directory of the project. E.g. libndGarageMib.so.

To have the **testagent** use them, you need to make a configuration change to the testagent. This change is made in the **ndconf.xml**. This file is located in the testagent “config” directory. By default this directory is “**./testagent/config**”, though this can be modified by the testagent command line options.

Once you find the ndconf.xml, open the file in a text editor. Locate the **<datadlls>** section of the file. Within this section are individual **<datadll></datadll>** entries. In the default file, there will be an existing entry in this section for the ‘**garage**’ (interfaces) datadll, which looks as follows:

```
<datadll name="if" dll="libndGarage.so.0" ></datadll>
```

There are two or three fields in a <datadll> </datadll> specification. These provide specific configuration information about a particular datadll. These fields are as follows.

`name`	This is the name that is given to a **datadll**. This is used in load, unload & status operations.
`dll`	This is the actual file name of the **datadll**. E.g. **libietfInterfaces.so.0**.
`path`	(optional). This is the path to a **datadll**. When not specified, the path is assumed to be **<SDK `installdir>/ndconf/lib64`**. If the **datadll’s** path is not this, then you need to specify it.

E.g.

```
<datadll name="garage" dll="libndGarageMib.so" path="../ndGarageMib/Linux" ></datadll>
```

4.2.1 ndGarage <option name="co-level-sampling-interval " ...

There is a parameter associated with then **ndGarageMib** shared object that controls the sampling rate of the simulated “co-level” evaulation. The default is 10 seconds, which may lead to a notification being sent as frequently as every two seconds. This rate can be modified as follows:

E.g.

```
<datadll name="garage" dll="libndGarageMib.so" path="../ndGarageMib/Linux" >
    <options>
        <option name="co-level-sampling-interval" value="30"></option>
    </options>
</datadll>
```

where “30” sets the new sample rate every 30 seconds.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

4.3 Startup Data

The file, **startup-cfg.xml**, may be used to provide initial configuration data to the **testagent**, on startup. Each source project includes a file named '**<project name>-startup-cfg-demo.xml**' that contains some sample initialization data.

Once the **testagent** has been configured to load either or both of the **datadll** projects, you can use the contents of these files to provide initial data for that datadll.

E.g. Adding initialization data for **ndGarageMib** to **startup-cfg.xml**.

From the default **startup-cfg.xml**, at the bottom is the close tag **</config>**. Depicted as follows.

```
        </nacm>
    </config>
```

Insert the entire content from ndGarageMib's **ndGarageMib-startup-cfg-demo.xml**, between the tags (shown above) , as depicted below:

```
    </nacm>
    <garage xmlns="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
        <garageObjects>
            <garageAddress>10 street</garageAddress>
            <garageCOLevelRisingThreshold>50</garageCOLevelRisingThreshold>
            <garageCOLevelFallingThreshold>10</garageCOLevelFallingThreshold>
        </garageObjects>
        <vehiclesEntry>
            <vehicleIndex>1</vehicleIndex>
            <vehicleLicencePlate>123 ABC</vehicleLicencePlate>
            <vehicleModel>Maserati Quattroporte</vehicleModel>
        </vehiclesEntry>
        <vehiclesEntry>
            <vehicleIndex>2</vehicleIndex>
            <vehicleLicencePlate>345 XYZ</vehicleLicencePlate>
            <vehicleModel>Jaguar F-Type</vehicleModel>
        </vehiclesEntry>
    </garage>
</config>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

5 Working with testagent or ndconfsvr

In the following sections, you may use **testagent** or **ndconfsvr** interchangeably, except as noted.

5.1 testagent and ndconfsvr Module Compliance

`testagent` and **`ndconfsvr`** implements the standard YANG modules:

- ietf-inet-types
- ietf-yang-types
- ietf-yang-smiv2
- SNMPv2-TC
- ietf-netconf-notifications
- ietf-netconf
- ietf-netconf-monitoring
- ietf-netconf-acm
- iana-crypt-hash
- ietf-x509-cert-to-name
- notifications
- nc-notifications
- ietf-crypto-types
- ietf-keystore
- ietf-truststore
- ietf-tcp-common
- ietf-tcp-client
- ietf-tcp-server
- ietf-tls-common
- ietf-tls-server
- ietf-ssh-common
- ietf-ssh-server
- ietf-netconf-server
- ietf-writable-running
- ietf-confirmed-commit
- ietf-partial-lock
- ietf-netconf-server.yang

ietfInterfaces data dll implements yang 1.0 module:

- ietf-interfaces (2014-05-08)

ietfYangLibrary data dll implements yang 1.0 module:

- ietf-yang-library (2016-06-21)

ietfInterfaces11 data dll implements yang 1.1 module:

- ietf-interfaces (2018-02-20)

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

ietfYangLibrary11 data dll implements yang 1.1 module:

- ietf-yang-library (2019-01-04)

openconfig-Interfaces data dll implements module:

- oc-interfaces

ndtaasnmp implements YANG modules:

- SNMPv2-MIB
- SNMP-FRAMEWORK-MIB
- ietf-snmp

  with submodules:

  ietf-snmp-common
  ietf-snmp-engine
  ietf-snmp-target
  ietf-snmp-notification
  ietf-snmp-proxy
  ietf-snmp-community
  ietf-snmp-usm
  ietf-snmp-vacm

ndtaarestconf implements YANG modules:

- nc-notifications
- ietf-restconf
- ietf-restconf-monitoring
- ietf-yang-patch
- nc-notifications

ietfSubscribedNotifications implements YANG modules:

- ietf-subscribed-notifications
- ndt-nc-subnotifs

ietfYangPush implements YANG modules:

- iietf-yang-push
- ietf-system-capabilities
- ietf-notification-capabilities

syslogMsgMib implements YANG modules:

- SYSLOG-TC-MIB
- SYSLOG-MSG-MIB

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

5.2 `testagent` Command Line Options

testagent *[-f cfgFile][-c configDir][-s schemaDir][-v vrootDir][-r startupDir] [-b bakDir][-l logDir] [-w] [-a] [-o]*

cfgFile	Path to file with agent configuration, default is 'none'.
configDir	Folder with agent configuration files, default is config sub-folder of the folder where vrootDir is set to.
schemaDir	Folder with YANG modules. Agent searches this folder while processing "get-schema" rpc. The default is the schema sub-folder to where vrootDir is set to.
vrootDir	This is the 'virtual' root for the agent. Any write operations are only allowed under this point. By default, this is the current directory from which the executable was run.
startupDir	This is the directory in which the startup configuration is. By default this is the same directory as '**configDir**'.
bakDir	Folder where backup files are stored. The default is the 'bak' sub-folder to where vrootDir is set to.
logDir	Folder where log files are stored. The default is the 'log' sub-folder to where vrootDir is set to.
-w	Create a new log file on each agent start, this is default. File name format is *testagent-xxx.log* where *xxx* is encoded date/time in format *YYMMDDhhmmss*. E.g. *testagent-151206191025.log*
-a	a single log file named *testagent.log* is created, log output is appended
-o	Operate the server in "original" mode. When in this mode, the server uses the "ndconf-original.xml" configuration file. The default mode of operation is in **NMDA** mode, which uses the "ndconf.xml" configuration file.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

5.3 ndconfsvr Command Line Options

ndconfsvr *[-f cfgFile][-c configDir] [-s schemaDir][-v vrootDir][-r startupDir][-b bakDir] [-l logDir] [-w] [-a] [-d] [-t] [-o]*

cfgFile	Path to file with agent configuration, default is 'none'.
configDir	Folder with agent configuration files, default is 'config' sub-folder of the folder where vrootDir is set to.
schemaDir	Folder with YANG modules. Agent searches this folder while processing "get-schema" rpc. The default is the 'schema' sub-folder to where vrootDir is set to.
vrootDir	This is the 'virtual' root for the agent. Any write operations are only allowed under this point. By default, this is the current directory from which the executable was run.
startupDir	This is the directory in which the start up configuration is. By default this is the same directory as 'configDir'.
bakDir	Folder where backup files are stored. The default is the 'bak' sub-folder to where vrootDir is set to.
logDir	Folder where log files are stored. The default is the 'log' sub-folder to where vrootDir is set to.
-w	create a new log file on each agent start.[18] File name format is testagent-xxx.log where xxx is encoded date/time in format YYMMDDhhmmss. E.g. testagent-151206191025.log
-a	a single log file named testagent.log is created, log output is appended
-d	Be default, ndconfsvr starts as a console application. To start as a daemon, use this option. To access the cli, once ndconfsvr has been started in the background, use ndcliclient to access the cli.
-t	When ndconfsvr is started as a background process with the -d option, using this option to stop it.
-o	Operate the server in "original" mode. When in this mode, the server uses the "ndconf-original.xml" configuration file. The default mode of operation is in **NMDA** mode, which uses the "ndconf.xml" configuration file.

18 If neither -w nor -a is specified, no logging is produced.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

5.4 Agent Configuration Parameter File

The following sections describes the format of the agent configuration parameter file. This is the file indicated by the '-f' command line options for **testagent** and **ndconfsvr**. This file offers additional operational parameters than does the command line. As an example, below is the contents of the cfgparams.xml file provided in the SDK's 'config' directory.

```
<?xml version="1.0"?>
<ndt>
  <cfgparams>
    <param name="dirConfig" value="/opt/ndConfBuilder-3.0.0-build002/ndconf/testagent/config" />
    <param name="dirSchema" value="/opt/ndConfBuilder-3.0.0-build002/ndconf/schema" />
    <param name="dirVRoot" value="/opt/ndConfBuilder-3.0.0-build002/ndconf/testagent" />
    <param name="dirStartupCfg" value="config" />
    <param name="dirLog" value="log" />
    <param name="dirBak" value="bak" />
    <param name="logFileAccessMode" value="w" />
    <param name="logLevel" value="1" />
    <param name="maxLogSize" value="100" />
    <param name="maxLogFiles" value="-1" />
    <param name="maxBackups" value="-1" />
    <param name="fileNdConfCfg" value="ndconf.xml" />
    <param name="fileNdConfErr" value="ndconf-err.xml" />
    <param name="fileStartupCfg" value="startup-cfg.xml" />
    <param name="fileFactoryCfg" value="factory-cfg.xml" />
  </cfgparams>
</ndt>
```

The file's parameters are defined as follows. Note: each of these elements is optional. If not specified, the agent's defaults relating to the parameter will apply as indicated in the previous two sections.

dirConfig	Folder with agent configuration files, default is **config** sub-folder of the folder where **vrootDir** is set to.
dirSchema	Folder with YANG modules. Agent searches this folder while processing "get-schema" rpc. The default is the **schema** sub-folder to where **vrootDir** is set to.
dirVRoot	This is the 'virtual' root for the agent. Any write operations are only allowed under this point. By default, this is the current directory from which the executable was run.
dirStartupCfg	This is the directory in which the start up configuration is. By default this is the same directory as '**configDir**'.
dirBak	Folder where backup files are stored. The default is the '**bak**' sub-folder to where **dirVRoot** is set to.
dirLog	Folder where backup files are stored. The default is the '**log**' sub-folder to where **dirVRoot** is set to.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

logFileAccessMode	One of ‘a’, ‘w’ or ‘n’. ‘a’ and ‘w’ are the logging ‘append’ and ‘over**w**rite’. ‘n’ indicates no logging.
logLevel	Controls the content levels in logs, (minimum)1<=logLevel<=5(maximum). Default:1.
maxLogSize	Specifies the maximum log file size (KB) to allow. If not specified, there is no file size constraint. If one is specified, then when the file grows past the constraint, a new file is created, subject to the limitation of ‘**maxLogFiles**’. Files names are serialized.
maxLogFiles	This specified the maximum number of log files that the server will maintain. As that number of log files grows to the limit, then the oldest files are deleted. A value of ‘-1’ indicates no constraint.
maxBackups	This specified the maximum number of backup files that the server will maintain. As that number of files grows to the limit, then the oldest backup files are deleted. A value of ‘-1’ indicates no constraint.
fileNdConfCfg	The filename of the configuration file used to specify access agents, data dlls and options pertaining to these items.
fileNdConfErr	The filename of the error file.
fileStartupCfg	The filename of the start up configuration file.
fileFactoryCfg	The filename of the ‘factory’ configuration file.

5.5 Agent Configuration Options in ndconf.xml

These options are managed from the “**ndconf.xml**” file. The location of this file is specified in the **cfgparams.cfg** file by the **dirConfig** option’s value.

Theses option are found under the **<ndt>** key in the **<options>** section.. E.g.

```
<?xml version="1.0"?>
   <ndt>
      <options>
         <option name="dlldir" value="/opt/ndConfBuilder-3.0.0-build002/ndconf/lib64" />
         <option name="maxevents" value="100" />
         <option name="loadConfig" value="relaxed" />
      </options>
  :
  :
```

dlldir	The directory from which to load datadlls.
maxevents	The number of events that the server will maintain internally for replay. The default is **500**.
loadConfig	When set to “relaxed”, the server will continue even if configuration data if found for which there is **no** information known. This may happen when a previously load datadll, which defines this information, is no longer loaded. By default the server will stop when this occurs. Caution should be exercised with this option as the server the use of this option as that information will be lost if a “copy-config” running to startup is invoked..

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

5.6 Using `testagent`

To start **`testagent`**, type **`testagent`** and any appropriate parameters, in a terminal window. E.g.

```
> ./testagent -c ./config -s ./schema -l . -a
```

(See the previous pages for more information about command line parameters.)

On startup **`testagent`** executes the following steps:

- read settings (from./config/ndconf.xml), here is an excerpt (openconfig-interfaces configuration depicted):

```
<agents>
  <agent name="cli" dll="libndtaacli.so.0"></agent>
  <agent name="nc" dll="libndtaanc.so.0" ></agent>
  <agent name="snmp" dll="libndtaasnmp.so.0"></agent>
  <agent name="rc" dll="libndtaarestconf.so.0"></agent>
</agents>
<datadlls>
  <datadll name="yanglib" dll="libietfYangLibrary.so.0" ></datadll>
  <datadll name="if" path="<install_dir>/ndconf/lib64/ietf-interfaces" dll="libietfInterfaces.so"
></datadll>
  <datadll name="ocExtension" dll="libopenconfigExtensions.so" ></datadll>
  <datadll name="ocYangTypes" dll="libopenconfigYangTypes.so" ></datadll>
  <datadll name="ocTypes" dll="libopenconfigTypes.so" ></datadll>
  <datadll name="ocInterfaces" dll="libopenconfigInterfaces.so" ></datadll>
  <datadll name="slm" dll="libsyslogMsgMib.so" ></datadll>

  <datadll name="garage" dll="libndGarageMib.so" ></datadll>
  <datadll name="rcvlan" dll="librcvlan.so" ></datadll>
</datadlls>
```

- load Access Agents (listed in section <agents>)

- load data dlls (listed in section <datadlls>, I.e., initialize and register modules implemented in each dll

- load **startup-cfg.xml** into RUNNING configuration data store

- copy RUNNING to CANDIDATE

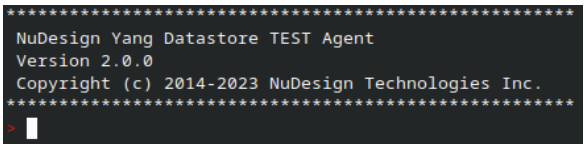
- initialize Access agents (at this point running configuration contains parameters required by AA to be initialized properly)

- refresh state data (get initial state of objects that are not part of configuration, e.g. counters)

- record start time

- start Access agents

- if the **`libndtaacli.so`** is successfully loaded, control is passed to it and it displays:



otherwise it displays:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

In other words, other Access agents may be running without **`libndtaacli.so`**, until ‘q’ is pressed.

`testagent` creates verbose log files. The name of the log file is "<executable-name>.log", e.g. "`testagent.log`". By default, log files are created in subdirectory "`./log`".

Lastly, output for commands that output to the console may be redirected with the ‘>’ character and specifying a filename. By default the file is located in the directory local to the server.

E.g.

```
> configure[Enter]
Candidate configuration is now locked.
[edit ]
# show running > running.txt
[edit ]
#
```

redirects the output of “show running” to the file “running.txt”

5.7 The “Interfaces” DataDlls.

The SDK comes with two different “Interfaces” implementations. Both provide “live” operational network interface data. One is based on the **openconfig-interfaces.yang** document and implemented in **libopenconfigInterfaces.so.0**. The other is based on the **ietf-interfaces.yang** document and implemented in **libietfInterfaces.so.0**.

Only one of these “datadlls” may handle interacting with the SNMP access agent at a time. To control which, there are two different mapping files. By default, the mapping file “m2y_IF-MIB.xml”, in the **config** directory is in place and thus **libietfInterfaces.so.0** handles SNMP interactions.

To change the configuration to have **libopenconfigInterfaces.so.0** instrument SNMP interactions, rename the following files in the **config** directory where the executable is. First rename “m2y_IF-MIB.xml” to “m2y_IF-MIB.xml.bak” then rename “m2y_openconfigInterfaces.xml.bak” to “m2y_openconfigInterfaces.xml” and restart the server.

Having the ‘.bak’ extension prevents the mapping file from being loaded at startup.

5.8 The Syslog DataDll

The SDK comes with an implementation of the SYSLOG-MSG-MIB (RFC5676). When this datadll is part of the configuration of the server, then you can access the syslog “stream” through either NETCONF or RESTCONF.

Note: On some distributions of Linux, if you’re not running the dataserver as root, then the datadll may only have access to non-privileged syslog entries. i.e. the same as the current user/group of the dataserver process.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

6 Working with the CLI

As noted, **ndconfsvr** starts as a console application, in which case, the following directly applies. When it is started as a background process, you may access the **CLI** from the **ndcliclient** application. Once you do, the following **CLI** interactions still apply.

The CLI emulates “Junos” and is started in operational mode, indicated by the "> " prompt. To get a list of available Commands type "?" (question mark).

```
> ?
Possible completions:
 about
 cls
 cmpcfg
 configure
 datadll
 exit
 help
 quit
 script
 session
 show
 sleep
>
```

Execute the "configure" command to enter configuration mode, indicated by the “[edit]” line above the prompt, which changes to “# “

```
> configure[Enter]
Candidate configuration is now locked.
[edit ]
#
```

In configuration mode user can modify data in the "candidate" datastore (which is locked by the CLI during execution of the "configure" command). Once the "candidate" is configured, to apply the changes to the "running" configuration, the user must execute the "commit" command.

6.1 ndGarageMib Example

[19]Here is an example of how to create a row in the vehiclesTable:

```
[edit ]
# show garage[Enter]                          <--- display current config
// candidate //
    garageObjects
      garageAddress 10 Street
      garageCOLevelRisingThreshold 50
      garageCOLevelFallingThreshold 10
    vehiclesEntry
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
    vehiclesEntry
```

19 This example assumes the you have followed the instruction in section 4.3 relating to startup data,

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
      vehicleIndex 2
      vehicleLicencePlate 345 XYZ
      vehicleModel Jaguar F-Type
[edit ]
# create garage vehiclesEntry 3[Enter]       <--- create new entry
  ok
[edit ]
# show garage[Enter]                         <--- display current config
// candidate //
    garageObjects
      garageAddress 10 Street
      garageCOLevelRisingThreshold 50
      garageCOLevelFallingThreshold 10
    vehiclesEntry
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
    vehiclesEntry
      vehicleIndex 2
      vehicleLicencePlate 345 XYZ
      vehicleModel Jaguar F-Type
    vehiclesEntry                           <--- new row created, note that vehicleLicencePlate
      vehicleIndex 3                        <--- and vehicleModel is not created yet
[edit ]
# create garage vehiclesEntry 3 vehicleLicencePlate "555 QQQ"[Enter] <--- create vehicleLicencePlate
  ok
[edit ]
# create garage vehiclesEntry 3 vehicleModel "Audi TT"[Enter]        <--- create vehicleModel
  ok
[edit ]
# show garage[Enter]                         <--- display current config
// candidate //
    garageObjects
      garageAddress 10 Street
      garageCOLevelRisingThreshold 50
      garageCOLevelFallingThreshold 10
    vehiclesEntry
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
    vehiclesEntry
      vehicleIndex 2
      vehicleLicencePlate 345 XYZ
      vehicleModel Jaguar F-Type
    vehiclesEntry
      vehicleIndex 3
      vehicleLicencePlate 555 QQQ
      vehicleModel Audi TT
[edit ]
# commit[Enter]                              <--- commit
  ok
[edit ]
# show running garage[Enter]                 <--- show running configuration
// running //
  garage
    garageObjects
      garageAddress 10 Street
      garageCOLevelRisingThreshold 50
      garageCOLevelFallingThreshold 10
    vehiclesEntry
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
    vehiclesEntry
      vehicleIndex 2
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
      vehicleLicencePlate 345 XYZ
      vehicleModel Jaguar F-Type
    vehiclesEntry
      vehicleIndex 3
      vehicleLicencePlate 555 QQQ
      vehicleModel Audi TT
[edit ]
#
```

6.2 rcvlan Example

[20]Here is the example how to create vlan:

```
[edit ]
# show vlanModule[Enter]                    <--- display current config
// candidate //
    vlan
      vlanId 1
      vlanName vlan#1
      stgId 0
      vlanType 1
      vlanPortMembers 43:04:00:40
      vlanProtocolId 1
      ip
        ifIndex 1
        addr 192.168.0.11
        netMask 255.255.0.0
        bcastAddrFormat 0
        reasmMaxSize 1234
        macOffset 0
        vrfId 15
[edit ]
# edit vlanModule[Enter]                    <--- navigate to vlanModule branch
  ok
[edit vlanModule]
# create vlan 2 vlanName vlan#2[Enter]      <--- create vlan, id# = 2, with vlanName
  ok
[edit vlanModule]
# commit[Enter]                             <--- commit
  ok
[edit vlanModule]
# show vlan 2
// candidate //
  vlanId 2
  vlanName vlan #2
[edit vlanModule]
# edit vlan 2[Enter]                       <--- navigate to vlan 2
  ok
[edit vlanModule vlan "2"]
# set stgId 0[Enter]                       <--- set  stgId
  ok
[edit vlanModule vlan "2"]
# set vlanType byPort[Enter]               <--- set  vlanType
  ok
[edit vlanModule vlan "2"]
# set vlanPortMembers 1/1/1-2/1/3[Enter]   <--- set  vlanPortMembers
  ok
[edit vlanModule vlan "2"]
# set vlanProtocolId 1[Enter]              <--- set  vlanProtocolId
  ok
[edit vlanModule vlan "2"]
```

20 Assumes you've provided startup data for this module, similar to the above ndGarageMib example

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
# create ip 1|1.2.3.4[Enter]                <--- create row in ip-adddr table, key = ifIndex|ipAddr
  ok
[edit vlanModule vlan "2"]
# edit ip 1|1.2.3.4[Enter]                  <--- navigate to ip
  ok
[edit vlanModule vlan "2" ip "1|1.2.3.4"]
# set netMask 255.255.128.0[Enter]          <--- set  netMask
  ok
[edit vlanModule vlan "2" ip "1|1.2.3.4"]
# set bcastAddrFormat 0[Enter]              <--- set  bcastAddrFormat
  ok
[edit vlanModule vlan "2" ip "1|1.2.3.4"]
# set reasmMaxSize 1234[Enter]              <--- set  reasmMaxSize
  ok
[edit vlanModule vlan "2" ip "1|1.2.3.4"]
# set macOffset 0[Enter]                    <--- set  macOffset
  ok
[edit vlanModule vlan "2" ip "1|1.2.3.4"]
#set vrfId 15[Enter]                        <--- set  vrfId
  ok
[edit vlanModule vlan "2" ip "1|1.2.3.4"]
#up[Enter]                                 <--- move up to vlan 2
[edit vlanModule vlan "2"]
#show[Enter]                               <--- display vlan 2
// candidate //
      vlanId 2
      vlanName vlan#2
      stgId 0
      vlanType byPort
      vlanPortMembers 1/1/1-2/1/3
      vlanProtocolId 1
      ip
        ifIndex 1
        addr 1.2.3.4
        netMask 255.255.128.0
        bcastAddrFormat 0
        reasmMaxSize 1234
        macOffset 0
        vrfId 15
[edit vlanModule vlan "2" ]
# commit[Enter]                             <--- commit
  ok
[edit vlanModule vlan "2" ]
# quit[Enter]
Candidate configuration is now unlocked.
> show running[Enter]                       <--- show running configuration
// running //
 ...
      vlanId 2
      vlanName vlan#2
      stgId 0
      vlanType byPort
      vlanPortMembers 1/1/1-2/1/3
      vlanProtocolId 1
      ip
        ifIndex 1
        addr 1.2.3.4
        netMask 255.255.128.0
        bcastAddrFormat 0
        reasmMaxSize 1234
        macOffset 0
        vrfId 15
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

6.3 ndcliclient

ndcliclient is the remote client for **ndconfsvr** when it is running in the background. The only distinction in the CLI provided between running this with **ndconfsvr** in the background and the foreground is that the “exit” and “quit” commands, when at the top level of the CLI, pertains to exiting **ndcliclient**, not causing the server to exit.

6.3.1 Command Line Options

ndcliclient *[-p port] [-t traceid]*

```
>  ndcliclient
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

7 Working with ncclient

ncclient is a command line NETCONF client. The command line handling is similar to the CLI AA's CLI. The difference is that ncclient does not have a configuration mode. Rather, commands are an implementation of NETCONF rpc commands, defined in the ietf-netconf Yang module. A few commands (E.g. edit-config) expect xml as an argument/payload, and it would be impractical to provide such input in a command line. So the argument for these commands is the name of the file that contains part of the **xml** string required.

ncclient also listens for 'call-home' connections from a netconf server, on TCP port 4334 (SSH) and TCP port 4335 (TLS). (See RFC 8071). When such a connection request is received, an automated connection request of the same kind (SSH/TLS) is attempted to the calling server.

ncclient supports multiple concurrent connections to one or more NETCONF servers.

7.1 `ncclient` Command Line Options

`ncclient` *`[-s <serverAddr>] [-a ssh|tls] [-r cert] [-u <username>] [-p password] [-c configDir] [-f cfgparams] [-e script] [-w waittime] [-h]`*

`-s`	Address of server to connect to. When specified, **ncclient** executes a 'connect server'.
`-a`	Specifies which security protocol to use **ssh** or **tls**.
`-r`	Certificate name (**tls** only)
`-u`	User name (**ssh** only). When specified with the -s option, **ncclient** executes a 'connect server user'.
`-p`	Password (**ssh** only). When specified with the -s and -u command, ncclient executes a 'connect server user pwd'.
`-c`	folder with ncclient configuration files, default is the **config** sub-folder of the folder where **testagent** is located.
`-l`	folder where log files are stored, default is ./**log**.
`-f`	Path to **cfgparams.xml** file
`-e`	Script file to execute.
`-w`	Specifies a period of time, in milliseconds to wait between commands. This option is used in conjunction with the -f option to allow the previous command to output results. The longer the server needs to process a command, the longer this value should be. Default 100 ms.
`-h`	Print usage.

7.2 Using `ncclient`

To start **ncclient**, from **<SDK installdir>/ndconf/ncclient**, type **./ncclient** in a terminal window.

7.2.1 connect

The **ncclient** "connect" command establishes communication with a netconf server. If the "connect" succeeds, **ncclient** sends a "hello" message (shown in light blue below), and reads a "hello" from server.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

7.2.1.1 Connecting using SSH

```
> connect 127.0.0.1 ssh <a valid user name here> pwd[Enter]
Password: ********
Fingerprint: A0 E0 6A E9 E1 29 85 3E 77 37 AB 20 97 88 03 88 83 5B 78 ED
Authentication methods: publickey,password,keyboard-interactive
Sending NETCONF client <hello>
<?xml version="1.0" encoding="UTF-8"?>
<hello>
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
</capabilities>
</hello>
]]>]]>
Reading NETCONF server <hello>
  capabilities
    capability urn:ietf:params:netconf:base:1.0
...
1:S admin@127.0.0.1>
```

The above example assumes the user name was ‘admin’.

SSH also supports the use of public keys and or certificates. For a key the format is:

```
> connect 127.0.0.1 ssh <a valid user name here> key clindt[Enter]
```

If ‘key’ authentication is specified, the value expected is the name of the public-key file without extension ".pub".

In the example above **ncclient** will look for **clindt.pub** file in the <install_dir>/ndconf/ncclient/config/ssh folder.

or using certificates, the format is:

```
> connect 127.0.0.1 ssh <a valid user name here> cert clindt[Enter]
```

where ‘**clindt**’ could be the name of a configured public key or certificate. If **cert** is specified as an authentication method, the value expected is the portion of the certificate file name without the suffix "-cert.pub" (ssh-keygen adds suffix -cert to the public-key file name when generating certificate).

In the example above **ncclient** will look for the **clindt-cert.pub** file in <install_dir>/ndconf/ncclient/config/ssh folder.

After a successful connection the format of the command prompt changes to something like the following:

```
> 1:T admin@localhost
```

Assuming this was the ‘pwd’ login of ‘admin’ mentioned above.

See the section for the “**nc**” command for information on prompt format.

A final note pertaining to **SSH**, ncclient **supports** both ~/.ssh/known_hosts and ~/.ssh/authorized_keys. See sshd documentation (i.e. man sshd) for details, format and use of these files. Use of these files is **recommended**. If ~/.ssh/known_hosts has been configured, then the server specified in the connect command **MUST** be a valid entry in the ~/.ssh/known_hosts file, otherwise the connect will fail and you’ll see a failure message similar to:

```
SSH Host check error: 1
FAILED to connect to 127.0.0.1:830
>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

7.2.1.2 Connecting using TLS[21]

```
> connect 127.0.0.1 tls <a valid certificate name here>[Enter]
Sending NETCONF client <hello>
<?xml version="1.0" encoding="UTF-8"?>
<hello>
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
</capabilities>
</hello>
]]>]]>
Reading NETCONF server <hello>
  capabilities
    capability urn:ietf:params:netconf:base:1.0
...
1:T cliadmin@127.0.0.1>
```

The above example assumes the certificate was ‘cliadmin’.

ncclient needs access to both certificate and the corresponding private key. It looks for these files in the <install_dir>/ndconf/ncclient/config/tls/certs and <install_dir>/ndconf/ncclient/config/tls/private folders respectively. At this point, a session is created, and the user can issue NETCONF commands.

7.2.2 When a netconf server “calls-home”

When a server calls home, the **CLI** of **ncclient** is interrupted and you will see an indication of the event, something like the following:

```
> AfterAccept callhome-1


Sending NETCONF client <hello>
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
</capabilities>
</hello>
]]>]]>
```

If the connection is successful and the server responds to the client hello message, the usual scroll of the server’s hello message will be seen, after which there will be a change in the command prompt as outlined in a next section.

7.2.3 nc – “netconf connection”

As stated above, **ncclient** supports multiple concurrent connections to one or more **netconf** servers. This command permits you to manage these connections.

Information about the connection context can be determined from the information in the prompt. The following is an example:

```
1:T admin@localhost>
```

21 By default there, are five example certificate provided in the <installation directory>/ncclient/config/tls/certs directory. The provided certificate names are cliadmin, clisan1, clisan2, clisan3,clisan4 and clisan5.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

The number '**1**' indicates the "connection-id", the letter '**T**' indicates that this is a **TLS** session, and '**admin**' is the **username**, and **localhost** is the target server. A letter '**S**' would indicate that a session is **SSH**.

Any NETCONF command (get, edit-config, lock, ...) will be sent over the connection indicated by the connection-id in the prompt. **nc** stands for "netconf connection".

When **nc** is executed without an argument, the command prints all currently established connections, e.g.:

```
1:T admin@localhost> nc
* 1: TLS - admin@127.0.0.1:6513 - session-id: 3 cert: cliadmin
```

Note that the connection id (the number at the beginning of line) is not usually the same as the NETCONF session id (printed after "session-id:"). Each connection has an unique connection-id, and, of course, session-id. The reason why there are 2 numbers, instead of just a session-id is the following scenario: Given 1 ncclient and 2 testagents.

```
Connecting to the 1st testagent, the session id will be "1".
Connecting to the 2nd testagent, the session id will be "1".
```

Now there are two NETCONF connections (i.e. to nc-s) and both have session-id "1". With a different connection-id it is easy to distinguish between these 2 NETCONF sessions.

A user can switch between connections by executing **nc <id>** where **<id>** is the connection-id.

The "*" indicates which connection is "active".

To create another session/connection execute **nc new**:

```
1:T admin@localhost> nc new
>
```

When the prompt changes to a single '>' the user can issue another 'connect' command:

```
> connect localhost ssh user1 pwd
Password: ****
Authentication methods: publickey,gssapi-keyex,gssapi-with-mic,password,keyboard-interactive
...
session-id 4
2:S user1@localhost>
```

Note that the new connection automatically becomes active. 2 is connection-id, "S" indicates that this is a SSH connection, user1 is the user name and localhost is the target server.

Executing nc again:

```
2:S user1@localhost> nc
1: TLS - admin@127.0.0.1:6513 - session-id: 3 cert: cliadmin
* 2: SSH - user1@127.0.0.1:830 - session-id: 4
```

To switch back to connection 1:

```
2:S user1@localhost> nc 1
1:T admin@localhost>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

7.2.4 `ndGarageMib` Example

Here is an example how to create a row in the **vehiclesTable**:

```
> get-config candidate xpath //ndg:garage[Enter]    <--- get the candidate configuration
  data
    garage
...
      vehiclesEntry
        vehicleIndex 2
        vehicleLicencePlate 345 XYZ
        vehicleModel Jaguar F-Type
      vehiclesEntry
        vehicleIndex 3
        vehicleLicencePlate 555 QQQ
        vehicleModel Audi TT
> lock candidate[Enter]                        <--- lock configuration
  ok
> list edit-config c1[Enter]                   <--- display content of file to be used as argument
  <default-operation>merge</default-operation>
  <test-option>set</test-option>
  <config>
    <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
      <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create" >
        <t:vehicleIndex>4</t:vehicleIndex>
        <t:vehicleLicencePlate>777 SWE</t:vehicleLicencePlate>
        <t:vehicleModel>Koenigsegg One:1</t:vehicleModel>
      </t:vehiclesEntry>
    </t:garage>
  </config>

> edit-config candidate c1[Enter]              <--- execute edit-config rpc
  ok
> lock running[Enter]                          <--- lock configuration
  ok
> commit[Enter]                                <--- commit changes to running configuraton
  ok
> unlock running[Enter]                        <--- unlock configuration
  ok
> unlock candidate[Enter]                      <--- unlock configuration
  ok
> get xpath //ndg:garage[Enter]                <--- get running configuration
  data
    garage
...
      vehiclesEntry
        vehicleIndex 2
        vehicleLicencePlate 345 XYZ
        vehicleModel Jaguar F-Type
      vehiclesEntry
        vehicleIndex 3
        vehicleModel Audi TT
        vehicleLicencePlate 555 QQQ
      vehiclesEntry                                <--- new entry
        vehicleIndex 4
        vehicleLicencePlate 777 SWE
        vehicleModel Koenigsegg One:1
>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

7.2.5 `rcvlan` based example

```
> get-config candidate xpath //rcv:vlan[Enter]    <--- get vlan branch in candidate configuration
rpc-reply
data
    vlan
      vlanId 1
      vlanName vlan#1
      stgId 0
      vlanType 1
      vlanPortMembers 43:04:00:40
      vlanProtocolId 1
      ip
        ifIndex 1
        addr 192.168.0.11
        netMask 255.255.0.0
        bcastAddrFormat 0
        reasmMaxSize 1234
        macOffset 0
        vrfId 15
    vlan
      vlanId 3
      vlanName vlan no3
      stgId 0
      vlanType byPort
      vlanPortMembers 1/1/1-2/1/2
      vlanProtocolId 1
      ip
        ifIndex 1
        addr 1.2.3.4
        netMask 255.255.128.0
        bcastAddrFormat 0
        reasmMaxSize 1234
        macOffset 0
        vrfId 15
> lock candidate[Enter]                        <--- lock configuration
  ok
> list edit-config createvlan[Enter]           <--- display content of createvlan.xml
  <test-option>set</test-option>
  <config>
    <t:vlanModule xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:rcvlan" >
      <t:vlan xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create" >
      <t:vlanId>2</t:vlanId>
      <t:vlanName>vlan#2</t:vlanName>
      <t:stgId>22</t:stgId>
      <t:vlanType>1</t:vlanType>
      <t:vlanPortMembers>13:04:11:20</t:vlanPortMembers>
      <t:vlanProtocolId>1</t:vlanProtocolId>
      <t:ip>
        <t:ifIndex>2</t:ifIndex>
        <t:addr>192.168.0.22</t:addr>
        <t:netMask >255.255.0.0</t:netMask>
        <t:bcastAddrFormat>0</t:bcastAddrFormat>
        <t:reasmMaxSize>1234</t:reasmMaxSize>
        <t:macOffset>0</t:macOffset>
        <t:vrfId>15</t:vrfId>
      </t:ip>
    </t:vlan>
    </t:vlanModule>
  </config>

> edit-config candidate createvlan[Enter]      <--- execute edit-config rpc
  ok
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
> lock running[Enter]                          <--- lock configuration
  ok
> commit[Enter]                                <--- commit changes to running configuraton
  ok
> unlock running[Enter]                        <--- unlock configuration
  ok
> unlock candidate[Enter]                      <--- unlock configuration
  ok
> get xpath //rcv:vlan[Enter]                  <--- get running configuration
rpc-reply
  data
    vlan
      vlanId 1
      vlanName vlan#1
      stgId 0
      vlanType 1
      vlanPortMembers 43:04:00:40
      vlanProtocolId 1
      ip
        ifIndex 1
        addr 192.168.0.11
        netMask 255.255.0.0
        bcastAddrFormat 0
        reasmMaxSize 1234
        macOffset 0
        vrfId 15
    vlan
      vlanId 3
      vlanName vlan no3
      stgId 0
      vlanType byPort
      vlanPortMembers 1/1/1-2/1/2
      vlanProtocolId 1
      ip
        ifIndex 1
        addr 1.2.3.4
        netMask 255.255.128.0
        bcastAddrFormat 0
        reasmMaxSize 1234
        macOffset 0
        vrfId 15
    vlan
      vlanId 2
      vlanName vlan#2
      stgId 22
      vlanType 1
      vlanPortMembers 13:04:11:20
      vlanProtocolId 1
      ip
        ifIndex 2
        addr 192.168.0.22
        netMask 255.255.0.0
        bcastAddrFormat 0
        reasmMaxSize 1234
        macOffset 0
        vrfId 15
>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

7.2.6 Notifications in `ncclient`

To check the notification streams supported:

```
> get xpath //manageEvent:stream[Enter]
rpc-reply
  data
     stream
       name AACLI
       description AACLI stream
       replaySupport true
       replayLogCreationTime 2019-08-06T20:54:05Z
    stream
      name NETCONF
      description NETCONF stream
      replaySupport true
      replayLogCreationTime 2019-08-06T20:54:05Z
    stream
      name SNMP
      description SNMP stream
      replaySupport false
```

To subscribe for notifications:

```
> create-subscription[Enter]
rpc-reply
  ok
>
```

Later (libndGarageMib.so sends coLevelFallingAlarm and coLevelRisingAlarm periodically):

```
notification
  eventTime 2019-08-07T15:01:21Z
  coLevelFallingAlarm
    object-1
      garageCOLevel 9
```

The subscription can be terminated by using <close-session>

```
> close-session[Enter]
rpc-reply
  ok
>
```

or <kill-session> (from a different client instance)

When subscribing for notifications with replay, a user must specify {startTime} or {startTime, stopTime}.

If only the startTime is specified, agent will replay all available notifications in the log since the startTime. After finishing the replay agent sends a "replayComplete" notification. The agent will continue to send notifications to the client after that.

```
> create-subscription startTime 2019-08-06T00:00:00Z[Enter]
>
rpc-reply
  ok

notification
  eventTime 2019-08-06T23:16:04Z
  netconf-session-start
    username cli
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
    session-id 1

notification
  eventTime 2019-08-06T23:16:04Z
  netconf-session-start
    username snmp
    session-id 2

notification
  eventTime 2019-08-06T23:16:10Z
  netconf-session-start
    username user1
    session-id 3

notification
  eventTime 2019-08-06T23:16:25Z
  coLevelRisingAlarm
    object-1
      garageCOLevel 51

notification
  eventTime 2019-08-06T23:16:41Z
  replayComplete        <<<--- after this notification, session becomes regular notif subscriber

...

notification
  eventTime 2019-08-06T23:20:00Z
  coLevelFallingAlarm
    object-1
      garageCOLevel 9
```

There are two scenarios when stopTime is specified:

1) If the stopTime is before current time, agent will replay notifications up to the stopTime and cancel the subscription.

```
> create-subscription startTime 2019-08-15T00:00:00Z stopTime 2017-04-06T23:17:00Z[Enter]
>
rpc-reply
  ok

notification
  eventTime 2019-08-06T23:16:04Z
  netconf-session-start
    username cli
    session-id 1

notification
  eventTime 2019-08-06T23:16:04Z
  netconf-session-start
    username snmp
    session-id 2

notification
  eventTime 2019-08-06T23:16:10Z
  netconf-session-start
    username user1
    session-id 3

notification
  eventTime 2019-08-06T23:16:25Z
  coLevelRisingAlarm
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
    object-1
      garageCOLevel 51

notification
  eventTime 2019-08-06T23:21:07Z
  replayComplete

notification
  eventTime 2019-08-06T23:21:07Z
  notificationComplete     <<<--- after this notification agent will stop sending new notifications
```

2) If the stopTime is in the future, the agent will replay all available notifications in the log since the startTime and send a "replayComplete" notification. The agent will continue to send any new notifications until the stopTime, at which point the agent will send a "notificationComplete" notification.

```
> create-subscription startTime 2019-08-06T23:23:40Z stopTime 2017-04-06T23:30:00Z[Enter]
>
rpc-reply
  ok

notification
  eventTime 2019-08-06T23:23:42Z
  coLevelRisingAlarm
    object-1
      garageCOLevel 51

notification
  eventTime 2019-08-06T23:24:16Z
  replayComplete     <<<--- after this notification agent will continue to send notifications ...

...

notification
  eventTime 2019-08-06T23:24:33Z
  coLevelFallingAlarm
    object-1
      garageCOLevel 9

...

notification
  eventTime 2019-08-06T23:30:00Z   <<<--- ... until stopTime
  notificationComplete
```

An error may be reported, depending on the validity of the subscription request. E.g. if the startTime is in the future:

```
> create-subscription startTime 2019-04-24T10:00:00Z[Enter]
>
rpc-reply
  rpc-error
    error-type protocol
    error-tag bad-element
    error-severity error
    error-app-tag invalid-data
    error-path /rpc/create-subscription/startTime
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

7.3 nccliout

nccliout is an application that can be used to receive the response output and notifications from **ncclient**. If this application is running when **ncclient** is started, it is enabled for use automatically. Otherwise if **nccliout** is started after **ncclient** is started, you need to use the **redirectqueue** command in ncclient.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

8 Working with SNMP

8.1 Configuring ndtaasnmp.so.0

There are several options available to the SNMP Access Agent. Documentation for these are as follows.

8.1.1 <option name="postCommitInterval" ...

By default, the SNMP access agent commits the changes to the datastore when SNMP Set processing completes successfully. By specifying a value for **postCommitInterval**, SNMP will defer committing changes to the datastore for the period in seconds to allow for subsequent successful SNMP Sets to be set at the same time. This is done in the configuration file for the **testagent** (or **ndconfsvr**). In **ndconf.xml**, change the line:

```
<agent name="snmp" dll="libndtaasnmp.so.0"></agent>
```

to include ‘options’ and specify the ‘option’ named “postCommitInterval” and provide the value of the interval in seconds you wish to configure. E.g.

```
<agent name="snmp" dll="libndtaasnmp.so.0">
    <options>
        <option name=”postCommitInterval” value=”10” />
    <options>
</agent>
```

As subsequent SNMP Sets are received, this interval is reset to the value specified to extend for another postCommitInterval period.

Setting a value less than 1 second will disable this feature.

8.1.2 <option name="postCommitMaxInterval" ...

When the above feature is activated, this option is used to specify the absolute maximum time the commit interval may extend to, in secodns. E.g.

```
<agent name="snmp" dll="libndtaasnmp.so.0">
    <options>
        <option name=”postCommitInterval” value=”10” />
        <option name=”postCommitMaxInterval” value=”25” />
    <options>
</agent>
```

Sets the maximum commit deferment interval to 25 seconds. The default is 100 seconds if not specified and postCommitInterval is active.

8.1.3 <option name="postCommitSaveToStartup" …

This option instructs the the SNMP access agent to also commit the changes to the :startup datastore.

E.g.

```
<option name=”postCommitSaveToStartup” value=”true” />
```

When the value is not “true”, this option is disabled. The default is false.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

8.2 Default SNMP Users

In order to work with an SNMP manager such are MIBrowser Pro, you 'll need information about the user configurations that ndCONF provides by default.

The SNMP users provided are not secure for deployment, since they are known in the public domain and intended only for test purposes. The configuration for SNMP users is found in the startup-cfg.xml file, in the <snmp> section and the <community><usm> and <vacm> subsections of <snmp> in that file.

8.2.1 SNMP v1/v2c

The community '**`public`**' is defined for v1/v2c. This 'user' has limited, read-only access to the overall MIB arc, allowing only the '**system**' (1.3.6.1.2.1.1) group of scalar objects and anything under the '**nuDesignRoot**' (1.3.6.1.4.1.4761) arc, including **`ND-GARAGE-MIB`** implementation provided with the SDK.

8.2.2 SNMP v3

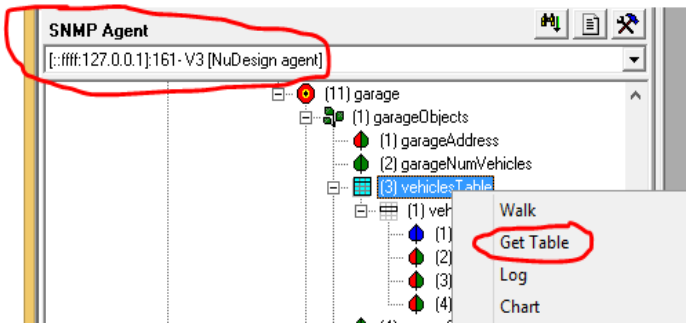
The USM users '**sha**' and '**shaaes**' are defined for v3. These users have read/write access to the entire MIB.

'**`sha`**' is a **`authNoPriv`** user with authentication password '**`shaauth`**'. It uses sha(128) authentication.

'**`shaaes`**' is a **`authPriv`** user with authentication password '**`shaaesauth`**' and privacy password '**`shaaespriv`**'. It uses sha(1) authentication and AES(128) privacy.

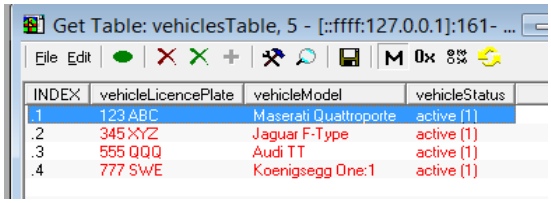
8.3 Working with MIBrowser

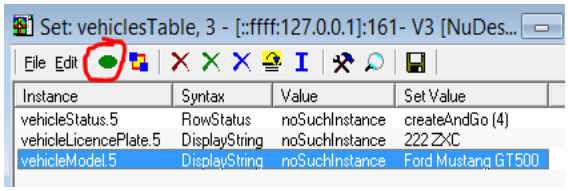
To interact with the SNMP access agent you need to use an SNMP manager. For ther purposes of this document, we are using NDT's MIBrowser Pro (You can download the evaluation version of MIBrowser Pro at NuDesign web site at https://www.ndt-inc.com/ndt/eval-download/). Start Visual MIBrowser Pro, load ND-GARAGE-MIB (NDGarageV2.mib), select the testagent's address as the target (for this example: 127.0.0.1), right click "vehicleTable" and select "Get Table" from the context menu (initiated using a right mouse click):



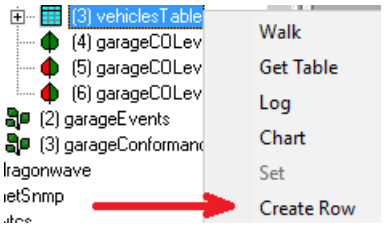
produces the following:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

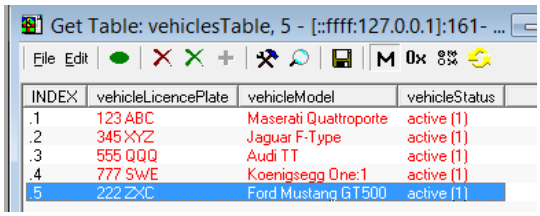




Now create a new row with MIBrowser Pro, select “Create Rpw”:

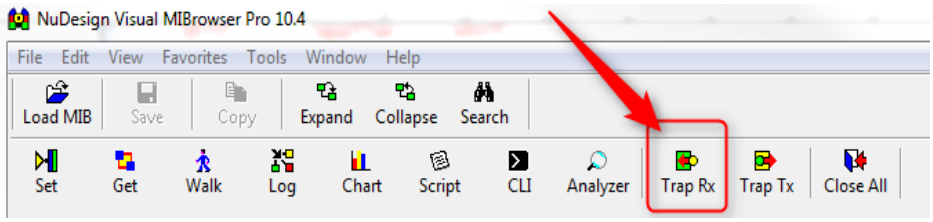


Fill in the details and then send the "set" request by clicking "Execute" button (circled in red). If there were not creation problems, then you can refresh the getTable window if it doesn't appear automatically:



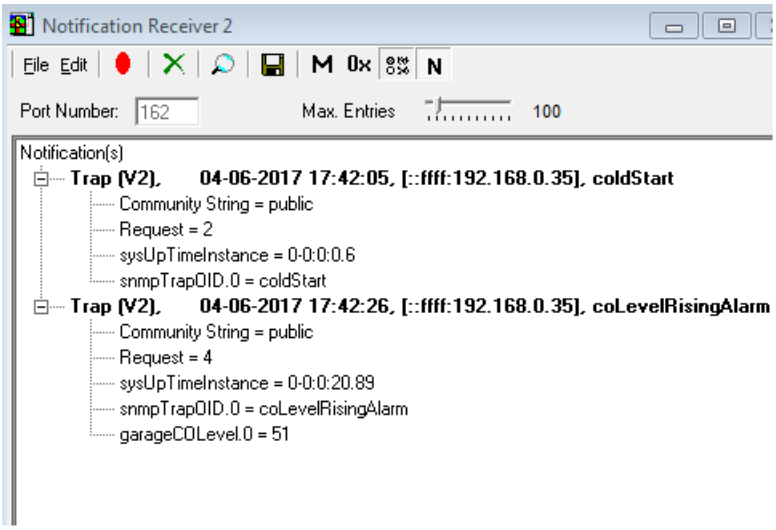
8.3.1 Notifications in MIBrowser

In the following section, it is assumed that the server has been configured with a target that is the same as the server (E.g. 127.0.0.1 default). If this is not the case, you'll need to configure the server to have a target that is the that of MIBrowser Pro. Open a trap receiver (Trap Rx):



then (re)start the **testagent**. You'll see something like the following:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



The first entry is an SNMP v3 trap notification indicating libndtsnmpaa.so has loaded and been started. i.e. the coldstart.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

8.4 Putting it All together

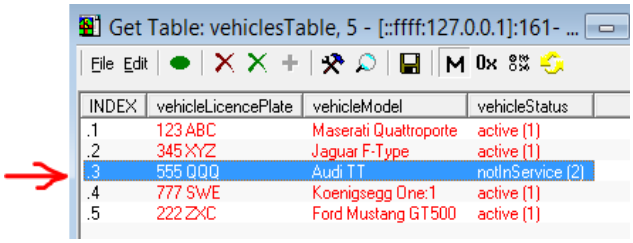
Retrieving configurations from **`ncclient`**:

```
> get-config candidate xpath //ndg:vehiclesEntry[Enter]    <--- get candidate configuration
  data
    garage
      vehiclesEntry
        vehicleIndex 1
        vehicleLicencePlate 123 ABC
        vehicleModel Maserati Quattroporte
      vehiclesEntry
        vehicleIndex 2
        vehicleLicencePlate 345 XYZ
        vehicleModel Jaguar F-Type
      vehiclesEntry
        vehicleIndex 3
        vehicleLicencePlate 555 QQQ
        vehicleModel Audi TT
      vehiclesEntry
        vehicleIndex 4
        vehicleLicencePlate 777 SWE
        vehicleModel Koenigsegg One:1
      vehiclesEntry
        vehicleIndex 5
        vehicleLicencePlate 222 ZXC
        vehicleModel Ford Mustang GT500
> get-config running xpath //ndg:vehiclesEntry[Enter]      <--- get running configuration
  data
    garage
      vehiclesEntry
        vehicleIndex 1
        vehicleLicencePlate 123 ABC
        vehicleModel Maserati Quattroporte
      vehiclesEntry
        vehicleIndex 2
        vehicleLicencePlate 345 XYZ
        vehicleModel Jaguar F-Type
      vehiclesEntry
        vehicleIndex 3
        vehicleLicencePlate 555 QQQ
        vehicleModel Audi TT
      vehiclesEntry
        vehicleIndex 4
        vehicleLicencePlate 777 SWE
        vehicleModel Koenigsegg One:1
      vehiclesEntry
        vehicleIndex 5
        vehicleLicencePlate 222 ZXC
        vehicleModel Ford Mustang GT500
>
```

The running and candidate configurations are the same. The reason is that, by default, SNMP AA synchronizes running and candidate after each successful set request.

Set one of the rows in the table to ‘notInService’ by right mouse clicking in the row and selecting it:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



Only 'active' rows are part of configuration. You can verify this in **`ncclient`**:

```
> get-config running xpath //ndg:vehiclesEntry[Enter]      <--- get running configuration
  data
    garage
      vehiclesEntry
        vehicleIndex 1
        vehicleLicencePlate 123 ABC
        vehicleModel Maserati Quattroporte
      vehiclesEntry
        vehicleIndex 2
        vehicleLicencePlate 345 XYZ
        vehicleModel Jaguar F-Type                          <--- #3 is deleted
      vehiclesEntry
        vehicleIndex 4
        vehicleLicencePlate 777 SWE
        vehicleModel Koenigsegg One:1
      vehiclesEntry
        vehicleIndex 5
        vehicleLicencePlate 222 ZXC
        vehicleModel Ford Mustang GT500
>
```

Delete a row using **`ncclient`**:

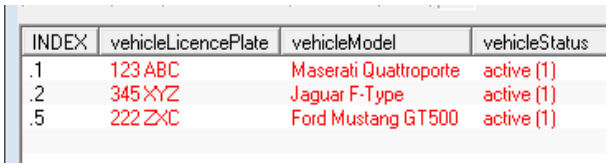
```
> list edit-config d1[Enter]                                <--- content of d1.xml
  <default-operation>merge</default-operation>
  <test-option>set</test-option>
  <config>
    <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
      <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="delete" >
        <t:vehicleIndex>4</t:vehicleIndex>
      </t:vehiclesEntry>
    </t:garage>
  </config>

> lock candidate[Enter]                          <--- lock configuration
  ok
> edit-config candidate d1[Enter]                <--- execute edit-config rpc (arg = d1.xml)
  ok
> lock running[Enter]                            <--- lock configuration
  ok
> commit[Enter]                                  <--- commit changes to running configuraton
  ok
> unlock running[Enter]                          <--- unlock configuration
  ok
> unlock candidate[Enter]                        <--- unlock configuration
  ok
> get-config running xpath //ndg:vehiclesEntry[Enter]      <--- get running configuration
  data
    garage
      vehiclesEntry
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
        vehicleIndex 1
        vehicleLicencePlate 123 ABC
        vehicleModel Maserati Quattroporte
      vehiclesEntry
        vehicleIndex 2
        vehicleLicencePlate 345 XYZ
        vehicleModel Jaguar F-Type                <--- #4 is deleted
      vehiclesEntry
        vehicleIndex 5
        vehicleLicencePlate 222 ZXC
        vehicleModel Ford Mustang GT500
>
```

Check with MIBrowser, by refreshing the window:



INDEX	vehicleLicencePlate	vehicleModel	vehicleStatus
.1	123 ABC	Maserati Quattroporte	active (1)
.2	345 XYZ	Jaguar F-Type	active (1)
.5	222 ZXC	Ford Mustang GT500	active (1)

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

9 Configuring DataServer Options

There are several options that are “global”. These are as follows and pertain to configuring for

9.1 <option name="nacmtrace" …

This option control NACM tracing. Default is **`false`**. E.g.

```
<ndt>
    <options>
        <option name="nacmtrace" value="true"></option>
            :
            :
    </options>
</ndt>
```

9.2 Configuring to Retrieve User groups Using an External Application

The data server supports using an external application to retrieve user group information[22]. The SDK includes an application to retrieve user group information from a FreeRADIUS server called ‘ndtrdsclient’. If your web server is using RADIUS user authentication, this allows a user to configure NACM user and group information in a single system[23].

Note: Using this mechanism requires that the NACM be enabled (<enable-nacm> is “true”) and that <enable-external-groups> is “true”.

Note: This requires certain data server configuration parameters to use the application as well as a suitable configuration in the RADIUS server.

9.2.1 ndtrdsclient

The executable is used by the data server to retrieve user group information from a suitably configured RADIUS server (See below). When successful, ndtrdsclient outputs a '|' separated list of groups the user is a member of to stdout.[24] E.g. “users|admin” would indicate the user are part of both “users” and “admin”.

The command line options are:

`ndtrdsclient [-n` *`port`* **`] [-t`** *`timeout`* **`] [-d`** *`retries`* **`] -s`** *`secret`* **`-a`** *`address`* **`-u`** *`user`* **`-p`** *`password`*

The options are as follows:

```
-n port       UDP port to access the RADIUS server. Default: 1812.
-t timeout    Specifies the request timeout period in seconds. Default: 1.
-r retries    Specifies the request retries. Default: 1.
-s secret     This is the RADIUS server "secret".
-a address    This is the address of the RADIUS server.
-u user       Specifies the user for which group information is being sought.
-p password   Specifies the user password.
```

22 Provided an application adhere to the same specifications, another similar application could be substituted.
23 Currently this provision only works with RESTCONF.
24 There is a man page for this application as well.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

9.2.2 RADIUS configuration requirements

The documentation below outlines the required configuration changes to successfully deploy ndtrdsclient. The examples below are specific to the FreeRADIUS 3.0 server, using a file based configuration.

Open the ‘users’ file and for each user, add each user’s specific information, using the following template:

```
user1   Cleartext-Password := "user1-password"
        Reply-Message := "User-Groups=user1-group1|user1-group2"       # I.e. “the group info”.
```

Where ‘user1’ is a hypothetical user’s user name; “**user1-password**” is the ‘**user1**‘ password and “**user1-group1**” and “**user1-group2**” are the groups to which ‘**user1**’ belongs.

Note: when a user is a member of several groups, the group names are separated by a ‘|’ as shown above.

In the ‘**default**’ file[25] under the **sites-available** directory, in the ‘post-auth {‘ section, find the ‘Post-Auth-Type REJECT {‘ section and add the following content

```
         Post-Auth-Type REJECT {
       :
       :

                 # when an auth fails, if "User-Group" exists in the
                 # replay message, then it's been introduced by our
                 # "group" handling in “users", so remove it so that
                 # we don't "leak" user / group information inadvertently
        # during an unsuccessful authentication.
                 if ( &reply:Reply-Message =~ /User-Groups/ ) {
                         update {
                                 &reply:Reply-Message := ""
                         }
                 }
       :
       :
         }
```

9.3 <option name="grpAuthExe" …

This option defines the path and name of the external application to retrieve user group information with. The current working directory is assumed if no or relative path information is provided.

```
<ndt>
    <options>
        <option name="grpAuthExe" value="ndtrdsclient"></option>
          :
    </options>
</ndt>
```

9.4 <option name="authAddr" …

This option defines the network address of the authentication (RADIUS) server.

```
<ndt>
        <options>
            <option name="authAddr" value="192.168.2.254"></option>
            :
        </options>
</ndt>
```

25 Only perform this step in a RESTCONF only configuration. i.e. the NETCONF server isn’t part of the configuration.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

9.5 <option name="authPswd" …

This option defines the ‘secret’ (password) for access the authentication (RADIUS) server.

```
<ndt>
       <options>
           <option name="authPswd" value="testing123"></option>
           :
           :
       </options>
</ndt>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

10 Configuring NETCONF

There is a several options available for the NETCONF Access Agent. Documentation for them follows.

10.1 <option name="numthreads" …

This options controls how many service threads this access agent will maintain. The default is three (‘3’) threads. To change this option, add an <option> section within the <agent> definition for NETCONF access agent. E.g.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="numthreads" value="5"></option>
    </options>
</agent>
```

10.2 <option name="enableSSH" …

This options controls whether the SSH security protocol is in use. By default, access via SSH is enabled. You may disable SSH access by setting the option “enableSSH” to “false”. E.g.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="enableSSH" value="false"></option>
    </options>
</agent>
```

10.3 <option name="enableTLS" …

This options controls whether the TLS security protocol is in use. By default, access via TLS is enabled, provided the required certificates and keys are in place (see below). You may disable TLS access by setting the option “`enableTLS`” to “`false`”. E.g.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="enableTLS" value="false"></option>
    </options>
</agent>
```

10.4 Configuring TLS certificates

For TLS to be enabled, by default a “TLS” directory hierarchy must exist, subordinate to the config directory. E.g.

```
.../config
     /tls
          /ca-certs
          /certs
          /private
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

By default “ca-certs” directory contains the CA certificate chain file “ca-chain.pem”, the “certs” directory contains the file “ndconfsrv.pem” and the “private” directory contains the file “ndconfsrv.key”.

The defaults identified above can be altered with the following options.

serverCert	The value contains the path and name of the server certificate file. By default this is “ndconfsrv.pem” as described above.
serverKey	The value contains the path and name of the server key file. By default this is “ndconfsrv.key” as described above.
CAFile	The value contains the path and name of the CA certificate chain file. By default this is “ca-chain.pem” as described above.

E.g.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options key=”cert”>
        <option name="serverCert"
value="<install_dir>/ndconf/testagent/config/tls/cert/ndconfsrv.pem"></option>
        <option name="serverKey"
value="<install_dir>/ndconf/testagent/config/tls/private/ndconfsrv.key"></option>
        <option name="CAFile" value="<install dir>/ndconf/testagent/config/tls/ca-cert/ca-
chain.pem"></option>
    </options>
</agent>
```

10.5 <option name="call-home-reconnect-interval" …

This option controls the re-connection attempt for a call-home connection, in seconds. Default: “10”.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="call-home-reconnect-interval" value="30"></option>
    </options>
</agent>
```

10.6 <option name="max-incoming-connections" …

This option controls the maximum number of incoming call-home connections. Default: “10”.

Default: “10”.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="max-incoming-connections" value="5"></option>
    </options>
</agent>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

10.7 <option name="call-home-sshd-debug" …

This option controls whether the call-home connection SSH server is started in debug mode or not. Default: “false”.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="call-home-sshd-debug" value="true"></option>
    </options>
</agent>
```

10.8 <option name="call-home-sshd-exe" …

This option specifies the path and name of the call-home SSH server. Default: “/usr/sbin/sshd”.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="call-home-sshd-exe" value="/usr/local/sshd"></option>
    </options>
</agent>
```

10.9 <option name="call-home-sshd-cfg" …

This option specifies the path and name of the call-home SSH server configuration file. Default: “ndchsshd_config”.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="call-home-sshd-cfg" value="some_other_path_and_name"></option>
    </options>
</agent>
```

10.10 <option name="call-home-sshd-ndncsub" …

This option specifies the name of the call-home NETCONF pipe process executable. Default: “ndncsub”.

```
<agent name="nc" dll="libndtaanc.so.0">
    <options>
        <option name="call-home-sshd-ndncsub" value="some_other_exe_name"></option>
    </options>
</agent>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

11 Working with RESTCONF

The provided RESTCONF server is normally expected to be used via a third party RESTCONF client application. However, as mentioned in the installation section, we provide a simple web page. This is ‘restconf-submit-test.html’. . This is provided strictly for limited testing and demonstration purposes.

As a foreword, the current configuration of the RESTCONF server and the associated web server provide two virtual directories features. These are are:

- `/.well-known/`
- `/restconf/`

The first is used to identify the latter and permits only a single subsequent string in the URI, ‘host-meta’. i.e. ‘/.well-known/host-meta’. When this URI is submitted, the server responds with something like the following:

```
<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
<Link rel='restconf' href='/restconf'/></XRD>
```

depending on the selected ‘Accept’ header. This basically informs a RESTCONF client where the RESTCONF ‘root’ is. See RFC 8040 for more information on this and subsequent discussions.

Given the above response, all further interactions are prefixed with ‘/restconf/’. There are four acceptable sub paths for all RESTCONF requests:

- data, which is where the combine configuration and state data resources are
- operations, which is used for Data-model-specific operations
- yang-library-version, which provides the ‘ietf-yang-library’ revision date.
- Streams, which provides access to the RESTCONF notification streams NETCONF, NETCONF-JSON, NETCONF-TEXT, SNMP, SNMP-JSON, SNMP-TEXT, SYSLOG, SYSLOG-JSON and SYSLOG-TEXT[26].

11.1 Configuring libndtaarestconf.so.0

There are several options available to the RESTCONF Access Agent. Documentation for these are as follows.

11.1.1 <option name="socket" ...

By default, the RESTCONF access agent uses **tcp port 9010**, to communicated with the web server. If for some reason this port is unavailable, you can change the port. This is done in the configuration file for the **testagent** (or **ndconfsvr**). In **ndconf.xml**, change the line:

```
<agent name="rc" dll="libndtaarestconf.so.0"></agent>
```

to include ‘options’ and specify the ‘option’ named “socket” and provide the value of the port you wish to use, preceded by a “:”.

E.g.

26 Documented later in this section.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
<agent name="rc" dll="libndtaarestconf.so.0">
 <options>
     <option name="socket" value=":9010"></option>
 </options>
</agent>
```

11.1.2 <option name="HttpsOnly" ...

By default the RESTCONF Access Agent will allow requests from either ‘http’ or ‘https’. You may configure it so that it will only accept requests using the ‘https’ protocol stream. This of course assumes you’ve configured your web server to operated with ‘https’[27]. To do so, this will require that you have the correct authentication certificates in place to allow this to happen. If you do not have certificates available from a CA, you can create ‘self-signed’ certificates for testing. Consult you web server’s documentation as to how to proceed with enabling https.

The following options controls limiting the RESTCONF access agent to https:

```
<agent name="rc" dll="libndtaarestconf.so.0">
 <options>
     <option name="Httpsonly" value="true"></option>
 </options>
</agent>
```

Any value other than “**true**” will allow the RESTCONF access agent to operate with either **http** or **https**. If a request is received on **http**, while **httpsonly** is **true**, then a ‘401 Unauthorized’ status is returned.

11.1.3 <option name="ClientCertRequired" ...

By default the RESTCONF Access Agent will allow requests without a client certificate being provided. The following option is used to enable client certificate checking.

```
<agent name="rc" dll="libndtaarestconf.so.0">
 <options>
     <option name="ClientCertRequired" value="true"></option>
 </options>
</agent>
```

When the client sends a certificate and this option is "true", RESTCONF will require that the web server received a valid certificate to process the request, otherwise an ‘401 Unauthorized’ status is returned.[28]

If the client sends a certificate and this option is "false" and the web server received an invalid certificate to process the request, then an ‘401 Unauthorized’ status is also returned.

11.1.4 <option name="AuthRequired" ...

By default the RESTCONF Access Agent will allow unauthenticated client requests. The following option is used to require web server authentication be enabled.

```
<agent name="rc" dll="libndtaarestconf.so.0">
 <options>
     <option name="Authrequired" value="true"></option>
 </options>
</agent>
```

27 ‘**httpsonly**’, ‘**authrequired**’, ‘**clientcertrequired**’ and ‘**passwebusertoNACM**’ **SHOULD** all be **ENABLED**, when deployed in an operational environment. The defaults should be considered for development and test purposes **only**.

28 See the httpd and lighttpd configuration sections for more information.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

Many methods for user authentication are possible with most web servers, the one you choose must provide the **FCGI** interface with that user's name in the variable '**REMOTE_USER**' and the variable '**REMOTE_PASSWD**'. If it does not provide this then you can use basic authentication, in which case the web server must provide the variable **HTTP_AUTHORIZATION**[29] with a 'Basic' authorization string.

Tip: You can look through the log file for the string '<WEB> AAWeb::HttpRequest::HttpRequest()'. What follows this is all the information a web server provides the RESTCONF access agent with each FCGI transaction. Note: you need to set the logLevel to at least '3' for see this information in a log.

11.1.5 <option name="PassWebUsertoNACM" ...

By default the RESTCONF Access Agent has the single user, "web" associated with it. This is used for processing requests to the data server, so when the NACM is enabled, it is used as the NACM user name for the request. If the user chooses, the authenticated web user name can be used for this purpose. This option is used to enable using the logged in web user name for the purposes of accessing the data server via the NACM mechanism. In other words, when enabled, there must be a NACM user configured that corresponds to the authenticated web user. Using this option allows for configuring multiple NACM profiles for RESTCONF access.

```
<agent name="rc" dll="libndtaarestconf.so.0">
 <options>
     <option name="PasswebusertoNACM" value="true"></option>
 </options>
</agent>
```

11.1.6 <option name="UseClientCertCN" …

When 'PassWebUserToNACM' is enabled, you may use this option to have RESTCONF use the client's CN as a user name to pass to the NACM, when a '**VALID**' client certificate has been encountered by the web server, regardless of that which could be provided by other authentication mechanisms. (E.g. RADIUS) The default is to not use this information.

11.1.7 <option name="DisableFlushToStartup" …

Use this option to have RESTCONF automatically flush changes to :startup. When set to "true" this feature is disabled. Default is "false".

```
<agent name="rc" dll="libndtaarestconf.so.0">
 <options>
     <option name="DisableFlushToStartup" value="true"></option>
 </options>
</agent>
```

11.1.8 <option name="FlushtoStartupPeriod" ...

Assuming that flushing hasn't been disabled (see above), when writes to the configuration are made through RESTCONF, the changed data is updated to the :startup datastore. This option controls how long after the last write that the update occurs. The default period is 5 seconds. When writes occurs prior to the previous period expiring, the expiry time is reset to this value.

```
<agent name="rc" dll="libndtaarestconf.so.0">
 <options>
     <option name="FlushtoStartupPeriod" value="10"></option>
 </options>
</agent>
```

29A configuration modification may be required to enable the webserver to pass **HTTP_AUTHORIZATION**. See the Apache/httpd section.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

11.1.9 <option name="UseCandidateAndCommit" …

By default, RESTCONF writes directly to the :running datastore. You can change this behavior with this option. Enabling this option to “true” will cause the transaction to first be written to the :candidate datastore, then it will commit the changes to running.

11.1.10 <option name="NotifSocket" ...

By default, the streaming/notification services of the RESTCONF access agent uses **tcp port 9524**, to communicated with the stream pipe process, ‘ndrcstream’, that is launched by the web server on a per stream basis for pushing notifications. If for some reason this port is unavailable, you can change the port. This is done in the configuration file for the **testagent** (or **ndconfsvr**). In **ndconf.xml**, change the line:

```
<agent name="rc" dll="libndtaarestconf.so.0"></agent>
```

to include ‘options’ and specify the ‘*option*’ named “*notifsocket*” and provide the value of the port you wish to use. E.g.

```
<agent name="rc" dll="libndtaarestconf.so.0">
    <options>
        <option name="notifsocket" value="9525"></option>
    </options>
</agent>
```

11.1.11 <option name="CalcEtagForNonConfig" …

By default, RESTCONF only provides an Etag response header field for GET and HEAD operations when the “content” query parameter has the value “config”. If the value for “CalcEtagForNonConfig” is true then an Etag is provided for any GET or HEAD request, independent for whether there is “content” query parameter or it’s value. E.g.

```
<agent name="rc" dll="libndtaarestconf.so.0">
    <options>
        <option name="CalcEtagForNonConfig" value="true"></option>
    </options>
</agent>
```

11.1.12 <option name="AllowPartialReadKeys" …

By default, RESTCONF requires that when key(s) for a list are being provided on the URI, values for all key(s) for the list must be provided. This option relaxes that requirement for GET (and HEAD) such that you can specify fewer. The data set returned will include all entries of the list for the missing key(s), given the values of the provided keys. E.g.

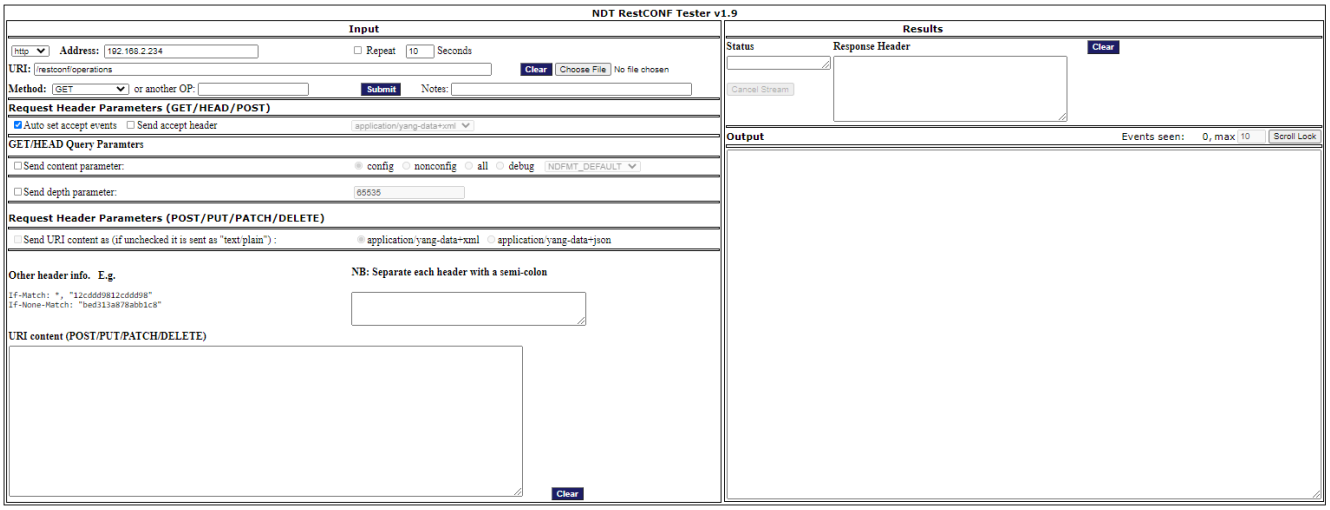
```
<agent name="rc" dll="libndtaarestconf.so.0">
  <options>
      <option name="AllowPartialReadKeys" value="true"></option>
  </options>
</agent>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

11.2 Using ‘restconf-submit-test.html’

To view, access either of the named pages via the web server, using most any current web browser. This will necessitate copying restconf-submit-test.html to the web server’s file space. Consult the web server’s documentation as to where that location is.

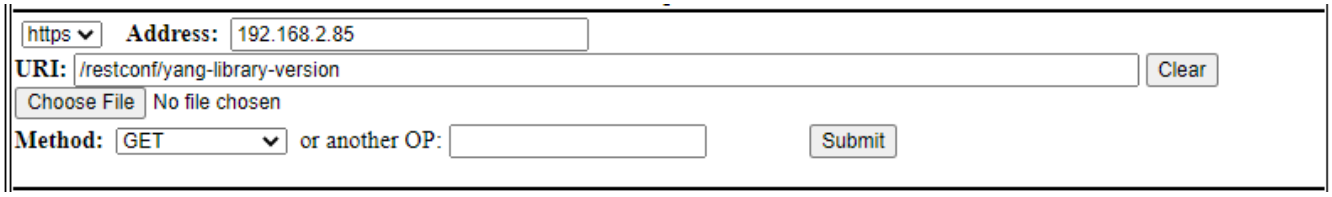
Once loaded you will see something like the following. The application is broken into the left (Input) and right (Result) sections.



The left side is where all data for a request is entered (‘Input’) and the right side where the result (’Results’) will be displayed.

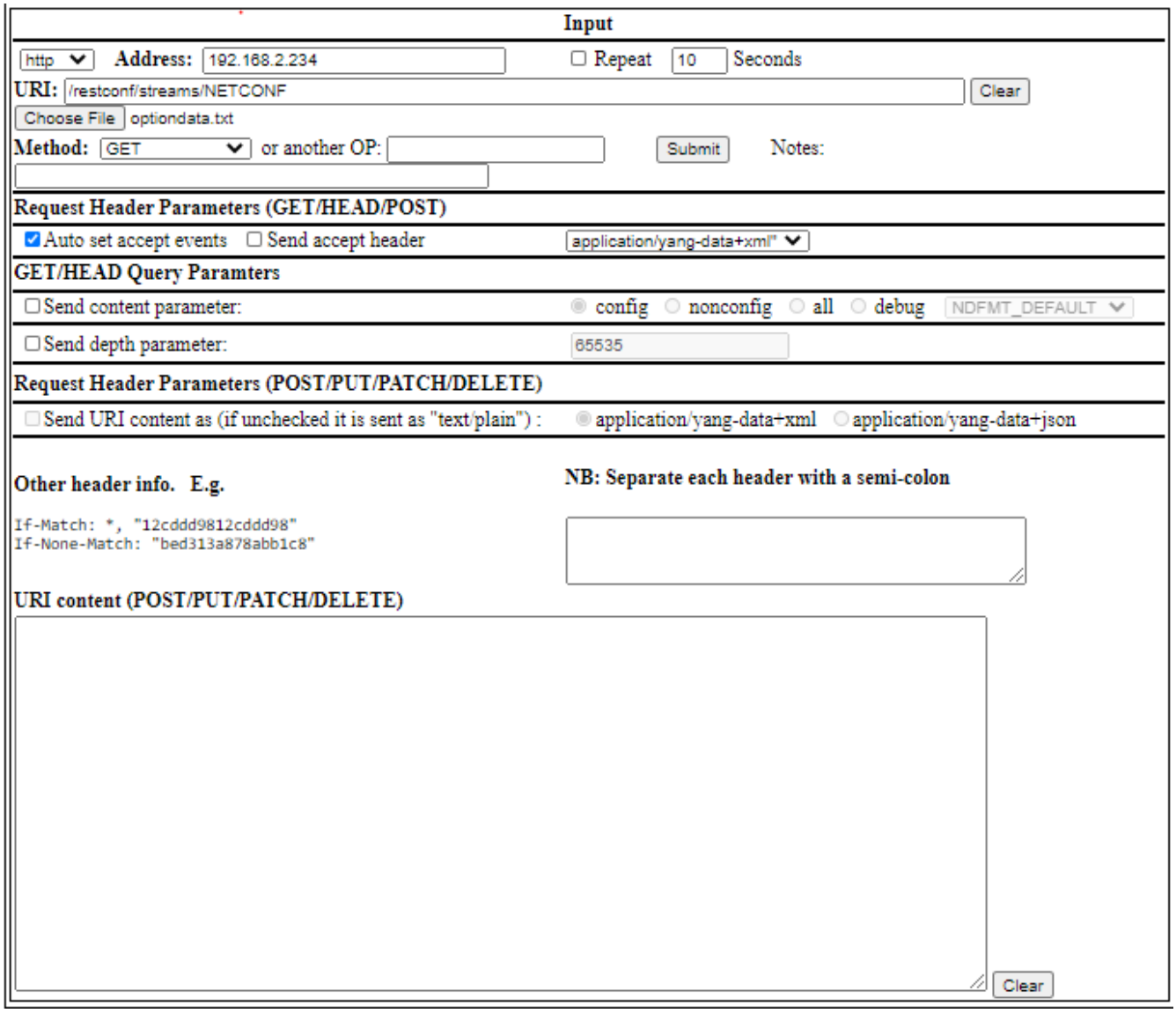
11.2.1 The ‘Input’ Section

The top section looks something list the following when loaded from the web server.



This is where the primary characteristics of the request are setup and the ‘Submit’ button, which is used to send the request, reside.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



The upper left of this section allows for selecting either “**http**” or “**https**”. The default is ‘**http**’. ‘**https**’ is only usable if the web server has been configured to support it. In normal deployment, ‘**https**’ should be configured and ‘http’ disallowed for RESTCONF requests, due to the typically sensitive nature of the data.

‘**Address**’ displays the current address of the RESTCONF server. If the page is loaded from the local file system directly (restconf-submit-test only), you can specify the address of the RESTCONF server.

‘**URI**:’ is where you can select or specific the RESTCONF URI for the request[30].

‘**Clear**’ is used to clear the current URI. This is useful enable viewing all options.

‘**Choose File**’ is used to select a local (to the browser) file to add additional or replace options in the **URI**: drop down list.

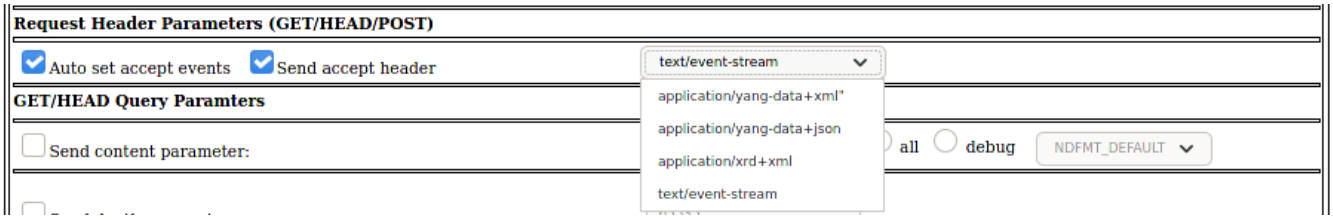
30 Some of the options provided are specific to particular datadlls. In particular, ietf-openconfig-interfaces and ietf-interfaces. These options only produce results when the corresponding datadll is loaded.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

‘**Method**’ is used to specify one of GET, HEAD, OPTIONS, POST, PUT, PATCH, DELETE[31].

‘Repeat’ is used to cause the page to automatically refresh when checked by the number of ‘**Seconds**’ specified.

The next section relates to setting of the Accept header.



By default, the request is sent with the Accept header set to “*/*”. To enable one of the listed choices, you must check “Send accept header for:”, in which case the request is sent with the selected option from the drop down.

If you select “Auto set accept events” (as shown), then when the URI contains the prefix “/restconf/streams”, the accept header will automatically be set to ‘text/event-steam’ when “Submit” is pressed.

Note: This then causes the page to establish a connection to the web server and wait for data. Additionally the “Scroll Lock” mode is initiated.

The next two sections related to sending one of two specific query parameters, ‘content’ and/or ‘depth’. In both cases, you need to check the option to enable it.

Note: the ‘debug’ option in the ‘content’ is an internal only option and is not part of the RESTCONF specification.

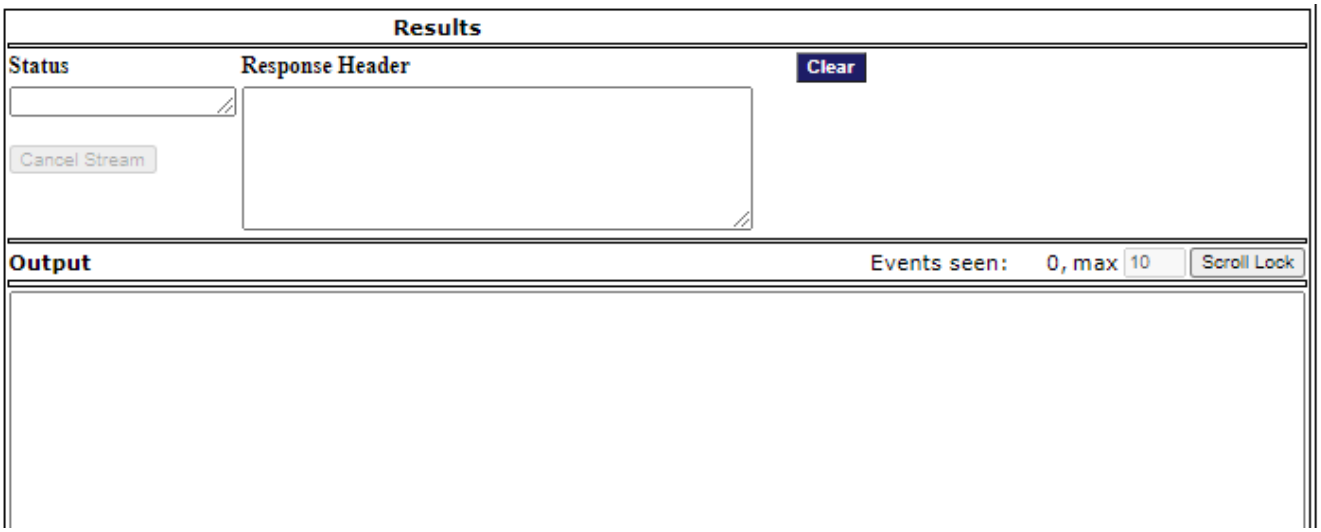
The next section relates the format of content in the “URI content ...” section. To enable the selection of format to either XML or JSON, you must check the “Send URI Content …” check-box.

11.2.2 The ‘Results’ Section

This section provides the **Status**, **Response Header** and the **Output** of the request and are self explanatory.

31 See the ‘**Current Implementation Limitations**’ section.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



When the page is in the event streaming mode, the “Cancel Stream” button will be enabled and the “Submit” button is disabled. When “Cancel Stream” is pressed, it changes to display “Cancelling” and will terminate the current stream on the next reception of a notification. The “Cancel Stream” button is then disabled and the “Submit” button is enabled again. If the server end terminates the connection, the “Cancel Stream” button is also disabled as is the “Submit” enabled.

You should note that browsers have limits to the amount of data that can be contained in the Output text area. Over time, the browser may slow down and eventually become unresponsive. To avoid this, the page automatically invokes scroll lock when an event stream is initiated to limit the amount of data. If you turn off scroll lock, then you can press the “Clear” button to clear the current data to avoid this problem.

The “Scroll Lock” button locks the bottom of the output to the bottom of the “Output” display area and changes to “Scroll Unlock”. As noted above, this mode is initiated automatically when a stream is initiated. When in the “Scroll Lock” mode, the application deletes the data that is out of view so the problem mention above does not manifest, hence you’ll always be able to keep the most recent notifications in view. The number of items retained in the output is based on the value (10) in the “max” box. This can be modified after the stream is initiated. The “Events seen:” area reports the total number of events that have been received.

11.2.3 Examples of Using NETCONF RPCs

The following are examples of invoking some of the NETCONF defined RPCs from restconf-submit-test.html. **Note**: any output from an RPC will be embedded inside <output></output> tags. See RFC 6241 for more information on these RPCs.

- URI: /restconf/operations/ietf-netconf:get
  Method: POST
  URI Content:

```
<input xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <filter type="subtree" xmlns:g="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
    <g:garage></g:garage>
  </filter>
</input>
```

  In this example, this “get” RPC will return for contents of the <garage> using a subtree filter.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

- URI: /restconf/operations/ietf-netconf:get
  Method: POST
  URI Content:

```
<input xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <filter type="xpath" xmlns:ndg="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB"
     select="//ndg:garage" />
</input>
```

  In this example, this “get” RPC will also return for contents of the <garage> using a xpath filter.

- URI: /restconf/operations/ietf-netconf:get-config
  Method: POST
  URI Content:

```
<input xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <source>
    <running></running>
  </source>
  <filter type="subtree" xmlns:g="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
    <g:garage></g:garage>
  </filter>
</input>
```

  In this example, this “get-config” RPC will return for contents of the <garage> using a subtree filter from the :running datastore.

- URI: /restconf/operations/ietf-netconf:get-config
  Method: POST
  URI Content:

```
<input xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <source>
    <startup></startup>
  </source>
  <filter type="subtree" xmlns:g="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
    <g:garage></g:garage>
  </filter>
</input>
```

  In this example, this “get-config” RPC will return for contents of the <garage> using a subtree filter from the :startup datastore.

- URI: /restconf/operations/ietf-netconf:copy-config
  Method: POST
  URI Content:

```
<input xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <source><running/></source>
   <target><startup/></target>
</input>
```

  In this example, this “copy-config” RPC will copy the :running datastore to the :startup datastore.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

- URI: /restconf/operations/ietf-netconf:commit
  Method: POST
  URI Content:

```
<input xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <confirmed/>
  <persist>IQ,d4668</persist>
</input>
```

  In this example, this “commit” RPC will initiate a persistent confirmed commit[32] with the identifier “IQ,d4668”.

- URI: /restconf/operations/ietf-netconf:commit
  Method: POST
  URI Content:

```
<input xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <persist-id>IQ,d4668</persist-id>
</input>
```

  In this example, this “commit” RPC will “confirm” the persistent confirmed commit, with identifier “IQ,d4668”.

- URI: /restconf/operations/ietf-netconf:cancel-commit
  Method: POST
  URI Content:

```
<input xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <persist-id>IQ,d4668</persist>
</input>
```

  In this example, this “cancel-commit” RPC will cancel the persistent confirmed commit, with identifier “IQ,d4668”.

11.2.4 RESTCONF Notifications

Notifications are received from RestCONF via an SSE encoded event stream. These streams are available under the ‘/restconf/streams’ URI path. All streams receive the same event notification and differ only in presentation. The following streams are available.

- /restconf/streams/NETCONF - XML formatted
- /restconf/streams/NETCONF-JSON - JSON formatted
- /restconf/streams/NETCONF-TEXT - text formatted

When the SNMP Access agent is load, the following are available:

- /restconf/streams/SNMP - XML formatted
- /restconf/streams/SNMP-JSON - JSON formatted
- /restconf/streams/SNMP-TEXT - text formatted

32 This example only performs anything when the RESTCONF AA feature “UseCommitAndConfirm” is set to “true”.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

When the SYSLOG data dll is loaded, the following are available:

- /restconf/streams/SYSLOG - XML formatted
- /restconf/streams/SYSLOG-JSON - JSON formatted
- /restconf/streams/SYSLOG-TEXT - etxt formatted

Each notification sent by RESTCONF is initiated to an SSE “event”. The event name for this is ‘’start’ and the data provided with the event is “***NDT Notification***”.

The data for the event is sent encoded in a SSE “data” sequence.

Note: you must select the “Send accept header for” with the “text/event” option for reception of event streams.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

11.3 Using curl

'curl' is a command line tool available on Linux. You may optionally access the RESTCONF server using this tool. Below is a list of some curl command examples that correlate closely to the options provided in the web applications (above).

```
curl http://127.0.0.1/.well-known/host-meta

curl http://127.0.0.1/restconf/

curl http://127.0.0.1/restconf/data

curl http://127.0.0.1/restconf/yang-library-version

curl http://127.0.0.1/restconf/data/ND-GARAGE-MIB:garage

curl http://127.0.0.1/restconf/data/ND-GARAGE-MIB:garage/vehiclesEntry

curl http://127.0.0.1/restconf/data/ND-GARAGE-MIB:garage/vehiclesEntry=2

curl http://127.0.0.1/restconf/data/ND-GARAGE-MIB:garage/vehiclesEntry=2/vehicleModel

curl 'http://127.0.0.1/restconf/data/ND-GARAGE-MIB:garage/vehiclesEntry=1?
fields=vehicleIndex;vehicleModel'

curl http://127.0.0.1/restconf/data/openconfig-interfaces:interfaces/interface

curl 'http://127.0.0.1/restconf/data/openconfig-interfaces:interfaces/interface?fields=config/
name;state/counters/in-octets'

curl 'http://127.0.0.1/restconf/data/openconfig-interfaces:interfaces/interface?
fields=config(name;mtu)'

curl http://127.0.0.1/restconf/data/ietf-interfaces

curl http://127.0.0.1/restconf/data/ietf-interfaces:interfaces/interface

curl http://127.0.0.1/restconf/data/ietf-interfaces:interfaces/interface=lo

curl 'http://127.0.0.1/restconf/data/ietf-yang-library:modules-state?
fields=module(name;revision)'

curl http://127.0.0.1/restconf/data/ietf-netconf-monitoring:netconf-state/capabilities -H
"Accept:application/yang-data+xml"

curl http://127.0.0.1/restconf/data/ietf-netconf-monitoring:netconf-state/capabilities -H
"Accept:application/yang-data+json"

curl http://127.0.0.1/restconf/data/ietf-netconf-monitoring:netconf-state/capabilities?depth=1
-H "Accept:application/yang-data+xml"

curl http://127.0.0.1/restconf/data/ietf-netconf-monitoring:netconf-state/capabilities?depth=1
-H "Accept:application/yang-data+json"
```

The following is a 'POST' curl sequence that will add a row to the 'vehiclesEntry' table of the ndGarageMib datadll. Replace "<username:password>" below with something appropriate. (Note these examples are single command lines)

```
curl -i -u <username:password> -h "accept: application/yang-data+xml" -h "content-type:
text/xml" -x POST --data-urlencode ="<vehiclesentry xmlns=\"urn:ietf:
params:xml:ns:yang:smiv2:nd-garage-mib\"><vehicleindex>4</vehicleindex><vehiclelicenceplate>777
swe</vehiclelicenceplate><vehiclemodel>koenigsegg one:1</vehiclemodel></vehiclesentry>"
 https://127.0.0.1/restconf/data/nd-garage-mib:garage
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

The following is a ‘PUT’ curl sequence that will change the values of the entries in the same row of the ‘vehiclesEntry’ table of the ndGarageMib datadll.

```
curl -i -u <username:password> -h "accept: application/yang-data+xml" -h "content-type:
text/xml" -x PUT --data-urlencode ="<vehiclesentry xmlns=\"urn:ietf:
params:xml:ns:yang:smiv2:nd-garage-mib\"><vehicleindex>4</vehicleindex><vehiclelicenceplate>888
ENG</vehiclelicenceplate><vehiclemodel>Camry</vehiclemodel></vehiclesentry>”
 https://127.0.0.1/restconf/data/nd-garage-mib:garage
```

The following is a ‘DELETE’ curl sequence that will delete the row.

```
curl -i -u <username:password> -H "Accept: application/yang-data+xml" -H "Content-type:
text/xml" -X DELETE https://127.0.0.1/restconf/data/ND-GARAGE-MIB:garage/vehiclesEntry=4
```

11.4 Current Implementation Status & Limitations

The RESTCONF implementation currently has the following limitations.

- HEAD, GET, POST, PUT, PATCH (plain), OPTIONS & DELETE methods are supported.
- Yang Patch (RFC 8072) is implemented.
- The ‘operations’ resource is implemented. i.e. you can invoke RPCs.
- Actions are implemented.
- The ‘depth’ query parameter is partially implemented.
- The ‘field’ query parameter is partially implemented.
- The ‘insert’ and ‘point’ operations are supported for POST and PUT.
- JSON input is implemented for methods sending content for an operation.
- The ietf-restconf-state module is implemented.
- notification streams with notification query parameters ‘filter’, ‘start-time’, ‘stop-time’ are implemented.
- ‘If-Match’ & ‘If-None-Match’ are implemented for all operation..
- ‘If-Modified-Since’ & ‘If-Unmodified-Since’ are not implemented.
- The notification query parameters ‘with-defaults’ is partially implemented.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

12 CLI

This section describes commands implemented by the CLI Access Agent (libndtaacli.so).

Type ? to see a list of the available commands:

```
> ?
Possible completions:
 about
 accessagent
 cls
 cmpcfg
 configure
 datadll
 exit
 help
 quit
 script
 session
 show
 sleep
>
```

12.1 Command completion

To complete a partial command enter <tab> immediately after a command prefix, e.g.

```
> c[Tab]
Ambiguous command: c
Possible completions:
 cls
 cmpcfg
 configure
> c
```

In the example above, there are 3 commands that start with prefix "c". List of possible completions is printed along with the error message "Ambiguous command". New command line is extended to the first 'non-ambiguous' letter (in this case new cmd line is not extended as "c" is already the longest common prefix for these 3 commands) and cursor positioned immediately after it.

Type next few letters and [Tab] :

```
> conf[Tab]
```

and command will be completed:

```
> configure
```

If there is no command that starts with prefix:

```
> cond[Tab]
Couldn't complete, no match found
> cond
```

12.2 Commands and parameters

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

12.2.1 ?

?
<prefix>?

List all commands or just commands that starts with <prefix>. E.g.

```
> s?
Possible completions:
 script
 session
 show
 sleep
>
```

12.2.2 about

about

Show **testagent** information:

```
> about[Enter]
************************************************
* NuDesign YANG Datastore Test Agent
* Version x.x.x.x
* Copyright 2014-2025 NDT
************************************************
>
```

12.2.3 accessagent

accessagent [load | loadAndStart | start | stop | unload] <name>

Perform one of the action on access agent. Argument <name> is the name attribute for agent element in "ndconf.xml":

```
    <agents>
        <agent name="nc" dll="libndtaanc.so.0" ></datadll>
          ...
```

```
> show agents[Enter]
  nc                                <<-- name attribute from ndconf.xml
      path libndtaanc.so.0
      name netconf                  <<-- name property of the access agent
      version 1.5.0.0
      status running
...
>
> accessagent unload nc[Enter]
  ok
> show agents[Enter]
  nc
      path libndtaanc.so.0
      status NOT LOADED
...
> accessagent load nc[Enter]
  ok
> show agents[Enter]
  nc
      path libndtaanc.so.0
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
      name netconf
      version 1.5.0.0
      status loaded
...
> accessagent start nc[Enter]
  ok
> show agents[Enter]
  nc
      path libndtaanc.so.0
      name netconf
      version 1.5.0.0
      status running
...
```

NOTE: access agent can not be unloaded from the session handled by the target of unload. E.g. executing from cli:

```
> accessagent unload cli
```

12.2.4 cls

cls

Clear the screen.

12.2.5 cmpcfg

cmpcfg <candidate|running|startup|url> <candidate|running|startup|url>

Compare configurations.

Please refer to the “diff” command for information on how to interpret the screen output of **cmpcfg**.

```
> cmpcfg startup running[Enter]

left: startup
right: running
```

12.2.5.1 configure

configure

Enter configuration mode (indicated by [edit] line above prompt, prompt changes to #)

```
> configure[Enter]
Candidate configuration is now locked.
[edit ]
#
```

See "Configuration Mode" section for details.

12.2.6 datadll

datadll [load | unload] <name>

Load or unload data dll. Argument <name> is the name attribute for datadll element in "ndconf.xml":

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
    <datadlls>
        <datadll name="garage" dll="Garage.dll" ></datadll>
```

Argument <name> is also displayed in the output of "show datadll":

```
> show datadll[Enter]
  data
    dll
      name if
      path <SDK installdir>/ndconf/lib64/libndGarageMib.so.0
      loaded true
>
> datadll unload if[Enter]
  ok
> datadll load if[Enter]
  ok
>
```

NOTE: Loading or unloading causes a "netconf-capability-change" notification. Also note, that if there is an interdependency between datadlls and you try to unload the dependent one, you'll receive the following error message:

```
  <rpc-error>
     <error-type>application</error-type>
     <error-tag>internal-error</error-tag>
     <error-severity>error</error-severity>
     <error-app-tag>general-error</error-app-tag>
     <error-message>Module: imported in other module</error-message>
  </rpc-error>
```

12.2.7 exit

exit

In operational mode, exit the **testagent**.

```
> exit
disconnected
[a-user@localhost testagent]$
```

In configuration mode, "exit" command moves the current position one level up. If at the root of configuration tree, exit configuration mode.

```
[edit ]                                          <--- /
# edit garage vehiclesEntry 1[Enter]
[edit garage vehiclesEntry "1" ]                 <--- /garage/vehiclesEntry[vehiclesIndex=1]
# exit[Enter]
[edit garage ]                                   <--- /garage
# exit[Enter]
[edit ]                                          <--- /
# exit[Enter]
>                                                <--- op mode
```

12.2.8 help

help <cmdname>

Print help for command specified in argument.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
> help configure[Enter]
Enter configuration mode
>
```

12.2.9 quit

exit the agent.

12.2.10 script

script <file>

Execute a script file.

12.2.11 session

session

Display CLI session info

```
> session[Enter]
 SESSION: user:cli, id:1, status:connected
>
```

12.2.12 show

show [agents | datadll | data | history | namespaces | remaining | running]

Display selected information.

12.2.12.1 show agents

Display Access agents status

```
> show agents[Enter]
cli
   path: libndtaacli.so.0
   name: cli
   version: 1.6.0.0
   status: OK
nc
   path: libndtaanc.so.0
   name: netconf
   version: 1.4.0.0
   status: OK
snmp
   path: libndtaasnmp.so.0
   name: snmp
   version: 1.5.1.0
   status: OK
rc
   path: libndtaarestconf.so.0
   name: restconf
   version: 0.9.2.0
   status: OK
>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

12.2.12.2 show data

Display all data (equivalent to netconf 'get')

```
> show data[Enter]
 nc:data
   snmp:snmp
   snmp:engine
...
> show data xpath //ncm:sessions[Enter]
 nc:data
   ncm:sessions
     ncm:session
       ncm:session-id 1
...
>
```

12.2.12.3 show datadll

Display list of configured data dlls with 'loaded' status

```
> show datadll[Enter]
  data
    dll
      name if
      path /home/a-user/ndconf/lib64/libietfInterfaces.so.0
      loaded true
   dll
     name yanglib
     path libietfYangLibrary.so.0
     loaded true
    dll
      name garage
      path /home/a-user/ndconf/ndGarageMib/Linux/libndGarageMib.so
      loaded true
>
```

12.2.12.4 show history

Display command line history.

```
> show history[Enter]
0: about
1: show running
>
```

12.2.12.5 show namespaces

Display all namespaces

```
> show namespaces[Enter]
 ndtmg:data
   namespace
     prefix y
     uri urn:ietf:params:xml:ns:yang:1
       :
       :
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
>
```

12.2.12.6 show remaining

(Applies to evaluation version only) Display remain time in the evaluation period.

```
> show remaining[Enter]
Remaining evaluation time (dd:hh:mm:ss): 28:01:00:56
>
```

12.2.12.7 show running

Display running configuration (equivalent to netconf 'get-config running')

```
> show running[Enter]
// running //
 snmp
    engine
      enabled true
      listen
        name lis-1
        udp
          ip 127.0.0.1
          port 4161
      version
        v1
        v2c
        v3
      engine-id 80:00:12:99:04:6e:64:74
...
>
```

12.2.13 sleep

sleep <seconds>

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

12.3 Configuration Mode

Configuration mode commands operate on configuration data. To enter configuration mode execute the configure command. The commands available in configuration mode at any position are:

```
[edit ]
# ?
Possible completions:
 commit
 create
 delete
 diff
 edit
 exit
 help
 load
 quit
 run
 save
 set
 show
 top
 up
[edit ]
#
```

Commands operate on the config data available at the current position, see edit, up, top, exit. On entering configuration mode, the current position is set to root. Excerpts from testagent configuration to be used in examples:

```
    <config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <garage xmlns="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
  ...
        <snmp xmlns="urn:ietf:params:xml:ns:yang:ietf-snmp">
  ...
```

Possible arguments for the "set" command at root level are:

```
[edit
# set ?
Possible completions:
 <[Enter]>           Execute this command
 garage
 interfaces
 mib-2.snmp
 nacm
 snmp
 system
[edit ]
# set
```

Change position to "garage"

```
[edit ]
# edit garage[Enter]
[edit garage ]
#
```

Possible args for "set" command at this level are:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
[edit garage ]
# set ?
Possible completions:
 <[Enter]>           Execute this command
 garageObjects
 vehiclesEntry
[edit garage ]
# set
```

12.3.1 commit

commit

Applies changes in candidate configuration to running.

```
# commit[Enter]
  ok
#
```

12.3.2 create

create arg1 [arg2] [...]

Create configuration data (corresponds to edit-config, operation="create").

```
[edit garage]
# create vehiclesEntry 5[Enter]
  ok
[edit garage ]
# show[Enter]
// candidate //
...
    vehiclesEntry
      vehicleIndex 1
...
    vehiclesEntry
      vehicleIndex 2
...
    vehiclesEntry
      vehicleIndex 5
[edit garage ]
#
```

Sometimes an entry can have multiple keys. In these cases use the '|' character as the separator(s) between keys.

```
# create workerEntry 101|David

or if there are spaces in the keys.

# create workerEntry “201|John Smith Jr.”
```

12.3.3 delete

delete arg1 [arg2] [...]

Delete configuration data.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
[edit garage ]
# delete vehiclesEntry 2[Enter]
  ok
[edit garage ]
# show[Enter]
// candidate //
...
    vehiclesEntry
      vehicleIndex 1
...
    vehiclesEntry
      vehicleIndex 5
...
[edit garage ]
#
```

12.3.4 diff

diff [candidate running | candidate <file> | running candidate | running <file>]

Perform a difference operation. Shows difference tree: '*' indciates that the node value is different (shown value is from the “left” datastore), '+' indicates added nodes (they do not exist in the “right” datastore), '-' indicates that the nodes are deleted (they exist in the “right” datastore)

```
[edit ]
# diff candidate running[Enter]
 xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
config
 xmlns="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB"
  garage
    garageObjects
      * garageAddress 10 street            <--- *  different value
    + vehiclesEntry                        <--- +  added node
      + vehicleIndex 5
    - vehiclesEntry                        <--- -  deleted node
      - vehicleIndex 2
      - vehicleLicencePlate abcd345
      - vehicleModel Jaguar F-Type
```

12.3.5 edit

edit arg1 [arg2] [...]

[edit] line shows the current position in configuration hierarchy. To navigate configuration hierarchy use **edit**, **up** and **top**.

```
[edit]
# edit garage vehiclesEntry 1[Enter]
[edit garage vehiclesEntry "1" ]              <--- /garage/vehiclesEntry[vehiclesIndex=1]33
# show[Enter]
// candidate //
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
[edit garage vehiclesEntry "1" ]              <--- /garage/vehiclesEntry[vehiclesIndex=1]
```

33 As outlined in “**create**” above, when providing key information for a list that has multiple keys, you need to separate the keys with a vertical bar symbol (‘|’). For example, “this|that” could be the keys “this” and “that” for a two key list. Also note, that the quotes around “this|that” are only required if there is a space in a key. E.g. “this one|that” would require quotes.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
# up[Enter]
[edit garage ]                                  <--- /garage
# show[Enter]
// candidate //
    garageObjects
      garageAddress 10 Street
...
[edit garage ]                                  <--- /garage
# up[Enter]
[edit ]                                         <--- /
# edit garage vehiclesEntry 1[Enter]
[edit garage vehiclesEntry "1" ]                <--- /garage/vehiclesEntry[vehiclesIndex=1]
# top[Enter]
[edit ]                                         <--- /
#
```

12.3.6 exit

exit [discard]

Exit configuration mode if the current position is configuration root. Otherwise move one level up (the same as 'up' command)

```
[edit ]
# exit[Enter]
>
```

If there are uncommitted changes made to candidate configuration

```
[edit ]
# exit[Enter]
There are uncommitted changes.
Use 'commit' to commit the changes, or 'exit discard' to discard them.
[edit ]
# exit discard[Enter]
>
```

`exit discard` executed at any level is the same as `quit`.

12.3.7 load

load <file> candidate

Load configuration from <file> into the candidate configuration.

```
[edit ]
# load newcfg.xml candidate[Enter]
  ok
[edit ]
#
```

12.3.8 quit

quit

Exit configuration mode discarding changes (if any), the same as 'exit discard'.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
[edit ]
# exit[Enter]
There are uncommitted changes.
Use 'commit' to commit the changes, or 'exit discard' to discard them.
[edit ]
# quit[Enter]
>
```

12.3.9 run

run arg1 [arg2] [...]

Execute system command.

```
[edit ]
# run ping 192.168.0.1[Enter]
Executing 'ping 192.168.0.1'
Press Ctrl-C to interrupt

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64
...
Approximate round trip times in milliseconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
[edit ]
#
```

12.3.10 save

save candidate|running <file>

Save running or candidate configuration to file.

```
[edit ]
# save candidate test[Enter]
  ok
[edit ]
#
```

12.3.11 set

set arg1 [arg2] [...]

Modify configuration data (corresponds to edit-config, operation="replace").

```
[edit ]
# show garage garageObjects garageAddress[Enter]
// candidate //
      garageAddress 10 Street                                    <-- current address
[edit ]
# set garage garageObjects garageAddress "123 avenue rd."[Enter]   <-- change address
  ok
[edit ]
# show garage garageObjects garageAddress[Enter]
// candidate //
      garageAddress 123 avenue rd.                               <-- new address
[edit ]
#
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

12.3.12 show

show [running] [arg1] [arg2] [...]

Display configuration. By default displays candidate configuration data.

```
[edit ]
# show[Enter]
// candidate //
  snmp
    engine
      enabled true
      listen
        udp
...
```

To display running configuraton:

```
[edit ]
# show running[Enter]
// running //
  snmp
    engine
      enabled true
      listen
        udp
...
```

Argument can be any node that is direct child of the current position.

```
# show ?
Possible completions:
 <[Enter]>           Execute this command
 garage
 mib-2.snmp
 history
 running
 snmp
 system
[edit ]
# show running ?
Possible completions:
 <[Enter]>           Execute this command
 garage
 mib-2.snmp
 snmp
 system
[edit ]
# show running
```

```
[edit ]
# show garage[Enter]
// candidate //
    garageObjects
      garageAddress Avenue road
[edit ]
#
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

12.3.13 top

top

Change current position to configuration root

See ***edit***

12.3.14 up

up

Change current position one level up

See ***edit***

12.4 Configuring libndtaacli.so.0

When **libndtaacli.so.0** is started in the background, as in when it is started from a daemonized ndconfsvr, it opens a service port to allow access to in in the background from **ndcliclient**. There is an option to allow changing this CLI server tcp port, from the default of 9523. To do so, in **ndconf.xml**, change the line:

```
<agent name="cli" dll="libndtaacli.so.0"></agent>
```

to include ‘options’ and specify the ‘option’ named “socket” and provide the value of the port you wish to use.

E.g.

```
<agent name="cli" dll="libndtaacli.so.0">
 <options>
     <option name="socket" value="9524"></option>
 </options>
</agent>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

13 `ncclient` - NETCONF client

`ncclient` is the command line NETCONF client. To start, from **`<SDK installdir>/ndconf/ncclient`** type **`./ncclient`** in a console:

```
**************************************************
 NuDesign NetConf CALL HOME Client
 Version 2.0.0.0
 Copyright (c) 2014-2025 NuDesign Technologies Inc.
**************************************************
>
```

The UI is similar to CLI . Type ? (question mark) to see the list of available commands, prior to being connected:

```
> ?
Possible completions:
 about
 cls
 connect
 exit
 help
 list
 nc
 now
 quit
 redirectqueue
 script
 show
 sleep
>
```

After the client is connected the following commands are available:

```
2:T chadmin@127.0.0.1> ?
Possible completions:
 about
 access-agent
 cancel-commit
 close-session
 cls
 commit
 copy-config
 create-subscription
 datadll
 delete-config
 delete-subscription
 discard-changes
 disconnect
 edit-config
 edit-data
 establish-subscription
 exit
 get
 get-access-agents
 get-config
 get-data
 get-data-dlls
 get-namespaces
 get-schema
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
 help
 kill-session
 kill-subscription
 list
 lock
 modify-subscription
 nc
 now
 partial-lock
 partial-unlock
 quit
 redirectqueue
 rpc
 script
 show
 sleep
 unlock
 validate
```

13.1 ncclient Commands and parameters

13.1.1 ?

?
<prefix>?

Lists all commands or just commands that start with <prefix>. E.g.

```
> c?
Possible completions:
 configure
 connect
> c
```

13.1.2 about

about

Show **ncclient** information:

```
> about[Enter]
*****************************************************
 NuDesign NetConf CALL HOME Client
 Version 2.0.0.0
 Copyright (c) 2014-2025 NuDesign Technologies Inc.
*****************************************************
>
```

13.1.3 access-agent

access-agent [load | loadAndStart | start | stop | unload] <name>

Perform one of the action on access agent. Argument <name> is the name attribute for agent element in "ndconf.xml":

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
    <agents>
        <agent name="snmp" dll="libndtaasnmp.so.0" ></datadll>
          ...
```

```
1:T admin@127.0.0.1> get-access-agents[Enter]
  data
    dll
      name snmp
      path libndtaasnmp.so.0
      version 1.5.0.0
      status running
...
1:T admin@127.0.0.1>
1:T admin@127.0.0.1> access-agent unload snmp[Enter]
  ok
1:T admin@127.0.0.1> get-access-agents[Enter]
  data
    dll
      name snmp
      path libndtaasnmp.so.0
      version
      status NOT LOADED
...
1:T admin@127.0.0.1> access-agent load snmp[Enter]
  ok
1:T admin@127.0.0.1> get-access-agents[Enter]
  data
    dll
      name snmp
      path libndtaasnmp.so.0
      version 1.5.0.0
      status loaded
...
1:T admin@127.0.0.1> access-agent start snmp[Enter]
  ok
1:T admin@127.0.0.1> get-access-agents[Enter]
  data
    dll
      name snmp
      path libndtaasnmp.so.0
      version 1.5.0.0
      status running
...
```

NOTE: access agent can not be unloaded from the session handled by the target of unload. E.g. executing from cli:

```
1:T admin@127.0.0.1> access-agent unload nc
```

will result in 'operation-failed' error.

13.1.4 cls

This command clears the current display.

13.1.5 connect

connect <host> ssh|tls [username] cert|key|pwd [password] | [certificate name] [key name]

Connect **ncclient** to NETCONF agent (**testagent**). The host address is a mandatory argument. When using

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

SSH, a User name must be specified as the second argument as well as one of ‘cert|key|pwd’ When ‘pwd’ is selected as below, the password should be provided at the "Password:" prompt. This is example of unsuccessful connection:

```
> connect 192.168.0.21 ssh admin pwd[Enter]
Password: ********
Failed to connect to 192.168.0.21!
>
```

or if using TLS

```
> connect 192.168.0.21 tls[Enter]
Certificate name: badcertname
Error loading client certificate /opt/ndConfBuilder-3.0.0-build002/nugen/ncclient/config/tls/certs/
badcertname.pem
>
```

If the connect succeeds, **ncclient** sends a "hello" message (shown in light blue below), and reads the "hello" from server (using SSH):

```
> connect 127.0.0.1 ssh admin pwd[Enter]
Password: ********
Fingerprint: A0 E0 6A E9 E1 29 85 3E 77 37 AB 20 97 88 03 88 83 5B 78 ED
Authentication methods: publickey,password,keyboard-interactive
Sending NETCONF client <hello>
<?xml version="1.0" encoding="UTF-8"?>
<hello>
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
</capabilities>
</hello>
]]>]]>
Reading NETCONF server <hello>
  capabilities
    capability urn:ietf:params:netconf:base:1.0
...
1:T admin@127.0.0.1 >
```

or if using TLS and if the connect succeeds, ncclient sends a "hello" message (shown in light blue below), and reads the "hello" from server (using SSH):

```
> connect 127.0.0.1 tls[Enter]
Certificate name: cliadmin
TLS Connected using certificate /opt/ndConfBuilder-3.0.0-build002/ndconf/ncclient/config/tls/certs/
cliadmin.pem
Sending NETCONF client <hello>
<?xml version="1.0" encoding="UTF-8"?>
<hello>
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
</capabilities>
</hello>
]]>]]>
Reading NETCONF server <hello>
  capabilities
    capability urn:ietf:params:netconf:base:1.0
...
1:T admin@127.0.0.1 >
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

At this point the ‘connected’ (including 'NETCONF') commands are available:

```
1:T admin@127.0.0.1 > ?
Possible completions:
 about
 cancel-commit
 close-session
 cls
 commit
 copy-config
 create-subscription
 datadll
 delete-config
 delete-subscription
 discard-changes
 disconnect
 edit-config
 establish-subscription
 exit
 get
 get-access-agents
 get-config
 get-data-dlls
 get-namespaces
 get-schema
 help
 kill-session
 kill-subscription
 list
 lock
 modify-subscription
 nc
 now
 partial-lock
 partial-unlock
 quit
 rpc
 show
 sleep
 unlock
 validate
1:T admin@127.0.0.1 >
```

* Red font commands are available only when ncclient is connected to the NETCONF agent

13.1.6 cancel-commit

cancel-commit [persist-id <id>]

Sends request to cancel a [specific] confirmed commit.

```
1:T admin@127.0.0.1> cancel-commit[Enter]
<?xml version="1.0" encoding="UTF-8" ?><rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="4">
      <cancel-commit>
      </cancel-commit>
</rpc>
  ok
1:T admin@127.0.0.1>
1:T admin@127.0.0.1> cancel-commit persist-id p10[Enter]
<?xml version="1.0" encoding="UTF-8" ?><rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="5">
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
      <cancel-commit>
             <persist-id>p10</persist-id>
      </cancel-commit>
</rpc>
  ok
1:T admin@127.0.0.1>
```

13.1.7 close-session

close-session

Sends request for graceful termination of a NETCONF session.

```
1:T admin@127.0.0.1 > close-session[Enter]
<?xml version="1.0" encoding="UTF-8" ?><rpc xmlns="urn:ietf:params:xml:ns:netcon
f:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <close-session xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" />
</rpc>
  ok
1:T admin@127.0.0.1>
```

13.1.8 commit

commit [confirmed | persist-id <id>]
commit confirmed [timeout <val>] [persist <id>]

The commit operation instructs the device to implement the configuration data contained in the candidate configuration, i.e candidate configuration is copied to running configuration.

```
                                                <-- … make changes to candidate

1:T admin@127.0.0.1> commit[Enter]                <-- commit immediately
  ok

                                                <-- … or make changes to candidate

1:T admin@127.0.0.1> commit confirmed persist ND5 timeout 1000[Enter] <-- timeout 1000 seconds,
persist id = "ND5"
  ok

                                                <-- … or make changes to candidate

1:T admin@127.0.0.1> commit confirmed persist ND5[Enter] <-- commit again (sets the timer to default
600sec)
  ok

                                          <-- …then either the system "reverts changes" on timeout or

1:T admin@127.0.0.1> commit persist-id ND5[Enter]    <-- confirming commit (if within the timeout value)
1:T admin@127.0.0.1> cancel-commit persist-id ND5[Enter]<-- cancel changes (if within the timeout value)
```

13.1.9 copy-config

copy-config <source> <target>
copy-config candidate|running|startup|<url> candidate|running|startup|<url>

Supported value for url is file URI scheme. Extension "xml" is appended to url.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
1:T admin@127.0.0.1> copy-config candidate abc[Enter]
source: candidate
target: file:///abc.xml
  ok
1:T admin@127.0.0.1>
```

13.1.10 create-subscription

create-subscription [startTime YYYY-MM-DDThh:mm:ssZ]

This operation initiates an event notification subscription that will send asynchronous event notifications to the **ncclient** until the subscription terminates.

```
1:T admin@127.0.0.1> create-subscription[Enter]
  ok
1:T admin@127.0.0.1>
```

Later:

```
1:T admin@127.0.0.1>
notification
  eventTime 2015-04-03T20:56:19Z
  netconf-capability-change
    changed-by
      username cli
      session-id 1
    added-capability urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB?module=ND-GARAGE-
MIB&revision=2004-07-09
```

Subscribe for notifications with replay:

```
1:T admin@127.0.0.1> create-subscription startTime 2015-04-01T00:00:00Z[Enter]
notification
  eventTime 2015-04-06T21:10:59Z
  netconf-session-start
    username cli
    session-id 1

notification
  eventTime 2015-04-06T21:10:59Z
  netconf-session-start
    username snmp
    session-id 2

notification
  eventTime 2015-04-06T21:11:17Z
  netconf-session-start
    username user1
    session-id 3

notification
  eventTime 2015-04-06T21:11:19Z
  coLevelRisingAlarm
    object-1
      garageCOLevel 51

rpc-reply
  ok
1:T admin@127.0.0.1>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

13.1.11 datadll

datadll [load | unload] <name>

Load or unload data dll. Argument <name> is the name attribute for datadll element in "ndconf.xml":

```
    <datadlls>
        <datadll name="garage" dll="Garage.dll" ></datadll>
```

Argument <name> is also displayed in the output of "show datadll":

```
1:T admin@127.0.0.1> get-data-dlls[Enter]
  data
    dll
      name garage
      path /ndconf/lib64/libndGarageMib.so.0
      loaded true
   ...
1:T admin@127.0.0.1> datadll unload garage[Enter]
  ok
1:T admin@127.0.0.1> get-data-dlls[Enter]
  data
    dll
      name garage
      path /ndconf/lib64/libndGarageMib.so.0
      loaded false
...
1:T admin@127.0.0.1> datadll load garage[Enter]
  ok
1:T admin@127.0.0.1> get-data-dlls[Enter]
  data
    dll
      name garage
      path /ndconf/lib64/libndGarageMib.so.0
      loaded true
>
```

NOTE: Loading or unloading causes a "netconf-capability-change" notification. Also note, that if there is an interdependency between datadlls and you try to unload the dependent one, you'll receive the following error message:

```
  <rpc-error>
     <error-type>application</error-type>
     <error-tag>internal-error</error-tag>
     <error-severity>error</error-severity>
     <error-app-tag>general-error</error-app-tag>
     <error-message>Module: imported in other module</error-message>
  </rpc-error>
```

13.1.12 delete-config

delete-config startup|<url>

Supported value for url is file URI scheme.

```
1:T admin@127.0.0.1> delete-config startup[Enter]
target: startup
ok
1:T admin@127.0.0.1>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

13.1.13 delete-subscription

delete-subscription <id>

The "delete-subscription" operation permits canceling an existing subscription.

```
1:T admin@127.0.0.1> delete-subscription 3[Enter]
  rpc-reply
    ok
1:T admin@127.0.0.1>
```

13.1.14 discard-changes

discard-changes

Revert the candidate configuration to the current running configuration

```
1:T admin@127.0.0.1> discard-changes[Enter]
ok
1:T admin@127.0.0.1>
```

13.1.15 disconnect

disconnect

Disconnects **ncclient** from the **testagent**, 'NETCONF' commands are removed.

```
1:T admin@127.0.0.1> disconnect[Enter]
Disconnected
>
```

13.1.16 edit-config

edit-config candidate|running <file-name>

The <edit-config> operation loads all or part of a specified configuration to the specified target configuration datastore. This command reads file specified as the argument (extension xml is assumed) from the folder .../config/edit-config, and creates **edit-config** rpc request to be executed by the target agent. E.g. file might contain:

```
    <default-operation>merge</default-operation>
    <test-option>set</test-option>
    <config>
      <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
        <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create" >
          <t:vehicleIndex>4</t:vehicleIndex>
          <t:vehicleLicencePlate>777 SWE</t:vehicleLicencePlate>
          <t:vehicleModel>Koenigsegg One:1</t:vehicleModel>
        </t:vehiclesEntry>
      </t:garage>
    </config>
```

This command will add an RPC wrapper around it:

```
  <rpc message-id="3" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <edit-config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
          <target>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
            <candidate />
          </target>
  CONTENT OF THE FILE IS INSERTED HERE
      </edit-config>
  </rpc>
```

before passing it to **testagent** for processing.

```
1:T admin@127.0.0.1> get-config candidate xpath //ndg:vehiclesEntry[Enter]  <--- get config
  data
    vehiclesEntry
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
...
1:T admin@127.0.0.1> list edit-config c1[Enter]                          <--- check content of 2.xml
  <default-operation>merge</default-operation>
  <test-option>set</test-option>
  <config>
    <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
      <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create">
        <t:vehicleIndex>4</t:vehicleIndex>
        <t:vehicleLicencePlate>777 SWE</t:vehicleLicencePlate>
        <t:vehicleModel>Koenigsegg One:1</t:vehicleModel>
      </t:vehiclesEntry>
    </t:garage>
  </config>
1:T admin@127.0.0.1> edit-config candidate c1[Enter]                     <--- send edit-config rpc
  ok
1:T admin@127.0.0.1> get-config candidate xpath //ndg:vehiclesEntry[Enter]  <--- get config
  data
    vehiclesEntry
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
...
    vehiclesEntry                                                        <--- new entry created
      vehicleIndex 4
      vehicleLicencePlate 777 SWE
      vehicleModel Koenigsegg One:1
1:T admin@127.0.0.1>
```

13.1.17 edit-data

edit-data candidate|running <file-name>

The <edit-data> operation loads all or part of a specified configuration to the specified target configuration datastore. Note that the <edit-data> operation does not support the "error-option" and the "test-option" parameters that were part of the <edit-config> operation. The error behavior of <edit-data> corresponds to the "rollback-on-error" value in the "error-option" parameter.

This command reads file specified as the argument (extension xml is assumed) from the folder .../config/edit-data, and creates **edit-data** rpc request to be executed by the traget agent. E.g. file “g1.xml” might contain:

```
    <default-operation>merge</default-operation>
    <config>
      <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
        <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create" >
          <t:vehicleIndex>4</t:vehicleIndex>
          <t:vehicleLicencePlate>777 SWE</t:vehicleLicencePlate>
          <t:vehicleModel>Koenigsegg One:1</t:vehicleModel>
        </t:vehiclesEntry>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
      </t:garage>
    </config>
```

The command “edit-data candidate g1” will add an RPC wrapper around the content of g1.xml:

```
  <rpc xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda"
       xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores" message-id="6">
     <edit-data>
       <datastore>ds:candidate</datastore>
  CONTENT OF THE FILE IS INSERTED HERE
     </edit-data>
   </rpc>
```

before sending it to the agent for processing.

```
1:T admin@127.0.0.1> get-data candidate all -1 no-origin xpath //ndg:vehiclesEntry[Enter]
  data
    vehiclesEntry
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
...
1:T admin@127.0.0.1> list edit-data g1[Enter]                          <--- check content of g1.xml
  <default-operation>merge</default-operation>
  <config>
    <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
      <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create">
        <t:vehicleIndex>4</t:vehicleIndex>
        <t:vehicleLicencePlate>777 SWE</t:vehicleLicencePlate>
        <t:vehicleModel>Koenigsegg One:1</t:vehicleModel>
      </t:vehiclesEntry>
    </t:garage>
  </config>
1:T admin@127.0.0.1> edit-data candidate g1[Enter]                       <--- send edit-data rpc
  ok
1:T admin@127.0.0.1> get-data candidate all -1 no-origin xpath //ndg:vehiclesEntry[Enter]
  data
    vehiclesEntry
      vehicleIndex 1
      vehicleLicencePlate 123 ABC
      vehicleModel Maserati Quattroporte
...
    vehiclesEntry                                                       <--- new entry created
      vehicleIndex 4
      vehicleLicencePlate 777 SWE
      vehicleModel Koenigsegg One:1
1:T admin@127.0.0.1>
```

13.1.18 establish-subscription

```
establish-subscription stream <stream-name> [filter <filename>] [start <start-datetime>]
                       [stop <stop-datetime>]
establish-subscription datastore <dsname> [filter <filename>] [trigger <filename>]
```

The "establish-subscription" RPC allows a subscriber to request the creation of a subscription.

The **establish-subscription stream** command reads the specified file (with an assumed .xml extension) from the .../config/stream-filter directory and constructs an establish-subscription stream RPC request, which is then executed by the target agent.

If the file “f1.xml” contains:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
<stream-xpath-filter>
  //ndg:coLevelFallingAlarm
</stream-xpath-filter>
```

then the command “establish-subscription stream NETCONF filter f2” will generate the following RPC:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="35">
  <n2:establish-subscription>
    <n2:stream>NETCONF</n2:stream>
    <n2:stream-xpath-filter>//ndg:coLevelFallingAlarm</n2:stream-xpath-filter>
  </n2:establish-subscription>
</rpc>
```

```
1:T admin@127.0.0.1> establish-subscription stream NETCONF filter f2
rpc-reply
    id 8
...

8[2]<

  notification
    eventTime 2025-08-15T21:12:05Z
    coLevelFallingAlarm
      object-1
        garageCOLevel 5
...
```

The **establish-subscription datastore** command reads the specified file (with an assumed .xml extension) from the .../config/yang-push directory and constructs an establish-subscription datastore RPC request, which is then executed by the target agent.

If the file “filt2.xml” contains:

```
<datastore-xpath-filter>
  /ndg:garage/ndg:garageObjects
</datastore-xpath-filter>
```

and trig2.xml:

```
<on-change xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <dampening-period>2000</dampening-period>
    <sync-on-start>true</sync-on-start>
    <excluded-change>create</excluded-change>
</on-change>
```

then the command “establish-subscription datastore operational filter filt2 trigger trig2” will generate the following RPC:

```
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
      xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="37">
    <n2:establish-subscription xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
      <yp:datastore>operational</yp:datastore>
      <yp:datastore-xpath-filter>/ndg:garage/ndg:garageObjects</yp:datastore-xpath-filter>
      <yp:on-change xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <yp:dampening-period>2000</yp:dampening-period>
        <yp:sync-on-start>true</yp:sync-on-start>
        <yp:excluded-change>create</yp:excluded-change>
      </yp:on-change>
    </n2:establish-subscription>
  </rpc>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
1:T admin@127.0.0.1> establish-subscription datastore operational filter filt2 trigger trig2[Enter]
rpc-reply
    id 7
...
7[2]<

  notification
    eventTime 2025-08-15T21:15:19Z
    push-change-update
      id 7
      datastore-contents
        yang-patch
          patch-id 11
          edit
            edit-id 1
            operation merge
            target /ndg:garageObjects/ndg:garageCOLevel
            value
              garageCOLevel 6
...
```

13.1.19 exit

exit

Shutdown the **ncclient**.

```
> exit[Enter]
$
```

13.1.20 get

get [path|subtree|xpath <filter>]

Retrieve running configuration and state data from the NETCONF server. Executing get without parameters retrieves all data. **testagent** supports subtree and xpath filtering. <filter> argument depends on argument preceding it.

```
1:T admin@127.0.0.1> get xpath //ncm:sessions[Enter]
  data
    sessions
      session
        session-id 1
...
```

```
1:T admin@127.0.0.1> get subtree <a:netconf-state xmlns:a="urn:ietf:params:xml:ns:yang:ietf-netconf-
monitoring"><a:sessions><a:session /></a:sessions></a:netconf-state>[Enter]
  data
    netconf-state
      sessions
        session
          session-id 1
...
```

Specifying subtree filter in command line is cumbersome. "path" is simpler way to specify subtree filter. E.g. "/netconf-state/sessions/session" is internally translated in "<netconf-state><sessions><session /></sessions></netconf-state>"

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
1:T admin@127.0.0.1> get path /ncm:netconf-state/sessions/session[Enter]
  data
    netconf-state
      sessions
        session
          session-id 1
...
```

13.1.21 get-access-agents

Display Access agents status

```
1:T admin@127.0.0.1> get-access-agents[Enter]
data
      dll
        name cli
        path /opt/ndConfBuilder-3.0.0-build002/ndconf/lib64/libndtaacli.so.0
        version 1.6.1.0
        status running
      dll
        name nc
        path /opt/ndConfBuilder-3.0.0-build002/ndconf/lib64/libndtaanc.so.0
        version 2.0.0.0
        status running
      dll
        name snmp
        path /opt/ndConfBuilder-3.0.0-build002/ndconf/lib64/libndtaasnmp.so.0
        version 1.7.0.0
        status running
      dll
        name rc
        path /opt/ndConfBuilder-3.0.0-build002/ndconf/lib64/libndtaarestconf.so.0
        version 1.1.0.0
        status running
>
```

13.1.22 get-config

get-config candidate|running|startup [path|subtree|xpath <filter>]

Retrieve all or part of a specified configuration datastore (specified as the 1st argument). If filter is omitted retrieves all configuration data.

Processing of filter is the same as in **get**.

```
1:T admin@127.0.0.1> get-config candidate[Enter]
  data
    garage
      garageObjects
        garageAddress 10 Street
        garageCOLevelRisingThreshold 50
        garageCOLevelFallingThreshold 10
      vehiclesEntry
        vehicleIndex 1
...
```

13.1.23 get-data

get-data candidate|intended|operational|running|startup [all|config-only|state-only [maxdepth [no-origin|with-origin [path|subtree|xpath <filter>]]]]

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

Retrieve data from the NETCONF server. The first parameter: datastore name is mandatory. Executing get-data <dsname> retrieves all data from datastore <dsname>.

2nd argument is config-filter: it is used to retrieve only "config true" or "config false" nodes or both.

3rd arg, the "max-depth" parameter is used to limit the number of subtree levels that are returned in the reply.

4th arg “with-origin” requests that the server includes "origin" metadata annotations in its response, as detailed in the NMDA. If no-origin is specified then "origin" metadata annotations is not required in the response.

5th arg is a content filter parameter, similar to the "filter" parameter of <get-config>. It uses explicit nodes for subtree filtering ("subtree") and XPath filtering ("xpath"). ndCONF supports “path” filter as well.

```
1:T admin@127.0.0.1> get-data operational[Enter]

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="21">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda">
    <netconf-state xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring">
      <capabilities>
        <capability>urn:ietf:params:netconf:base:1.0</capability>
        <capability>urn:ietf:params:netconf:base:1.1</capability>
...
        <vehicleModel>Model 2</vehicleModel>
      </vehiclesEntry>
    </garage>
  </data>
</rpc-reply>

1:T admin@127.0.0.1> get-data operational config-only 2 no-origin xpath /ndg:garage[Enter]

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="22">
  <data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda">
    <garage xmlns="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
      <garageObjects/>
      <vehiclesEntry/>
      <vehiclesEntry/>
    </garage>
  </data>
</rpc-reply>
```

13.1.24 get-data-dlls

Retrieve list of configured data dlls with 'loaded' status

```
1:T admin@127.0.0.1> get-data-dlls[Enter]
  data
    dll
      name if
      path /home/a-user/ndconf/lib64/libietfInterfaces.so.0
      loaded true
   dll
     name yanglib
     path libietfYangLibrary.so.0
     loaded true
    dll
      name garage
      path /home/a-user/ndconf/ndGarageMib/Linux/libndGarageMib.so
      loaded true
1:T admin@127.0.0.1>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

13.1.25 get-namespaces

get-namespaces

Retrieve all namespaces and namespace prefixes.

```
1:T admin@127.0.0.1> get-namespaces[Enter]
  data
      namespace
       prefix y
       uri urn:ietf:params:xml:ns:yang:1
     namespace
       prefix yin
       uri urn:ietf:params:xml:ns:yang:yin:1
...
```

13.1.26 get-schema

get-schema identifier [<version> [<format>]]

Retrieve "schema" from the NETCONF server:

```
1:T admin@127.0.0.1> get-schema ietf-netconf[Enter]
 data  module ietf-netconf {

  // the namespace for NETCONF XML definitions is unchanged
  // from RFC 4741 which this document replaces

  namespace "urn:ietf:params:xml:ns:netconf:base:1.0";

  prefix nc;

  import ietf-inet-types {
    prefix inet;
  }

  organization
    "IETF NETCONF (Network Configuration) Working Group";
```

<version> and <format> are optional.

<version> is specified as YYYY-MM-DD, e.g. 2011-03-08

The only supported format is "**ncm:yang**".

13.1.27 help

help <cmdname>

Print help for command specified in argument.

```
1:T admin@127.0.0.1> help disconnect[Enter]
Disconnect from netconf server/agent
```

13.1.28 kill-session

kill-session <session-id>

Force the termination of a NETCONF session. "snmp" session can not be terminated this way.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
1:T admin@127.0.0.1> kill-session 3[Enter]
  ok
1:T admin@127.0.0.1>
```

13.1.29 kill-subscription

kill-subscription <id>

The "kill-subscription" operation permits an operator to end a dynamic subscription that is not associated with the transport session used for the RPC.

```
1:T admin@127.0.0.1> kill-subscription 3[Enter]
  rpc-reply
    ok
1:T admin@127.0.0.1>
```

13.1.30 list

list edit-config | rpc [<file-name>]

Print content of file from 'edit-config' or 'rpc' folder'. If file name is missing print list of files in corresponding folder.

```
> list edit-config[Enter]
 sub1.xml
 sub2.xml
 f1.xml
 x1.xml
>
```

If the file name is missing it prints a list of files in corresponding folder.

```
> list edit-config c1[Enter]
  <default-operation>merge</default-operation>
  <test-option>set</test-option>
  <config>
    <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
      <t:vehiclesEntry xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="create">
        <t:vehicleIndex>4</t:vehicleIndex>
        <t:vehicleLicencePlate>777 SWE</t:vehicleLicencePlate>
        <t:vehicleModel>Koenigsegg One:1</t:vehicleModel>
      </t:vehiclesEntry>
    </t:garage>
  </config>
>
```

13.1.31 lock

lock running|candidate|startup

Lock the entire configuration datastore system of a device.

```
1:T admin@127.0.0.1> lock candidate[Enter]
  ok
1:T admin@127.0.0.1>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

13.1.32 modify-subscription

```
modify-subscription stream <id> [filter <filename>] [stop <stop-datetime>]
modify-subscription datastore <id> [filter <filename>] [trigger <filename>]
```

The "modify-subscription" operation permits changing the terms of an existing dynamic subscription.

The **modify-subscription stream** command reads the specified file (with an assumed .xml extension) from the .../config/stream-filter directory and constructs an “modify-subscription stream” RPC request, which is then executed by the target agent.

If the file “f1.xml” contains:

```
<stream-subtree-filter xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
    <t::coLevelRisingAlarm >
  </t::coLevelRisingAlarm>
</stream-subtree-filter>
```

then the command “modify-subscription stream NETCONF filter f2” will generate the following RPC:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="49">
  <n2:modify-subscription>
    <n2:id>3</n2:id>
    <n2:stream-subtree-filter xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
      <t:coLevelRisingAlarm></t:coLevelRisingAlarm>
    </n2:stream-subtree-filter>
  </n2:modify-subscription>
</rpc>
```

```
1:T admin@127.0.0.1> modify-subscription stream NETCONF filter f2
rpc-reply
    id 8
...

8[2]<

  notification
    eventTime 2025-08-15T21:12:05Z
    coLevelRisingAlarm
      object-1
        garageCOLevel 67
...
```

The **modify-subscription datastore** command reads the specified file (with an assumed .xml extension) from the .../config/yang-push directory and constructs an “modify-subscription datastore” RPC request, which is then executed by the target agent.

If the file “filt5.xml” contains:

```
<datastore-subtree-filter>
  <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
      <t:vehiclesEntry></t:vehiclesEntry>
  </t:garage>
</datastore-subtree-filter>
```

and trig1.xml:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
  <periodic>
      <period>3000</period>
      <anchor-time>2024-02-01T12:00:00Z</anchor-time>
  </periodic>
```

then the command “modify-subscription datastore operational filter filt5 trigger trig1” will generate the following RPC:

```
   <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
        xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="37">
      <n2:modify-subscription xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <n2:id>5</n2:id>
        <datastore-subtree-filter>
          <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
            <t:vehiclesEntry></t:vehiclesEntry>
          </t:garage>
        </datastore-subtree-filter>
        <periodic>
          <period>3000</period>
          <anchor-time>2024-02-01T12:00:00Z</anchor-time>
        </periodic>
      </n2:modify-subscription>
    </rpc>
```

```
1:T admin@127.0.0.1> modify-subscription datastore 5 filter filt5 trigger 1[Enter]
rpc-reply
    id 7
...

  notification
    eventTime 2025-08-15T22:06:00Z
    push-update
      id 5
      datastore-contents
        garage
          vehiclesEntry
            vehicleIndex 1
            vehicleLicencePlate CAR001
            vehicleModel Model 1
          vehiclesEntry
            vehicleIndex 2
            vehicleLicencePlate CAR002
            vehicleModel Model 2
...
```

13.1.33 nc

nc [[<connection-id] [new]]

Manages connections.

```
1:T admin@127.0.0.1> nc[Enter]
* 5: TLS - chadmin@127.0.0.1:53166 - session-id: 8 CALL-HOME
1:T admin@127.0.0.1>
```

Displays the connections table. The item with an ‘*’ is the current connection context.

```
1:T admin@127.0.0.1> nc 3[Enter]
1:T admin@127.0.0.1>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

Sets the specified connection-id as the current connection. If the connection specified doesn't exit,

```
no client with id <n>
```

is displayed.

```
1:T admin@127.0.0.1> nc new[Enter]
>
```

Allows the creation of a new connect, with the '**connect**' command.

13.1.34 now

Display the current local and UTC time.

```
1:T admin@127.0.0.1> now[Enter]
LOCAL: Tues May 16 16:13:38 2023
UTC:   2023-05-16T20:13:38Z
1:T admin@127.0.0.1>
```

13.1.35 partial-lock

partial-lock <xpath>

Perform a partial lock on a portion of the datastore system of a device, given by **<xpath>**.

```
1:T admin@127.0.0.1> partial-lock //ncm:netconf-state[Enter]
  lock-id 100
  locked-node /netconf-state
1:T admin@127.0.0.1>
```

13.1.36 partial-unlock

partial-unlock <lock-id>

Perform an unlock on a portion of the datastore system of a device, given by **<lock-id>**.

```
1:T admin@127.0.0.1> partial-unlock 100[Enter]
1:T admin@127.0.0.1>
rpc-reply
ok
1:T admin@127.0.0.1>
```

13.1.37 quit

quit

Shutdown the ncclient.

```
> quit[Enter]
$
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

13.1.38 redirectqueue

redirectqueue [disconnect|reconnect]

Manages connections to the auxiliary display application, **nccliout**. When a connect has been established, all responses and notification are forwarded to **nccliout**.

(re)connect to **nccliout**.

```
1:T admin@127.0.0.1> redirectqueue reconnect[Enter]
Received NETCONF messages will be redirected to the message queue '6'
1:T admin@127.0.0.1>
```

Disconnect from **nccliout**.

```
1:T admin@127.0.0.1> redirectqueue disconnect[Enter]
Received NETCONF messages will be printed in the console
1:T admin@127.0.0.1>
```

13.1.39 rpc

rpc <file-name>

This command reads the file <file-name> from the **<SDK installdir>/ndconfndconf/ncclient/config/rpc** folder. <file-name> can be specified without extension, in that case ".xml" is appended. The file should contain valid NETCONF rpc request. An example of the content:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
      <filter type="subtree">
          <garage>
          </garage>
      </filter>
  </get>
</rpc>
```

message-id is replaced by CLI before passing request to **testagent** for processing.

Assuming content of f1.xml is shown above:

```
1:T admin@127.0.0.1> rpc f1[Enter]
    garage
      garageObjects
        garageAddress 10 Street
        garageNumVehicles 2
...
```

If the file does not exist:

```
1:T admin@127.0.0.1> rpc f2[Enter]
Failed to open file config\rpc\f2.xml
1:T admin@127.0.0.1>
```

To display content of the file:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
1:T admin@127.0.0.1> rpc show f0[Enter]
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<get>
<filter type="subtree">
</filter>
</get>
1:T admin@127.0.0.1>
```

13.1.40 show

show history

Display command line history.

```
> show history[Enter]
0: connect
1: lock candidate
```

13.1.41 sleep

sleep <seconds>

Suspend execution for a specified number of seconds.

```
1:T admin@127.0.0.1> sleep 10[Enter]
```

Primarily intended for use in scripts, to allow output from a command to be sent, prior to executing the next command in the script.

13.1.42 unlock

unlock running|candidate|startup

Lock the entire configuration datastore system of a device.

```
1:T admin@127.0.0.1> unlock candidate[Enter]
  ok
```

13.1.43 validate

validate running|candidate|startup|<url>|config <filename>

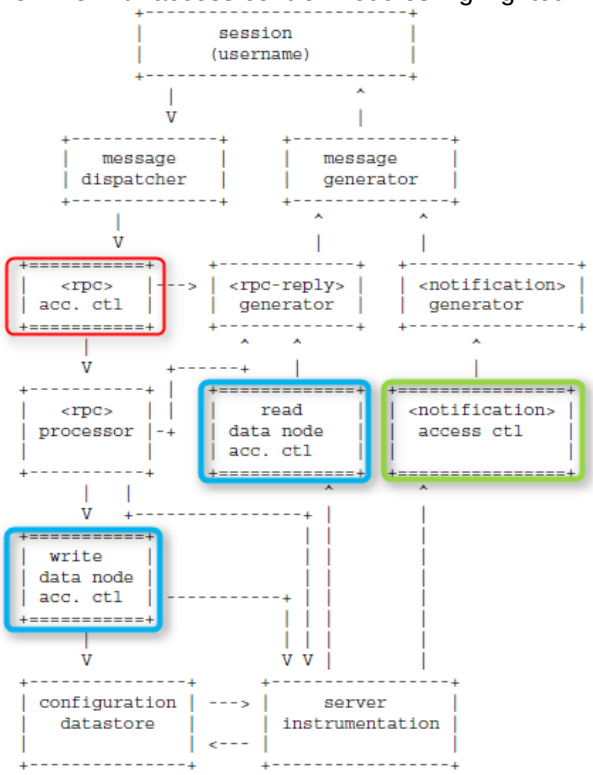
Validate the contents of the specified configuration.

```
1:T admin@127.0.0.1> validate startup[Enter]
source: startup  ok
1:T admin@127.0.0.1> delete-config startup[Enter]
  target: startup
  ok
1:T admin@127.0.0.1> validate startup[Enter]
source: startup  rpc-error
    error-type application
    error-tag internal-error
    error-severity error
    error-app-tag general-error
    error-message Data store is deleted
>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

14 NACM - Access Control

Access Control is specified in RFC 6536: Network Configuration Protocol (NETCONF) Access Control Model. Here is the diagram from the RFC with access control modules highlighted:



NOTE: **testagent** is shipped with the "**nacm**" disabled in the startup configuration. To enable access control, the user should modify the "nacm" section and restart the agent. Also, the user should add **their** user-name to each group, The examples in this document are for user "**test**".

NOTE 2: **testagent** creates a backup of **startup-cfg.xml** before overwriting it. See section "**testagent**" Command Line Options" for the backup location.

Access control configuration is located under the **nacm** element in **startup-cfg.xml**. Here is the excerpt:

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <enable-nacm>true</enable-nacm>              nacm is enabled
  <read-default>permit</read-default>          if no rule is found allow read and notify
  <write-default>deny</write-default>          if no rule is found deny write operations
  <exec-default>permit</exec-default>          if no rule is found allow execution of rpc
  <groups>
      <group>...</group>
      ... more groups ...
  </groups>
  <rule-list>
      <rule>...</rule>
      ... more rules ...
  </rule-list>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
</nacm>
```

14.1 Incoming RPC Message Validation

The following nacm configuration allows execution of all rpcs except "validate" rpc for user "test"

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <enable-nacm>true</enable-nacm>
  <exec-default>permit</exec-default>              default is permit
  <groups>
      <group>
           <name>admin</name>
           <user-name>test</user-name>               replace "test" with your login
      </group>
  </groups>
  <rule-list>
       <name>rule1</name>
       <group>*</group>
      <rule>
           <name>rule1-2</name>
           <module-name>*</module-name>
           <rpc-name>validate</rpc-name>
           <access-operations>*</access-operations>
           <action>deny</action>
      </rule>
  </rule-list>
</nacm>
```

"get" is allowed, "validate" is denied:

```
1:T admin@127.0.0.1> get[Enter]
rpc-reply
  data
    snmp
      engine
...
1:T admin@127.0.0.1> validate candidate[Enter]
rpc-reply
  rpc-error
    error-type application
    error-tag access-denied
    error-severity error
    error-app-tag no-access
    error-path /nc:ietf-netconf/nc:validate
    error-message NC: access denied
>
```

14.2 Outgoing <notification> Authorization

The ndGarageMib .so in **testagent** generates coLevelFallingAlarm and coLevelRisingAlarm notifications in random intervals.

The following nacm configuration allows all outgoing notifications.

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <enable-nacm>true</enable-nacm>
  <read-default>permit</read-default>              default is permit
```

In **ncclient**:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
1:T admin@127.0.0.1> create-subscription[Enter]
rpc-reply
  ok

notification
  eventTime 2015-12-11T20:06:37Z
  coLevelRisingAlarm                          <<<---  "rising"
    object-1
      garageCOLevel 51
  id 4

notification
  eventTime 2015-12-11T20:07:28Z
  coLevelFallingAlarm                         <<<---  "falling"
    object-1
      garageCOLevel 9
  id 5
...
```

The following addition to nacm configuration will deny "coLevelRisingAlarm" notification defined in "ND-GARAGE-MIB" YANG module to be sent to session with user "test".

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
...
  <groups>
      <group>
           <name>admin</name>
           <user-name>test</user-name>                replace "test" with your login
      </group>
  </groups>
  <rule-list>
       <name>rule1</name>
       <group>*</group>
      <rule>
           <name>rule1-2</name>
           <module-name>ND-GARAGE-MIB</module-name>
           <notification-name>coLevelRisingAlarm</notification-name>
           <access-operations>*</access-operations>
           <action>deny</action>
      </rule>
  </rule-list>
</nacm>
```

In **ncclient**, note that "rising" notifications are missing:

```
1:T admin@127.0.0.1> create-subscription[Enter]
1:T admin@127.0.0.1>
rpc-reply
  ok

notification
  eventTime 2015-12-11T20:12:51Z
  coLevelFallingAlarm                         <<<---  "falling"
    object-1
      garageCOLevel 9
  id 5

notification
  eventTime 2015-12-11T20:14:13Z
  coLevelFallingAlarm                         <<<---  "falling"
    object-1
      garageCOLevel 9
  id 7
...
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

14.3 Data Node Access Validation

The following nacm configuration allows read access to all objects in datastore.

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
 <enable-nacm>true</enable-nacm>
 <read-default>permit</read-default>            default is permit
```

In **ncclient**:

```
1:T admin@127.0.0.1> get[Enter]
 data
   snmp
     engine
       enabled true
...
```

Actually "permit" applies to all objects defined without nacm extension "default-deny-all". Definition of "nacm" container in **module ietf-netconf-acm** is:

```
container nacm {
  nacm:default-deny-all;
```

hence, "nacm" objects will be missing from the reply to "get" rpc. Add the following rule to be able to read "nacm" objects:

```
    <rule>
        <name>rule1-4</name>
        <module-name>*</module-name>
        <path>//nacm</path>
        <access-operations>read</access-operations>
        <action>permit</action>
    </rule>
```

When this rule is present (providing "user-name" is a member of the "group" specified in the "rule-list" to which this "rule" belongs):

```
1:T admin@127.0.0.1> get[Enter]
 data
   snmp
     engine
       enabled true
...
   nacm
     enable-nacm true
     read-default permit
...
```

Here is the example how to exclude object from the "read" view,

```
     <rule>
        <name>rule1-1</name>
        <module-name>ND-GARAGE-MIB</module-name>
        <path>//garageAddress</path>
        <access-operations>read</access-operations>
        <action>deny</action>
      </rule>
```

Without the rule above:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
1:T admin@127.0.0.1> get xpath //ndg:garageObjects[Enter]
  data
    garageObjects
      garageAddress 10 Street
      garageNumVehicles 2
      garageCOLevel 22
      garageCOLevelRisingThreshold 50
      garageCOLevelFallingThreshold 10
1:T admin@127.0.0.1>
```

After adding the rule (note that garageAddress is missing):

```
1:T admin@127.0.0.1> get xpath //ndg:garageObjects[Enter]
  data
    garageObjects
      garageNumVehicles 2
      garageCOLevel 22
      garageCOLevelRisingThreshold 50
      garageCOLevelFallingThreshold 10
```

By default write operation is denied:

```
  <nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
   <enable-nacm>true</enable-nacm>
   <write-default>deny</write-default>            default is deny
```

Using c1.xml as an example (see 6.1.1 edit-config):

```
   <config>
       <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
           <t:garageObjects>
               <t:garageAddress>123 eglinton</t:garageAddress>
           </t:garageObjects>
       </t:garage>
   </config>
```

In **ncclient**:

```
1:T admin@127.0.0.1> edit-config candidate c1[Enter]
  rpc-error
    error-type application
    error-tag access-denied
    error-severity error
    error-app-tag no-access
    error-path /nc:config-candidate/nc:config/ndg:garage
    error-message NC: access denied
```

After adding the following rule:

```
       <rule>
          <name>rule1-1</name>
          <module-name>ND-GARAGE-MIB</module-name>
          <path>//garageAddress</path>
          <access-operations>read create update delete</access-operations>
          <action>permit</action>
        </rule>
```

edit-config should succeed.

```
1:T admin@127.0.0.1> edit-config candidate c1[Enter]
  ok
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

14.4 Debugging Access Control

Access control configuration can be quite complicated and sometimes hard to find the problem when unexpected result occurs. YangDEMOAgent outputs debug messages while processing access control to the log file. Such lines start with:

nacmr - rpc access control

nacmn - notification access control

nacmd - data access control

The 1st line of access control processing contains session id, user name and rpc/notification or object name. E.g. notification "netconf-session-start" for session id =1, user = "test",

```
nacmn [1:test] netconf-session-start
```

One or more log lines following the 1st line are related to processing that rpc/notification or object name. The last one contains either "permit" or "deny". E.g.

```
nacmn 11 permit
```

Here is the example of multi-line output while processing edit-config:

```
nacmd [2:test] /config-candidate/config/garage/garageObjects/garageAddress
nacmd 6.1, rule=rule1-1, module-match=ND-GARAGE-MIB
nacmd 6.2 A match=/config-candidate/config/garage/garageObjects/garageAddress
nacmd 6.3.2 create
nacmd 7 permit
```

Example of denied rpc:

```
nacmr [2:test] /ietf-netconf/validate
nacmr 8 deny
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

15 NETCONF call-home

15.1 The truststore and keystore

15.1.1 truststore

The "**truststore**" is the section of the configuration that stores the CA certificates:

```
<truststore xmlns="urn:ietf:params:xml:ns:yang:ietf-truststore">
    <certificate-bags>
      <certificate-bag>
        <name>ca-bag</name>
        <certificate>
            <name>ca-root</name>
            <cert-data>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdNVENDQkJtZ0F3SUJBZ0l
...
dDNZb0xVCkVoeW1CVFE9Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K</cert-data>
        </certificate>
        <certificate>
            <name>ca-int</name>
            <cert-data>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdMekNDQkJlZ0F3
...
1NlcKLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=</cert-data>
        </certificate>
      </certificate-bag>
    </certificate-bags>
</truststore>
```

The **cert-data** element contains the certificate data. The type of **cert-data** is binary, so this value is the base64 encoded information from the certificate PEM file.

User may use the Linux utility **base64** to get the value to input into these fields. For example, assuming you have the file “abcd.pem”, then:

```
> base64 abcd.pem > abcd.pem.base64
```

Open the abcd.pem.base64 (it is a text file) and copy the content to value of <cert-data>

IMPORTANT: There must be no space between <cert-data> and the beginning of a string. Each new line must start at the beginning of line. There must be no space between the end of string and </cert-data>.

The following examples are common issues that are all invalid examples of copying data from a “.base64” text file (Note: in the examples below yellow regions indicate “white-space”):

a) Having a new line after <cert-data>:

```
<cert-data>
    LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdNVENDQkJtZ0F3SUJBZ0l
    ...
```

b) Having space(s) after <cert-data>:

```
<cert-data>   LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdNVENDQkJtZ0F3SUJBZ0l
    ...
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

c) Having space(s) or tab(s) inserted in the text:

```
<cert-data>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdNVENDQkJtZ0F3SUJBZ0l
                        DRUF3d0RRWUpL
     ...
```

d) Having a new line before </cert-data>:

```
     ...
     QVRFLS0tLS0K
     </cert-data>
```

e) Having space(s) or tab(s) before </cert-data>:

```
     ...
     QVRFLS0tLS0K    </cert-data>
```

The **truststore** is configured in the **config/start-cfg.xml** file for both **ncclient** and **testagent** (and **ncfonfsvr**).

Note: The **SDK**, as installed has a working configuration to allow for testing call-home based on certificates created by NDT only for that purpose. You will need to use your own certificates for you own deployments.

15.1.2 keystore

The **"keystore"** contains keys and certificates:

```
<keystore xmlns="urn:ietf:params:xml:ns:yang:ietf-keystore"
          xmlns:ct="urn:ietf:params:xml:ns:yang:ietf-crypto-types">
  <asymmetric-keys>
    <asymmetric-key>
      <name>serverkey</name>
      <public-key-format>ct:subject-public-key-info-format</public-key-format>
      <private-key-format>rsa-private-key-format</private-key-format>
      <public-key>MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvdn5m1e02QH5EoRHni2b
...
VQIDAQAB</public-key>
      <cleartext-private-key>MIIEowIBAAKCAQEAvdn5m1e02QH5EoRHni2bfmx/YEEBTQH+jKoN
...
OuMxcQhVb45NNUgLNrZ3WojPpMSEZSNp1St6VWueUkVlB8cbrMfw</cleartext-private-key>
      <certificates>
        <certificate>
          <name>servercert</name>
          <cert-data>MIIFxTCCA62gAwIBAgICEAwwDQYJKoZIhvcNAQELBQAwgawxCzAJBgNVBAYTAkNB
MRAwDgYDVQQIDAdPbnRhcmlvMR4wHAYDVQQKDBVOdURlc2lnbiBUZWNobm9sb2dp
...
y4u6E7O29zgTdk++HQZqgRvRdSX+hZUF0aUo10i2Zyc9r13YYsnH6xU=</cert-data>
        </certificate>
      </certificates>
    </asymmetric-key>
  </asymmetric-keys>
</keystore>
```

The type of <public-key>, <cert-data> and <cleartext-private-key> is binary, so the value in the config is base64 encoded. See <cert-data> in the truststore section above for an example.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

15.2 Server

The section of the server configuration[34] related to call-home is highlighted below:

```
<ncs:netconf-server xmlns:ncs="urn:ietf:params:xml:ns:yang:ietf-netconf-server"
                    xmlns:ct="urn:ietf:params:xml:ns:yang:ietf-crypto-types"
                    xmlns:cmn="urn:ietf:params:xml:ns:yang:ietf-tls-common"
                    xmlns:x509c2n="urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name">
  <ncs:listen>
  ...
  </ncs:listen>
  <ncs:call-home>
    <ncs:netconf-client>
      <ncs:name>chadmin</ncs:name>
      <ncs:endpoints>
        <ncs:endpoint>
          <ncs:name>ep-one</ncs:name>
     ...
        </ncs:endpoint>
      </ncs:endpoints>
      <ncs:connection-type>
        <ncs:persistent />
      </ncs:connection-type>
      <ncs:reconnect-strategy>
        <ncs:start-with>random-selection</ncs:start-with>
        <ncs:max-attempts>3</ncs:max-attempts>
      </ncs:reconnect-strategy>
    </ncs:netconf-client>
  ... MORE CLIENTS
  </ncs:call-home>
</ncs:netconf-server>
```

The netconf-client is identified by it's **name**, and for each client a list of endpoints has to be specified. **endpoint** can be either **TLS** or **SSH**.

The **connection-type** specifies how the server should maintain connections. Options are persistent and periodic.

The **reconnect-strategy** directs how a server reconnects to a client, after discovering its connection to the client has dropped, even if due to a reboot.

15.2.1 TLS endpoint

The endpoint is identified by it's **name**. The <**tls**> element indicates that this is a **TLS** endpoint.

```
<ncs:endpoint>
   <ncs:name>ep-one</ncs:name>
   <ncs:tls>
     <ncs:tcp-client-parameters>
        ...
     </ncs:tcp-client-parameters>
     <ncs:tls-server-parameters>
        ...
     </ncs:tls-server-parameters>
     <ncs:netconf-server-parameters>
        ...
     </ncs:netconf-server-parameters>
   </ncs:tls>
</ncs:endpoint>
```

34 The server configuration is contained in the testagent/ or ndconfsvr/ config/startup-cfg.xml file.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

There are 3 groups to configure:

1. tcp-client-parameters, **remote-address** is the address of the client; **local-address** is the address of the **TLS** server listening for connections. By default the NETCONF client that is a TLS call-home target, should listen on port **4335**, the NETCONF TLS server should listen on port **6513**. **keepalives** defines what the server should do if the session is idle.

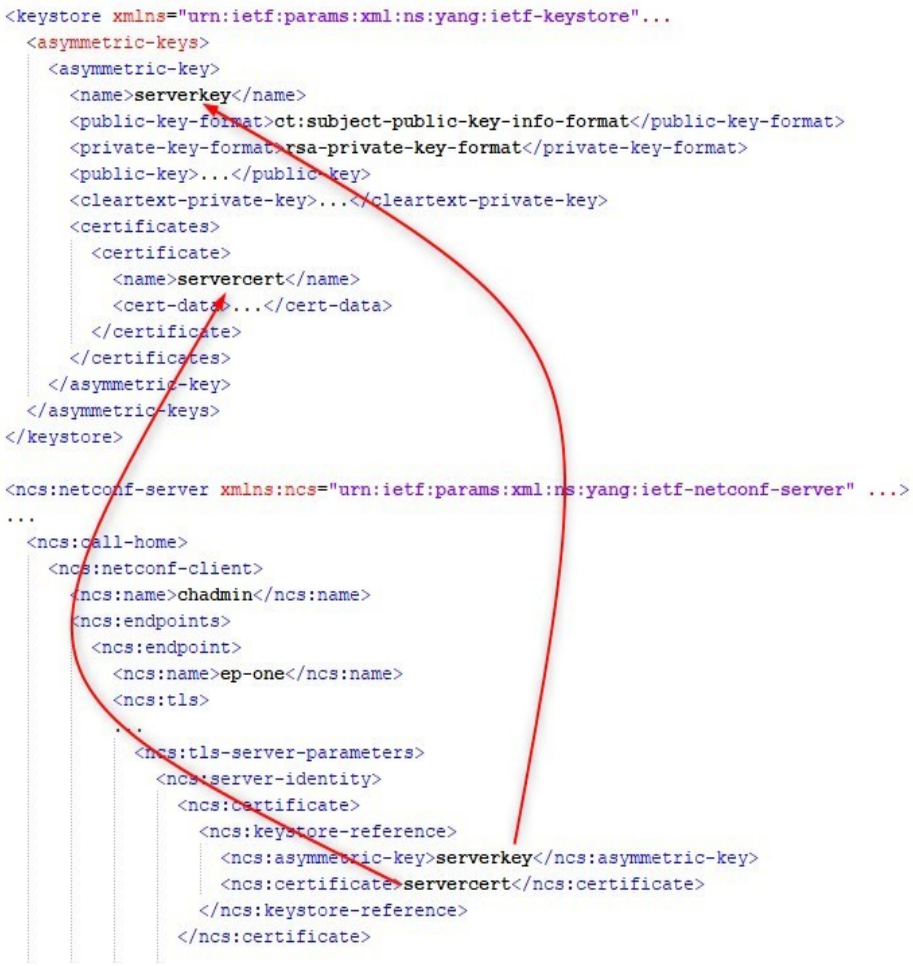
```
<ncs:tcp-client-parameters>
         <ncs:remote-address>127.0.0.1</ncs:remote-address>
         <ncs:remote-port>4335</ncs:remote-port>
         <ncs:local-address>0.0.0.0</ncs:local-address>
         <ncs:local-port>6513</ncs:local-port>
         <ncs:keepalives>
 …
```

2. tls-server-parameters, this is where server identity is specified. There are multiple options (see ietf-tls-server.yang). Here we are using key/certificate pair.

```
<ncs:tls-server-parameters>
         <ncs:server-identity>
            <ncs:certificate>
               <ncs:keystore-reference>
                  <ncs:asymmetric-key>serverkey</ncs:asymmetric-key>
                  <ncs:certificate>servercert</ncs:certificate>
               </ncs:keystore-reference>
            </ncs:certificate>
```

In the example above, the asymmetric-key and certificate are referencing the keystore:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



3. netconf-server-parameters, specifies mappings through which NETCONF client certificates are used to determine a NETCONF username (see ietf-netconf-server.yang and RFC 7407).

```
<ncs:netconf-server-parameters>
        <ncs:client-identity-mappings>
           <ncs:cert-to-name>
              <ncs:id>1</ncs:id>
              <ncs:fingerprint>04:43:58:af:e6:de:...:3b:72:e7</ncs:fingerprint>
              <ncs:map-type>x509c2n:specified</ncs:map-type>
              <ncs:name>chadmin</ncs:name>
           </ncs:cert-to-name>
        </ncs:client-identity-mappings>
     </ncs:netconf-server-parameters>
```

In the example above, map-type '**specified**' means that the name '**chadmin**' will be used.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

15.2.2 SSH endpoint

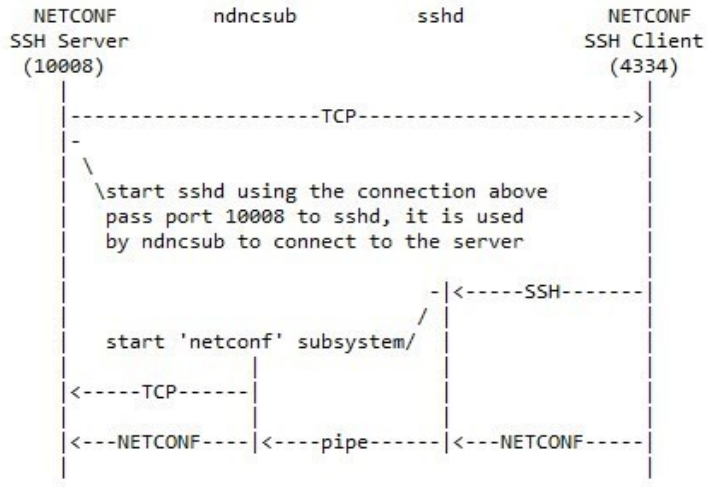
The endpoint is identified by it's **name**. The <**ssh**> element indicates that this is a **SSH** endpoint.

```
<ncs:endpoint>
   <ncs:name>ep2</ncs:name>
   <ncs:ssh>
     <ncs:t cp-client-parameters>
          ...
     </ncs:tcp-client-parameters>
   </ncs:ssh>
</ncs:endpoint>
```

For ssh connections, only tcp-client parameters should be configured, the other 2 groups are not used.

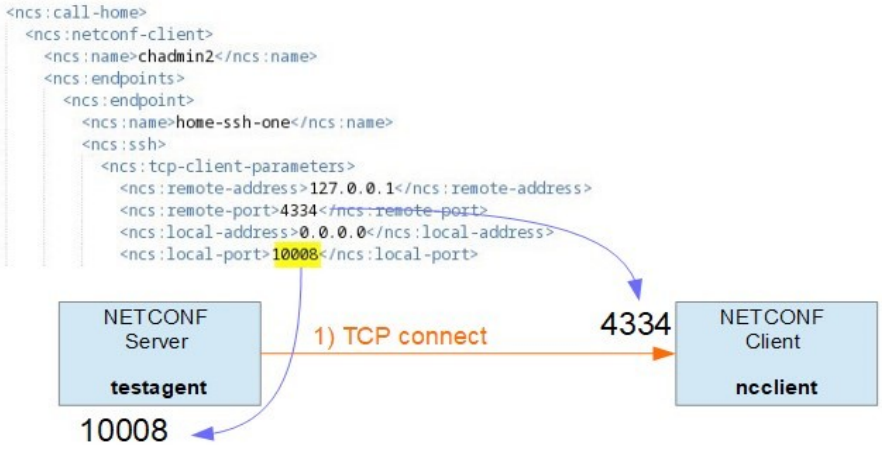
```
<ncs:tcp-client-parameters>
   <ncs:remote-address>127.0.0.1</ncs:remote-address>
   <ncs:remote-port>4334</ncs:remote-port>
   <ncs:local-address>0.0.0.0</ncs:local-address>
   <ncs:local-port>10008</ncs:local-port>
   <ncs:keepalives>
     ...
   </ncs:keepalives>
</ncs:tcp-client-parameters>
```

A Call home connection sequence is shown in the following diagram:

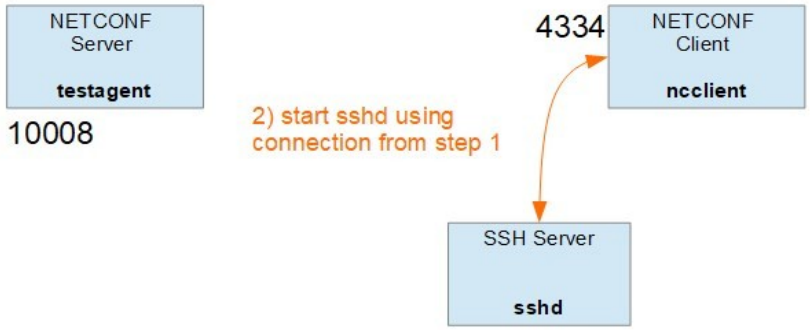


This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

1. The server establishes a TCP connection with client listening on port 4334.



2. The server starts sshd using the connection from step 1, the client is connected to sshd.

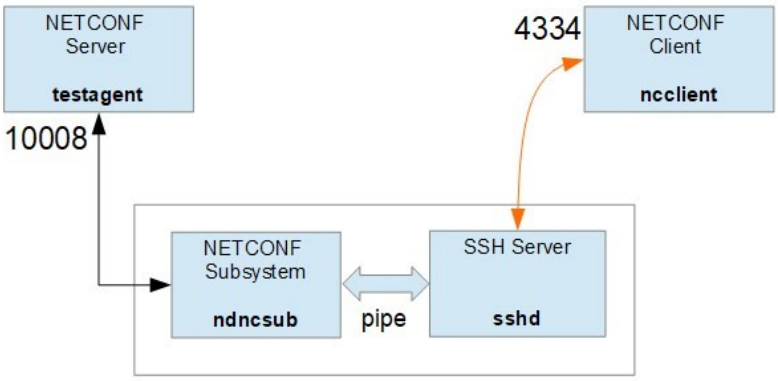


3. The client starts a SSH session and authenticates itself (password or public key/certificate)

4. **sshd** starts the NETCONF subsystem ndncsub

5. **ndncsub** connects to the server listening on port 10008 (this port is passed to **sshd** in step 2). The reason for not using 9522 (our standard SSH port) is that we won't be able to distinguish between a call-home and an ordinary SSH connection. By using a unique port, the server can match the NETCONF session to the correct call-home configuration.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



6. At this point the client can issue NETCONF requests.

15.3 client

The client is also a server. It listens for call-home TCP connections from a server seeking to “call home”.

The section of the client configuration related to call-home is depicted below:

```
<ncc:netconf-client xmlns:ncc="urn:ietf:params:xml:ns:yang:ietf-netconf-client" >
  <ncc:listen>
     <ncc:idle-timeout>3600</ncc:idle-timeout>
     <ncc:endpoint>
        <ncc:name>callhome1</ncc:name>
     ...
     </ncc:endpoint>
  ... MORE ENDPOINTS
  </ncc:call-home>
</ncc:netconf-server>
```

The client listens for incoming connections on endpoints. An **endpoint** can be either **TLS** or **SSH**.

15.3.1 TLS endpoint

The endpoint is identified by it’s **name**. The <**tls**> element indicates that this is a **TLS** endpoint.

```
 <ncc:endpoint>
    <ncc:name>callhome1</ncc:name>
    <ncc:tls>
      <ncc:tcp-server-parameters>
         ...
      </ncc:tcp-server-parameters>
      <ncc:tls-client-parameters>
         ...
      </ncc:tls-client-parameters>
    </ncc:tls>
</ncc:endpoint>
```

There are 2 groups to configure (the yang model defines a 3rd group as well, "netconf-client-parameters" but at present this group has no objects specified).

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

1. tcp-server-parameters, **local-address** is the address the TCP server is listening to for connections. By default the NETCONF client that is a TLS call-home target should listen on port **4335**. **keepalives** defines what this server should do if the session is idle.

```
<ncc:tcp-server-parameters>
   <ncc:local-address>127.0.0.1</ncc:local-address>
   <ncc:local-port>4335</ncc:local-port>
   <ncc:keepalives>
      ...
   </ncc:keepalives>
</ncc:tcp-server-parameters>
```

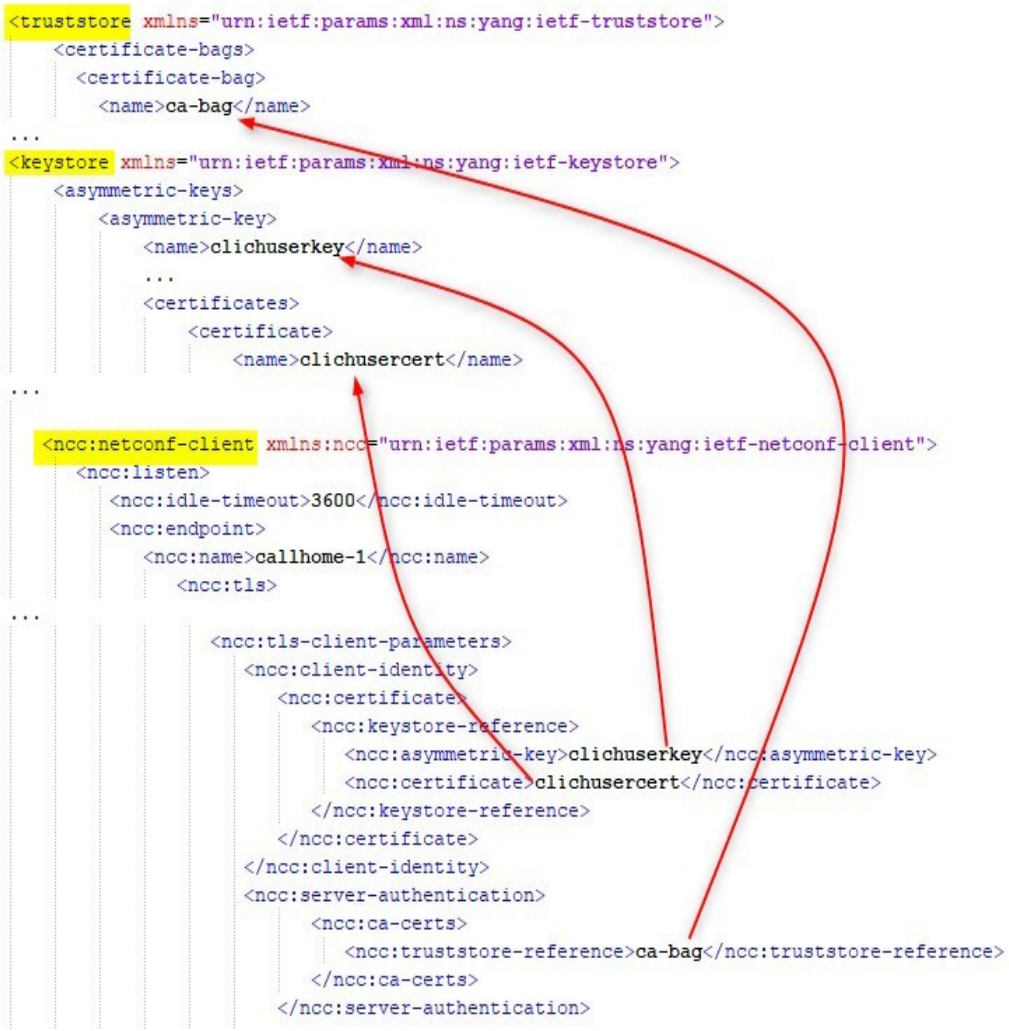
2. tls-client-parameters, this is where the client identity is specified. There are multiple options (see ietf-tls-client.yang). Here we are using a key/certificate pair.

```
<ncc:tls-client-parameters>
   <ncc:client-identity>
      <ncc:certificate>
         <ncc:keystore-reference>
            <ncc:asymmetric-key>clichuserkey</ncc:asymmetric-key>
            <ncc:certificate>clichusercert</ncc:certificate>
         </ncc:keystore-reference>
      </ncc:certificate>
   </ncc:client-identity>
   <ncc:server-authentication>
      <ncc:ca-certs>
         <ncc:truststore-reference>ca-bag</ncc:truststore-reference>
      </ncc:ca-certs>
   </ncc:server-authentication>
   <ncc:hello-params>
      <ncc:tls-versions>
         <ncc:tls-version>tls-1.2</ncc:tls-version>
      </ncc:tls-versions>
   </ncc:hello-params>
   <ncc:keepalives>
           ...
 </ncc:keepalives>
</ncc:tls-client-parameters>
```

In the example above, the asymmetric-key and certificate are referencing the **keystore** objects, and server-authentication is referencing the **truststore** object:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

15.3.2 SSH endpoint



The endpoint is identified by it's **name**. The <**ssh**> element indicates that this is a **SSH** endpoint.

```
<ncc:endpoint>
   <ncc:name>callhome1</ncc:name>
   <ncc:ssh>
     <ncc:tcp-server-parameters>
           ...
     </ncc:tcp-server-parameters>
     <ncc:ssh-client-parameters>
           ...
     </ncc:ssh-client-parameters>
   </ncc:ssh>
</ncc:endpoint>
```

There are 2 groups to configure (the yang model defines a 3rd group as well, "netconf-client-parameters" but at present this group has no objects specified).

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

1. tcp-server-parameters, the **local-address** is the address the TCP server is listening to for connections. By default the NETCONF client that is a SSH call-home target should listen on port **4334**. **keepalives** defines what this server should do if the session is idle.

```
<ncc:tcp-server-parameters>
   <ncc:local-address>127.0.0.1</ncc:local-address>
   <ncc:local-port>4334</ncc:local-port>
   <ncc:keepalives>
          ...
   </ncc:keepalives>
</ncc:tcp-server-parameters>
```

2. ssh-client-parameters, this is where the client identity is specified. There are multiple options (see ietf-ssh-client.yang). **Password**:

```
<ncc:ssh-client-parameters>
   <ncc:client-identity>
      <ncc:username>alice</ncc:username>
      <ncc:password>
         <ncc:cleartext-password>1234</ncc:cleartext-password>
      </ncc:password>
   </ncc:client-identity>
   <ncc:keepalives>
          ...
   </ncc:keepalives>
</ncc:ssh-client-parameters>
```

and **publickey**:

```
<ncc:ssh-client-parameters>
   <ncc:client-identity>
      <ncc:username>bob</ncc:username>
      <ncc:public-key>
         <ncc:keystore-reference>clindt</ncc:keystore-reference>
      </ncc:public-key>
   </ncc:client-identity>
   <ncc:keepalives>
          ...
   </ncc:keepalives>
</ncc:ssh-client-parameters>
```

authentication are supported.

15.4 A call-home example

The server starts "calling-home" immediately after starting, if there is a "call-home" configuration in the startup-cfg.xml file. The client listens for incoming call-home connections (on endpoints specified in the configuration).

If the call-home connection is successful, ncclient will output a hello exchange and an initial rpc **get** request to retrieve the NETCONF session info.

The client's console might look like as follows (for an SSH call-home connection):

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
AfterAccept callhome-2

Authentication methods: publickey

Sending NETCONF client <hello>
...
  capabilities
    capability urn:ietf:params:netconf:base:1.0
...
  session-id 3
CallHomeServer  Connected 127.0.0.1:52524

<?xml version="1.0" encoding="UTF-8"?>  <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-
id="2">
    <get>
      <filter type="xpath" select="//ncm:session[ncm:session-id=3]"></filter>
    </get>
  </rpc>

1[3]<

  rpc-reply
    data
      session
        session-id 3
        transport netconf-ssh
        username user1
        source-host 127.0.0.1
        login-time 2023-02-03T01:01:31Z
        in-rpcs 1
        in-bad-rpcs 0
        out-rpc-errors 0
        out-notifications 0
1:S user1@127.0.0.1>
```

The **AfterAccept** line indicates which endpoint successfully accepted the connection. Next the authentication method used is printed. Then the client sends a hello message and receives hello message from the server. If everything is ok, CallHomeServer line indicates the remote address.

The client then immediately sends a "get" rpc to retrieve the session info. On a successful reply this info is printed.

Finally the prompt is changed to indicate that this is now the active connection (see the "nc" section for details about the prompt and connections). Note that the prompt change occurs only if there is no active connection when a call-home connection is being processed.

Executing nc:

```
1:S user1@127.0.0.1> nc
* 1: SSH - @127.0.0.1:52524 - session-id: 3 CALL-HOME
```

At this point the client can manage the server. E.g.

```
1:S user1@127.0.0.1> get xpath //ndg:garage

<?xml version="1.0" encoding="UTF-8"?>  <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-
id="4">
    <get>
      <filter type="xpath" select="//ndg:garage"></filter>
    </get>
  </rpc>

1[3]<
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
rpc-reply
  data
    garage
      garageObjects
        garageAddress 10 street
        garageNumVehicles 2
        garageCOLevel 50
        garageCOLevelRisingThreshold 50
        garageCOLevelFallingThreshold 10
      vehiclesEntry
        vehicleIndex 1
        vehicleLicencePlate 123 ABC
        vehicleModel Maserati Quattroporte
      vehiclesEntry
        vehicleIndex 2
        vehicleLicencePlate 345 XYZ
        vehicleModel Jaguar F-Type
```

Note 1[3]< in the output above. This is an indication that an incoming message was received on connection 1, session 3. This is helpful when "create-subscription" is called on multiple sessions. It can be used to distinguish from which subscription the notification arrived.

What happens if the call-home session is closed?

The server will try to reconnect according to it's reconnect strategy. Our server introduces a delay between reconnect attempts. This is controlled by the "**call-home-reconnect-interval**" option for the NETCONF access agent (see ndconf.xml). If not specified, the **default** interval is **10** seconds. Setting this option to zero makes the re-connection to be tried immediately (as specified in the netconf client/server models). The server will then continuously try to reconnect, so setting a reconnect interval to zero may introduce a flood of connection attempts in the network.

15.5 Keys and Certificates

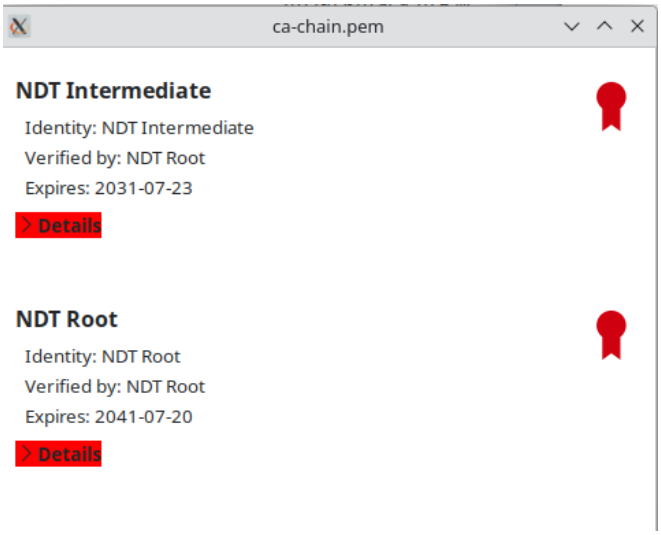
15.5.1 TLS

For a TLS connection to work, both the client and server have to be configured with valid certificates. The server authenticates the client by verifying the certificate sent by a client during the TLS handshake. The client authenticates the server by verifying certificate sent by a server during the TLS handshake.

For this purpose and only for testing, the SDK provides the same "self-signed" certificate in both the **<install_dir>/ndconf/ncclient/config/tls/ca-certs/** and **<install_dir>/testagent/config/tls/ca-certs/** (and also under **ndconvsvr**) directories, called '**ca-chain.pem**'.

This file contains two certificates:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



"NDT Root", valid until 2041 and "NDT Intermediate" valid until 2031. All client and server certificates are signed by "NDT Intermediate".

However, the content of these certificates (as documented previously) are embedded directly in the <truststore> section of the startup-cfg.xml files of both the server and the client.

So from either of the startup-cfg.xml files, the root certificate is the certificate named "ca-root" and the intermediate certificate is named "ca-int".

```
<truststore xmlns="urn:ietf:params:xml:ns:yang:ietf-truststore">
  <certificate-bags>
      <certificate-bag>
          <name>ca-bag</name>
          <certificate>
              <name>ca-root</name>
              <cert-data>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdNVENDQkJtZ0F3SUJBZ0lVTkM4VVlsZ
                  ...
          </certificate>
          <certificate>
              <name>ca-int</name>
              <cert-data>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUdMekNDQkJlZ0F3SUJBZ0lDRUFBd0RRWU
                  ...
          </certificate>
```

The server is configured with certificate and private key. These are loaded from the startup configuration. See the section **tls-server-parameters** for more details.

The call-home server in the client **does not** need server certificate and private keys; it is just a TCP server, serving only to establish the initial connection.

The client **does** need client certificate and private key to send to the server to establish a TLS connection.

For "regular" connections (initiated from the command prompt using the "connect" command) the certificate/key is looked up in **.../nugen/ncclient/config/tls/certs** and **.../nugen/ncclient/config/tls/private** folders respectively.

The name of the client certificate/key starts with the prefix "cli". The file name is irrelevant, all that matters is its content. E.g. in order for

```
> connect localhost tls cliadmin
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

to work, **cliadmin.pem** must be in the **<install_dir>/ndconf/ncclient/config/tls/certs** and **cliadmin.key** must be in the **<install_dir>/nugen/ncclient/config/tls/private** folder.

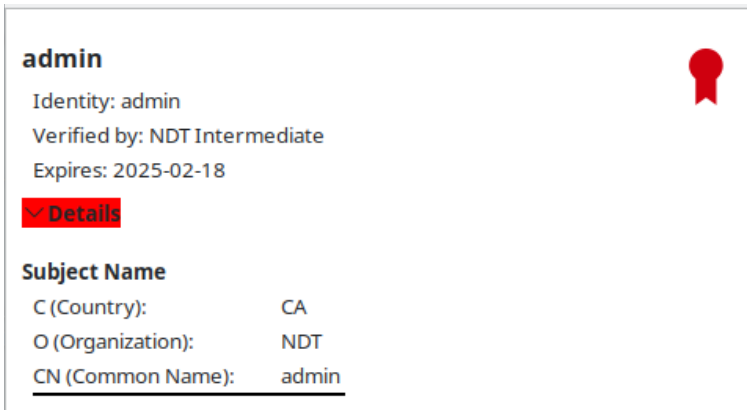
For the call-home connection, the certificate and key used are the ones configured for the particular call-home endpoint, see the tls-client-parameters section above.

15.5.1.1 Client Identity Mapping

For the client, under **<install_dir>/ndconf/ncclient/config/tls/certs**, are several certificates. These are:

- cliadmin.pem
- clisan1.pem
- clisan2.pem
- clisan3.pem
- clisan4.pem
- clisan5.pem

Taking 'cliadmin' as an example



Note that in the certificate above, "**admin**" is the **CN** or "**Common Name**". In the server configuration, the **CN** is specified for mapping the certificate to a user name, as follows:

```
<ncs:netconf-server xmlns:ncs="urn:ietf:params:xml:ns:yang:ietf-netconf-server"...>
    <ncs:listen>
      <ncs:idle-timeout>1800</ncs:idle-timeout>
      <ncs:endpoint>
        <ncs:name>mytls</ncs:name>
        <ncs:tls>
...
          <ncs:netconf-server-parameters>
            <ncs:client-identity-mappings>
              <ncs:cert-to-name>
                <ncs:id>1</ncs:id>
                <ncs:fingerprint>04:0E:D3:...:F5:22:9D[35]</ncs:fingerprint>
                <ncs:map-type>x509c2n:common-name</ncs:map-type>
              </ncs:cert-to-name>
```

35 The first octet of the fingerprint is the value of the algorithm signature used to create the fingerprint. In this case '04' indicates 'sha256'. See RFC5246 for more information.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

The configuration above specifies that for the client certificate with fingerprint **`04:0E:D3:.:F5:22:9D`**, the user's name is the certificate's common name.

To check the fingerprint:

```
$ openssl x509 -noout -fingerprint -sha256 -in cliadmin.pem
sha256 Fingerprint=0E:D3:A7:13:06:D9:90:9B:…:E2:B9:CE:B5:F5:F5:22:9D
```

Note: the first octet for the fingerprint above does not include the algorithm signature and needs to be included in the fingerprint in the server's configuration.

Like the above example for '**cliadmin**'. The rest of the certificates in this directory have the common names 'san1', 'san2', 'san3', 'san4' and 'san5'.

However, they also have a "X509v3 Subject Alternate Name" (SAN) set. If you do:

```
openssl x509 -noout -text -in clisan1.pem
```

You'll see that, under

```
X509v3 extensions:
      :
      X509v3 Subject Alternative Name:
            DNS:guest
```

This is set to "DNS:guest". So you can use "**guest**" as a user name, with the following configuration segment.

```
<ncs:cert-to-name>
    <ncs:id>2</ncs:id>
    <ncs:fingerprint>04:B6:7C:E8:...:21:47:A6:1E:7A</ncs:fingerprint>
    <ncs:map-type>x509c2n:san-dns-name</ncs:map-type>
</ncs:cert-to-name>
```

Where the specified fingerprint is that of 'clisan1'. The other clisan?.pem files use other SAN map types, as can be seen in the other cert-to-name sections of the server-authentications configuration.

Corresponding to each .pem file, each key file is in **`<install_dir>/ndconf/ncclient/config/tls/private`**.

15.5.2 SSH

For an SSH connection to work (when using keys), the server has to be configured with a valid private/public key pair. If the server accepts publickey authentication then the client could use a private/public key pair to authenticate itself to the server.

sshd (ssh server) is by default configured with 3 keys, (see sshd_config in /etc/ssh):

```
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
```

All the "HostKey"s specified are private keys.

If the client wants to verify that it is connecting to a trusted server, it has to know the address of the server and its public-key. This information is stored in the "**known_hosts**" file in the **~/.ssh** folder (~ indicates the home folder for the current user). The client reads this file at start up and uses this information to verify the server during the SSH handshake.

Here is an example of the content of a "**known_hosts**" file could be (note: keys have been shortened!):

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
localhost ecdsa-sha2-nistp256 E2VjZHNhLXNoYTItbmlz...9o1nw=
localhost ssh-ed25519    AAAAC3NzaC1lZDI1NT...S7Q2oA9b2Z
localhost ssh-rsa AAAAB3NzaC1y...EjZYV7DPP+Rji/cI/48QfxWiDGcz8E=
```

Each line consists of the address of the server (‘localhost’), followed by the server's public key. The **sshd** server stores these public keys in the ‘/etc/ssh’ directory. In the example above, the public key is the content of the files: ssh_host_ecdsa_key.pub, ssh_host_ed25519_key.pub, and ssh_host_rsa_key.pub respectively.

Note that the verification procedure requires an exact match of the address specified. With the settings above; a connection to ‘127.0.0.1’ would be rejected. It is possible to specify multiple lines for the same key and different address. E.g.

```
localhost ssh-rsa AAAAB3NzaC1y...EjZYV7DPP+Rji/cI/48QfxWiDGcz8E=
127.0.0.1 ssh-rsa AAAAB3NzaC1y...EjZYV7DPP+Rji/cI/48QfxWiDGcz8E=
192.168.0.55 ssh-rsa AAAAB3NzaC1y...EjZYV7DPP+Rji/cI/48QfxWiDGcz8E=
```

The server (**sshd**) can authenticate a client by password or public-key authentication.

For password authentication sshd_config has to contain the line

```
PasswordAuthentication yes
```

For public key authentication, sshd_config has to contain the line

```
PubkeyAuthentication yes
```

For publickey authentication to work, the server has to know the public keys of all clients that can be authenticated using this method. **sshd** reads these public keys from the file "**authorized_keys**" in **~/.ssh** folder (~ indicates home folder for the current user).

Here is the example of the content of "**authorized_keys**" file (key is shortened!):

```
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoY...VC+TSdHo37v5w= client
```

Each line consists of the client's public key. The last item in the line ("client") is ignored. Note that there is no address, client can log on from anywhere.

SSH keys

The **SDK** includes several SSH keys. To aid in creating alternative keys, below shows how these included keys are created.

ssh-keygen is a utility on Linux used to create SSH keys.

Host key: **ssh_host_ndt** (no passphrase!). Note: "identification" is "private key".

```
$ ssh-keygen -t ecdsa -f ssh_host_ndt -C nuDesignTech
Generating public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh_host_ndt
Your public key has been saved in ssh_host_ndt.pub
The key fingerprint is:
SHA256:2lcBNzIzoi39rnux8iVMBjHJjTmf7qbz/Cr3EgaDufo nuDesignTech
The key's randomart image is:
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
+---[ECDSA 256]---+
|       .+=* o    |
|       +*+.B .   |
|      oo+o ..    |
|      o.ooo  .   |
|       .S++ .    |
|      .o ==.     |
|     .. .o=+.    |
|    .   ++Bo     |
|     .E +%==o    |
+----[SHA256]-----+
```

Client key: **clindt** (no passphrase!)

```
$ ssh-keygen -t ecdsa -f clindt -C client
Generating public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in clindt
Your public key has been saved in clindt.pub
The key fingerprint is:
SHA256:LIhKQm4MnQfpZ//IG6z/KcsA42Xwcu+ZXccGsKxg2t4 client
The key's randomart image is:
+---[ECDSA 256]---+
|  ..             |
| ..o             |
|.o+ .   .        |
|= .=o. o o       |
|.==oO.. S .      |
|+o @ +.o   o     |
|. o o.=o  . +    |
|   . *+=.o o     |
|    o.E=+        |
+----[SHA256]-----+
```

The server (host) keys: **ssh_host_ndt** and **ssh_host_ndt.pub** can be found in **<install_dir>/ndconf/testagent[36]/config/ssh**.

The client keys: **clindt** and **clindt.pub** can be found in **<install_dir>/ndconf/ncclient/config/ssh**.

15.5.2.1 Configuring sshd and ncclient to use the include (NDT) SSH keys

To configure "**known_hosts**" for your SSH server, you **may** copy the information from one or more of the keys provided by the SSH server, as depicted in **15.5.2 SSH**

or

you can configure the SSH server to use the SSH keys provided by the SDK. These can be found in the **<install_dir>/ndconf/testagent[37]/config/ssh/** directory (as well as in the **ndconfsvr** directory).

First, copy the public and private host key files (ssh_host_ndt and ssh_host_ndt.pub) to **/etc/ssh**. Eg.

```
$ cd <install_dir>/ndconf/testagent/config/ssh
$ sudo cp ssh_host_ndt* /etc/ssh
```

Make sure the files are owned by root and are in the root group and that they are are not otherwise accessible to users.

Note: some distributions may require that the key file, ssh_host_ndt be in the “ssh_keys” group[38].

36 also under the equivalent **ndconfsvr** directory.
37 also under the equivalent **ndconfsvr** directory.
38 You can use “`$ getent group | grep -f ssh_keys`” to see if the group exists. If it does, you probably should consider it.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

Second, edit **/etc/ssh/sshd_config** to add the line:

```
Hostkey /etc/ssh/ssh_host_ndt
```

The server (**sshd**) can authenticate a client by password or public-key authentication. For password authentication **sshd_config** has to contain the line

```
PasswordAuthentication yes
```

For public key authentication sshd_config has to contain the line

```
PubkeyAuthentication yes
```

Now restart **sshd** daemon.

Assuming we will be using ncclient and sshd on the same machine, add the following 2 lines to **~/.ssh/known_hosts:**

```
localhost ecdsa-sha2-nistp256 E2VjZHNhLXNoYTItbmlz...9o1nw= [39]
127.0.0.1 ecdsa-sha2-nistp256 E2VjZHNhLXNoYTItbmlz...9o1nw=
```

where the high-lighted text in each line is the content of **ssh_host_ndt.pub** file. There is a copy of this file for your use, based on the above, in **<install_dir>/ndconf/testagent/config/ssh/**, so that you can just copy it to **~/.ssh/** to satisfy this requirement.

Add the content of **clindt.pub**, found in **<install_dir>/ndconf/ncclient/config/ssh/** (the entire single line) to **~/.ssh/authorized_keys**:

```
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoY...VC+TSdHo37v5w= client [40]
```

There is also a copy of this file for your use, based on the above, in **<install_dir>/ndconf/testagent/config/ssh/**, so that you can just copy it to ~/.ssh/ to satisfy this requirement.

Start the **testagent** and **ncclient**. From ncclient, execute (replace 'user1' with your username)

```
> nc new
> connect localhost ssh user1 key clindt
```

and you should be able to connect to the **testagent** without **sshd** asking you for password. If you have a problem connecting this way you may run **sshd** in debug mode.

To do so, first stop the **sshd** daemon:

```
$ sudo systemctl stop sshd
```

Confirm that **sshd** is not running:

```
$ systemctl status sshd
○ sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled; vendor preset:
disabled)
Active: inactive (dead) since ...
```

39 This string will likely not be the same string in the file as depicted.
40 This string will likely not be the same string in the file as depicted.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
Docs:  man:sshd(8)
man:sshd_config(5)
...
```

Open a console and execute:

```
$ sudo /usr/sbin/sshd -ddd
[sudo] password for user1:
debug2: load_server_config: filename /etc/ssh/sshd_config
debug2: load_server_config: done config len = 4308
...
```

Try to connect from ncclient and save the text from the sshd terminal to a file, it may be helpful in debugging the issue.

Once **ncclient** can use publickey authentication through "connect" we are ready to configure SSH callhome.

Note: if you make a change to your SSH server's configuration (i.e. sshd_config), remember to restart the SSH server to put the changes into effect.

E.g.

```
$ sudo systemctl restart sshd
```

15.5.2.2 Call-home over SSH using keys

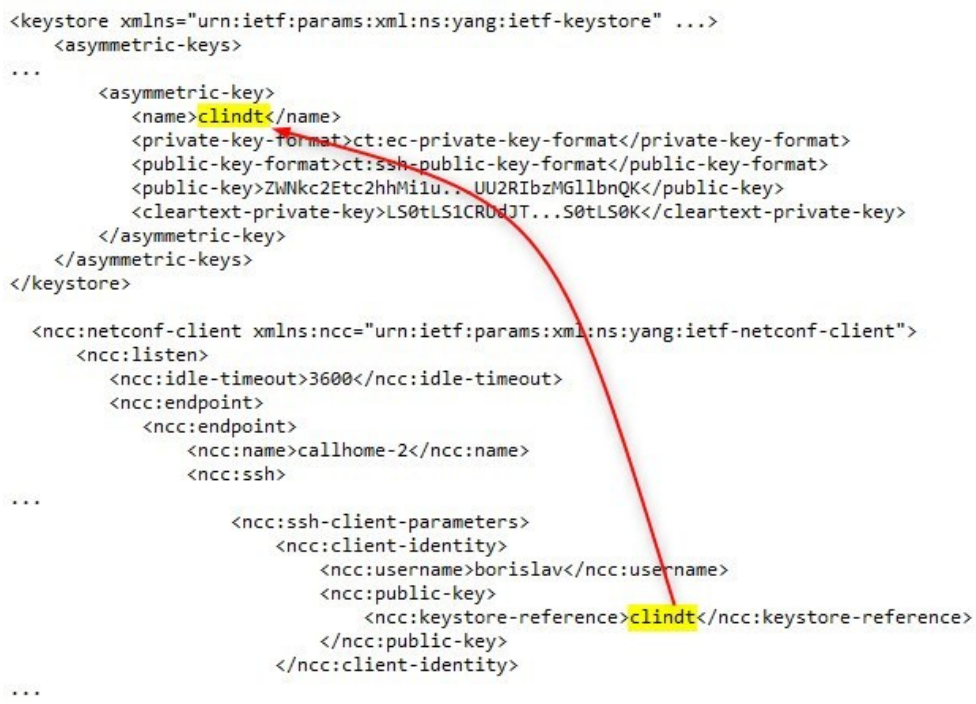
As explained in the server **SSH Endpoint** section, the server creates a TCP connection with a call-home server (ncclient) and then starts a new instance of **sshd** for that connection. This instance of **sshd** is initialized from the configuration file **ndchsshd_config:**

```
Port 2592
UsePrivilegeSeparation no
TCPKeepAlive yes
HostKey <install_dir>/nugen/testagent/config/ssh/ssh_host_ndt
PubkeyAuthentication yes
PasswordAuthentication no
LogLevel DEBUG3
ChallengeResponseAuthentication no
```

This instance of **sshd** will use the same **~/.ssh/authorized_keys** file to authenticate clients. Since we will be using the same client key it is already set in the previous section. If you plan to use a different client key make sure that it's public-key is added to **~/.ssh/authorized_keys.**

The call-home client keys are specified in the startup-cfg.xml file.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.



Note: the **public-key** and **cleartext-private-key entries** are binary values. The value of these 2 fields are base64 encodings of the corresponding key files.

By default, the startup configuration file for **ncclient**, **startup-cfg.xml**, in **<install_dir>/ndconf/ncclient/config/ssh/** needs to be modified to enable **call-home** for **SSH**. The configuration for SSH callhome is included, but is commented out to prevent it from being invoked prior to the required changes described above. To make these changes, open **ncclient's startup-cfg.xml** file in an editor and look for "callhome-2". You will see the following sequence when found:

```
<!--<ncc:endpoint>
     <ncc:name>callhome-2</ncc:name>
     <ncc:ssh>
```

Remove the comment introducer "**<!--**" from the line before "`<ncc:name>callhome-2...`", then search for "`</ncc:endpoint>-->`" and remove the trailing "**-->**". This will allow the SSH call-home configuration to be processed.

The configuration needs at least one further small change to be complete. Search for "**`<ncc:username>`**" within this section and note it's content is current "**user1**". Change this to your user name. If you intend to use call-home SSH with a password, change the current value of "**password1**" to the appropriate password, noting that this is a clear text password and is not recommended for deployment in a production environment. Our recommendation being to comment on the entire section introduced by "**`<ncc:password>`**".

At this point call-home should be working. If it does not you should enable debugging for **sshd**. This is done in **ndconf.xml**:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
 <ndt>
  <agents>
...
    <agent name="nc" dll="libndtaanc.so.0">
      <options>
        <option name="call-home-sshd-debug" value="true" />
      </options>
...
```

With the **call-home-sshd-debug** option set to "true" sshd started for call-home will create a log file with name **ndchsshdXXX.log** where **XXX** is the port number specified in the configuration for the corresponding endpoint.

15.5.3 SSH Certificates

Using SSH certificates simplifies the configuration and deployment of keys as the use of the **authorized_keys** file is **not required** in this case. The only file required to be deployed is the **CA** (Certificate Authority) public key. The **CA** private key is used to sign the server and client certificates. So neither the server nor the client needs to know about the other side, the certificate received during the handshake could be verified by **CA** public key (to confirm that they are signed by the **CA**'s private key).

IMPORTANT: due to limitation of libssh2 host verification does not work with host certificates.

This is a superior solution in a single administration domain with thousands of users. The issue is that the server certificate is tied to a domain, so the user that wants to try our SSH call home and wants certificate based authentication would have to create their own CA key, and server and client certificates.

Below demonstrates how to create the certificates included with the SDK. You will need to generate your own.

Using **ssh-keygen**[41], create a CA key called "**NDTCA**":

```
$ ssh-keygen -t ecdsa -f NDTCA -C nuDesignTech
Generating public/private ecdsa key pair.
Enter passphrase (empty for no passphrase): 1234
Enter same passphrase again: 1234
Your identification has been saved in NDTCA
Your public key has been saved in NDTCA.pub
The key fingerprint is:
SHA256:V2A8qGAAxm7B3LiQUgwetZclgQKLcr/g2/0vQrj9dOA nuDesignTech
The key's randomart image is:
+---[ECDSA 256]---+
|OO*o.o.. oo      |
|BOoo+ + ..o.     |
|B.=o + .   ..    |
|.= .. .    .     |
|. . ..  S .      |
| . ..... o       |
|  . .+  E .      |
|   o..o...       |
|  . . .+oo.      |
+----[SHA256]-----+
```

We will be using the keys created previously. The certificate is simply a wrapper for the public key with extra info: valid period, list of principals, etc. The important part that it is signed by CS's private key.

41 On Linux, see “man ssh-keygen” for more information.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

Creating the server certificate:

-h means create a host certificate as oppose to a user certificate.

-s sign using the specified (private) . E.g.'NDTCA'.

-I the key identity, not significant in this case.

-n is a list of prinicipals, (user or host names). In this case the name of the host is ‘fed36’.

-V the validity interval. In this case ‘52w’ meaning 52 weeks.

- the pubkey file. Path and file name of the public key file. E.g. ‘ssh_host_ndt.pub’

```
      # ssh-keygen -h -s NDTCA -n fed36 -I ndtserver -V +52w ssh_host_ndt.pub
Enter passphrase: 1234 <<--- this is passphrase for NDTCA
Signed host key ssh_host_ndt-cert.pub: id "ndtserver" serial 0 for fed36 valid from 2023-02-
06T15:44:00 to 2024-02-05T15:45:32
```

Creating the client certificate:

- note: no -h option, means this a user certificate

-s sign using the specified (private) . E.g.'NDTCA'.

-I the key identity, not significant in this case.

-n is a list of prinicipals, (user or host names). In this case the name of the user is ‘user1’.

-V the validity interval. Again, in this case ‘52w’ meaning 52 weeks.

- the pubkey file. Path and file name of the public key file. E.g. ‘clindt.pub’

```
# ssh-keygen -s NDTCA -n user1 -I bllogin -V +52w clindt.pub
Enter passphrase: 1234
Signed user key clindt-cert.pub: id "bllogin" serial 0 for user1 valid from 2023-diff 02-06T15:52:00
to 2024-02-05T15:53:47
```

To view the content of the certificate you can execute **ssh-keygen** with the -L option:

```
$ ssh-keygen -L -f ssh_host_ndt-cert.pub
ssh_host_ndt-cert.pub:
    Type: ecdsa-sha2-nistp256-cert-v01@openssh.com host certificate
    Public key: ECDSA-CERT SHA256:oX6EDrt+7ZOonUr2L4rWFlBV9qKPc386tZeeVsEr2wQ
    Signing CA: RSA SHA256:qudiHFV4/DMn6I2H1xmuScmFots6mSSezjLzl5tTA9g (using rsa-sha2-512)
    Key ID: "ndtserver"
    Serial: 0
    Valid: from 2023-02-06T15:44:00 to 2024-02-05T15:45:32
    Principals:
            fed36
    Critical Options: (none)
    Extensions: (none)

$ ssh-keygen -L -f clindt-cert.pub
clindt-cert.pub:
    Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
    Public key: ECDSA-CERT SHA256:LIhKQm4MnQfpZ//IG6z/KcsA42Xwcu+ZXccGsKxg2t4
    Signing CA: RSA SHA256:qudiHFV4/DMn6I2H1xmuScmFots6mSSezjLzl5tTA9g (using rsa-sha2-512)
    Key ID: "bllogin"
    Serial: 0
    Valid: from 2023-02-06T15:52:00 to 2024-02-05T15:53:47
    Principals:
            user1
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
    Critical Options: (none)
    Extensions:
            permit-X11-forwarding
            permit-agent-forwarding
            permit-port-forwarding
            permit-pty
            permit-user-rc
```

the “**Signing CA**” line is a fingerprint of the CA key. To see the fingerprint of NDTCA:

```
$ ssh-keygen -lf NDTCA.pub
3072 SHA256:qudiHFV4/DMn6I2H1xmuScmFots6mSSezjLzl5tTA9g nuDesignTech (RSA)
```

15.5.3.1 Configuring sshd and ncclient to use the "ndt" SSH certificates

Copy the CA public-key and server certificate to **/etc/ssh**.

```
$ sudo cp NDTCA.pub /etc/ssh
[sudo] password for xxxxx:

$ sudo cp ssh_host_ndt-cert.pub /etc/ssh
[sudo] password for xxxxx:
```

Edit **sshd_config** and add the following lines:

```
HostCertificate /etc/ssh/ssh_host_ndt-cert.pub
TrustedUserCAKeys /etc/ssh/NDTCA.pub
```

From the previous edit the following line should already be there:

```
HostKey /etc/ssh/ssh_host_ndt[42]
```

Make sure there is a line:

```
PubkeyAuthentication yes
```

Restart **sshd** after these changes.

(Note: the following is what is supposed to be working but is ignored by libssh2!)

Add the following line to **~/.ssh/known_hosts**

```
@cert-authority fed36 ssh-rsa  AAAAB3Nz...UyR0= nuDesignTech
```

where highlighted text is the content of **NDTCA.pub** file. Line starts with **@cert-authority**, and the second token is server's address/domain name (in this case **fed36**).

Once again, the above has effect.

Copy **clindt** (the private key) and **clindt-cert.pub** (the certificate) to **<install_dir>/ndconf/ncclient/config/ssh**.

42 Note: you will need to comment out the line “HostKey /etc/ssh/ssh_host_ecdsa_key” from sshd_config if you use HostKey /etc/ssh/ssh_host_ndt, since it replaces the ecdsa key for the server.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

Start the test agent and ncclient. From ncclient execute (note replace 'user1' with your username)

```
> connect localhost ssh user1 cert clindt
```

and you should be able to connect to the testagent without sshd asking you for password. In case of a problem connecting this way you should run sshd in debug mode (see the previous section).

15.5.3.2 Call-home over SSH using certificates

As explained in section **SSH Endpoint**, server creates TCP connection with call-home server (ncclient) and then starts the new instance of sshd with that connection. This instance of sshd is initialized from the configuration file **ndchsshd_config**. To use SSH certificates add the high lighted lines:

```
Port 2592
UsePrivilegeSeparation no
TCPKeepAlive yes
HostKey <install_dir>/ndconf/testagent/config/ssh/ssh_host_ndt
HostCertificate <install_dir>/ndconf/testagent[43]/config/ssh/ssh_host_ndt-cert.pub
TrustedUserCAKeys <install_dir>/ndconf/testagent/config/ssh/NDTCA.pub
PubkeyAuthentication yes
PasswordAuthentication no
LogLevel DEBUG3
ChallengeResponseAuthentication no
```

If you created you own files, make sure host key, host certificate and CA public key are configured (above) according to .

As explained **authorized_keys** file is not required in this case.

call-home client keys and certificates are specified in the startup-cfg.

43 or ndconfsvr

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
<keystore xmlns="urn:ietf:params:xml:ns:yang:ietf-keystore" ...>
    <asymmetric-keys>
        <asymmetric-key>
            <name>clindtkey</name>
            <private-key-format>ct:ec-private-key-format</private-key-format>
            <public-key-format>ct:ssh-public-key-format</public-key-format>
            <public-key>ZWNkc2Etc...bGllbnQK</public-key>
            <cleartext-private-key>LS0tLS1...S0tLS0K</cleartext-private-key>
            <certificates>
                <certificate>
                    <name>clindtcert</name>
                    <cert-data>ZWNkc2...W50Cg==</cert-data>
                </certificate>
            </certificates>
        </asymmetric-key>
    </asymmetric-keys>
</keystore>

   <ncc:netconf-client xmlns:ncc="urn:ietf:params:xml:ns:yang:ietf-netconf-client">
      <ncc:listen>
          <ncc:endpoint>
              <ncc:name>callhome-2</ncc:name>
              <ncc:ssh>
...
                  <ncc:ssh-client-parameters>
                     <ncc:client-identity>
                        <ncc:username>borislav</ncc:username>
                        <ncc:certificate>
                           <ncc:keystore-reference>
                              <ncc:asymmetric-key>clindtkey</ncc:asymmetric-key>
                              <ncc:certificate>clindtcert</ncc:certificate>
                           </ncc:keystore-reference>
                        </ncc:certificate>
...
```

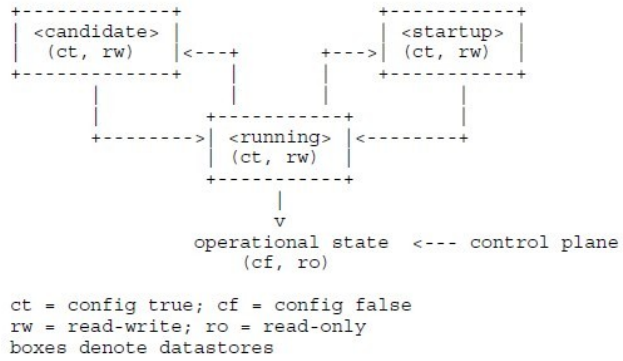
Note: **public-key, cleartext-private-key** and **cert-data** are binary values. The value of these 3 fields is a base64 encoding of the corresponding key/cert files.

At this point call-home should be working. If it does not you should enable debugging for sshd.

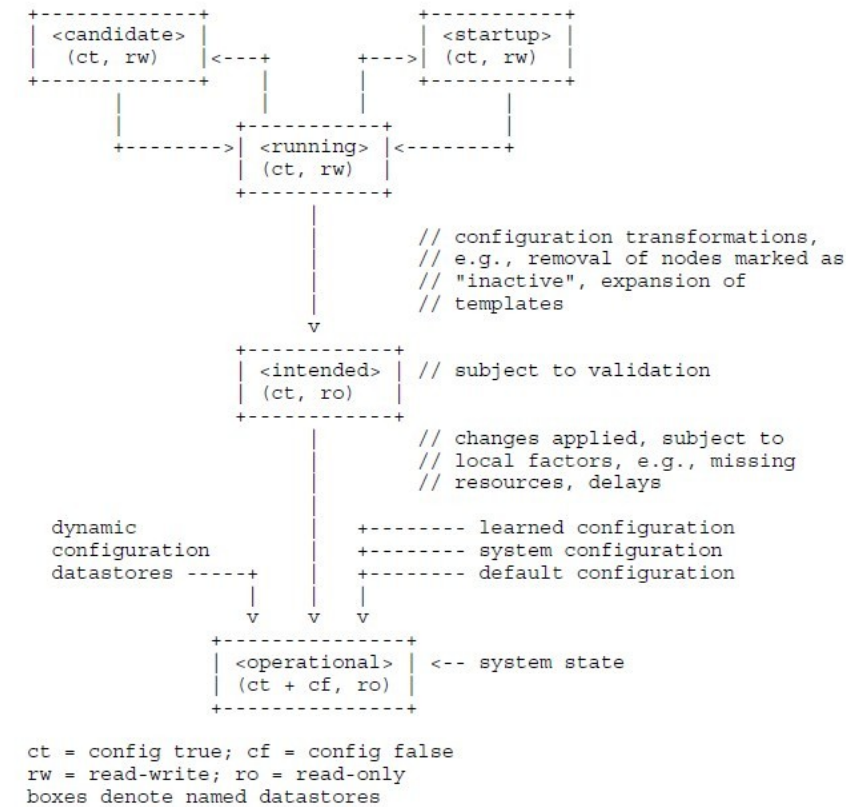
This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

16 Datastores

ndCONF supports both "original" (RFC 6241)



and NMDA (RFC 8342) model of datastores.



This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

Note that in the present ndCONF implementation <running> and <intended> are identical, see section 5.1.4. The Intended Configuration Datastore (<intended>) in RFC 8342.

Both ndconfsvr and testagent by default start in NMDA mode. To switch to “original” mode add “-o” switch to the command line. E.g.

```
> ndconfsvr -o -c
```

When the agent/server starts in the “original“ datastore mode, it expects implementation of “yanglib” and “interfaces” modules defined with yang 1.0. In other words, ndconf.xml has to have these entries in <datadlls> section:

```
<datadll name="yanglib" dll="libietfYangLibrary.so.0"/>
<datadll name="if" dll="libietfInterfaces.so.0" />
```

When running in NMDA datastore mode, the agent expects implementation of “yanglib” and “interfaces” modules defined with yang 1.1:

```
<datadll name="yanglib11" dll="libietfYangLibrary11.so.0"/>
<datadll name="if11" dll="libietfInterfaces11.so.0" />
```

The suffix “11” in the name indicates that this is implementation of yang 1.1 module.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

17 Subscribed Notifications

libietfSubscribedNotifications shared library contains implementation of RFC 8639 “Subscription to YANG Notifications”. "ietf-subscribed-notifications" yang module depends on “ietf-network-instance", which depends on "ietf-ip", "ietf-yang-schema-mount", and “ietf-interfaces” modules.

In order to use the ‘subscribed notifications’ the agent/server must use NMDA Datastore model and the following shared libs are required:

```
<datadll name="yanglib11" dll="libietfYangLibrary11.so.0"/>
<datadll name="if11" dll="libietfInterfaces11.so.0" />
<datadll name="ip" dll="libietfIp.so.0" ></datadll>
<datadll name="ism" dll="libietfYangSchemaMount.so.0" ></datadll>
<datadll name="ini" dll="libietfNetworkInstance.so.0" ></datadll>
<datadll name="isn" dll="libietfSubscribedNotifications.so.0" ></datadll>
```

The next sections provide examples of how to subscribe to notifications from “ND-GARAGE-MIB” yang module. This module is implemented in libndGarageMib.so library. To try these examples make sure that this shared lib is added to ndconf.xml:

```
<datadll name="garage" dll="libndGarageMib.so.0">
  <options>
      <option name="co-level-sampling-interval" value="20"/>
  </options>
</datadll>
```

Implementation generates random values for “garageCOLevel”. Depending on the previous and the current value of “garageCOLevel”, notification “coLevelRisingAlarm” or “coLevelFallingAlarm” may be generated.

“coLevelRisingAlarm” and “coLevelFallingAlarm” are configured in startup-cfg.xml:

```
<garage xmlns="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
  <garageObjects>
      <garageAddress>10 street</garageAddress>
      <garageCOLevelRisingThreshold>50</garageCOLevelRisingThreshold>
      <garageCOLevelFallingThreshold>10</garageCOLevelFallingThreshold>
  </garageObjects>
...
</garage>
```

Option “co-level-sampling-interval" (in the example above 20 seconds) for libndGarageMib.so specifies how often random value for coLevel is going to be generated.

When the current sampled coLevel is greater than or equal to garageCORisingFallingThreshold, and the coLevel at the last sampling interval was less than garageCORisingFallingThreshold, a single event coLevelRisingAlarm will be generated. After a rising event is generated, another such event will not be generated until the sampled value falls below this threshold and reaches the garageCOLevelFallingThreshold.

When the current sampled coLevel is less than or equal to garageCOLevelFallingThreshold, and the coLevel at the last sampling interval was greater than garageCOLevelFallingThreshold, a single event (coLevelFallingAlarm) will be generated. After a falling event is generated, another such event will not be generated until the sampled value rises above this threshold and reaches the garageCOLevelRisingThreshold.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

17.1 Filters

The filters container stores all subscription filters that persist beyond the lifecycle of a single subscription, allowing predefined filters to be shared across multiple subscriptions. Here is the example for xpath and subtree filters from startup-cfg.xml:

```
<filters xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
  <stream-filter>
      <name>fx1</name>
      <stream-xpath-filter>//ndg:coLevelFallingAlarm</stream-xpath-filter>
  </stream-filter>
  <stream-filter>
      <name>fx2</name>
      <stream-subtree-filter>
          <t:coLevelRisingAlarm xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" />
      </stream-subtree-filter>
  </stream-filter>
</filters>
```

17.2 Dynamic Subscriptions

To subscribe to “coLevelFallingAlarm” notification NETCONF client could send:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="4">
   <n2:establish-subscription>
      <n2:stream>NETCONF</n2:stream>
      <n2:stream-xpath-filter>/ndg:coLevelFallingAlarm</n2:stream-xpath-filter>
   </n2:establish-subscription>
</rpc>
```

or (assuming filters specified in the previous section are loaded by the agent):

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="5">
   <n2:establish-subscription>
      <n2:stream>NETCONF</n2:stream>
      <n2:stream-filter-name>fx1</n2:stream-filter-name>
   </n2:establish-subscription>
</rpc>
```

If the agent accepts this subscription it replies with subscription id, in this example “3”:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="4">
  <id xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">3</id>
</rpc-reply>
```

After a while, client should start seeing notifications like this one:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
   xmlns:ndg="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
  <eventTime>2025-07-31T19:50:23Z</eventTime>
  <ndg:coLevelFallingAlarm>
      <ndg:object-1>
          <ndg:garageCOLevel>1</ndg:garageCOLevel>
      </ndg:object-1>
  </ndg:coLevelFallingAlarm>
</notification>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

The existing subscription can be modified without stopping it. This example shows how to change the filter:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
 xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="15">
   <n2:modify-subscription>
     <n2:id>3</n2:id>
     <n2:stream-xpath-filter>//ndg:coLevelRisingAlarm</n2:stream-xpath-filter>
   </n2:modify-subscription>
 </rpc>
```

After accepting this request, the agent will be sending ‘rising” notifications, instead of “falling”.

To stop receiving notifications, client can “delete” or “kill” subscriptions. “delete” can be used for client’s own subscriptions only (i.e. this command works from the session that created subscription), while “kill” can stop notification subscription for any session. E.g. to stop subscription established above:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="11">
   <n2:delete-subscription>
   <n2:id>3<</n2:id>
   </n2:delete-subscription>
</rpc>
```

If this was valid request the agent would send <ok> reply and the client won’t be receiving notifications any more.

17.3 Configured Subscriptions

A configured subscription is one that has been established based on parameters defined in the configuration. The example below configures subscription with id 301, filter is ‘fx2’ (I.e. “coLevelRisingAlarm”, see Filters section), and the stream is default: “NETCONF”.

```
<subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"
               xmlns:nnsn="http://ndt-inc.com/ns/ndt-nc-subnotifs">
 <subscription>
     <id>301</id>
     <stream-filter-name>fx2</stream-filter-name>
     <stream>NETCONF</stream>
     <!-- <stop-time>2025-08-02T23:47:00Z</stop-time> -->
     <transport>nnsn:netconf</transport>
     <receivers>
         <receiver>
             <name>receiver1</name>
             <nnsn:netconf-endpoint>home-tls-one</nnsn:netconf-endpoint>
         </receiver>
     </receivers>
 </subscription>
</subscriptions>
```

There is one receiver: “receiver1” with netconf-endpoint “home-tls-one”. This is configured in the “netconf-server” section of the configuration:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
<netconf-server xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-server"
                xmlns:ct="urn:ietf:params:xml:ns:yang:ietf-crypto-types"
                xmlns:cmn="urn:ietf:params:xml:ns:yang:ietf-tls-common"
                xmlns:x509c2n="urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name">

  <listen>
  . . .
  </listen>
  <call-home>
    <netconf-client>
      <name>chadmin</name>
        <endpoints>
          <endpoint>
            <name>home-tls-one</name>
            <tls>
               <tcp-client-parameters>
                  <remote-address>127.0.0.1</remote-address>
                   <remote-port>4335</remote-port>
                    ...
```

Based on the configuration above, notifications will be sent to 127.0.0.1:4335, provided that the call-home connection has been established.

Note that “ietf-subscribed-notifications” does not define transport and receiver objects to be used in the configuration, from RFC 8639, section 2.5.2. “Creating a Configured Subscription”:

```
Receiver-specific transport connectivity parameters MUST be configured via
transport-specific augmentations to this specification.
```

Augmentation for the receiver-specific parameters is defined in “**ndt-nc-subscribed-notifications**” yang module.

```
  identity netconf {
    base sn:transport;
  }

  augment "/sn:subscriptions/sn:subscription/sn:receivers/sn:receiver" {
      when 'derived-from(../../transport, "nnsn:netconf")';
      leaf netconf-endpoint {
        type leafref {
          path "/ncs:netconf-server/ncs:call-home/ncs:netconf-client" +
               "/ncs:endpoints/ncs:endpoint/ncs:name";
        }
        mandatory true;
      }
  }
```

Implementation of the “ndt-nc-subscribed-notifications” yang module is in libietfSubscribedNotifications.so library.

Based on the configuration example above, on startup the agent will “call-home”, and if this is successful, notification streaming will start, and the client wil receive “subscription-started” notification:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
              xmlns:sn="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
  <eventTime>2025-08-01T19:45:54Z</eventTime>
  <sn:subscription-started>
    <sn:id>301</sn:id>
    <sn:stream-xpath-filter>//ndg:coLevelRisingAlarm</sn:stream-xpath-filter>
    <sn:stream>NETCONF</sn:stream>
    <sn:transport xmlns:nnsn="http://ndt-inc.com/ns/ndt-nc-subnotifs">nnsn:netconf</sn:transport>
    <sn:encoding>encode-xml</sn:encoding>
  </sn:subscription-started>
</notification>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

and later the subscribed notifications:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
              xmlns:ndg="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
  <eventTime>2025-08-01T19:46:02Z</eventTime>
  <ndg:coLevelRisingAlarm>
    <ndg:object-1>
      <ndg:garageCOLevel>54</ndg:garageCOLevel>
    </ndg:object-1>
  </ndg:coLevelRisingAlarm>
</notification>
```

17.4 Subscription monitoring

The client can retrieve the list of subscriptions from the agent by querying the ‘subscriptions’ container defined in “ietf-subscribed-notifications” yang module:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="5">
   <get>
      <filter type="xpath" select="/sn:subscriptions"></filter>
   </get>
</rpc>
```

Reply might look like:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="12">
  <data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
      <subscription>
        <id>301</id>
        <stream-filter-name>fx2</stream-filter-name>
        <stream>NETCONF</stream>
        <replay-start-time>2025-08-01T19:57:02Z</replay-start-time>
        <transport xmlns:nnsn="http://ndt-inc.com/ns/ndt-nc-subnotifs">nnsn:netconf</transport>
        <configured-subscription-state>valid</configured-subscription-state>
        <receivers>
          <receiver>
            <name>receiver1</name>
            <sent-event-records>40</sent-event-records>
            <excluded-event-records>39</excluded-event-records>
            <state>active</state>
     <netconf-endpoint xmlns="http://ndt-inc.com/ns/ndt-nc-subnotifs">home-tls-one</netconf-endpoint>
          </receiver>
        </receivers>
       </subscription>
       <subscription>
        <id>3</id>
        <stream-filter-name>fx1</stream-filter-name>
        <stream>NETCONF</stream>
        <receivers>
          <receiver>
            <name>127.0.0.1</name>
            <sent-event-records>13</sent-event-records>
            <excluded-event-records>12</excluded-event-records>
            <state>active</state>
          </receiver>
        </receivers>
      </subscription>
    </subscriptions>
  </data>
</rpc-reply>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

In this reply there are 2 subscriptions:

301 - this is a configured subscription, distinguished from dynamic subscriptions by the presence of the “configured-subscription-state” object.

3 – this is a dynamic subscription

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

18 Yang push

Yang push allows applications to subscribe to updates from a datastore and enables the server (the "publisher") to push and, in effect, stream those updates.

Similar to “Subscribed Notifications”, yang-push subscription can be dynamic or configured.

n order to use the ‘yang-push’ the agent/server must use NMDA Datastore model and the following shared libs are required:

```
<datadll name="yanglib11" dll="libietfYangLibrary11.so.0"/>
<datadll name="if11" dll="libietfInterfaces11.so.0" />
<datadll name="ip" dll="libietfIp.so.0" ></datadll>
<datadll name="ism" dll="libietfYangSchemaMount.so.0" ></datadll>
<datadll name="ini" dll="libietfNetworkInstance.so.0" ></datadll>
<datadll name="isn" dll="libietfSubscribedNotifications.so.0" ></datadll>
<datadll name="push" dll="libietfYangPush.so.0" ></datadll>
```

The next sections provide examples of how to subscribe to yang-push from “ND-GARAGE-MIB” yang module.

18.1 Filters

The filters container stores all subscription filters that persist beyond the lifecycle of a single subscription, allowing predefined filters to be shared across multiple subscriptions. Here is the example for xpath and subtree filters for yang-push from startup-cfg.xml:

```
<filters xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
  <selection-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <filter-id>ypf1</filter-id>
    <datastore-subtree-filter>
      <t:garage xmlns:t="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB" >
        <t:vehiclesEntry >
        </t:vehiclesEntry>
      </t:garage>
    </datastore-subtree-filter>
  </selection-filter>
  <selection-filter xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <filter-id>ypf2</filter-id>
    <datastore-xpath-filter>/ndg:garage/ndg:garageObjects</datastore-xpath-filter>
  </selection-filter>
</filters>
```

Note that the yang-push filters (“selection-filter”) are nested in the same container “filters” as notification filters (“stream-filter”).

18.2 System capabilities

The agent implements ietf-system-capabilities and ietf-notification-capabilities yang modules. These modules specify parameters for subscriptions, including which subset(s) of the datastores the clients can subscribe to. The agent loads this configuration from “yp-capabilities.xml”.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
<yp-capabilities xmlns="http://ndt-inc.com/yp-capabilities">
  <system-capabilities xmlns="urn:ietf:params:xml:ns:yang:ietf-system-capabilities">
    <datastore-capabilities>
      <datastore>operational</datastore>
      <per-node-capabilities>
        <node-selector>/ndg:garage</node-selector>
        <subscription-capabilities
               xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-capabilities">
          <max-nodes-per-update>50</max-nodes-per-update>
          <periodic-notifications-supported>
              config-changes state-changes
          </periodic-notifications-supported>
          <minimum-update-period>1200</minimum-update-period>
          <on-change-supported>state-changes</on-change-supported>
          <minimum-dampening-period>1200</minimum-dampening-period>
          <supported-excluded-change-type>create</supported-excluded-change-type>
          <supported-excluded-change-type>move</supported-excluded-change-type>
        </subscription-capabilities>
      </per-node-capabilities>
...
```

In the example above, the agent could provide notifications for data nodes under /ndg:garage node selector in operational datastore: on-change for state nodes, periodic for both configuration and state nodes.

Before subscribing to “yang-push”, the client should query “datastore-capabilities” (for details see RFC 9196: YANG Modules Describing Capabilities for Systems and Datastore Update Notifications).

18.3 Dynamic Subscriptions

18.3.1 Periodic subscriptions

Assuming:

```
<yp-capabilities xmlns="http://ndt-inc.com/yp-capabilities">
  <system-capabilities xmlns="urn:ietf:params:xml:ns:yang:ietf-system-capabilities">
    <datastore-capabilities>
      <datastore>operational</datastore>
      <per-node-capabilities>
        <node-selector>/ndg:garage</node-selector>
        <subscription-capabilities
               xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-capabilities">
          <max-nodes-per-update>50</max-nodes-per-update>
          <periodic-notifications-supported>
              config-changes state-changes
          </periodic-notifications-supported>
          <minimum-update-period>1200</minimum-update-period>
...
```

the agent could provide periodic notifications for all data nodes under `/ndg:garage` node selector in the operational datastore.

To receive automatic and continuous updates from a targeted subset of a datastore, a client can send a request for periodic updates:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
 <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
      xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="4">
    <n2:establish-subscription xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
      <yp:datastore xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">operational</yp:datastore>
      <yp:selection-filter-ref>ypf2</yp:selection-filter-ref>
      <yp:periodic>
        <yp:period>6000</yp:period>
        <yp:anchor-time>2024-02-01T12:00:00Z</yp:anchor-time>
      </yp:periodic>
    </n2:establish-subscription>
  </rpc>
```

On accepting the RPC, the agent returns the subscription ID in its response:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="4">
  <id xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">3</id>
</rpc-reply>
```

and the agent starts sending updates every 60 seconds (period is specified in centiseconds). The time of update is ‘anchored” to the date/time specified in the ‘anchor-time’ element. In this example updates will be sent on full minute.The update looks like:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <eventTime>2025-08-11T21:26:00Z</eventTime>
  <yp:push-update>
    <yp:id>3</yp:id>
    <yp:datastore-contents>
      <ndg:garageObjects xmlns:ndg="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
        <ndg:garageAddress>1 Front St.</ndg:garageAddress>
        <ndg:garageNumVehicles>2</ndg:garageNumVehicles>
        <ndg:garageCOLevel>1</ndg:garageCOLevel>
        <ndg:garageCOLevelRisingThreshold>50</ndg:garageCOLevelRisingThreshold>
        <ndg:garageCOLevelFallingThreshold>10</ndg:garageCOLevelFallingThreshold>
      </ndg:garageObjects>
    </yp:datastore-contents>
  </yp:push-update>
</notification>
```

18.3.2 On change subscriptions

Assuming:

```
<yp-capabilities xmlns="http://ndt-inc.com/yp-capabilities">
  <system-capabilities xmlns="urn:ietf:params:xml:ns:yang:ietf-system-capabilities">
    <datastore-capabilities>
      <datastore>operational</datastore>
      <per-node-capabilities>
        <node-selector>/ndg:garage</node-selector>
        <subscription-capabilities
               xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-capabilities">
          <max-nodes-per-update>50</max-nodes-per-update>
          <on-change-supported>state-changes</on-change-supported>
          <minimum-dampening-period>1200</minimum-dampening-period>
...
        </subscription-capabilities>
      </per-node-capabilities>
...
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

the agent could provide on-change notifications for changes in “config false“ data nodes under `/ndg:garage` node selector in the operational datastore.

To subscribe to changes in `/garage/garageObjects` subset the client could send:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    xmlns:n2="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications" message-id="9">
    <n2:establish-subscription xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
      <yp:datastore>operational</yp:datastore>
      <yp:datastore-xpath-filter>/ndg:garage/ndg:garageObjects</yp:datastore-xpath-filter>
      <yp:on-change xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <yp:dampening-period>2000</yp:dampening-period>
        <yp:sync-on-start>true</yp:sync-on-start>
        <yp:excluded-change>create</yp:excluded-change>
      </yp:on-change>
    </n2:establish-subscription>
  </rpc>
```

On accepting the RPC, the agent returns the subscription ID in its response:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="9">
  <id xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">4</id>
</rpc-reply>
```

and immediately “push-update” notification (in the request: **sync-on-start = true**):

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
              xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <eventTime>2025-08-11T21:37:49Z</eventTime>
  <yp:push-update>
    <yp:id>4</yp:id>
    <yp:datastore-contents>
      <ndg:garageObjects xmlns:ndg="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
        <ndg:garageNumVehicles>2</ndg:garageNumVehicles>
        <ndg:garageCOLevel>1</ndg:garageCOLevel>
      </ndg:garageObjects>
    </yp:datastore-contents>
  </yp:push-update>
</notification>
```

After that the agent will send notification only when the change happens. E.g. on the next sampling interval, garageCOLevel value is 20. The agent detects this change and sends “push-change-update” notification:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <eventTime>2025-08-11T21:38:39Z</eventTime>
  <yp:push-change-update>
    <yp:id>4</yp:id>
    <yp:datastore-changes>
      <yang-patch xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
        <patch-id>0</patch-id>
        <edit>
          <edit-id>1</edit-id>
          <operation>merge</operation>
          <target>/ndg:garageObjects/ndg:garageCOLevel</target>
          <value xmlns:n2="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
            <n2:garageCOLevel>20</n2:garageCOLevel>
          </value>
        </edit>
      </yang-patch>
    </yp:datastore-changes>
  </yp:push-change-update>
</notification>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

The change is detected by the agent by retrieving data for the filters specified (in the “yp-capabilities.xml”) and comparing to the previous values. It would be extremely CPU intensive to track changes in the datastore(s) by continuously retrieving data for the filters specified. So how does the agent know when the change happens?

The agent retains the result of the most recent retrieval operation for each “on-change” subscription, along with the timestamp when it occurred. The implementation must provide its own timestamp when a change occurs, enabling the agent to verify it and generate a notification if necessary.

The implementation of yang-module is in a class derived from DataObjGroup. E.g. in ndGarageMib project:

```
class IfData_NdGarageMib : public DataObjGrp
```

and the actual data is in:

```
class Data_NdGarageMib : public IfData_NdGarageMib
```

and this class is instantiated as a global variable:

```
Data_NdGarageMib data_ndg;
```

In the implemetation add:

```
data_ndg.SetLastChange()
```

after the monitored object/s is/are changed. Note that the time of changes of configuration objects done through netconf/restconf/… is automatically recorded. If the yang-push is configured for config-only objects and no cganges are done internally by the implementation code, then no explicit recording of the last change is necessary.

18.4 Configured Subscriptions

A configured subscription is one that has been established based on parameters defined in the configuration. The example below configures subscription with id 400, filter is ‘ypf2’ (I.e. datastore-xpath-filter: /ndg:garage/ndg:garageObjects, see Filters section), and datastore is “operational”.

```
<subscriptions xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications"
               xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push"
               xmlns:nnsn="http://ndt-inc.com/ns/ndt-nc-subnotifs">

    <subscription>
          <id>400</id>
          <yp:datastore>operational</yp:datastore>
          <yp:selection-filter-ref>ypf2</yp:selection-filter-ref>
          <transport>nnsn:netconf</transport>
          <receivers>
              <receiver>
                  <name>receiver1</name>
                  <nnsn:netconf-endpoint>home-tls-one</nnsn:netconf-endpoint>
              </receiver>
          </receivers>
          <yp:periodic>
              <yp:period>3000</yp:period>
              <yp:anchor-time>2025-03-18T12:00:00Z</yp:anchor-time>
          </yp:periodic>
   </subscription>
</subscriptions>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

There is one receiver: “recever1” with netconf-endpoint “home-tls-one” which is configured in the “netconf-server” section of the configuration:

```
<netconf-server xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-server"
                xmlns:ct="urn:ietf:params:xml:ns:yang:ietf-crypto-types"
                xmlns:cmn="urn:ietf:params:xml:ns:yang:ietf-tls-common"
                xmlns:x509c2n="urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name">

  <listen>
  . . .
  </listen>
  <call-home>
    <netconf-client>
      <name>chadmin</name>
        <endpoints>
          <endpoint>
            <name>home-tls-one</name>
            <tls>
               <tcp-client-parameters>
                  <remote-address>127.0.0.1</remote-address>
                   <remote-port>4335</remote-port>
                    ...
    </netconf-client>
  </call-home>
</netconf-server>
```

Based on the configuration above, notifications will be sent to 127.0.0.1:4335, provided that the call-home connection has been established.

Note that “ietf-subscribed-notifications” does not define transport and receiver objects to be used in the configuration, from RFC 8639, section 2.5.2. “Creating a Configured Subscription”:

```
Receiver-specific transport connectivity parameters MUST be configured via
transport-specific augmentations to this specification.
```

Augmentation for the receiver-specific parameters is defined in “**ndt-nc-subscribed-notifications**” yang module.

```
identity netconf {
  base sn:transport;
}

augment "/sn:subscriptions/sn:subscription/sn:receivers/sn:receiver" {
    when 'derived-from(../../transport, "nnsn:netconf")';
    leaf netconf-endpoint {
      type leafref {
        path "/ncs:netconf-server/ncs:call-home/ncs:netconf-client" +
             "/ncs:endpoints/ncs:endpoint/ncs:name";
      }
      mandatory true;
    }
}
```

Implementation of the “ndt-nc-subscribed-notifications” yang module is in libietfSubscribedNotifications.so library.

Based on the configuration example above, on startup the agent will “call-home”, and if this is successful, yang-push will start, and the client wil receive “subscription-started” notification:

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
              xmlns:sn="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
  <eventTime>2025-08-01T19:45:54Z</eventTime>
 <notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
               xmlns:sn="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
  <eventTime>2025-08-18T14:11:04Z</eventTime>
  <sn:subscription-started>
    <sn:id>400</sn:id>
    <sn:datastore-xpath-filter>/ndg:garage/ndg:garageObjects</sn:datastore-xpath-filter>
    <sn:datastore>operational</sn:datastore>
    <sn:transport xmlns:nnsn="http://ndt-inc.com/ns/ndt-nc-subnotifs">nnsn:netconf</sn:transport>
    <sn:encoding>encode-xml</sn:encoding>
  </sn:subscription-started>
</notification>
```

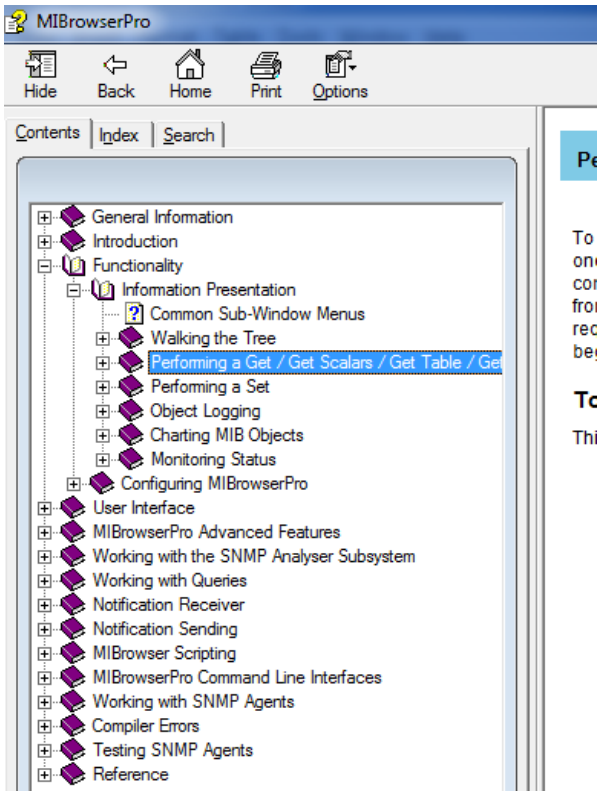
and later the subscribed notifications:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0"
              xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
  <eventTime>2025-08-18T14:11:30Z</eventTime>
  <yp:push-update>
    <yp:id>400</yp:id>
    <datastore-contents>
      <ndg:garageObjects xmlns:ndg="urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB">
        <ndg:garageAddress>1 Front St.</ndg:garageAddress>
          <ndg:garageNumVehicles>2</ndg:garageNumVehicles>
          <ndg:garageCOLevel>1</ndg:garageCOLevel>
          <ndg:garageCOLevelRisingThreshold>50</ndg:garageCOLevelRisingThreshold>
          <ndg:garageCOLevelFallingThreshold>10</ndg:garageCOLevelFallingThreshold>
        </ndg:garageObjects>
      </datastore-contents>
    </yp:push-update>
</notification>
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

19 SNMP

Use MIBrowser to access **testagent**. Please check MIBrowser help for more information.



This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

20 Code Generation

20.1 Code Generator

Using the NuDesign YANG code generator, the developer generates a C++ project that produces a data dll, from an input YANG document. The code provides a basis for providing instrumentation for the XML tree defined by the input document.

The resulting code is loadable from the NuDesign YANG Data Server. Access to this instrumentation is from the NuDesign Data (Demo) Server via an "Access Agent" (AA). The AA is Dynamic-Link Library also loaded by NuDesign Data (Demo) Server at start up. It is a protocol handler (any of NETCONF, RESTCONF, SNMP, CLI...).

The code generator itself is a pyang plugin called ndt3.pyc. The code generator plugin is dependent on version 2.4.0 of pyang. During installation, the installer checks for the presence of pyang 2.4.0 and if It doesn't exist, installs it.

20.2 ndt3.pyc Plugin

20.2.1 Plugin Command Line Options

The following generation options are available:

`-p DIRECTORY`	Where DIRECTORY is a path to a repository of YANG documents. (Note: this is a pyang option, not specific to the NDT YANG Code Generator).
`-f FORMAT`	The pyang format selector. FORMAT must be 'ndt'.
`--ndt-output=DIRECTORY`	Generate output in DIRECTORY.
`--ndt-template=TEMPLATE`	Template directory for the project files. The path should point to the ndt-template directory.
`--ndt-help`	Print help on usage of NDT YANG Code Generator and exit.
`--ndt-version`	Print the version of NDT YANG Code Generator and exit.
`--ndt-verbose`	Verbose mode: Print detailed debug messages.
`--ndt-debug`	Print debug messages. Redundant if verbose mode is on.
`--ndt-ignore-errors`	Ignore errors from validation.
`--ndt-sdkdir=SDKDIR`	Directory of where SDK is installed.
`--ndt-hidetimestamps`	Prevents time stamp information being added to the source code files, to make analysis of difference ('diff') on the project files, easier.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

20.3 Generating Code with the plugin

To create a project, you may invoke the NDT code generator via pyang as follows:

```
pyang <-p path> yang-file -f ndt <ndt options>
```

See the previous sections for the specifics of the various options.

The following is a potential invocation given the default SDK installation directories.

E.g. This example builds a project from the ND-GARAGE-MIB.yang found in the installation's **schema** directory.

(Note: replace <user> below with a directory specific to your system.)

```
pyang      -p /opt/ndConf-x.y-buildnnn/ndconf/schema/ND-GARAGE-MIB.yang \
           -f ndt \
           --ndt-output=/home/<user>/src \
           --ndt-sdkdir=/opt/ndConf-x.y-buildnnn/ndconf \
           --ndt-template=/opt/ndConf-x.y-buildnnn/ndconf/CodeGenerator/template
```

20.4 ndtGen

'ndtGen' is an executable that simplifies invocation of 'ndt3.pyc'.

20.4.1 ndtGen Synopsis

ndtGen *[-p <path>] [--ndt-projschema <path>] [--ndt-output <path>]*
[--ndt-template<path>] [--ndt-gensubprojs] [--ndt-f <document.yang>] …

20.4.2 ndtGen Command Line Options

The following generation options are available:

`-p <path>`	An optional parameter. Where **path** is a directory path list to one or more repositories of YANG documents. Note: this is a pyang option, not specific to the NDT YANG Code Generator, so use with caution. You may be replacing default content for it and unexpected errors may occur. Generally you should use '—-ndt-projschema' instead.
`--ndt-projschema <path>`	An optional parameter. Where **path** is a directory path to a secondary repository, perhaps project specific, YANG documents.
`--ndt-output <path>`	Specifies an optional path to the directory to put the created project files. The default is the current directory.
`--ndt-template <path>`	Optional parameter. Where **path** is a directory path to the template repository.
`--ndt-gensubprojs`	Optional generate sub projects. Sub projects are determined by the **import** statements in the project YANG document. Default: do not.
`--ndt-f <file>`	Optionally specifies the project YANG document. If the file is not in the current directory the schema directory is checked for existence of the file

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

20.4.3 Generating code with ndtGen

To create a project, you may invoke the NDT code generator via **ndtGen** as follows:

```
ndtGen --ndt-f /opt/ndConfBuilder-x.yy-buildnnn/ndconf/schema/ND-GARAGE-MIB.yang
```

The above command line would generate the ndGarageMib project in the current directory, assuming the same as the previous section, /home/<user>/src. See the man page for ndtGen. For more information.

20.5 The Generated Project

From the above example project, the contents of the project would be as follows.

```
Linux/
     makefile
     StdAfx.h
yang/
     gen-info.txt
     ND-GARAGE-MIB.yang
ndGarageMib_data.cpp
ndGarageMib_data_if.h
ndGarageMib_notif.cpp
ndGarageMib_stx.h
m2y_ndGarageMib.xml
ndGarageMib_data.h
ndGarageMib_impl.cpp
ndGarageMib_notif.h
ndGarageMib_template.cpp
ndGarageMib.cpp
ndGarageMib_data_if.cpp
ndGarageMib_impl.h
ndGarageMib_stx.cpp
ndGarageMib_template.h
m2y_ndGarageMib.xml
ndGarageMib-startup-cfg-demo.xml
```

20.6 Building the Project

Change to the ' Linux ' directory of the project and to build it, from type:

```
$ make
```

Additionally, there is a make command line option, MODE. MODE can be used to specify debug or release builds. E.g.

```
$ make MODE=release
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

20.7 Testing the Project

When configuring ndconf.xml to test your project, you have a choice to make. Namely this where to put you project's '.so'. In the simple case, which is when there is only one project '.so', it Is sufficient to use the "path=" construct when creating the entry for your datadll and leave it in the project area.

However, if the yang document for your project uses the '**import**' statement to import other modules, you usually need to specify the '**`--ndt-gensubprojs`**' to successfully build a project for your document. In this case, a project typically has one or more sub-projects in addition to the main project, meaning it is possible that one or more of the '**.so**'s is dependent on another in the project.

When this occurs, load dependencies need to be satisfied in order to successfully load a particular **datadll**. To simplify testing, we suggest that you copy all project '**.so**'s into the **/lib64** directory of the ndCONF Builder installation. Each of the start up scripts (E.g. **teststart** & **ndconfsvrstart**), will ensure that this '**lib64**' directory is on the library load path and there should be no difficulty loading a module.

This also means that you do not have to specific a '**path**' in the '<**datadll**>' specification for any of the project's '**.so**'s.

Note: When building to deploying a finished project, you may have to adjust the build to accommodate the deployment target configuration, usually in the link specification of the projects that have dependencies.

20.8 Building with the Source SDK

If you have the source SDK installed, building the produced project with it is as simple as changing the first line of the project makefile. Below is the typical first line of the every makefile (given the ndConfBuilder SDK's version is 1.3 and this SDK is installed in the default location):

```
NUDESIGNDIR=/opt/ndConfBuilder-3.0-build002/ndconf
```

Place a comment on this line and create a new line that defines the variable NUDESIGNDIR so that it contains the path of the ndconf directory of the source SDK. Give the default installation location of the source SDK install and user name 'nudesign' and the Source SDK's version is 1.8, this would look like

```
NUDESIGNDIR=/home/nudesign/ndConfSrcSDK-3.0-build002/ndconf
```

So the first two lines of the makefile now contains:

```
#NUDESIGNDIR=/opt/ndConfBuilder-3.0-build002/ndconf
NUDESIGNDIR=/home/nudesign/ndConfSrcSDK-3.0-build002/ndconf
```

Now just build as given in the previous sections.

20.9 Configuring `testagent` to use the Project's shared object output

To use the generated code's shared object, it needs to be made available to the NuDesign Data (Demo) Server. This is done via the testagent's configuration file,

```
/opt/ndConfBiulder-x.y-buildnnn/ndconf/testagent/config/ndconf.xml.
```

In the <datadlls> section,

```
<datadlls>
```

You can disable the provide configuration for ndGarageMib by commenting it out.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
<!--<datadll name="garage" dll="libndGarageMib.so.0" ></datadll>-->
```

Now add a new <datadll> section for the generated and built shared object.

```
<datadll name="garage" path="/home/<user>/src/ndGarageMib/Linux" dll="libndGarageMib.so"></datadll>
```

Note the use of the “path” parameter, to override the default path for shared objects in the SDK and is assumed to be under some <user>’s src directory.

20.10 Enabling SNMP access to a Project's shared object output

To continue with the ndGarageMIB project example, in order to enable SNMP access to its objects the following files are required and are included in the ndCONF Eval Agent as follows:

1. The project's **SNMP MIB module** (*/ndconf/mib2xml/mib/NDGarageV2.mib) and a corresponding **YANG module** (*/ndconf/schema/ND-GARAGE-MIB.yang), where MIB tree object's OIDs map into leafs defined in the YANG model (see Section 18 - Appendix). The YANG object’s definitions contain MIB tree OID annotation directives.

    1. Such a MIB tree OID annotated YANG module can be auto-generated from an existing MIB module using NuDesign’s Visual MIBuilder tool (an RFC6643 compliant process) as was the case with the included ndGarageMIB (and rcvlan) projects.

    2. For a pre-existing YANG module, a SNMP MIB module can be manually created, using NuDesign’s Visual MIBuilder tool, with the objects defined in the YANG model. Once completed, MIB tree OID annotation directives should be manually added to the original YANG module's object definitions before the project’s generation with the ndCONF Builder tool.

2. The MIB to YANG mapping file that binds the MIB objects to appropriate leafs of a YANG module. The ndCONF Builder generates this **m2y_ndGarageMIB.xml** OID mapping file automatically for any MIB OID annotated YANG modules, during the source code generation, see Section 14.5. The **m2y_ndGarageMIB.xml** needs to be copied into */ndconf/testagent/config directory.
3. The initial start-up data file, the **ndGarageMib-startup-cfg-demo.xml** file, created by the build process in ndGarageMIB Project's directory. Merge its content into the agent default **startup-cfg.xml** file, present in the */ndconf/testagent/config directory, see the section 4.3 Startup Data of this User Guide.
4. A **compiled MIB file**, mib_ND-GARAGE-MIB.xml file, created using the provided mib_2_xml compiler tool. The compiled MIB file provides the SNMP Access Agent with the MIB's SMI properties.

    The mib_2_xml compiler tool is in */ndconf/mib2xml directory. Insert your MIB in the */ndconf/mib2xml/MIB directory. Type from the */ndconf/mib2xml directory:

    ```
    # ./mib2xml  MIB/NDGarageV2.mib
    ```

5. The mib_ND-GARAGE-MIB.xml file shall be generated into */ndconf/mib2xml/out directory and needs to be copied into */ndconf/testagent/config directory.

    The same process applies to user's custom MIB / YANG module pair when creating the YANG Datastore extension for access from any supported protocol Access Agent.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

20.11 Other Resources

20.11.1 Project Design Notes

The installation contains two PDF documents that provide design documentation for the two datadll projects included in the install. Both are installed in the ./doc directory and are called

- NuDesign ndCONF Builder – CodeGen Tech Reference.pdf
- NuDesign ndCONF Builder – UsingExistingImpl Tech Reference.pdf

20.11.2 Class Reference

The installation contains references for the underlying C++ class libraries. Each class reference is contained in a PDF file. These are installed in the ./doc directory. Also in the ./doc directory is an html file called **Main.htm**, which serves as a front end to these PDF files.

20.11.3 man pages

Included with the installation are several Linux ‘man’ pages. These pages are available from a command line, for the following topics:

- **testagent**
- **teststart**
- **ndncsub**
- **mib2xml**
- **ndtGen**[44]
- **ndtrdsclient**
- **ndncsub**
- **ncclient**
- **mib2xml**
- **ndconfsvr**
- **ndconfsvrstart**

There is one for each of the executables included by the installation.

20.11.4 HTML formatted “man” pages

There are also HTML formatted man pages included in the SDK. These are accessible from ‘**main-man.html**’ located in the installation ‘doc' directory.

44 Only available on the full release SDK.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

20.11.5 Default TCP Ports

This product uses a number of TCP/UDP ports. The default ports used by the various access agents are listed below.

- 161 (UDP), SNMP AA, SNMP service port.
- 9010 (TCP), RESTCONF AA, FCGI service port.
- 9522 (TCP), NETCONF AA, SSH pipe communication port
- 9523 (TCP), CLI AA, local CLI communication port.
- 9524 (TCP), RESTCONF AA, stream notification service (“push”) port
- 6513 (TCP), NETCONF AA, listen port for TLS connection

These are all configurable, See the individual sections in this document for more information.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

21 Products for ndCONF Development

The **ndCONF Agent Development Studio** contains the following products, downloadable from NuDesign's secure site.

21.1 ndCONF Builder

The ndCONF Builder generates C++ loadable data handlers (LDH) extensions projects from YANG documents. LDHs are used with the YANG Datastore Server to extend the data model. The server is included with the ndCONF SDK.

The YANG Datastore Server and the ndCONF Builder are included with the ndCONF SDK installation program.

The ndCONF Builder also has its own, standalone installation program that comes with the following documents:

- ndCONF User Guide - this User Guide, with the chapter 13 covering the ndCONF Builder operations
- NuDesign ndCONF Builder – CodeGen Tech Reference.pdf
- NuDesign ndCONF Builder – UsingExistingImpl Tech Reference.pdf

21.2 ndCONF SDK

The ndCONF SDK contains the extensible YANG Datastore Server, NETCONF / YANG SDKs with basic CLI access and APIs to the protocol Access Agents and LDH extensions, the YANG Datastore's object providers.

Its installation comes with the following document, in addition to the C++ class references identified in section 14.2.

- ndCONF User Guide - this User Guide, it describes the installation and use of ndCONF SDK in a process of developing ndCONF Agent for user selected YANG modules.

21.3 NETCONF, RESTCONF, CLI and SNMP Access Agents

Each protocol Access Agent when combined with YANG Datastore Server provides a full management and configuration Agent providing access to YANG objects via NETCONF, CLI, RESTCONF or SNMP client applications.

The Access Agent's installs come with the C++ class reference documentation identified in section 14.2.

21.4 Visual MIBuilder

The SNMP Acess Agent is accompanied by NuDesign’s Visual MIBuilder. This product provides a visual user interface for designing the ASN.1 MIBs used by SNMP. It also provides for exporting a MIB as a YANG document. This exported YANG document contains OID annotation directives used by ndCONF Builder when generating extension sub-agent projects for use with SNMP, NETCONF or RESTCONF protocols.

Please see the section “**Enabling SNMP access to a Project's shared object output**” for more details.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

22 About NuDesign

NuDesign Technologies, Inc., headquartered in Toronto, ON, Canada, specializes in providing NETCONF, RESTCONF, SNMP, CLI and web agent & manager development applications, tools, libraries and consulting services to developers and networking OEMs worldwide. The benefits of deploying NuDesign's management software technologies are reliable, low risk, well integrated and quick-to-market solutions, supporting management operations over IPv4 / IPv6 networks in carrier, campus and enterprise settings.

Our focus is on industry standard management protocols such as NETCONF. RESTCONF, SNMP and protocols using HTTP transport. Our customers are Original Equipment Manufacturers, System Integrators, Service Providers and End Users worldwide.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

23 Appendix

23.1 'ND-GARAGE-MIB' Yang module

```
/*
This Yang module was created using NuDesign Technologies' Visual MIBuilder (Ver 5.2).
MIB File : NDGarageV2.mib
Module : ND-GARAGE-MIB
*/

module ND-GARAGE-MIB {

  namespace "urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB";
  prefix "nd-garage";

  import ietf-yang-types          { prefix "yang"; }
  import ietf-yang-smiv2          { prefix "smiv2"; }
  import SNMPv2-TC                { prefix "snmpv2-tc"; }

  organization
    "NuDesign Technologies, Inc.";
  contact
    "NuDesign Technologies, Inc.
    Web site: www.ndt-inc.com
    Email:    contact@ndt-inc.com";
  description
    "The MIB module for managing car garage.";

  revision 2004-07-09 {
    description
        "New version released that is to be used across all NuDesign Products. sh";
  }
  revision 2002-10-11 {
    description
        "Added objects garageCOLevel, garageCOLevelRisingThreshold and
        garageCOLevelFallingThreshold and notifications
        coLevelRisingAlarm and coLevelFallingAlarm.
        Updated notif-groups and compliance. bl";
  }
  revision 2002-01-17 {
    description
        "Modified the access of vehicleModel to ReadCreate. sh";
  }
  revision 2001-03-21 {
    description
        "Addedd notifications and conformance statements. bl";
  }
  revision 2000-05-01 {
    description
        "Initial version. bl";
  }

  container garage {
    config false;

    container garageObjects {

        smiv2:oid "1.3.6.1.4.1.4761.99.11.1";
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
        leaf garageAddress {
            type snmpv2-tc:DisplayString {
                length "0..32";
            }
            description
                "Garage address. For example: 10 St, SomePlace.";
            smiv2:max-access "read-write";
            smiv2:oid "1.3.6.1.4.1.4761.99.11.1.1";
        }
        leaf garageNumVehicles {
            type int32;
            description
                "Number of vehicles currently parked in the garage.";
            smiv2:max-access "read-only";
            smiv2:oid "1.3.6.1.4.1.4761.99.11.1.2";
        }
        leaf garageCOLevel {
            type int32;
            description
                "CO toxicity level in garage (airborne concentration level expressed in
                parts per million or PPM).";
            smiv2:max-access "read-only";
            smiv2:oid "1.3.6.1.4.1.4761.99.11.1.4";
        }
        leaf garageCOLevelRisingThreshold {
            type int32;
            description
                "Level of CO toxicity level in garage (in parts per million or PPM)
                considered dangerous for humans. 400 PPM causes serious headache
                after 1-2 hours of exposure. This level is life threatening after 3 hours.
                When the current sampled value is greater than or equal to
                this threshold, and the value at the last sampling interval
                was less than this threshold, a single event (coLevelRisingAlarm)
                will be generated.
                After a rising event is generated, another such event will not be
                generated until the sampled value falls below this threshold and
                reaches the garageCOLevelFallingThreshold.";
            smiv2:max-access "read-write";
            smiv2:oid "1.3.6.1.4.1.4761.99.11.1.5";
        }
        leaf garageCOLevelFallingThreshold {
            type int32;
            description
                "Level of CO toxicity level in garage (in parts per million or PPM)
                considered harmless for humans. 35 PPM is maximum exposure allowed by
                OSHA in the workplace over an eight hour period.
                When the current sampled value is less than or equal to this threshold,
                and the value at the last sampling interval was greater than this threshold,
                a single event (coLevelFallingAlarm) will be generated.
                After a falling event is generated, another such event will not be
                generated until the sampled value rises above this threshold and
                reaches the garageCOLevelRisingThreshold.";
            smiv2:max-access "read-write";
            smiv2:oid "1.3.6.1.4.1.4761.99.11.1.6";
        }
    }

    /*
       container vehiclesTable {
            description
                "Table of vehicles parked in the garage.";
            smiv2:oid "1.3.6.1.4.1.4761.99.11.1.3";
    */
            list vehiclesEntry {
                key "vehicleIndex";
                description
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
                "Row in the vehiclesTable.";

            smiv2:oid "1.3.6.1.4.1.4761.99.11.1.3.1";

            leaf vehicleIndex {
                type int32;
                description
                    "Index into vehiclesTable.";
                smiv2:max-access "not-accessible";
                smiv2:oid "1.3.6.1.4.1.4761.99.11.1.3.1.1";
            }
            leaf vehicleLicencePlate {
                type snmpv2-tc:DisplayString {
                    length "6..7";
                }
                description
                    "License plate of the vehicle, for example 123XYZ or ABCD987.";
                smiv2:max-access "read-create";
                smiv2:oid "1.3.6.1.4.1.4761.99.11.1.3.1.2";
            }
            leaf vehicleModel {
                type snmpv2-tc:DisplayString;
                description
                    "Make and model of the vehicle; e.g 'NuDesign speedmaster'.
                    NuDesign does not manufacture vehicles (yet). This is just
                    for demonstration purpose.";
                smiv2:max-access "read-create";
                smiv2:oid "1.3.6.1.4.1.4761.99.11.1.3.1.3";
            }
        }
/*
  }
*/
 }

 notification notPaidAlarm {
   description
       "The SNMP trap that is generated when the vehicle leaves garage
       but did not pay the bill.";

   smiv2:oid "1.3.6.1.4.1.4761.99.11.2.1";

   container object-1 {
       leaf vehicleIndex {
           type leafref {
               path "/nd-garage-mib:ND-GARAGE-MIB/nd-garage-mib:vehiclesTable" +
                    "/nd-garage-mib:vehiclesEntry/nd-garage-mib:vehicleIndex";
           }
       }
       leaf vehicleLicencePlate {
           type leafref {
               path "/nd-garage-mib:ND-GARAGE-MIB/nd-garage-mib:vehiclesTable" +
                    "/nd-garage-mib:vehiclesEntry/nd-garage-mib:vehicleLicencePlate";
           }
       }
   }
 }

 notification coLevelRisingAlarm {
   description
       "The SNMP trap that is generated when CO toxicity level in garage
       crosses its rising threshold (garageCOLevelRisingThreshold).";

   smiv2:oid "1.3.6.1.4.1.4761.99.11.2.2";

   container object-1 {
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
        leaf garageCOLevel {
            type leafref {
                path "/nd-garage-mib:ND-GARAGE-MIB/nd-garage-mib:garageObjects/nd-garage-
mib:garageCOLevel";
            }
        }
    }
  }

  notification coLevelFallingAlarm {
    description
        "The SNMP trap that is generated when CO toxicity level in garage
        crosses its falling threshold (garageCOLevelFallingThreshold).";

    smiv2:oid "1.3.6.1.4.1.4761.99.11.2.3";

    container object-1 {
        leaf garageCOLevel {
            type leafref {
                path "/nd-garage-mib:ND-GARAGE-MIB/nd-garage-mib:garageObjects/nd-garage-
mib:garageCOLevel";
            }
        }
    }
  }

  smiv2:alias "nuDesignTech" {
      smiv2:oid "1.3.6.1.4.1.4761";
  }

  smiv2:alias "ndtExperimental" {
      smiv2:oid "1.3.6.1.4.1.4761.99";
  }

  smiv2:alias "garage" {
      smiv2:oid "1.3.6.1.4.1.4761.99.11";
  }

  smiv2:alias "garageObjects" {
      smiv2:oid "1.3.6.1.4.1.4761.99.11.1";
  }

  smiv2:alias "garageEvents" {
      smiv2:oid "1.3.6.1.4.1.4761.99.11.2";
  }

  smiv2:alias "garageConformance" {
      smiv2:oid "1.3.6.1.4.1.4761.99.11.3";
  }

  smiv2:alias "garageCompliances" {
      smiv2:oid "1.3.6.1.4.1.4761.99.11.3.1";
  }

  smiv2:alias "garageGroups" {
      smiv2:oid "1.3.6.1.4.1.4761.99.11.3.2";
  }
}
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

23.2 'ND-GARAGE-MIB' MIB module

```
-- ************************************************************************
-- *
-- * Copyright (c) 2000-2025 NuDesign Technologies, Inc. All Rights Reserved.
-- *
-- * Module Name:  ND-GARAGE-MIB
-- *
-- * Description:  Defines objects for managing garage for motor vehicles
-- *
-- ************************************************************************

ND-GARAGE-MIB DEFINITIONS ::= BEGIN

IMPORTS
 NOTIFICATION-TYPE, OBJECT-TYPE, MODULE-IDENTITY,
 enterprises, Integer32
     FROM SNMPv2-SMI
 NOTIFICATION-GROUP, OBJECT-GROUP, MODULE-COMPLIANCE
     FROM SNMPv2-CONF
 RowStatus, DisplayString
     FROM SNMPv2-TC;

garage MODULE-IDENTITY
 LAST-UPDATED "201303140000Z"
 ORGANIZATION
     "NuDesign Technologies, Inc."
 CONTACT-INFO
     "NuDesign Technologies, Inc.
      Web site: www.ndt-inc.com
      Email:    contact@ndt-inc.com"
 DESCRIPTION
     "The MIB module for managing car garage."

 REVISION "201303140000Z"
 DESCRIPTION
     "Administrative changes. dp"

 REVISION "200407091226Z"
 DESCRIPTION
     "New version released that is to be used across all NuDesign Products. sh"

 REVISION "200210110900Z"
 DESCRIPTION
     "Added objects garageCOLevel, garageCOLevelRisingThreshold and
            garageCOLevelFallingThreshold and notifications
            coLevelRisingAlarm and coLevelFallingAlarm.
      Updated notif-groups and compliance. bl"

 REVISION "200201171512Z"
 DESCRIPTION
     "Modified the access of vehicleModel to ReadCreate. sh"

 REVISION "200103210000Z"
 DESCRIPTION
     "Addedd notifications and conformance statements. bl"

 REVISION "200005010000Z"
 DESCRIPTION
     "Initial version. bl"
::= { ndtExperimental 11 }

nuDesignTech         OBJECT IDENTIFIER ::= { enterprises 4761 }
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
ndtExperimental      OBJECT IDENTIFIER ::= { nuDesignTech 99 }
garageObjects        OBJECT IDENTIFIER ::= { garage 1 }
garageEvents         OBJECT IDENTIFIER ::= { garage 2 }
garageConformance    OBJECT IDENTIFIER ::= { garage 3 }

-- Conformance Information

garageCompliances    OBJECT IDENTIFIER ::= { garageConformance 1 }
garageGroups         OBJECT IDENTIFIER ::= { garageConformance 2 }

garageAddress  OBJECT-TYPE
 SYNTAX     DisplayString (SIZE(0..32))
 MAX-ACCESS read-write
 STATUS     current
 DESCRIPTION
     "Garage address. For example: 10 St, SomePlace."
 ::= { garageObjects 1 }

garageNumVehicles  OBJECT-TYPE
 SYNTAX     Integer32
 MAX-ACCESS read-only
 STATUS     current
 DESCRIPTION
     "Number of vehicles currently parked in the garage."
 ::= { garageObjects 2 }

vehiclesTable  OBJECT-TYPE
 SYNTAX SEQUENCE OF VehiclesEntry
 MAX-ACCESS not-accessible
 STATUS     current
 DESCRIPTION
     "Table of vehicles parked in the garage."
 ::= { garageObjects 3 }

vehiclesEntry  OBJECT-TYPE
 SYNTAX  VehiclesEntry
 MAX-ACCESS not-accessible
 STATUS     current
 DESCRIPTION
     "Row in the vehiclesTable."
 INDEX { vehicleIndex }
 ::= { vehiclesTable 1 }

VehiclesEntry ::= SEQUENCE {
 vehicleIndex
     Integer32,
 vehicleLicencePlate
     DisplayString,
 vehicleModel
     DisplayString,
 vehicleStatus
     RowStatus
}

vehicleIndex  OBJECT-TYPE
 SYNTAX     Integer32
 MAX-ACCESS not-accessible
 STATUS     current
 DESCRIPTION
     "Index into vehiclesTable."
 ::= { vehiclesEntry 1 }

vehicleLicencePlate  OBJECT-TYPE
 SYNTAX     DisplayString (SIZE(6..7))
 MAX-ACCESS read-create
 STATUS     current
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
  DESCRIPTION
      "Licence plate of the vehicle, for example 123XYZ or ABCD987."
  ::= { vehiclesEntry 2 }

vehicleModel  OBJECT-TYPE
  SYNTAX     DisplayString
  MAX-ACCESS read-create
  STATUS     current
  DESCRIPTION
      "Make and model of the vehicle; e.g 'NuDesign speedmaster'.
        NuDesign does not manufacture vehicles (yet). This is just
        for demonstration purpose."
  ::= { vehiclesEntry 3 }

vehicleStatus  OBJECT-TYPE
  SYNTAX     RowStatus
  MAX-ACCESS read-create
  STATUS     current
  DESCRIPTION
      "Status of this row."
  ::= { vehiclesEntry 4 }

garageCOLevel  OBJECT-TYPE
  SYNTAX     Integer32
  MAX-ACCESS read-only
  STATUS     current
  DESCRIPTION
      "CO toxicity level in garage (airborne concentration level expressed in
      parts per million or PPM)."
  ::= { garageObjects 4 }

garageCOLevelRisingThreshold  OBJECT-TYPE
  SYNTAX     Integer32
  MAX-ACCESS read-write
  STATUS     current
  DESCRIPTION
      "Level of CO toxicity level in garage (in parts per million or PPM)
      considered dangerous for humans. 400 PPM causes serious headache
      after 1-2 hours of exposure. Thislevel is life threatening after 3 hours.
      When the current sampled value is greater than or equal to
            this threshold, and the value at the last sampling interval
            was less than this threshold, a single event (coLevelRisingAlarm)
            will be generated.
            After a rising event is generated, another such event will not be
            generated until the sampled value falls below this threshold and
            reaches the garageCOLevelFallingThreshold."
  ::= { garageObjects 5 }

garageCOLevelFallingThreshold  OBJECT-TYPE
  SYNTAX     Integer32
  MAX-ACCESS read-write
  STATUS     current
  DESCRIPTION
      "Level of CO toxicity level in garage (in parts per million or PPM)
      considered harmless for humans. 35 PPM is maximum exposure allowed by
      OSHA in the workplace over an eight hour period.
      When the current sampled value is less than or equal to this threshold,
      and the value at the last sampling interval was greater than this threshold,
      a single event (coLevelFallingAlarm) will be generated.
            After a falling event is generated, another such event will not be
            generated until the sampled value rises above this threshold and
            reaches the garageCOLevelRisingThreshold."
  ::= { garageObjects 6 }

-- Events
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
notPaidAlarm  NOTIFICATION-TYPE
 OBJECTS { vehicleLicencePlate }
 STATUS     current
 DESCRIPTION
     "The SNMP trap that is generated when the vehicle leaves garage
      but did not pay the bill."
 ::= { garageEvents 1 }

coLevelRisingAlarm  NOTIFICATION-TYPE
 OBJECTS { garageCOLevel }
 STATUS     current
 DESCRIPTION
     "The SNMP trap that is generated when CO toxicity level in garage
           crosses its rising threshold (garageCOLevelRisingThreshold)."
 ::= { garageEvents 2 }

coLevelFallingAlarm  NOTIFICATION-TYPE
 OBJECTS { garageCOLevel }
 STATUS     current
 DESCRIPTION
     "The SNMP trap that is generated when CO toxicity level in garage
           crosses its falling threshold (garageCOLevelFallingThreshold)."
 ::= { garageEvents 3 }

-- Compliance statements

garageCompliance  MODULE-COMPLIANCE
 STATUS     current
 DESCRIPTION
     "The compliance statement for agents which
      implement the ND-GARAGE-MIB. Note that garageNotifGroup2 is
     unconditionally optional for compliance to this MIB module."
 MODULE
     MANDATORY-GROUPS { garageBasicGroup,
                garageNotifGroup1 }
 ::= { garageCompliances 1 }

-- Units of compliance

garageBasicGroup  OBJECT-GROUP
 OBJECTS { garageAddress,
     garageNumVehicles,
     vehicleLicencePlate,
     vehicleModel,
     vehicleStatus,
     garageCOLevel,
     garageCOLevelRisingThreshold,
     garageCOLevelFallingThreshold }
 STATUS     current
 DESCRIPTION
     "A collection of objects providing the info for the garage."
 ::= { garageGroups 1 }

garageNotifGroup1  NOTIFICATION-GROUP
 NOTIFICATIONS { coLevelRisingAlarm,
     coLevelFallingAlarm }
 STATUS     current
 DESCRIPTION
     "The threshold notifications."
 ::= { garageGroups 2 }

garageNotifGroup2  NOTIFICATION-GROUP
 NOTIFICATIONS { notPaidAlarm }
 STATUS     current
 DESCRIPTION
     "The not-paid notification."
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
  ::= { garageGroups 3 }
END

-- This MIB was created using NuDesign Technology's Visual MIBuilder (Ver 4.4).
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

23.3 'RAPID-CITY-MIB' (rcvlan) Yang Module

```
module rcvlan {
  namespace "urn:ietf:params:xml:ns:yang:smiv2:rcvlan";
  prefix "rcv";

  import ietf-yang-smiv2          { prefix "smiv2"; }
  import ietf-inet-types          { prefix "inet"; }
  import SNMPv2-TC                { prefix "snmpv2-tc"; }

  organization
    "Avaya ...";
  contact
    "...";
  description
    "...";

  revision 2016-09-10 {
  }
  typedef InterfaceIndex {
     type int32; // {length "1..2147483647"; }
     description
         "Port IfIndex.";
  }
  typedef PortSet {
    type binary {
        length "0..255";
    }
    description
        "The string is 88 octets long, for a total of 704 bits. Each bit
        corresponds to a port, as represented by its ifIndex value . When a
        bit has the value one(1), the corresponding port is a member of the
        set. When a bit has the value zero(0), the corresponding port is not
        a member of the set. The encoding is such that the most significant
        bit of octet #1 corresponds to ifIndex 0, while the least significant
        bit of octet #88 corresponds to ifIndex 703. In order to accommodate
        future uses of this the string has a max size of 255 bytes.";
  }

  typedef RcVlanIdOrNone {
    type int32 {
        range "1..4094";
    }
    description
        "Range of VLAN IDs supported for application";
  }

  typedef VrfIdentifier {
    type uint32 {
        range "0..512";
    }
    description
        "Virtual Router Identifier.
        VRFID 0 is reserved for the Administrative VRF
        and cannot be used to create VRF's.
        ";
  }

  container vlanModule {
    config true;
    //container rcVlanTable {
    //    description
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
//        "A list of Virtual LAN entries. The number of entries
//        is given by rcVlanNumVlans.";
//    smiv2:oid "1.3.6.1.4.1.2272.1.3.2";

    list vlan { // rcVlanEntry {
        key "vlanId";
        description
            "Entry containing configuration information for a
            particular Virtual LAN.

            The relationship between the various port sets in a
            VLAN Entry are :

            o The set of ports defined by PortMembers must be a
            subset of the set of ports in the underlying STG of
            the VLAN.

            o The bitwise AND of PortMembers and NotAllowToJoin must
            be the empty set.

            o The bitwise OR of PortMembers and NotAllowToJoin must
            be the set of ports in the underlying STG of the VLAN.

            o The set of ports defined by StaticMembers must be a
            subset of the set of ports defined by PortMembers.

            o The bitwise XOR of PortMembers and StaticMembers defines
            the set of dynamic (potential) members of the VLAN.

            o The set of ports defined by ActiveMembers must be a subset
            of the set of ports defined by PortMembers.
            ";

        smiv2:oid "1.3.6.1.4.1.2272.1.3.2.1";

        leaf vlanId { // rcVlanId {
            type rcv:RcVlanIdOrNone;
            description
                "A value that uniquely identifies the Virtual LAN
                associated with this entry. This value corresponds
                to the lower 12 bits in the IEEE 802.1Q VLAN Tag.";
            smiv2:max-access "read-only";
            smiv2:oid "1.3.6.1.4.1.2272.1.3.2.1.1";
        }
        leaf vlanName { // rcVlanName {
            type snmpv2-tc:DisplayString {
                length "0..64";
            }
            description
                "An administratively-assigned name for this VLAN.";
            smiv2:max-access "read-write";
            smiv2:oid "1.3.6.1.4.1.2272.1.3.2.1.2";
        }
        leaf stgId {  // rcVlanStgId {
            type int32 {
                range "0..128";
            }
            description
                "Indicates the Spanning Tree Group (STG) used by
                this VLAN to determine the state of its ports.
                If this VLAN is not associated with any STG, this
                value should be set to zero.";
            smiv2:defval "1";
            smiv2:max-access "read-write";
            smiv2:oid "1.3.6.1.4.1.2272.1.3.2.1.9";
        }
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
            leaf vlanType { // rcVlanType {
                type enumeration {
                    enum "byPort" { value "1"; }
                    enum "byIpSubnet" { value "2"; }
                    enum "byProtocolId" { value "3"; }
                    enum "bySrcMac" { value "4"; }
                    enum "byDstMcast" { value "5"; }
                    enum "bySvlan" { value "6"; }
                    enum "byIds" { value "7"; }
                    enum "spbm-bvlan" { value "11"; }
                    enum "private" { value "13"; }
                }
                description
                    "The type of VLAN, distinguished according to the
                    policy used to define its port membership.
                    VSP9000 does not support bySvlan(6) and ByIds(7).";
                smiv2:max-access "read-write";
                smiv2:oid "1.3.6.1.4.1.2272.1.3.2.1.10";
            }
            leaf vlanPortMembers { // rcVlanPortMembers {
                type rcv:PortSet;
                description
                    "The set of ports that are members (static or
                    dynamic) of this VLAN.";
                smiv2:max-access "read-write";
                smiv2:oid "1.3.6.1.4.1.2272.1.3.2.1.11";
            }
            leaf vlanProtocolId { // rcVlanProtocolId {
                type enumeration {
                    enum "none" { value "0"; }
                    enum "ip" { value "1"; }
                    enum "ipx802dot3" { value "2"; }
                    enum "ipx802dot2" { value "3"; }
                    enum "ipxSnap" { value "4"; }
                    enum "ipxEthernet2" { value "5"; }
                    enum "appleTalk" { value "6"; }
                    enum "decLat" { value "7"; }
                    enum "decOther" { value "8"; }
                    enum "sna802dot2" { value "9"; }
                    enum "snaEthernet2" { value "10"; }
                    enum "netBios" { value "11"; }
                    enum "xns" { value "12"; }
                    enum "vines" { value "13"; }
                    enum "ipV6" { value "14"; }
                    enum "usrDefined" { value "15"; }
                    enum "rarp" { value "16"; }
                    enum "pPPoE" { value "17"; }
                }
                description
                    "The protocol identifier of this VLAN. This value
                    is meaningful only if rcVlanType is equal to
                    byProtocolId(3). For other VLAN types it should
                    have the value none(0).";
                smiv2:max-access "read-write";
                smiv2:oid "1.3.6.1.4.1.2272.1.3.2.1.15";
            }

        list ip { // rcIpAddrTable {
            key "addr";
            description
            "The table of addressing information relevant to
            this entity's IP addresses.";

                leaf ifIndex {
                    type rcv:InterfaceIndex;
                    description
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
            "The index value which uniquely identifies the
            interface to which this entry is applicable.  The
            interface identified by a particular value of this
            index is the same interface as identified by the
            same value of ifIndex.";
        smiv2:max-access "read-only";
        smiv2:oid "1.3.6.1.4.1.2272.1.8.2.1.1";
    }
    leaf addr {  // rcIpAdEntAddr {
        type inet:ipv4-address;
        description
            "The IP address to which this entry's addressing
            information pertains.";
        smiv2:max-access "read-only";
        smiv2:oid "1.3.6.1.4.1.2272.1.8.2.1.2";
    }
    leaf netMask { // rcIpAdEntNetMask {
        type inet:ipv4-address;
        description
            "The subnet mask associated with the IP address of
            this entry.  The value of the mask is an IP
            address with all the network bits set to 1 and all
            the hosts bits set to 0.";
        smiv2:max-access "read-write";
        smiv2:oid "1.3.6.1.4.1.2272.1.8.2.1.3";
    }
    leaf bcastAddrFormat{ // rcIpAdEntBcastAddrFormat {
        type enumeration {
            enum "zeros" { value "0"; }
            enum "ones" { value "1"; }
        }
        description
            "The IP broadcast address format used on this
            interface.";
        smiv2:max-access "read-only";
        smiv2:oid "1.3.6.1.4.1.2272.1.8.2.1.4";
    }
    leaf reasmMaxSize { // rcIpAdEntReasmMaxSize {
        type int32 {
            range "0..65535";
        }
        description
            "The size of the largest IP datagram which this
            entity can re-assemble from incoming IP fragmented
            datagrams received on this interface.";
        smiv2:max-access "read-only";
        smiv2:oid "1.3.6.1.4.1.2272.1.8.2.1.5";
    }
    /*
    leaf rcIpAdEntVlanId {
        type int32 {
            range "0..4096";
        }
        description
            "A value that uniquely identifies the Virtual LAN
            associated with this entry. This value corresponds
            to the lower 12 bits in the IEEE 802.1Q VLAN Tag.";
        smiv2:max-access "read-write";
        smiv2:oid "1.3.6.1.4.1.2272.1.8.2.1.7";
    }
    */
    leaf macOffset { // rcIpAdEntMacOffset {
        type int32 {
            range "0..511";
        }
        description
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
                    "Used to translate the ip address into mac address.
                    The system has 512 mac addresses of which 0-507 are reserved for
                    Boardwalk box. 508-511 is reserved for MG. One can either
                    mention a mac offset while configuring an ip on the
                    vlan or it can be alloted by the system within the above range.";
                smiv2:max-access "read-write";
                smiv2:oid "1.3.6.1.4.1.2272.1.8.2.1.9";
            }
            leaf vrfId { // rcIpAdEntVrfId {
                type rcv:VrfIdentifier;
                description
                    "The VrfId used in this table is to associate VLANs
                    or Brouter ports to a VRF after the creation of VLANs
                    or Brouter ports.
                    VRFID 0 is reserved for the Administrative VRF.";
                smiv2:max-access "read-create";
                smiv2:oid "1.3.6.1.4.1.2272.1.8.2.1.12";
            }
         }
        }
  }
 }
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

23.4 'RAPID-CITY-MIB' (rcvlan) MIB Module

```
RAPID-CITY DEFINITIONS ::= BEGIN

IMPORTS
 OBJECT-TYPE, MODULE-IDENTITY, enterprises,
 iso, Unsigned32, Integer32, IpAddress
     FROM SNMPv2-SMI
 TEXTUAL-CONVENTION, RowStatus, MacAddress,
 TruthValue, DisplayString
     FROM SNMPv2-TC;

-- ISSUES
--
-- None outstanding

rapidCity MODULE-IDENTITY
 LAST-UPDATED "201205150000Z"
 ORGANIZATION
     "Bay Networks, Rapid-City Group"
 CONTACT-INFO
     "
     Edwin Tsang
                      Postal: Bay Networks, Inc.
                            4401 Great America Parkway
                            Santa Clara, CA 95052-8185

                         Tel: 408-495-6159
                         Fax: 408-495-5215
                      E-mail:edwin_tsang@@baynetworks.com
     "
 DESCRIPTION
     "Enterprise MIB for the Accelar product family."
::= { enterprises 2272 }


-- May 15, 2012
--
-- Local defines to avoid having to pull-in other RFC's.
--

BridgeId ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "An identifier that identifies the Bridge Id"
 SYNTAX  OCTET STRING (SIZE(8))


EnableValue ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "Enable/Disable value."
 SYNTAX  INTEGER {
     enable (1),
     disable (2)
 }


IdList ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "An identifier for a list of Ids."
 SYNTAX  OCTET STRING
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the "NuDesign Software License Agreement" associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
--
-- This variable type is used through out the Rapid-City enterprise
-- MIB to denote the standard ifIndex in mib-2.
--

InterfaceIndex ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "Port IfIndex."
 SYNTAX  Integer32


InterfaceIndexOrZero ::= TEXTUAL-CONVENTION
 DISPLAY-HINT   "d"
 STATUS     current
 DESCRIPTION
     "This textual convention is an extension of the
                 InterfaceIndex convention.  The latter defines a greater
                 than zero value used to identify an interface or interface
                 sub-layer in the managed system.  This extension permits the
                 additional value of zero.  the value zero is object-specific
                 and must therefore be defined as part of the description of
                 any object which uses this syntax.  Examples of the usage of
                 zero might include situations where interface was unknown,
                 or when none or all interfaces need to be referenced."
 SYNTAX  Integer32 (0..2147483647)


IpIsisPlsbNodeNickName ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "Plsb Node Nick name."
 SYNTAX  OCTET STRING (SIZE(3))


Ipv6NextHdr ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "The ipv6 next header to look for."
 SYNTAX  INTEGER {
     hop-by-hop (0),
     icmpv4 (1),
     igmpv4 (2),
     ipInIp (4),
     tcp (6),
     egp (8),
     udp (17),
     ipv6 (41),
     routing (43),
     frag (44),
     rsvp (46),
     ipsecESP (50),
     ipsecAh (51),
     icmpv6 (58),
     noNxtHdr (59),
     destOptions (60),
     undefined (255)
 }


--
-- The string is 88 octets long, for a total of 704 bits. Each bit
-- corresponds to a port, as represented by its ifIndex value . When a
-- bit has the value one(1), the corresponding port is a member of the
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
-- set. When a bit has the value zero(0), the corresponding port is not
-- a member of the set. The encoding is such that the most significant
-- bit of octet #1 corresponds to ifIndex 0, while the least significant
-- bit of octet #88 corresponds to ifIndex 703."
--

PortSet ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "The string is 88 octets long, for a total of 704 bits. Each bit
              corresponds to a port, as represented by its ifIndex value . When a
              bit has the value one(1), the corresponding port is a member of the
              set. When a bit has the value zero(0), the corresponding port is not
              a member of the set. The encoding is such that the most significant
              bit of octet #1 corresponds to ifIndex 0, while the least significant
              bit of octet #88 corresponds to ifIndex 703. In order to accommodate
              future uses of this the string has a max size of 255 bytes."
 SYNTAX  OCTET STRING (SIZE(0..255))


RcLongDisplayString ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "The semantics are identical to the standard DisplayString
                      in RFC1213 and RFC2579, except for the longer length that
                      is allowed with this TC. "
 SYNTAX  OCTET STRING (SIZE(0..65535))


RcVlanIdOrNone ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "Range of VLAN IDs supported for application"
 SYNTAX  Integer32 (1..4094)


Timeout ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "An identifier that identifies the timeout value."
 SYNTAX  Integer32


VrfIdentifier ::= TEXTUAL-CONVENTION
 STATUS     current
 DESCRIPTION
     "Virtual Router Identifier.
                  VRFID 0 is reserved for the Administrative VRF
                  and cannot be used to create VRF's.
                 "
 SYNTAX  Unsigned32 (0..512)

org                 OBJECT IDENTIFIER ::= { iso 3 }
dod                 OBJECT IDENTIFIER ::= { org 6 }

--
-- Additions to resolve some SNMP V2 dependencies so we don't
-- have to pull-in a lot of mib modules.
--

internet            OBJECT IDENTIFIER ::= { dod 1 }
snmpV2              OBJECT IDENTIFIER ::= { internet 6 }
snmpModules         OBJECT IDENTIFIER ::= { snmpV2 3 }
snmpMIB             OBJECT IDENTIFIER ::= { snmpModules 1 }
snmpMIBObjects      OBJECT IDENTIFIER ::= { snmpMIB 1 }
snmpTraps           OBJECT IDENTIFIER ::= { snmpMIBObjects 5 }
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
--
-- Enterprise specific MIB groups
--

rcMgmt               OBJECT IDENTIFIER ::= { rapidCity 1 }

-- APLS device end

rcSystem             OBJECT IDENTIFIER ::= { rcMgmt 1 }
rcTftp               OBJECT IDENTIFIER ::= { rcMgmt 2 }
rcVlan               OBJECT IDENTIFIER ::= { rcMgmt 3 }
rcIp                 OBJECT IDENTIFIER ::= { rcMgmt 8 }

-- VLAN Table

rcVlanNumVlans  OBJECT-TYPE
 SYNTAX     Integer32 (1..128)
 MAX-ACCESS read-only
 STATUS     current
 DESCRIPTION
     "The number of VLANs currently defined in the switch."
 ::= { rcVlan 1 }

rcVlanTable  OBJECT-TYPE
 SYNTAX SEQUENCE OF RcVlanEntry
 MAX-ACCESS not-accessible
 STATUS     current
 DESCRIPTION
     "A list of Virtual LAN entries. The number of entries
                          is given by rcVlanNumVlans."
 ::= { rcVlan 2 }

rcVlanEntry  OBJECT-TYPE
 SYNTAX  RcVlanEntry
 MAX-ACCESS not-accessible
 STATUS     current
 DESCRIPTION
     "Entry containing configuration information for a
                      particular Virtual LAN.

                      The relationship between the various port sets in a
                      VLAN Entry are :

                      o The set of ports defined by PortMembers must be a
                        subset of the set of ports in the underlying STG of
                        the VLAN.

                      o The bitwise AND of PortMembers and NotAllowToJoin must
                        be the empty set.

                      o The bitwise OR of PortMembers and NotAllowToJoin must
                        be the set of ports in the underlying STG of the VLAN.

                      o The set of ports defined by StaticMembers must be a
                        subset of the set of ports defined by PortMembers.

                      o The bitwise XOR of PortMembers and StaticMembers defines
                        the set of dynamic (potential) members of the VLAN.

                      o The set of ports defined by ActiveMembers must be a subset
                        of the set of ports defined by PortMembers.
                      "
 INDEX { rcVlanId }
 ::= { rcVlanTable 1 }
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
RcVlanEntry ::= SEQUENCE {
 rcVlanId
     RcVlanIdOrNone,
 rcVlanName
     DisplayString,
 rcVlanColor
     Integer32,
 rcVlanHighPriority
     TruthValue,
 rcVlanRoutingEnable
     TruthValue,
 rcVlanIfIndex
     InterfaceIndex,
 rcVlanAction
     INTEGER,
 rcVlanResult
     INTEGER,
 rcVlanStgId
     Integer32,
 rcVlanType
     INTEGER,
 rcVlanPortMembers
     PortSet,
 rcVlanActiveMembers
     PortSet,
 rcVlanStaticMembers
     PortSet,
 rcVlanNotAllowToJoin
     PortSet,
 rcVlanProtocolId
     INTEGER,
 rcVlanSubnetAddr
     IpAddress,
 rcVlanSubnetMask
     IpAddress,
 rcVlanAgingTime
     Integer32,
 rcVlanMacAddress
     MacAddress,
 rcVlanRowStatus
     RowStatus
}

rcVlanId  OBJECT-TYPE
 SYNTAX     RcVlanIdOrNone
 MAX-ACCESS read-only
 STATUS     current
 DESCRIPTION
     "A value that uniquely identifies the Virtual LAN
                          associated with this entry. This value corresponds
                          to the lower 12 bits in the IEEE 802.1Q VLAN Tag."
 ::= { rcVlanEntry 1 }

rcVlanName  OBJECT-TYPE
 SYNTAX     DisplayString (SIZE(0..64))
 MAX-ACCESS read-write
 STATUS     current
 DESCRIPTION
     "An administratively-assigned name for this VLAN."
 ::= { rcVlanEntry 2 }

rcVlanColor  OBJECT-TYPE
 SYNTAX     Integer32 (0..32)
 MAX-ACCESS read-write
 STATUS     current
 DESCRIPTION
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
      "An administratively-assigned color code for this
                           VLAN. The value of this object is used by the VLAN
                           Manager GUI tool to select a color when it draws
                           this VLAN on the screen."
  ::= { rcVlanEntry 3 }

rcVlanHighPriority  OBJECT-TYPE
  SYNTAX     TruthValue
  MAX-ACCESS read-write
  STATUS     obsolete
  DESCRIPTION
      "A flag to note whether frames in this VLAN should
                           be assigned a high switching priority."
  DEFVAL  { false }
  ::= { rcVlanEntry 4 }

rcVlanRoutingEnable  OBJECT-TYPE
  SYNTAX     TruthValue
  MAX-ACCESS read-write
  STATUS     obsolete
  DESCRIPTION
      "A flag to note whether IP routing is enabled in
                           this VLAN."
  DEFVAL  { false }
  ::= { rcVlanEntry 5 }

rcVlanIfIndex  OBJECT-TYPE
  SYNTAX     InterfaceIndex
  MAX-ACCESS read-only
  STATUS     current
  DESCRIPTION
      "When rcVlanRoutingEnable is set to true(1), this
                           value indicates the 'logical' ifIndex assigned to
                           this VLAN.  Otherwise, this value is meaningless
                           and should be set to zero."
  ::= { rcVlanEntry 6 }

rcVlanAction  OBJECT-TYPE
  SYNTAX     INTEGER {
      none (1),
      flushMacFdb (2),
      flushArp (3),
      flushIp (4),
      flushDynMemb (5),
      all (6),
      flushSnoopMemb (7),
      triggerRipUpdate (8),
      flushSnoopMRtr (9),
      flushIpRsmltEdgePeer (10)
  }
  MAX-ACCESS read-write
  STATUS     current
  DESCRIPTION
      "VLAN related actions."
  DEFVAL  { none }
  ::= { rcVlanEntry 7 }

rcVlanResult  OBJECT-TYPE
  SYNTAX     INTEGER {
      none (1),
      inProgress (2),
      success (3),
      fail (4)
  }
  MAX-ACCESS read-only
  STATUS     current
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
    DESCRIPTION
        "The result from the last VLAN action."
    DEFVAL  { none }
    ::= { rcVlanEntry 8 }

  rcVlanStgId  OBJECT-TYPE
    SYNTAX     Integer32 (0..128)
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "Indicates the Spanning Tree Group (STG) used by
                            this VLAN to determine the state of its ports.
                            If this VLAN is not associated with any STG, this
                            value should be set to zero."
    DEFVAL  { 1 }
    ::= { rcVlanEntry 9 }

  rcVlanType  OBJECT-TYPE
    SYNTAX     INTEGER {
        byPort (1),
        byIpSubnet (2),
        byProtocolId (3),
        bySrcMac (4),
        byDstMcast (5),
        bySvlan (6),
        byIds (7),
        spbm-bvlan (11),
        private (13)
    }
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The type of VLAN, distinguished according to the
                            policy used to define its port membership.
                            VSP9000 does not support bySvlan(6) and ByIds(7)."
    ::= { rcVlanEntry 10 }

  rcVlanPortMembers  OBJECT-TYPE
    SYNTAX     PortSet
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The set of ports that are members (static or
                            dynamic) of this VLAN."
    ::= { rcVlanEntry 11 }

  rcVlanActiveMembers  OBJECT-TYPE
    SYNTAX     PortSet
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The set of ports that are currently active in
                            this VLAN. Active ports include all static ports
                            and any dynamic ports where the VLAN policy was
                            met."
    ::= { rcVlanEntry 12 }

  rcVlanStaticMembers  OBJECT-TYPE
    SYNTAX     PortSet
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "The set of ports that are static members of this
                            VLAN. A static member of a VLAN is always active
                            and is never aged out."
    ::= { rcVlanEntry 13 }
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
rcVlanNotAllowToJoin  OBJECT-TYPE
 SYNTAX     PortSet
 MAX-ACCESS read-write
 STATUS     current
 DESCRIPTION
     "The set of ports that are not allowed to become
                         members of this VLAN."
 ::= { rcVlanEntry 14 }

rcVlanProtocolId  OBJECT-TYPE
 SYNTAX     INTEGER {
     none (0),
     ip (1),
     ipx802dot3 (2),
     ipx802dot2 (3),
     ipxSnap (4),
     ipxEthernet2 (5),
     appleTalk (6),
     decLat (7),
     decOther (8),
     sna802dot2 (9),
     snaEthernet2 (10),
     netBios (11),
     xns (12),
     vines (13),
     ipV6 (14),
     usrDefined (15),
     rarp (16),
     pPPoE (17)
 }
 MAX-ACCESS read-write
 STATUS     current
 DESCRIPTION
     "The protocol identifier of this VLAN. This value
                         is meaningful only if rcVlanType is equal to
                         byProtocolId(3). For other VLAN types it should
                         have the value none(0)."
 ::= { rcVlanEntry 15 }

rcVlanSubnetAddr  OBJECT-TYPE
 SYNTAX     IpAddress
 MAX-ACCESS read-write
 STATUS     current
 DESCRIPTION
     "The IP subnet address of this VLAN. This value
                         is meaningful only if rcVlanType is equal to
                         byIpSubnet(2). For other VLAN types it should
                         have the value 0.0.0.0."
 ::= { rcVlanEntry 16 }

rcVlanSubnetMask  OBJECT-TYPE
 SYNTAX     IpAddress
 MAX-ACCESS read-write
 STATUS     current
 DESCRIPTION
     "The IP subnet mask of this VLAN. This value
                         is meaningful only if rcVlanType is equal to
                         byIpSubnet(2). For other VLAN types it should
                         have the value 0.0.0.0."
 ::= { rcVlanEntry 17 }

rcVlanAgingTime  OBJECT-TYPE
 SYNTAX     Integer32 (0|10..1000000)
 MAX-ACCESS read-write
 STATUS     current
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
    DESCRIPTION
        "The timeout period (in seconds) used for aging
                            out dynamic members of this VLAN. This field is
                            only relevant for policy-based VLANs."
    DEFVAL  { 600 }
    ::= { rcVlanEntry 18 }

  rcVlanMacAddress  OBJECT-TYPE
    SYNTAX     MacAddress
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The MAC address assigned to the virtual router
                            interface of this VLAN. This field is meaningful
                            only if rcVlanRoutingEnable is equal to true(1)."
    ::= { rcVlanEntry 19 }

  rcVlanRowStatus  OBJECT-TYPE
    SYNTAX     RowStatus
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
        "Used to create/delete entries in the rcVlanTable."
    ::= { rcVlanEntry 20 }

  -- IP Address Table

  rcIpAddrTable  OBJECT-TYPE
    SYNTAX SEQUENCE OF RcIpAddrEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The table of addressing information relevant to
                            this entity's IP addresses.

                            This table is identical to the ipAddrTable in
                            MIB2 except the columns rcIpAdEntIfIndex and
                            rcIpAdEntAddr are reversed.  Also, the table
                            is indexed by both these variables."
    ::= { rcIp 2 }

  rcIpAddrEntry  OBJECT-TYPE
    SYNTAX  RcIpAddrEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The addressing information for one of this
                            entity's IP addresses."
    INDEX { rcIpAdEntIfIndex,
        rcIpAdEntAddr }
    ::= { rcIpAddrTable 1 }

  RcIpAddrEntry ::= SEQUENCE {
    rcIpAdEntIfIndex
        InterfaceIndex,
    rcIpAdEntAddr
        IpAddress,
    rcIpAdEntNetMask
        IpAddress,
    rcIpAdEntBcastAddrFormat
        INTEGER,
    rcIpAdEntReasmMaxSize
        Integer32,
    rcIpAdEntRowStatus
        RowStatus,
    rcIpAdEntVlanId
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
      Integer32,
   rcIpAdEntBrouterPort
      TruthValue,
   rcIpAdEntMacOffset
      Integer32,
   rcIpAdEntIfType
      INTEGER,
   rcIpAdEntVrfId
      VrfIdentifier
  }

  rcIpAdEntIfIndex  OBJECT-TYPE
   SYNTAX     InterfaceIndex
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
      "The index value which uniquely identifies the
                           interface to which this entry is applicable.  The
                           interface identified by a particular value of this
                           index is the same interface as identified by the
                           same value of ifIndex."
   ::= { rcIpAddrEntry 1 }

  rcIpAdEntAddr  OBJECT-TYPE
   SYNTAX     IpAddress
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
      "The IP address to which this entry's addressing
                           information pertains."
   ::= { rcIpAddrEntry 2 }

  rcIpAdEntNetMask  OBJECT-TYPE
   SYNTAX     IpAddress
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
      "The subnet mask associated with the IP address of
                           this entry.  The value of the mask is an IP
                           address with all the network bits set to 1 and all
                           the hosts bits set to 0."
   ::= { rcIpAddrEntry 3 }

  rcIpAdEntBcastAddrFormat  OBJECT-TYPE
   SYNTAX     INTEGER {
      zeros (0),
      ones (1)
   }
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
      "The IP broadcast address format used on this
                           interface."
   ::= { rcIpAddrEntry 4 }

  rcIpAdEntReasmMaxSize  OBJECT-TYPE
   SYNTAX     Integer32 (0..65535)
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
      "The size of the largest IP datagram which this
                           entity can re-assemble from incoming IP fragmented
                           datagrams received on this interface."
   ::= { rcIpAddrEntry 5 }

  rcIpAdEntRowStatus  OBJECT-TYPE
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
    SYNTAX     RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "Used to create/delete entries"
    ::= { rcIpAddrEntry 6 }

  rcIpAdEntVlanId  OBJECT-TYPE
    SYNTAX     Integer32 (0..4096)
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "A value that uniquely identifies the Virtual LAN
                              associated with this entry. This value corresponds
                              to the lower 12 bits in the IEEE 802.1Q VLAN Tag."
    ::= { rcIpAddrEntry 7 }

  rcIpAdEntBrouterPort  OBJECT-TYPE
    SYNTAX     TruthValue
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Used to indicate whether this entry correponds
              to a brouter port (as oppose to a routable VLAN).
               This value cannot be changed after the row is
              created."
    ::= { rcIpAddrEntry 8 }

  rcIpAdEntMacOffset  OBJECT-TYPE
    SYNTAX     Integer32 (0..511)
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "Used to translate the ip address into mac address.
                               The system has 512 mac addresses of which 0-507 are reserved for
                               Boardwalk box. 508-511 is reserved for MG. One can either
                               mention a mac offset while configuring an ip on the
                               vlan or it can be alloted by the system within the above range."
    ::= { rcIpAddrEntry 9 }

  rcIpAdEntIfType  OBJECT-TYPE
    SYNTAX     INTEGER {
        circuitLessIP (1),
        brouterPort (2),
        vlan (3)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The address entry IfType indicates the type of the
          interface. In the case of circuit-less-IP there is
                              no association with any ports. In other words, it can
                              be thought of as a virtual interface. The value
                              'brouter port' indicates this interface is associated
                              with a specific physical port. The value 'vlan'
                              indicates a logical port that contains one or physical
                              ports/MLT ports."
    ::= { rcIpAddrEntry 10 }

  rcIpAdEntVrfId  OBJECT-TYPE
    SYNTAX     VrfIdentifier
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "The VrfId used in this table is to associate VLANs
             or Brouter ports to a VRF after the creation of VLANs
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
          or Brouter ports.
          VRFID 0 is reserved for the Administrative VRF."
  ::= { rcIpAddrEntry 12 }

-- FlowContorl Group

rcIpFlowTable  OBJECT-TYPE
  SYNTAX SEQUENCE OF RcIpFlowEntry
  MAX-ACCESS not-accessible
  STATUS     current
  DESCRIPTION
      "IpFlow Table.  This table contain entries that
            correspond to RSVP records in the switching
            fabric Address Resolution table.

            A connection that has a matching entry in this
                           table will be processed with a higher priority
                           than connections that do not have an entry."
  ::= { rcIp 3 }

rcIpFlowEntry  OBJECT-TYPE
  SYNTAX  RcIpFlowEntry
  MAX-ACCESS not-accessible
  STATUS     current
  DESCRIPTION
      "Proprietary flow control variables."
  INDEX { rcIpFlowSrcIpAddress,
      rcIpFlowSrcIpPort,
      rcIpFlowDstIpAddress,
      rcIpFlowDstIpPort,
      rcIpFlowProtocol }
  ::= { rcIpFlowTable 1 }

RcIpFlowEntry ::= SEQUENCE {
  rcIpFlowSrcIpAddress
      IpAddress,
  rcIpFlowSrcIpPort
      Integer32,
  rcIpFlowDstIpAddress
      IpAddress,
  rcIpFlowDstIpPort
      Integer32,
  rcIpFlowProtocol
      INTEGER,
  rcIpFlowRowStatus
      RowStatus
}

rcIpFlowSrcIpAddress  OBJECT-TYPE
  SYNTAX     IpAddress
  MAX-ACCESS read-only
  STATUS     current
  DESCRIPTION
      "The source IP address of an IP packet."
  ::= { rcIpFlowEntry 1 }

rcIpFlowSrcIpPort  OBJECT-TYPE
  SYNTAX     Integer32 (0..65535)
  MAX-ACCESS read-only
  STATUS     current
  DESCRIPTION
      "The source port of an IP packet.  A zero value
                           in this field is used as a wildcard value."
  DEFVAL  { 0 }
  ::= { rcIpFlowEntry 2 }
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
rcIpFlowDstIpAddress  OBJECT-TYPE
 SYNTAX     IpAddress
 MAX-ACCESS read-only
 STATUS     current
 DESCRIPTION
     "The destination IP address of an IP packet."
 ::= { rcIpFlowEntry 3 }

rcIpFlowDstIpPort  OBJECT-TYPE
 SYNTAX     Integer32 (0..65535)
 MAX-ACCESS read-only
 STATUS     current
 DESCRIPTION
     "The destination port of an IP packet.  A zero
                          value in this field is used as a wildcard value."
 DEFVAL  { 0 }
 ::= { rcIpFlowEntry 4 }

rcIpFlowProtocol  OBJECT-TYPE
 SYNTAX     INTEGER {
     ip (4),
     tcp (6),
     udp (17)
 }
 MAX-ACCESS read-only
 STATUS     current
 DESCRIPTION
     "The protocol type of an IP packet.  A zero value
                          in this field is used as a wildcard value."
 DEFVAL  { ip }
 ::= { rcIpFlowEntry 5 }

rcIpFlowRowStatus  OBJECT-TYPE
 SYNTAX     RowStatus
 MAX-ACCESS read-write
 STATUS     current
 DESCRIPTION
     "Status of entry."
 ::= { rcIpFlowEntry 6 }
END
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

23.5 Capabilities

The following capabilities are available in the ndCONF server:

```
urn:ietf:params:netconf:base:1.0
urn:ietf:params:netconf:base:1.1
urn:ietf:params:netconf:capability:writable-running:1.0
urn:ietf:params:netconf:capability:candidate:1.0
urn:ietf:params:netconf:capability:confirmed-commit:1.0
urn:ietf:params:netconf:capability:confirmed-commit:1.1
urn:ietf:params:netconf:capability:rollback-on-error:1.0
urn:ietf:params:netconf:capability:validate:1.0
urn:ietf:params:netconf:capability:validate:1.1
urn:ietf:params:netconf:capability:startup:1.0
urn:ietf:params:netconf:capability:url:1.0
urn:ietf:params:netconf:capability:xpath:1.0
urn:ietf:params:netconf:capability:notification:1.0
urn:ietf:params:netconf:capability:interleave:1.0
urn:ietf:params:netconf:capability:partial-lock:1.0
urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=explicit
urn:ietf:params:xml:ns:yang:ietf-inet-types?module=ietf-inet-types&revision=2013-07-15
urn:ietf:params:xml:ns:yang:ietf-yang-types?module=ietf-yang-types&revision=2013-07-15
urn:ietf:params:xml:ns:yang:ietf-yang-smiv2?module=ietf-yang-smiv2&revision=2012-06-22
urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-TC?module=SNMPv2-TC&revision=1999-04-01
urn:ietf:params:xml:ns:yang:ietf-netconf-notifications?module=ietf-netconf-notifications&revision=2011-08-07
urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring?module=ietf-netconf-monitoring&revision=2010-10-04
urn:ietf:params:xml:ns:yang:ietf-netconf-acm?module=ietf-netconf-acm&revision=2011-10-04
http://ndt-inc.com/ns/ndt-mgmt?module=ndt-mgmt&revision=2020-08-01
http://ndt-inc.com/ns/ndt-tags?module=ndt-tags&revision=2021-11-25
urn:ietf:params:xml:ns:yang:iana-crypt-hash?module=iana-crypt-hash&revision=2014-08-06&features=crypt-hash-md5,
                crypt-hash-sha-256,crypt-hash-sha-512
urn:ietf:params:xml:ns:yang:ietf-x509-cert-to-name?module=ietf-x509-cert-to-name&revision=2014-12-10
urn:ietf:params:xml:ns:netconf:notification:1.0?module=notifications&revision=2015-12-14
urn:ietf:params:xml:ns:netmod:notification?module=nc-notifications&revision=2008-07-14
urn:ietf:params:xml:ns:yang:ietf-crypto-types?module=ietf-crypto-types&revision=2021-05-18
urn:ietf:params:xml:ns:yang:ietf-keystore?module=ietf-keystore&revision=2021-05-18&
                features=central-keystore-supported, local-definitions-supported
urn:ietf:params:xml:ns:yang:ietf-truststore?module=ietf-truststore&revision=2021-05-18&
                features=central-truststore-supported, local-definitions-supported,certificates,public-keys
urn:ietf:params:xml:ns:yang:ietf-tcp-common?module=ietf-tcp-common&revision=2021-05-18&features=keepalives-supported
urn:ietf:params:xml:ns:yang:ietf-tcp-client?module=ietf-tcp-client&revision=2021-05-18&
                features=local-binding-supported, tcp-client-keepalives,proxy-connect,socks5-gss-api,
                socks5-username-password
urn:ietf:params:xml:ns:yang:ietf-tcp-server?module=ietf-tcp-server&revision=2021-05-18&features=tcp-server-keepalives
urn:ietf:params:xml:ns:yang:ietf-tls-common?module=ietf-tls-common&revision=2021-05-18&features=tls-1_2,tls-1_3,
                tls-ecc, tls-dhe,tls-3des,tls-gcm,tls-sha2
urn:ietf:params:xml:ns:yang:ietf-tls-server?module=ietf-tls-server&revision=2021-05-18&
                features=tls-server-hello-params-config, tls-server-keepalives,server-ident-x509-cert,
                client-auth-supported,client-auth-x509-cert
urn:ietf:params:xml:ns:yang:ietf-ssh-common?module=ietf-ssh-common&revision=2021-05-18&features=ssh-ecc,
                ssh-x509-certs, ssh-dh-group-exchange,ssh-ctr,ssh-sha2
urn:ietf:params:xml:ns:yang:ietf-ssh-server?module=ietf-ssh-server&revision=2021-05-18&
                features=ssh-server-transport-params-config, ssh-server-keepalives,local-users-supported,
                userauth-publickey,userauth-password,userauth-hostbased,userauth-none
```

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.

```
urn:ietf:params:xml:ns:yang:ietf-netconf-server?module=ietf-netconf-server&revision=2021-05-18&
                features=ssh-call-home,tls-listen,tls-call-home
urn:ietf:params:xml:ns:yang:ietf-snmp?module=ietf-snmp&revision=2014-12-10&features=notification-filter,proxy
http://ndt-inc.com/ns/ndt-snmp?module=ndt-snmp&revision=2020-04-30
urn:ietf:params:xml:ns:yang:smiv2:SNMP-FRAMEWORK-MIB?module=SNMP-FRAMEWORK-MIB&revision=2002-10-14
urn:ietf:params:xml:ns:yang:smiv2:SNMPv2-MIB?module=SNMPv2-MIB&revision=2002-10-16
urn:ietf:params:xml:ns:yang:ietf-restconf?module=ietf-restconf&revision=2017-01-26
http://ndt-inc.com/ns/ndt-rest?module=ndt-rest&revision=2022-08-17
urn:ietf:params:xml:ns:yang:ietf-restconf-monitoring?module=ietf-restconf-monitoring&revision=2017-01-26
urn:ietf:params:xml:ns:yang:ietf-yang-library?module=ietf-yang-library&revision=2016-06-21
urn:ietf:params:xml:ns:yang:ietf-interfaces?module=ietf-interfaces&revision=2014-05-08&features=if-mib
urn:ietf:params:xml:ns:yang:iana-if-type?module=iana-if-type&revision=2014-05-08
http://openconfig.net/yang/openconfig-ext?module=openconfig-extensions&revision=2018-10-17
http://openconfig.net/yang/types/yang?module=openconfig-yang-types&revision=2018-11-21
http://openconfig.net/yang/openconfig-types?module=openconfig-types&revision=2018-11-21
http://openconfig.net/yang/interfaces?module=openconfig-interfaces&revision=2018-11-21
urn:ietf:params:xml:ns:yang:smiv2:SYSLOG-TC-MIB?module=SYSLOG-TC-MIB&revision=2009-03-30
urn:ietf:params:xml:ns:yang:smiv2:SYSLOG-MSG-MIB?module=SYSLOG-MSG-MIB&revision=2009-08-13
urn:ietf:params:xml:ns:yang:smiv2:ND-GARAGE-MIB?module=ND-GARAGE-MIB&revision=2004-07-09
urn:ietf:params:xml:ns:yang:smiv2:rcvlan?module=rcvlan&revision=2016-09-10
```

This list is associated with the RESTCONF access agent:

```
urn:ietf:params:restconf:capability:depth:1.0
urn:ietf:params:restconf:capability:fields:1.0
urn:ietf:params:restconf:capability:filter:1.0
urn:ietf:params:restconf:capability:replay:1.0
urn:ietf:params:restconf:capability:defaults:1.0?basic-mode=report-all
```

This list varies with the configuration of the server, in particular with the datadlls loaded by the configuration. The above depicts the datadlls for **openconfig-interfaces** and the two buildable module projects, “**ND-GARAGE-MIB**” and “**rcvlan**”.

If the ietf-interfaces configuration is loaded, then the following entries would not be depicted.

```
urn:ietf:params:xml:ns:yang:ietf-interfaces?module=ietf-interfaces&revision=2018-02-20
urn:ietf:params:xml:ns:yang:iana-if-type?module=iana-if-type&revision=2017-01-19
http://openconfig.net/yang/openconfig-ext?module=openconfig-extensions&revision=2018-10-17
http://openconfig.net/yang/types/yang?module=openconfig-yang-types&revision=2018-11-21
http://openconfig.net/yang/openconfig-types?module=openconfig-types&revision=2018-11-21
http://openconfig.net/yang/interfaces?module=openconfig-interfaces&revision=2018-11-21
```

Instead the following two entries would appear:

```
urn:ietf:params:xml:ns:yang:ietf-interfaces?module=ietf-interfaces&revision=2014-05-08
urn:ietf:params:xml:ns:yang:iana-if-type?module=iana-if-type&revision=2014-05-08
```

You may notice that the modules ietf-interfaces and iana-if-type exist in both configurations (though with different revisions). This is due to openconfigInterfaces being “dependent” of these modules.[45]

45 Care should be take to not use the wrong modules in the two configurations as there is overlap in the SNMP implementation provided by both.

This manual and the associated NuDesign Software are provided on an "As Is" basis. By using them, you are consenting to the Terms of the “NuDesign Software License Agreement” associated with the Product or Product Evaluation, that you licensed from NuDesign. This manual may not be copied.