



YumaPro Installation Guide

YANG-Based Unified Modular Automation Tools

YumaPro SDK Package Installation

Version 19.10-12

Table of Contents

1	Preface.....	4
1.1	Legal Statements.....	4
1.2	Additional Resources.....	4
1.2.1	WEB Sites.....	4
1.2.2	Mailing Lists.....	5
1.3	Conventions Used in this Document.....	5
2	Introduction.....	6
3	Installing YumaPro SDK.....	7
3.1	Pre-requisites.....	7
3.1.1	External Package Needed by the Server and Client.....	7
3.1.2	External Package Needed by the Server.....	8
3.1.3	External Packages Needed by the Client.....	8
3.2	YumaPro Packages.....	9
3.2.1	Installing YumaPro SDK Binary Packages.....	10
3.2.2	Installing YumaPro SDK from Source Code.....	12
3.2.3	Uninstalling YumaPro SDK.....	12
3.3	Configure SSH.....	13
3.3.1	Installing SSH keys.....	13
3.4	SELinux Security Configuration.....	14
3.5	Restart the SSH Server.....	14
3.6	Start netconfd-pro.....	16
3.6.1	Server Startup Issues.....	16
3.7	yangcli-pro Connect.....	17
3.8	Configure TLS.....	19
3.8.1	Server Setup.....	19
3.8.2	Client Setup.....	22
3.8.3	Test the TLS Connection.....	23
3.9	Starting netconfd-pro with yppwatcher Program.....	24
4	RESTCONF Installation.....	25
4.1	Pre-requisites.....	25
4.1.1	NGINX Support.....	25
4.1.2	External Packages Needed by the Server.....	26
4.2	Configuring the Apache Server File.....	27
4.2.1	Setup TLS Certificates for Your RESTCONF Site.....	29
4.3	Restart the Apache Server.....	30
4.4	HTTP Connect.....	31
5	yp-shell – adding a CLI.....	32
5.1	Pre-requisites.....	32
5.2	Installing yp-shell.....	32
5.2.1	Add yp-shell to the /etc/shells File.....	32
5.2.2	Change the Shell for the Login User to yp-shell.....	32
A	Appendix: Installed Files.....	35
A.1	Release Notes.....	35
A.2	YANG Modules.....	35
A.3	Shared Libraries.....	35
A.4	Binary Programs.....	36
A.5	User Manual Pages.....	37
A.6	User Documentation.....	37

YumaPro SDK Installation Guide

A.7 Sample Configuration Files.....	38
A.8 Developer Files.....	38
A.9 License and Packaging Information.....	41
B Appendix: Next Steps.....	42
B.1 Solutions.....	42
B.2 More Documentation.....	42
B.3 Unix 'man' pages.....	43
B.4 Program -help option.....	43

1 Preface

1.1 Legal Statements

Copyright 2009 – 2012, Andy Bierman, All Rights Reserved.

Copyright 2012 - 2020, YumaWorks, Inc., All Rights Reserved.

1.2 Additional Resources

Other documentation includes:

- YumaPro Quickstart Guide
- YumaPro User Manual
- YumaPro netconfd-pro Manual
- YumaPro yangcli-pro Manual
- YumaPro ypclient-pro Manual
- YumaPro yangdiff-pro Manual
- YumaPro yangdump-pro Manual
- YumaPro Developer Manual
- YumaPro API Quickstart Guide
- YumaPro ypgnmi Guide
- YumaPro yp-system API Guide
- YumaPro yp-show API Guide
- YumaPro Yocto Linux Quickstart Guide
- YumaPro yp-snmp Manual

To obtain additional support you may contact YumaWorks technical support department:

support@yumaworks.com

1.2.1 WEB Sites

- **YumaWorks**
 - <https://www.yumaworks.com>
 - Offers support, training, and consulting for YumaPro.
- **Netconf Central**
 - <http://www.netconfcentral.org/>
 - Free information on NETCONF and YANG, tutorials, on-line YANG module validation and documentation database
- **Yang Central**
 - <http://www.yang-central.org>
 - Free information and tutorials on YANG, free YANG tools for download

YumaPro SDK Installation Guide

- **NETCONF Working Group Wiki Page**
 - <http://trac.tools.ietf.org/wg/netconf/trac/wiki>
 - Free information on NETCONF standardization activities and NETCONF implementations
- **NETCONF WG Status Page**
 - <http://tools.ietf.org/wg/netconf/>
 - IETF Internet draft status for NETCONF documents
- **libsmi Home Page**
 - <http://www.ibr.cs.tu-bs.de/projects/libsmi/>
 - Free tools such as smidump, to convert SMIV2 to YANG



1.2.2 Mailing Lists

- **NETCONF Working Group**
 - <https://mailarchive.ietf.org/arch/browse/netconf/>
 - Technical issues related to the NETCONF protocol are discussed on the NETCONF WG mailing list. Refer to the instructions on <https://www.ietf.org/mailman/listinfo/netconf> for joining the mailing list.
- **NETMOD Working Group**
 - <https://datatracker.ietf.org/wg/netmod/documents/>
 - Technical issues related to the YANG language and YANG data types are discussed on the NETMOD WG mailing list. Refer to the instructions on the WEB page for joining the mailing list.

1.3 Conventions Used in this Document

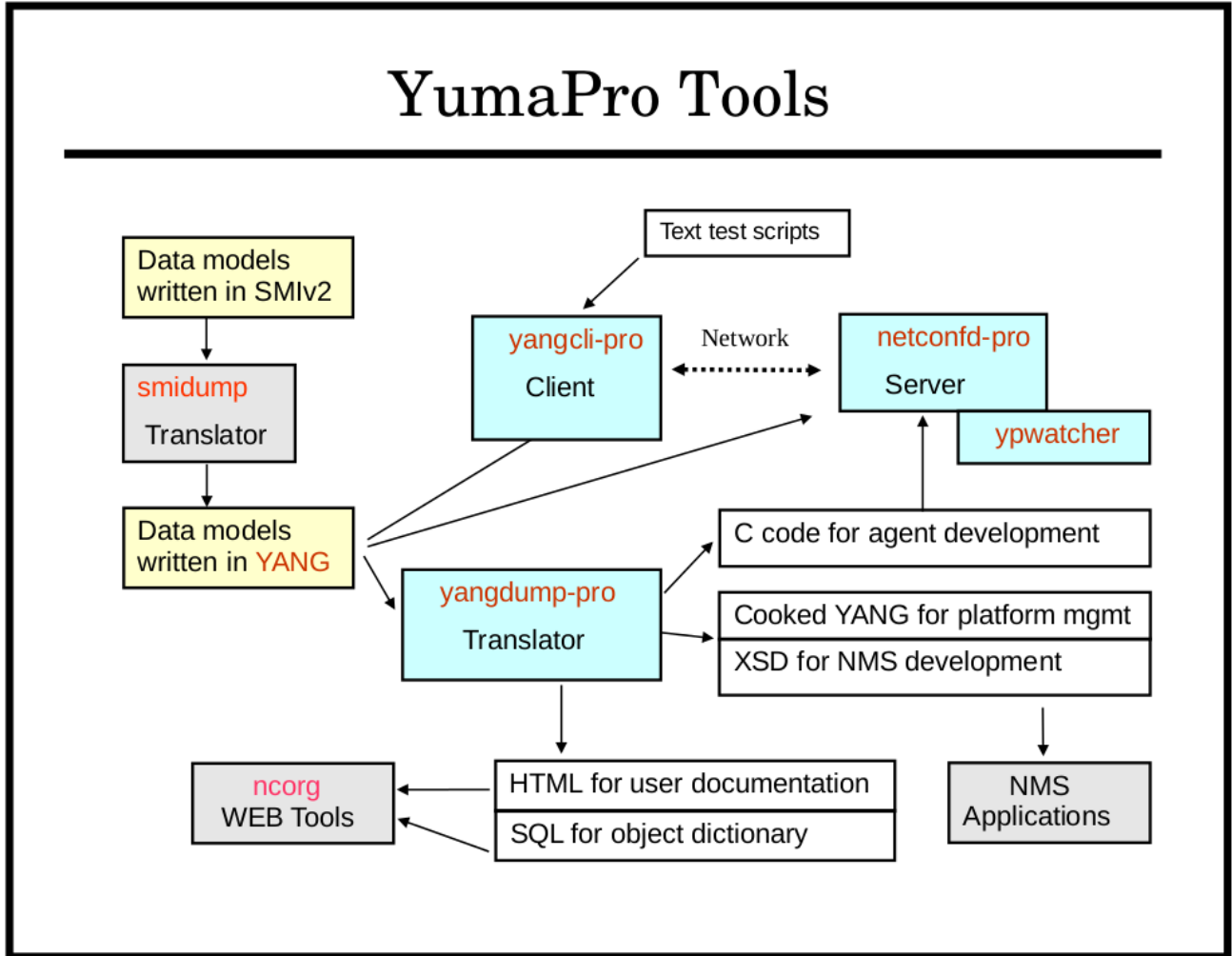
The following formatting conventions are used throughout this document:

Documentation Conventions

Convention	Description
<code>--foo</code>	CLI parameter foo
<code><foo></code>	XML element foo
<code>foo</code>	netconfd-pro command or parameter
<code>\$FOO</code>	Environment variable FOO
some text	Example command or PDU
some text	Plain text
 Informational text	Useful or expanded information
 Warning text	Warning information indicating possibly unexpected side-effects

2 Introduction

YumaPro SDK provides a complete YANG Software Development Kit to build servers and clients that support multi-protocol interfaces to device or virtual service configuration and operational data.



The following lessons provide an introduction to YumaPro SDK including:

- basics of installation
- network connection
- installing optional features
- working with YANG modules
- creating and testing server instrumentation

3 Installing YumaPro SDK

To start using YumaPro SDK it first needs to be installed along with some support libraries on a Linux system. The following platforms are supported and maintained for the YumaPro SDK binary packages:

- Ubuntu 16.04 LTS, 18.04 LTS* and 20.04 LTS* (64-bit AMD, * indicates AARCH64 64-bit ARM support)
- Debian 8, 9 and 10 (64-bit AMD)
- Fedora 30, 31 and 32* (64-bit x86_64, * indicates AARCH64 64-bit ARM support)
- CentOS 6, 7 and 8 (64-bit x86_64)
- Red Hat Enterprise Linux 7, 8 (64-bit x86_64)
- OpenSUSE Leap 42 (64-bit x86_64)
- Raspian for Raspberry Pi 2 (ARMv6)
- Raspian for Raspberry Pi 3 (ARMv7)

Other platforms are available upon request.

3.1 Pre-requisites

YumaPro SDK installs on most modern Linux systems with very few requirements. You will need:

- \$HOME needs to be set on your system or you need to run netconfd-pro with the option `-home=/home/<YOUR-USER-NAME>`. To set your \$HOME environment variable see: <https://help.ubuntu.com/community/EnvironmentVariables>
- a current version of one of the supported Linux systems
- access to the External Open Source packages listed below
- root access to your Linux system to install packages
- access to YumaWorks' dld site: <https://www.yumaworks.com/dld>

You should have received a username and password for access to YumaWorks' download site that provides access to the YumaWorks' products you have licensed. If you have not received access please contact sales@yumaworks.com.

3.1.1 External Package Needed by the Server and Client

libxml2: is needed by some of the XML parsing functions This is usually installed by default on most Linux systems. If you are building YumaPro from source you will also need the associated developer package called **libxml2-dev** on DEB based systems and **libxml2-devel** on RPM based systems.

```
Ubuntu version:  
mydir> sudo apt-get install libxml2-dev
```

```
Fedora version:  
mydir> sudo dnf install libxml2-devel
```

3.1.2 External Package Needed by the Server

openssh-server: is needed by the netconfd-pro server for NETCONF over SSH support. This package may not be installed on some Linux systems.

```
Ubuntu version:  
mydir> sudo apt-get install openssh-server
```

```
Fedora version:  
mydir> sudo dnf install openssh-server
```



NOTE: The `/etc/ssh/sshd_config` file must be properly configured and the SSH server restarted before the netconfd-pro server will work. The configuration is covered in a section below.

libcurl: is needed by the netconfd-pro server to support `<copy-config>` to/from an FTP or TFTP URL. This package may not be installed on some Linux systems. If you are building the software from source code, this library is only required if the `WITH_CURL=1` or `EVERYTHING=1` make parameter is used. There are several Ubuntu variants of the libcurl4 package. The “gnutls” variant is shown below:

```
Ubuntu version:  
mydir> sudo apt-get install libcurl4-gnutls-dev
```

```
Fedora version:  
mydir> sudo dnf install libcurl-devel
```

3.1.3 External Packages Needed by the Client

The following packages are needed for the yangcli-pro client to function. If you are building YumaPro from source you will also need the associated developer packages.

libssh2: is needed in order to connect to NETCONF servers using the SSH protocol. The developer version of this package. It is called libssh2-1-dev on DEB based systems and libssh2-devel on RPM based systems.



NOTE: CentOS 5 users may need to use the RPMForge repository to download libssh2 and libssh2-devel. For help with acquiring RPMForge please refer to:

YumaPro SDK Installation Guide

<http://wiki.centos.org/AdditionalResources/Repositories/RPMForge>

Users of Red Hat and CentOS version 8 and above may find libssh2 is not provided. Please refer to:

[I can't find libssh2 in Red Hat and CentOS](#)

```
Ubuntu version:  
mydir> sudo apt-get install libssh2-1-dev
```

```
Fedora version:  
mydir> sudo dnf install libssh2-devel
```

ncurses: is needed for some terminal support. This package is installed by the default Linux installation process. The developer version of this package is called libncurses5-dev on DEB based systems and ncurses-devel on RPM based systems.

```
Ubuntu version:  
mydir> sudo apt-get install libncurses5-dev
```

```
Fedora version:  
mydir> sudo dnf install ncurses-devel
```

zlib1g: is needed for data compression support, used by other libraries that YumaPro imports. This package is installed by the default Linux installation process. The developer version of this package. It is called zlib1g-dev on DEB based systems.

```
Ubuntu version:  
mydir> sudo apt-get install zlib1g-dev
```

```
Fedora version:  
mydir> sudo dnf install zlib-devel
```

3.2 YumaPro Packages

YumaPro SDK Installation Guide

There are three main variants of YumaPro SDK. The installation is similar for all packages. The <version#> below is the version and release number of the package you choose.

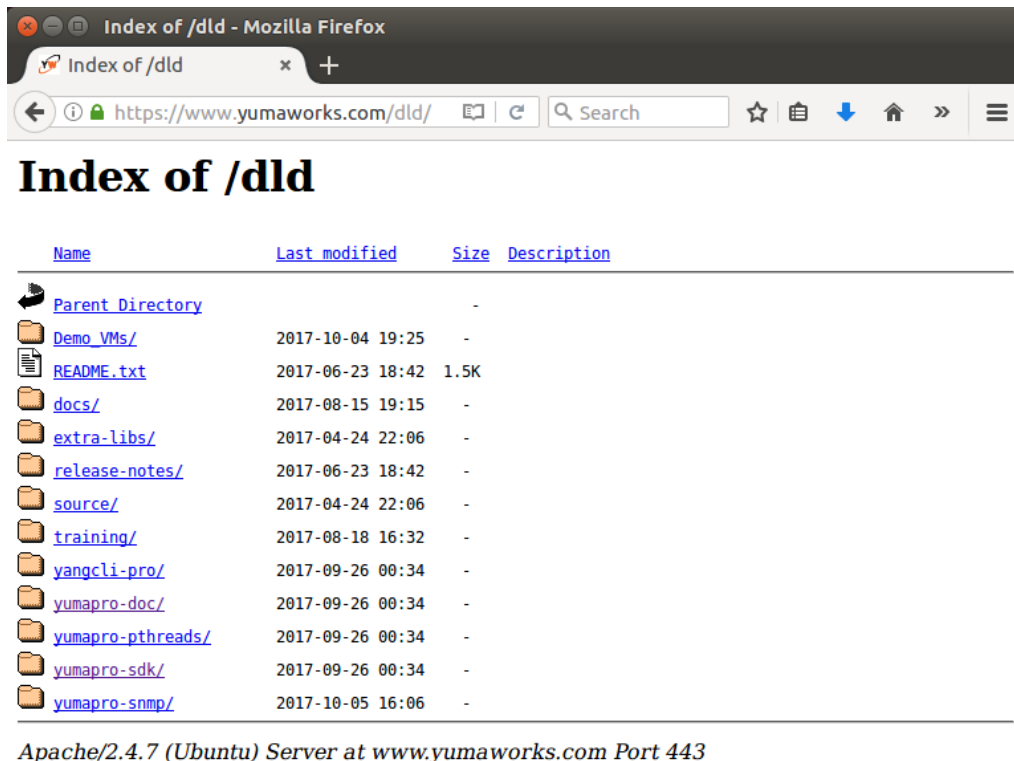


Initially you should just use the latest version of the SDK. For more information on choosing a package see: [Which YumaPro Release Train Should I Use?](#)

- yumapro-sdk-<version#>
 - This package contains the full SDK without POSIX pthreads supported
 - yumapro-pthreads-<version#>
 - This package contains the full SDK with POSIX pthreads supported
 - yumapro-snmp-<version#>
 - This package contains the full SDK with the Simple Network Management Protocol (SNMP) supported
- There is also a package to install the YumaPro SDK documentation.
- yumapro-doc-<version#>
 - This package contains the YumaPro SDK documentation (except Linux “man” files).

3.2.1 Installing YumaPro SDK Binary Packages

- in your browser navigate to <https://www.yumaworks.com/dld> and login with the credentials YumaWorks provided to you. You should see something similar to this:



YumaPro SDK Installation Guide

- navigate to the yumapro- version you want to install. In a later lesson we will go through YumaPro's SNMP support so for now choose yumapro-sdk or yumapro-threads
- you will see a list of the releases you can download. Unless you have a specific need for a particular version of YumaPro SDK it is best to navigate to the bottom of the page and select the "latest" folder. This contains the latest main release
- In the "latest" folder you will see listed the supported platforms each in its own folder. Select the one appropriate for the platform you are going to install YumaPro SDK and then select the binary needed, usually either 64-bit or 32-bit, that is appropriate for your system
- download the package and then in a terminal window navigate to where the package was saved and execute the following command (the package and version number should reflect the package you are installing):

You should see something like:

```
Ubuntu version:
```

```
mydir> ls -al
total 12332
drwxr-xr-x  2 john john    4096 Oct 14 13:36 .
drwxr-xr-x 19 john john    4096 Oct 14 13:36 ..
-rw-rw-r--  1 john john 2391468 Sep 28 18:32 yumapro-sdk-17.10-
1.u1604.amd64.deb

mydir> sudo dpkg -i yumapro-sdk-17.10-1.u1604.amd64.deb
```

```
Fedora version:
```

```
mydir> ls -al
total 2780
drwxr-xr-x.  2 john john    4096 Oct 16 02:45 ./
drwx----- 15 john john    4096 Oct 15 21:47 ../
-rw-rw-r--.  1 john john 2837870 Oct 16 02:44 yumapro-sdk-17.10-
1.fc26.x86_64.rpm

mydir> sudo dnf install yumapro-sdk-17.10-1.fc26.x86_64.rpm
```

To quickly test the install run the server to just display its version:

```
mydir> netconfd-pro --version
```

```
Starting netconfd-pro...
Copyright (c) 2008-2012, Andy Bierman, All Rights Reserved.
Copyright (c) 2012-2017, YumaWorks, Inc., All Rights Reserved.

netconfd-pro version 17.10-1
```

YumaPro SDK Installation Guide



To see which files are installed in more detail look at Appendix A “Installed Files” at the end of this document.

If you need to install the documentation go to <https://yumaworks.com/dld/yumapro-doc/latest/> , download the documentation that matches the version and release number of the SDK package you installed.

```
Ubuntu version:  
mydir> sudo dpkg -i yumapro-doc_17.10-1_all.deb
```

```
Fedora version:  
mydir> sudo dnf install yumapro-doc-17.10-1.noarch.rpm
```

3.2.2 Installing YumaPro SDK from Source Code



To install YumaPro SDK from source files make sure you have installed the developer versions of the External Packages listed previously. Navigate to the YumaPro top level directory. You will find the file “README” that describes how to build YumaPro SDK and the build variables available. Below is an example. It is best to build, install, and if needed uninstall using the same build variables.

```
mydir> make DEBUG=1 DEBUG2=1 EVERYTHING=1 USE_WERROR=1  
mydir> sudo make install DEBUG=1 DEBUG2=1 EVERYTHING=1 USE_WERROR=1
```



NOTE: If you have Server source code (yumapro-server2-*) you will not have access to the yangcli-pro/yp-shell source code and as such EVERYTHING=1 will not work. You should:

- substitute EVERYTHING=1 for the features you want to include such as RESTCONF=1. Consult the README file in the source code for options or search

<https://yumaworks.freshdesk.com/a/solutions/> for “server build options”

- to use yangcli-pro and/or yp-shell use a version from one of the available binary packages and install:

- libyumapro_mgr.so
- libyumapro_ycli.so
- yangcli-pro
- yp-shell

3.2.3 Uninstalling YumaPro SDK



If you need to uninstall YumaPro SDK see the article: [How do I remove YumaPro SDK from my system?](#)

3.3 Configure SSH

To use the server you must modify the `/etc/ssh/sshd_config` file and add the netconf subsystem to the file. From a terminal edit the file:

```
mydir> sudo <your_editor> /etc/ssh/sshd_config
```



Replace `<your_editor>` with the editor of your choice such as `vi`, `vim`, `emacs`, `gedit`, etc.

Add the following commands to `sshd_config` (Port 22 will probably already exist in the file though it may be commented out)

```
Port 22
Port 830
Subsystem netconf /usr/sbin/netconf-subsystem-pro
```

3.3.1 Installing SSH keys

YumaPro SDK provides support for SSH keys. If you do not have SSH keys installed already the easiest way to install these is as follows:

```
mydir> ssh-keygen

Generating public/private rsa key pair.
Enter file in which to save the key (/<your_$HOME>/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /<your_$HOME>/.ssh/id_rsa.
Your public key has been saved in /<your_$HOME>/.ssh/id_rsa.pub.
The key fingerprint is:
...
```



NOTE: You will be asked if you want to store the keys somewhere other than the standard location.

You will also be asked if you want to enter a passphrase. If you enter a passphrase you will need to provide that passphrase each time you connect to the server.

This process creates a public and a private key. The public key needs to be placed on the server. The easiest way to do this is with `ssh-copy-id`. The output should look something like this:

YumaPro SDK Installation Guide

```
mydir> ssh-copy-id <your_username>@<server's_IP_address>

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to
filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '<username>@<server>'"
and check to make sure that only the key(s) you wanted were added.
```

If you have not connected to the server before you may be asked to confirm it is OK to connect to the server.

You should validate that everything worked by using ssh to connect to the server as the output from ssh-copy-id suggests.

If you do not have ssh-copy-id on your system there are other ways to put the public key on the server. Consult with your sys-admin the preferred way to do this.

3.4 SELinux Security Configuration

For SELinux systems like CentOS and Fedora you must configure SELinux to allow the port and subsystem changes. Edit the SELinux configuration file, show below, and change SELINUX from “enforcing” to “permissive”. Then reboot the system.

```
Fedora version:
mydir> sudo <your_editor> /etc/selinux/config

SELINUX=permissive

mydir> sudo reboot
```



NOTE: Check with your system administrator for the network security policies that are required for the server. A “permissive” SELinux level could be too lax for your network’s policy.

3.5 Restart the SSH Server

Restart the SSH Server with:

```
Ubuntu version:
mydir> sudo service ssh restart
```

```
Fedora version:
mydir> sudo service sshd restart
```

YumaPro SDK Installation Guide



NOTE: SELinux systems usually do not start sshd on reboot so you will need to restart it each time you reboot your system or have it started with your system initialization.

3.6 Start netconfd-pro

Next start the netconfd-pro server:

```
mydir> netconfd-pro --log-level=debug4 --access-control=off
```

There are many parameters you can use to tune the server to perform the way you desire. The two parameters shown above start the server with the most verbose debug message level to let you monitor what the server is doing and disabling the access-control so you can manipulate the YANG data without having to setup NETCONF Access Control.



For more information of log-level see: [What is the --log-level parameter and how is it used?](#)

For more information on the netconfd-pro server command line parameters see Chapter 3 “CLI Reference” of the yumapro-netconfd-manual.pdf or the HTML version of the manual located at: [CLI Reference](#)

3.6.1 Server Startup Issues



If you have previously run the netconfd-pro server and you see the message below the server is either still running in another process or was not shut down cleanly the last time it ran. Either stop the other server running or follow the instructions in the message to clean up the files left by the previous run.

```
Error: program netconfd-pro appears to be running as PID 5125
Error: Cannot create PID file
*** If no other instances of netconfd-pro are running,
*** try deleting /tmp/ncxserver.sock and $HOME/.yumapro/netconfd-pro.pid
*** > rm /tmp/ncxserver.sock
*** > rm $HOME/.yumapro/netconfd-pro.pid
```

```
netconfd-pro: init returned (operation failed)
Server Cleanup Starting...
```

```
mydir> sudo rm /tmp/ncxserver.sock
rm: cannot remove '/tmp/ncxserver.sock': No such file or directory
mydir> sudo rm $HOME/.yumapro/netconfd-pro.pid
```


3.7 yangcli-pro Connect

In a separate terminal window start the yangcli-pro client:

```
mydir> yangcli-pro
```

The login message from yangcli-pro will be displayed followed by the command prompt “>”. Part of the login message displays some of the help and command completion options available:

```
Type 'help' or 'help <command-name>' to get started
Use the <tab> key for command and value completion
Use the <enter> key to accept the default value in brackets

These escape sequences are available when filling parameter values:

?      help
??     full help
?s     skip current parameter
?se    skip rest of optional parameters
?c     cancel current command
```

Connect to the server:



NOTE: for <your-username> and <your-password> use your system login user name and password.

```
> connect server=localhost user=<your-username> password=<your-password>
```

The server’s “hello” message will be displayed detailing the servers capabilities and other information. In the terminal window where the server is running you will see its debug information displayed at the debug level the server was started with.



If you experience problems connecting to the server see the article: [Cannot Connect to the Server](#)

YumaPro SDK Installation Guide

You can now issue some commands to display YANG data, such as:

```
<your-username>@localhost> sget /netconf-state/sessions

Filling container /netconf-state/sessions:
RPC Data Reply 5 for session 3 [default]:

rpc-reply {
  data {
    netconf-state {
      sessions {
        session 3 {
          session-id 3
          transport ncm:netconf-ssh
          username <your-username>
          source-host 127.0.0.1
          login-time 2018-11-20T07:00:17Z
          in-rpcs 4
          in-bad-rpcs 0
          out-rpc-errors 0
          out-notifications 0
        }
      }
    }
  }
}
```

With each command issued from yangcli-pro you will be able to see the corresponding debug information displayed from the server in the terminal window in which the server is running.



For more information on yangcli-pro and the commands available see yumapro-yangcli-manual.pdf or the HTML version of the manual located at: [yumapro yangcli manual](#)

To exit yangcli-pro type “quit”:

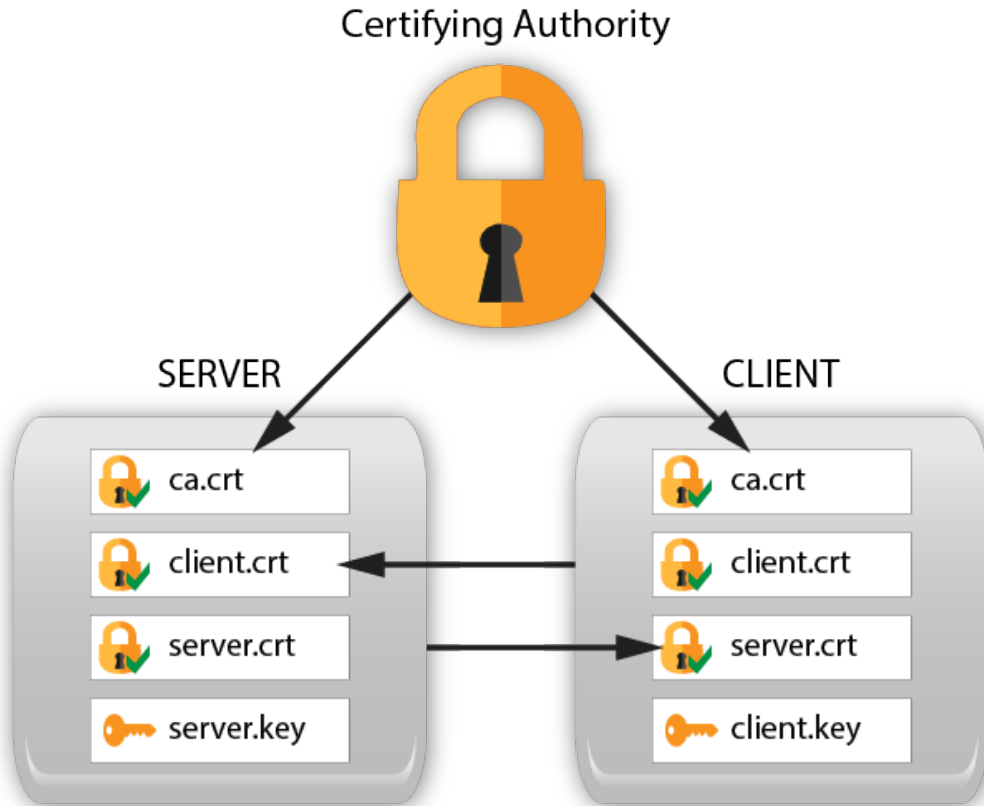
```
<your-username>@localhost> quit

mydir>
```

To exit the server type <Ctrl>-c in the window it is running in.

3.8 Configure TLS

To enable Transport Layer Security (TLS) between the server and client you need to setup both the server and the client with appropriate certificates and configuration. The following instructions walk you through the setup process.



Graphical representation of the server, client, and authority certificates and keys for TLS.

3.8.1 Server Setup

On the server create a couple of directories for working with the certificates and copy the generate-keys.sh script to the buildcerts directory that was just created:

```
mydir> mkdir $HOME/buildcerts
mydir> mkdir $HOME/certs
mydir> cp /usr/share/yumapro/util/generate-keys.sh $HOME/buildcerts
```

YumaPro SDK Installation Guide

cd to the buildcerts directory, run the key generation script and check the files were created:

```
mydir> cd buildcerts
buildcerts> ./generate-keys.sh

buildcerts> ls -l

-rw-rw-r-- 1 andy andy 956 Mar 16 15:05 ca.crt
-rw-rw-r-- 1 andy andy 883 Mar 16 15:05 ca.csr
-rw-rw-r-- 1 andy andy 1708 Mar 16 15:05 ca.key
-rw-rw-r-- 1 andy andy 17 Mar 16 15:05 ca.srl
-rw-rw-r-- 1 andy andy 969 Mar 16 15:05 client.crt
-rw-rw-r-- 1 andy andy 891 Mar 16 15:05 client.csr
-rw-rw-r-- 1 andy andy 1708 Mar 16 15:05 client.key
-rwxrwxr-x 1 andy andy 1513 Feb 23 16:29 generate-keys.sh
-rw-rw-r-- 1 andy andy 969 Mar 16 15:05 server.crt
-rw-rw-r-- 1 andy andy 891 Mar 16 15:05 server.csr
-rw-rw-r-- 1 andy andy 1704 Mar 16 15:05 server.key
```



The generate-keys.sh script will generate keys and certs for the “restconf” site. You can use the keys created by this script to setup TLS for your restconf site as described in Section 4 of this document.

Copy the certificates to their proper places:

```
buildcerts> sudo cp ca.crt /usr/local/share/ca-certificates/
buildcerts> cp server.crt ../certs/
buildcerts> cp server.key ../certs/
```

Go to the /etc/ssl/certs directory, run updates and check the results:

```
buildcerts> cd /etc/ssl/certs
certs> sudo update-ca-certificates

certs> ls -l | grep ca.crt
lrwxrwxrwx 1 root root 39 Mar 16 15:52 ca.pem -> /usr/local/share/ca-
certificates/ca.crt
```



The script update-ca-certificates uses the ca-certificates package. If you do not have this package on your system, for example if you are building a minimal footprint system, then the following steps, instead of the section above, will create the required certificate links:

```
buildcerts> mkdir temp
buildcerts> sudo ln -s /usr/local/share/ca-certificates/ca.crt temp/ca.pem
```

YumaPro SDK Installation Guide


```
buildcerts> sudo c_rehash temp
Doing temp
buildcerts> sudo mv temp/* /etc/ssl/certs
buildcerts> ls -l /etc/ssl/certs | grep ca.pem
lrwxrwxrwx 1 root root      6 Mar 16 08:00 56c899cd.0 -> ca.pem
lrwxrwxrwx 1 root root      6 Mar 16 08:00 b2457b50.0 -> ca.pem
lrwxrwxrwx 1 root root     39 Mar 16 08:00 ca.pem -> /usr/local/share/ca-certificates/ca.crt
buildcerts>
```

Generate the client Fingerprint:

```
certs> cd $HOME/buildcerts
buildcerts> openssl x509 -noout -fingerprint -sha1 \
    -inform pem -in client.crt

SHA1
Fingerprint=4B:A7:05:1E:12:F7:BC:FF:2D:9E:48:66:0A:8B:CC:D7:A5:65:E5:97
```


Next you need to configure the server with the parameters needed to use TLS by editing the netconfd-pro.conf file. If you have an existing netconfd-pro.conf then add the four parameter lines, within “netconfd-pro {“ and “}”, to the existing file. If you don't have an existing netconfd-pro.conf file then run your editor as shown and a netconfd-pro.conf file will be created, then add the lines shown.

 NOTE: the cert-usermap parameter required for netconfd-pro.conf will be <YOUR_USERNAME>@<first_six_pairs_of_the_SHA1_Fingerprint> from the “Generate the client Fingerprint:” step above.

For example if user=andy creates the SHA1 Fingerprint then the cert-usermap parameter line will be:

```
cert-usermap andy@4B:A7:05:1E:12:F7
```

```
buildcerts> sudo <your_editor> /etc/yumapro/netconfd-pro.conf
```

 Replace <your_editor> with the editor of your choice such as vi, vim, emacs, gedit, etc.

Add following four parameters lines, substituting cert-usermap with your version – see above:

```
netconfd-pro {
  with-netconf-tls true
  netconf-tls-certificate ~/certs/server.crt
  netconf-tls-key ~/certs/server.key
  cert-usermap <YOUR_USERNAME>@<first_six_pairs_of_the_SHA1_Fingerprint>
}
```

3.8.2 Client Setup

On the client create a couple of directories for working with the certificates:

```
CLIENT:  
  
mydir> mkdir $HOME/buildcerts  
mydir> mkdir $HOME/certs
```

On the server copy the files you created to the client machine using sftp:

```
SERVER:  
  
mydir> cd $HOME/buildcerts  
buildcerts> sftp CLIENT_USERNAME@CLIENT  
sftp> cd buildcerts  
sftp> put *  
sftp> bye
```

Now copy the certificates on the client to their proper places:

```
CLIENT:  
  
mydir> cd $HOME/buildcerts  
buildcerts> sudo cp ca.crt /usr/local/share/ca-certificates  
buildcerts> cp client.crt $HOME/certs/  
buildcerts> cp client.key $HOME/certs/
```

Go to the /etc/ssl/certs directory, run updates and check the results:

```
CLIENT:  
  
buildcerts> cd /etc/ssl/certs  
certs> sudo update-ca-certificates  
  
certs> ls -l | grep ca.crt  
  
lrwxrwxrwx 1 root root    39 Mar 16 16:25 ca.pem -> /usr/local/share/ca-  
certificates/ca.crt
```

YumaPro SDK Installation Guide



Similarly, if you do not have the ca-certificates package see the note in the Server Setup section above on using `c_rehash`.

Next you need to configure the client with the parameters needed to use TLS by editing the `yangcli-pro.conf` file. If you have an existing `yangcli-pro.conf` then add the two parameter lines, within “`yangcli-pro {`” and “`}`”, to the existing file. If you don't have an existing `yangcli-pro.conf` file then run your editor as shown and a `yangcli-pro.conf` file will be created, then add the lines shown.

```
buildcerts> sudo <your_editor> /etc/yumapro/yangcli-pro.conf
```



Replace `<your_editor>` with the editor of your choice such as `vi`, `vim`, `emacs`, `gedit`, etc.

Add the following two parameters:

```
yangcli-pro {  
  ssl-certificate ~/certs/client.crt  
  ssl-key ~/certs/client.key  
}
```

3.8.3 Test the TLS Connection

To test the TLS connection run the server as you would normally, for example:

```
SERVER:  
mydir> netconfd-pro log-level=debug4 access-control=off
```

Run `yangcli-pro` on the client and connect using the command show below:

```
CLIENT:  
mydir> yangcli-pro  
...  
> connect user=<andy> server=<SERVER_HOST> no-password transport=tls
```



NOTE: replace `<andy>` with your user name and `<SERVER_HOST>` with the name or address of the server.

3.9 Starting netconfd-pro with ywatcher Program

The **ywatcher** is a program that provides monitoring mechanism to **netconfd-pro** server and its state. **Ypwatcher** program periodically checks the server's state and determine if the server is still running. If the server is no longer running it cleans up the state, restarts the server, and generates a syslog message.

The **ywatcher** program will be launched by the server by default unless **--no-watcher** parameter will be specified or the program is already running.

The **ywatcher** program is running continuously and attempting to restart the server any time it exits unexpectedly.

The **ywatcher** program will be invoked automatically whether the server starts interactively or in the background mode:

- To start the server interactively, with **ywatcher** program:

```
mydir> netconfd-pro
```

- To start the server interactively, with no **ywatcher** program:

```
mydir> netconfd-pro --no-watcher
```

The **--watcher-interval** parameter specifies the sleep interval between **ywatcher** program attempts to check availability of the server.

- To start the server interactively, with **ywatcher** program and set the watcher interval:

```
mydir> netconfd-pro --watcher-interval=10
```


4 RESTCONF Installation

Configure the optional RESTCONF protocol.



NOTE: RESTCONF is not available with YumaPro SDK Basic.

The restconf program is the FastCGI thin client that connects Apache2 (or other WEB server) to the netconf-subsystem-pro program. This lesson describes how to setup the restconf program as a WEB site on your system.



NOTE: if you are using an SELinux system (RHEL, CentOS, Fedora) you will need to set SELinux to permissive mode and start netconfd-pro as root with the `-fileloc-fhs` parameter set to true if you wish to use restconf:

```
mydir> sudo netconfd-pro -fileloc-fhs=true
```

Please consult your System Administrator for assistance in managing SELinux on your system.



NOTE: if you are using Ubuntu 18.04, then the “restconf” FGCI program called from the WEB server will not be permitted to access local sockets in the /tmp directory such as /tmp/ncxserver.sock. The fileloc-fhs parameter must be set to true for RESTCONF to function in Ubuntu 18.04.

```
mydir> sudo netconfd-pro -fileloc-fhs=true
```

Please consult your System Administrator for assistance in managing SELinux on your system.

4.1 Pre-requisites

You should have completed “3 Installing YumaPro SDK”. If you have installed YumaPro SDK with a binary package then RESTCONF is included. If you have installed the SDK from source code then you need to have built and installed using `EVERYTHING=1` or `WITH_RESTCONF=1` build variables.

4.1.1 NGINX Support

The following steps will show you how to integrate the restconf program into an **Apache** WEB server. If you would rather integrate restconf into an **NGINX** WEB server then please review our FAQ: [Setting up RESTCONF on the NGINX WEB server](#).

4.1.2 External Packages Needed by the Server

To use the RESTCONF protocol a WEB server is required. It must support the FastCGI API which is used by the restconf program for REST access to the netconfd-pro server.



NOTE: The 'fcgid' module is needed. Do not use the older 'fastcgi' module.

```
Ubuntu version:  
mydir> sudo apt-get install apache2 libapache2-mod-fcgid
```

```
Fedora version:  
mydir> sudo dnf install httpd  
mydir> sudo dnf install fcgi-devel  
mydir> sudo dnf install mod_fcgid
```



If commands shown above are not successful, install and build FastCGI developer kit from the source:
The archived WEB site for FastCGI:

<https://fastcgi-archives.github.io/>

Download latest libfcgi:

https://github.com/FastCGI-Archives/FastCGI.com/blob/master/original_snapshot/fcgi-2.4.1-SNAP-0910052249.tar.gz

Build and install from the source

If you have built and installed YumaPro SDK from source code then the restconf program will be installed in the correct location. If you installed YumaPro SDK from a binary package you will need take an additional steps, creating sudo mkdir /var/www/yang-api/ if it does not exists and moving the restconf program as show below:

```
mydir> sudo mkdir /var/www/yang-api/  
mydir> sudo mv /usr/sbin/restconf /var/www/yang-api/  
mydir> sudo chmod 775 /var/www/yang-api/restconf  
mydir> sudo chown www-data:www-data /var/www/yang-api/restconf
```

For Fedora systems you need to change the user and group ownership of restconf:

```
Fedora version:  
mydir> sudo chown apache:apache /var/www/yang-api/restconf
```

4.2 Configuring the Apache Server File



NOTE: Before making any changes to your Apache configuration, be sure to back up the configuration file:

```
Ubuntu version:
mydir> sudo cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf.backup
```

```
Fedora version:
mydir> sudo cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.backup
```

Enable modules in Apache:

```
Ubuntu version:
mydir> sudo a2enmod fcgid status headers
Module fcgid already enabled
Module status already enabled
Enabling module headers.
To activate the new configuration, you need to run:
service apache2 restart
```



On Fedora/CentOS you don't need to explicitly enable individual Apache modules like mod-fcgid, as these modules are enabled automatically upon installation.

Apache mod_status offers an option called ExtendedStatus, which provides additional information about each request made to Apache and FastCGI. To enable ExtendedStatus edit your Apache configuration file:

```
Ubuntu version:
mydir> sudo <your_editor> /etc/apache2/apache2.conf
```

and add to the end of the configuration file:

```
ExtendedStatus On
```



NOTE: Enabling ExtendedStatus consumes additional system resources.

YumaPro SDK Installation Guide



Apache mod_headers is used to provide support for the RESTCONF client discovery of the root of the RESTCONF API. The client discovers this by getting the "/.well-known/host-meta" resource and using the <Link> element containing the "restconf" attribute. Refer to Virtual Host configuration file below for more details.

Restart Apache:

```
Ubuntu version:  
mydir> sudo service apache2 restart
```

You need a restconf configuration file. One has been provided in /usr/share/yumapro/util. To place the configuration file in the correct location for Apache:

```
Ubuntu version:  
mydir> sudo cp /usr/share/yumapro/util/restconf.conf \  
/etc/apache2/sites-available/
```

```
Fedora version:  
mydir> sudo cp /usr/share/yumapro/util/restconf.conf \  
/etc/httpd/conf.d/
```

Fedora version: In the /etc/httpd/conf.d/restconf.conf file you need to comment out the sections for logging.

```
mydir> sudo <your_editor> /etc/httpd/conf.d/restconf.conf  
  
...  
  
#### CHANGE to preferred logging location if desired  
#### MUST Change if not Apache2!!!  
#ErrorLog ${APACHE_LOG_DIR}/error.log  
  
# Possible values include: debug, info, notice, warn, error, crit,  
# alert, emerg.  
#LogLevel warn  
  
#### CHANGE to preferred logging location if desired  
#### MUST Change if not Apache2!!!  
#CustomLog ${APACHE_LOG_DIR}/access.log combined
```

YumaPro SDK Installation Guide

Enable the restconf site:

```
Ubuntu version:
mydir> sudo a2ensite restconf.conf
Enabling site restconf.
To activate the new configuration, you need to run:
    service apache2 reload
```

Set up Apache authentication for a user “admin”:



NOTE: Using password authentication in the way shown below is considered DEPRECATED. This is a relic of yang-api, a pre-standard implementation of the RESTCONF protocol. To set up security for your restconf site please configure your own TLS certificates and keys to suit your installation. Refer to Section 3.8 “Configure TLS” of this document.

```
mydir> sudo mkdir /var/www/passwords
mydir> cd /var/www/passwords
passwords> sudo htpasswd -c passwd admin
New password:
Re-type new password:
Adding password for user admin
```

4.2.1 Setup TLS Certificates for Your RESTCONF Site

To use TLS Certificates with your restconf site you will first need to be sure your WEB server has the mod_ssl module installed and enabled. The SSL module will be installed by default if you have followed this guide to this point. You will have to enable the SSL module explicitly on Ubuntu/Debian systems:

```
Ubuntu version:
mydir> sudo a2enmod ssl
mydir> sudo service apache2 reload
```

Next you will need to edit your etc/apache2/sites-available/restconf.conf site configuration file and uncomment the optional TLS configuration block shown below:



NOTE: Be sure to insert your path to server.crt and server.key for the SSLCertificateFile and SSLCertificateKeyFile parameters instead of /home/<user-name>/certs/...

YumaPro SDK Installation Guide

```
#
# Uncomment this block to enable RESTCONF over TLS
#
#
##### CHANGE /home/user/certs/... to your username or the location of the
##### certificates, e.g./home/fred/certs/... (in two locations below)
#
#####
#####
#<IfModule mod_ssl.c>

    ...

#     SSLCertificateFile /home/<user-name>/certs/server.crt
#     SSLCertificateKeyFile /home/<user-name>/certs/server.key
#

    ...

#</IfModule>
#
#####
```

4.3 Restart the Apache Server

To restart the Apache Server follows the steps:

```
Ubuntu version:
mydir> sudo service apache2 reload
mydir> sudo service apache2 restart
```

```
Fedora version:

You need to reboot the system:
mydir> sudo reboot

When the system is back up:

mydir> sudo service sshd start
mydir> sudo service httpd start
```

Start the **netconfd-pro** as root using the `fileloc-fhs=true` parameter:

```
mydir> sudo netconfd-pro --fileloc-fhs=true
```

4.4 HTTP Connect

After WEB server reboot you can verify that the configuration are correct by using “curl” tool, for example. Send the following request to the RESTCONF server to verify that it is running and configured properly:

```
mydir> curl -u admin:<your-password> \
          http://localhost/restconf/data/netconf-state/sessions
{
  "sessions": {
    "session": [
      {
        "session-id":5,
        "transport":"yumaworks-ids:netconf-http",
        "username":"restconf",
        "source-host":"127.0.0.1",
        "login-time":"2018-11-20T07:39:14Z",
        "in-rpcs":0,
        "in-bad-rpcs":0,
        "out-rpc-errors":0,
        "out-notifications":0
      }
    ]
  }
}
```



For more web requests using curl see the article: [How can I execute web requests with tools like curl?](#)

For more information on using RESTCONF see the articles in the section: [RESTCONF](#)

5 yp-shell – adding a CLI

Configure the optional yp-shell Command Line Interface (CLI).



NOTE: yp-shell is not available with YumaPro SDK Basic.

5.1 Pre-requisites

You should have completed “3 Installing YumaPro SDK”. If you have installed YumaPro SDK with a binary package then the CLI is included. If you have installed the SDK from source code then you need to have built and installed using `EVERYTHING=1` or `WITH_CLI=1` build variables.

5.2 Installing yp-shell

5.2.1 Add yp-shell to the /etc/shells File

The file `/etc/shells` must be edited with root access. The line `/usr/bin/yp-shell` needs to be added anywhere.

```
mydir> sudo <your_editor> /etc/shells
```

The following example file shows yp-shell added at the end:

```
mydir> cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/yp-shell
```

5.2.2 Change the Shell for the Login User to yp-shell

The user(s) that will be used to login to the OpenSSH server need to be created. In this example the user 'cli' is being created (the `-m` option creates a home folder for the new user):

```
mydir> sudo useradd -m --shell /usr/bin/yp-shell cli
```


YumaPro SDK Installation Guide

Next create a password for the new user:

```
mydir> sudo passwd cli
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

If the user account already exists, then use the chsh command instead:

```
Ubuntu version:
mydir> sudo chsh --shell /usr/bin/yp-shell cli

Fedora version:
mydir> sudo usermod -s /usr/bin/yp-shell cli
```



NOTE: chsh dictates that the user can ONLY use the specified shell. Therefore **DO NOT** perform this command on a user that needs to access a bash terminal or any other type of shell other than yp-shell.

The system may need to be rebooted to activate the new shell.

Start the netconfd-pro server, the CLI must be enabled (it is by default).

A normal SSH login will invoke the yp-shell program when the user logs into the system:

```
extdir> ssh cli@192.168.1.9
cli@192.168.1.9's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

27 packages can be updated.
0 updates are security updates.

Last login: Tue Oct 10 15:38:26 2017 from 192.168.1.9

Starting NETCONF session for cli on u16-vm over netconf-tcp-ncx on port 830
cli@u16-vm>
```

You can now enter CLI commands similar to the ones in yangcli-pro. The same help and command completion options are available.

YumaPro SDK Installation Guide

```
cli@lu16-vm> sget /netconf-state/sessions

Filling container /netconf-state/sessions:
RPC Data Reply 3 for session 3 [default]:

rpc-reply {
  data {
    netconf-state {
      sessions {
        session 3 {
          session-id 3
          transport yid:netconf-cli
          username cli
          source-host 127.0.0.1
          login-time 2018-11-20T07:27:30Z
          in-rpcs 2
          in-bad-rpcs 0
          out-rpc-errors 0
          out-notifications 0
        }
      }
    }
  }
}
```

The two CLIs are very similar, the main difference is yp-shell is connected directly to the server it runs on and so does not have commands to create sessions to multiple servers.



To see the differences between yp-shell and yangcli-pro CLIs see the article: [What is the difference between yp-shell in netconfd-pro and yangcli-pro?](#)

A Appendix: Installed Files

This section describes all the files and/or directories installed by the YumaPro packages.

A.1 Release Notes

- `/usr/share/yumapro/ReleaseNotes` directory contains all the detailed release notes for the entire release train.

A.2 YANG Modules

- `/usr/share/yumapro/modules` directory contains all the YANG modules:
 - `ietf/`
 - `mib/`
 - `netconfcentral/`
 - `test/`
 - `yumaworks/`

A.3 Shared Libraries

This section describes the shared libraries needed for the server to operate.

- `/usr/lib/` directory contains the NCX and AGT libraries **NOTE: `/usr/lib64/` for Fedora and CentOS 64-bit systems**
 - `libyumapro_ncx.so.19.10`
 - `libyumapro_agt.so.19.10`
 - `libyumapro_db-api.so.19.10`
 - `libyumapro_sil-sa.so.19.10`
 - `libyumapro_ycontrol.so.19.10`
 - `libyumapro_mgr.so.19.10`
 - `libyumapro_subsys-pro.so.19.10`
 - `libyumapro_ycli.so.19.10`
- `/usr/lib/yuma` directory contains the following file: **NOTE: `/usr/lib64/yuma` for Fedora and CentOS 64-bit systems**
 - `libtoaster.so`
- `/usr/lib/yumapro` directory contains the following files: **NOTE: `/usr/lib64/yumapro` for Fedora and CentOS 64-bit systems**
 - `libget2-test.so`
 - `libhooks-test.so`
 - `libietf-interfaces.so`

- libietf-restconf-monitoring.so
- libietf-yang-library.so
- libif-linux.so
- libsil-error.so
- libtoaster.so
- libyp_system-example.so
- libyuma-arp.so
- libyuma-interfaces.so
- libyuma-mysession.so
- libyuma-proc.so
- libyumaworks-server.so

A.4 Binary Programs

This section describes the binary programs installed.

- **/usr/bin** directory contains the following programs:
 - db-api-app
 - make_sil_bundle
 - make_sil_dir_pro
 - make_sil_sa_bundle
 - make_sil_sa_dir
 - sil-sa-app
 - support-save-app
 - yangcli-pro
 - yangdiff-pro
 - yangdump-pro
 - yangdump-sdk
 - ypcontrol
 - yp-ha-app
 - yp-shell
 - ypwatcher
- **/usr/sbin** directory contains the following server programs:
 - netconfd-pro

- netconf-subsystem-pro
- restconf

A.5 User Manual Pages

- **/usr/share/man/man1** directory contains the following files:
 - db-api-app.1.gz
 - make_sil_bundle.1.gz
 - make_sil_dir_pro.1.gz
 - make_sil_sa_bundle.1.gz
 - make_sil_sa_dir.1.gz
 - netconfd-pro.1.gz
 - netconf-subsystem-pro.1.gz
 - restconf.1.gz
 - sil-sa-app.1.gz
 - support-save-app.1.gz
 - yangcli-pro.1.gz
 - yangdiff-pro.1.gz
 - yangdump-pro.1.gz
 - yangdump-sdk.1.gz
 - ypcontrol.1.gz
 - yp-ha-app.1.gz
 - yp-shell.1.gz
 - yppwatcher.1.gz

A.6 User Documentation

- **/usr/share/doc/yumapro** directory contains the following files:
 - index.html
- **/usr/share/doc/yumapro/pdf** directory containing the following files:
 - yumapro-installation-guide.pdf
 - yumapro-quickstart-guide.pdf

YumaPro SDK Installation Guide

- yumapro-user-cmn-manual.pdf
- yumapro-netconfd-manual.pdf
- yumapro-yangcli-manual.pdf
- yumapro-ypclient-manual.pdf
- yumapro-yangdiff-manual.pdf
- yumapro-yangdump-manual.pdf
- yumapro-dev-manual.pdf
- yumapro-api-quickstart-guide.pdf
- yumapro-ypsystem-api-guide.pdf
- yumapro-ypshow-api-guide.pdf
- yumapro-yocto-quickstart-guide.pdf
- yumapro-ypsnmp-manual.pdf
- **/usr/share/doc/yumapro/html** directory containing the HTML files for all the manuals
 - (directories for each HTML manual)

A.7 Sample Configuration Files

- **/etc/yumapro** directory contains the following sample configuration files:
 - yangcli-pro-sample.conf
 - yangdiff-pro-sample.conf
 - yangdump-pro-sample.conf
 - netconfd-pro-sample.conf
 - snmpd.conf

A.8 Developer Files

- **/usr/include/yumapro** directory contains H files needed to compile SIL code so it can be loaded into the server at runtime.
 - ncx/*.h
 - agt/*.h
 - db-api/*.h
 - mgr/*.h
 - platform/procdefs.h

YumaPro SDK Installation Guide

- sil-sa/*.h
- ycli/*.h
- ycontrol/*.h
- **/usr/share/yumapro/src/libtoaster** directory is an example of a SIL implementation, with the following contents:
 - Makefile
 - src directory
 - Makefile
 - toaster.c
 - toaster.c.start
 - toaster.h
 - toaster.h.start
 - bin directory
 - lib directory
- **/usr/share/yumapro/src/libsystem** directory is an example of a user external system API (yp-system) library with the following contents:
 - Makefile
 - src directory
 - Makefile
 - example-system.c
 - example-system.h
 - bin directory
 - lib directory
- **/usr/share/yumapro/src/sil-error** directory is an example of a SIL library that can be used to force errors during editing transactions to help test corner-case error handling. It has the following contents:
 - Makefile
 - src directory
 - Makefile
 - sil-error.c
 - sil-error.h
 - bin directory
 - lib directory
- **/usr/share/yumapro/src/get2-test** directory is an example of a SIL-SA library that shows examples of “get2” callback functions for the module get2-test.yang. It has the following contents:

YumaPro SDK Installation Guide

- Makefile
- src directory
 - Makefile
 - get2-test.c
 - get2-test.h
- bin directory
- lib directory
- **/usr/share/yumapro/src/libif-mib** directory is an example of the IF-MIB converted to YANG library that includes SIL code with the following contents:
 - Makefile
 - src directory
 - Makefile
 - IF-MIB.c
 - IF-MIB.h
 - bin directory
 - lib directory
- **/usr/share/yumapro/src/sil-sa-app** directory is an example of Server Instrumentation Library for Sub-Agent (SIL-SA) application. A SIL-SA application runs as a separate sub-agent process within the same system as the netconfd-pro server. It is used for asynchronous server callbacks.
 - Makefile
 - main.c
- **/usr/share/yumapro/util** directory contains the following files and directories.
 - errmsg-lang/
 - errmsg-tr.py
 - generate-keys.sh
 - host-meta
 - makefile.sil
 - makefile.sil-sa
 - makefile-top.sil
 - restconf.conf
 - restconf-nginx.conf
- **/usr/share/yumapro/src/yp-client** **NOTE:** this directory will only be present if you are using a PTHREADS version of YumaPro SDK, either a binary containing the name yumapro-pthreads- or built from source with PTHREADS=1 as a parameter supplied to make. If present the directory will contain the following yp-client C++ client library examples, each in their own directory:

- c-toaster directory
 - c-toaster.c
 - Makefile
 - README
- capabilities directory
 - capabilities.cpp
 - Makefile
 - README
- libtest directory
 - libtest.cpp
 - Makefile
 - README
- sget-system directory
 - sgtet-system.cpp
 - Makefile
 - README
- toaster directory
 - toaster.cpp
 - Makefile
 - README
- README

A.9 License and Packaging Information

- **/usr/share/doc/yumapro** directory containing the following files:
 - AUTHORS
 - README
 - yumapro-legal-notice.pdf
 - <package-license-file>.pdf
- **/usr/share/doc/<package-name>** directory containing the following files.
NOTE: This directory is only created by Ubuntu:
 - copyright
 - changelog.Debian

B Appendix: Next Steps

After Installation you can start learning more about YumaPro SDK with a set of lessons can be found on the YumaWorks' Knowledge Base "Solutions – Server Developer – Getting Started" at:
<https://yumaworks.freshdesk.com/solution/folders/1000229490>

B.1 Solutions

YumaWorks hosts a Knowledge Base with solutions to many Frequently Asked Questions (FAQs) located at:
<https://yumaworks.freshdesk.com/solution/categories>

The Knowledge Base has the following sections:

- General
- netconfd-pro
- yangcli-pro
- YANG
- Server Developer

B.2 More Documentation

- YumaPro Quickstart Guide:
 - [/usr/share/doc/yumapro/pdf/yumapro-quickstart-guide.pdf](#)
- YumaPro Common User Manual:
 - [/usr/share/doc/yumapro/pdf/yumapro-user-cmn-manual.pdf](#)
- YumaPro Program Specific User Manuals:
 - [/usr/share/doc/yumapro/pdf/yumapro-netconfd-manual.pdf](#)
 - [/usr/share/doc/yumapro/pdf/yumapro-yangcli-manual.pdf](#)
 - [/usr/share/doc/yumapro/pdf/yumapro-yangdiff-manual.pdf](#)
 - [/usr/share/doc/yumapro/pdf/yumapro-yangdump-manual.pdf](#)
 - [/usr/share/doc/yumapro/pdf/yumapro-ypclient-manual.pdf](#)
- YumaPro Developer Manuals:
 - [/usr/share/doc/yumapro/pdf/yumapro-dev-manual.pdf](#)
 - [/usr/share/doc/yumapro/pdf/yumapro-ypsystem-api-guide.pdf](#)
 - [/usr/share/doc/yumapro/pdf/yumapro-api-quickstart-guide.pdf](#)
 - [/usr/share/doc/yumapro/pdf/yumapro-yocto-quickstart-guide.pdf](#)

- `/usr/share/doc/yumapro/pdf/yumapro-ypshow-api-guide.pdf`
- `/usr/share/doc/yumapro/pdf/yumapro-ypsnmp-manual.pdf`

B.3 Unix 'man' pages

The unix 'man' program can be used to get documentation about each program. For example:

- `man yangcli-pro`
- `man yangdump-pro`
- `man yangdump-sdk`
- `man yangdiff-pro`
- `man netconfd-pro`
- `man netconf-subsystem-pro`
- `man make_sil_dir_pro`
- `man make_sil_bundle`
- `man make_sil_sa_dir`
- `man make_sil_sa_bundle`
- `man ypcontrol`
- `man yp-shell`
- `man ypwatcher`
- `man yang-api`
- `man restconf`
- `man ypgnmi-app`
- `man yp-ha-app`
- `man db-api-app`
- `man sil-sa-app`

B.4 Program `--help` option

Each program also has extensive help information available with the `--help` CLI parameter. For example:

- `yangcli-pro --help`
- `yangdump-pro --help`
- `yangdiff-pro --help`
- `netconfd-pro --help`