MG-SOFT Corporation

NetConf Browser 2015
Professional Edition

GETTING STARTED GUIDE

(Document Version: 4.6)

Document published on Thursday, 12-November-2015

Depending on your license, certain functions described in this document may not be available in the version of the software that you are currently using.

Screenshots used in this document may slightly differ from those on your display.

TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

This guide contains instructions for completing basic operations in MG-SOFT NetConf Browser Professional application. Majority of instructions are provided on a step-by-step basis, which should help the reader start using the software effectively.

It is supposed that you are familiar with using a graphical computer environment, such as choosing a main menu command or a pop-up command, selecting items, closing windows and dialog boxes, etc.

All program commands in this manual are written in bold and italic letters. Individual commands in combinations of commands are separated by the “/” character. For example:

Edit / Preferences – which means: click the “Edit” entry in the menu bar and select the “Preferences” command from the “View” menu.

All hyperlinks in text are marked with blue colored letters, e.g., Starting NetConf Browser. Clicking a hyperlink opens the page, which the hyperlink points to.

The content of this guide is listed in the Table of Contents.

1.1 Product Description

MG-SOFT NetConf Browser Professional Edition is powerful and user-friendly NETCONF client application that lets you retrieve, modify, install and delete the configuration of any NETCONF server device in the network.

The software can load any standard or vendor specific YANG or YIN module and display its contents in a visual manner, where module elements are represented in a hierarchical tree structure, containing nodes on which NETCONF operations can be performed.

NetConf Browser offers an intuitive user interface that lets you easily retrieve the device configuration and state data as well as modify the device configuration via the NETCONF v1.1 (RFC 6241) or NETCONF v1.0 (RFC 4741) protocol operations (get, get-config, lock, unlock, edit-config, copy-config, delete-config, commit, etc.). The software supports establishing NETCONF sessions over SSH2 and TLS v1.2 secure transport protocols. In addition to providing quick access to common NETCONF operations, the software also implements advanced tools, like the RFC 6110-compliant NETCONF Content Editor and Validator that lets you easily compose any type of NETCONF XML document and validate it using the DSDL schemas, which are automatically generated from the selected YANG modules. Furthermore, the software support subscribing to and receiving NETCONF notifications (as specified in RFC 5277). In addition, NetConf Browser also supports the NETCONF get-schema operation (specified in RFC 6022) that lets you download schema definitions (e.g., YANG and YIN modules) from remote NETCONF servers.

MG-SOFT NetConf Browser is a Java^{TM} application that can be installed and used on Windows, Linux and Mac OS X operating systems with Java Runtime Environment version 7.0 (a.k.a. JRE 1.7) or later installed.

2 INSTALL NETCONF BROWSER PROFESSIONAL EDITION

This section presents the basic system requirements your computer has to meet to install and use MG-SOFT NetConf Browser 2015 Professional Edition, and it describes the procedure of installing MG-SOFT NetConf Browser on Windows, Linux, Mac OS X operating systems.

2.1 Requirements

MG-SOFT NetConf Browser is a Java^{TM} application that can be installed and used on Windows, Linux and Mac OS X operating systems with Oracle **Java Runtime Environment version 7.0 (a.k.a. JRE 1.7) or later** installed. You can download Oracle Java from the following web page: www.java.com

Additionally, **administrator/root** user privileges are required to install the software.

2.1.1 Windows Operating System

The Windows version of MG-SOFT NetConf Browser 2015 v4.6 requires one of the following 32-bit (where applicable) or 64-bit Microsoft Windows operating systems:

- Windows Vista,
- Windows Server 2008,
- Windows 7,
- Windows Server 2012,
- Windows 8.x,
- Windows 10.

2.1.2 Linux Operating System

The Linux version of MG-SOFT NetConf Browser 2015 v4.6 requires one of the following 32-bit or 64-bit Linux distributions running on Intel x86/x86_64 architecture:

- Red Hat Enterprise Linux 5 or newer,
- Fedora Core 8 or newer,
- SUSE 11 or newer,
- Ubuntu 9 or newer,
- Slackware 13 or newer.

For the most recent information about the supported distributions, please refer to the release notes (READ_ME.TXT) of the current software release.

2.1.3 Mac OS X Operating System

MG-SOFT NetConf Browser 2015 v4.6 requires one of the following Mac OS X versions running on Intel-based Mac platform:

- Mac OS X v10.7.3+ Lion
- Mac OS X v10.8.x Mountain Lion
- Mac OS X v10.9.x Mavericks
- Mac OS X v10.10.x Yosemite
- Mac OS X v10.11.x El Capitan

2.2 Installing NetConf Browser Professional Edition

Before you install NetConf Browser Professional Edition on your computer, first make sure your computer meets the system requirements described in the Requirements section.

2.2.1 Windows Operating System

1. Put the MG-SOFT NetConf Browser Professional Edition CD into your CD or DVD drive.

Note: To install the software on Windows, you need to have administrative privileges.

2. Click the ***Start*** button and select the ***Run*** command.
3. The Run dialog box appears (Figure 1).

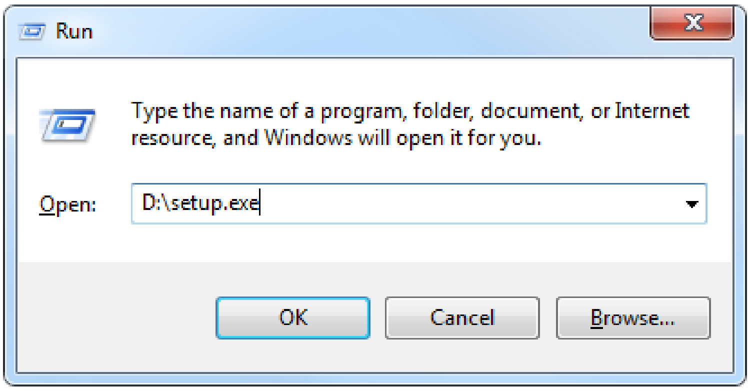

Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
D:\setup.exe
OK
Cancel
Browse...

Figure 1: Run dialog box

4. Into the ***Open*** input line, type `D:\setup` and click the ***OK*** button.

Note: `D` is the letter assigned to the CD or DVD drive. If your CD or DVD has a different letter, type that one instead of `D`.

5. Follow the instructions displayed on the screen.

Once the installation is complete, you can start MG-SOFT NetConf Browser program.

2.2.2 Linux Operating System

Before the installation, please close all running MG-SOFT applications and uninstall any previous version of MG-SOFT NetConf Browser Professional Edition.

1. Put the MG-SOFT NetConf Browser Professional Edition CD into your CD-ROM drive and mount the CD.
2. The software comes in three different software packages (`rpm`, `deb` and `tgz`). Depending on your Linux distribution, run one of the following commands in a Terminal window to install the software:

   a) Linux distributions with the **RPM** package manager (RHEL, Fedora, SUSE, Mandriva, etc):

   - ❑ On a 32-bit (i386) Linux distribution with the RPM package manager, install the RPM package:

     ```
     # rpm -ivh mgNetConfBrowser_2015-X.X-X.i386.rpm
     ```

   - ❑ On a 64-bit (x86_64) Linux distribution with the RPM package manager, install the RPM package containing the 64-bit build of the software, as follows:

     ```
     # rpm -ivh mgNetConfBrowser_2015-X.X-X.x86_64.rpm
     ```

   b) Linux distributions with the **DPKG** package manager (Debian, Ubuntu, etc.):

   - ❑ On a 32-bit (i386) Linux distribution with the DPKG package manager, install the DEB package:

     ```
     # dpkg -i mgNetConfBrowser-2015_X.X-X_i386.deb
     ```

   - ❑ On a 64-bit (x86_64/amd64) Linux distribution with the DPKG package manager, install the DEB package containing the 64-bit build of the software, as follows:

     ```
     # dpkg -i mgNetConfBrowser-2015_X.X-X_x86_64.deb
     ```

   c) Linux distributions with the **installpkg** package manager (e.g., Slackware):

   - ❑ On a 32-bit (i386) Linux distribution with the installpkg package manager, install the 32-bit TGZ package:

     ```
     # installpkg mgNetConfBrowser_2015-X.X-i386-X.tgz
     ```

   - ❑ On a 64-bit (x86_64) Linux distribution with the installpkg package manager, install the corresponding TGZ package:

     ```
     # installpkg mgNetConfBrowser_2015-X.X-x86_64-X.tgz
     ```

In case you have KDE or GNOME Environments installed on your machine, the installation will add an entry to the K Menu or Gnome Menu respectively.

Once the installation is complete, you can start MG-SOFT NetConf Browser program.

2.2.3 Mac OS X Operating System

1. Double-click the MG-SOFT NetConf Browser disk image file (`.dmg`) that you have downloaded from MG-SOFT's Website or obtained on a removable medium.

   Tip: Use **Finder** to navigate to the DMG file if it is not located on your desktop.

2. The contents of the double-clicked disk image displays in a Finder window. MG-SOFT NetConf Browser virtual drive appears on the desktop. Drag&drop the "MG-SOFT NetConf Browser.app" from the MG-SOFT NetConf Browser virtual drive to the "Applications" folder.

Once the installation is complete, you can start MG-SOFT NetConf Browser for Mac OS X from the Finder.

3 START NETCONF BROWSER PROFESSIONAL EDITION

3.1 Starting NetConf Browser

3.1.1 Windows Operating System

1. In Windows operating systems, select the ***Start*** / ***Programs*** / ***MG-SOFT NetConf Browser*** / ***NetConf Browser*** command from the Windows taskbar.
2. The NetConf Browser desktop appears and you can start using the software. Please refer to the Apply License Key section for instructions on how to apply your license.

3.1.2 Linux Operating System

The easiest way to start NetConf Browser under Linux operating system is to use the start menu. The start menu can be displayed from the desktop taskbar.

1. If you have the KDE or GNOME desktop environment installed, display the ***K/Gnome*** start menu by clicking the button in the left corner of your taskbar.
2. To start NetConf Browser, search for and use the ***MG-SOFT NetConf Browser*** / ***NetConf Browser*** command.
3. The NetConf Browser desktop appears and you can start using the software. Please refer to the Apply License Key section for instructions on how to apply your license.

Tip: To start the software from the command prompt, open the Terminal window, CD to the directory where the "`mgNetConfBrow`.jar" file is (/usr/local/mg-soft/mgnetconfbrowser/java/) and run the following command:

```
# java -jar mgNetConfBrow.jar
```

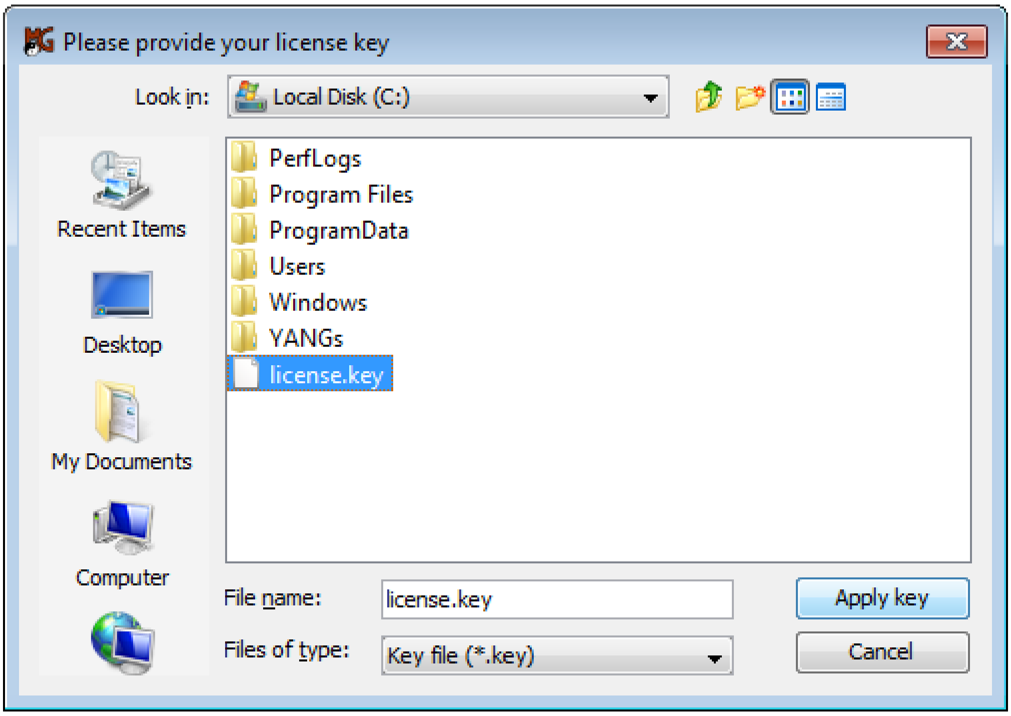
3.1.3 Mac OS X Operating System

1. Open the **Finder** and select the ***Applications*** entry in the panel on the left.
2. Select and double-click the "MG-SOFT NetConf Browser.app" icon to start the NetConf Browser application.
3. The NetConf Browser desktop appears and you can start using the software. Please refer to the Apply License Key section for instructions on how to apply your license.

4 APPLY LICENSE KEY

To use MG-SOFT NetConf Browser without limitations, you need to apply a valid `license.key` file to the software, as follows:

1. Select the ***Help / Apply License Key*** command from the main menu or click the ***Apply License*** toolbar button (the latter is displayed only when the software is run without a valid license key file in place).
2. A dialog box appears that lest you select and apply your license key (Figure 2).



File name: license.key

Files of type: Key file (*.key)

Figure 2: Selecting and applying the license.key file

3. Navigate to the drive and folder containing your `license.key` file for MG-SOFT NetConf Browser Professional Edition, select the `license.key` file and click the ***Apply License*** button in the license key selection dialog box.
4. The software will copy the selected `license.key` file to the proper location in order for NetConf Browser to read it and unlock its features respectively after a restart.
5. Exit NetConf Browser by choosing the ***File / Exit*** command and restart it, as described in the Starting NetConf Browser section. Now, the selected license should be applied and you can start using the software without licensing restrictions.

Tip: You can check if the license key has been properly applied by verifying if the About NetConf Browser dialog box (accessible via the ***Help / About*** command) displays your license details correctly.

5 CONNECT TO REMOTE NETCONF SERVER

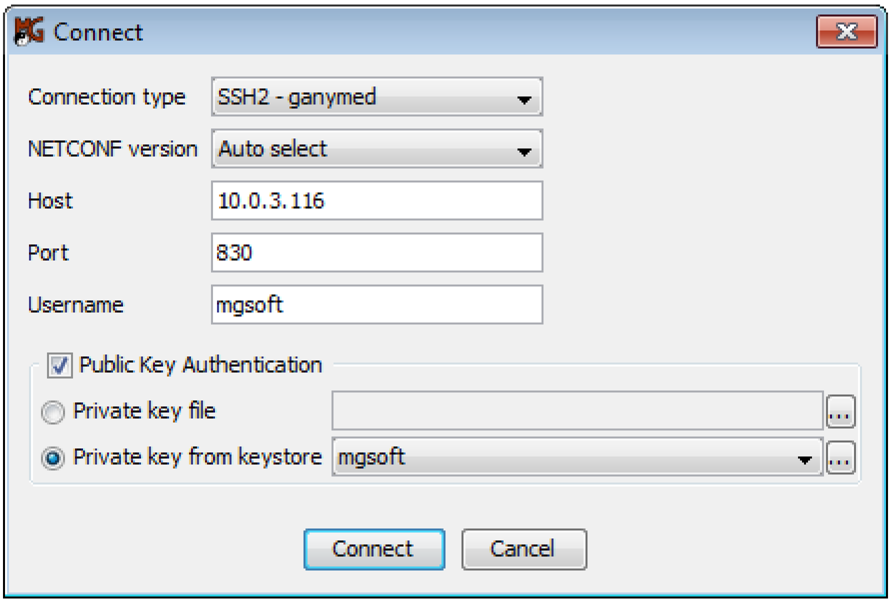
MG-SOFT NetConf Browser supports NETCONF over SSH2 and NETCONF over TLS 1.2. In order to manage a NETCONF device (server), you first need to establish a NETCONF connection to it using either SSH (Secure Socket Shell) or TLS (Transport Layer Security) transport protocol. Both transports can include public key authentication mechanism, as described in this section.

Furthermore, in the process of establishing a connection, the NETCONF server reports its capabilities to the NetConf Browser, as described in this section.

5.1 Connecting to Remote Server Using NETCONF over SSH

This section describes how to connect NetConf Browser to a NETCONF server using SSH2 transport protocol and either simple password authentication method or the public key authentication method.

1. In the main window, choose the ***File / Connect*** command or click the ***Connect*** toolbar button.
2. The Connect dialog box appears (Figure 3), which is used for connecting NetConf Browser to a NETCONF server device.



Connect

Connection type: SSH2 - ganymed

NETCONF version: Auto select

Host: 10.0.3.116

Port: 830

Username: mgsoft

☑ Public Key Authentication

○ Private key file

● Private key from keystore: mgsoft

Figure 3: Specifying the NETCONF over SSH connection parameters

3. In the ***Connection type*** drop-down list in the Connect dialog box, select the SSH2 transport protocol implementation to be used for the NETCONF connection (e.g., `SSH2-ganymed`).

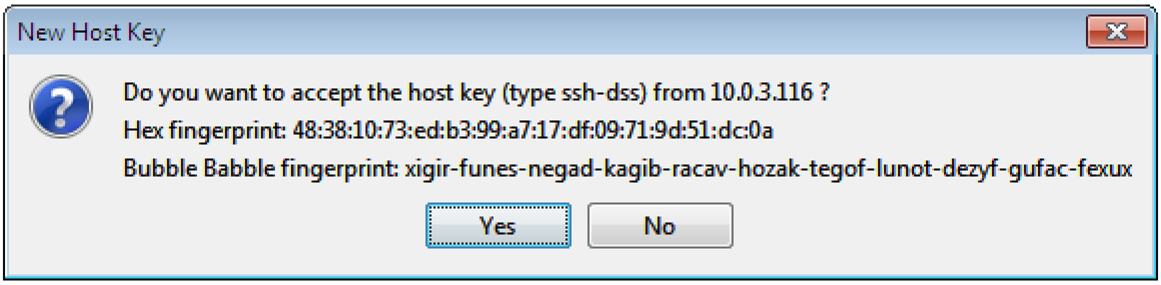
> **Tip:** In case you experience SSH connection problems, try selecting a different SSH2 connection type (protocol implementation) from this drop-down list.

4. In the ***NETCONF version*** drop-down list leave the **`Auto select`** option selected for the NetConf Browser to automatically select and use the highest version of

NETCONF protocol supported by both peers. If you want to use a specific version of the NETCONF protocol, select the corresponding entry from this drop-down list (e.g., **`1.1`** or **`1.0`**). When a specific version is selected, NetConf Browser will not establish NETCONF session with the server if the latter does not support the selected version of the NETCONF protocol.

5. In the ***Host*** input line, enter the IP address or the hostname of the NETCONF server you wish to connect to.
6. Into the ***Port*** input line, enter the TCP port number on which the NETCONF server listens to for incoming client connections. The default port number for NETCONF over SSH is **830**.
7. Into the ***Username*** input line, enter your username.
8. Select the desired user authentication method:
   - ❑ To use the password authentication mechanisms with SSH, make sure the ***Public key authentication*** checkbox is not checked. This is the default option.
   - ❑ To use the public key authentication mechanism with SSH instead of password authentication, check the ***Public key authentication*** checkbox and specify the private key to be used by:
     - ❑ Selecting the external file that contains the user's private key in PEM format, or
     - ❑ Selecting the digital certificate containing the user's private key from the built-in keystore.

   Note: The corresponding user's public key must be installed on the respective NETCONF server device.
9. Click the ***Connect*** button. NetConf Browser will try to establish an SSH connection to the specified device using the connection settings configured above. If successful and the given server has been contacted for the first time, the New Host Key dialog box appears (Figure 4).



Do you want to accept the host key (type ssh-dss) from 10.0.3.116 ?

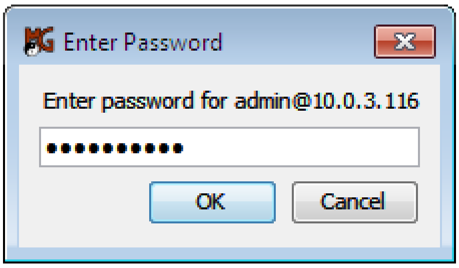
Hex fingerprint: 48:38:10:73:ed:b3:99:a7:17:df:09:71:9d:51:dc:0a

Bubble Babble fingerprint: xigir-funes-negad-kagib-racav-hozak-tegof-lunot-dezyf-gufac-fexux

Figure 4: Viewing the server SSH host key

10. The New Host Key dialog box displays the SSH host key fingerprint of the server (in hexadecimal as well as in human readable "Bubble Babble" form) and prompts you to either accept or reject it. If you trust the given server, click the ***Yes*** button. This will accept the key and store it in the NetConf Browser cache so you will not be prompted again when connecting to the same server. If you click the ***No*** button, the connection is aborted.
11. After accepting the SSH host key, the Enter Password dialog box appears (Figure 5) if you have selected the password authentication mechanism in the Connect dialog box. If you have selected the ***Public key authentication*** option in the Connect dialog box,

you do not need to enter the password for authenticating the user. Instead, the software may prompt you with a dialog box to enter the password for the private key. In such case, enter the password to access the private key.



Enter password for admin@10.0.3.116

●●●●●●●●●●●

Figure 5: Entering a user password for the NETCONF connection

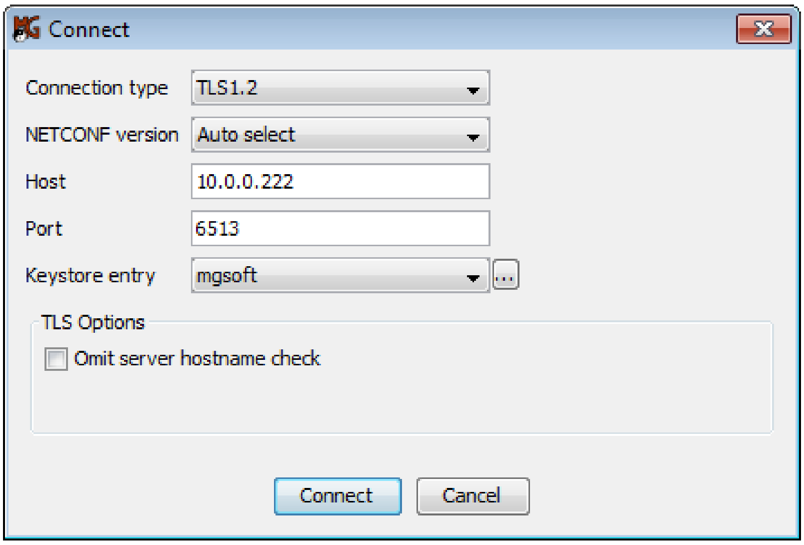
12. Into the ***Password*** input line, enter your password and click the ***OK*** button.
13. After successfully authenticating the user on remote device, the NETCONF session handshake occurs where the server and client exchange the Hello messages with NETCONF capabilities they support. When the capabilities are successfully exchanged, the NETCONF session is established (indicated by the **`Connected to`** status in the status bar – see Figure 9) and you can start managing the device configuration and retrieve device state information using NetConf Browser.

5.2 Connecting to Remote Server Using NETCONF over TLS

This section describes how to connect NetConf Browser to a NETCONF server using TLS 1.2 transport protocol and public key authentication method using X.509 digital certificates.

Note: Java 7.0+ (a.k.a. JRE 1.7+) must be installed on your computer to use NETCONF over TLS.

1. In the main window, choose the ***File / Connect*** command or click the ***Connect*** toolbar button.
2. The Connect dialog box appears (Figure 6), which is used for connecting NetConf Browser to a NETCONF server device.



Connection type: TLS1.2

NETCONF version: Auto select

Host: 10.0.0.222

Port: 6513

Keystore entry: mgsoft

TLS Options

□ Omit server hostname check

Figure 6: Specifying the NETCONF over TLS connection parameters

3. In the ***Connection type*** drop-down list in the Connect dialog box, select the **`TLS 1.2`** transport protocol to be used for the NETCONF connection.

4. In the ***NETCONF version*** drop-down list, leave the **`Auto select`** option selected for the NetConf Browser to automatically select and use the highest version of NETCONF protocol supported by both peers. If you want to use a specific version of the NETCONF protocol, select the corresponding entry from this drop-down list (e.g., **`1.1`** or **`1.0`**). When a specific version is selected, NetConf Browser will not establish NETCONF session with the server if the latter does not support the selected version of the NETCONF protocol.

5. In the ***Host*** input line, enter the IP address or the hostname of the NETCONF server you wish to connect to.

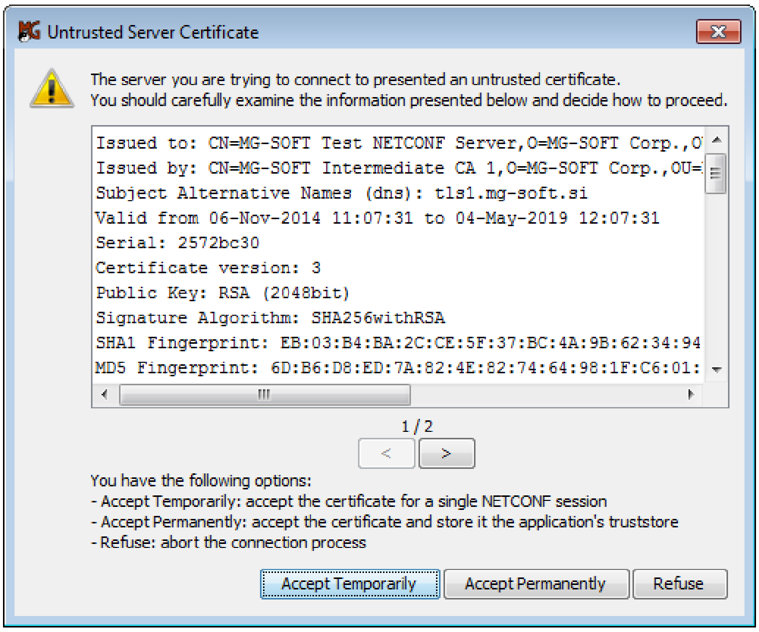
6. Into the ***Port*** input line, enter the port number on which the NETCONF server listens to for incoming client connections. The default TCP port number for NETCONF over TLS is **6513**.

7. In the ***Keystore entry*** drop-down list, select the digital certificate containing the user's private and public key from the built-in keystore. The ***Browse*** (**…**) button next to this input line lets you open the Manage Certificates dialog box where you can manage (import, generate, delete, etc.) digital certificates.

> **Note:** The corresponding user's public key must be available on the NETCONF server device.

8. In the TLS Options frame, select the ***Omit server hostname check*** option if you do not want the NetConf Browser to check if the server hostname matches the one specified in the server's digital certificate.

9. Click the ***Connect*** button. NetConf Browser will try to establish a TLS connection to the specified device using the connection settings configured above. If successful and the given server presents a digital certificate that has not been signed by a Certificate Authority (CA) you trust, the Untrusted Server Certificate dialog box appears (Figure 7).



Untrusted Server Certificate

The server you are trying to connect to presented an untrusted certificate.
You should carefully examine the information presented below and decide how to proceed.

```
Issued to: CN=MG-SOFT Test NETCONF Server,O=MG-SOFT Corp.,O
Issued by: CN=MG-SOFT Intermediate CA 1,O=MG-SOFT Corp.,OU=
Subject Alternative Names (dns): tls1.mg-soft.si
Valid from 06-Nov-2014 11:07:31 to 04-May-2019 12:07:31
Serial: 2572bc30
Certificate version: 3
Public Key: RSA (2048bit)
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: EB:03:B4:BA:2C:CE:5F:37:BC:4A:9B:62:34:94
MD5 Fingerprint: 6D:B6:D8:ED:7A:82:4E:82:74:64:98:1F:C6:01:
```

1 / 2

You have the following options:
- Accept Temporarily: accept the certificate for a single NETCONF session
- Accept Permanently: accept the certificate and store it the application's truststore
- Refuse: abort the connection process

Figure 7: Examining the NETCONF server certificate information

10. Carefully examine the information about the digital certificate or certificate chain displayed in the Untrusted Server Certificate dialog box and proceed as follows:
    - If you trust this server certificate, click either the ***Accept Temporarily*** or the ***Accept Permanently*** button to accept the certificate and continue with establishing the NETCONF over TLS connection.
    - If you do not trust this server certificate, click the ***Refuse*** button to reject the certificate and abort the connection.
11. Depending on the preferences settings for accessing the built-in keystore, i.e., if the *Prompt for password on first access* or the *Prompt for password on each access* option is selected, the **Enter Keystore Password** dialog box appears (Figure 8). In such case, enter the password to access the specified digital certificate in the keystore and click the ***OK*** button.

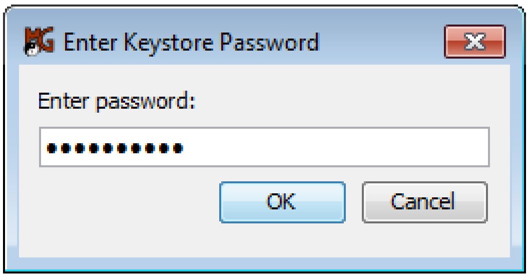


Figure 8: Entering a password for accessing the built-in keystore

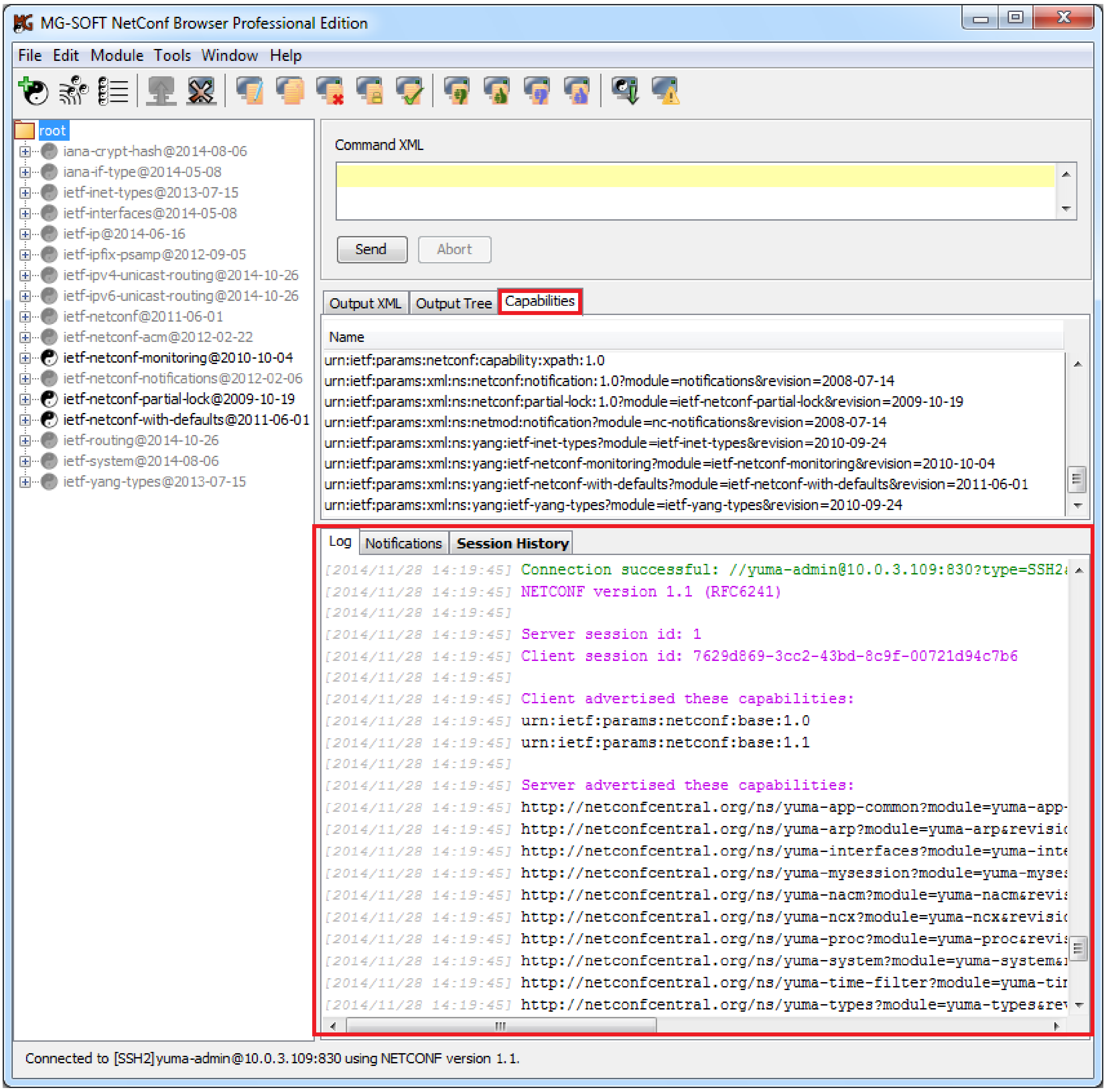
12. After successfully authenticating the user on remote device, the NETCONF session handshake occurs where the server and client exchange the Hello messages with NETCONF capabilities they support. When the capabilities are successfully exchanged, the NETCONF session is established (indicated by the **`Connected to`** status in the status bar – see Figure 9) and you can start managing the device configuration and retrieve device state information using NetConf Browser.

5.3 Information About NETCONF Session and Server Capabilities

After successfully establishing a connection with a NETCONF server, the **Log** panel at the bottom of the NetConf Browser main window displays the following information (Figure 9):

- **status of the connection**
- **session id** (client and server session ID)
- **client and server capabilities**

Capabilities of the NETCONF server the NetConf Browser is currently connected to are also listed in the **Capabilities tab** in the central panel of the main window (Figure 9).



```
iana-crypt-hash@2014-08-06
iana-if-type@2014-05-08
ietf-inet-types@2013-07-15
ietf-interfaces@2014-05-08
ietf-ip@2014-06-16
ietf-ipfix-psamp@2012-09-05
ietf-ipv4-unicast-routing@2014-10-26
ietf-ipv6-unicast-routing@2014-10-26
ietf-netconf@2011-06-01
ietf-netconf-acm@2012-02-22
ietf-netconf-monitoring@2010-10-04
ietf-netconf-notifications@2012-02-06
ietf-netconf-partial-lock@2009-10-19
ietf-netconf-with-defaults@2011-06-01
ietf-routing@2014-10-26
ietf-system@2014-08-06
ietf-yang-types@2013-07-15
```

```
urn:ietf:params:netconf:capability:xpath:1.0
urn:ietf:params:xml:ns:netconf:notification:1.0?module=notifications&revision=2008-07-14
urn:ietf:params:xml:ns:netconf:partial-lock:1.0?module=ietf-netconf-partial-lock&revision=2009-10-19
urn:ietf:params:xml:ns:netmod:notification?module=nc-notifications&revision=2008-07-14
urn:ietf:params:xml:ns:yang:ietf-inet-types?module=ietf-inet-types&revision=2010-09-24
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-with-defaults?module=ietf-netconf-with-defaults&revision=2011-06-01
urn:ietf:params:xml:ns:yang:ietf-yang-types?module=ietf-yang-types&revision=2010-09-24
```

```
[2014/11/28 14:19:45] Connection successful: //yuma-admin@10.0.3.109:830?type=SSH2
[2014/11/28 14:19:45] NETCONF version 1.1 (RFC6241)
[2014/11/28 14:19:45]
[2014/11/28 14:19:45] Server session id: 1
[2014/11/28 14:19:45] Client session id: 7629d869-3cc2-43bd-8c9f-00721d94c7b6
[2014/11/28 14:19:45]
[2014/11/28 14:19:45] Client advertised these capabilities:
[2014/11/28 14:19:45] urn:ietf:params:netconf:base:1.0
[2014/11/28 14:19:45] urn:ietf:params:netconf:base:1.1
[2014/11/28 14:19:45]
[2014/11/28 14:19:45] Server advertised these capabilities:
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-app-common?module=yuma-app-
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-arp?module=yuma-arp&revisi
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-interfaces?module=yuma-inte
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-mysession?module=yuma-myse
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-nacm?module=yuma-nacm&revi
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-ncx?module=yuma-ncx&revisi
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-proc?module=yuma-proc&revi
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-system?module=yuma-system&
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-time-filter?module=yuma-ti
[2014/11/28 14:19:45] http://netconfcentral.org/ns/yuma-types?module=yuma-types&re
```

Connected to [SSH2]yuma-admin@10.0.3.109:830 using NETCONF version 1.1.

Figure 9: Viewing connection details and sever capabilities in the Log panel

A NETCONF **capability** is a set of functionality that supplements the base NETCONF specification. NETCONF protocol allows a client to discover the set of protocol extensions supported by a server (e.g., new operations, modifications of operations, etc.), as well as the implemented modules, their revisions, and optional features and deviations. These "capabilities" permit the client to adjust its behavior to take advantage of the features exposed by the device. Additional standard and vendor-specific capabilities can be defined over time.

When the NETCONF session is opened, client and server first exchange Hello messages (displayed in the **Session History** tab at the bottom window panel) providing information about the capabilities they support. Each peer must support at least the base NETCONF capability. A NETCONF capability is identified with a URI (Uniform Resource Identifier), e.g., the base NETCONF capability URI is: "`urn:ietf:params:netconf:base:1.0`" (it can be shorter referenced as "`:base`"). For more information about NETCONF capabilities, please refer to the NETCONF specification.

Depending on the reported server capabilities, some features may be disabled or enabled in NetConf Browser. For example: if the server reports the standard "`:candidate`" capability (`urn:ietf:params:netconf:capability:candidate:1.0`), meaning that the device supports the candidate configuration datastore, the candidate datastore can be used as argument (target or source - where appropriate) of the get-config, edit-config, copy-config, lock and unlock operations, and additional (non-base) NETCONF operations are enabled (commit and discard-changes). Likewise, if the standard "`:validate`" capability is reported, meaning that the device supports the validate protocol operation (checking configuration for errors before applying it), the NETCONF validate operation is enabled in the software, etc.

Similarly, YANG/YIN modules that are not supported (i.e., advertised in Hello message) by the given server are disabled by default. Nodes of disabled modules are depicted with grayed-out icons (e.g.: ●) and performing NETCONF operations (e.g., get, get-config, edit-config, etc.) on these modules is disabled (Figure 10). If you want to change this behavior, i.e., enable also modules not advertised by the server, open the Preferences dialog box (***Edit/Preferences***) and uncheck the ***Disable unsupported modules on connect*** checkbox in the General view of the Preferences dialog box.

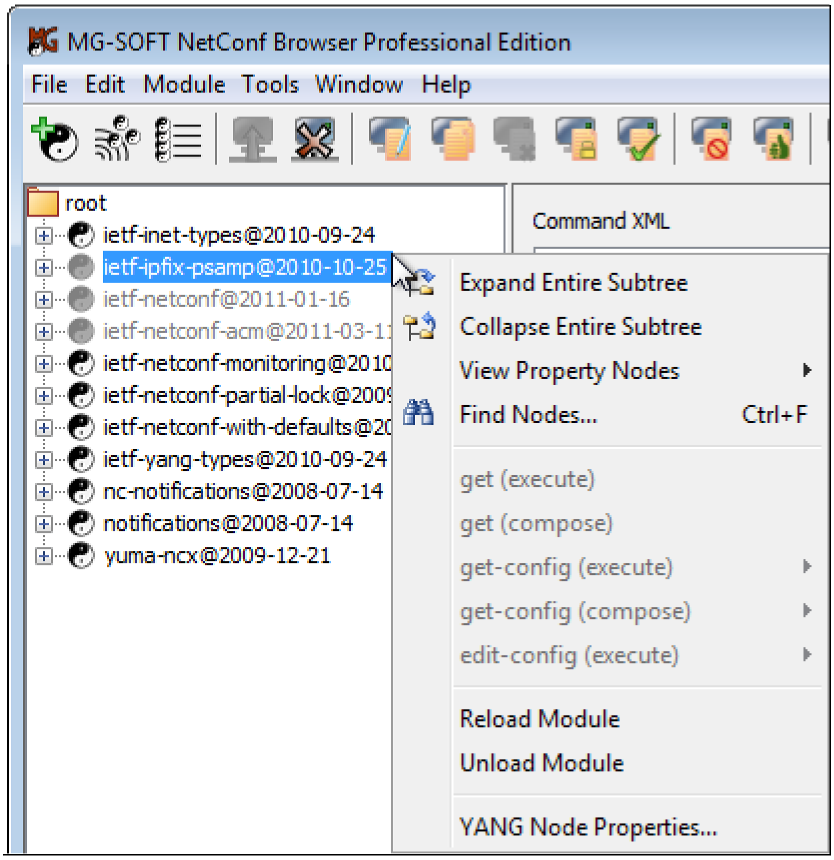


Figure 10: By default, YANG modules not advertised by the connected server are disabled

5.4 Managing Digital Certificates (X.509)

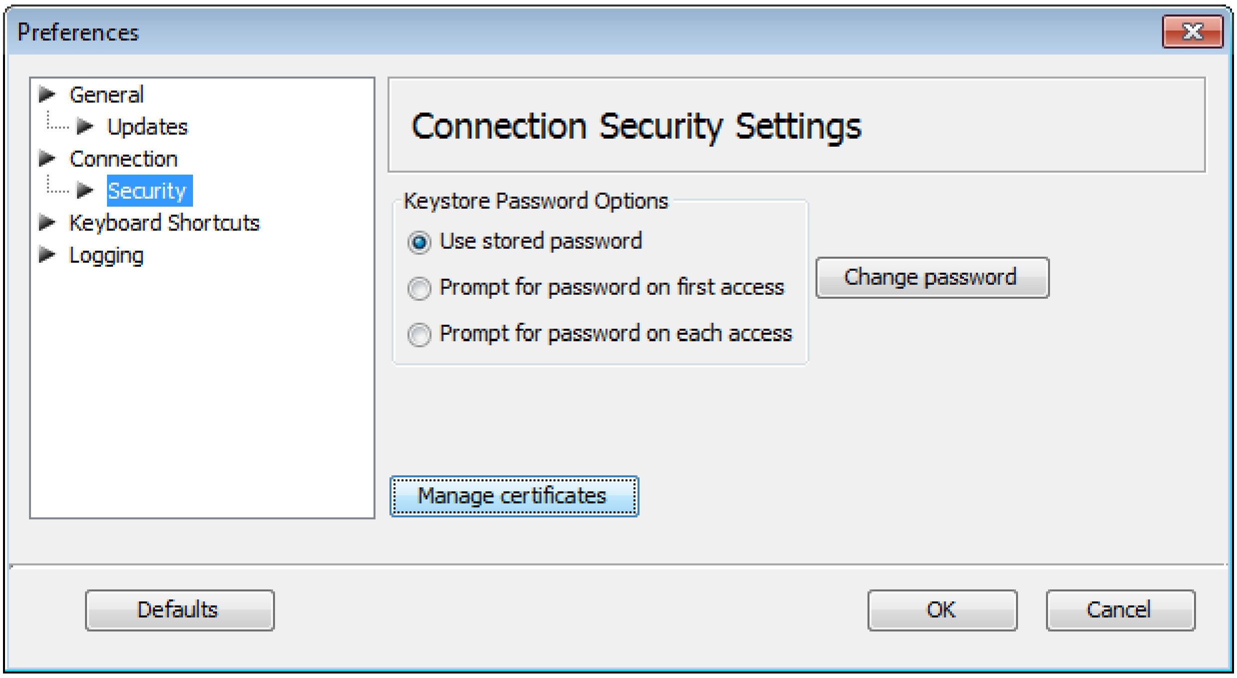
MG-SOFT NetConf Browser features a built-in keystore that stores digital certificates used for establishing NETCONF over TLS connections and NETCONF over SSH connections with public key authentication. This section describes how to view and manage the contents of the keystore in the Manage Certificate window.

5.4.1 Opening Manage Certificate Window

The Manage Certificate window lets you view, generate, import, export, and delete digital certificates in NetConf Browser.

To open the Manage Certificates window, proceed as follows:

1. Select the ***View / Preferences*** command from the main menu to open the Preferences dialog box.



Connection Security Settings

Keystore Password Options

● Use stored password

○ Prompt for password on first access

○ Prompt for password on each access

Figure 11: NetConf Browser Preferences dialog box, Connection Security Settings panel

2. In the navigation panel on the left hand-side, click the ***Connection / Security*** entry to display the Connection Security Settings panel in the right portion of the dialog box.
3. Click the **Manage Certificates** button in the Connection Security Settings panel.
4. Depending on the **keystore password option** selected in the Connection Security Settings panel, the software may prompt you with a dialog box to enter the keystore password, as follows:

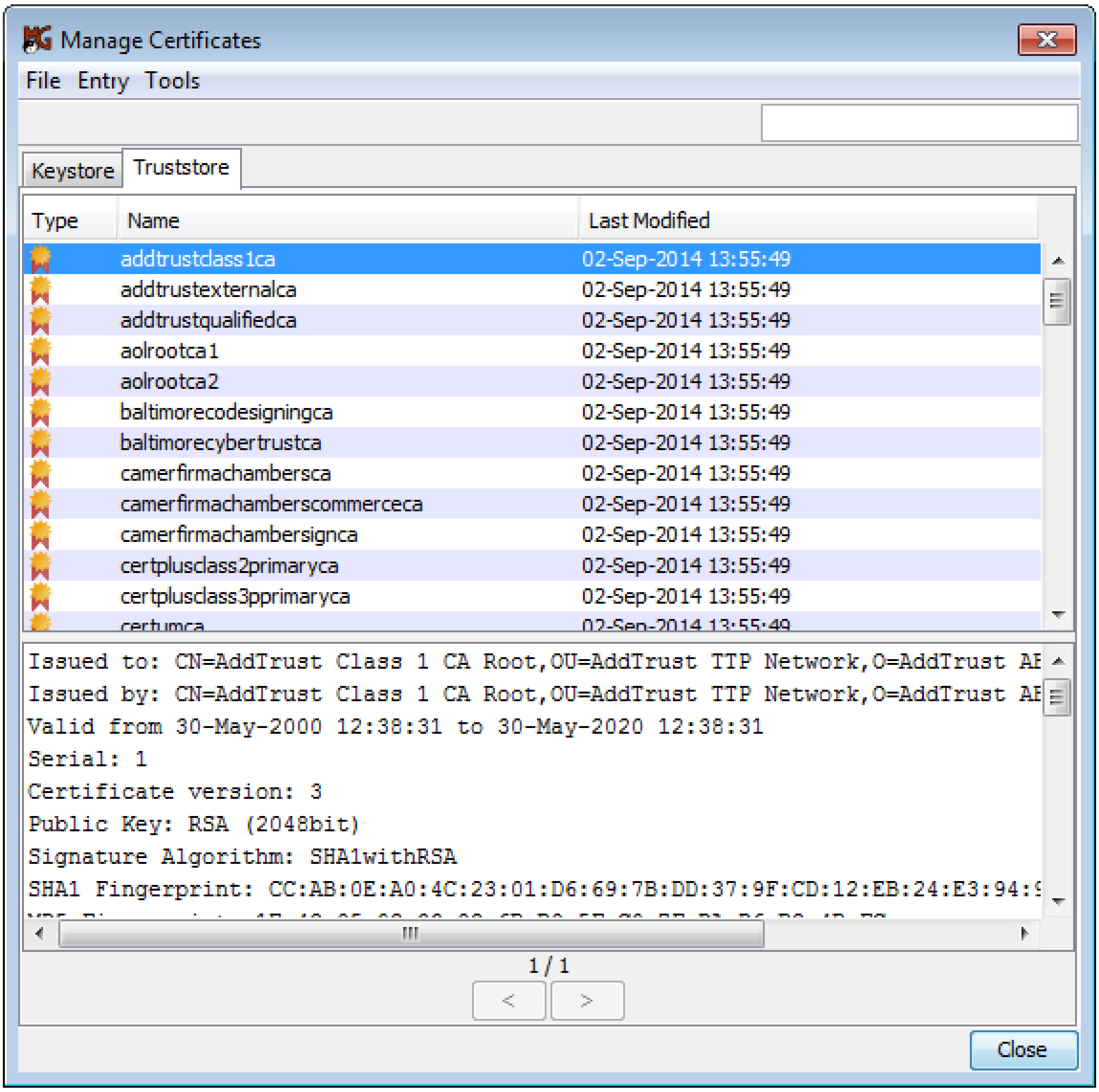
Note 1: The keystore must be protected with a password, but this password can be handled in three different ways, depending on the keystore password option selected – as described below.

Note 2: Keystore password is used for checking the integrity of the keystore and for encrypting the digital certificates (private keys) in it.

Note 3: NetConf Browser comes with a pre-defined stored password (`mgpassword`). You can change the keystore password by clicking the ***Change Password*** button and enter the old and new password. **If you change the keystore password, make sure not to forget or lose it, otherwise you will not be able to access digital certificates in the keystore anymore** (unless the *Use stored password* option is selected).

- ❑ If the ***Use stored password*** option is selected (default), you are not prompted for the password. NetConf Browser stores the (encrypted) password in a file and uses the password automatically whenever needed (i.e., every time you open the Manage Certificates window or establish a NETCONF session that requires a certificate). This is the most convenient, but least secure of the three options available.
- ❑ If the ***Prompt for password on first access*** option is selected, you are prompted for the password the first time (after starting the program) you open the Manage Certificates window or establish a NETCONF session that requires a certificate from the keystore. Once you have entered the correct password, NetConf Browser keeps it in memory and supplies it automatically instead of you whenever needed. This is a relatively convenient and secure option.
- ❑ If the ***Prompt for password on each access*** option is selected, you are prompted for the password every time you access a digital certificate in the keystore (i.e., every time you open the Manage Certificates window or establish a NETCONF session that requires a certificate from the keystore). If this option is selected, NetConf Browser does not store the password anywhere. Hence, this may be considered the most secure option.

5. If the **Enter Keystore Password** dialog box appears (Figure 8), enter the password to open the keystore in the Manage Certificate window and click the ***OK*** button.
6. The Manage Certificates window appears, containing two tabs:
   - ❑ **Keystore**
     A store of user digital certificates containing public-private key pairs, either generated by or imported into NetConf Browser.
   - ❑ **Truststore**
     A trusted root store of certificate authorities (CAs) and their digital certificates (this trusted root store is a part of the Oracle Java 7 distribution).



Type	Name	Last Modified
	addtrustclass1ca	02-Sep-2014 13:55:49
	addtrustexternalca	02-Sep-2014 13:55:49
	addtrustqualifiedca	02-Sep-2014 13:55:49
	aolrootca1	02-Sep-2014 13:55:49
	aolrootca2	02-Sep-2014 13:55:49
	baltimorecodesigningca	02-Sep-2014 13:55:49
	baltimorecybertrustca	02-Sep-2014 13:55:49
	camerfirmachambersca	02-Sep-2014 13:55:49
	camerfirmachamberscommerceca	02-Sep-2014 13:55:49
	camerfirmachambersignca	02-Sep-2014 13:55:49
	certplusclass2primaryca	02-Sep-2014 13:55:49
	certplusclass3pprimaryca	02-Sep-2014 13:55:49
	certumca	02-Sep-2014 13:55:49

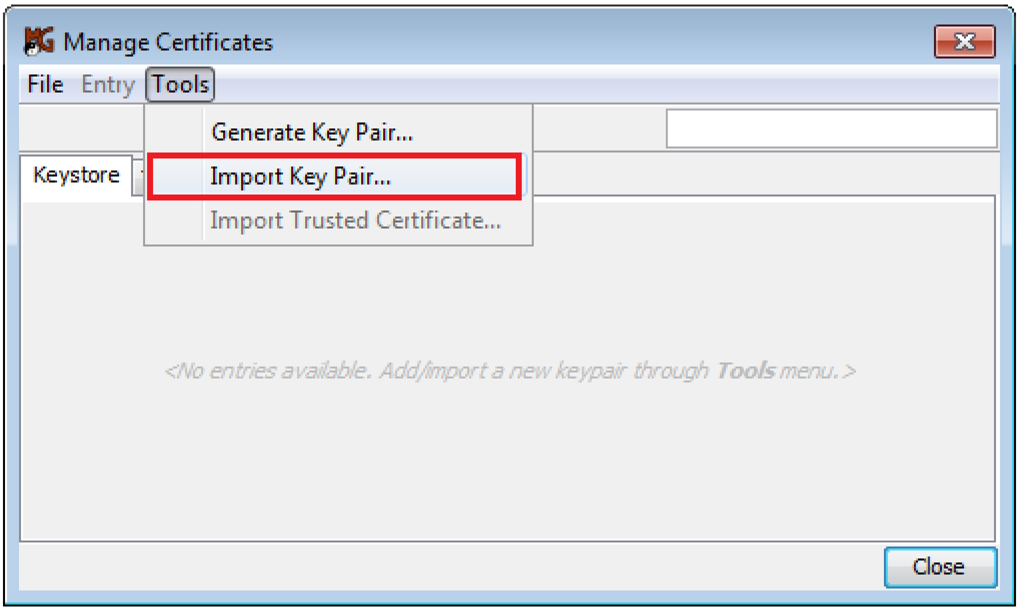
```
Issued to: CN=AddTrust Class 1 CA Root,OU=AddTrust TTP Network,O=AddTrust AE
Issued by: CN=AddTrust Class 1 CA Root,OU=AddTrust TTP Network,O=AddTrust AE
Valid from 30-May-2000 12:38:31 to 30-May-2020 12:38:31
Serial: 1
Certificate version: 3
Public Key: RSA (2048bit)
Signature Algorithm: SHA1withRSA
SHA1 Fingerprint: CC:AB:0E:A0:4C:23:01:D6:69:7B:DD:37:9F:CD:12:EB:24:E3:94:9
```

Figure 12: The Manage Certificate window, Truststore tab contains trusted root CAs

5.4.2 Importing Digital Certificates

If you already have a digital certificate (X.509) in PKCS #12 format (`*.p12` or `*.pfx`), containing a public-private key pair that you would like to use for NETCONF over TLS or NETCONF over SSH with public key authentication session, you can import this certificate into NetConf Browser, as described in this section.

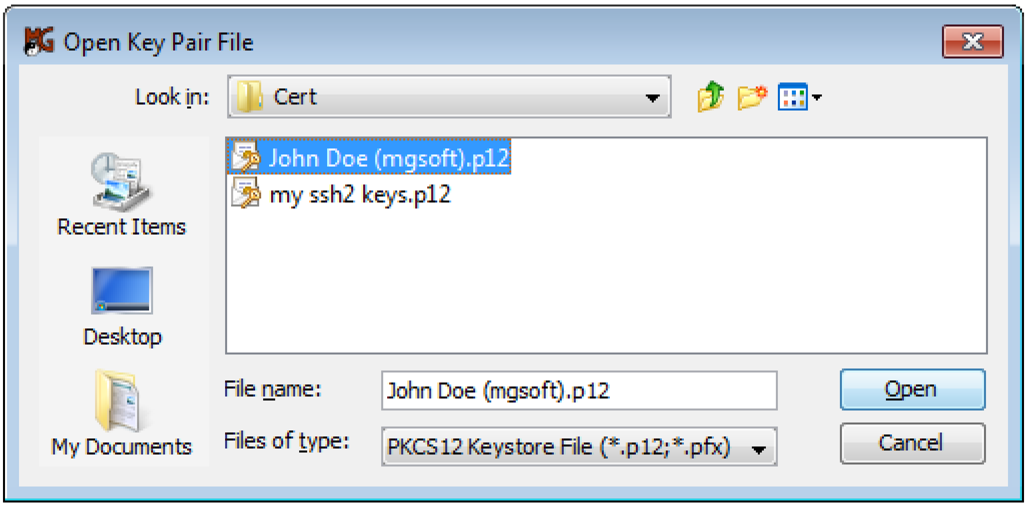
1. Open the Manage Certificates window, as described in the previous section.
2. Select the **Keystore** tab in the Manage Certificates window.
3. Select the ***Tools / Import Key Pair*** command from the menu in the Manage Certificates window (Figure 13).



<No entries available. Add/import a new keypair through **Tools** menu.>

Figure 13: Choosing the command to Import a digital certificate

4. The **Open Key Pair File** dialog box appears (Figure 14).



Look in: Cert

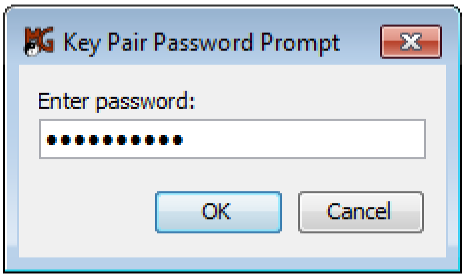
File name: John Doe (mgsoft).p12

Files of type: PKCS12 Keystore File (*.p12;*.pfx)

Figure 14: Importing a digital certificate (PKCS#12) containing public-private key pair

5. Navigate to the location containing the certificate in PKCS #12 file format (e.g., with the `*.p12` or `*.pfx` filename extension), select the file on disk and click the ***Open*** button.

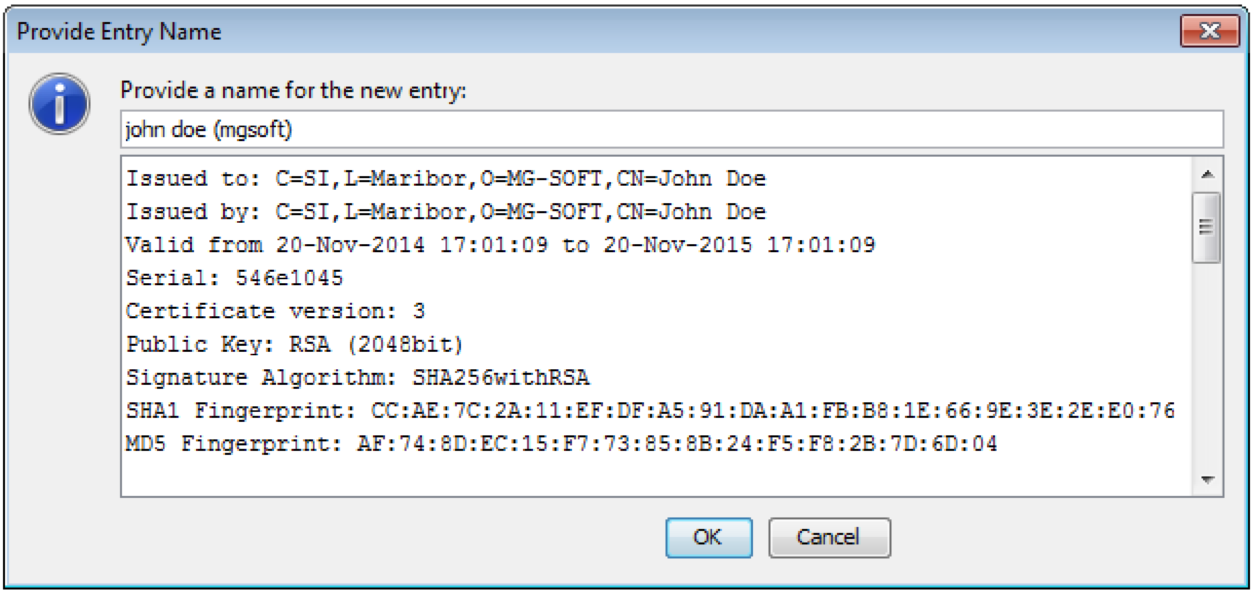
6. If the selected file has been protected with a password, the Enter Key Pair Password dialog box appears.



Enter password: ●●●●●●●●●●●

Figure 15: Entering a password for a digital certificate to be imported

7. Enter the password to decrypt the selected certificate file and click the ***OK*** button.
8. The Provide Entry Name dialog box appears, prompting you to enter a name by which the imported digital certificate will be referenced in the keystore (Figure 16).

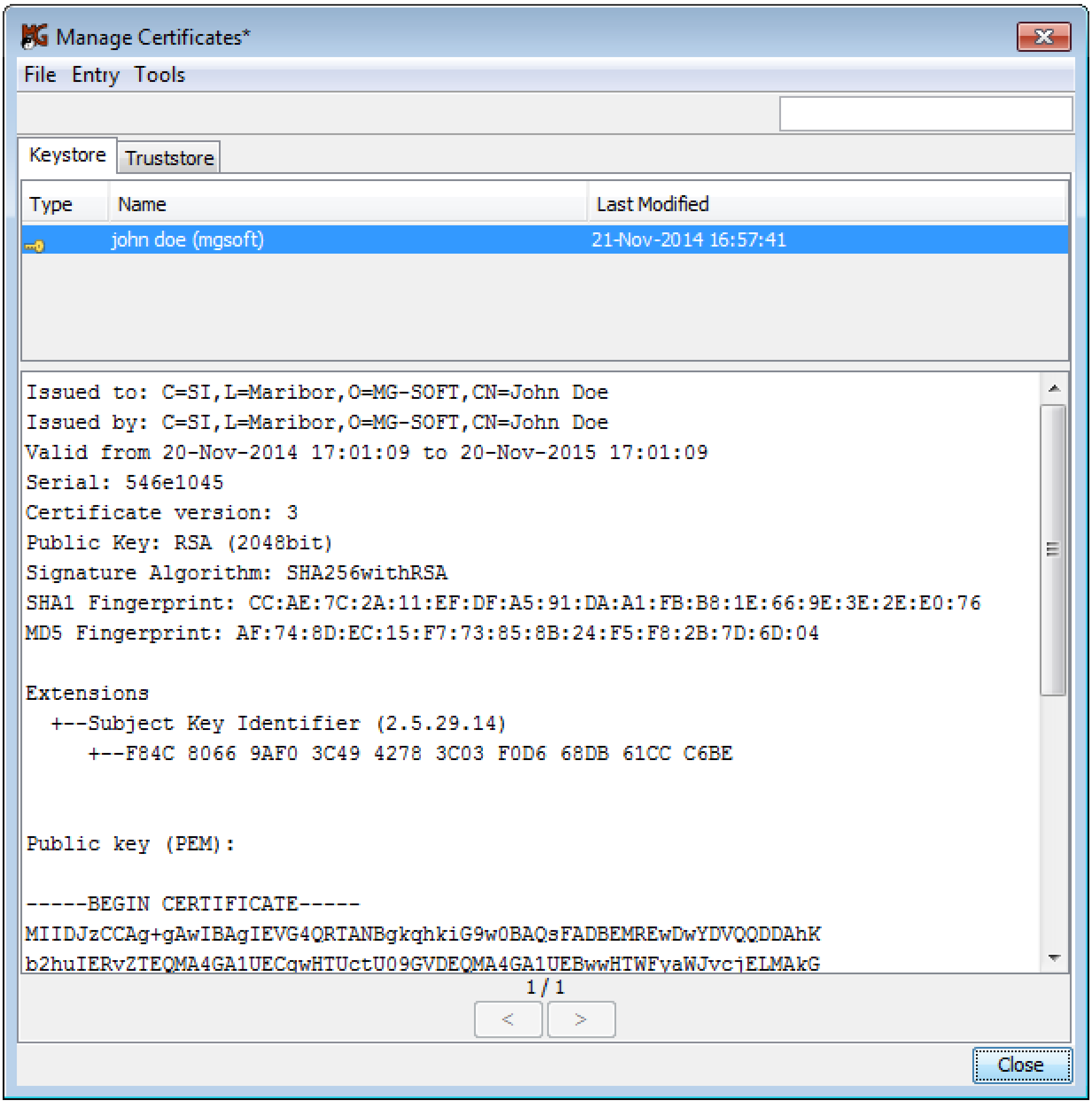


john doe (mgsoft)

```
Issued to: C=SI,L=Maribor,O=MG-SOFT,CN=John Doe
Issued by: C=SI,L=Maribor,O=MG-SOFT,CN=John Doe
Valid from 20-Nov-2014 17:01:09 to 20-Nov-2015 17:01:09
Serial: 546e1045
Certificate version: 3
Public Key: RSA (2048bit)
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: CC:AE:7C:2A:11:EF:DF:A5:91:DA:A1:FB:B8:1E:66:9E:3E:2E:E0:76
MD5 Fingerprint: AF:74:8D:EC:15:F7:73:85:8B:24:F5:F8:2B:7D:6D:04
```

Figure 16: Entering a keystore entry name (alias) for the imported digital certificate

9. Enter the certificate keystore entry name and click the ***OK*** button to close the Provide Entry Name dialog box.
10. The certificate is imported into the keystore - it is displayed in the **Keystore** tab in the Manage Certificates window (Figure 17). Click the entry in the upper window panel, to see the certificate and public key details in the lower window panel. Note that only the public key is displayed (the private key is not shown for security reasons).
11. Select the ***File / Save*** command in the Manage Certificates window to save the changes to the keystore you have made.



Type	Name	Last Modified
⊸	john doe (mgsoft)	21-Nov-2014 16:57:41

```
Issued to: C=SI,L=Maribor,O=MG-SOFT,CN=John Doe
Issued by: C=SI,L=Maribor,O=MG-SOFT,CN=John Doe
Valid from 20-Nov-2014 17:01:09 to 20-Nov-2015 17:01:09
Serial: 546e1045
Certificate version: 3
Public Key: RSA (2048bit)
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: CC:AE:7C:2A:11:EF:DF:A5:91:DA:A1:FB:B8:1E:66:9E:3E:2E:E0:76
MD5 Fingerprint: AF:74:8D:EC:15:F7:73:85:8B:24:F5:F8:2B:7D:6D:04

Extensions
  +--Subject Key Identifier (2.5.29.14)
     +--F84C 8066 9AF0 3C49 4278 3C03 F0D6 68DB 61CC C6BE


Public key (PEM):

-----BEGIN CERTIFICATE-----
MIIDJzCCAg+gAwIBAgIEVG4QRTANBgkqhkiG9w0BAQsFADBEMREwDwYDVQQDDAhK
b2huIERvZTEQMA4GA1UECgwHTUctU09GVDEQMA4GA1UEBwwHTWFyaWJvcjELMAkG
```

1 / 1

< >

Figure 17: Viewing the details of an imported digital certificate

5.4.3 Generating a Key Pair for Use with NETCONF over TLS or SSH

NetConf Browser can generate a digital certificate containing a public/private key pair to be used by a user for NETCONF over SSH with public key authentication or NETCONF over TLS session. The generated certificate is self-issued and self-signed, but can be subsequently signed by other party, like a certificate authority (CA). This section describes how to generate a (self-signed) certificate with public-private key pair, export the public key to .crs file format for signing by a CA, and import a signed certificate back into the keystore (replacing the self-signed certificate).

Generating a Digital Certificate with a Public-Private Key Pair

1. Open the Manage Certificates window.
2. Select the **Keystore** tab in the Manage Certificates window.
3. Select the ***Tools / Generate Key Pair*** command from the menu in the Manage Certificates window (Figure 18).

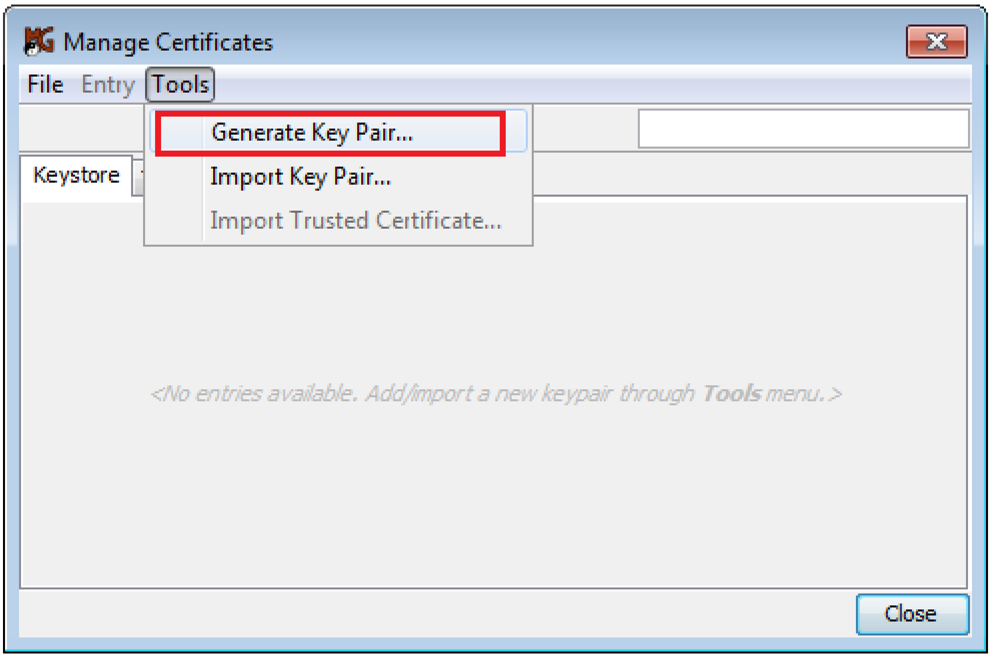
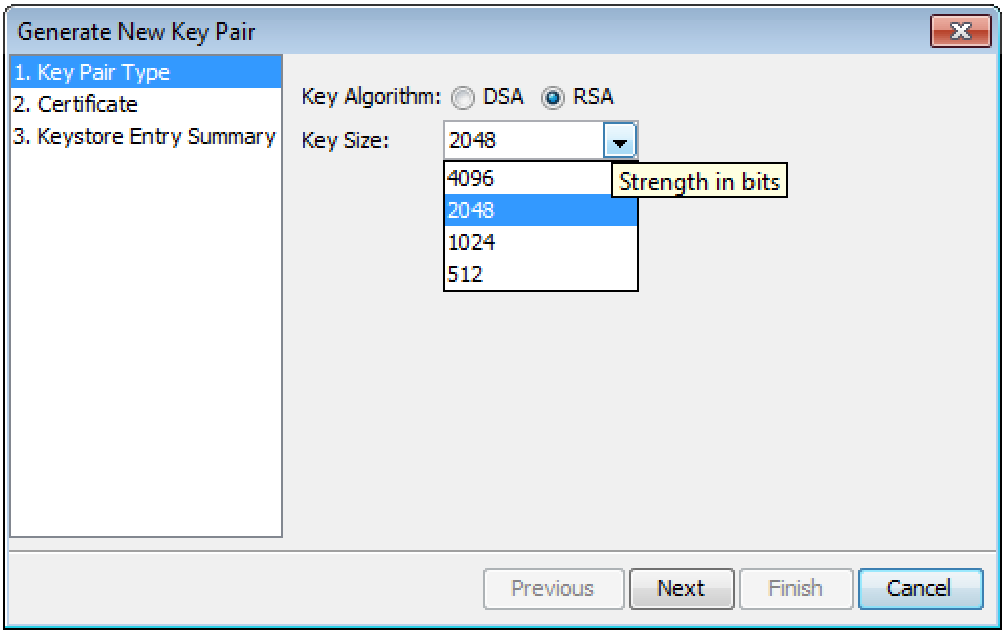

Manage Certificates
File
Entry
Tools
Generate Key Pair...
Import Key Pair...
Import Trusted Certificate...
Keystore
<No entries available. Add/import a new keypair through Tools menu.>
Close

Figure 18: Choosing the command to generate a digital certificate with public-private key pair

4. The Generate New Key Pair dialog box appears (Figure 19). This dialog box helps you specify information needed for generating a new key pair in three steps.

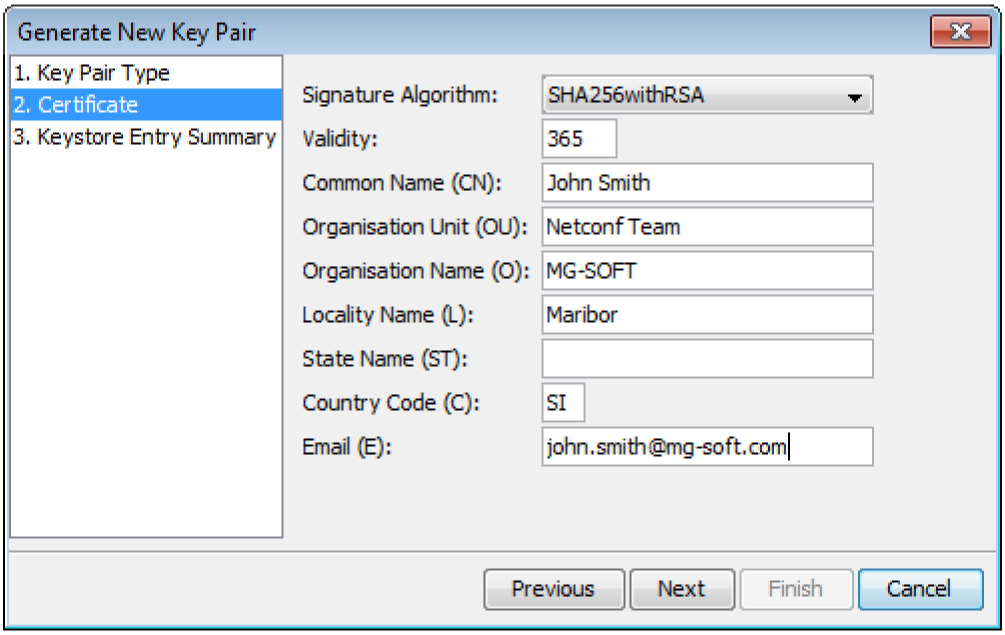


Key Algorithm: ○ DSA ● RSA

Key Size: 2048

Figure 19: The Generate New Key Pair dialog box, first screen

5. In the first screen of the Generate New Key Pair dialog box (Figure 19), select the desired ***Key algorithm*** (e.g., RSA) and ***Key size*** in bits (e.g., 2048). In general, the longer the key the more secure it is against brute-force attack, but large keys may adversely affect the application performance. Click the ***Next*** button to proceed to the second step.



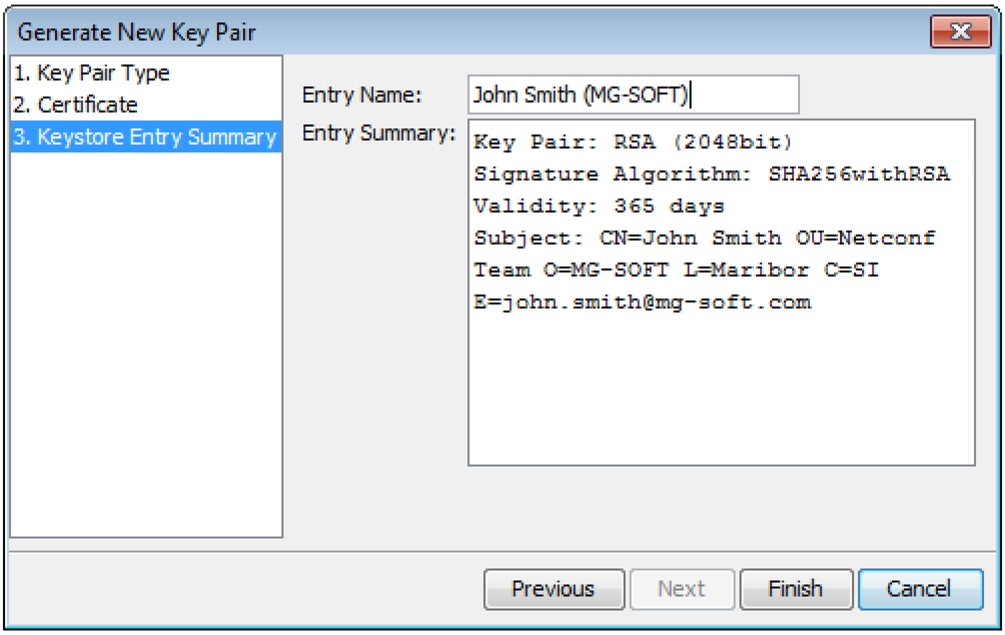
Signature Algorithm: SHA256withRSA
Validity: 365
Common Name (CN): John Smith
Organisation Unit (OU): Netconf Team
Organisation Name (O): MG-SOFT
Locality Name (L): Maribor
State Name (ST):
Country Code (C): SI
Email (E): john.smith@mg-soft.com

Figure 20: The Generate New Key Pair dialog box, second screen

6. In the second screen of the Generate New Key Pair dialog box (Figure 20), specify the digital certificate signature algorithm, validity and distinguished name details, as follows:
   - **Signature algorithm** – select the desired digital signature algorithm (e.g., SHA256withRSA),
   - **Validity** – enter the certificate validity in days (e.g., 365),
   - **Common Name (CN)** – optionally enter the name of the subject the certificate will be issued to (for example, your name),
   - **Organization Unit (OU)** – optionally enter the name of the organizational unit (e.g., department) the subject belongs to,
   - **Organization (O)** – optionally enter the name of the organization (e.g., a company) the subject belongs to,
   - **Locality (L)** – optionally enter the location (e.g., city) of the subject residence,
   - **State (ST)** – optionally enter the state (e.g., California) or province of the subject,
   - **Country Code (C)** – optionally enter the two letter ISO country code of the subject (e.g., US for USA, DE for Germany, etc.),
   - **E-mail (E)** – optionally enter the e-mail address of the subject.

Note: CN, OU, O, L, ST, C and E are the attributes of the so-called X.509 distinguished name for the certificate subject. At least one of these attributes must be specified.

7. After setting the above details, click the ***Next*** button to proceed to the final step.
8. In the third screen of the Generate New Key Pair dialog box (Figure 20), specify the keystore entry name (alias) for the new digital certificate, as follows:
   - **Entry name** – enter a name (alias) under which the digital certificate will be stored in the built-in keystore,
   - **Entry summary** – review the certificate details.



Entry Name: John Smith (MG-SOFT)

Entry Summary:

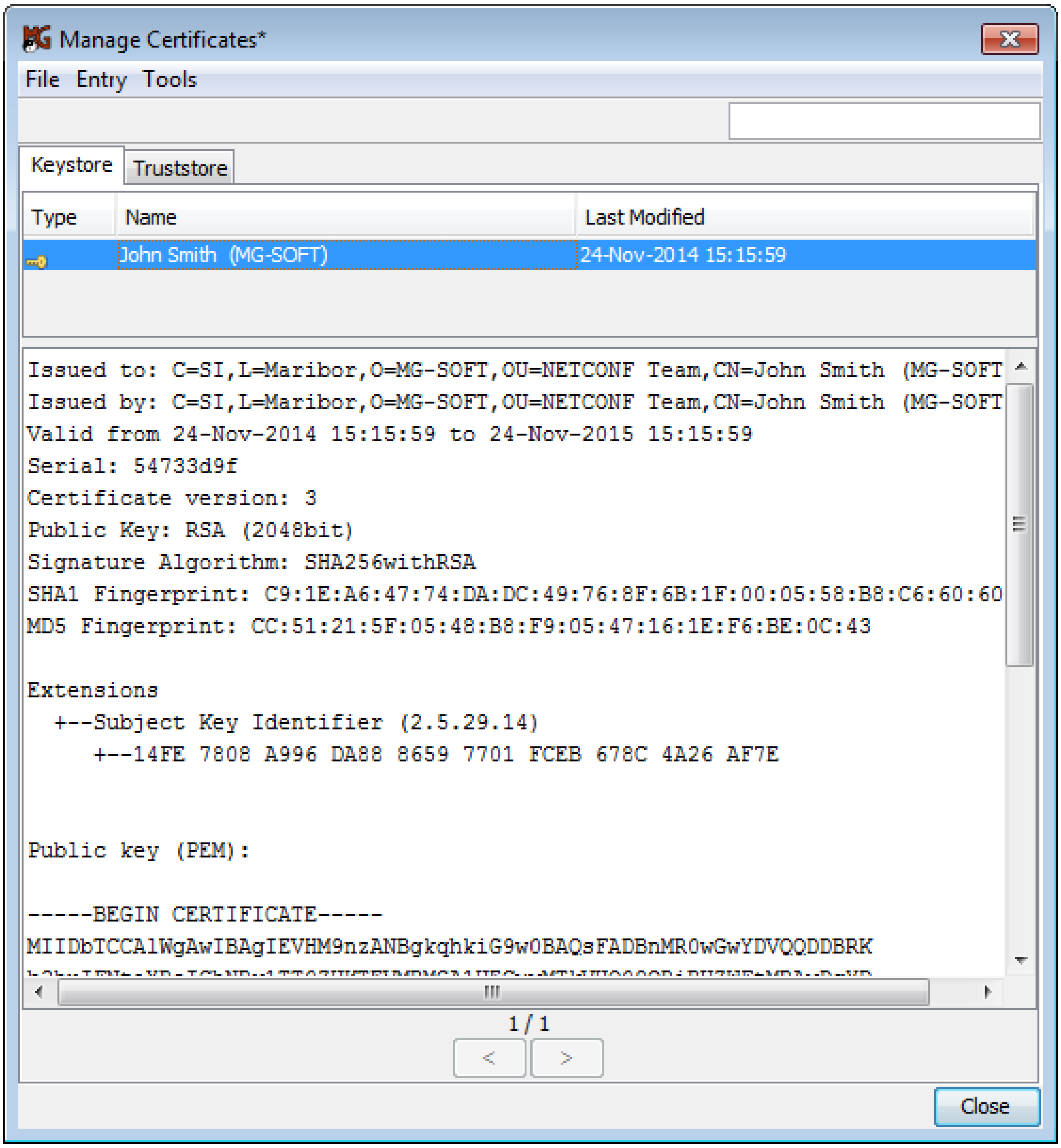
```
Key Pair: RSA (2048bit)
Signature Algorithm: SHA256withRSA
Validity: 365 days
Subject: CN=John Smith OU=Netconf
Team O=MG-SOFT L=Maribor C=SI
E=john.smith@mg-soft.com
```

Figure 21: The Generate New Key Pair dialog box, third screen

9. After you have specified the keystore entry name and reviewed the certificate details, click the ***Finish*** button to close the Generate New Key Pair dialog box generate a new self-issued and self-signed certificate with the corresponding public and private keys.
10. A new certificate entry appears in the **Keystore** tab in the Manage Certificates window (Figure 22). Click the entry in the upper window panel, to see the certificate and public key details in the lower window panel. Note that only the public key is displayed (the private key is not shown for security reasons).
11. Select the ***File / Save*** command in the Manage Certificates window to save the changes to the keystore you have made.

Tip: If you would like to use a self-issued and self-signed certificate (and the accompanying key pair) for NETCONF over TLS sessions with a specific NETCONF server, you need to export the certificate and the public key from the keystore to the .PEM format and import it to the trusted root store of the respective NETCONF server that supports TLS transport. The same is needed for NETCONF over SSH with public key authentication, unless the given NETCONF server utilizes the OpenSSH library. In such case, you can use the ***View OpenSSH authorized_keys Entry*** command to view and copy the public key in the format accepted by the OpenSSH library. Please refer to the NETCONF server documentation for more details on where to store the key.

Instead of using a self-issued and self-signed certificate, one would usually generate a certificate signing request (CSR) and submit it to a certificate authority (CA) to digitally sign the user certificate, as described in the next section. In such case, the NETCONF server only needs to have access to the certificate and the public key of this trusted CA (there is no need to copy the self-issued and self-signed certificate of each user to the NETCONF server).



Type	Name	Last Modified
	John Smith (MG-SOFT)	24-Nov-2014 15:15:59

```
Issued to: C=SI,L=Maribor,O=MG-SOFT,OU=NETCONF Team,CN=John Smith (MG-SOFT
Issued by: C=SI,L=Maribor,O=MG-SOFT,OU=NETCONF Team,CN=John Smith (MG-SOFT
Valid from 24-Nov-2014 15:15:59 to 24-Nov-2015 15:15:59
Serial: 54733d9f
Certificate version: 3
Public Key: RSA (2048bit)
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: C9:1E:A6:47:74:DA:DC:49:76:8F:6B:1F:00:05:58:B8:C6:60:60
MD5 Fingerprint: CC:51:21:5F:05:48:B8:F9:05:47:16:1E:F6:BE:0C:43

Extensions
  +--Subject Key Identifier (2.5.29.14)
     +--14FE 7808 A996 DA88 8659 7701 FCEB 678C 4A26 AF7E


Public key (PEM):

-----BEGIN CERTIFICATE-----
MIIDbTCCAlWgAwIBAgIEVHM9nzANBgkqhkiG9w0BAQsFADBnMR0wGwYDVQQDDBRK
```

Figure 22: Viewing the details of a generated digital certificate/key pair (self-signed)

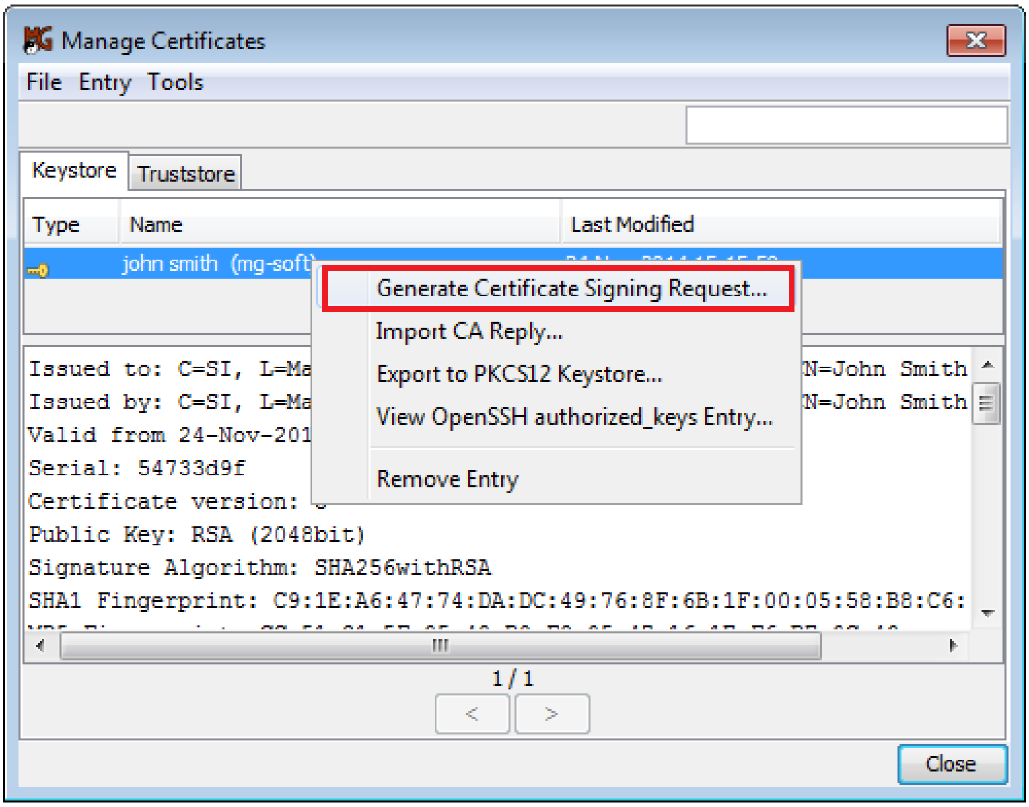
Generating a Certificate Signing Request (CSR)

Previous section describes how to generate a self-issued and self-signed X.509 certificate and a public-private key pair.

Typically, one would then generate a certificate signing request (CSR) and submit it to a public certificate authority (CA), such as, e.g., VeriSign or Thawte, or your organization’s own certificate authority in order to digitally sign the certificate, as described in this section.

1. Open the Manage Certificates window.
2. Select the **Keystore** tab in the Manage Certificates window.

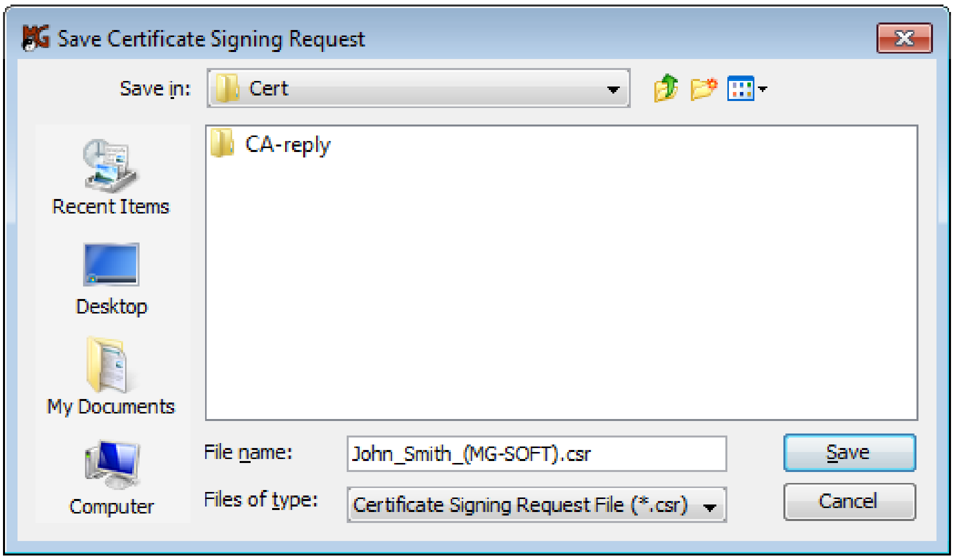
3. In the in the upper window panel in the **Keystore** tab, select the certificate you have generated and choose the ***Entry / Generate Certificate Signing Request*** command from the menu in the Manage Certificates window. Alternatively, right-click the certificate and choose the ***Generate Certificate Signing Request*** command from the context menu (Figure 23).



```
Issued to: C=SI, L=Ma                          N=John Smith
Issued by: C=SI, L=Ma                          N=John Smith
Valid from 24-Nov-201
Serial: 54733d9f
Certificate version: 
Public Key: RSA (2048bit)
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: C9:1E:A6:47:74:DA:DC:49:76:8F:6B:1F:00:05:58:B8:C6:
```

Figure 23: Choosing the command to generate a certificate signing request (CSR)

4. The ***Save Certificate Signing Request*** dialog box appears (Figure 24). Specify the location and name of the certificate signing request file and click the ***Save*** button to create the certificate signing request file (.csr) that includes your public key.



File name: John_Smith_(MG-SOFT).csr

Files of type: Certificate Signing Request File (*.csr)

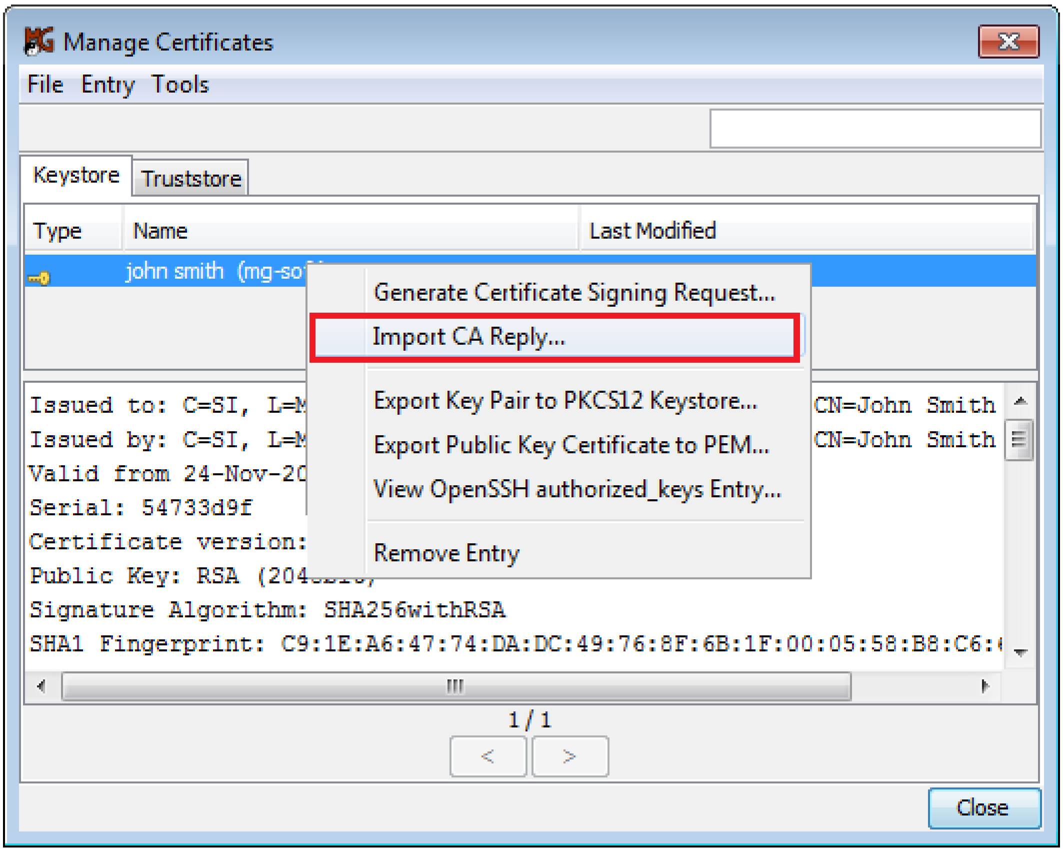
Figure 24: Saving a certificate signing request file

5. Submit the generated .CSR file to a trusted public certificate authority (CA), such as, e.g., VeriSign or Thawte, or to your organization's own certificate authority in order to digitally sign the certificate. You should import the signed certificate back to the keystore, as described in the next section.

Importing a Signed Certificate (CA-Reply) into Keystore

This section describes how to import a digital certificate that has been signed by a CA back into the keystore, replacing the self-signed certificate. CA reply certificate can be stored in various file formats, like .pem, .cer, .cert, .crt, .p7b, .spc, .pkipath. CA reply certificate file may contain a single certificate or a chain of certificates. In the latter case, the entire chain can be imported.

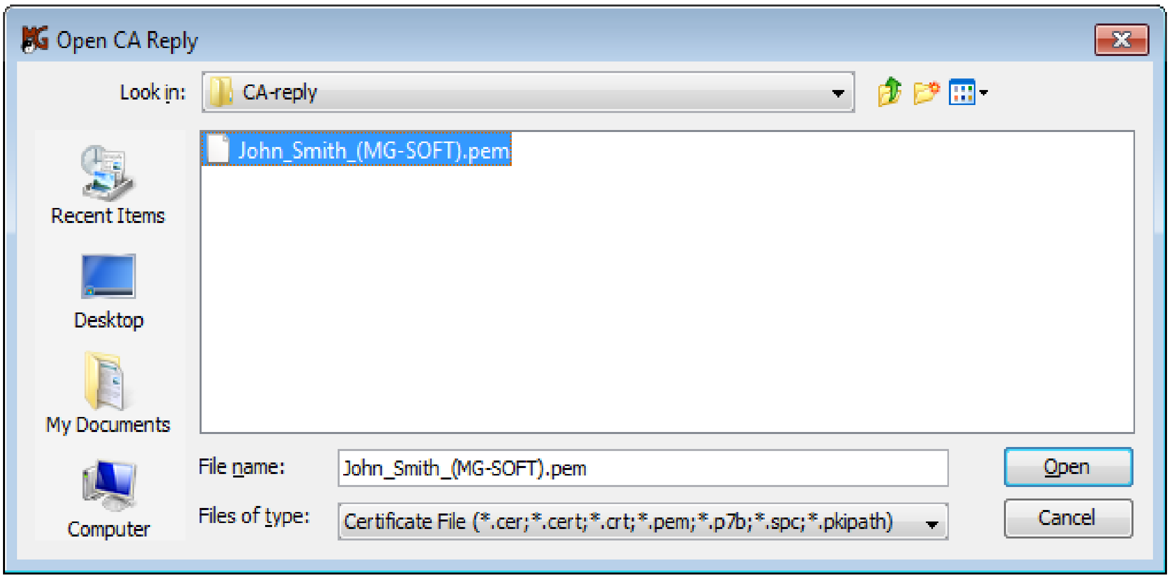
1. Open the Manage Certificates window.
2. Select the **Keystore** tab in the Manage Certificates window.
3. In the in the upper window panel in the **Keystore** tab, select the respective certificate entry and choose the ***Entry / Import CA Reply*** command from the menu in the Manage Certificates window. Alternatively, right-click the certificate and choose the ***Import CA Reply*** command from the context menu (Figure 25).


Manage Certificates
File Entry Tools
Keystore
Truststore
Type
Name
Last Modified
john smith (mg-so
Generate Certificate Signing Request...
Import CA Reply...
Export Key Pair to PKCS12 Keystore...
Export Public Key Certificate to PEM...
View OpenSSH authorized_keys Entry...
Remove Entry
1 / 1
<
>
Close

```
Issued to: C=SI, L=M                          CN=John Smith
Issued by: C=SI, L=M                          CN=John Smith
Valid from 24-Nov-20
Serial: 54733d9f
Certificate version:
Public Key: RSA (204
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: C9:1E:A6:47:74:DA:DC:49:76:8F:6B:1F:00:05:58:B8:C6:
```

Figure 25: Choosing the command to import a CA signed certificate (or certificate chain)

4. The **Open CA Reply** dialog box appears (Figure 26). Navigate to the location containing the CA reply certificate file in supported format (.pem, .cer, .cert, .crt, .p7b, .spc, .pkipath), select the file from disk and click the ***Open*** button.



Look in: CA-reply

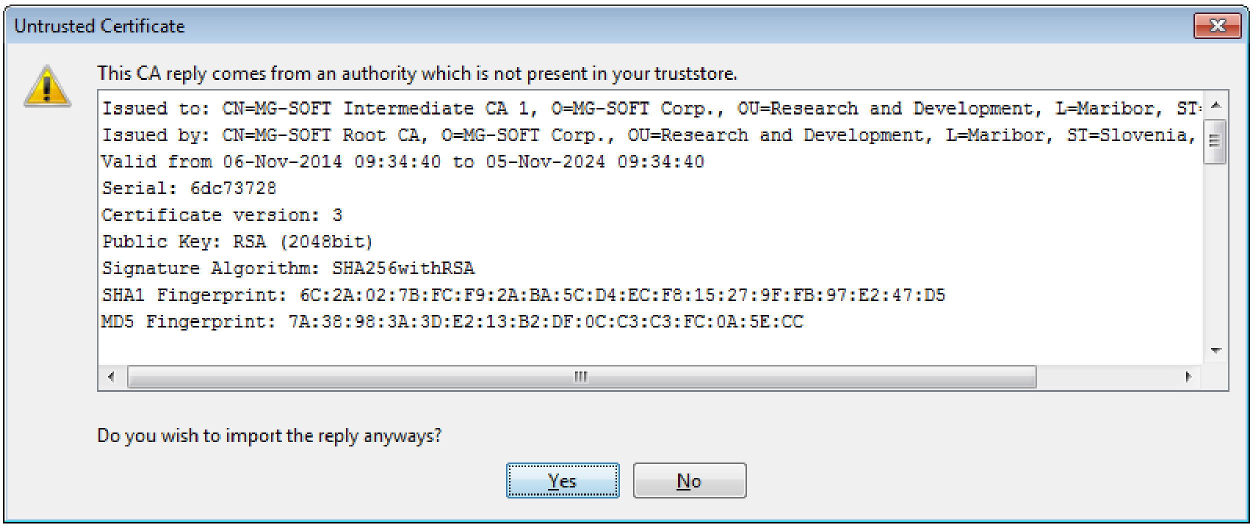
John_Smith_(MG-SOFT).pem

File name: John_Smith_(MG-SOFT).pem

Files of type: Certificate File (*.cer;*.cert;*.crt;*.pem;*.p7b;*.spc;*.pkipath)

Figure 26: Importing a CA signed certificate (or certificate chain)

5. If the CA reply contains a certificate that is not in your truststore, the **Untrusted Certificate** dialog box appears, presenting the details of the 'untrusted' root CA certificate (Figure 27). Carefully examine the digital certificate information displayed in the Untrusted Certificate dialog box and proceed as follows:
   - Click the **Yes** button if you wish to trust the certificate (chain) and import the CA reply.
   - Click the **No** button to reject the certificate and abort the import.



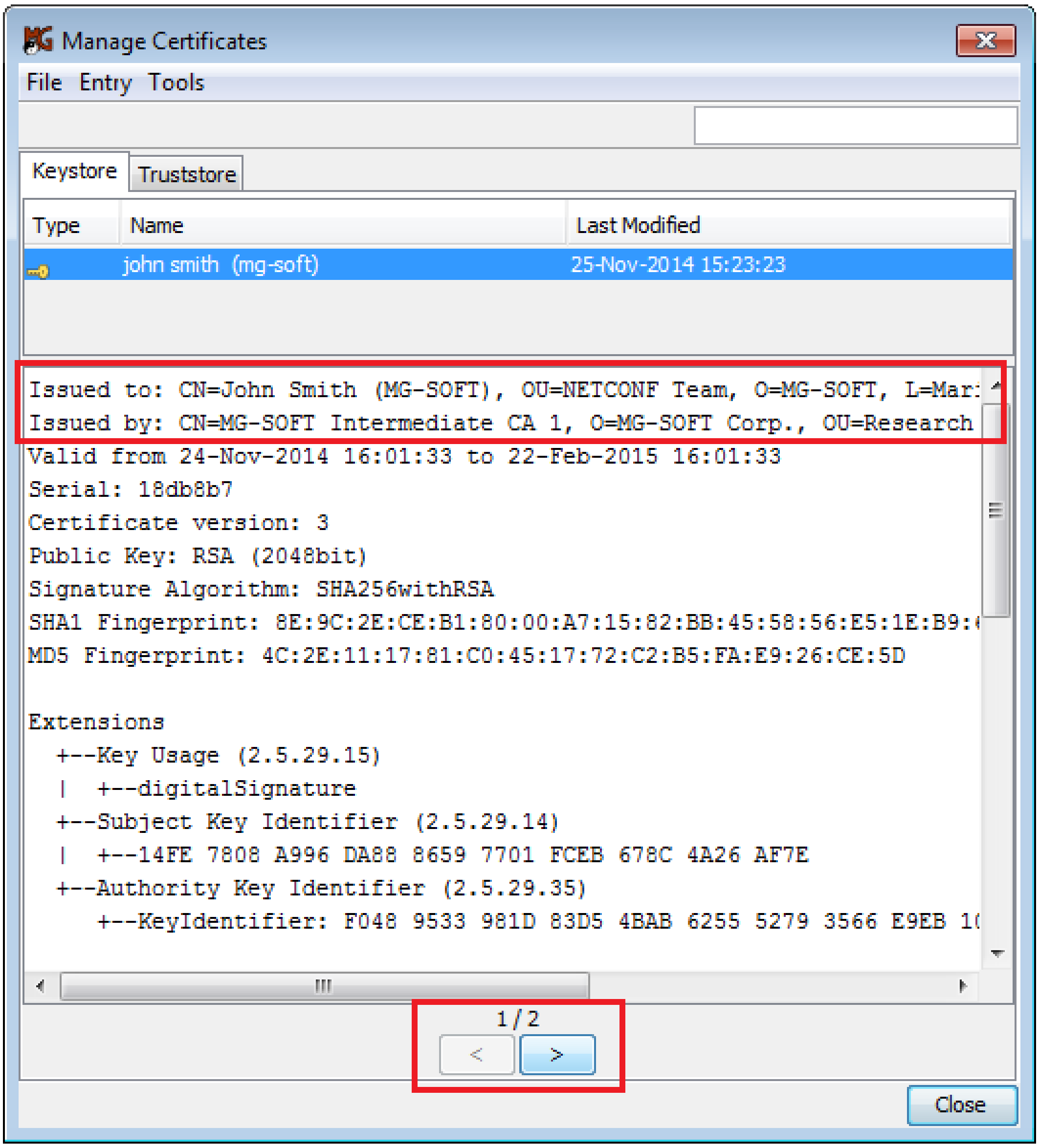
This CA reply comes from an authority which is not present in your truststore.

```
Issued to: CN=MG-SOFT Intermediate CA 1, O=MG-SOFT Corp., OU=Research and Development, L=Maribor, ST=
Issued by: CN=MG-SOFT Root CA, O=MG-SOFT Corp., OU=Research and Development, L=Maribor, ST=Slovenia,
Valid from 06-Nov-2014 09:34:40 to 05-Nov-2024 09:34:40
Serial: 6dc73728
Certificate version: 3
Public Key: RSA (2048bit)
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: 6C:2A:02:7B:FC:F9:2A:BA:5C:D4:EC:F8:15:27:9F:FB:97:E2:47:D5
MD5 Fingerprint: 7A:38:98:3A:3D:E2:13:B2:DF:0C:C3:C3:FC:0A:5E:CC
```

Do you wish to import the reply anyways?

Figure 27: Examining the details of an 'untrusted' CA reply certificate

6. By importing the CA reply, the keystore entry will be updated to reflect the content of the CA reply (i.e., user's self-signed certificate is replaced with the CA-signed one – see Figure 28).

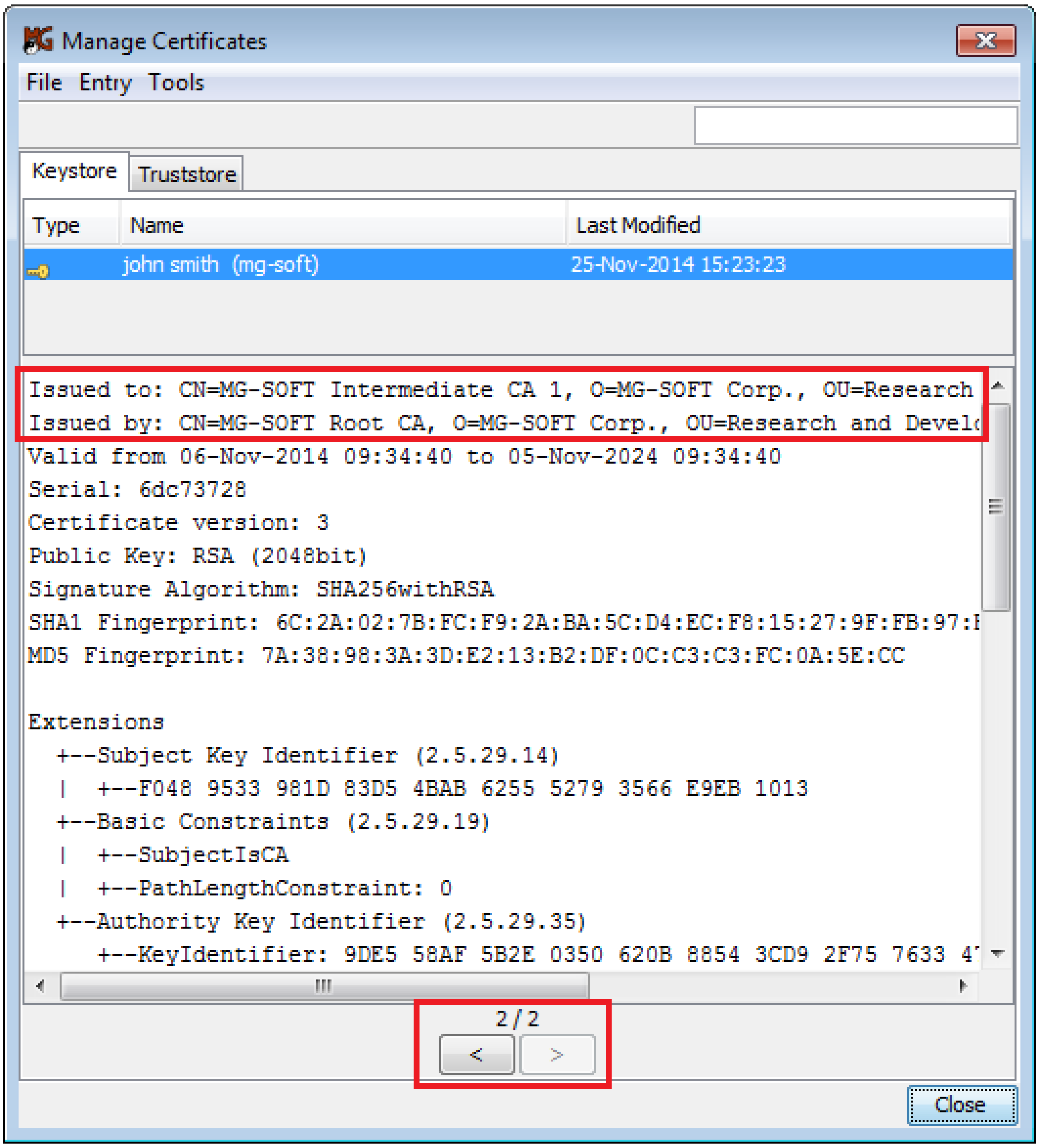


```
Issued to: CN=John Smith (MG-SOFT), OU=NETCONF Team, O=MG-SOFT, L=Mar:
Issued by: CN=MG-SOFT Intermediate CA 1, O=MG-SOFT Corp., OU=Research
Valid from 24-Nov-2014 16:01:33 to 22-Feb-2015 16:01:33
Serial: 18db8b7
Certificate version: 3
Public Key: RSA (2048bit)
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: 8E:9C:2E:CE:B1:80:00:A7:15:82:BB:45:58:56:E5:1E:B9:
MD5 Fingerprint: 4C:2E:11:17:81:C0:45:17:72:C2:B5:FA:E9:26:CE:5D

Extensions
  +--Key Usage (2.5.29.15)
  |   +--digitalSignature
  +--Subject Key Identifier (2.5.29.14)
  |   +--14FE 7808 A996 DA88 8659 7701 FCEB 678C 4A26 AF7E
  +--Authority Key Identifier (2.5.29.35)
      +--KeyIdentifier: F048 9533 981D 83D5 4BAB 6255 5279 3566 E9EB 1
```

Figure 28: Viewing the details of a CA reply certificate chain (first certificate)

7. If a CA reply contained a chain of certificates, you can view all certificates in the chain by clicking the **>** button at the bottom of the Manage Certificates widow. In our example, the CA reply file contained a chain of two certificates, one is the user's certificate signed by the company's Intermediate CA and the other is the Intermediate CA certificate signed by the company's root CA (Figure 29).



```
Issued to: CN=MG-SOFT Intermediate CA 1, O=MG-SOFT Corp., OU=Research
Issued by: CN=MG-SOFT Root CA, O=MG-SOFT Corp., OU=Research and Develo
Valid from 06-Nov-2014 09:34:40 to 05-Nov-2024 09:34:40
Serial: 6dc73728
Certificate version: 3
Public Key: RSA (2048bit)
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: 6C:2A:02:7B:FC:F9:2A:BA:5C:D4:EC:F8:15:27:9F:FB:97:I
MD5 Fingerprint: 7A:38:98:3A:3D:E2:13:B2:DF:0C:C3:C3:FC:0A:5E:CC

Extensions
  +--Subject Key Identifier (2.5.29.14)
  |  +--F048 9533 981D 83D5 4BAB 6255 5279 3566 E9EB 1013
  +--Basic Constraints (2.5.29.19)
  |  +--SubjectIsCA
  |  +--PathLengthConstraint: 0
  +--Authority Key Identifier (2.5.29.35)
     +--KeyIdentifier: 9DE5 58AF 5B2E 0350 620B 8854 3CD9 2F75 7633 4
```

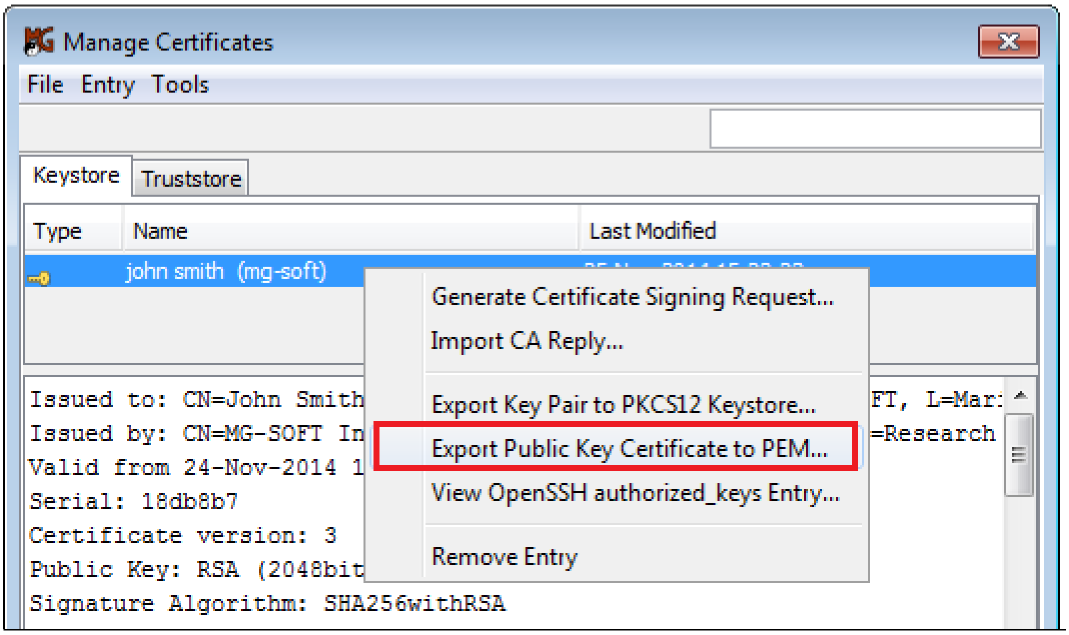
Figure 29: Viewing the details of a CA reply certificate chain (second certificate)

Exporting a Public Key Certificate to PEM Format

This section describes how to export a digital certificate and the accompanying public key to a .PEM file format. This certificate and public key then needs to be imported from the PEM file into the NETCONF server that you wish to manage by using NETCONF over TLS or NETCONF over SSH with public key authentication. For details on how to import a public key from a .PEM file and add it to trusted certificates/keys, please refer to the documentation of the respective NETCONF server.

1. Open the Manage Certificates window.
2. Select the **Keystore** tab in the Manage Certificates window.
3. In the in the upper window panel in the **Keystore** tab, select the certificate entry that you would like to export and choose the ***Entry / Export Public Key Certificate to***

PEM command from the menu in the Manage Certificates window. Alternatively, right-click the certificate and choose the ***Export Public Key Certificate to PEM*** command from the context menu (Figure 30).


Manage Certificates
File
Entry
Tools
Keystore
Truststore
Type
Name
Last Modified
john smith (mg-soft)
Generate Certificate Signing Request...
Import CA Reply...
Export Key Pair to PKCS12 Keystore...
Export Public Key Certificate to PEM...
View OpenSSH authorized_keys Entry...
Remove Entry

```
Issued to: CN=John Smith                          FT, L=Mar:
Issued by: CN=MG-SOFT In                          =Research
Valid from 24-Nov-2014 1
Serial: 18db8b7
Certificate version: 3
Public Key: RSA (2048bit
Signature Algorithm: SHA256withRSA
```

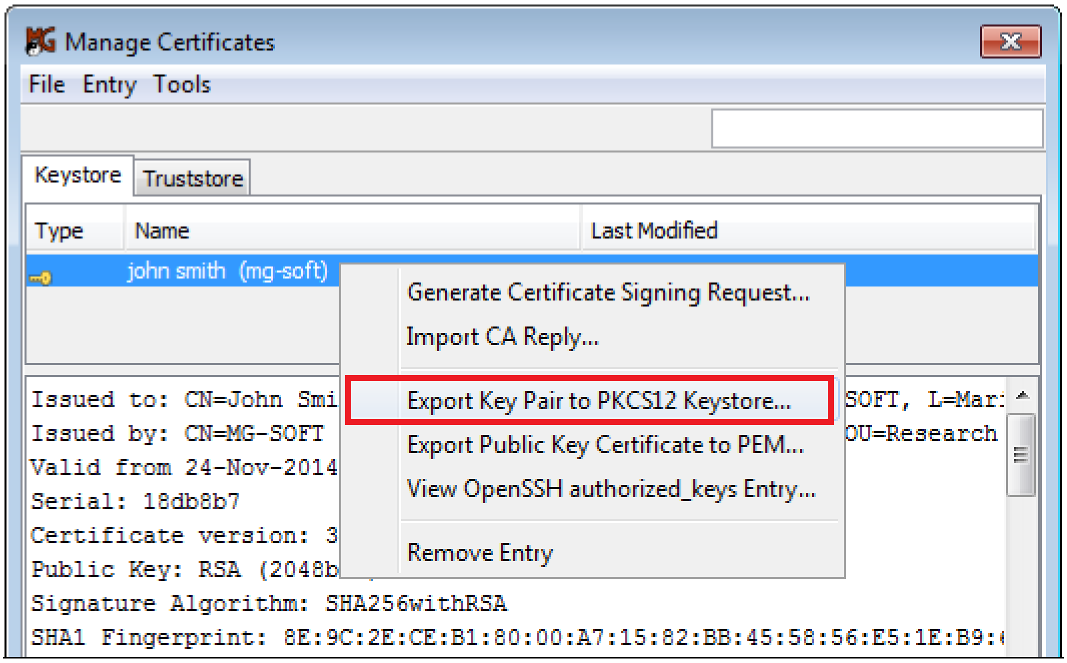
Figure 30: Selecting the command to export a certificate and public key to PEM format

4. The **Save *Public Key Certificate to PEM*** dialog box appears, resembling the standard Save As dialog box. Specify the location and name for the PEM file and click the ***Save*** button to save the file (.pem) that includes your digital certificate and public key.

5.4.4 Exporting a Key Pair to PKCS#12 Format

This section describes how to export a digital certificate and the accompanying public and private key to a PKCS#12 file format for external use or backup purposes.

1. Open the Manage Certificates window.
2. Select the **Keystore** tab in the Manage Certificates window.
3. In the in the upper window panel in the **Keystore** tab, select the certificate entry that you would like to export and choose the ***Entry / Export Key Pair to PKCS12 Keystore*** command from the menu. Alternatively, right-click the certificate and choose the ***Export Key Pair to PKCS12 Keystore*** command from the context menu (Figure 31).

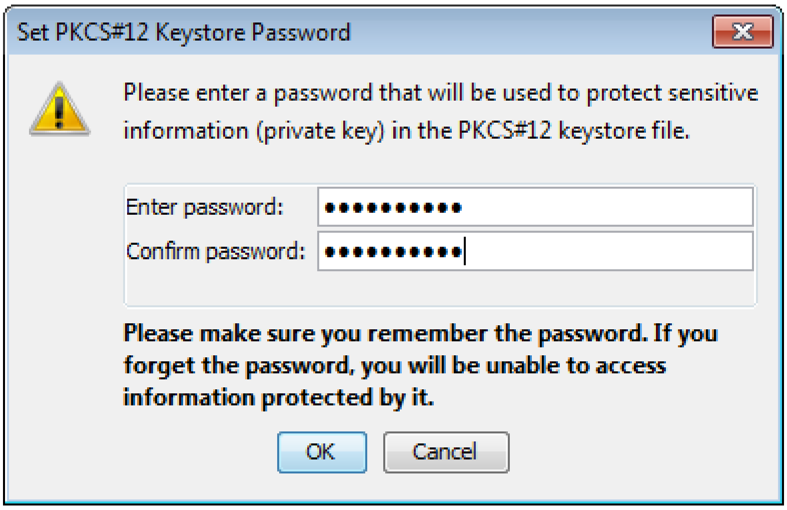

Manage Certificates
File Entry Tools
Keystore
Truststore
Type
Name
Last Modified
john smith (mg-soft)
Generate Certificate Signing Request...
Import CA Reply...
Export Key Pair to PKCS12 Keystore...
Export Public Key Certificate to PEM...
View OpenSSH authorized_keys Entry...
Remove Entry

```
Issued to: CN=John Smi                                   SOFT, L=Mar:
Issued by: CN=MG-SOFT                                     OU=Research
Valid from 24-Nov-2014
Serial: 18db8b7
Certificate version: 3
Public Key: RSA (2048b
Signature Algorithm: SHA256withRSA
SHA1 Fingerprint: 8E:9C:2E:CE:B1:80:00:A7:15:82:BB:45:58:56:E5:1E:B9:
```

Figure 31: Selecting the command to export a certificate (chain) and key pair to PKCS#12 format

4. The Set PKCS#12 Keystore Password dialog box appears (Figure 32), prompting you to enter a password that will be used to protect (encrypt) the private key in the PKCS#12 keystore file. After entering the password in both input lines, click the ***OK*** button.

> **Note:** Please make sure you remember the password; otherwise you will not be able to decrypt and retrieve the private key from the PKCS#12 keystore file.


Set PKCS#12 Keystore Password

Please enter a password that will be used to protect sensitive information (private key) in the PKCS#12 keystore file.

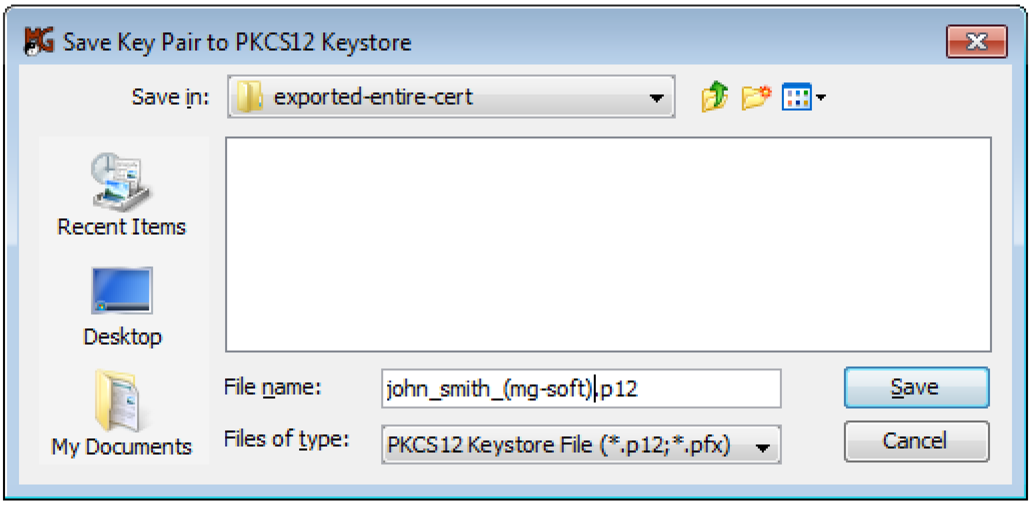
Enter password: ••••••••••

Confirm password: ••••••••••

Please make sure you remember the password. If you forget the password, you will be unable to access information protected by it.

Figure 32: Specifying a password to protect the private key in the PKCS#12 file

5. The Save Key Pair to PKCS#12 dialog box appears (Figure 33), resembling the standard Save As dialog box. Specify the location and name for the PKCS file and click the ***Save*** button to save the PKCS#12 (`.p12`) file that includes the selected digital certificate (or certificate chain) and the accompanying public and private keys.



Save in: exported-entire-cert

File name: john_smith_(mg-soft).p12

Files of type: PKCS12 Keystore File (*.p12;*.pfx)

Figure 33: Saving a certificate and public-private keys in PKCS#12 file format

Note: Please make sure to keep the PKCS#12 file in a safe location, as it contains also you secret private key.

6 NAVIGATING YANG TREE AND SELECTING NODES

When NetConf Browser is started for the first time, it automatically loads all standard YANG modules that are included in the distribution and graphically displays loaded modules in the YANG Tree panel in the left portion of the main window. Additional, vendor-specific YANG or YIN modules can be loaded by the user.

YANG is a data modeling language for the Network Configuration Protocol (NETCONF).

YANG module defines a hierarchy of data that can be used for NETCONF-based operations, including configuration, state data, Remote Procedure Calls, and notifications. Typically, a YANG module defines a tree of data elements that represent the configuration and runtime status of a particular network element managed via NETCONF. A YANG module is normally stored in a file with the .yang extension.

YANG modules can be translated into an equivalent XML syntax called **YIN** (YANG Independent Notation), allowing applications using XML parsers to operate on the models. The conversion from YANG to YIN is lossless. Typically, a YIN module is stored in a file with the .yin extension.

Submodules are partial modules that contribute definitions to a module. A module may include zero or more submodules, but each submodule may belong to only one module.

Loaded YIN and YANG modules are hierarchically organized and represented in the tree structure, containing nodes of different types. You can expand and view the tree structure in the YANG Tree panel (Figure 35) in the main window, as well as view the YANG properties of any selected node, as described in this section.

6.1 Expanding YANG Tree and Selecting Nodes

1. In the YANG Tree panel, select the root node (📁) if you want to expand the hierarchical tree structure of all the loaded modules, or a module (☯), or submodule (☯) node to expand and display the hierarchical tree structure of that (sub)module only.
2. Right-click the selected node to display the mouse context (pop-up) menu and select the ***Expand Entire Subtree*** pop-up command (Figure 34).

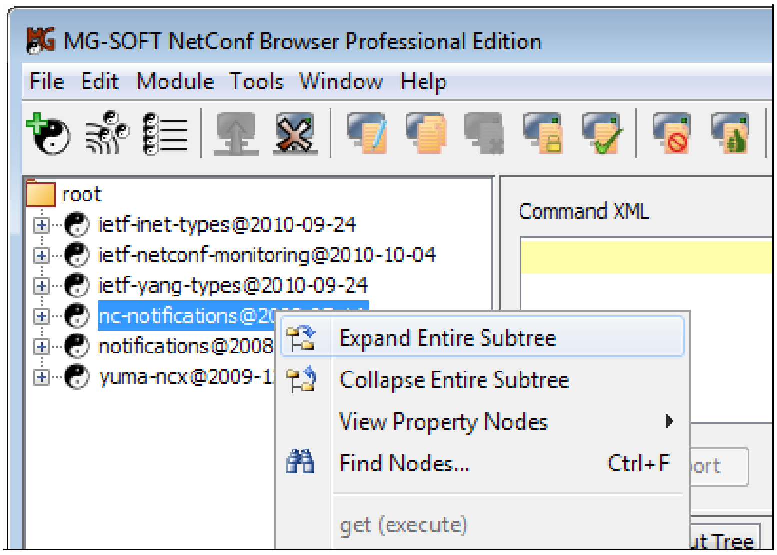


Figure 34: Selecting the Expand command from the context menu in the YANG Tree panel

3. In the expanded subtree, right-click the node to see what commands can be performed on it (e.g., on state data nodes, the **get-config and edit-config** operations are disabled, etc.).

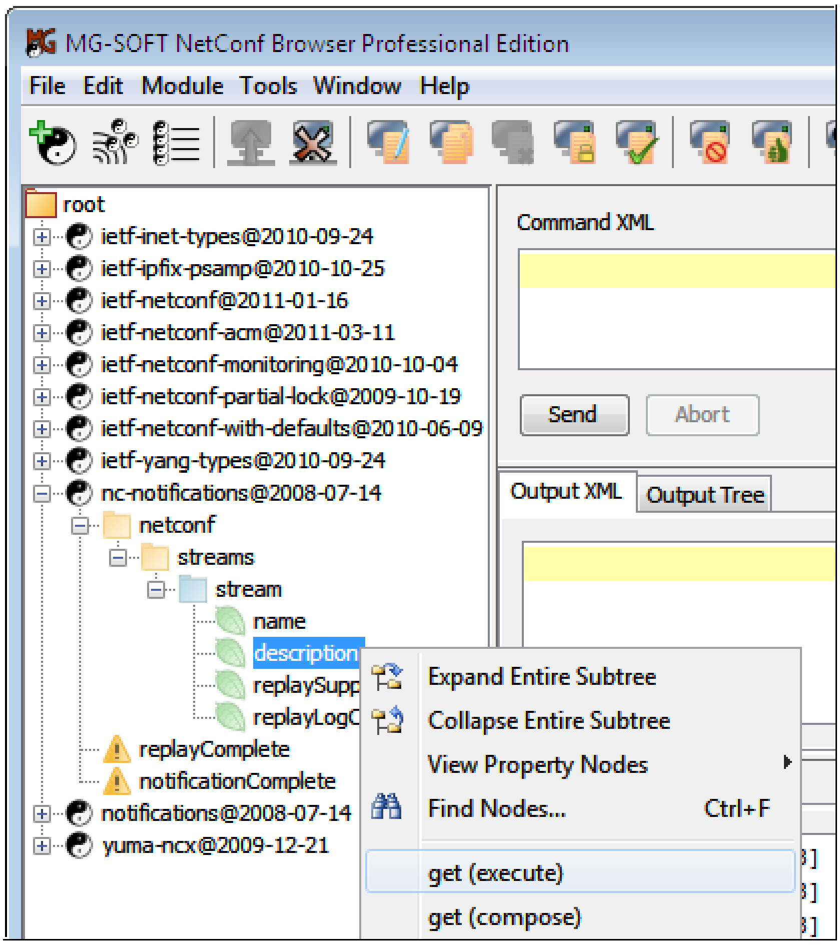

MG-SOFT NetConf Browser Professional Edition
File Edit Module Tools Window Help
root
ietf-inet-types@2010-09-24
ietf-ipfix-psamp@2010-10-25
ietf-netconf@2011-01-16
ietf-netconf-acm@2011-03-11
ietf-netconf-monitoring@2010-10-04
ietf-netconf-partial-lock@2009-10-19
ietf-netconf-with-defaults@2010-06-09
ietf-yang-types@2010-09-24
nc-notifications@2008-07-14
netconf
streams
stream
name
description
replaySupp
replayLogC
replayComplete
notificationComplete
notifications@2008-07-14
yuma-ncx@2009-12-21
Command XML
Send
Abort
Output XML
Output Tree
Expand Entire Subtree
Collapse Entire Subtree
View Property Nodes
Find Nodes... Ctrl+F
get (execute)
get (compose)

Figure 35: Context menu opened on the selected state data node in the YANG Tree panel

6.2 Viewing Property Sub-Nodes

The properties of any node can be displayed as sub-nodes in the YANG Tree window panel. These property sub-nodes are not displayed by default.

1. Right-click a node and select the ***View Property Nodes / Show for Subtree*** pop-up command (Figure 36) to display the property sub-nodes for all nodes in the selected subtree.

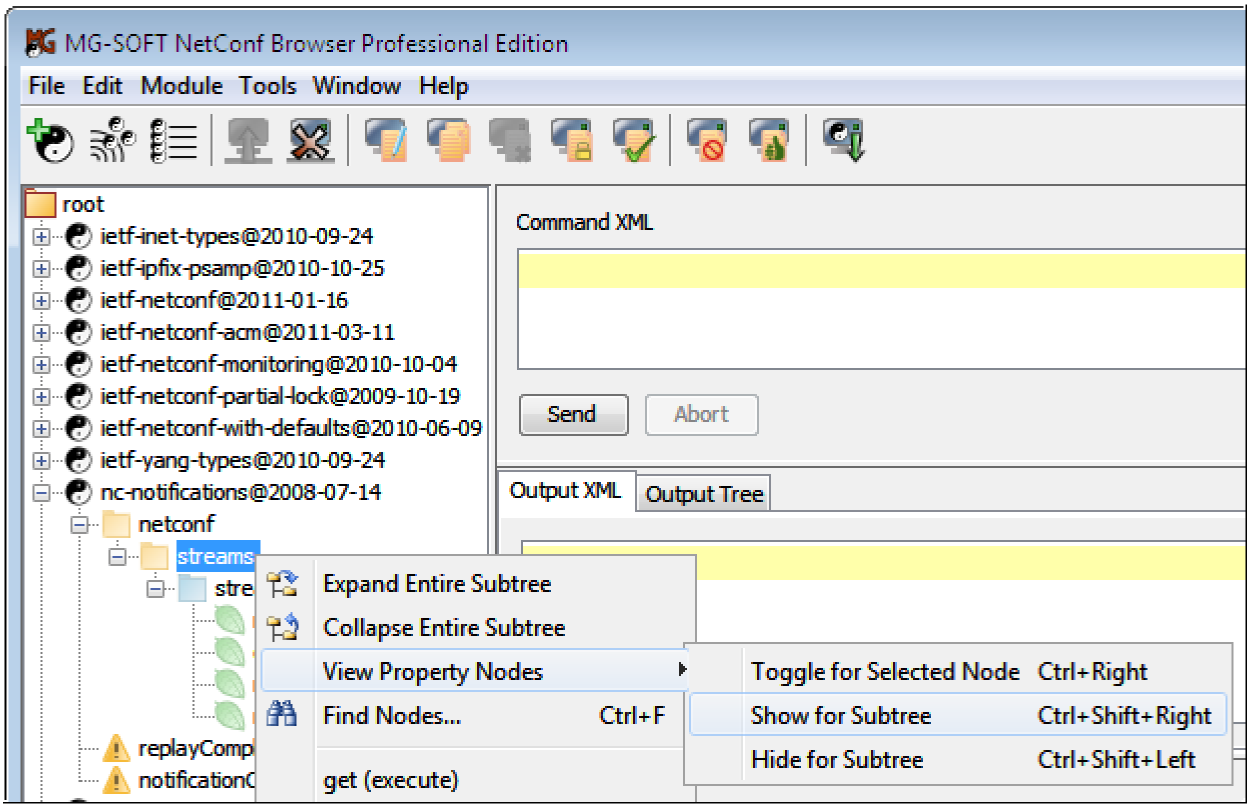

MG-SOFT NetConf Browser Professional Edition
File Edit Module Tools Window Help
root
ietf-inet-types@2010-09-24
ietf-ipfix-psamp@2010-10-25
ietf-netconf@2011-01-16
ietf-netconf-acm@2011-03-11
ietf-netconf-monitoring@2010-10-04
ietf-netconf-partial-lock@2009-10-19
ietf-netconf-with-defaults@2010-06-09
ietf-yang-types@2010-09-24
nc-notifications@2008-07-14
netconf
streams
stre
replayComp
notificationC
Command XML
Send
Abort
Output XML
Output Tree
Expand Entire Subtree
Collapse Entire Subtree
View Property Nodes
Find Nodes... Ctrl+F
get (execute)
Toggle for Selected Node Ctrl+Right
Show for Subtree Ctrl+Shift+Right
Hide for Subtree Ctrl+Shift+Left

Figure 36: Selecting the View Property Nodes for Subtree option in the YANG Tree panel

2. The property sub-nodes (●) are displayed for all nodes in the selected subtree (Figure 37).

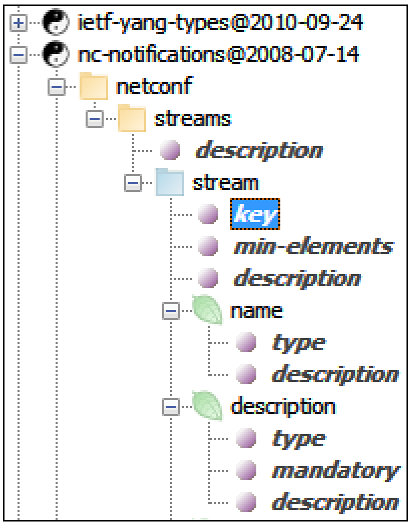


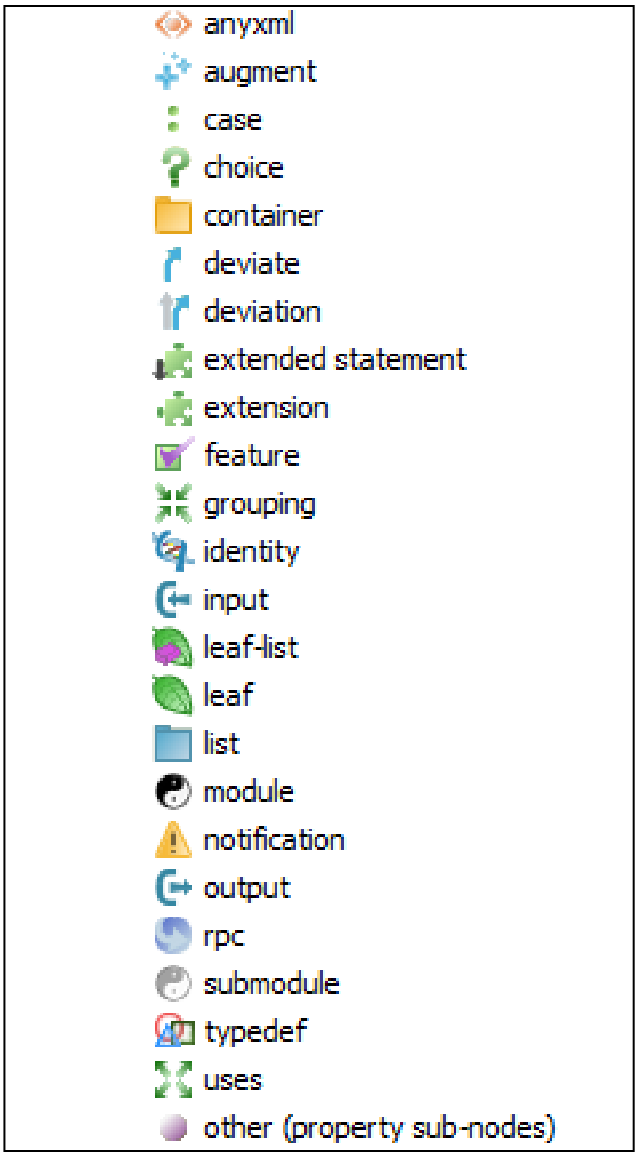
Figure 37: Property (sub)nodes displayed in the YANG Tree panel

3. To hide the properties (sub)nodes of the selected node only, right-click the node and choose the ***View Property Nodes / Toggle for Selected Node*** pop-up command.
4. To hide the properties (sub)nodes for all nodes in the selected subtree, right-click the subtree node and choose the ***View Property Nodes / Hide for Subtree*** pop-up command.

6.3 Different Types of YANG Tree Nodes and Sub-Nodes

6.3.1 Node Icons Representing Different Types of YANG Statements

MG-SOFT NetConf Browser uses the following **node icons** to present different types of YANG statements in the YANG Tree panel:



anyxml
augment
case
choice
container
deviate
deviation
extended statement
extension
feature
grouping
identity
input
leaf-list
leaf
list
module
notification
output
rpc
submodule
typedef
uses
other (property sub-nodes)

Figure 38: Node icons in the YANG Tree panel representing different types of YANG statements

For a description of YANG statements, please refer to the YANG specification: RFC 6020.

In addition, the following **overlay symbols** are displayed on some of the node icons listed above to depict special ‘expanded’ nodes in the YANG tree that represent either a usage of a reusable statement (e.g., `uses` of a `grouping`), or nodes that originate from other statements (e.g., `augment`, `extension`,...):

✚	node originating from an `augment` statement (e.g., 🍃, 📁, ❓,...)
⬇	node originating from **use** of a `grouping` or `extension` (e.g., 🍃, 🧩,…)
⬇ ✚	node originating from **use** of a `grouping` in an `augment` statement (e.g., 🍃,…)
●	‘leafref’ node (`leaf` or `leaf-list` node with the *`leafref`* type property) (e.g., 🍃, 🍃)
🔑	`leaf` node that is a `key` of a `list` (i.e., 🍃, 🍃)

6.3.2 Configuration and State Data Nodes

In contrast to **configuration data** nodes, which are depicted with normal-colored icons, the **state data** nodes are depicted with light-colored (translucent) icons in the YANG Tree panel (Figure 39). This principle lets you quickly distinguish between configuration (e.g., read-write) and state data (read-only) nodes in the YANG tree, for example:

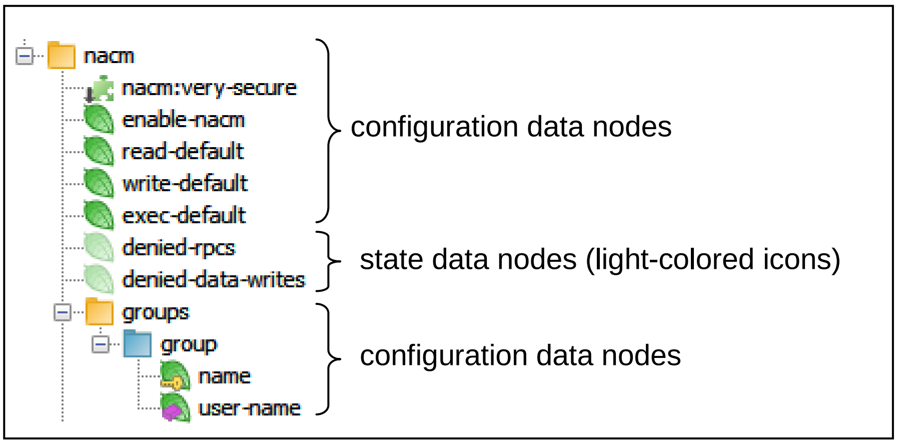


Figure 39: Example of configuration and state data nodes displayed in the YANG Tree panel

> RFC 6241, section 1.4: The information that can be retrieved from a running system is separated into two classes, **configuration data** and **state data**. Configuration data is the set of writable data that is required to transform a system from its initial default state into its current state. State data is the additional data on a system that is not configuration data such as read-only status information and collected statistics.

6.4 Searching for Nodes

NetConf Browser lets you search the YANG tree for nodes and sub-nodes, whose argument contains a user-specified text string. For main nodes, this argument is node **name**, while for sub-nodes the argument can be any text (e.g., the argument of a `description` property sub-node is the entire description text).

Furthermore, the software lets you search for and find all nodes and sub-nodes of a certain type. To do this, select the desired type (e.g., a `container` or `leaf` or `description` or `config`, or `base`, etc.) from the ***Node type*** drop down list and repeatedly press the ***Find Next*** button in the Find Nodes dialog box to “walk” through nodes of selected type.

Optionally, you can combine both search conditions to find a (sub)node of a certain type whose argument contains a user-specified string.

Search is performed on all the loaded YANG and YIN modules and one can start searching up or down from the selected node in the YANG tree.

To find a YANG tree node (start searching from the root node):

5. To enable searching by all node properties, not only the name (e.g., description, type, reference, mandatory, etc.), right-click the node from which you wish to start searching and select the ***View Property Nodes / Show for Subtree*** pop-up command to display the property sub-nodes (●) in the selected subtree.
6. Right-click the **root** node in the YANG Tree panel and select the ***Find Nodes*** pop-up command (Figure 40).

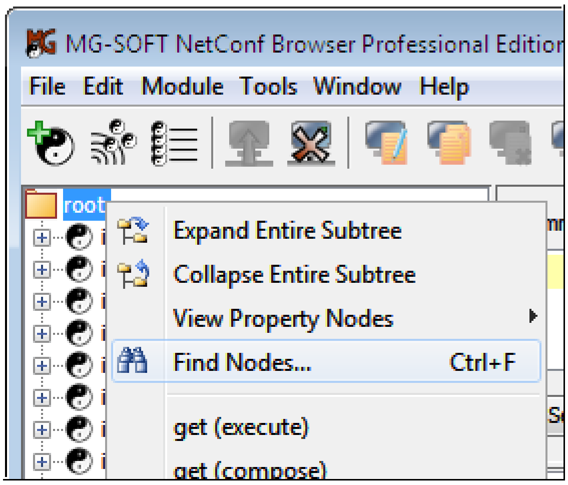
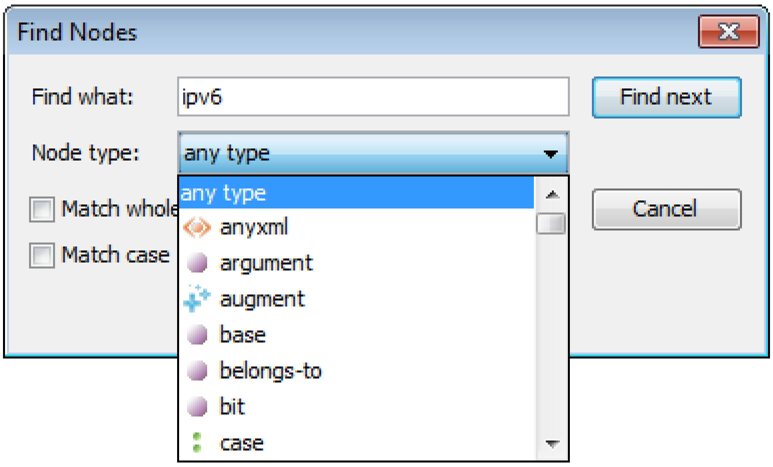


Figure 40: Selecting the *Find Nodes* command in the YANG Tree panel

7. The **Find Nodes** dialog box appears (Figure 41). Into the ***Find what*** input line in the Find Nodes dialog box, enter one or more characters (for example, name or part of the name of the node) you are looking for.

Tip: If you only want to find a particular (sub)type of node, you can leave the ***Find what*** input line empty and select the desired (sub)type from the ***Node type*** drop-down list below.



Find what: ipv6

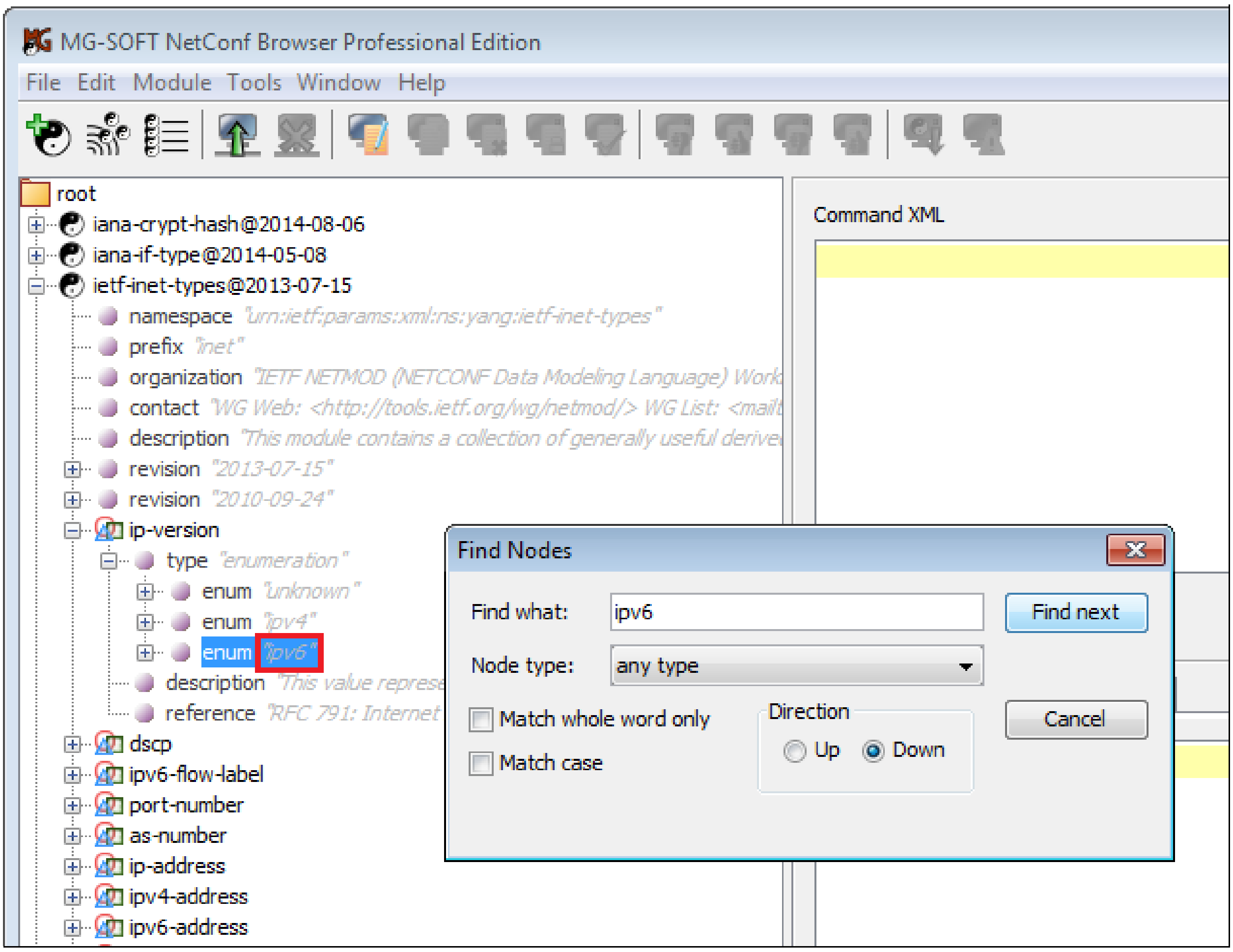
Node type: any type

☐ Match whole

☐ Match case

Figure 41: Specifying the search options in the Find Nodes dialog box

8. From the ***Node type*** drop-down list, optionally select the type of the node or sub-node (●) you are looking for. If the **any type** option is selected, NetConf Browser will search all types of nodes and sub-nodes and find the first one from the selected node (i.e., root node in this case), whose argument contains the string specified in the ***Find what*** input line.
9. Select the ***Down*** radio button in the **Direction** frame (see Figure 42) to enable searching in the direction downward from the selected node.
10. Optionally, select the ***Match case*** (it makes search case sensitive) and ***Match whole word only*** (it finds only occurrences that are whole word, not part of a larger word) checkboxes if they are applicable to your search.
11. Click the ***Find next*** button.
12. NetConf Browser starts searching for the matching node or sub-node in all loaded modules from the root downwards. If a matching node is found, the tree structure from the root to the matching node is expanded and the node is selected in the YANG Tree panel (Figure 42).


MG-SOFT NetConf Browser Professional Edition
File Edit Module Tools Window Help
root
iana-crypt-hash@2014-08-06
iana-if-type@2014-05-08
ietf-inet-types@2013-07-15
namespace "urn:ietf:params:xml:ns:yang:ietf-inet-types"
prefix "inet"
organization "IETF NETMOD (NETCONF Data Modeling Language) Work
contact "WG Web: <http://tools.ietf.org/wg/netmod/> WG List: <mailt
description "This module contains a collection of generally useful derive
revision "2013-07-15"
revision "2010-09-24"
ip-version
type "enumeration"
enum "unknown"
enum "ipv4"
enum "ipv6"
description "This value represe
reference "RFC 791: Internet
dscp
ipv6-flow-label
port-number
as-number
ip-address
ipv4-address
ipv6-address
Command XML
Find Nodes
Find next
Cancel

Find what: ipv6

Node type: any type

□ Match whole word only

□ Match case

Direction: ○ Up ● Down

Figure 42: The found (sub)node is selected in the YANG Tree panel

13. If you would like to continue the search, click the ***Find next*** button again or press the ***F3*** keyboard key to search for the next node whose argument matches the search criteria. Note: The ***F3*** key lets you find the next matching node even after closing the Find Nodes dialog box.

6.5 Viewing Node Properties

NetConf Browser lets you view the properties of any YANG Tree node as it is defined in YANG or YIN definition module. Node properties are displayed in the YANG Node Properties window.

To view the properties of a node in the YANG tree:

1. Right-click the node, whose properties you want to view, and choose ***YANG Node Properties*** command from the pop-up menu (Figure 43).

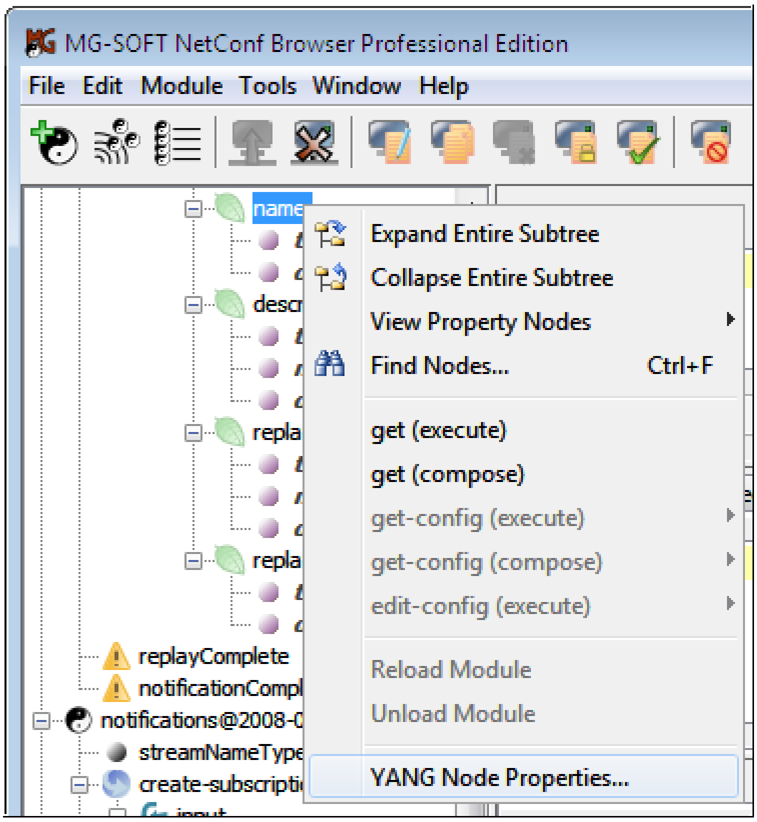
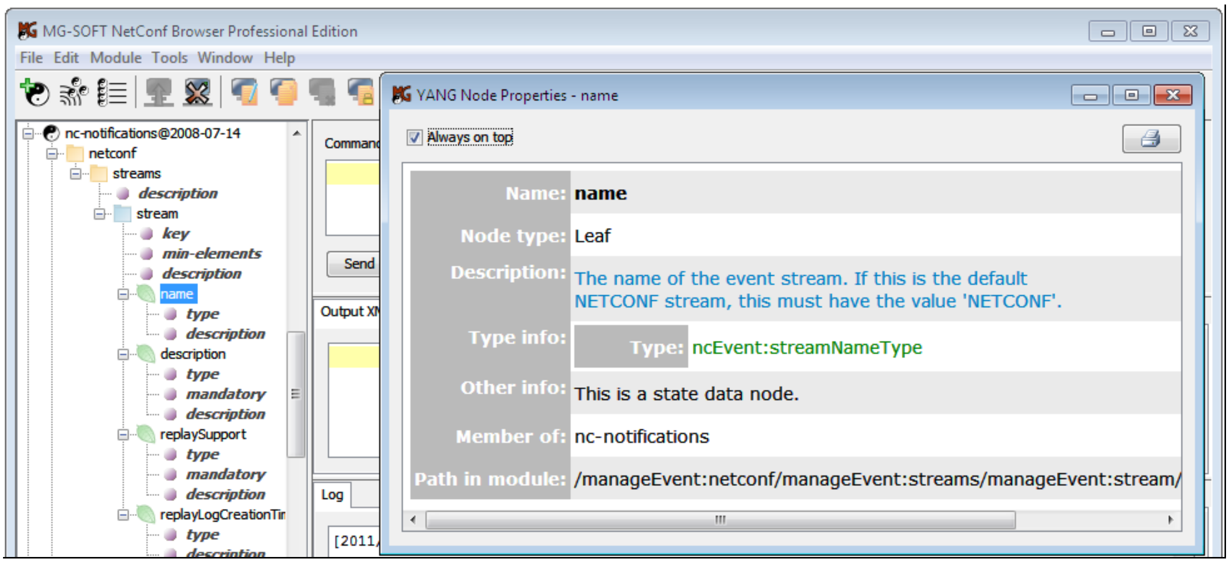


Figure 43: Selecting the YANG Node Properties command in the YANG Tree panel

2. The YANG Node Properties window opens (Figure 44) displaying all the properties of the selected node as defined in the corresponding YANG or YIN definition file.



☑ Always on top

Name: name

Node type: Leaf

Description: The name of the event stream. If this is the default NETCONF stream, this must have the value 'NETCONF'.

Type info: Type: ncEvent:streamNameType

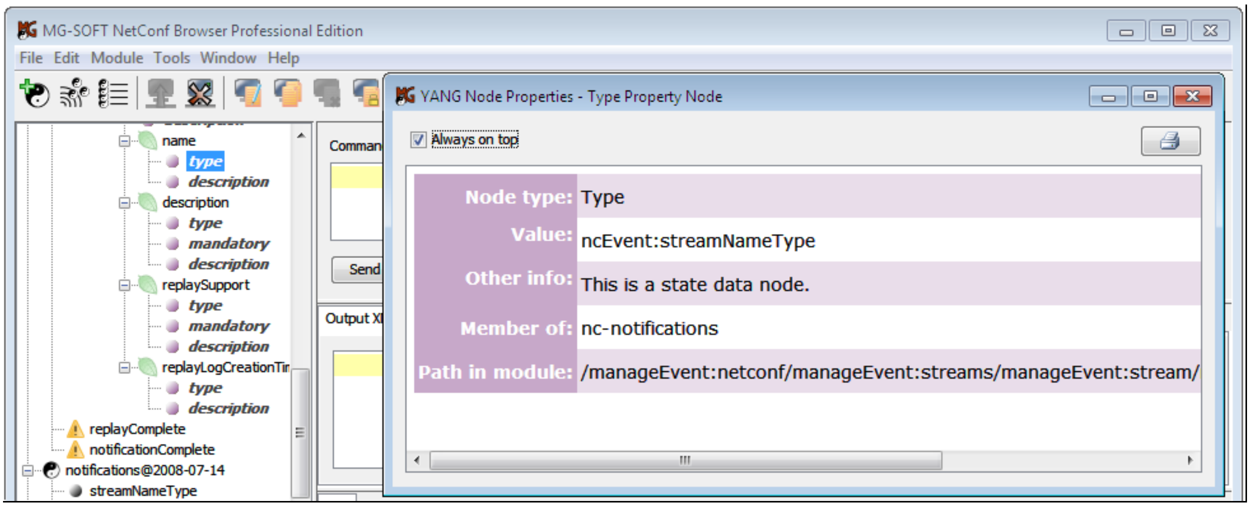
Other info: This is a state data node.

Member of: nc-notifications

Path in module: /manageEvent:netconf/manageEvent:streams/manageEvent:stream/

Figure 44: Viewing properties of the selected node in the YANG Node Properties window

3. Select the ***Stay on top*** checkbox in the Yang Node Properties window to keep this window in the foreground, automatically updating its contents while you click other nodes or sub-nodes in the YANG tree.



☑ Always on top

Node type:	Type
Value:	ncEvent:streamNameType
Other info:	This is a state data node.
Member of:	nc-notifications
Path in module:	/manageEvent:netconf/manageEvent:streams/manageEvent:stream/

Figure 45: Viewing properties of the property sub-node in the YANG Node Properties window

7 LOAD YANG AND YIN MODULES IN NETCONF BROWSER

NetConf Browser supports loading NETCONF modules in YANG and YIN format.

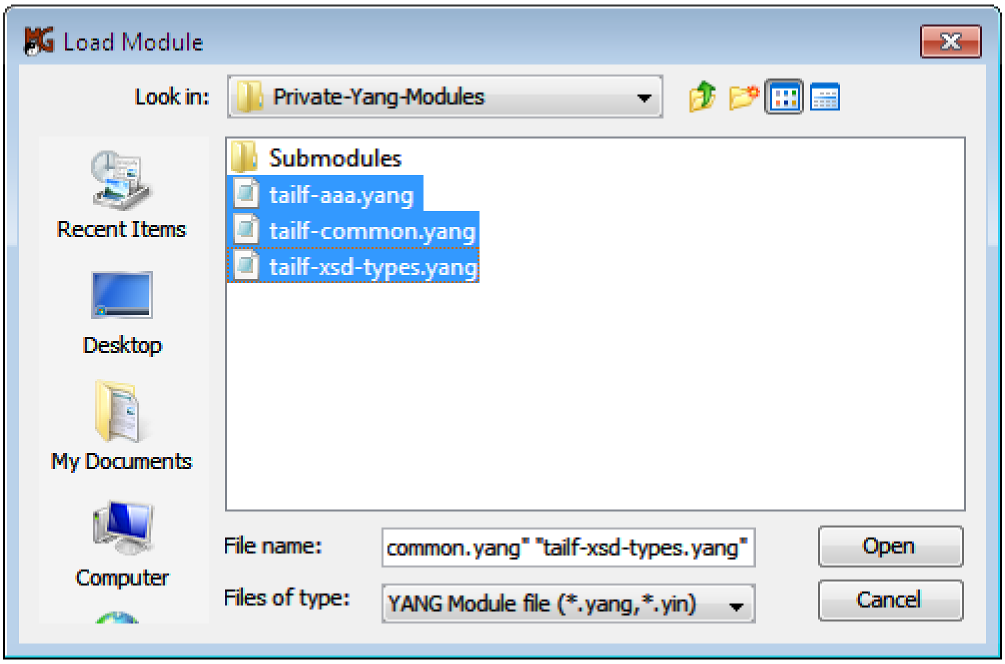
When NetConf Browser is started for the first time, it automatically loads all standard YANG modules that are included in the distribution and graphically displays loaded modules in the YANG Tree panel in the left portion of the main window. Additional, vendor-specific YANG or YIN modules can be loaded by the user and existing modules can be unloaded.

The loading of private modules in NetConf Browser is an important step that will provide you with a clear overview of the node hierarchy and node attributes representing the configuration and state data implemented in the managed NETCONF device.

7.1 Loading YANG and YIN Modules

To load a YANG or YIN module (and all dependent modules it imports and includes):

1. Copy the YANG or YIN module(s) you received from the vendor of the NETCONF device to a local folder of your choice.
2. In NetConf Browser, select the ***Module / Load Module*** command from the main menu.
3. The Load Module dialog box appears (Figure 46). In this dialog box, navigate to the folder containing the private YANG or YIN module(s), select one or more modules (use the CTRL or SHIFT keyboard key to select more than one module), and click the ***Open*** button.



Look in: Private-Yang-Modules

Submodules
tailf-aaa.yang
tailf-common.yang
tailf-xsd-types.yang

File name: common.yang" "tailf-xsd-types.yang"

Files of type: YANG Module file (*.yang,*.yin)

Figure 46: Selecting the YANG modules to load into NetConf Browser

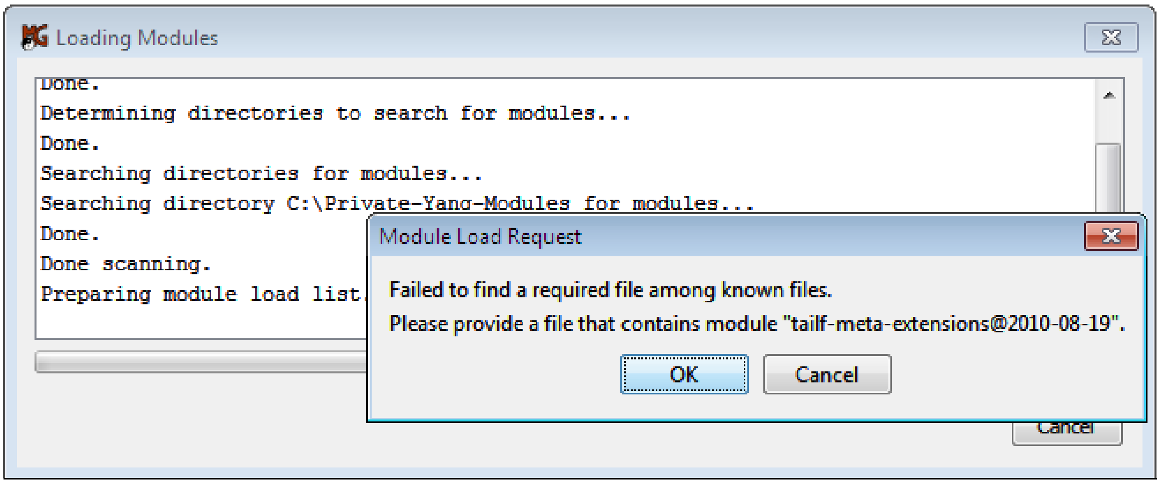
Tip: To select more than one file to load, hold down the ***Ctrl*** key on the keyboard and click the desired module files.

4. NetConf Browser first starts scanning the current folder for files that contain valid YANG and YIN modules to build a list of known modules, i.e., for all modules it detects,

the module name, its revision and full path of the file defining this module is stored in the program cache (known modules list) for future use. When done, NetConf Browser starts parsing the selected YANG or YIN module(s) and checking their consistency. While parsing the module(s), the progress messages and error messages (if any) are being logged in the Log window panel at the bottom of the main window. During this process, every module is also checked for dependencies, i.e., the modules it imports and submodules it includes (this is done recursively for all referenced modules).

- If any dependency is found that is not "known" to NetConf Browser, the Module Load Request dialog box appears (Figure 47) prompting you to specify the location of the file that defines the referenced (sub)module.

> **Note:** "Known modules" are those modules for which information in the program cache already exists, i.e., the standard modules that ship with NetConf Browser and those private modules that have already been loaded in NetConf Browser.

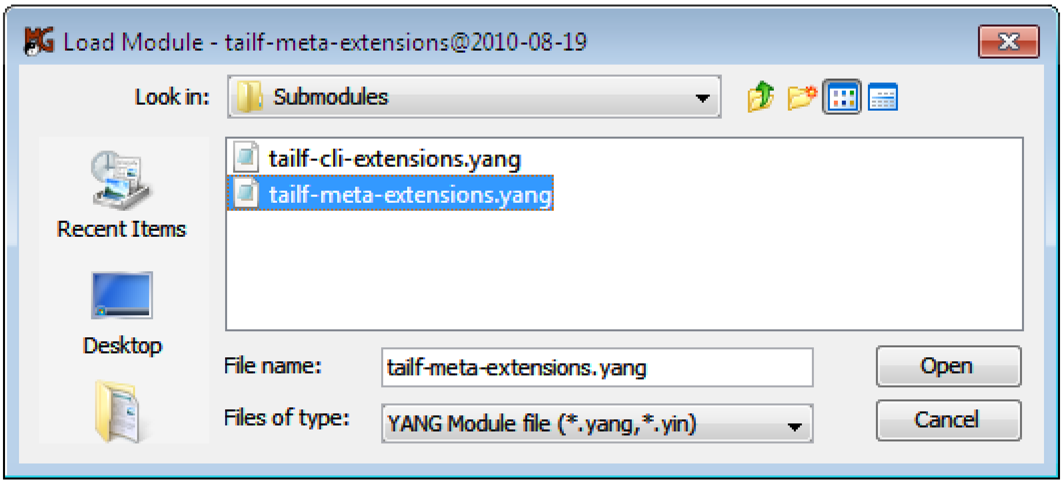


```
Done.
Determining directories to search for modules...
Done.
Searching directories for modules...
Searching directory C:\Private-Yang-Modules for modules...
Done.
Done scanning.
Preparing module load list.
```

Failed to find a required file among known files.
Please provide a file that contains module "tailf-meta-extensions@2010-08-19".

Figure 47: NetConf Browser prompts you to provide the location of the required module

- Click the ***OK*** button in the Module Load Request dialog box to close it and display the Load Module dialog box. In the Load Module dialog box, navigate to the file containing the definition of the required module, select it, and click the ***Open*** button (Figure 48).



Look in: Submodules

tailf-cli-extensions.yang
tailf-meta-extensions.yang

File name: tailf-meta-extensions.yang

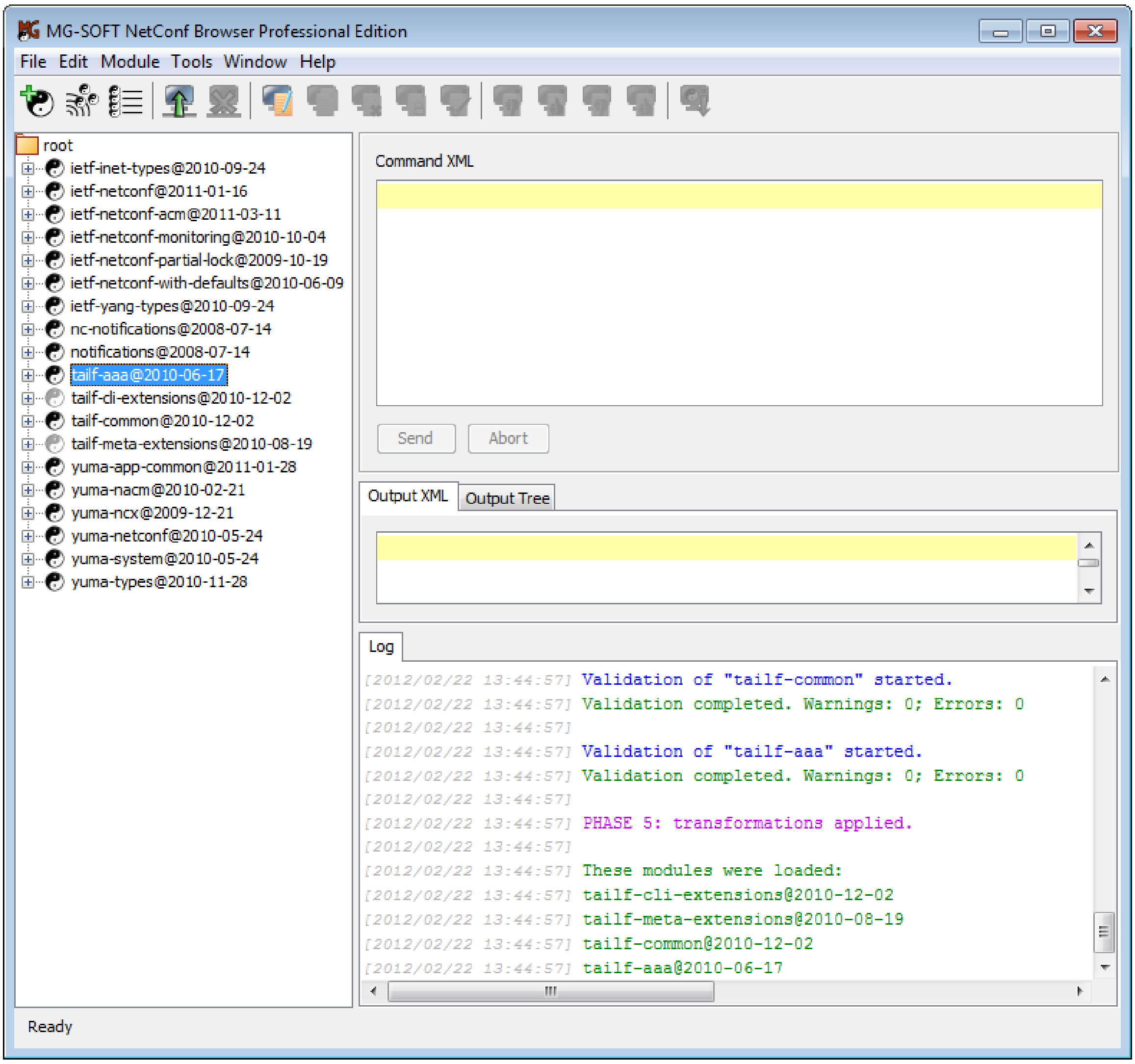
Files of type: YANG Module file (*.yang,*.yin)

Figure 48: Loading a YANG module

- Again, the specified folder is first scanned for files that contain valid YANG and YIN modules to build a list of known modules, then the selected module validated and loaded.
- If you are prompted for any other missing module, repeat the above procedure to specify its location.

Note: Once a module has been loaded from disk, this module and all other modules from the same folder (and optionally all its subfolders) are "registered" and can later be loaded from the **Known Modules** dialog box.

5. After the selected modules have been successfully parsed and validated (no syntax or semantic errors were found), the modules are loaded and displayed in the YANG Tree panel in the left portion of the main window (Figure 49). You can expand the loaded modules to view their tree structure and the properties of nodes, as described in the next section.


MG-SOFT NetConf Browser Professional Edition
File Edit Module Tools Window Help
Command XML
Send
Abort
Output XML
Output Tree
Log
Ready

```
root
  ietf-inet-types@2010-09-24
  ietf-netconf@2011-01-16
  ietf-netconf-acm@2011-03-11
  ietf-netconf-monitoring@2010-10-04
  ietf-netconf-partial-lock@2009-10-19
  ietf-netconf-with-defaults@2010-06-09
  ietf-yang-types@2010-09-24
  nc-notifications@2008-07-14
  notifications@2008-07-14
  tailf-aaa@2010-06-17
  tailf-cli-extensions@2010-12-02
  tailf-common@2010-12-02
  tailf-meta-extensions@2010-08-19
  yuma-app-common@2011-01-28
  yuma-nacm@2010-02-21
  yuma-ncx@2009-12-21
  yuma-netconf@2010-05-24
  yuma-system@2010-05-24
  yuma-types@2010-11-28
```

```
[2012/02/22 13:44:57] Validation of "tailf-common" started.
[2012/02/22 13:44:57] Validation completed. Warnings: 0; Errors: 0
[2012/02/22 13:44:57]
[2012/02/22 13:44:57] Validation of "tailf-aaa" started.
[2012/02/22 13:44:57] Validation completed. Warnings: 0; Errors: 0
[2012/02/22 13:44:57]
[2012/02/22 13:44:57] PHASE 5: transformations applied.
[2012/02/22 13:44:57]
[2012/02/22 13:44:57] These modules were loaded:
[2012/02/22 13:44:57] tailf-cli-extensions@2010-12-02
[2012/02/22 13:44:57] tailf-meta-extensions@2010-08-19
[2012/02/22 13:44:57] tailf-common@2010-12-02
[2012/02/22 13:44:57] tailf-aaa@2010-06-17
```

Figure 49: Newly loaded modules displayed in the YANG Tree window panel

The location of the standard YANG modules bundled with NetConf Browser depends on the operating system used:

Windows: My Documents\MGSOFTNetconfBrowser\modules\

Windows Vista, Windows Server 2008, Windows 7, Windows Server 2012, Windows 8.x, Windows 10:
C:\Users\[username]\Documents\MGSOFTNetconfBrowser\modules\yang

Windows XP, Windows Server 2003:
C:\Documents and Settings\[username]\My Documents\ MGSOFTNetconfBrowser\modules\yang

Linux: ~/Documents/MGSOFTNetconfBrowser/modules/
/home/[username]/Documents/MGSOFTNetconfBrowser/modules/yang/

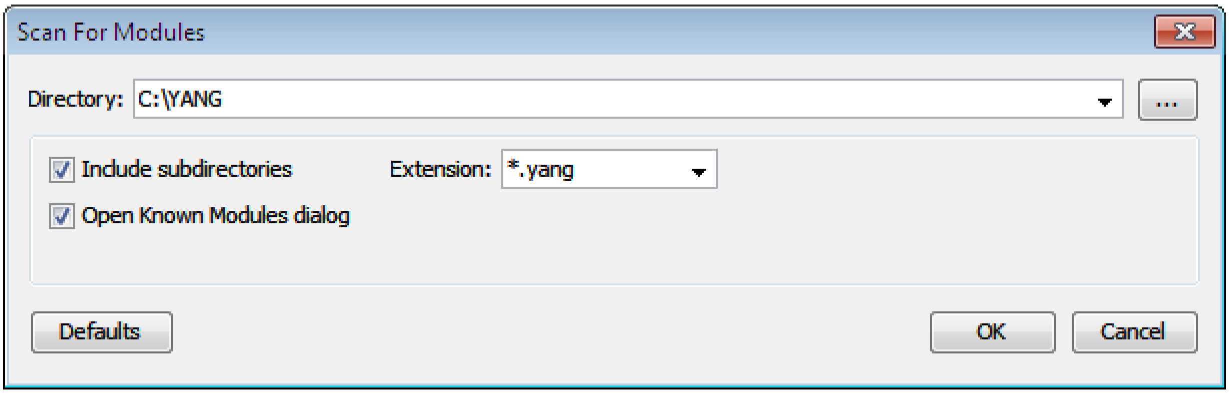
Mac OS X: ~/Documents/MGSOFTNetconfBrowser/modules/
/Users/[username]/Documents/MGSOFTNetconfBrowser/modules/yang

7.2 Scanning Folders for YANG and YIN Modules

NetConf Browser incorporates a convenient functionality that lets you scan selected folders (and optionally also subfolders) for files that contain YANG or YIN modules and automatically "register" them for use with NetConf Browser. Registered files are added to the known modules cache and appear in the Known Modules dialog box, from where they can be loaded into NetConf Browser.

To scan a folder for YANG or YIN module files:

1. In NetConf Browser, select the ***Module / Scan for Modules*** command from the main menu. The Scan for Modules dialog box appears (Figure 46).



Scan For Modules

Directory: C:\YANG

☑ Include subdirectories

Extension: *.yang

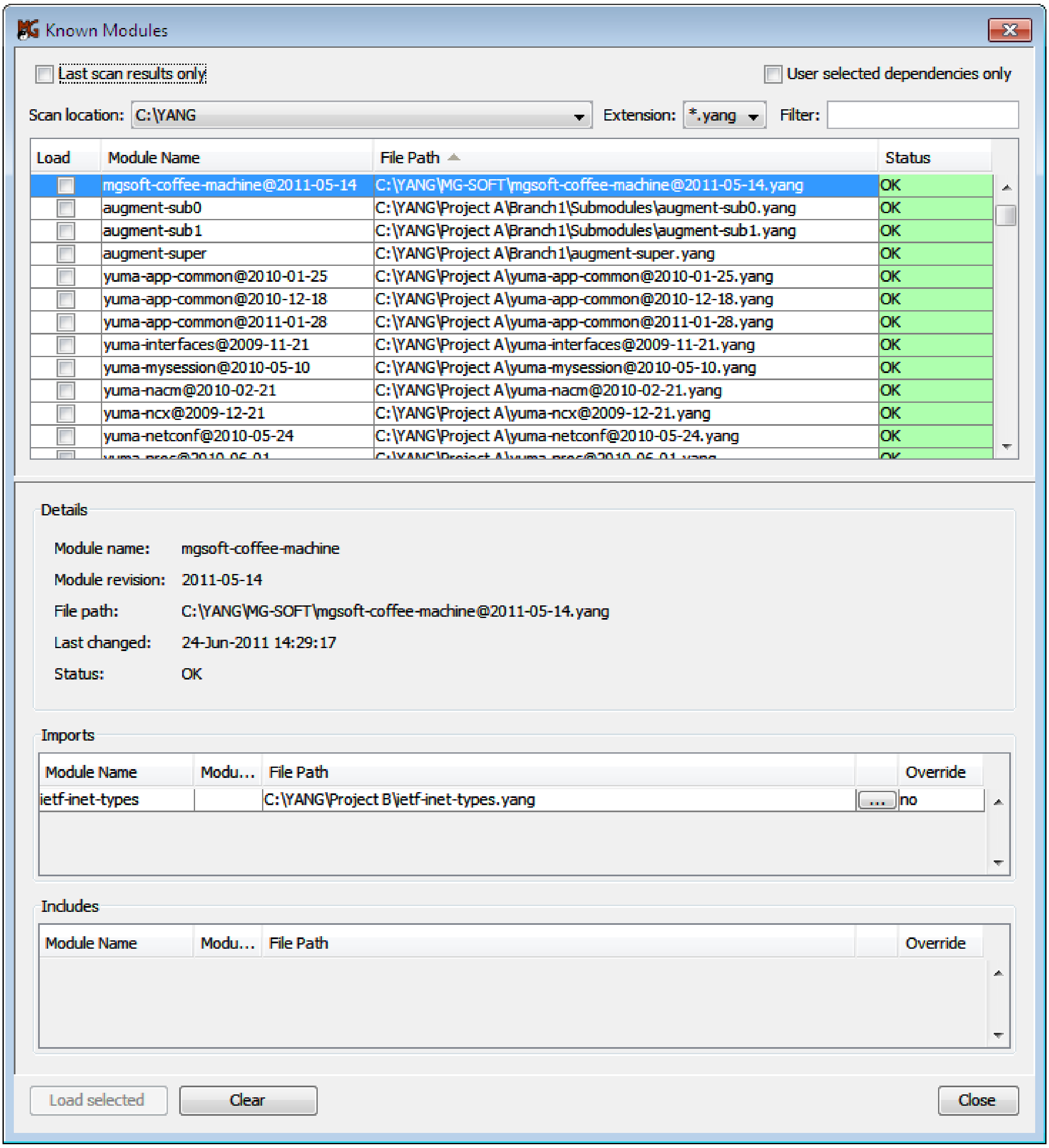
☑ Open Known Modules dialog

Figure 50: The Scan for Modules dialog box

2. In the Scan for Modules dialog box, specify the following:
   - In the ***Directory*** drop-down list, enter the full path of the folder containing (private) YANG or YIN modules,
   - Check the ***Include subdirectories*** checkbox if you want to scan also all the subfolders of the specified folder (directory).
   - In the ***Extension*** drop-down list, select the file mask (e.g., *.yang, *yin, all) to be used for finding the YANG and YIN modules. Only files with the selected extension(s) will be taken into account when scanning for modules.

- ❑ Check the ***Open Known Modules dialog*** checkbox if you want to view the results of the scan operation in the **Known Modules** dialog box.
- ❑ Click the ***Defaults*** button to revert the settings in this dialog box to the default values.

3. Click the ***OK*** button to close the Scan for Modules dialog box and start scanning the selected folder. NetConf Browser scans the specified folder (and its subfolder if selected so) for files that contain valid YANG and YIN modules to build a list of known modules, i.e., for all modules it detects, the module name, its revision and full path of the file defining this module is stored in the program cache (known modules list) for future use. During the scan operation, every module is also checked for dependencies, i.e., NetConf Browser verifies if all the module(s) it imports and submodule(s) it includes are available. The progress of the scan operation is displayed in the Module scan dialog box.
4. When the scan operation finishes, the results are displayed in the Known Modules dialog box (Figure 51):
   - ❑ Select a module in the list of known module files in the upper panel, to view the module details (location, last modified date, etc.) in the middle section, and the module files it imports and submodules files it includes in the lower panel.
   - ❑ The status column in the list of known modules indicates the **`OK`** status if all dependencies (imported module files and included submodule files) for the given module are available. If the status of a module is not **`OK`**, you can specify the location of the missing (sub)module as described in this section.
5. To view only the new modules that have been found in the last scan operation, select the ***Last scan results only*** checkbox in the Known Modules dialog box.
6. To filter results by file extension, i.e., to view only modules saved in files with a specific filename extension, select the desired extension (e.g., *.yang, *.yin) form the ***Extension*** drop-down list in the Known Modules dialog box.
7. To filter results by text, i.e., to view only those lines that contain a specific text string, enter the desired text string into the ***Filter*** input line in the Known Modules dialog box.
8. To view the entire known modules cache (all the registered modules), select the **`All`** entry from the ***Scan location*** drop-down list in the Known Modules dialog box.
9. To load one or more modules (and their dependencies), check the ***Load*** checkbox of the modules you want to load in the list of known module files and click the ***Load Selected*** button at the bottom of the Known Modules dialog box (Figure 59).



□ Last scan results only
□ User selected dependencies only

Scan location: C:\YANG
Extension: *.yang
Filter:

Load	Module Name	File Path	Status
□	mgsoft-coffee-machine@2011-05-14	C:\YANG\MG-SOFT\mgsoft-coffee-machine@2011-05-14.yang	OK
□	augment-sub0	C:\YANG\Project A\Branch1\Submodules\augment-sub0.yang	OK
□	augment-sub1	C:\YANG\Project A\Branch1\Submodules\augment-sub1.yang	OK
□	augment-super	C:\YANG\Project A\Branch1\augment-super.yang	OK
□	yuma-app-common@2010-01-25	C:\YANG\Project A\yuma-app-common@2010-01-25.yang	OK
□	yuma-app-common@2010-12-18	C:\YANG\Project A\yuma-app-common@2010-12-18.yang	OK
□	yuma-app-common@2011-01-28	C:\YANG\Project A\yuma-app-common@2011-01-28.yang	OK
□	yuma-interfaces@2009-11-21	C:\YANG\Project A\yuma-interfaces@2009-11-21.yang	OK
□	yuma-mysession@2010-05-10	C:\YANG\Project A\yuma-mysession@2010-05-10.yang	OK
□	yuma-nacm@2010-02-21	C:\YANG\Project A\yuma-nacm@2010-02-21.yang	OK
□	yuma-ncx@2009-12-21	C:\YANG\Project A\yuma-ncx@2009-12-21.yang	OK
□	yuma-netconf@2010-05-24	C:\YANG\Project A\yuma-netconf@2010-05-24.yang	OK
□	yuma-proc@2010-06-01	C:\YANG\Project A\yuma-proc@2010-06-01.yang	OK

Details

Module name: mgsoft-coffee-machine
Module revision: 2011-05-14
File path: C:\YANG\MG-SOFT\mgsoft-coffee-machine@2011-05-14.yang
Last changed: 24-Jun-2011 14:29:17
Status: OK

Imports

Module Name	Modu...	File Path		Override
ietf-inet-types		C:\YANG\Project B\ietf-inet-types.yang	...	no

Includes

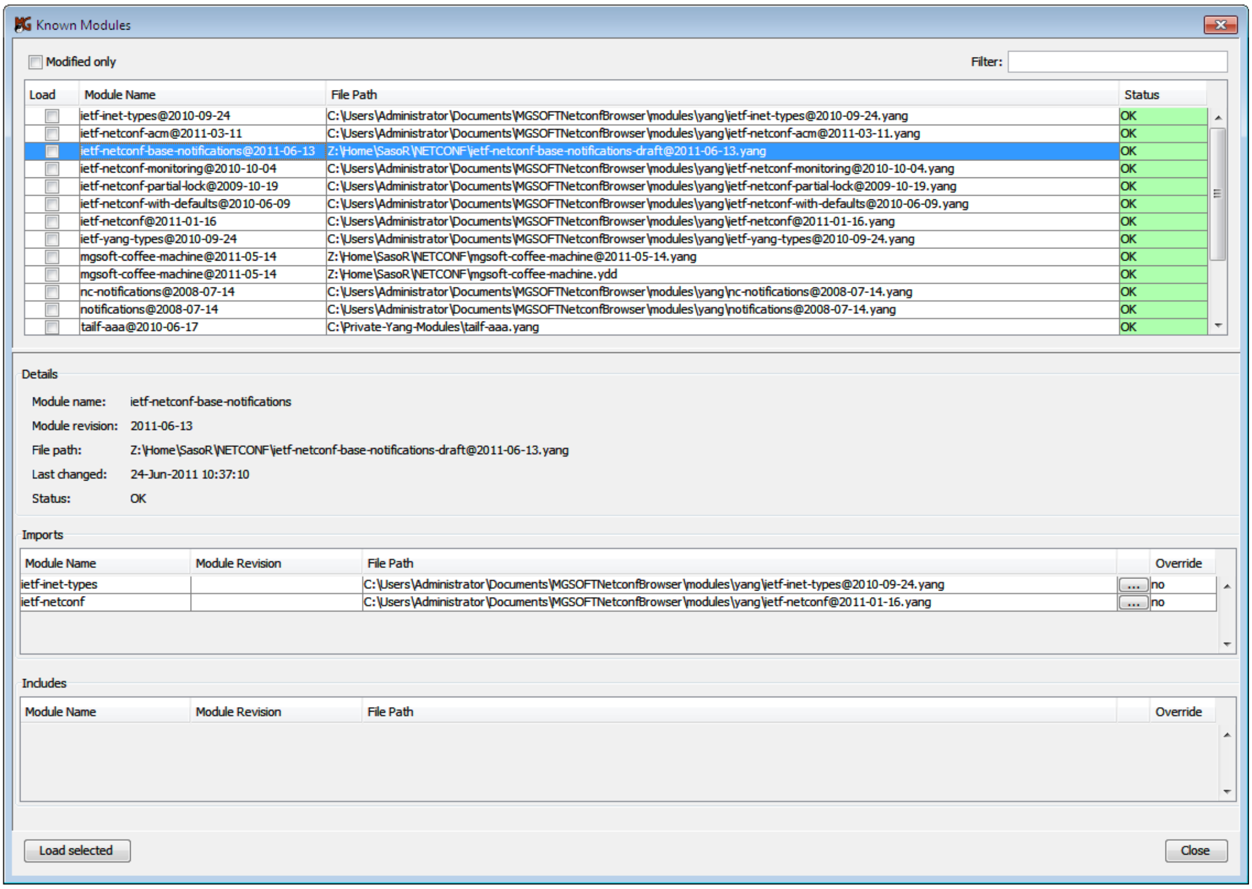
Module Name	Modu...	File Path		Override

Figure 51: The Known Modules dialog box listing YANG modules found in the scanned folder tree

7.3 Loading Known YANG and YIN Modules

Once a folder has been scanned for files that contain YANG or YIN modules, all module files from that folder (and optionally from all its subfolders) are “registered” and can later be loaded from the Known Modules dialog box (if they pass the validation).

The Known Modules dialog box also lets you view and configure dependencies for each module, that is, the module(s) it imports and the submodule(s) it includes.



□ Modified only

Load	Module Name	File Path	Status
□	ietf-inet-types@2010-09-24	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-inet-types@2010-09-24.yang	OK
□	ietf-netconf-acm@2011-03-11	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-netconf-acm@2011-03-11.yang	OK
□	ietf-netconf-base-notifications@2011-06-13	Z:\Home\SasoR\NETCONF\ietf-netconf-base-notifications-draft@2011-06-13.yang	OK
□	ietf-netconf-monitoring@2010-10-04	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-netconf-monitoring@2010-10-04.yang	OK
□	ietf-netconf-partial-lock@2009-10-19	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-netconf-partial-lock@2009-10-19.yang	OK
□	ietf-netconf-with-defaults@2010-06-09	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-netconf-with-defaults@2010-06-09.yang	OK
□	ietf-netconf@2011-01-16	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-netconf@2011-01-16.yang	OK
□	ietf-yang-types@2010-09-24	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-yang-types@2010-09-24.yang	OK
□	mgsoft-coffee-machine@2011-05-14	Z:\Home\SasoR\NETCONF\mgsoft-coffee-machine@2011-05-14.yang	OK
□	mgsoft-coffee-machine@2011-05-14	Z:\Home\SasoR\NETCONF\mgsoft-coffee-machine.ydd	OK
□	nc-notifications@2008-07-14	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\nc-notifications@2008-07-14.yang	OK
□	notifications@2008-07-14	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\notifications@2008-07-14.yang	OK
□	tailf-aaa@2010-06-17	C:\Private-Yang-Modules\tailf-aaa.yang	OK

Details

Module name: ietf-netconf-base-notifications
Module revision: 2011-06-13
File path: Z:\Home\SasoR\NETCONF\ietf-netconf-base-notifications-draft@2011-06-13.yang
Last changed: 24-Jun-2011 10:37:10
Status: OK

Imports

Module Name	Module Revision	File Path		Override
ietf-inet-types		C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-inet-types@2010-09-24.yang	...	no
ietf-netconf		C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\ietf-netconf@2011-01-16.yang	...	no

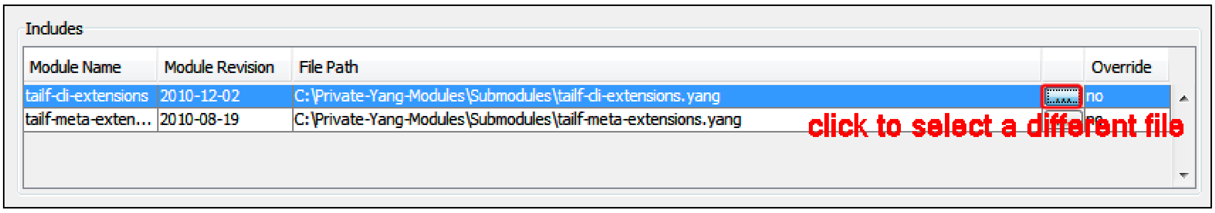
Includes

Module Name	Module Revision	File Path		Override

Figure 52: The Known Modules dialog box (viewing details of a selected module)

To load a YANG or YIN module (and all dependent modules it imports and includes) from the Known Modules dialog box:

1. In NetConf Browser, select the ***Module / Known Modules*** command from the main menu.
2. The Known Modules dialog box appears, listing all YANG and YIN modules that are currently known to NetConf Browser (Figure 52).
   - Select a module in the list of known module files in the upper panel, to view the module details (location, last modified date, etc.) in the middle section, and the module files it imports and submodules files it includes in the lower panel.
   - The status column in the list of known modules indicates the **OK** status if all dependencies (imported module files and included submodule files) for the given module are available.
   - If the status of a module is not **OK**, or if the status is **OK**, but you would like to change the path of the files that will be imported or included by the module, click the ***Browse …*** button next to the **Override** column in the list of Imports or Includes (Figure 53).



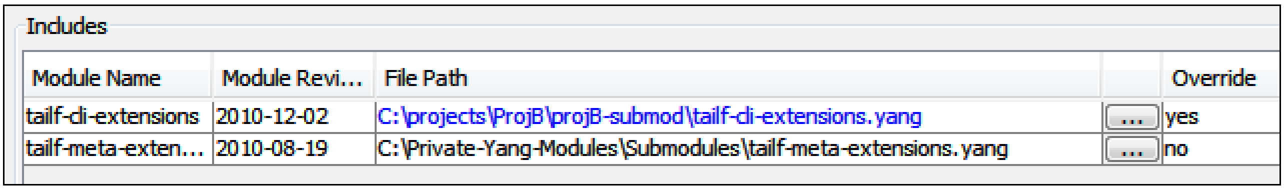
Includes

Module Name	Module Revision	File Path		Override
taif-di-extensions	2010-12-02	C:\Private-Yang-Modules\Submodules\tailf-di-extensions.yang	...	no
tailf-meta-exten...	2010-08-19	C:\Private-Yang-Modules\Submodules\tailf-meta-extensions.yang		no

click to select a different file

Figure 53: Specifying a different include file in the Known Modules dialog box

- ❑ In the Edit Override dialog box that appears, select the ***Browse…*** entry and click the ***OK*** button to open the Load Module dialog box (Figure 46) that lets you browse the file system and select a different file containing the definition of the given submodule.

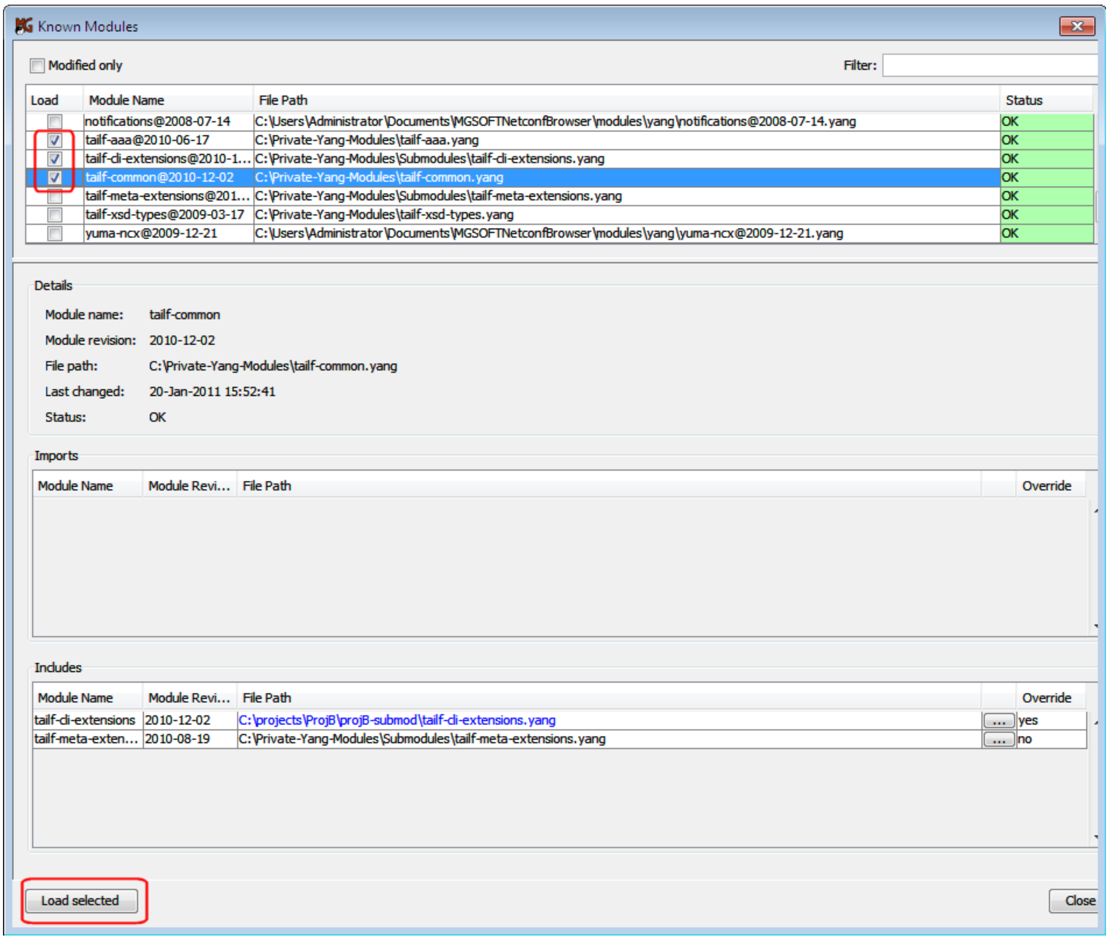


Includes

Module Name	Module Revi...	File Path		Override
tailf-di-extensions	2010-12-02	C:\projects\ProjB\projB-submod\tailf-di-extensions.yang	...	yes
tailf-meta-exten...	2010-08-19	C:\Private-Yang-Modules\Submodules\tailf-meta-extensions.yang	...	no

Figure 54: User overridden include submodule is displayed in blue

3. Check the ***Load*** checkbox of the modules (and their dependencies) you want to load in the list of known module files and click the ***Load Selected*** button at the bottom of the Known Modules dialog box (Figure 55).



Known Modules

□ Modified only

Filter:

Load	Module Name	File Path	Status
□	notifications@2008-07-14	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\notifications@2008-07-14.yang	OK
☑	tailf-aaa@2010-06-17	C:\Private-Yang-Modules\tailf-aaa.yang	OK
☑	tailf-di-extensions@2010-1...	C:\Private-Yang-Modules\Submodules\tailf-di-extensions.yang	OK
☑	tailf-common@2010-12-02	C:\Private-Yang-Modules\tailf-common.yang	OK
□	tailf-meta-extensions@201...	C:\Private-Yang-Modules\Submodules\tailf-meta-extensions.yang	OK
□	tailf-xsd-types@2009-03-17	C:\Private-Yang-Modules\tailf-xsd-types.yang	OK
□	yuma-ncx@2009-12-21	C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\yang\yuma-ncx@2009-12-21.yang	OK

Details

Module name: tailf-common
Module revision: 2010-12-02
File path: C:\Private-Yang-Modules\tailf-common.yang
Last changed: 20-Jan-2011 15:52:41
Status: OK

Imports

Module Name	Module Revi...	File Path		Override

Includes

Module Name	Module Revi...	File Path		Override
tailf-di-extensions	2010-12-02	C:\projects\ProjB\projB-submod\tailf-di-extensions.yang	...	yes
tailf-meta-exten...	2010-08-19	C:\Private-Yang-Modules\Submodules\tailf-meta-extensions.yang	...	no

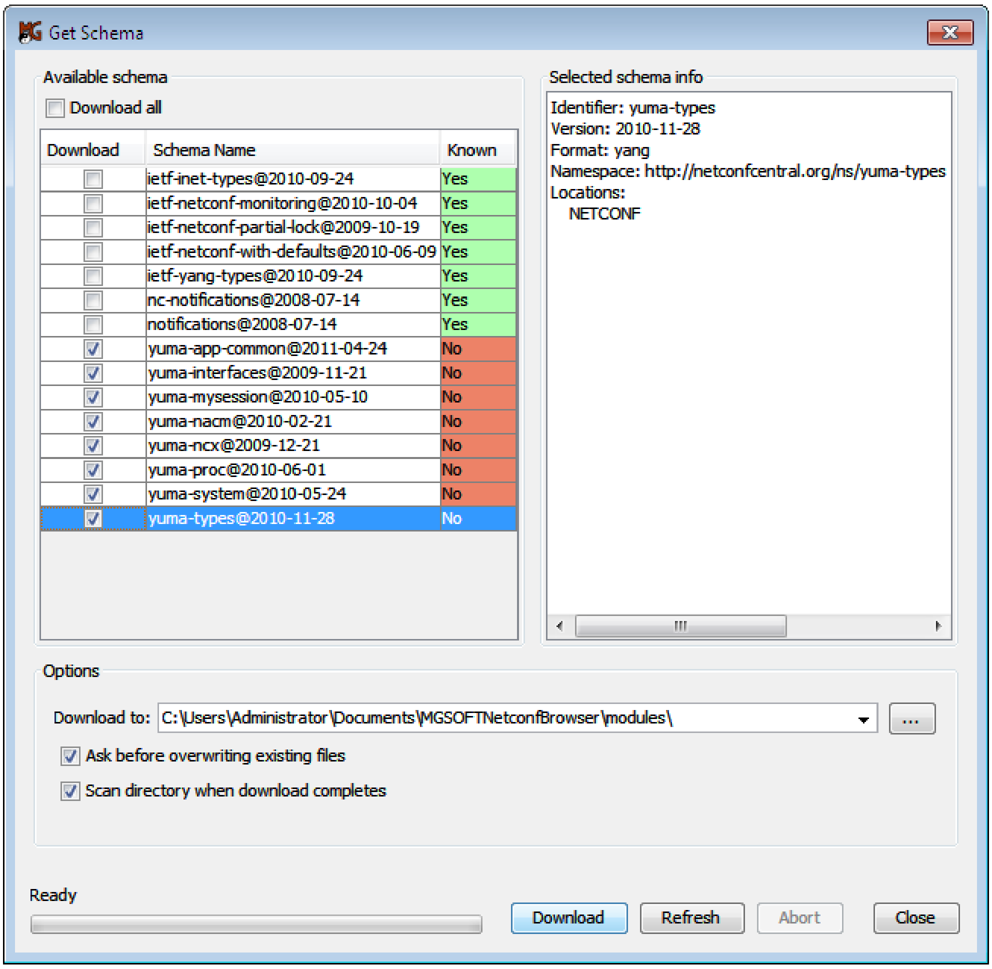
Figure 55: Loading selected modules from the Known Modules dialog box

4. Known Modules dialog box closes and the selected modules and all dependent (sub)modules are validated (checked for syntax and semantic errors). If the modules pass the validation, they are loaded into the YANG tree panel in the main window (unless modules with the same names and revisions are already loaded – in such case you need to unload the currently loaded modules first and then repeat the loading procedure).

7.4 Downloading YANG and YIN Modules from NETCONF Server by Using get-schema Operation

If the currently connected NETCONF server supports the **:ietf-netconf-monitoring** capability, you can use NetConf Browser to discover data models (schema definitions) supported by the NETCONF server and retrieve them from the server using the NETCONF **<get-schema>** operation (specified in RFC 6022), as described in this section.

1. In NetConf Browser, select the ***Tools / Get Schema*** command from the main menu or click the ***Get Schema*** toolbar button ( ).
2. The Get Schema dialog box appears, listing available schema definitions supported by the given NETCONF server (Figure 56).



Download	Schema Name	Known
□	ietf-inet-types@2010-09-24	Yes
□	ietf-netconf-monitoring@2010-10-04	Yes
□	ietf-netconf-partial-lock@2009-10-19	Yes
□	ietf-netconf-with-defaults@2010-06-09	Yes
□	ietf-yang-types@2010-09-24	Yes
□	nc-notifications@2008-07-14	Yes
□	notifications@2008-07-14	Yes
☑	yuma-app-common@2011-04-24	No
☑	yuma-interfaces@2009-11-21	No
☑	yuma-mysession@2010-05-10	No
☑	yuma-nacm@2010-02-21	No
☑	yuma-ncx@2009-12-21	No
☑	yuma-proc@2010-06-01	No
☑	yuma-system@2010-05-24	No
☑	yuma-types@2010-11-28	No

Identifier: yuma-types
Version: 2010-11-28
Format: yang
Namespace: http://netconfcentral.org/ns/yuma-types
Locations:
NETCONF

Download to: C:\Users\Administrator\Documents\MGSOFTNetconfBrowser\modules\

☑ Ask before overwriting existing files

☑ Scan directory when download completes

Figure 56: Selecting schema definitions to download in the Get Schema dialog box

3. Select a schema in the list to view its properties, like the Name (Identifier), Version, Format, Location, Namespace, etc. in the panel on the right side (Figure 56).
4. Select the schema definitions you want to download by checking the checkboxes in front of them in the Get Schema dialog box. To be able to download a schema file, its **Location** must be **NETCONF**. To be able to use the schema file in NetConf Browser, its **Type** must be **YANG** or **YIN**.

> **Note 1:**: Schema definitions may be in different formats, like YANG, YIN, XSD, RNG, and RNC. You can download scheme definitions in any format, however, NetConf Browser can directly load and use only the schema definitions in YANG and YIN format.
>
> **Note 2:** If a schema is marked as **known** in the Get Schema dialog box, then the same schema (YANG or YIN module) already exists on the system and there is not need to download it.

5. In the ***Download to*** input line specify the location where the downloaded schema files will be saved to.
6. Check the ***Ask before overwriting existing files*** checkbox, if you want NetConf Browser to prompt you wit a dialog box whether to overwrite the existing files with the same name (if they exist on disk) or not.
7. Check the ***Scan directory when download completes*** checkbox, if you want NetConf Browser to automatically scan the download location for new YANG and YIN modules and automatically register them for use with NetConf Browser.
8. Click the ***Download*** button at the bottom of the Get Schema dialog box to start downloading the selected schema files. NetConf Browser downloads the schema files and scans them if this options is selected.
9. The downloaded and scanned schema files (modules) appear in the Known Modules dialog box, from where you can load them into NetConf Browser.
10. Click the ***Close*** button at the bottom of the Get Schema dialog box to close it.

8 RETRIEVE CONFIGURATION BY USING NETCONF GET-CONFIG OPERATION

In this section, you will learn how to use the NETCONF get-config operation in NetConf Browser to retrieve all or part of a specified configuration from the NETCONF server it is connected to.

All NETCONF servers must support the “running” configuration datastore, which is the complete configuration currently active on the network device. Additional configuration datastores may be defined and advertised as capabilities, such as the “candidate” and “startup” configuration datastores.

8.1 Retrieving Complete Configuration with NETCONF get-config Operation

The NETCONF protocol uses a remote procedure call (RPC) communication model. A client encodes a request in XML and sends a complete XML document containing <rpc> element to the server. The server responds with a complete XML document containing <rpc-reply> element.

To retrieve the entire active (running) configuration:

1. In the **YANG Tree** window panel in the left portion of the main window, select the **`root`** node.
2. Right-click the root node to display the mouse context (pop-up) menu and select the ***get-config (execute)*** / ***running*** command from the context menu (Figure 57).

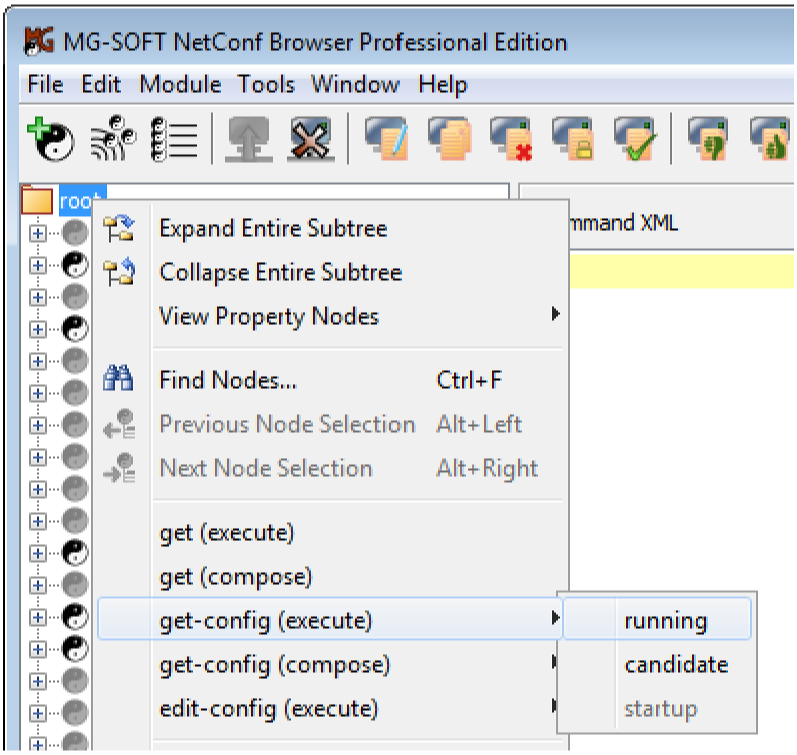


Figure 57: Selecting the get-config (execute) command from the pop-up menu

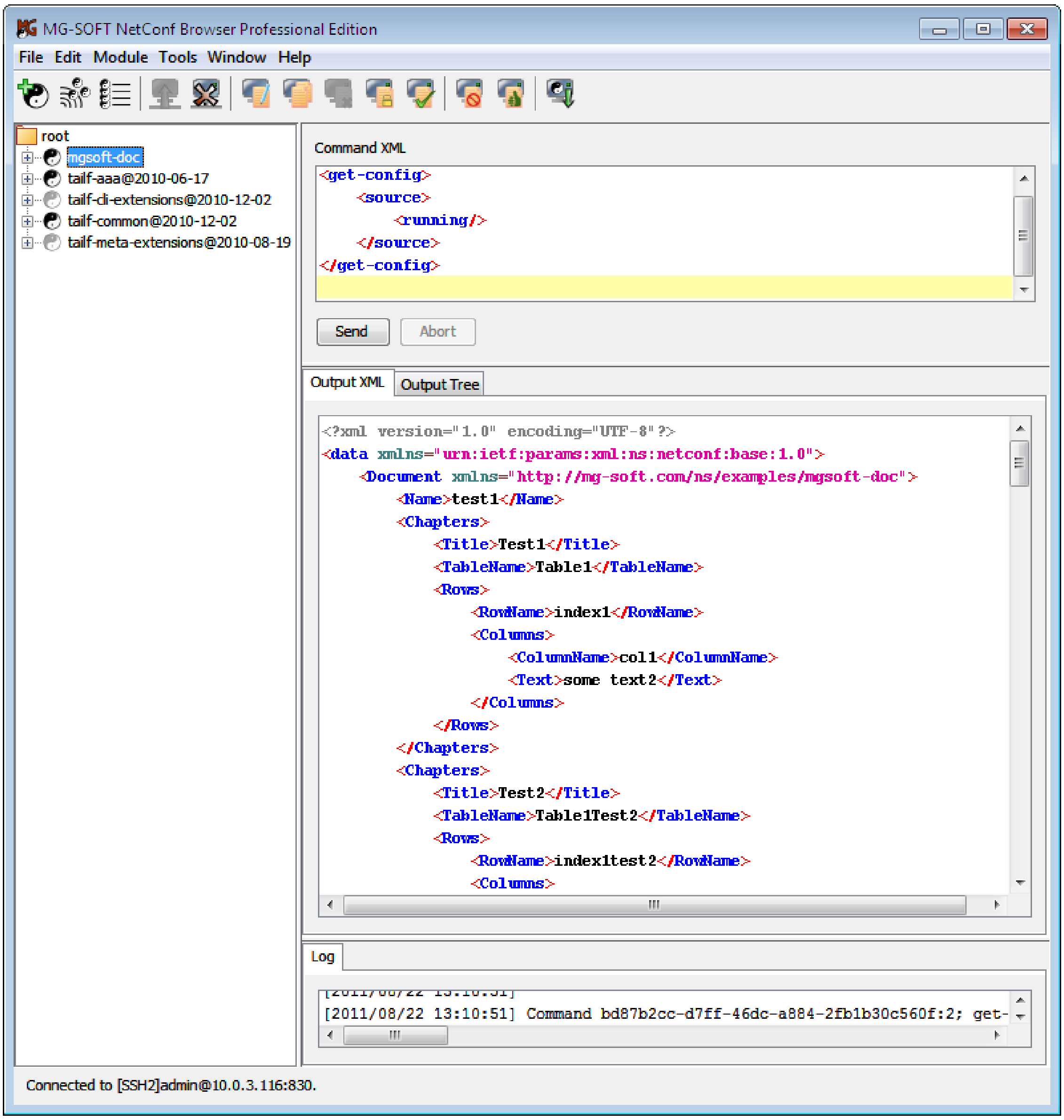
> **Tip:** If a device supports also additional configuration datastores, such as the "`candidate`" or "`startup`", it will advertise these supported datastores in the capabilities exchange procedure that occurs at the beginning of each NETCONF session. When NetConf Browser receives these capabilities, it will enable NETCONF operations on other datastores supported by the given NETCONF server. For example, if the server reports the standard :`candidate` capability, the `candidate` datastore can be used as argument (source or target - where applicable) of the NETCONF get-config, edit-config, copy-config, lock and unlock operations.
>
> To retrieve all the configuration data of the `candidate` or `startup` datastore, select the corresponding datastore from the get-config cascading context menu, e.g.:
>
> ***get-config (execute)*** / ***candidate***
>
> ***get-config (execute)*** / ***startup***

3. NetConf Browser creates and sends the NETCONF <get-config> request to the server. It contains a <source> tag element and the <running/> tag within an <rpc> tag element, e.g.:

```
<rpc message-id="101"
 xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <get-config>
        <source>
          <running/>
        </source>
      </get-config>
    </rpc>
```

To conserve space and increase the readability, the outer <rpc> message wrapper is not displayed in the NetConf Browser (Figure 58) **and in the remaining sections of this document.**

4. In response, the server sends an <rpc-reply> message containing a <data> element with the results of the query, which in this case is the entire running configuration (Figure 58).


MG-SOFT NetConf Browser Professional Edition
File Edit Module Tools Window Help
root
mgsoft-doc
tailf-aaa@2010-06-17
tailf-cli-extensions@2010-12-02
tailf-common@2010-12-02
tailf-meta-extensions@2010-08-19
Command XML
Send
Abort
Output XML
Output Tree
Log

```
<get-config>
    <source>
        <running/>
    </source>
</get-config>
```

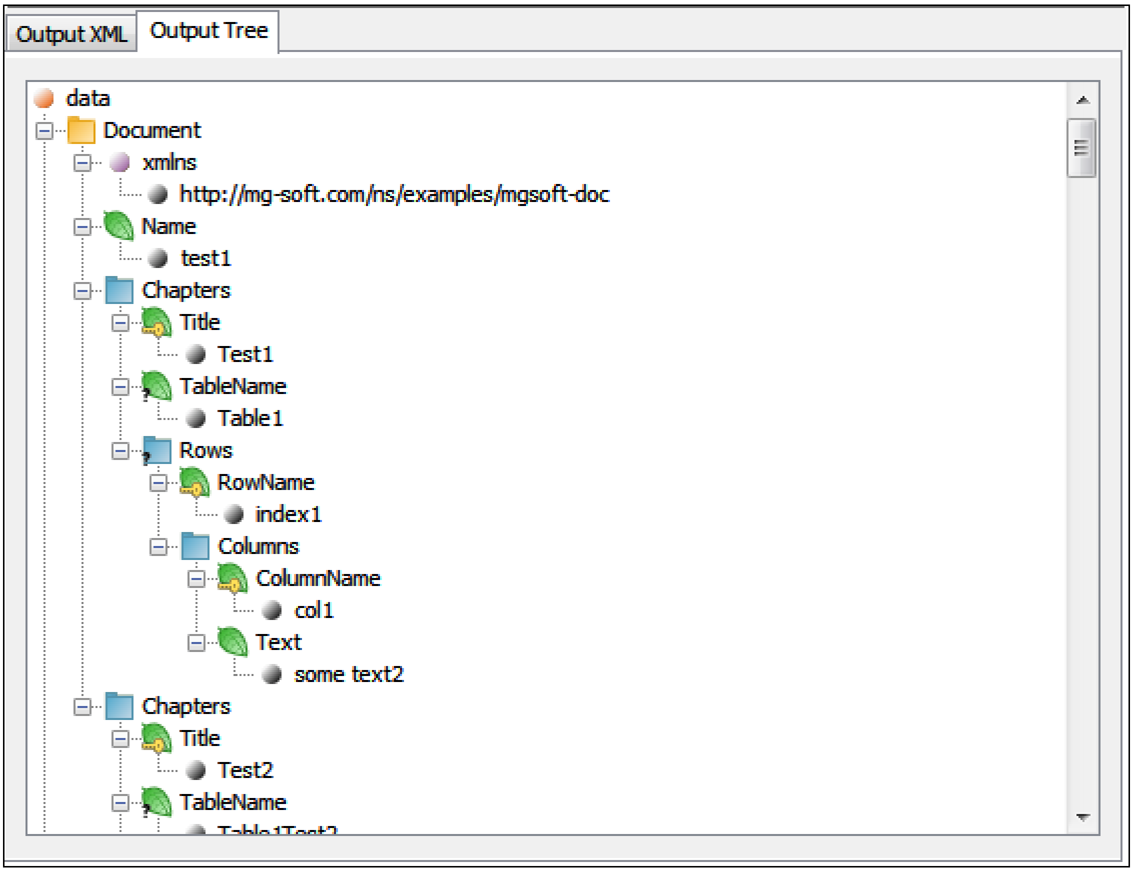
```
<?xml version="1.0" encoding="UTF-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <Document xmlns="http://mg-soft.com/ns/examples/mgsoft-doc">
        <Name>test1</Name>
        <Chapters>
            <Title>Test1</Title>
            <TableName>Table1</TableName>
            <Rows>
                <RowName>index1</RowName>
                <Columns>
                    <ColumnName>col1</ColumnName>
                    <Text>some text2</Text>
                </Columns>
            </Rows>
        </Chapters>
        <Chapters>
            <Title>Test2</Title>
            <TableName>Table1Test2</TableName>
            <Rows>
                <RowName>index1test2</RowName>
                <Columns>
```

```
[2011/08/22 13:10:51]
[2011/08/22 13:10:51] Command bd87b2cc-d7ff-46dc-a884-2fb1b30c560f:2; get-
```

Connected to [SSH2]admin@10.0.3.116:830.

Figure 58: Viewing a get-config request (upper panel) and response (middle panel) in XML form

5. To view the retrieved results in form of a hierarchically arranged tree, containing nodes and their values, select the **Output Tree** tab in the central panel of the main window (Figure 59).



```
data
  Document
    xmlns
      http://mg-soft.com/ns/examples/mgsoft-doc
    Name
      test1
    Chapters
      Title
        Test1
      TableName
        Table1
      Rows
        RowName
          index1
        Columns
          ColumnName
            col1
          Text
            some text2
    Chapters
      Title
        Test2
      TableName
        Table1Test2
```

Figure 59: Viewing retrieved configuration in a tree view

8.2 Retrieving Parts of Configuration with NETCONF get-config Operation

YANG defines four types of nodes for configuration data modeling:

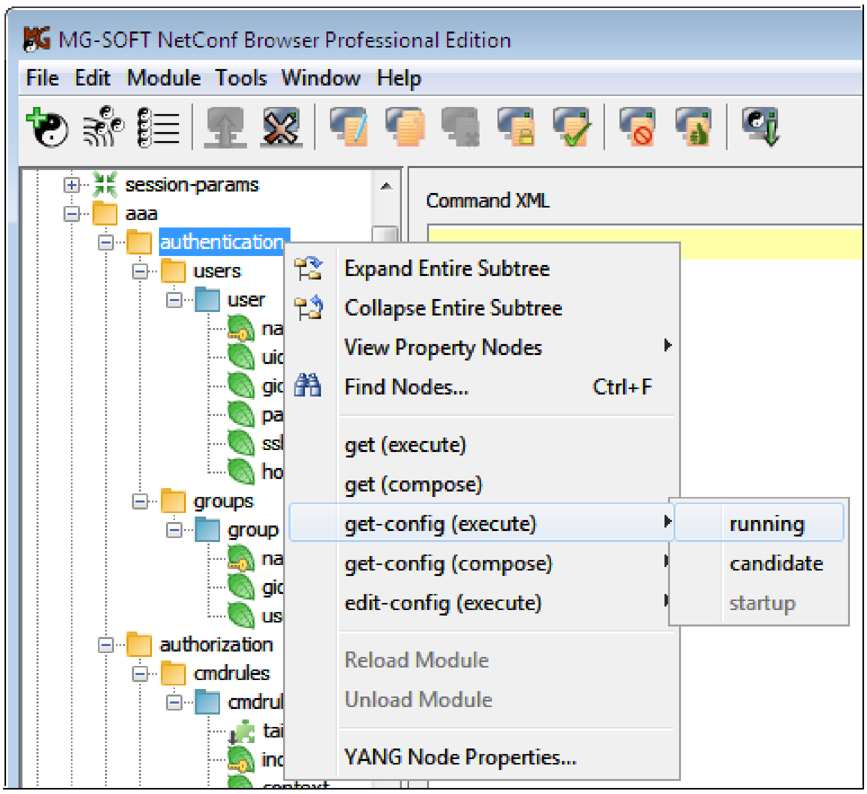
Node type	Node icon in NetConf Browser
leaf	🍃
leaf-list	🍃
container	📁
list	🟦

get-config operation can be performed on the above node types to retrieve only the values of the selected leaf or leaf-list node instances or the entire subtree specified by the selected container or list node.

Retrieving a part of the configuration is achieved by XML subtree filtering that allows an application to select particular XML subtrees to include in a response to a NETCONF get or get-config operation.

To retrieve a part of the running configuration:

1. In the **YANG Tree** window panel in the left portion of the main window, select the leaf or leaf-list node or the subtree (container node or list node) that you wish to retrieve.
2. Right-click the selected node to display the mouse context (pop-up) menu and select the ***get-config (execute)*** / ***running*** command from the context menu (Figure 60).



Use the following commands to retrieve the configuration data for the selected subtree from the `candidate` or the `startup` datastore (if supported by the device):

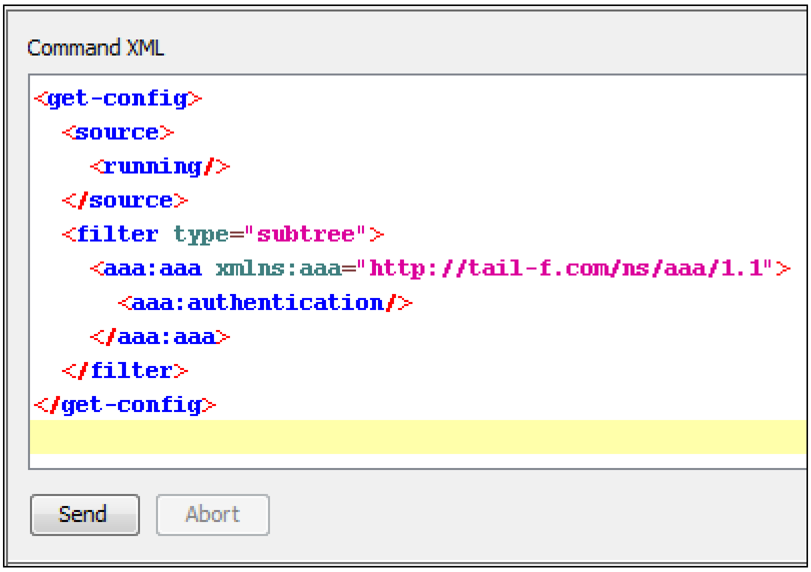
get-config (execute) / ***candidate***

or

get-config (execute) / ***startup***

Figure 60: Selecting the get-config (execute) command on a subtree node

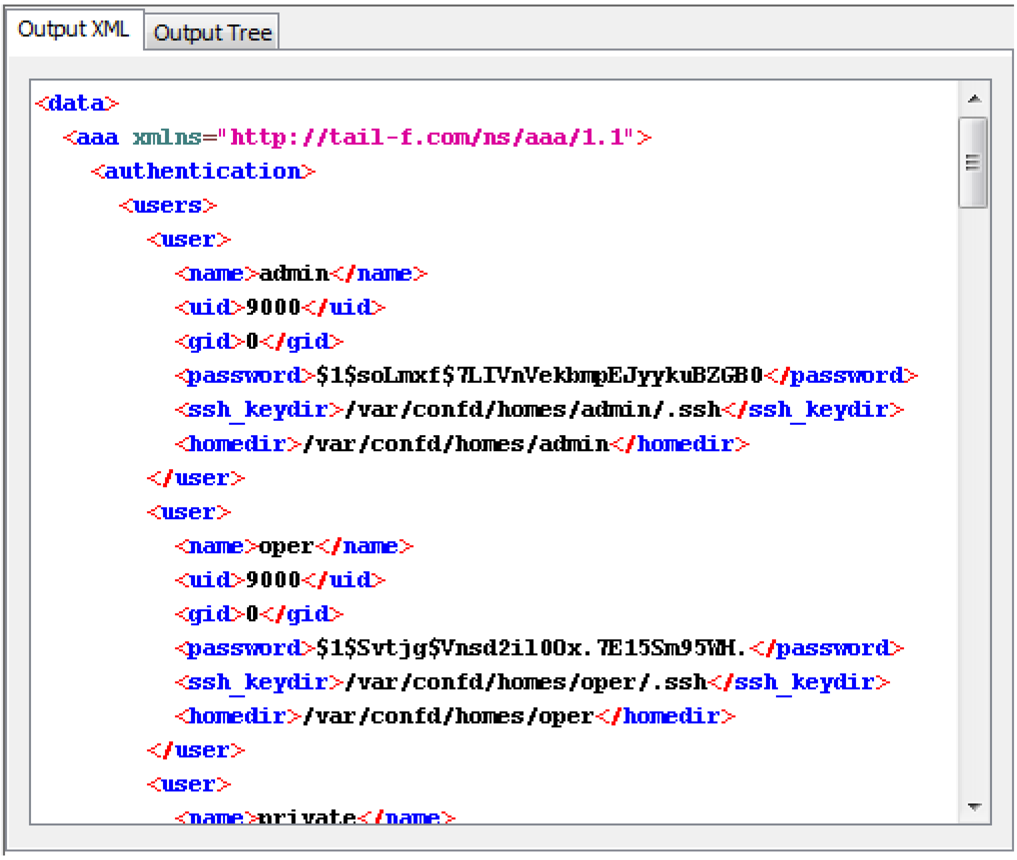
3. NetConf Browser creates and sends the NETCONF get-config request containing the <filter> element for the selected subtree to the server (Figure 61):



```
<get-config>
  <source>
    <running/>
  </source>
  <filter type="subtree">
    <aaa:aaa xmlns:aaa="http://tail-f.com/ns/aaa/1.1">
      <aaa:authentication/>
    </aaa:aaa>
  </filter>
</get-config>
```

Figure 61: get-config request for a subtree (aaa:authentication)

4. In response, the server sends an <rpc-reply> message containing a <data> element with the results of the query, which in this case is the `aaa:authentication` subtree (Figure 58).



```
<data>
  <aaa xmlns="http://tail-f.com/ns/aaa/1.1">
    <authentication>
      <users>
        <user>
          <name>admin</name>
          <uid>9000</uid>
          <gid>0</gid>
          <password>$1$soLmxf$7LIVnVekbmpEJyykuBZGB0</password>
          <ssh_keydir>/var/confd/homes/admin/.ssh</ssh_keydir>
          <homedir>/var/confd/homes/admin</homedir>
        </user>
        <user>
          <name>oper</name>
          <uid>9000</uid>
          <gid>0</gid>
          <password>$1$Svtjg$Vnsd2il0Ox.7E15Sm95WH.</password>
          <ssh_keydir>/var/confd/homes/oper/.ssh</ssh_keydir>
          <homedir>/var/confd/homes/oper</homedir>
        </user>
        <user>
          <name>private</name>
```

Figure 62: Retrieved subtree in XML form (aaa:authentication)

5. Use the scrollbars in the **Output XML** window panel to view the rest of the response in XML form. To view the retrieved results in form of a hierarchically arranged tree, containing nodes and their values, select the **Output Tree** tab in the central panel of the main window.

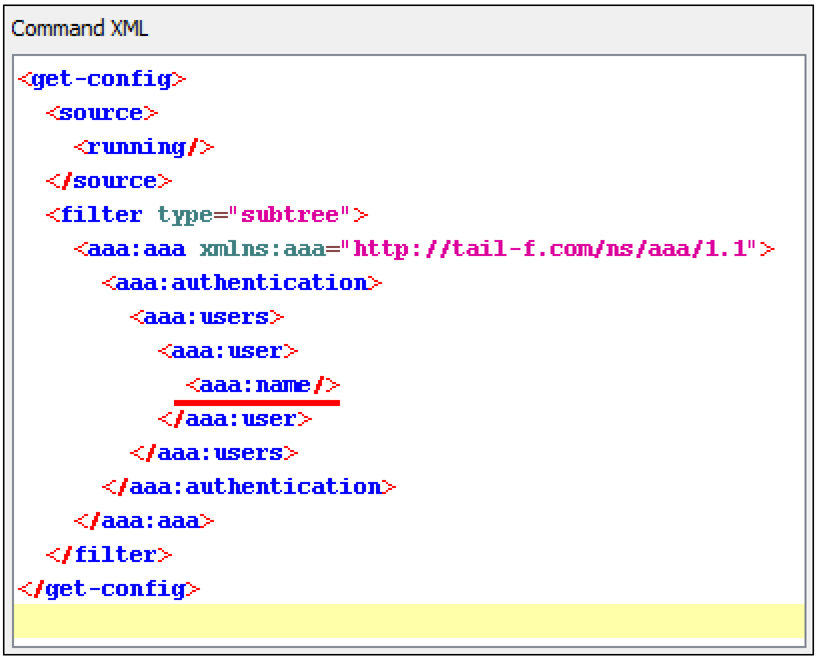
Example: How to retrieve data for the “admin” user only (edit filter)

This example refers to the `tailf-aaa` module (prefix: `aaa`). A part of this module models the authentication scheme based on users and user groups that are hierarchically structured as displayed in Figure 64. The principle of editing the subtree filter described in this example can be applied to any other YANG/YIN module that is similarly structured.

1. In the **YANG Tree window** panel in the left portion of the main window, select the leaf **name**.

2. Right-click the selected node to display the mouse context (pop-up) menu and select the ***get-config (compose)*** / ***running*** command from the context menu (Figure 60).

3. The **get-config** message for retrieving all instances of the **`name`** leaf node from the running configuration datastore is automatically created and displayed in the upper widow panel (Command XML). The get-config request has not been sent to the network yet.

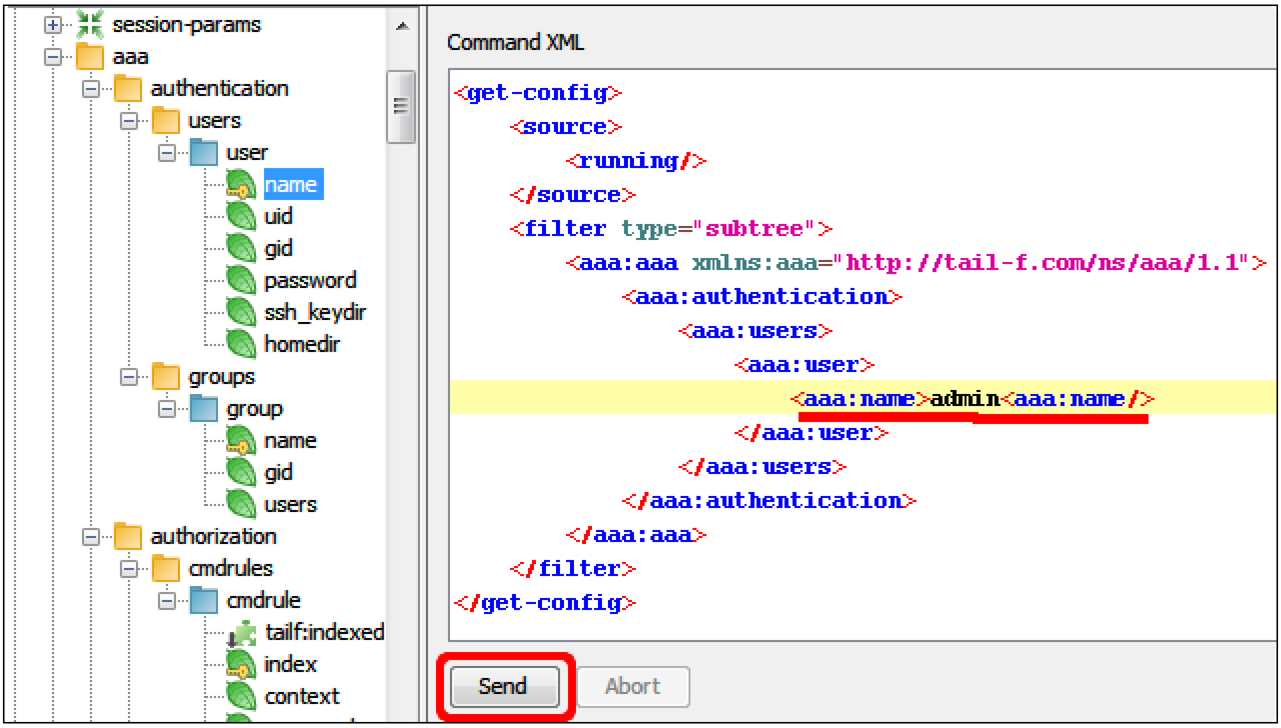


```
<get-config>
  <source>
    <running/>
  </source>
  <filter type="subtree">
    <aaa:aaa xmlns:aaa="http://tail-f.com/ns/aaa/1.1">
      <aaa:authentication>
        <aaa:users>
          <aaa:user>
            <aaa:name/>
          </aaa:user>
        </aaa:users>
      </aaa:authentication>
    </aaa:aaa>
  </filter>
</get-config>
```

Figure 63: Automatically generated get-config (compose) command

4. Click into the **Command XML** window panel and edit the filter to match your preferences, e.g., `<aaa:name>admin</aaa:name>` (Figure 64).

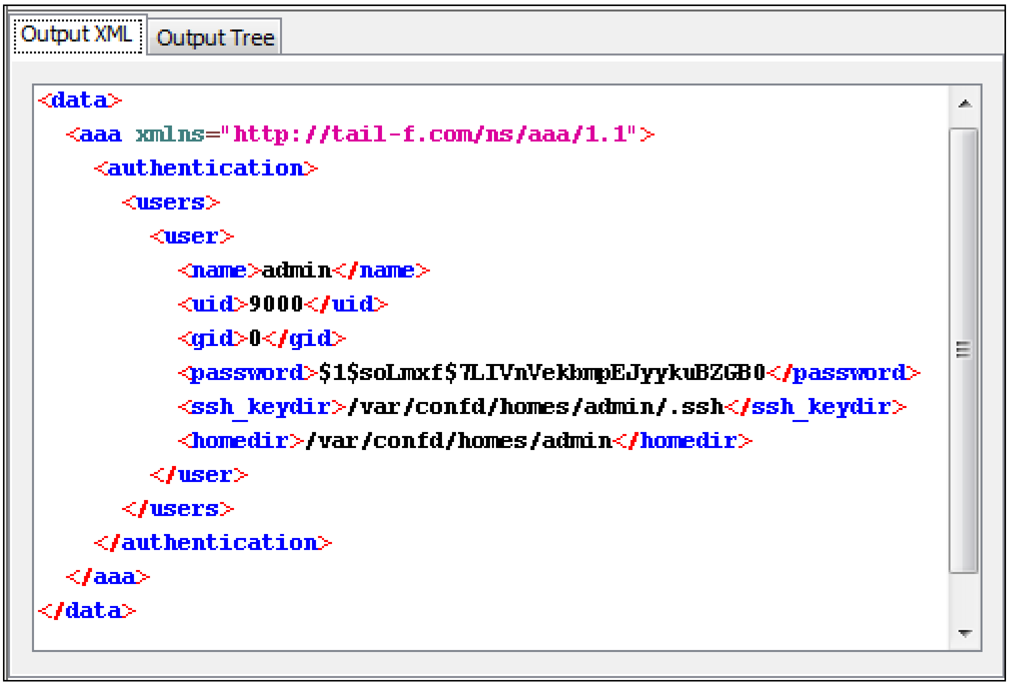
5. After you have edited the filter, click the ***Send*** button at the bottom of the Command XML window panel (Figure 64) to send the edited NETCONF request to the device.



```
<get-config>
    <source>
        <running/>
    </source>
    <filter type="subtree">
        <aaa:aaa xmlns:aaa="http://tail-f.com/ns/aaa/1.1">
            <aaa:authentication>
                <aaa:users>
                    <aaa:user>
                        <aaa:name>admin<aaa:name/>
                    </aaa:user>
                </aaa:users>
            </aaa:authentication>
        </aaa:aaa>
    </filter>
</get-config>
```

Figure 64: Sending a get-config request containing a filter edited by user

6. The queried device responds with a reply message containing all instances of the sibling set containing <aaa:name> for which the value of <aaa:name> equals **`admin`** (Figure 65).



```
<data>
  <aaa xmlns="http://tail-f.com/ns/aaa/1.1">
    <authentication>
      <users>
        <user>
          <name>admin</name>
          <uid>9000</uid>
          <gid>0</gid>
          <password>$1$soLmxf$7LIVnVekbmpEJyykuBZGB0</password>
          <ssh_keydir>/var/confd/homes/admin/.ssh</ssh_keydir>
          <homedir>/var/confd/homes/admin</homedir>
        </user>
      </users>
    </authentication>
  </aaa>
</data>
```

Figure 65: Retrieved configuration data for the user ‘admin’

9 RETRIEVE CONFIGURATION AND STATE DATA BY USING NETCONF GET OPERATION

In contrast to NETCONF get-config operation, which retrieves only configuration data, the NETCONF get operation retrieves both, the configuration and state data from the active configuration datastore. Using the NETCONF get operation is especially useful for monitoring the device that runs the NETCONF server, like monitoring the status and performance of the device, its network interfaces and other resources.

> RFC 6241, section 1.4: The information that can be retrieved from a running system is separated into two classes, **configuration data** and **state data**. Configuration data is the set of writable data that is required to transform a system from its initial default state into its current state. State data is the additional data on a system that is not configuration data such as read-only status information and collected statistics.

9.1 Retrieving Configuration and State Data Using NETCONF get Operation

This section describes how to retrieve all the information (configuration and state data) from the running configuration:

7. In the **YANG Tree** window panel in the left portion of the main window, select the 📁 **`root`** node.

1. Right-click the root node to display the mouse context (pop-up) menu and select the ***get (execute)*** command from the context menu (Figure 66).

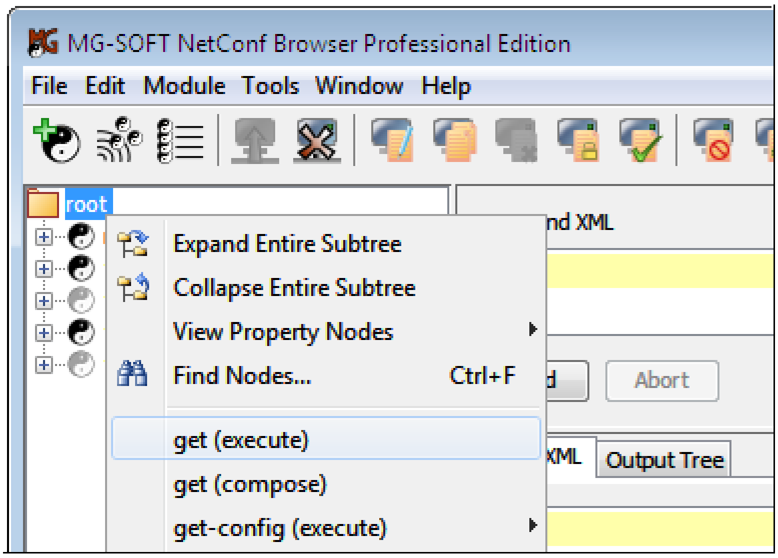
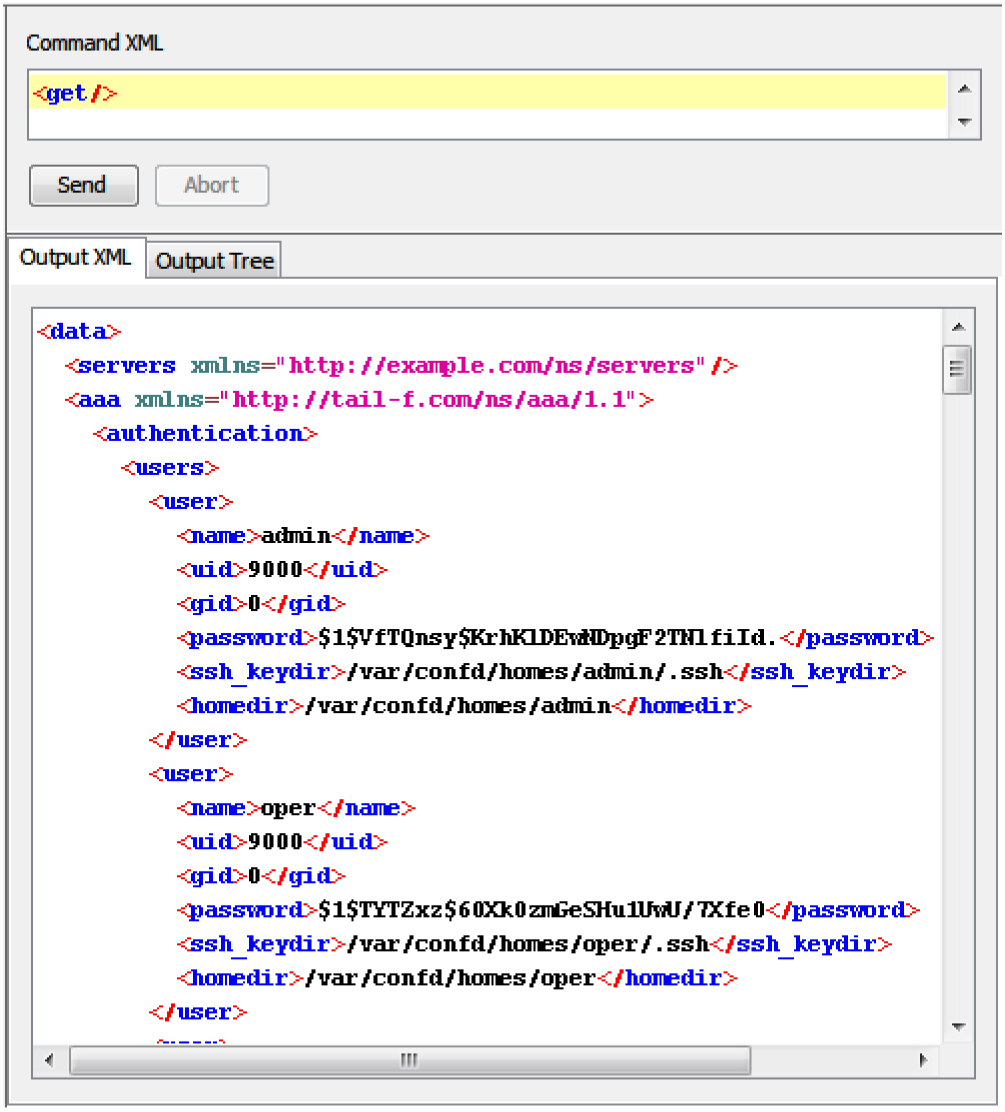


Figure 66: Selecting the NETCONF get operation from the context menu

2. NetConf Browser creates and sends the NETCONF **get** request with no elements (<get/>) to the server. In response, the server sends an <rpc-reply> message containing a <data> element with the results of the query, which in this case is the entire running configuration and device state information (Figure 67).



```
<get/>
```

```
<data>
  <servers xmlns="http://example.com/ns/servers"/>
  <aaa xmlns="http://tail-f.com/ns/aaa/1.1">
    <authentication>
      <users>
        <user>
          <name>admin</name>
          <uid>9000</uid>
          <gid>0</gid>
          <password>$1$VfTQnsy$KrhKlDEwNDpgF2TNlfiId.</password>
          <ssh_keydir>/var/confd/homes/admin/.ssh</ssh_keydir>
          <homedir>/var/confd/homes/admin</homedir>
        </user>
        <user>
          <name>oper</name>
          <uid>9000</uid>
          <gid>0</gid>
          <password>$1$TYTZxz$60Xk0zmGeSHu1UwU/7Xfe0</password>
          <ssh_keydir>/var/confd/homes/oper/.ssh</ssh_keydir>
          <homedir>/var/confd/homes/oper</homedir>
        </user>
```

Figure 67: An example of the NETCONF get request and response

3. Use the scrollbars in the **Output XML** window panel to view the rest of the reply message in XML form. You can also select and copy selected text to the clipboard using the context menu ***Copy*** command. To view the retrieved results in form of a hierarchically arranged tree, containing nodes and their values, select the **Output Tree** tab in the central panel of the main window.

9.2 Retrieving Device State Data Using NETCONF get Operation

This section describes how to retrieve the state data information from a device that supports the `ietf-netconf-monitoring` YANG module.

1. In the **YANG Tree** window panel in the left portion of the main window, **select a state data node**, e.g., **`netconf-state`** node from the **`ietf-netconf-monitoring`** module.

2. Right-click the selected node to display the mouse context (pop-up) menu and select the ***get (execute)*** command from the context menu (Figure 68). Note that the get-config and the edit-config commands are disabled, because the selected node is a state data node.

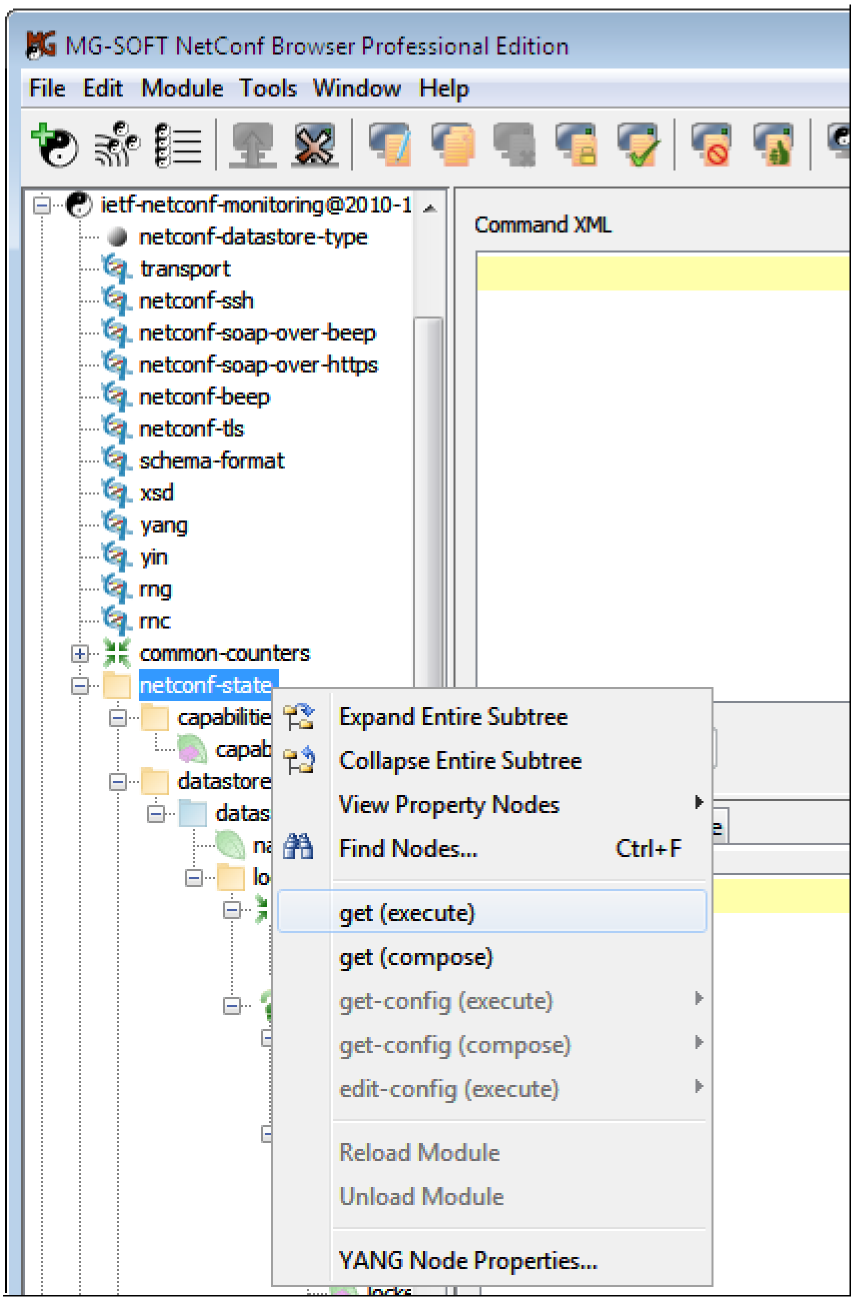
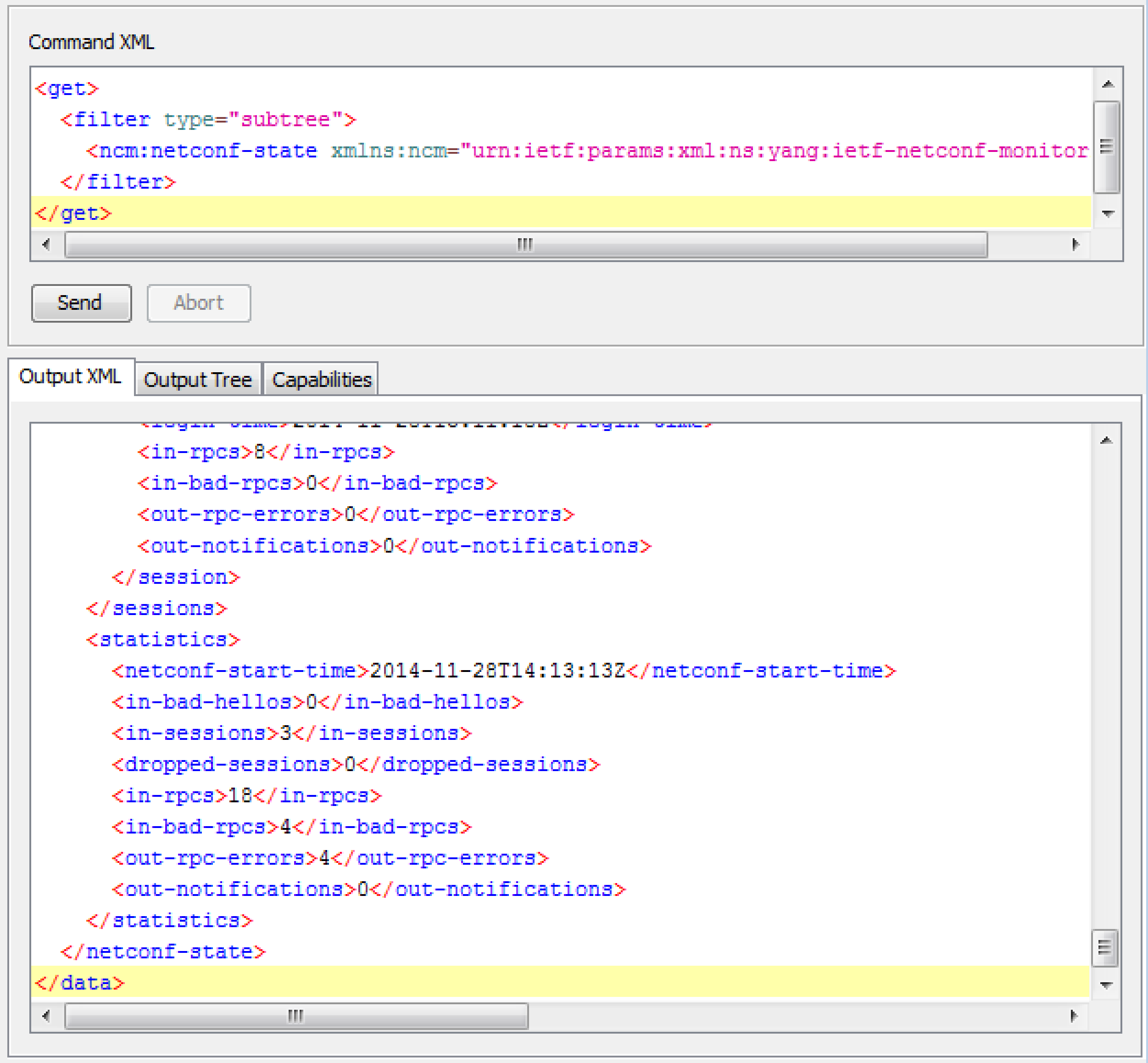


Figure 68: Executing the NETCONF get operation on a state data node

3. NetConf Browser creates and sends the NETCONF **get** request with the appropriate filter element to the server. In response, the server sends an <rpc-reply> message containing a <data> element with the results of the query, which in this case is all the state information from the **`netconf-state`** subtree (Figure 69).



```
<get>
  <filter type="subtree">
    <ncm:netconf-state xmlns:ncm="urn:ietf:params:xml:ns:yang:ietf-netconf-monitor
  </filter>
</get>
```

```
        <in-rpcs>8</in-rpcs>
        <in-bad-rpcs>0</in-bad-rpcs>
        <out-rpc-errors>0</out-rpc-errors>
        <out-notifications>0</out-notifications>
      </session>
    </sessions>
    <statistics>
      <netconf-start-time>2014-11-28T14:13:13Z</netconf-start-time>
      <in-bad-hellos>0</in-bad-hellos>
      <in-sessions>3</in-sessions>
      <dropped-sessions>0</dropped-sessions>
      <in-rpcs>18</in-rpcs>
      <in-bad-rpcs>4</in-bad-rpcs>
      <out-rpc-errors>4</out-rpc-errors>
      <out-notifications>0</out-notifications>
    </statistics>
  </netconf-state>
</data>
```

Figure 69: netconf-state information retrieved by a NETCONF get request

10 MODIFY CONFIGURATION IN REMOTE NETCONF SERVER

The NETCONF edit-config operation is used for changing the specified configuration.

Before using the NETCONF edit operations (**edit-config**, **commit**, **copy-config**), you need to determine which configuration datastore to use as the target by examining the **capabilities** advertised by the server while establishing NETCONF session. Typically, one would proceed as follows:

- ❑ If the server supports the candidate configuration datastore, the candidate configuration should be used as the target for edit operations. Changes to the candidate configuration will be later applied to the running configuration by means of the NETCONF **commit** or **copy-config** operation.
- ❑ If the server does not support the candidate configuration datastore and does support the ":writable-running" capability, then the running configuration should be used as the target for NETCONF editing operations.

This section describes both scenarios above.

Furthermore, since NETCONF servers usually support multiple concurrent sessions, the problem of concurrent write by different clients may occur. The configuration locking mechanism is used to deal with this problem. The **lock** operation allows the client to lock the configuration system of a device. Such locks are intended to be short-lived and allow a client to make a change without fear of interaction with other NETCONF clients, non-NETCONF clients (e.g., SNMP and command line interface (CLI) scripts), and human users. Once the configuration changes have been applied to the desired configuration, the configuration lock should be released using the **unlock** operation, so other clients (or other methods) can make changes to the configuration.

10.1 Modifying Running Configuration Directly

This section describes how to make changes to the currently active configuration (running) on the fly during runtime. The servers that support this type of configuration changes, advertise the **:writable-running** capability during the initiation of the NETCONF session.

This process typically includes the following steps:

1. lock <running/> database
2. edit <running/> database
3. unlock <running/> database

10.1.1 Lock the Running Configuration

In environments where more than one client can connect to a NETCONF server, it is recommend to lock the configuration before editing it. When a configuration is locked, only the client that acquired the lock is allowed to edit it.

1. In the main window, select the ***Tools / Manage Locks*** command from the main menu (Figure 70).

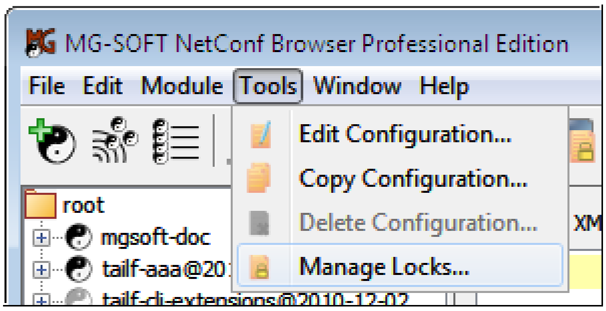
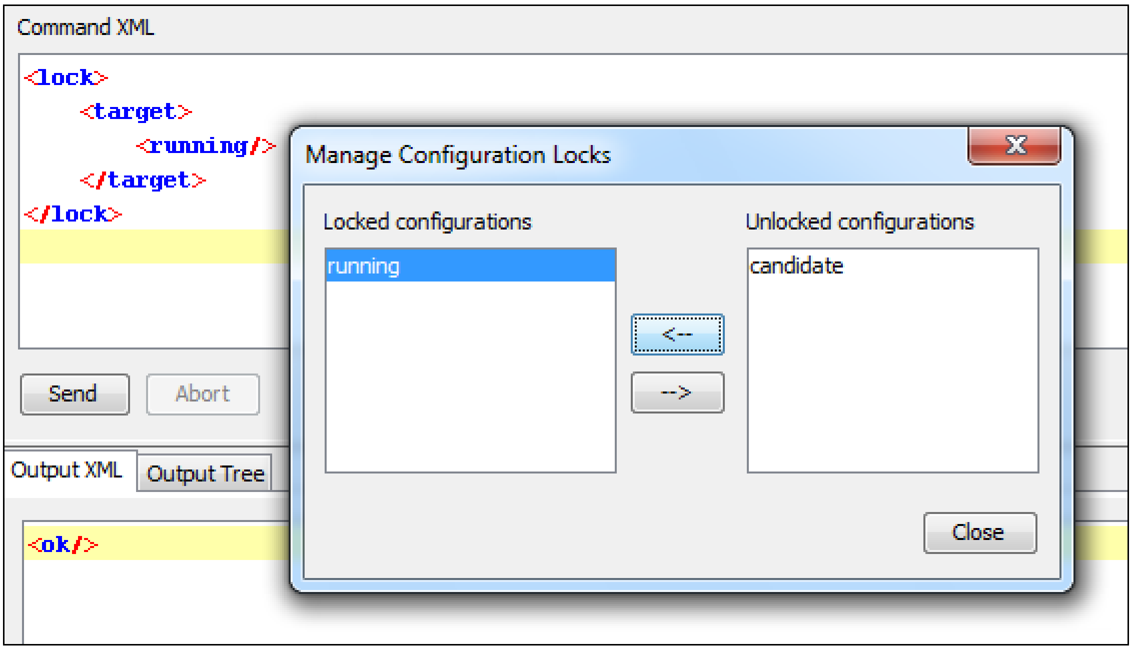

MG-SOFT NetConf Browser Professional Edition
File Edit Module Tools Window Help
Edit Configuration...
Copy Configuration...
Delete Configuration...
Manage Locks...
root
mgsoft-doc
tailf-aaa@201
tailf-cli-extensions@2010-12-02
XM

Figure 70: Selecting the Manage Locks command from the main menu

2. In the ***Manage Configuration Locks*** dialog box that appears (Figure 71), select the **`running`** configuration in the **Unlocked configurations** list and click the ***left-arrow button*** (<--) to move it to the **Locked configurations** list.



```
<lock>
    <target>
        <running/>
    </target>
</lock>
```

```
<ok/>
```

Figure 71: Locking the active (running) configuration

3. NetConf Browser creates and sends the NETCONF **lock** request to the server, attempting to lock the running configuration (see the **Command XML** panel in Figure 71). If the lock operation succeeds, the server responds with a reply message containing the <**ok>** element (see the **Output XML** panel in Figure 71). If the lock operation does not succeed, the server responds with the reply message containing the error description (i.e., configuration is already locked, etc.).

10.1.2 Edit the Running Configuration

This section describes how to use the NETCONF **edit-config** operation to configure network interfaces in the running configuration of a NETCONF server, which implements the `ietf-interfaces` and `ietf-ip` standard YANG modules.

This section explains how to perform the **edit-config** operation in NETCONF Content Editor window. For general instructions on using the NETCONF Content Editor window, refer to the Using NETCONF Content Editor section.

1. In the **YANG Tree** window panel in the left portion of the main window, **select the container or list node** that contains elements you would like to edit, e.g., **`interfaces`** node from the `ietf-interfaces` module.
2. Right-click the selected node to display the mouse context (pop-up) menu and select the ***edit-config (compose) / running*** command from the context menu (Figure 72).

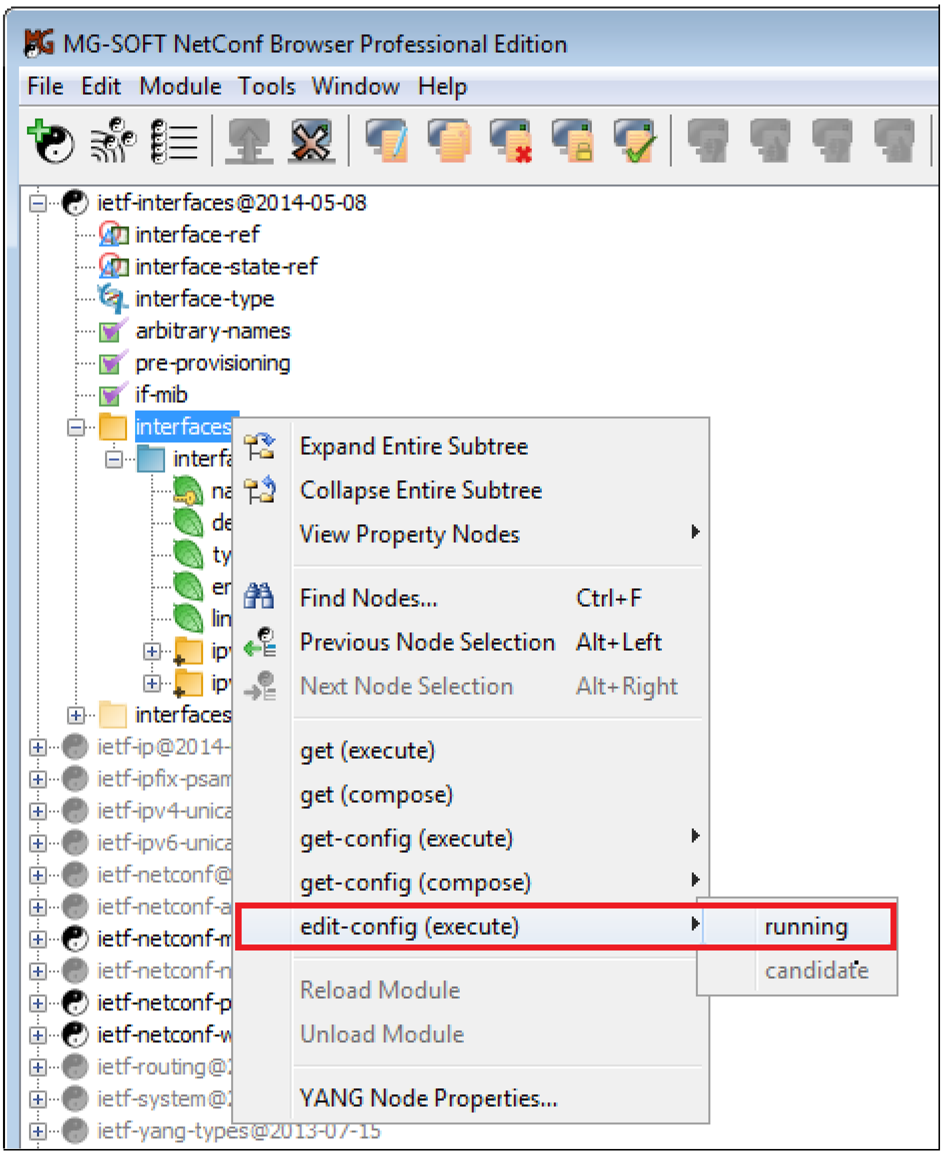
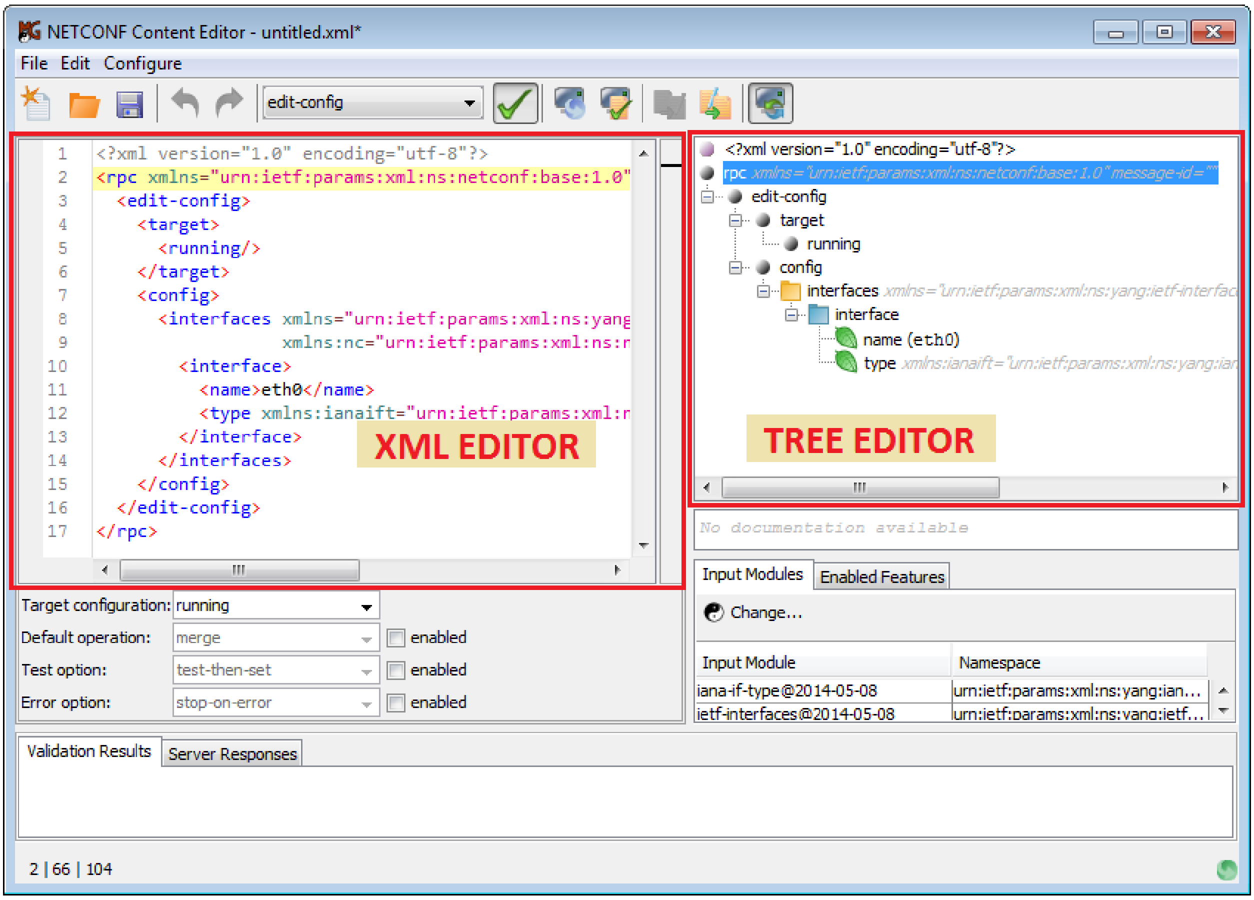


Figure 72: Choosing the *edit-config/running* command on a subtree node

3. NetConf Browser creates and sends a NETCONF **get-config** request to retrieve the configuration data of the selected subtree from the running configuration datastore (the retrieved configuration serves as a template for composing the **edit-config** request) and displays it in the NETCONF Content Editor window (Figure 73). Note that the **edit-config** content type option is automatically selected in the NETCONF Content Editor window.



```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang
                  xmlns:nc="urn:ietf:params:xml:ns:r
        <interface>
          <name>eth0</name>
          <type xmlns:ianaift="urn:ietf:params:xml:r
        </interface>
      </interfaces>
    </config>
  </edit-config>
</rpc>
```

```
<?xml version="1.0" encoding="utf-8"?>
rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id=""
  edit-config
    target
      running
    config
      interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfac
        interface
          name (eth0)
          type xmlns:ianaift="urn:ietf:params:xml:ns:yang:ian
```

Target configuration: running

Default operation: merge □ enabled

Test option: test-then-set □ enabled

Error option: stop-on-error □ enabled

Input Module	Namespace
iana-if-type@2014-05-08	urn:ietf:params:xml:ns:yang:ian...
ietf-interfaces@2014-05-08	urn:ietf:params:xml:ns:yang:ietf...

Figure 73: The NETCONF Content Editor window displaying the retrieved `interfaces` configuration subtree in both, textual and graphical manner

4. The NETCONF Content Editor window contains two main panels (Figure 73), which let you compose the content of the **edit-config** request in either textual or graphical manner, as follows:

- **NETCONF XML Editor** (upper left panel)

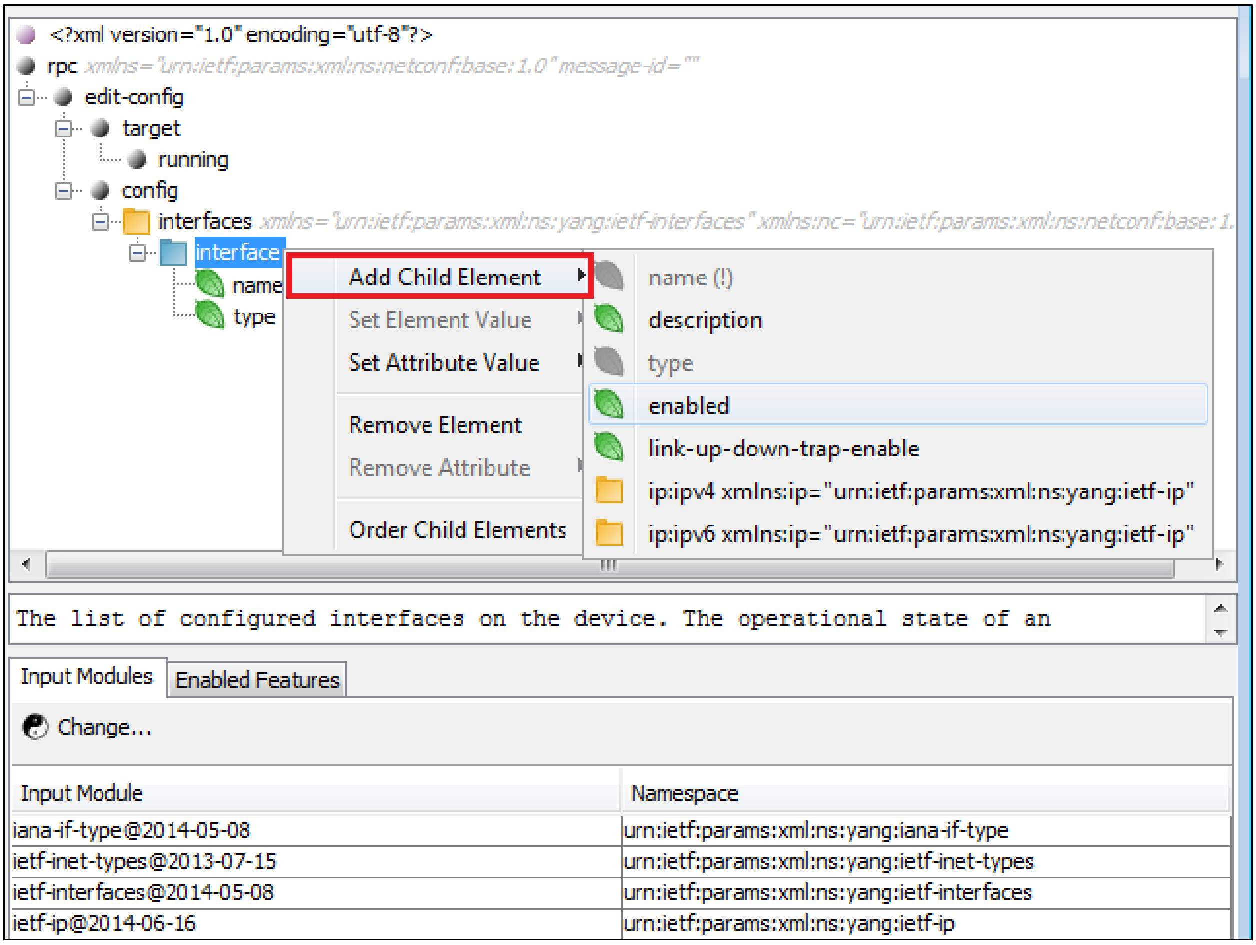
  Full-featured NETCONF XML document editor and validator that complies with the RFC 6110 specification. It features the XML syntax coloring and autocomplete feature. This editor lets you compose the contents of the **edit-config** request in a textual manner by writing the respective XML content. When you edit the XML document, its tree representation in the Tree Editor panel on the right side changes accordingly.

- **NETCONF Tree Editor** (upper right panel)

  It is used for viewing and composing the configuration tree for the edit-config operation. When you select the ***edit-config*** command in the main window, the NETCONF Content Editor window is opened and automatically populated with the configuration (sub)tree retrieved from the connected device by means of the get-config operation. You can edit the tree (change values, add nodes, delete nodes, etc.) to compose the edit-config RPC in a visual manner. While you edit the tree, the equivalent edit-config request in XML form is generated in the XML Editor on the left-hand side. You can switch between the Tree Editor and XML Editor at any time and further edit the content there.

> This section describes how to compose the edit-config RPC content in a visual manner by using the Tree Editor. For more information on using the XML Editor, please refer to the Using NETCONF Content Editor section.

5. To configure an existing interface (e.g., enable interface, configure IP address and netmask, etc.) using the **Tree Editor** panel, add additional nodes to the existing `interface` list node and set the values of these nodes as described in the following steps. If no interface is configured yet, you can configure one by adding the `interfaces` subtree to the `config` element, as depicted in Figure 73.
6. For example, to enable an interface, right-click the respective `interface` list node in the Tree Editor panel and select the ***Add Child Element / enabled*** command from the pop-up menu (Figure 74).



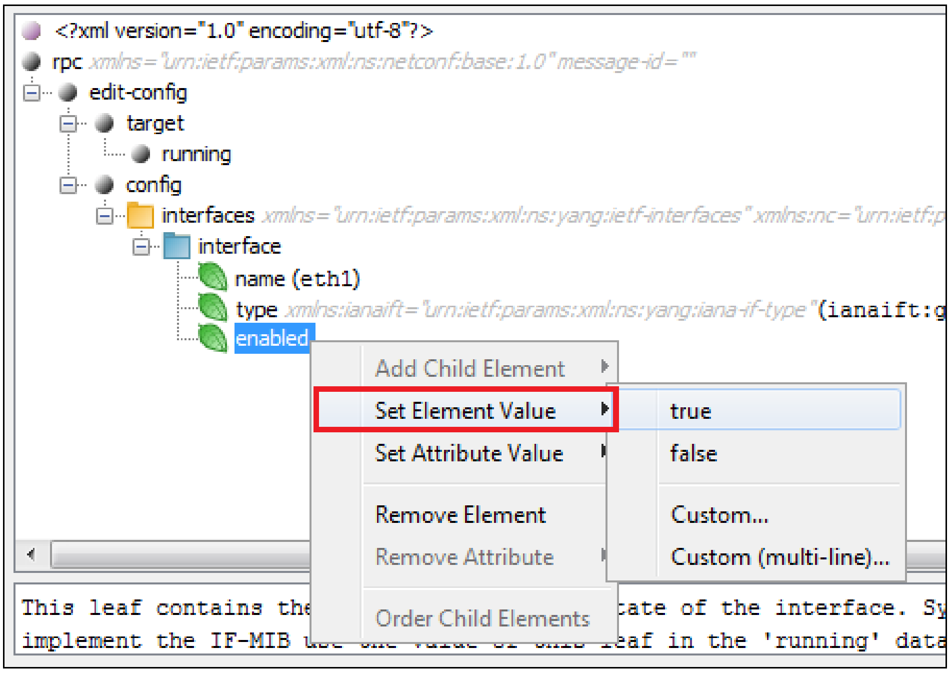
<?xml version="1.0" encoding="utf-8"?>
rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id=""
edit-config
target
running
config
interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.
interface
name
type
Add Child Element
Set Element Value
Set Attribute Value
Remove Element
Remove Attribute
Order Child Elements
name (!)
description
type
enabled
link-up-down-trap-enable
ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
ip:ipv6 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
Input Modules
Enabled Features
Change...

```
The list of configured interfaces on the device. The operational state of an
```

Input Module	Namespace
iana-if-type@2014-05-08	urn:ietf:params:xml:ns:yang:iana-if-type
ietf-inet-types@2013-07-15	urn:ietf:params:xml:ns:yang:ietf-inet-types
ietf-interfaces@2014-05-08	urn:ietf:params:xml:ns:yang:ietf-interfaces
ietf-ip@2014-06-16	urn:ietf:params:xml:ns:yang:ietf-ip

Figure 74: Adding a new element to the configuration tree

7. The `enabled` node is added as a child node to the `interface` list node in the Tree Editor. To set the value of this node to “true”, right-click it and select the ***Set Element Value / true*** command from the pop-up menu (Figure 75). The selected value (true) appears in brackets next to the `enabled` node.



<?xml version="1.0" encoding="utf-8"?>
rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id=""
edit-config
target
running
config
interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:p
interface
name (eth1)
type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type" (ianaift:g
enabled
Add Child Element
Set Element Value
Set Attribute Value
Remove Element
Remove Attribute
Order Child Elements
true
false
Custom...
Custom (multi-line)...
This leaf contains the
implement the IF-MIB
tate of the interface. Sy
eaf in the 'running' data

Figure 75: Setting the value of a leaf element in the configuration tree

8. For example, to configure an IPv4 address of an interface, right-click the respective `interface` list node in the Tree Editor panel and select the ***Add Child Element / ip:ipv4*** command from the pop-up menu (Figure 76).

Note 1: The **`ietf-ip`** module must be selected in the **Input Modules** list below the Tree Editor panel to enable adding `IPv4` elements defined in the **`ietf-ip`** YANG module.

Note 2: By default, the **Automatically Adapt Input Modules and Features to Session** toggle button is enabled (pressed) in the NETCONF Content Editor window, meaning that the set of **input modules** and **enabled features** is automatically adapted to the capabilities advertised by the currently connected NETCONF server (if any). Note that the content validation DSDL schemas in the NETCONF Content Editor window are generated from the set of enabled input modules and features.

To enable an input module or a feature not advertised by the server, click the **Automatically Adapt Input Modules and Features to Session** toggle button to disable this feature. Then, use the ***Configure / Input Modules*** or the ***Configure / Enabled Features*** command and select the desired YANG module(s) or feature(s) to refine the schemas for validating the NETCONF content. The list of available input modules and features that you can choose from is taken from the list of YANG modules loaded in the main window.

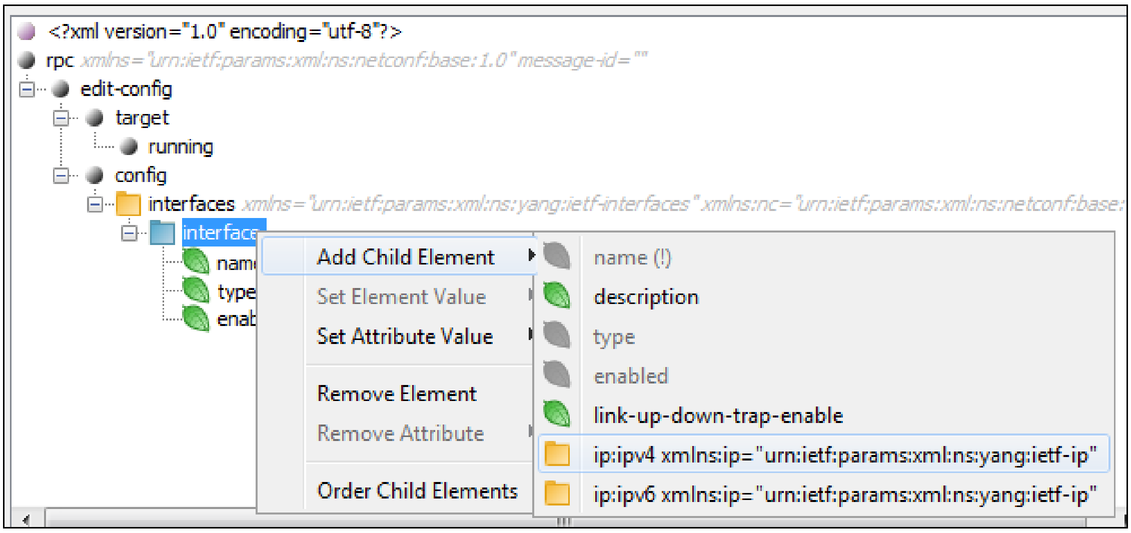

<?xml version="1.0" encoding="utf-8"?>
rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id=""
edit-config
target
running
config
interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:
interface
nam
type
enab
Add Child Element
Set Element Value
Set Attribute Value
Remove Element
Remove Attribute
Order Child Elements
name (!)
description
type
enabled
link-up-down-trap-enable
ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
ip:ipv6 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"

Figure 76: Adding a *ip:ipv4* element to the configuration tree

9. The `ip:ipv4` node is added as a child node to the `interface` list node in the Tree Editor. Right-click it and select the ***Add Child Element / ip:address*** command from the pop-up menu (Figure 77).

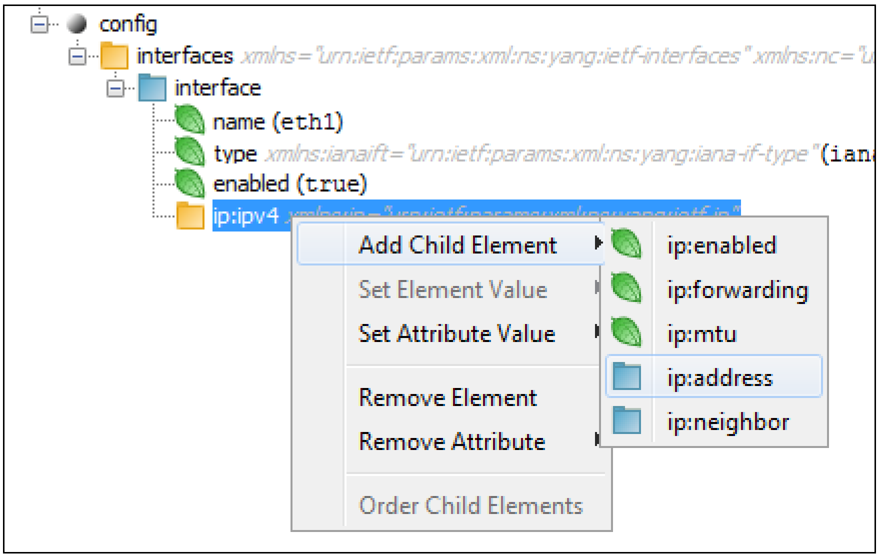

config
interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="u
interface
name (eth1)
type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type" (ian
enabled (true)
ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
Add Child Element
Set Element Value
Set Attribute Value
Remove Element
Remove Attribute
Order Child Elements
ip:enabled
ip:forwarding
ip:mtu
ip:address
ip:neighbor

Figure 77: Adding a *ip:address* list element to the configuration tree

10. The `ip:address` list node is added as a child node to the `interface` list node in the Tree Editor (Figure 78). Notice the error symbol (!) on the `ip:address` icon indicating a validation error.

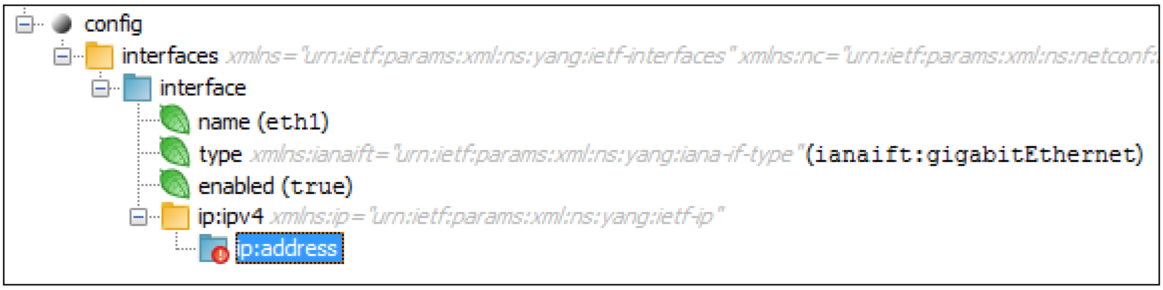
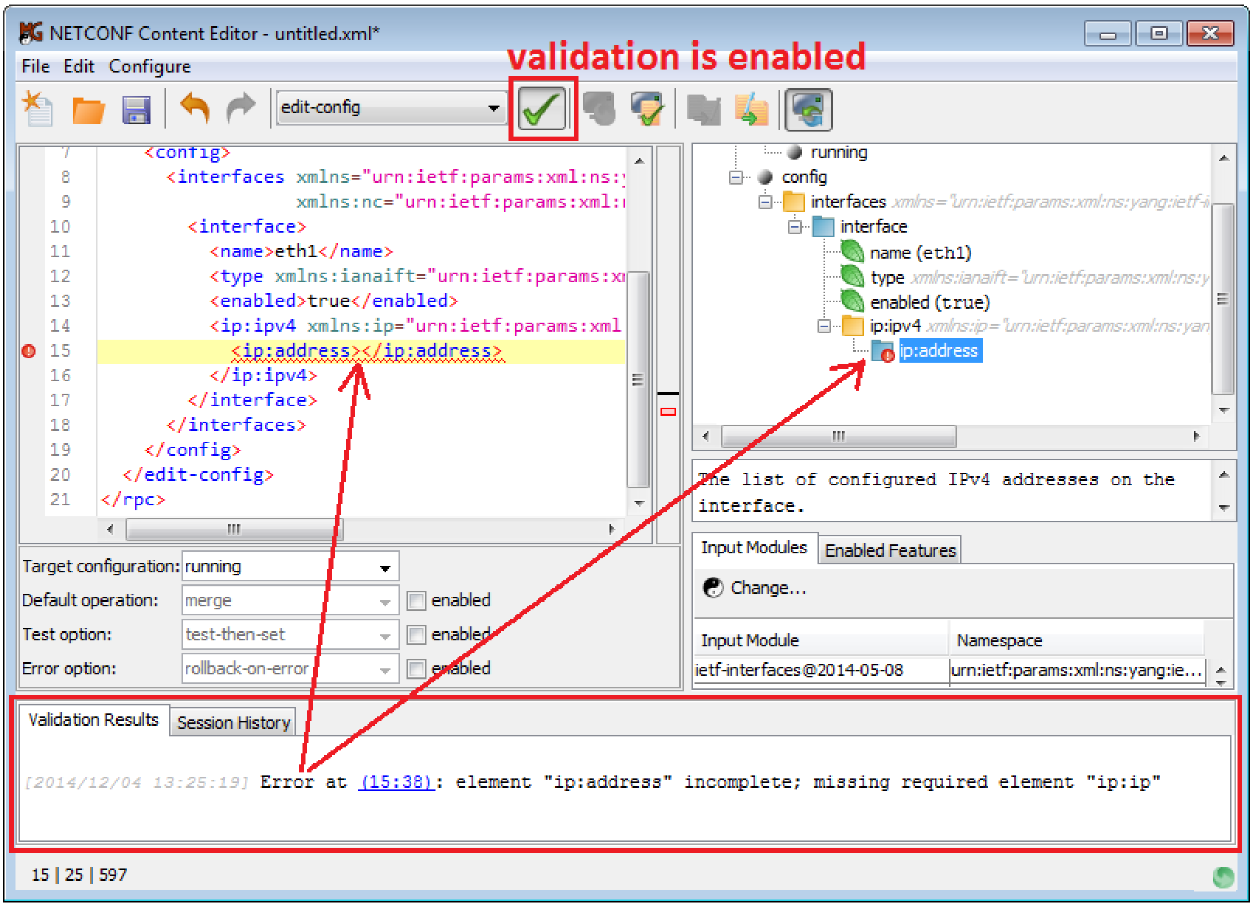

config
interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:params:xml:ns:netconf:
interface
name (eth1)
type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type" (ianaift:gigabitEthernet)
enabled (true)
ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
ip:address

Figure 78: New *ip:address* list element in the configuration tree

11. By default, the real-time content validation (as specified in RFC 6110) is enabled in the NETCONF Content Editor window (Figure 79), meaning that the software constantly checks if the content is syntactically and semantically correct according to the Document Schema Definition Languages (DSDL) schemas, which are automatically generated from selected YANG modules and features. In other words, NetConf Browser automatically checks the validity of every change you make in the document, either in XML Editor or in the Tree Editor. If any inconsistency is detected, the corresponding error or warning message appears in the **Validation Results** tab at the bottom of the window (Figure 79). The software also underlines the erroneous elements in the XML Editor and marks all nodes in the Tree Editor and all lines in XML Editor that are the source of the validation error/warning with the corresponding error (❗) or warning (⚠) overlay symbol. **It is highly recommended to fix all inconsistencies reported by the error and warning messages before sending the document as edit-config request to a NETCONF server.**

Tip: You can disable validation by clicking the ***Validation*** (✓) toggle button in the toolbar.



```
 7     <config>
 8       <interfaces xmlns="urn:ietf:params:xml:ns:y
 9                   xmlns:nc="urn:ietf:params:xml:n
10         <interface>
11           <name>eth1</name>
12           <type xmlns:ianaift="urn:ietf:params:xm
13           <enabled>true</enabled>
14           <ip:ipv4 xmlns:ip="urn:ietf:params:xml
15             <ip:address></ip:address>
16           </ip:ipv4>
17         </interface>
18       </interfaces>
19     </config>
20   </edit-config>
21 </rpc>
```

```
running
config
  interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-i
    interface
      name (eth1)
      type xmlns:ianaift="urn:ietf:params:xml:ns:y
      enabled (true)
      ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yan
        ip:address
```

The list of configured IPv4 addresses on the interface.

Target configuration: running
Default operation: merge ☐ enabled
Test option: test-then-set ☐ enabled
Error option: rollback-on-error ☐ enabled

Input Module	Namespace
ietf-interfaces@2014-05-08	urn:ietf:params:xml:ns:yang:ie...

```
[2014/12/04 13:25:19] Error at (15:38): element "ip:address" incomplete; missing required element "ip:ip"
```

15 | 25 | 597

Figure 79: An example of a validation error indicating a missing mandatory element (ip:ip)

In example above the error message states that the `ip:address` element is incomplete because the mandatory element `ip:ip` is missing. To fix the error, we need to add the missing element and set its value. To start doing this in the visual editor, right-click the `ip:address` node and select the ***Add Child Element / ip:ip*** command from the pop-up menu (Figure 80).

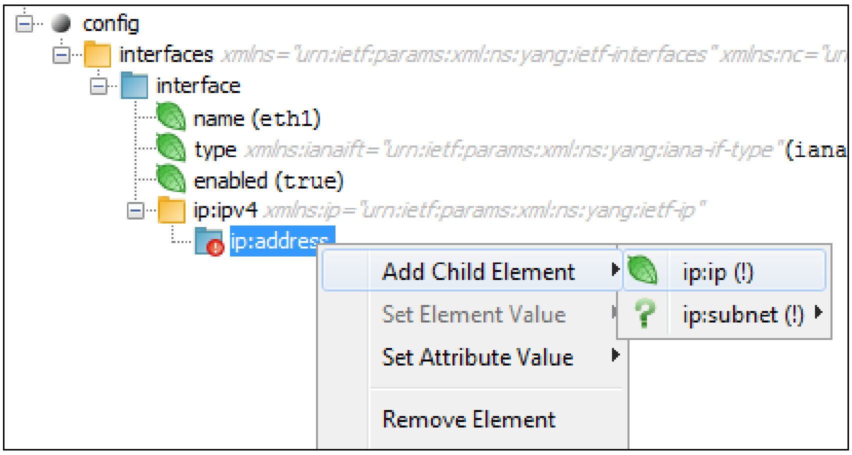

config
interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="ur
interface
name (eth1)
type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type" (iana
enabled (true)
ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
ip:address
Add Child Element
Set Element Value
Set Attribute Value
Remove Element
ip:ip (!)
ip:subnet (!)

Figure 80: Adding an *ip:ip* mandatory leaf element to the configuration tree

12. The `ip:ip` leaf node is added as a child node to the `ip:address` list node in the Tree Editor. Right-click it and select the ***Set Element Value / custom*** command from the pop-up menu (Figure 81).

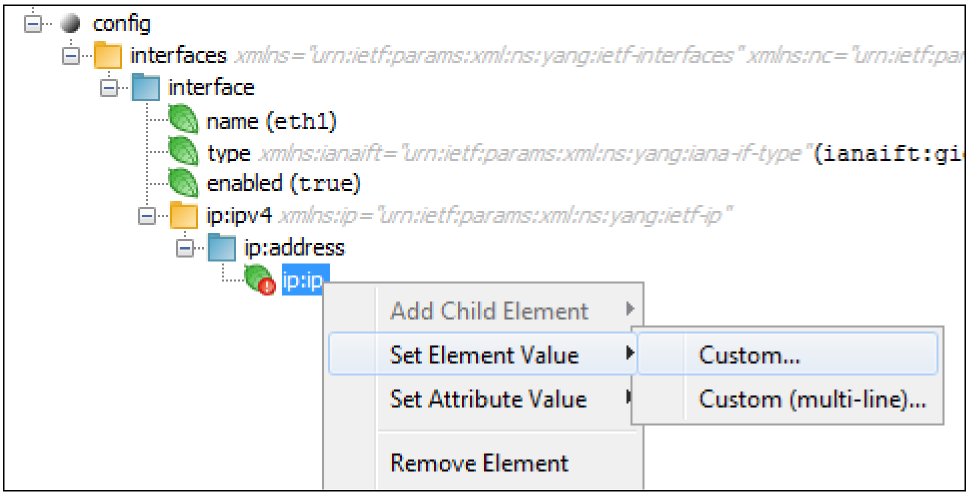
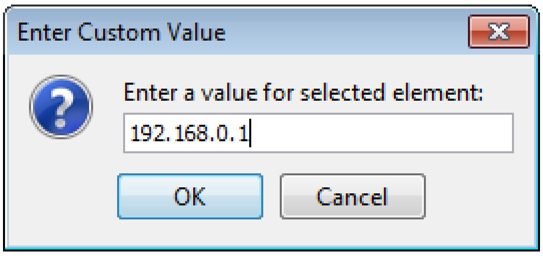

config
interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:par
interface
name (eth1)
type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type" (ianaift:gi
enabled (true)
ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
ip:address
ip:ip
Add Child Element
Set Element Value
Set Attribute Value
Remove Element
Custom...
Custom (multi-line)...

Figure 81: Setting the value of an *ip:ip* leaf element

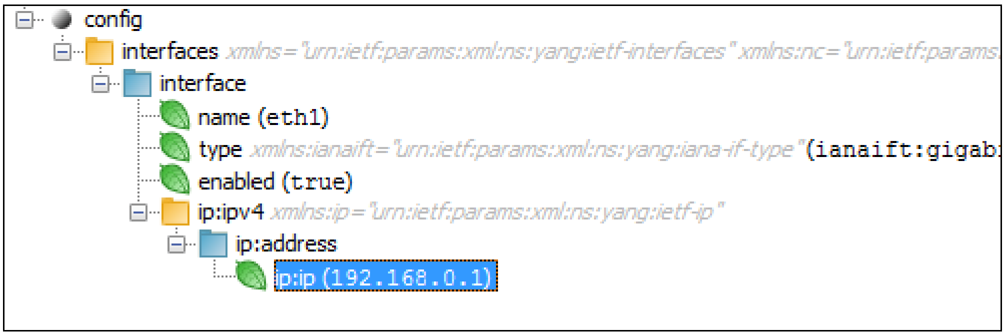
13. The Enter Custom Value dialog box appears (Figure 82). Enter the IP address of the respective interface into the input line and click the ***OK*** button to set the value.



Enter a value for selected element:
192.168.0.1

Figure 82: Setting the custom value (IP address) of an *ip:ip* leaf element

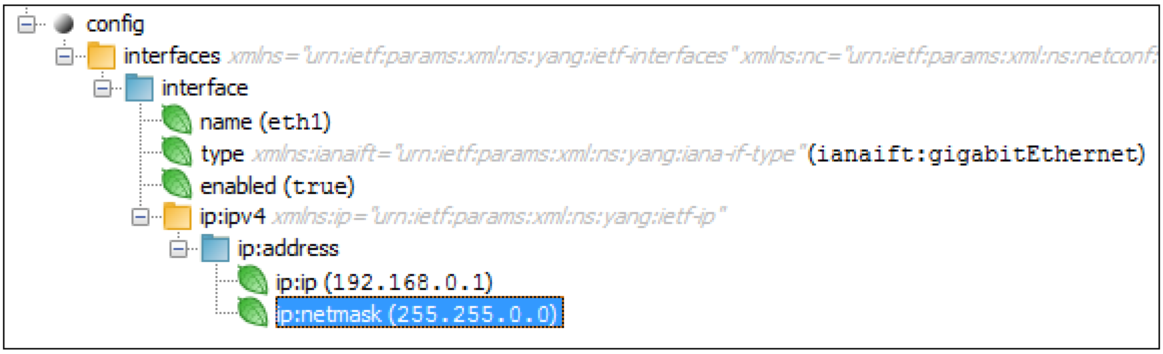
14. The entered value (IP address) appears in brackets next to the `ip:ip` leaf node (Figure 83). Notice that once you set a proper value (IP address) the validation error and the corresponding error symbol disappears from the `ip:ip` leaf node.



```
config
  interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:params
    interface
      name (eth1)
      type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type"(ianaift:gigab
      enabled (true)
      ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
        ip:address
          ip:ip (192.168.0.1)
```

Figure 83: Example of an interface configuration tree with a configured IP address

15. In the same manner, configure the netmask of the interface by adding the mandatory (!) `ip:netmask` leaf element and setting its value (Figure 84).

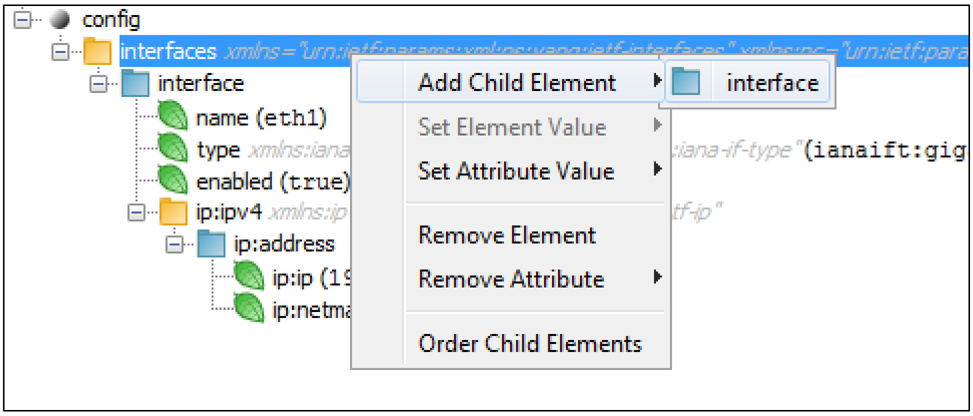


```
config
  interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:params:xml:ns:netconf:
    interface
      name (eth1)
      type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type"(ianaift:gigabitEthernet)
      enabled (true)
      ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
        ip:address
          ip:ip (192.168.0.1)
          ip:netmask (255.255.0.0)
```

Figure 84: Example of an interface configuration tree with a configured IP address and netmask

16. Use the procedure above to add the optional elements of the interface and IPv4 subtree to the configuration tree (e.g., `description`, `ip.mtu`, `ip.forwarding`, etc.) and set their properties.

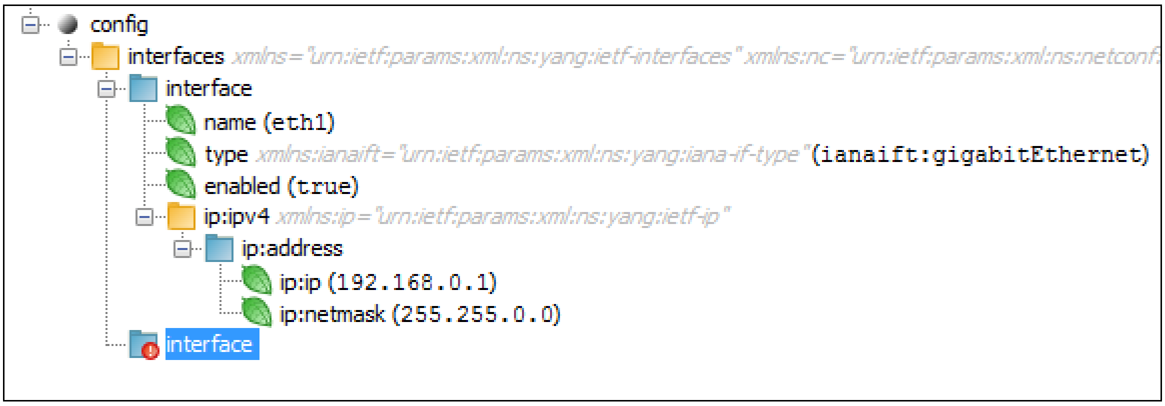
17. To configure an additional interface and its properties in the Tree Editor panel, right-click the `interfaces` container node and choose the ***Add Child Element / interface*** command from the pop-up menu (Figure 85).



```
config
  interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:para
    interface
      name (eth1)
      type xmlns:iana                    :iana-if-type"(ianaift:gig
      enabled (true)
      ip:ipv4 xmlns:ip                   tf-ip"
        ip:address
          ip:ip (19
          ip:netma
```

Figure 85: Adding a new interface element to the configuration tree

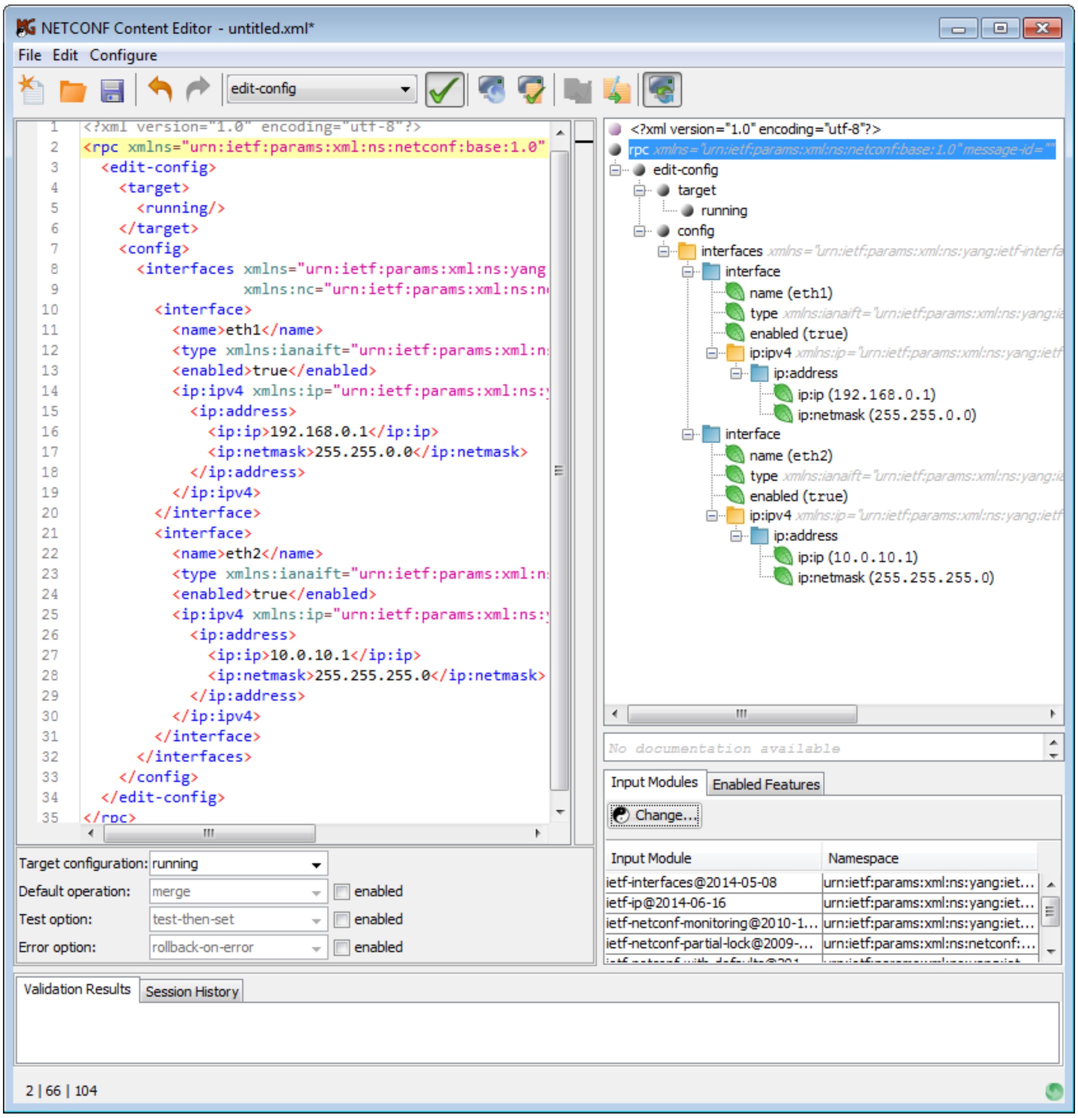
18. A new `interface` node (representing a new network interface) is added as a child node to the `interfaces` container node in the Tree Editor (Figure 86).



```
config
  interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces" xmlns:nc="urn:ietf:params:xml:ns:netconf
    interface
      name (eth1)
      type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type" (ianaift:gigabitEthernet)
      enabled (true)
      ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
        ip:address
          ip:ip (192.168.0.1)
          ip:netmask (255.255.0.0)
    interface
```

Figure 86: A new interface element in the configuration tree

19. Add the child elements to the new `interface` node (i.e., `name`, `type`, `enabled`, `ip:ipv4`, etc.) and set their values as described in previous steps to produce the content as shown in Figure 87.



```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang
                  xmlns:nc="urn:ietf:params:xml:ns:n
        <interface>
          <name>eth1</name>
          <type xmlns:ianaift="urn:ietf:params:xml:n
          <enabled>true</enabled>
          <ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:
            <ip:address>
              <ip:ip>192.168.0.1</ip:ip>
              <ip:netmask>255.255.0.0</ip:netmask>
            </ip:address>
          </ip:ipv4>
        </interface>
        <interface>
          <name>eth2</name>
          <type xmlns:ianaift="urn:ietf:params:xml:n
          <enabled>true</enabled>
          <ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:
            <ip:address>
              <ip:ip>10.0.10.1</ip:ip>
              <ip:netmask>255.255.255.0</ip:netmask>
            </ip:address>
          </ip:ipv4>
        </interface>
      </interfaces>
    </config>
  </edit-config>
</rpc>
```

```
<?xml version="1.0" encoding="utf-8"?>
rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id=""
  edit-config
    target
      running
    config
      interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfa
        interface
          name (eth1)
          type xmlns:ianaift="urn:ietf:params:xml:ns:yang:ia
          enabled (true)
          ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf
            ip:address
              ip:ip (192.168.0.1)
              ip:netmask (255.255.0.0)
        interface
          name (eth2)
          type xmlns:ianaift="urn:ietf:params:xml:ns:yang:ia
          enabled (true)
          ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf
            ip:address
              ip:ip (10.0.10.1)
              ip:netmask (255.255.255.0)
```

No documentation available

Target configuration: running

Default operation: merge □ enabled

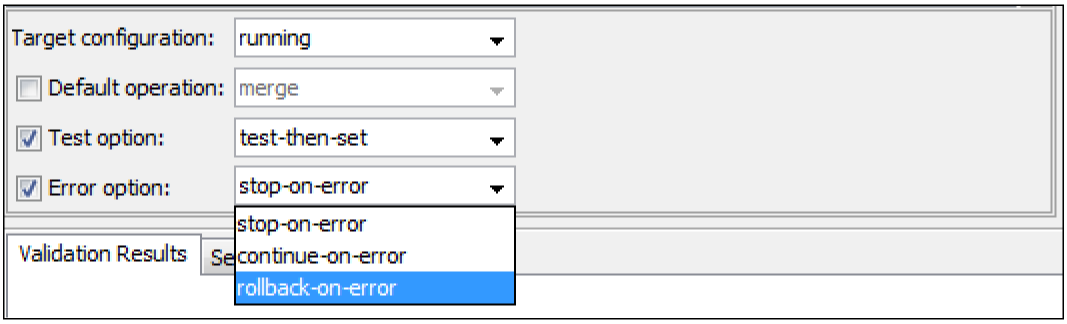
Test option: test-then-set □ enabled

Error option: rollback-on-error □ enabled

Input Module	Namespace
ietf-interfaces@2014-05-08	urn:ietf:params:xml:ns:yang:iet...
ietf-ip@2014-06-16	urn:ietf:params:xml:ns:yang:iet...
ietf-netconf-monitoring@2010-1...	urn:ietf:params:xml:ns:yang:iet...
ietf-netconf-partial-lock@2009-...	urn:ietf:params:xml:ns:netconf:...

Figure 87: An example of edit-config message content presented in textual (left panel) and graphical manner (right-panel)

20. After you have finished modeling the configuration tree for the edit-config request, specify the settings for the **edit-config** operation in the Quick options panel under the XML Editor (Figure 88), as follows:
    - In the ***Target configuration*** drop-down list, select the ***running*** entry to make the change directly to the currently active configuration.
    - Leave the ***Default operation*** checkbox unchecked, since merge is already the default operation and we will be merging the old configuration with the new one.
    - If the server supports the **:validate** capability, check the ***Test option*** checkbox and select the ***test-than-set*** option from the accompanying drop-down list. This way, the server will perform a validation test before attempting to set the configuration. If any validation error occurs, the edit-config operation will not be performed.
    - Check the ***Error option*** checkbox and select the ***rollback-on-error*** option if available (it depends on the **:rollback-on-error** capability) from the drop-down list. If this option is selected, the server will restore the configuration to the previous state if an error occurs while performing the edit-config operation. If this option is not available, leave the ***Error option*** checkbox unchecked (this aborts the edit-config operation on first error – if any).



Target configuration: running

□ Default operation: merge

☑ Test option: test-then-set

☑ Error option: stop-on-error

Figure 88: Setting the quick options for edit-config operation

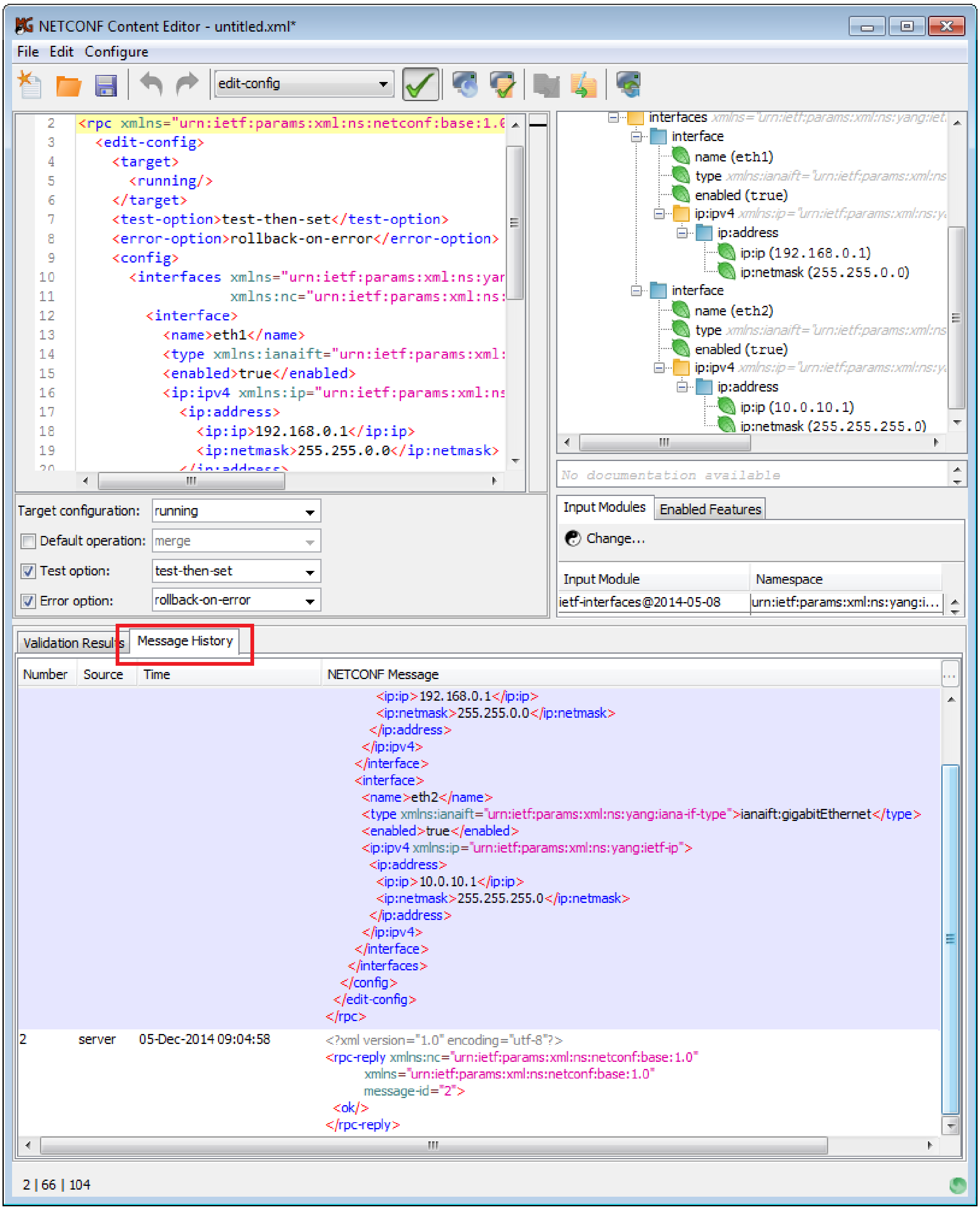
21. After you have configured the quick options for the edit-config operation, click the ***Send as RPC*** ( ) button in the toolbar to send the edit-config request to the server.

22. The **Message History** tab at the bottom of the NETCONF Content Editor window becomes active and displays the actual **edit-config** RPC message sent to the server and the corresponding RPC reply received from the server Figure 89).

23. The NETCONF server will attempt to perform the configuration change according to your settings. If the **edit-config** operation succeeds, the server will respond with a reply message containing the **<ok>** element (see the Message History list in Figure 89). If the edit-config operation fails, the server will respond with the reply message containing the error description.

> The operation status icon in the right section of the status bar in the NETCONF Content Editor window indicates whether the operation succeeded successfully (🟢) or resulted in error (🔴).
>
> See also the **Output XML** panel, the **Log** tab and **Session History** tab in the main window for the server response.

24. To save the entire content of the XML Editor (i.e., edit-config RPC) to a file for future use, select the ***File / Save*** command in the NETCONF Content Editor window and in the Save dialog box specify the location and name of the resulting XML file. You can later load the XML file back into the NETCONF Content Editor window by using the ***File / Load*** command.



```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0
  <edit-config>
    <target>
      <running/>
    </target>
    <test-option>test-then-set</test-option>
    <error-option>rollback-on-error</error-option>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:ns:yar
                  xmlns:nc="urn:ietf:params:xml:ns:
        <interface>
          <name>eth1</name>
          <type xmlns:ianaift="urn:ietf:params:xml:
          <enabled>true</enabled>
          <ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns
            <ip:address>
              <ip:ip>192.168.0.1</ip:ip>
              <ip:netmask>255.255.0.0</ip:netmask>
```

```
interfaces xmlns="urn:ietf:params:xml:ns:yang:iet
  interface
    name (eth1)
    type xmlns:ianaift="urn:ietf:params:xml:ns
    enabled (true)
    ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:y
      ip:address
        ip:ip (192.168.0.1)
        ip:netmask (255.255.0.0)
  interface
    name (eth2)
    type xmlns:ianaift="urn:ietf:params:xml:ns
    enabled (true)
    ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:y
      ip:address
        ip:ip (10.0.10.1)
        ip:netmask (255.255.255.0)
```

No documentation available

Target configuration: running

☐ Default operation: merge

☑ Test option: test-then-set

☑ Error option: rollback-on-error

Input Module	Namespace
ietf-interfaces@2014-05-08	urn:ietf:params:xml:ns:yang:i...

Number	Source	Time	NETCONF Message
2	server	05-Dec-2014 09:04:58	

```
            <ip:ip>192.168.0.1</ip:ip>
            <ip:netmask>255.255.0.0</ip:netmask>
          </ip:address>
        </ip:ipv4>
      </interface>
      <interface>
        <name>eth2</name>
        <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">ianaift:gigabitEthernet</type>
        <enabled>true</enabled>
        <ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip">
          <ip:address>
            <ip:ip>10.0.10.1</ip:ip>
            <ip:netmask>255.255.255.0</ip:netmask>
          </ip:address>
        </ip:ipv4>
      </interface>
    </interfaces>
  </config>
 </edit-config>
</rpc>
```

```
<?xml version="1.0" encoding="utf-8"?>
<rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
      xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
      message-id="2">
 <ok/>
</rpc-reply>
```

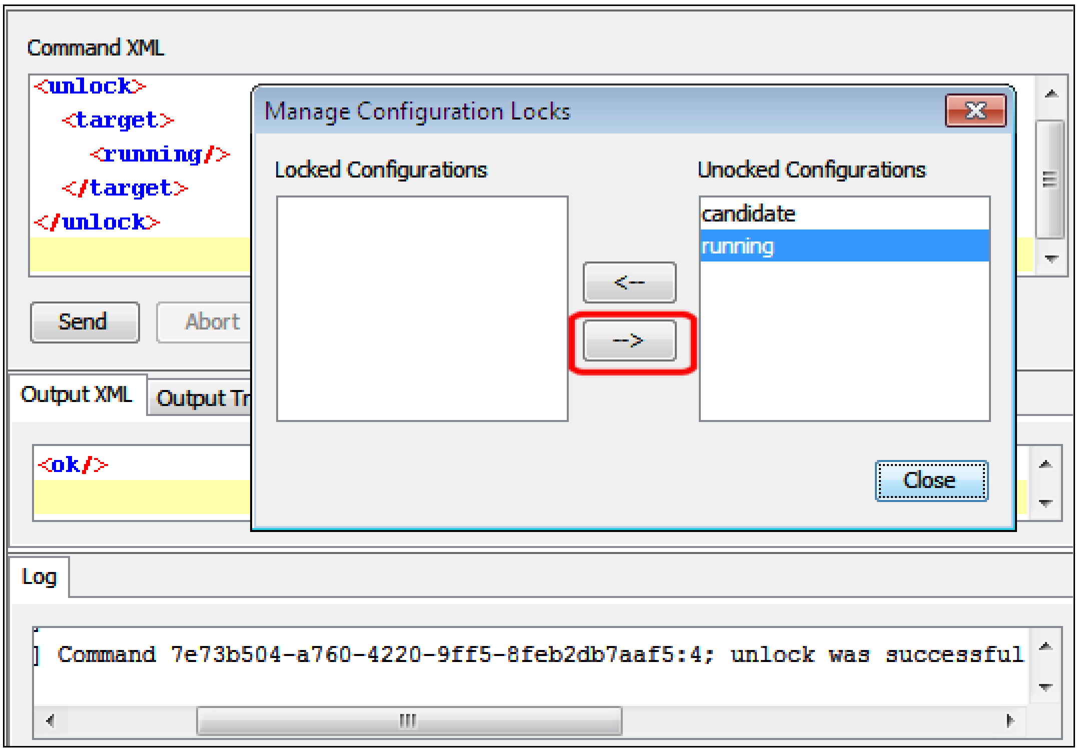
2 | 66 | 104

Figure 89: Viewing the edit-config request and reply messages exchanged with the server

10.1.3 Unlock the Running Configuration

After you have made the change to the running configuration, you should unlock it to allow for other NETCONF clients and methods (e.g., SNMP, CLI, …) to access it in write mode.

1. In the main window, select the ***Tools / Manage Locks*** command from the main menu (Figure 70).
2. In the ***Manage Configuration Locks*** dialog box that appears (Figure 90), select the **`running`** configuration in the **Locked configurations** list and click the ***right-arrow button*** to move it to the **Unlocked configurations** list (Figure 90).



```
<unlock>
  <target>
    <running/>
  </target>
</unlock>
```

```
<ok/>
```

```
] Command 7e73b504-a760-4220-9ff5-8feb2db7aaf5:4; unlock was successful
```

Figure 90: Unlocking the active (running) configuration

3. NetConf Browser creates and sends the NETCONF **unlock** request to the server, attempting to unlock the running configuration (see the Command XML panel in Figure 90). If the lock operation succeeds, the server responds with a reply message containing the **<ok>** element (see the Output XML panel in Figure 90).

10.2 Modifying Candidate Configuration and Committing Changes

If the **:candidate** capability is supported and advertised by the server, it means that it supports the (conceptual) candidate configuration datastore. The candidate configuration is a full configuration data set that serves as a work place for creating and manipulating configuration data without impacting the running configuration. Additions, deletions, and changes can be made to this data to construct the desired configuration data.

Unlike the changes made to the running configuration, any changes made to the candidate configuration do not take effect immediately within the network device.

When ready, the client can use the **commit** operation to activate the changes embodied in the candidate database, and make them part of the running configuration.

This section describes how to modify the candidate configuration and then apply the changes to the active configuration. This process typically includes the following steps:

1. lock <running/> database
2. lock <candidate/> database
3. edit <candidate/> database
4. commit <candidate/> database
5. unlock <candidate/> database
6. unlock <running/> database

10.2.1 Lock the Running and Candidate Configuration

In environments where more than one client can connect to a NETCONF server, it is recommend to lock the configuration before editing it. Before starting to edit the candidate configuration, you should lock the running and candidate configuration datastores.

When a configuration is locked, only the client that has acquired the lock is allowed to modify it.

1. In the main window, select the ***Tools / Manage Locks*** command from the main menu (Figure 91).

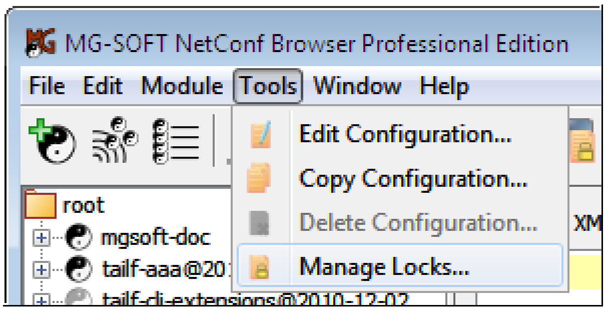
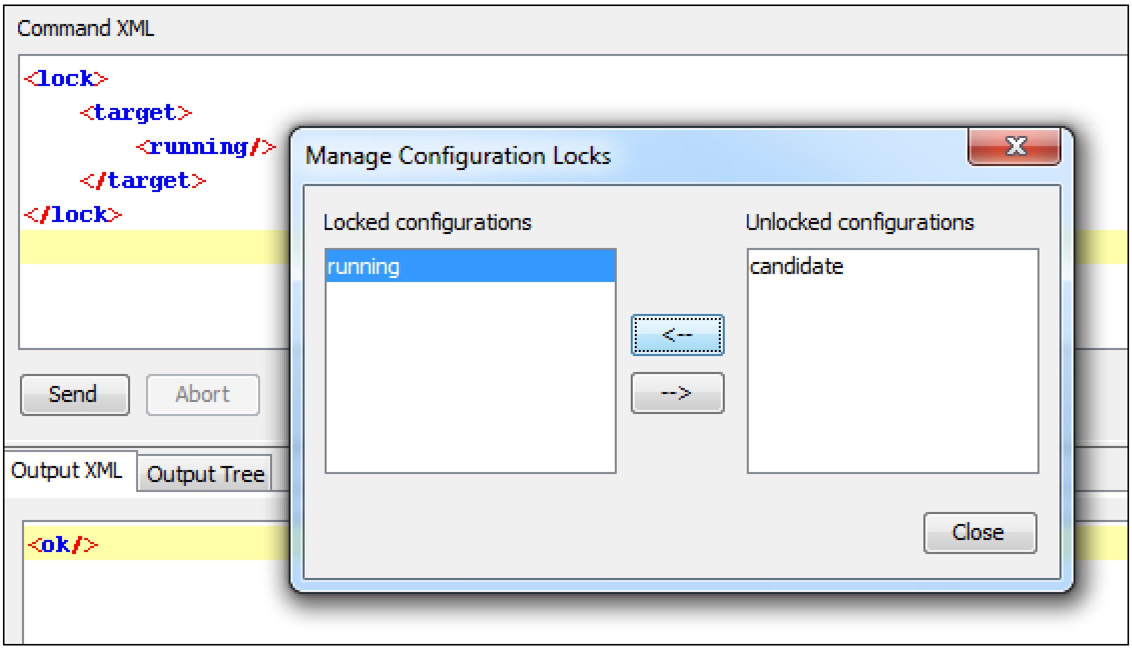


Figure 91: Selecting the Manage Locks command from the main menu

2. In the ***Manage Configuration Locks*** dialog box that appears (Figure 92), select the `running` configuration in the **Unlocked configurations** list and click the ***left-arrow button*** (<-) to move it to the **Locked configurations** list.



```
<lock>
    <target>
        <running/>
    </target>
</lock>
```

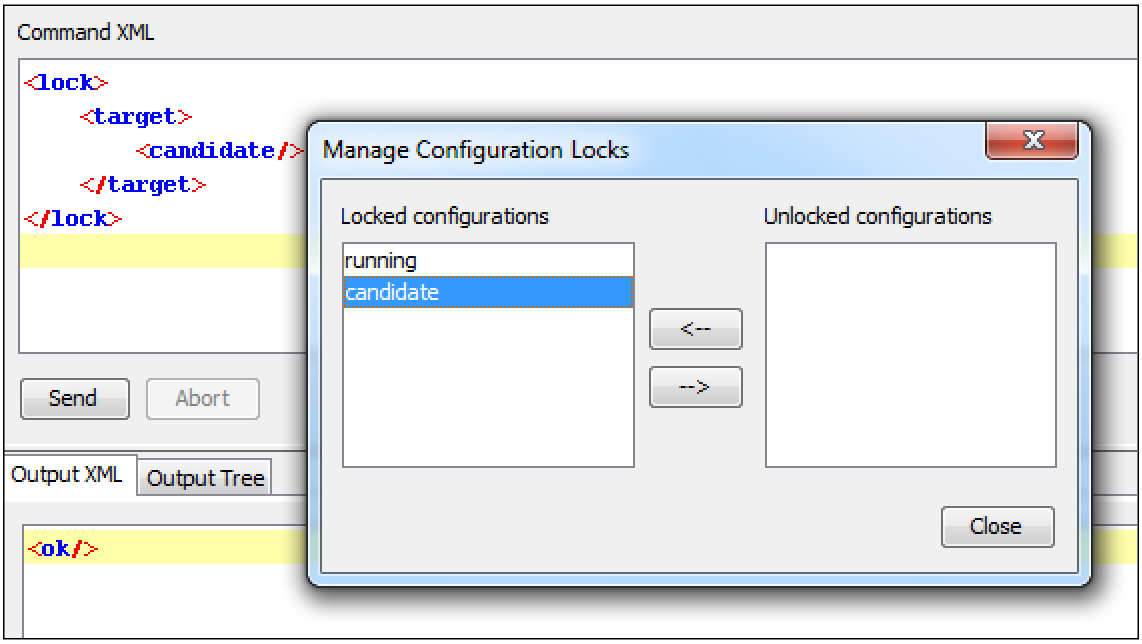
```
<ok/>
```

Figure 92: Locking the active (running) configuration

3. NetConf Browser creates and sends the NETCONF **lock** request to the server, attempting to lock the running configuration (see the **Command XML** panel in Figure 92). If the lock operation succeeds, the server responds with a reply message containing the **<ok>** element (see the **Output XML** panel in Figure 92). If the lock operation does not succeed, the server respond with the reply message containing the error description (i.e., configuration is already locked, etc.).

4. In the ***Manage Configuration Locks*** dialog box, select the **`candidate`** configuration in the **Unlocked configurations** list and click the ***left-arrow button*** (<--) to move it to the **Locked configurations** list (Figure 93).

5. NetConf Browser creates and sends the NETCONF **lock** request to the server, attempting to lock the candidate configuration. If the lock operation succeeds, the server responds with a reply message containing the **<ok>** element (see the **Output XML** panel in Figure 93). If the lock operation does not succeed, the server responds with the reply message containing the error description (i.e., configuration is already locked, etc.).



```
<lock>
    <target>
        <candidate/>
    </target>
</lock>
```

```
<ok/>
```

Figure 93: Locking also the candidate configuration

6. Click the ***Close*** button at the bottom of the ***Manage Configuration Locks*** dialog box to close it.

10.2.2 Edit the Candidate Configuration

This section describes how to use the NETCONF **edit-config** operation to configure network interfaces in the **candidate** configuration of a NETCONF server, which implements the `ietf-interfaces` and `ietf-ip` standard YANG modules.

This section explains how to perform the **edit-config** operation in NETCONF Content Editor window. For general instructions on using the NETCONF Content Editor window, refer to the Using NETCONF Content Editor section.

1. In the **YANG Tree** window panel in the left portion of the main window, **select the container or list node** that contains elements you would like to edit, e.g., **`interfaces`** node from the `ietf-interfaces` module.
2. Right-click the selected node to display the mouse context (pop-up) menu and select the ***edit-config (compose) / candidate*** command from the context menu ().

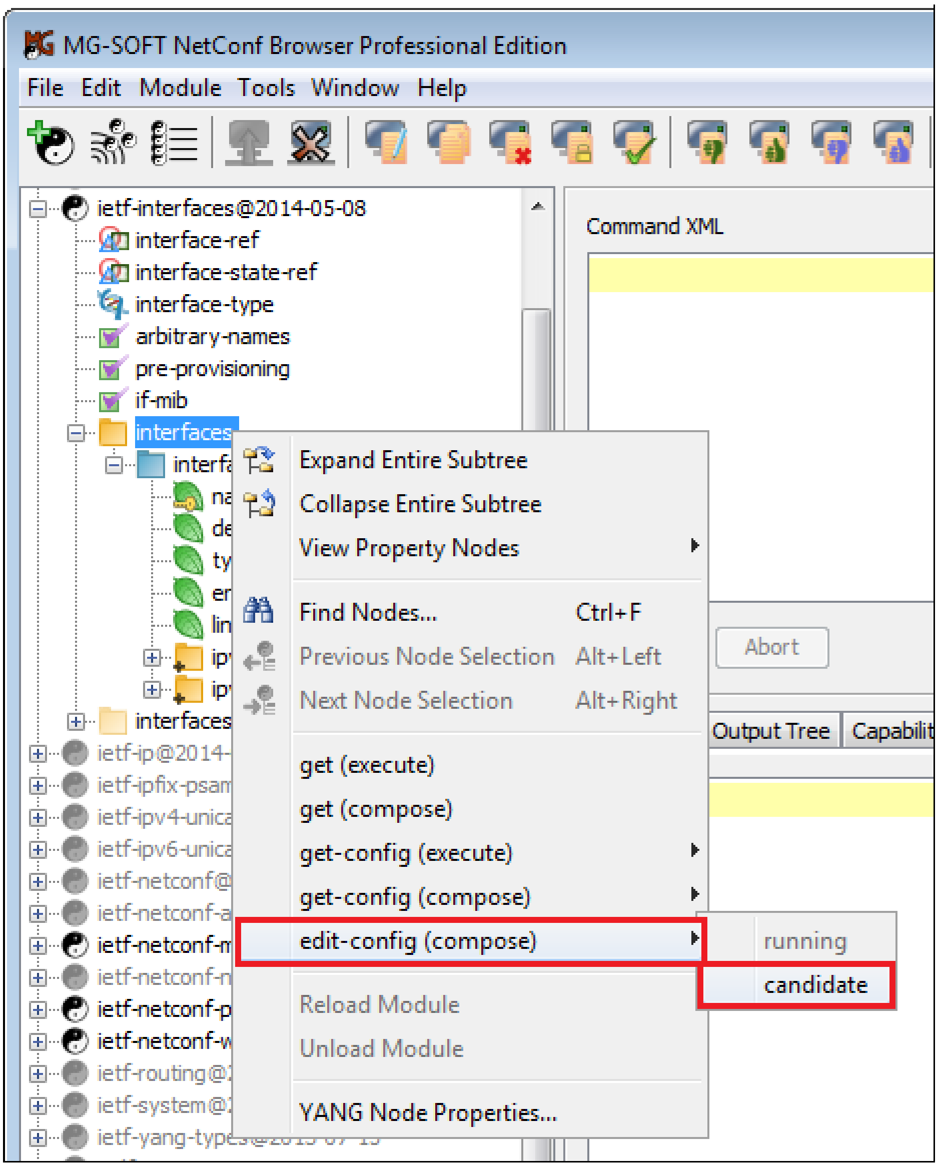
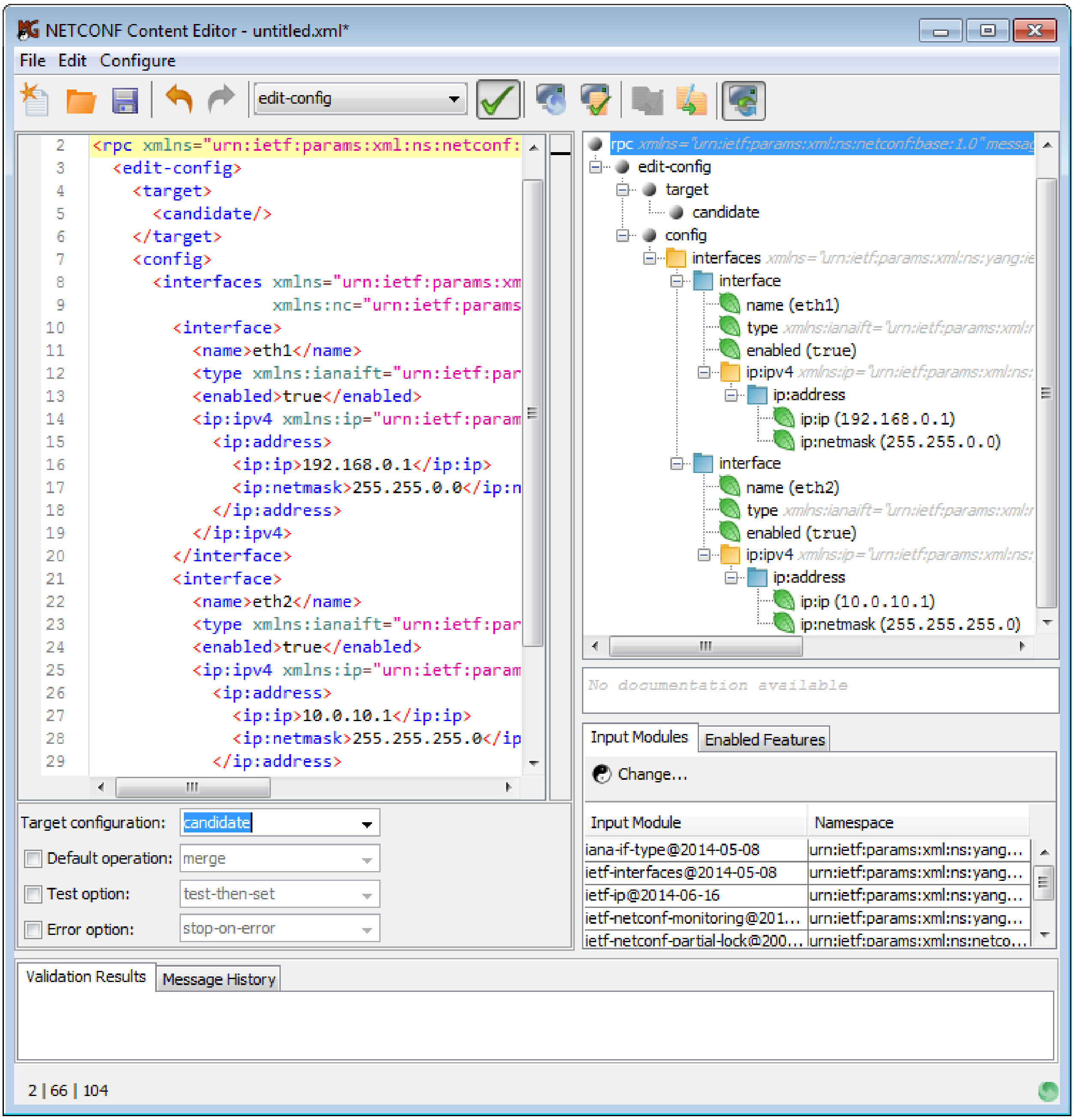


Figure 94: Choosing the *edit-config/candidate* command on a subtree node

3. NetConf Browser creates and sends a NETCONF **get-config** request to retrieve the configuration data of the selected subtree from the candidate configuration datastore (the retrieved configuration data serves as a template for composing the **edit-config** request) and displays it in the NETCONF Content Editor window (Figure 95). Note that the **edit-config** content type option is automatically enabled in the NETCONF Content Editor window.



```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:
  <edit-config>
    <target>
      <candidate/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xm
                  xmlns:nc="urn:ietf:params
        <interface>
          <name>eth1</name>
          <type xmlns:ianaift="urn:ietf:par
          <enabled>true</enabled>
          <ip:ipv4 xmlns:ip="urn:ietf:param
            <ip:address>
              <ip:ip>192.168.0.1</ip:ip>
              <ip:netmask>255.255.0.0</ip:n
            </ip:address>
          </ip:ipv4>
        </interface>
        <interface>
          <name>eth2</name>
          <type xmlns:ianaift="urn:ietf:par
          <enabled>true</enabled>
          <ip:ipv4 xmlns:ip="urn:ietf:param
            <ip:address>
              <ip:ip>10.0.10.1</ip:ip>
              <ip:netmask>255.255.255.0</ip
            </ip:address>
```

```
rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" messag
  edit-config
    target
      candidate
    config
      interfaces xmlns="urn:ietf:params:xml:ns:yang:ie
        interface
          name (eth1)
          type xmlns:ianaift="urn:ietf:params:xml:r
          enabled (true)
          ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:
            ip:address
              ip:ip (192.168.0.1)
              ip:netmask (255.255.0.0)
        interface
          name (eth2)
          type xmlns:ianaift="urn:ietf:params:xml:r
          enabled (true)
          ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:
            ip:address
              ip:ip (10.0.10.1)
              ip:netmask (255.255.255.0)
```

No documentation available

Target configuration: candidate
☐ Default operation: merge
☐ Test option: test-then-set
☐ Error option: stop-on-error

Input Module	Namespace
iana-if-type@2014-05-08	urn:ietf:params:xml:ns:yang...
ietf-interfaces@2014-05-08	urn:ietf:params:xml:ns:yang...
ietf-ip@2014-06-16	urn:ietf:params:xml:ns:yang...
ietf-netconf-monitoring@201...	urn:ietf:params:xml:ns:yang...
ietf-netconf-partial-lock@200...	urn:ietf:params:xml:ns:netco...

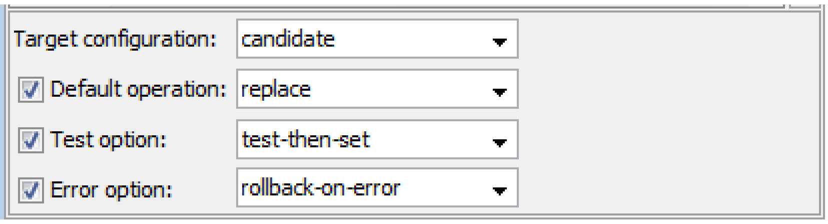
Figure 95: Example of the NETCONF Content Editor window displaying a retrieved `interfaces` configuration subtree (to be modified)

4. For detailed step-by-step instructions on how to model a configuration tree for the edit-config operation in a visual manner, please refer to the Edit the Running Configuration section, steps 4-19.

5. After you have finished modeling the configuration tree for the edit-config request, specify the settings for the **edit-config** operation in the Quick options panel under the XML Editor (Figure 96), as follows:

    - In the ***Target configuration*** drop-down list, select the ***candidate*** entry to make the change to the candidate configuration.

- Check the ***Default operation*** checkbox and select the ***replace*** option from the accompanying drop-down list to replace the existing configuration entries in the candidate datastore with the ones configured in the NETCONF Content Editor window.
- If the server supports the **:validate** capability, check the ***Test option*** checkbox and select the ***test-than-set*** option from the accompanying drop-down list. This way, the server will perform a validation test before attempting to set the configuration. If any validation error occurs, the edit-config operation will not be performed.
- Check the ***Error option*** checkbox and select the ***rollback-on-error*** option if available (it depends on the **:rollback-on-error** capability) from the drop-down list. If this option is selected, the server will restore the configuration to the previous state if an error occurs while performing the edit-config operation. If this option is not available, leave the ***Error option*** checkbox unchecked (this aborts the edit-config operation on first error – if any).



Target configuration: candidate

☑ Default operation: replace

☑ Test option: test-then-set

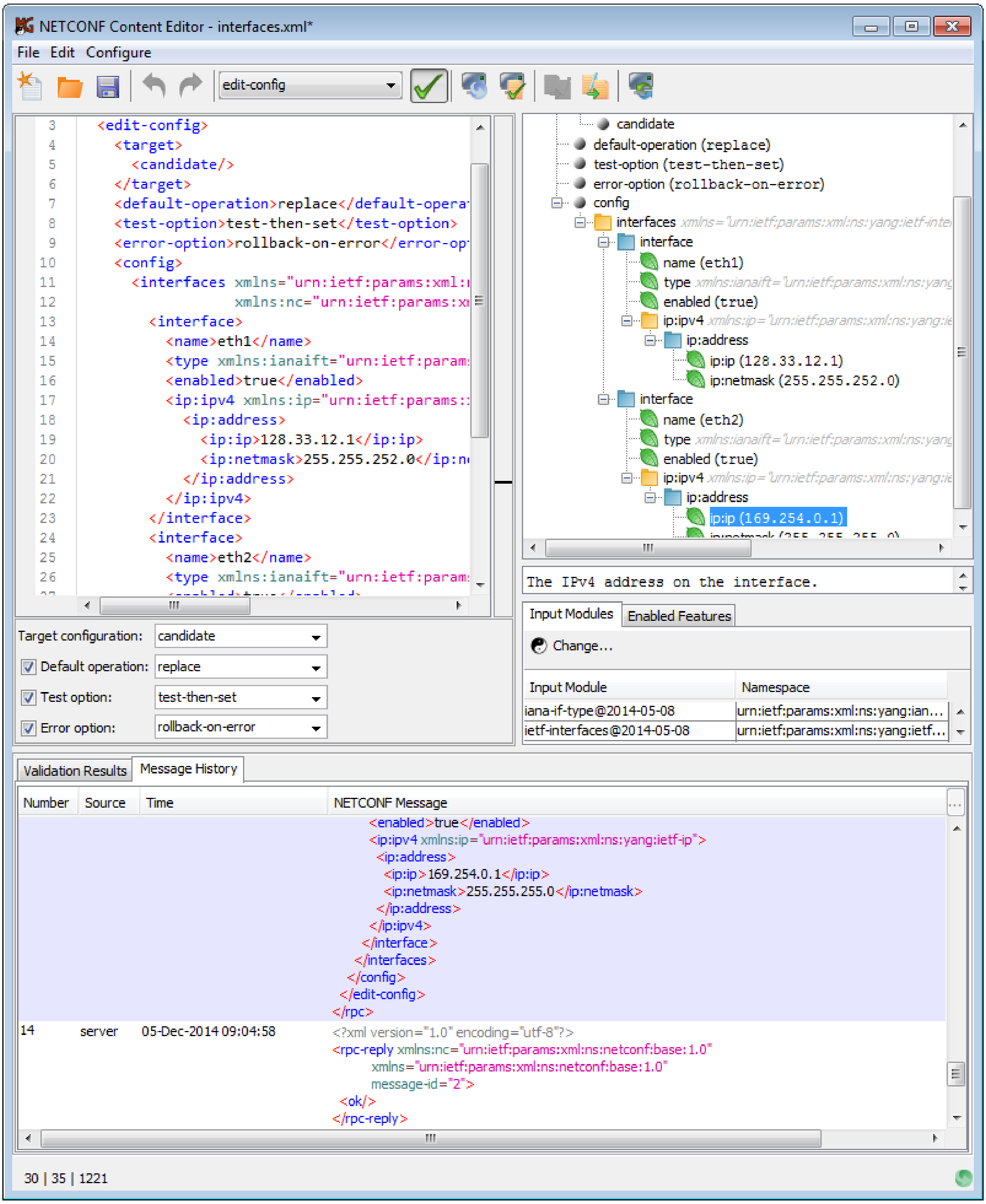
☑ Error option: rollback-on-error

Figure 96: Selecting the quick options for edit-config operation on the candidate datastore

6. After you have configured the quick options for the edit-config operation, click the ***Send as RPC*** () button in the toolbar to send the edit-config request to the server.
7. The **Message History** tab at the bottom of the NETCONF Content Editor window becomes active and displays the actual **edit-config** RPC message sent to the server and the corresponding RPC reply received from the server (Figure 97).
8. The NETCONF server will attempt to perform the configuration change according to your settings. If the **edit-config** operation succeeds, the server will respond with a reply message containing the **<ok>** element (see the Message History list in Figure 97). If the edit-config operation fails, the server will respond with the reply message containing the error description.

> The operation status icon in the right section of the status bar in the NETCONF Content Editor window indicates whether the operation completed successfully (🟢) or resulted in error (🔴).
>
> See also the **Output XML** panel, the **Log** tab and **Session History** tab in the main window for the server response.

9. To save the entire content of the XML Editor (i.e., edit-config RPC) to a file for future use, select the ***File / Save*** command in the NETCONF Content Editor window and in the Save dialog box specify the location and name of the resulting XML file. You can later load the XML file back into the NETCONF Content Editor window by using the ***File / Load*** command.



```
<edit-config>
  <target>
    <candidate/>
  </target>
  <default-operation>replace</default-opera
  <test-option>test-then-set</test-option>
  <error-option>rollback-on-error</error-op
  <config>
    <interfaces xmlns="urn:ietf:params:xml:i
                xmlns:nc="urn:ietf:params:xı
      <interface>
        <name>eth1</name>
        <type xmlns:ianaift="urn:ietf:param:
        <enabled>true</enabled>
        <ip:ipv4 xmlns:ip="urn:ietf:params::
          <ip:address>
            <ip:ip>128.33.12.1</ip:ip>
            <ip:netmask>255.255.252.0</ip:n
          </ip:address>
        </ip:ipv4>
      </interface>
      <interface>
        <name>eth2</name>
        <type xmlns:ianaift="urn:ietf:param:
```

```
    candidate
  default-operation (replace)
  test-option (test-then-set)
  error-option (rollback-on-error)
  config
    interfaces xmlns='urn:ietf:params:xml:ns:yang:ietf-inte
      interface
        name (eth1)
        type xmlns:ianaift='urn:ietf:params:xml:ns:yang
        enabled (true)
        ip:ipv4 xmlns:ip='urn:ietf:params:xml:ns:yang:ie
          ip:address
            ip:ip (128.33.12.1)
            ip:netmask (255.255.252.0)
      interface
        name (eth2)
        type xmlns:ianaift='urn:ietf:params:xml:ns:yang
        enabled (true)
        ip:ipv4 xmlns:ip='urn:ietf:params:xml:ns:yang:ie
          ip:address
            ip:ip (169.254.0.1)
```

The IPv4 address on the interface.

Target configuration: candidate

☑ Default operation: replace

☑ Test option: test-then-set

☑ Error option: rollback-on-error

Input Module	Namespace
iana-if-type@2014-05-08	urn:ietf:params:xml:ns:yang:ian…
ietf-interfaces@2014-05-08	urn:ietf:params:xml:ns:yang:ietf…

Number	Source	Time	NETCONF Message

			`<enabled>true</enabled>`
			`<ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip">`
			`<ip:address>`
			`<ip:ip>169.254.0.1</ip:ip>`
			`<ip:netmask>255.255.255.0</ip:netmask>`
			`</ip:address>`
			`</ip:ipv4>`
			`</interface>`
			`</interfaces>`
			`</config>`
			`</edit-config>`
			`</rpc>`

14	server	05-Dec-2014 09:04:58	`<?xml version="1.0" encoding="utf-8"?>`
			`<rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"`
			`xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"`
			`message-id="2">`
			`<ok/>`
			`</rpc-reply>`

30 | 35 | 1221

Figure 97: Viewing the edit-config request and reply messages exchanged with the server

10.2.3 Commit Changes to Running Configuration

After successfully performing configuration changes to the candidate configuration datastore, it is time to impact the running system. This is done by using the NETCONF **commit** or **confirmed commit** (if supported) operation, which sets the running configuration to the value of the candidate configuration, as described in this section.

Using the Commit Operation

1. To perform the NETCONF **commit** operation, select the ***Tools / Commit*** command from the main menu (Figure 98).

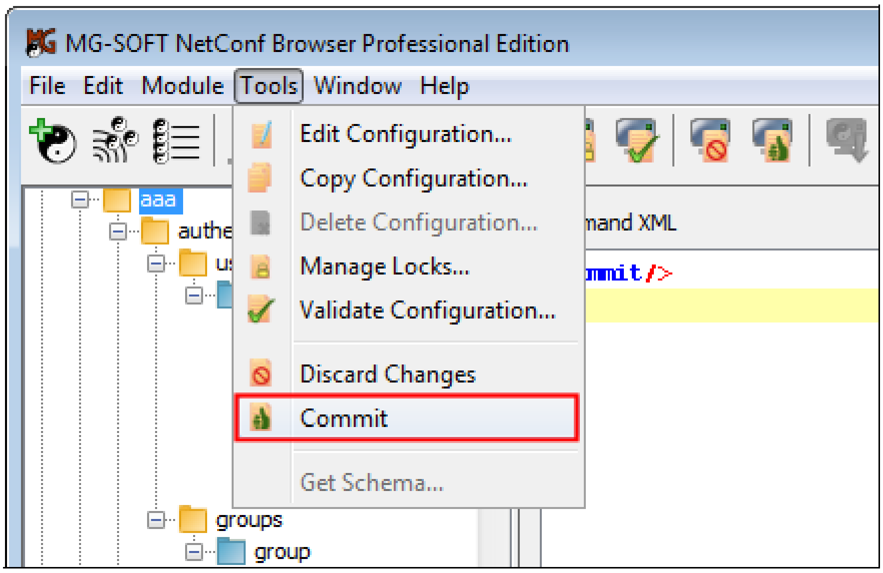
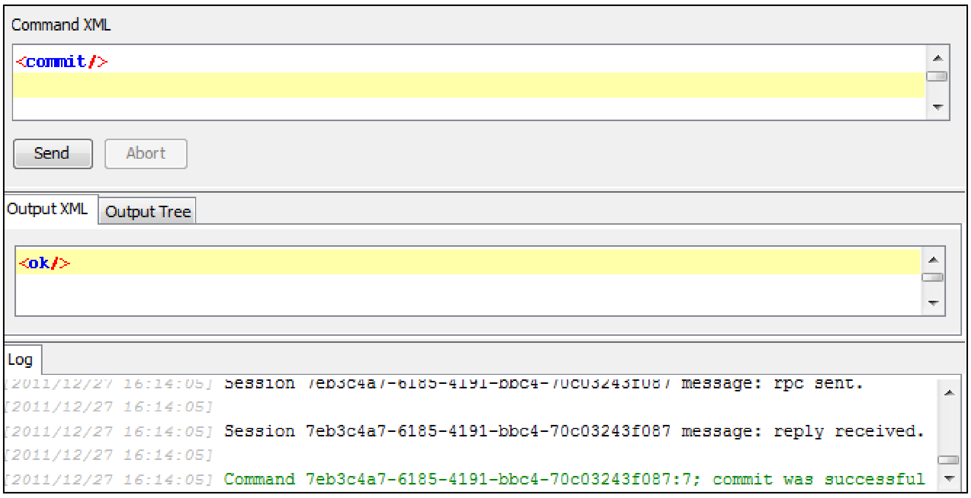


Figure 98: Selecting the Commit command from the main menu

2. NetConf Browser creates and sends the NETCONF **commit** request to the server (see the Command XML panel in Figure 99), attempting to set the running configuration to the current value of the candidate configuration. If the **commit** operation succeeds, the server responds with a reply message containing the **<ok>** element (see the Output XML panel in Figure 99).



```
<commit/>
```

```
<ok/>
```

```
[2011/12/27 16:14:05] Session 7eb3c4a7-6185-4191-bbc4-70c03243f087 message: rpc sent.
[2011/12/27 16:14:05]
[2011/12/27 16:14:05] Session 7eb3c4a7-6185-4191-bbc4-70c03243f087 message: reply received.
[2011/12/27 16:14:05]
[2011/12/27 16:14:05] Command 7eb3c4a7-6185-4191-bbc4-70c03243f087:7; commit was successful
```

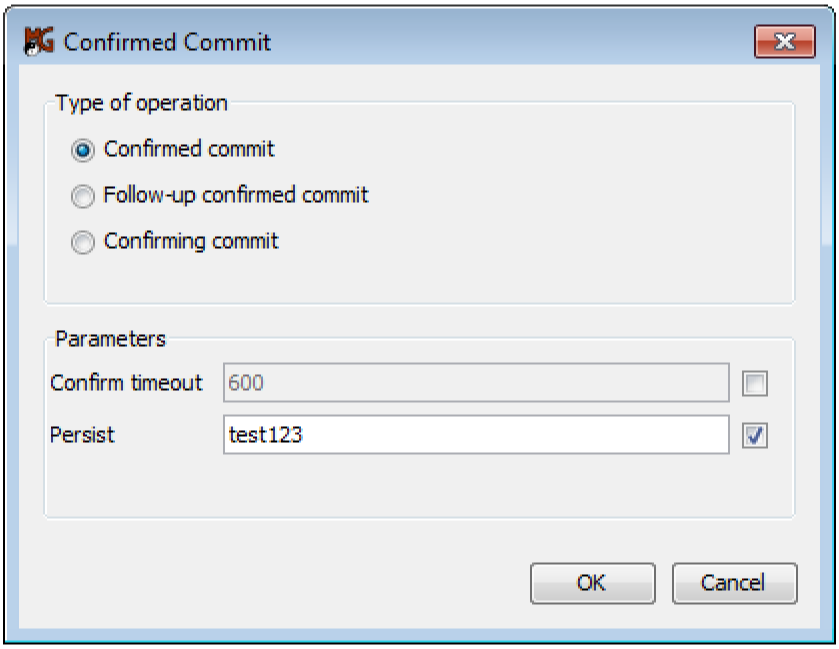
Figure 99: Viewing the results of a *commit* operation

Using the Confirmed Commit Operation

If the NETCONF server supports the **`:confirmed-commit`** capability, the **confirmed commit** operation can be used to apply changes to the running configuration. The confirmed commit operation consists of sending (at least) two distinct commit requests. First, a commit request containing the <confirmed> parameter is sent to the NETCONF server - this is called **confirmed commit**. Then, within a defined time frame, another commit request called **confirming commit** request is sent to the NETCONF server to confirm the commit operation. The advantage of the confirmed commit operation is that the running configuration is automatically reverted to its previous state if the NETCONF server does not receive the confirming commit request within the defined time period (10 minutes by default). This is particularly useful in cases when applying new configuration could unintentionally isolate the remote NETCONF device from the network (e.g., new firewall rules, passwords, etc.).

This section describes how to use MG-SOFT NetConf Browser to perform the **confirmed commit** operation.

1. To perform the NETCONF **confirmed commit** operation, select the ***Tools / Confirmed Commit*** command from the main menu. The Confirmed Commit dialog box appears (Figure 100).



Confirmed Commit

Type of operation

● Confirmed commit
○ Follow-up confirmed commit
○ Confirming commit

Parameters

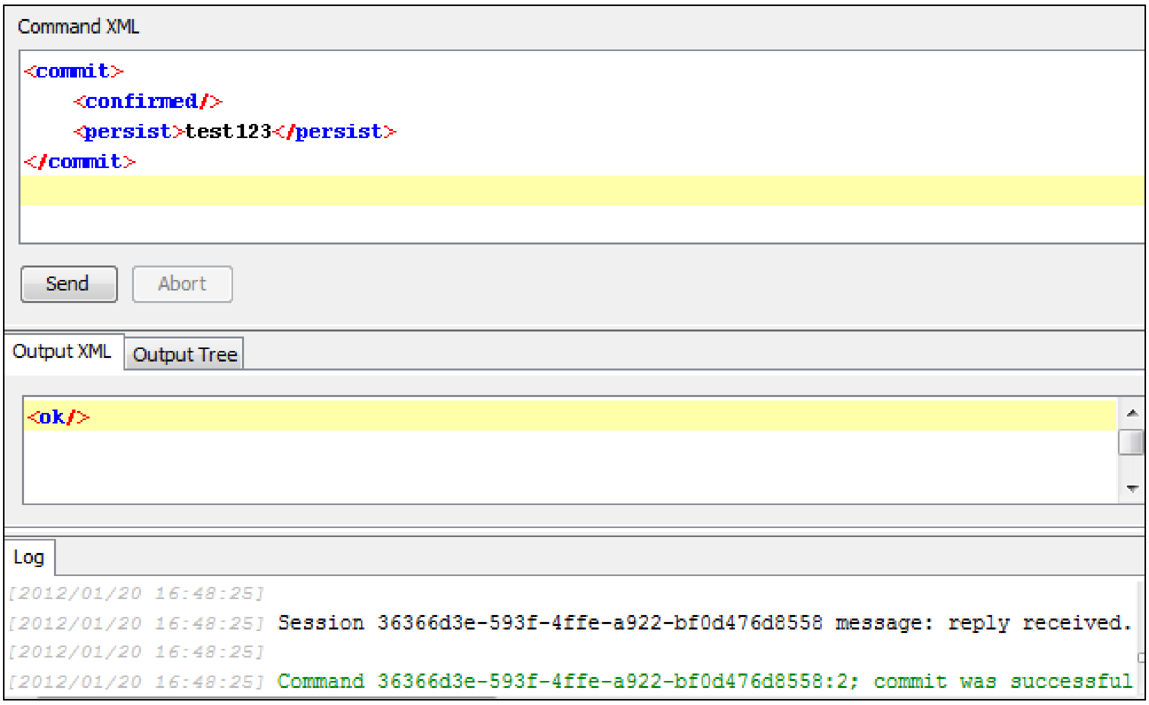
Confirm timeout 600 □
Persist test123 ☑

Figure 100: Setting the *confirmed commit* operation parameters

2. In the Confirmed Commit dialog box, select the ***Confirmed commit*** radio button in the **Type of operation** frame.
3. In the **Parameters** frame, you can optionally configure the following:
   - To set the confirm commit timeout to a value other than default (600 seconds), check the checkbox next to the ***Confirm timeout*** input line and enter a new value (in seconds) into the ***Confirm timeout*** input line. This setting determines how long the NETCONF server will wait for the confirming commit (or a follow-up commit) request before performing the automatic rollback.

- If the NETCONF server supports the **`:confirmed-commit:1.1`** capability, you can make the confirmed commit operation persistent (surviving a session termination, and set a token on the ongoing confirmed commit). To do this, check the checkbox next to the ***Persist*** input line and enter a persist phrase into the accompanying input line. If the persist phrase is set, the confirming commit request can be sent from a different session but must include the same persist phrase in persist-id parameter in order for the confirming commit to succeed. The **`:confirmed-commit:1.0`** capability does not support this feature.

4. Click the ***OK*** button to send the NETCONF **confirmed commit** request to the server (see the Command XML panel in Figure 103), to set the value of the running configuration to the current value of the candidate configuration. If the **confirmed commit** operation succeeds, the server responds with a reply message containing the **<ok>** element (see the Output XML panel in Figure 103).



Command XML

```
<commit>
    <confirmed/>
    <persist>test123</persist>
</commit>
```

```
<ok/>
```

```
[2012/01/20 16:48:25]
[2012/01/20 16:48:25] Session 36366d3e-593f-4ffe-a922-bf0d476d8558 message: reply received.
[2012/01/20 16:48:25]
[2012/01/20 16:48:25] Command 36366d3e-593f-4ffe-a922-bf0d476d8558:2; commit was successful
```

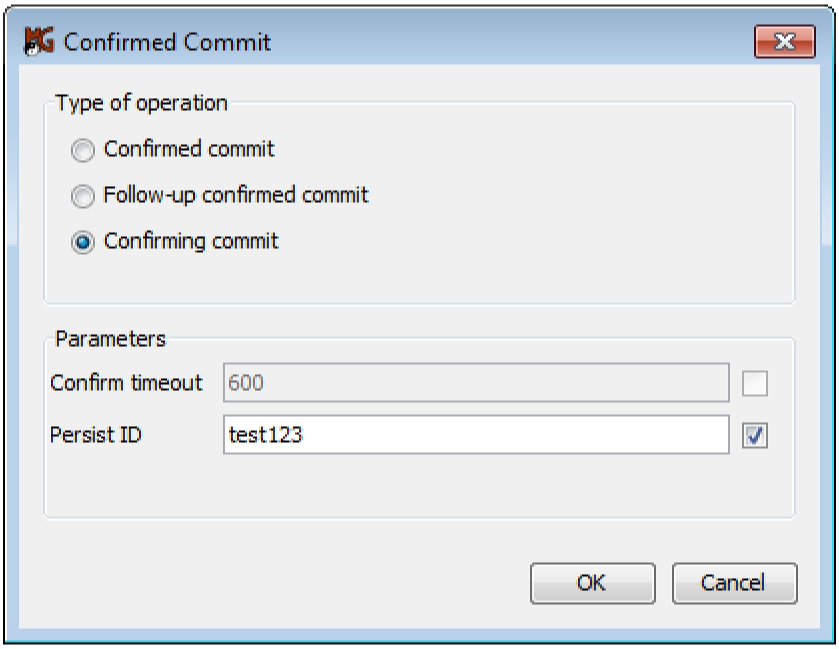
Figure 101: Viewing results of the confirmed commit operation

5. After testing the device that implements the temporarily applied configuration and determining that it functions as intended, you need to send the confirming commit request within the confirm timeout interval to the device. This makes the configuration changes permanent (otherwise, automatic configuration rollback occurs after timeout). To send the confirming commit request, select the ***Tools / Confirmed Commit*** command from the main menu. The Confirmed Commit dialog box appears again (Figure 102).

> **Tip:** To cancel the ongoing confirmed commit operation, select the ***Tools / Cancel Confirmed Commit*** command. In case of persistent confirmed commit operation, enter the corresponding persist identification string into the ***Persist ID*** input line in the dialog box that appears and click the ***OK*** button.

6. In the **Type of operation** frame, select the ***Confirming commit*** radio button.

7. If you have set the **persist** parameter with the initial confirmed commit request, check the checkbox next to the ***Persist-id*** input line and enter the same persist phrase into the accompanying input line. Otherwise, ignore this setting.



Confirmed Commit

Type of operation

○ Confirmed commit

○ Follow-up confirmed commit

● Confirming commit

Parameters

Confirm timeout 600 □

Persist ID test123 ☑

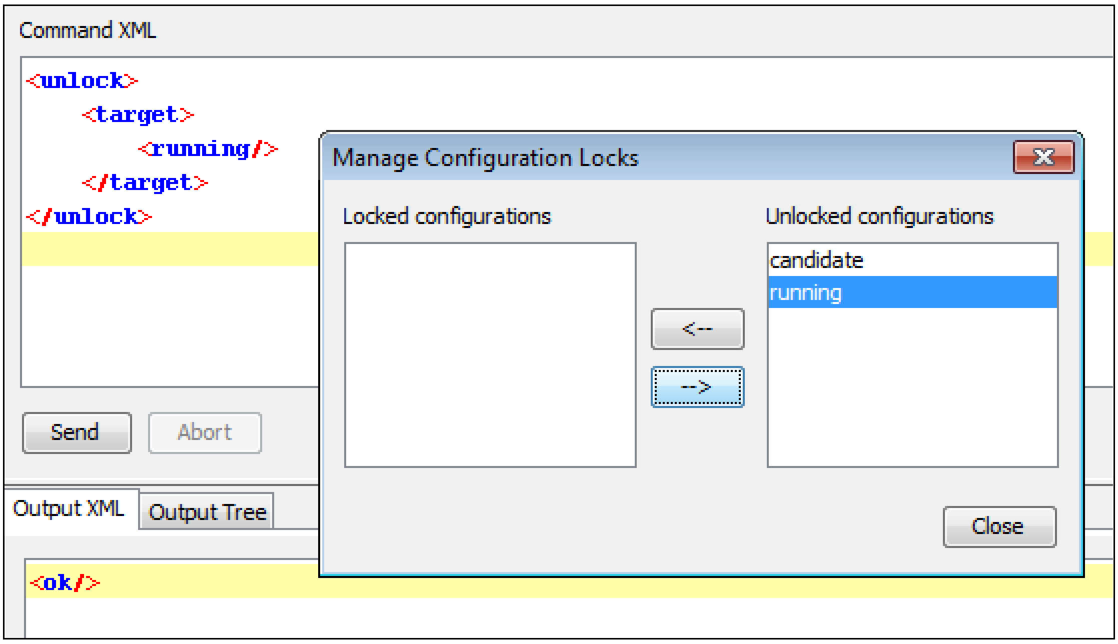
Figure 102: Setting the *confirming commit* operation parameters

8. Click the ***OK*** button to send the NETCONF **confirming commit** request to the server (see the Command XML panel in Figure 103), to set the value of the running configuration to the current value of the candidate configuration. If the **commit** operation succeeds, the server responds with a reply message containing the **<ok>** element (see the Output XML panel in Figure 103).

10.2.4 Unlock the Candidate and Running Configuration

After you have applied the change to the running configuration, you should unlock the candidate and running configurations to allow for other NETCONF clients and other methods (e.g., SNMP, CLI, …) to access it in write mode.

1. In the main window, select the ***Tools / Manage Locks*** command from the main menu (Figure 70).
2. In the ***Manage Configuration Locks*** dialog box that appears, select the **`candidate`** configuration in the **Locked configurations** list and click the ***right-arrow button*** (-->) to move it to the **Unlocked configurations** list (Figure 103).
3. NetConf Browser creates and sends the NETCONF **unlock** request to the server, attempting to unlock the unning configuration (see the Command XML panel in Figure 103). If the lock operation succeeds, the server responds with a reply message containing the **<ok>** element (see the Output XML panel in Figure 103).
4. Repeat the procedure in step 2 to unlock also the **`running`** configuration.



```
<unlock>
    <target>
        <running/>
    </target>
</unlock>
```

Unlocked configurations: candidate, running

```
<ok/>
```

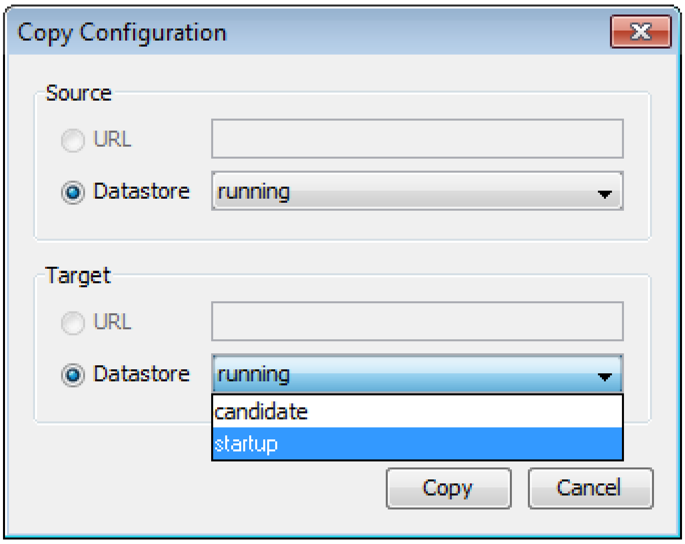
Figure 103: Unlocking the candidate and running configuration

10.3 Applying Changes to Startup Configuration

If the device supports the **`:startup`** capability, the changes you have applied to the running configuration (as described in previous sections) should be saved also to the startup configuration, as described in this section.

> If supported by a given device, the **startup** configuration datastore holds the configuration loaded by the device when it boots. Operations that affect the running configuration will not be automatically copied to the startup configuration. An explicit **copy-config** operation from the `running` to the `startup` is used to update the startup configuration to the current contents of the running configuration.

1. To copy the contents of the running configuration to the startup configuration, select the ***Tools / Copy Configuration*** command from the main menu.
2. In the Copy Configuration dialog box that appears (Figure 104), select the source and target for the **copy-config** operation, as follows:



Source

○ URL

● Datastore running

Target

○ URL

● Datastore running

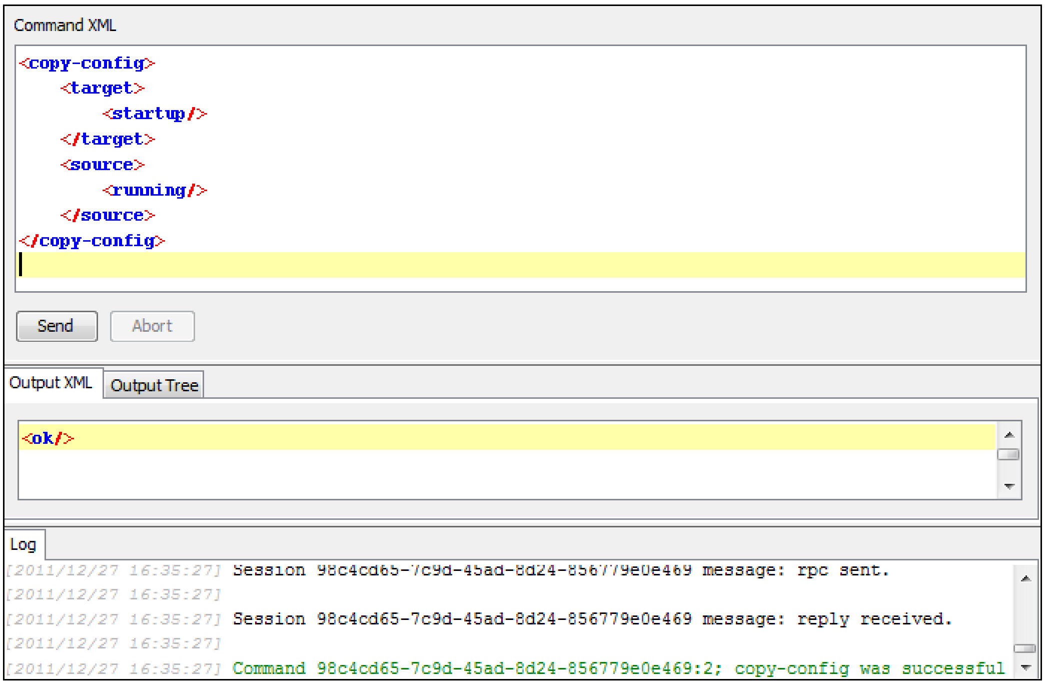
candidate

startup

Figure 104: Copy Configuration dialog box

- In the **Source** frame, select the ***Datastore*** radio button and the **`running`** entry from drop the accompanying drop-down list.
- In the **Target** frame, select the ***Datastore*** radio button and the **`startup`** entry from drop the accompanying drop-down list.

3. Click the ***Copy*** button at the bottom of the Copy Configuration dialog box to start the **copy-config** operation.
4. NetConf Browser creates and sends the NETCONF **copy-config** request to the server, attempting to set the startup configuration to the current value of the running configuration (see the Command XML panel in Figure 105). If the **copy-config** operation succeeds, the server responds with a reply message containing the **<ok>** element (see the Output XML panel in Figure 105).



```
<copy-config>
    <target>
        <startup/>
    </target>
    <source>
        <running/>
    </source>
</copy-config>
```

```
<ok/>
```

```
[2011/12/27 16:35:27] Session 98c4cd65-7c9d-45ad-8d24-856779e0e469 message: rpc sent.
[2011/12/27 16:35:27]
[2011/12/27 16:35:27] Session 98c4cd65-7c9d-45ad-8d24-856779e0e469 message: reply received.
[2011/12/27 16:35:27]
[2011/12/27 16:35:27] Command 98c4cd65-7c9d-45ad-8d24-856779e0e469:2; copy-config was successful
```

Figure 105: Viewing the copy-config operation command and its results

11 USING NETCONF CONTENT EDITOR

MG-SOFT NetConf Browser includes an XML editor and validator for NETCONF content that complies with the RFC 6110 specification. This feature allows you to edit or validate new and existing content according to Document Schema Definition Languages (DSDL), which are automatically generated from selected YANG modules in the background.

1. To start editing, load the desired YANG or YIN modules into NetConf Browser and select the ***Tools / Edit NETCONF Content*** command. This will open the NETCONF Content Editor window (Figure 106), which contains the following components:

   1) Menu bar

   2) Toolbar

   3) NETCONF content XML Editor panel – XML document editor with syntax coloring and autocomplete feature

   4) NETCONF content Tree Editor panel - represents the currently edited XML document in a graphical tree form and lets you edit it

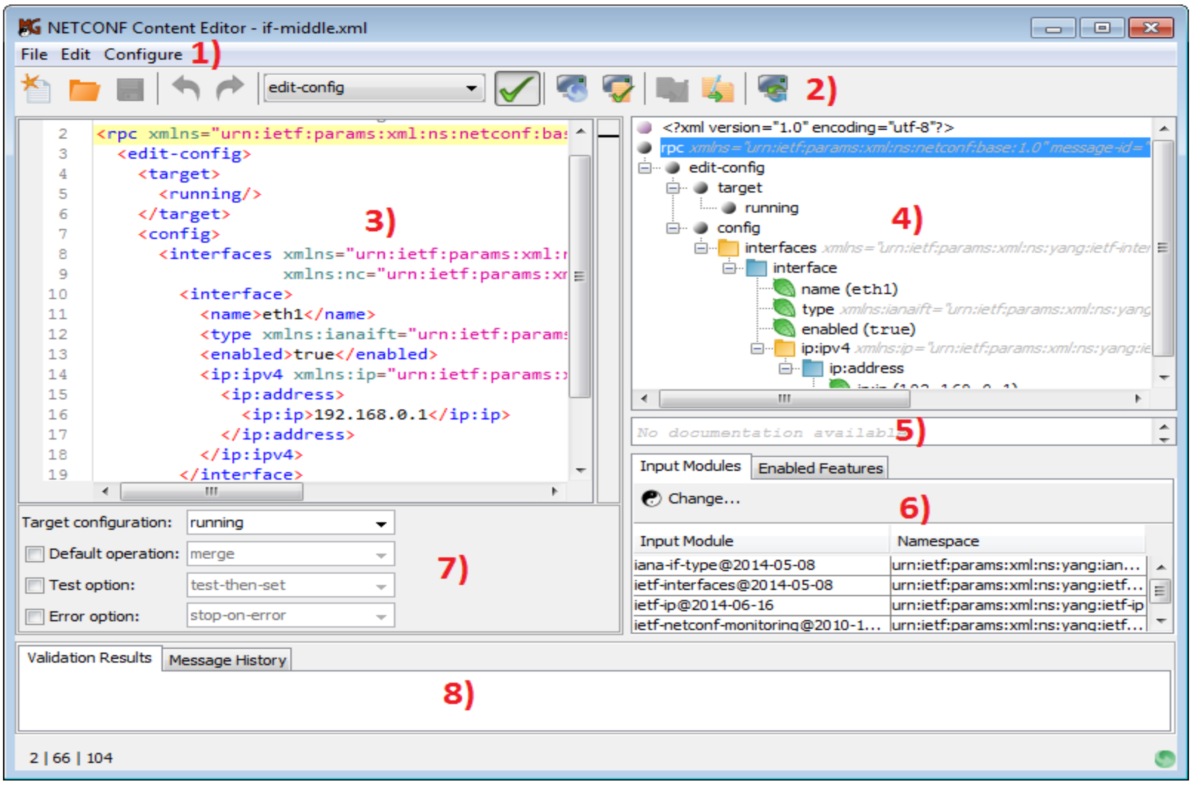
   5) Informational text area containing description of the currently selected tree node (if the corresponding YANG module provides it)

   6) Configuration tabs (Input Modules and Enabled Features) let you refine the current validation schema (YANG modules, features)

   7) Edit-config quick options panel

   8) Output panel, containing two tabs:

      - ❑ Validation Results tab (displays the validation errors and warnings)
      - ❑ Message History tab (displays a chronological list of NETCONF RPC requests sent from this window and responses received)



```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:ba
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <interfaces xmlns="urn:ietf:params:xml:r
                  xmlns:nc="urn:ietf:params:xr
        <interface>
          <name>eth1</name>
          <type xmlns:ianaift="urn:ietf:params
          <enabled>true</enabled>
          <ip:ipv4 xmlns:ip="urn:ietf:params:x
            <ip:address>
              <ip:ip>192.168.0.1</ip:ip>
            </ip:address>
          </ip:ipv4>
        </interface>
```

```
<?xml version="1.0" encoding="utf-8"?>
rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="
  edit-config
    target
      running
    config
      interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-inter
        interface
          name (eth1)
          type xmlns:ianaift="urn:ietf:params:xml:ns:yang
          enabled (true)
          ip:ipv4 xmlns:ip="urn:ietf:params:xml:ns:yang:ie
            ip:address
```

No documentation availabl

Target configuration: running

□ Default operation: merge

□ Test option: test-then-set

□ Error option: stop-on-error

Input Module	Namespace
iana-if-type@2014-05-08	urn:ietf:params:xml:ns:yang:ian...
ietf-interfaces@2014-05-08	urn:ietf:params:xml:ns:yang:ietf...
ietf-ip@2014-06-16	urn:ietf:params:xml:ns:yang:ietf-ip
ietf-netconf-monitoring@2010-1...	urn:ietf:params:xml:ns:yang:ietf...

Figure 106: NETCONF Content Editor window

2. After opening the NETCONF Content Editor window, select the type of content you wish to edit or validate by selecting it from the ***Content type*** drop-down list in the toolbar. You can choose between these content types:

    - **`data`**: Use this content type to edit/validate entire datastores – XML instance files, which contain both state and configuration data,

    - **`config`**: Use this content type when you wish to edit/validate configuration data only,

    - **`get`**: Use this content type when you wish to compose an XML document for a NETCONF get request. This content type lets you add get operation specific elements in addition to being able to edit a configuration. For example, it will let you specify a NETCONF filter element. You are also able to send a valid document to the currently connected NETCONF server as an get request by clicking the ***Send As RPC*** button () in the toolbar,

    - **`get-config`**: Use this content type when you wish to compose an XML document for a NETCONF get-config request. This content type lets you add get-config specific elements in addition to being able to edit a configuration. For example, it will let you specify a NETCONF filter element. You are also able to send a valid document to the currently connected NETCONF server as an actual get-config request by clicking the ***Send As RPC*** button () in the toolbar,

    - **`edit-config`**: Use his content type when you wish to compose an XML document for a NETCONF edit-config request. This content type differs from “config” in that it lets you add edit-config specific elements and attributes in addition to being able to edit a configuration. For example, it will let you specify a NETCONF operation attribute (e.g., create, merge, replace, delete, etc.) for each element of the config subtree. You are also able to send a valid document to the currently connected NETCONF server as an actual edit-config request by clicking the ***Send As RPC*** button () in the toolbar,

    - **`copy-config`**: Use this content type if you wish to compose an XML document for a NETCONF copy-config request. This content type lets you add copy-config specific elements (e.g., source and target configuration), as well as compose an entire configuration when the source is specified to be an inline config element. You are also able to send a valid document to the currently connected NETCONF server as an actual copy-config request by clicking the ***Send As RPC*** button () in the toolbar.

    - **`create-subscription`**: Use this content type if you wish to compose an XML document for a NETCONF create-subscription request. This content type lets you add create-subscription specific elements. For example, it will let you specify the start and stop time for the create-subscription operation so that a notification replay may be requested as specified in RFC 5277. You are also able to send a valid document to the currently connected NETCONF server as an actual create-subscription request by clicking the ***Send As RPC*** button () in the toolbar.

    - **`rpc`**: Allows you to create RPC operation requests based on RPC definitions available in the input modules and also lets you send a valid document as an RPC request, the same way this is possible with the `get`, `get-config` and `edit-config` content type,

- **`notification`**: Lets you validate or create examples of notifications defined in your input modules,
- **`get-reply`**: Enables you to validate or create examples of possible rpc-reply messages defined by your input modules, which would be created on server with a NETCONF get operation,
- **`get-config-reply`**: Enables you to validate or create examples of possible rpc-reply messages defined by your input modules, which would be created on server with a NETCONF get-config operation,
- **`rpc-reply`**: Enables you to validate or create examples of possible rpc-reply messages defined by your input modules, which would be created on server with any NETCONF RPC operation request.

3. Once you have chosen a content type, you can refine the schema behind it via the **Configuration** tabs. There are two ways to refine the schema:
   - by selecting **input modules** for the schema generation algorithm among all loaded YANG/YIN modules, and
   - by selecting enabled **features**, defined by the input modules.

   To automatically select the input YANG/YIN modules and features supported by the currently connected NETCONF server, click the ***Adapt Modules and Features to Session*** () toolbar button.

   To manually include/exclude a YANG/YIN module, select the ***Input Modules*** tab and click the ***Change***... button below it to open a dialog which will let you select the input modules. Similarly, you can enable or disable features in the ***Enabled Features*** tab (provided that the selected input modules define features).

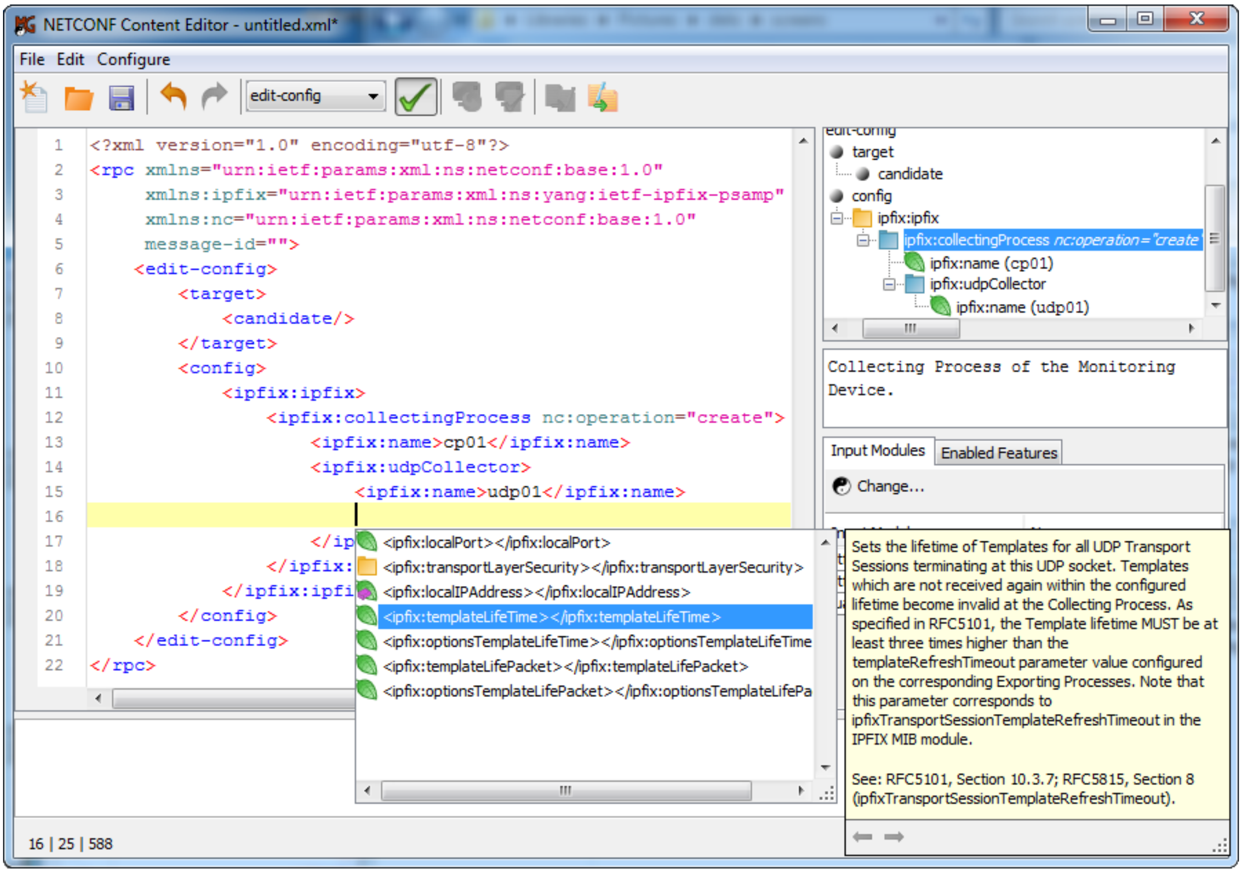
   Each time you make a change to the settings above, the schema that is used to validate your document may have to be re-generated. A progress bar dialog will appear each time this occurs, temporarily preventing you from editing the document. The main purpose of the schema is to enable document validation. NETCONF Content Editor, also uses it for the autocomplete feature that is available when writing XML documents.

   > **Note:** Actually, three different schemas are automatically generated in the background – a Relax NG schema, a DSRL schema (Document Schema Renaming Language schema) and an ISO Schematron schema – but for simplicity reasons we refer to them as if it were a single schema. (the schema files are generated in the following folder: $USER_HOME/`.mgnetconfbrowser/schemas`).

4. Edit your document in the NETCONF content editor. When you start writing an XML tag, the editor will assist you with the **autocomplete** feature, which displays a list of all possible elements defined by the schema. The autocomplete drop-down list of choices appears in the NETCONF content editor panel when you type in the "<" character or when you press the ***CTRL+Space*** keyboard keys (when the cursor is placed where completions are possible). Select an item in the autocomplete drop-down list to view its description (from YANG module) in a tooltip next to the selected item (Figure 107). Press the ***Enter*** key to insert the selected item into the NETCONF content editor panel.

   > **Note:** Autocomplete is provided for XML elements, attributes and their values. Note that completions for attribute and element values will only be provided if a set of possible values can be determined for the current attribute/element – for example this is possible if the element is specified by a YANG leaf statement of type enumeration, bits or similar. The autocomplete feature is XML namespace aware.

5. As you edit the XML document, its tree representation in the NETCONF content tree panel on the right side changes accordingly. The NETCONF content tree panel represents your document's structure and may be used to quickly navigate large documents by double-clicking the tree nodes. It also gives you some information on individual nodes – node type in YANG terms and its YANG description if available.



```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:ipfix="urn:ietf:params:xml:ns:yang:ietf-ipfix-psamp"
     xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
     message-id="">
    <edit-config>
        <target>
            <candidate/>
        </target>
        <config>
            <ipfix:ipfix>
                <ipfix:collectingProcess nc:operation="create">
                    <ipfix:name>cp01</ipfix:name>
                    <ipfix:udpCollector>
                        <ipfix:name>udp01</ipfix:name>

                    </ip
                </ipfix:
            </ipfix:ipfi
        </config>
    </edit-config>
</rpc>
```

```
<ipfix:localPort></ipfix:localPort>
<ipfix:transportLayerSecurity></ipfix:transportLayerSecurity>
<ipfix:localIPAddress></ipfix:localIPAddress>
<ipfix:templateLifeTime></ipfix:templateLifeTime>
<ipfix:optionsTemplateLifeTime></ipfix:optionsTemplateLifeTime
<ipfix:templateLifePacket></ipfix:templateLifePacket>
<ipfix:optionsTemplateLifePacket></ipfix:optionsTemplateLifePa
```

Collecting Process of the Monitoring Device.

Sets the lifetime of Templates for all UDP Transport Sessions terminating at this UDP socket. Templates which are not received again within the configured lifetime become invalid at the Collecting Process. As specified in RFC5101, the Template lifetime MUST be at least three times higher than the templateRefreshTimeout parameter value configured on the corresponding Exporting Processes. Note that this parameter corresponds to ipfixTransportSessionTemplateRefreshTimeout in the IPFIX MIB module.

See: RFC5101, Section 10.3.7; RFC5815, Section 8 (ipfixTransportSessionTemplateRefreshTimeout).

Figure 107: Using the auto-complete feature in the NETCONF Content Editor window

6. As you edit the document, the validation error/warning messages appear in the notification area at the bottom of the window (Figure 106). You can locate the source of most errors by clicking links in the messages that appear in the notification area. The editor will also indicate error sources by underlining erroneous elements and by showing error icons in front of the relevant lines in the content editor panel. If you hover your mouse over an underlined element, a tooltip with the error message appears.

   The editor automatically validates the document as you edit it. The validation contains several steps:

   - Ensures that the document is well-formed (XML syntax compliant),
   - Checks if the document is valid according to the current RelaxNG schema (the element/attribute structure must be as defined by the input YANG modules),
   - Creates a copy of the current document in the background and fills in the missing default values using DSRL (prerequisite for the final step),

- Ensures that the copy of the document satisfies all semantic constraints specified by the input modules, such as XPath expressions from the YANG *when* statements (ISO Schematron).

You can disable the validation at any time by clicking the ***Validation Enabled*** (✓) toggle button in the toolbar.

7. Depending on the currently selected content type you may use additional features of the editor, for example:

   - **Quick Options for edit-config** (middle left panel)

     When the `edit-config` document type is selected, additional Quick options panel is displayed under the XML Editor. These options let you quickly set the target configuration datastore and specify how the edit-config operation is performed (e.g., default operation (merge, replace, none), test options (test-then-set, set, test-only), error options (stop-on-error, continue-on-error, rollback-on-error). The edit-config settings can be configured by checking/unchecking the respective checkboxes and selecting the corresponding options from the drop-down lists. Availability of some of the options depends on the server capabilities.

   - Use the ***Send as RPC ( )*** button to send valid documents to server when the `rpc, get, get-config` or `edit-config` document type is selected. The operation status icon in the right section of the status bar indicates whether the operation succeeded successfully or resulted in error. The exchanged RPC messages are displayed in the **Message History** tab of the Output panel.

   - Use the ***Validate Config*** button ( ) when your selected content type is `config` or `edit-config` and your NETCONF server supports this capability. With the `edit-config` content type selected, the config element will be extracted and wrapped into a validate operation, whereas if the `config` document type is selected only the wrapping part is performed. Both of these features require that an active session with a NETCONF server.

   - When the `config` or `edit-config` content type is selected, you can use the corresponding toolbar buttons (/), to convert the document between the two types, i.e., between an edit-config operation with a <config> element and a config datastore part – note that this may result in loss of certain information (e.g., NETCONF operation attributes within an edit-config's <config> element will be discarded if the content is converted to the `config` type, etc.).

   - Besides the standard text editing features such as ***Find***, ***Replace*** and file operations (***Save***, ***Open***, ***New***), the editor also offers XML pretty-printing capabilities (***Edit / Format XML***), which will transform the entire document into a form that is easily readable.

12 RECEIVE NETCONF NOTIFICATIONS

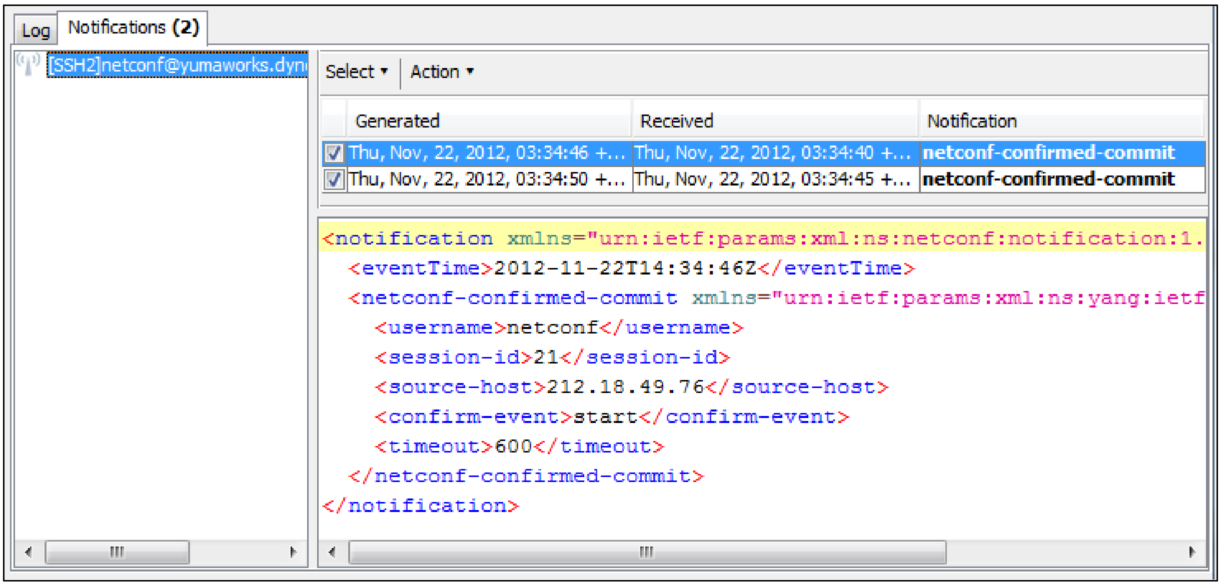
NetConf Browser supports receiving NETCONF notifications, as specified in RFC 5277.

If the connected NETCONF server supports the "**`:notification:1.0`**" capability, you can use NetConf Browser to subscribe to and receive asynchronous event notifications from it, as described in this section.

1. To subscribe to receiving NETCONF notifications from the currently connected NETCONF server, click the **Create *Subscription (Simple)*** toolbar button ().
2. NetConf Browser sends the <create-subscription> NETCONF request to the server and the server responds with an <rpc-reply> message containing **`OK`**. With this message exchange, the subscription to receiving all notifications from the default (NETCONF) stream is established. The subscription is valid until the active NETCONF session is terminated.

> **Tip:** To use advanced create-subscription method, where you can specify all valid subscription parameters (e.g., event stream), use the NETCONF Content Editor, as follows:
>
> 1. Open the NETCONF Content Editor by selecting the ***Tools / Edit NETCONF Content*** command.
> 2. Choose the **`create-subscription`** content type in the NETCONF Content Editor drop-down list. This will give you complete control over the XML payload to be sent to server.
> 3. Use NETCONF Content Editor auto-complete feature to form a valid payload.
> 4. Click the ***Send As RPC*** button to create the subscription.

3. Once the subscription has been set up, the NETCONF server starts sending the event notifications asynchronously over the connection, as the events occur within the system. NetConf Browser receives notifications and displays them in the **Notifications** tab in the bottom panel of the main window (Figure 108).



	Generated	Received	Notification
☑	Thu, Nov, 22, 2012, 03:34:46 +...	Thu, Nov, 22, 2012, 03:34:40 +...	netconf-confirmed-commit
☑	Thu, Nov, 22, 2012, 03:34:50 +...	Thu, Nov, 22, 2012, 03:34:45 +...	netconf-confirmed-commit

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.
  <eventTime>2012-11-22T14:34:46Z</eventTime>
  <netconf-confirmed-commit xmlns="urn:ietf:params:xml:ns:yang:ietf
    <username>netconf</username>
    <session-id>21</session-id>
    <source-host>212.18.49.76</source-host>
    <confirm-event>start</confirm-event>
    <timeout>600</timeout>
  </netconf-confirmed-commit>
</notification>
```

Figure 108: Viewing received NETCONF notifications

4. To view all notifications from a NETCONF session, click the session in the left portion **Notifications** tab. The upper-right panel of the **Notifications** tab displays all received notifications from the selected session. Click a notification in the upper-right panel to view its details in the lower-right panel (Figure 108)
5. If a notification is selected for more than 1 second, it is automatically marked as read (its typeface changes from bold to normal). To mark a notification as not read (unseen), right click it in the upper panel and select the ***Mark as unread*** pop-up command.