

Ericsson NETCONF Browser User Guide

USER GUIDE

Copyright

© Ericsson AB 2015 – 2016. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

License Agreements associated with the third party software used by this product can be found in the document [Third Party License Agreements](#).

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document [Trademark Information](#).



Contents

1	About This Guide	1
1.1	Target Groups	1
1.2	How This Guide is Organized	1
1.3	Conventions Used in This Guide	2
1.4	Comments About the Documentation	2
2	Overview	3
2.1	Prerequisites	3
3	Getting Started	3
3.1	Prerequisites	3
3.2	Working in Microsoft Windows	4
3.3	Working in Linux	5
4	Working with Node Connections	7
4.1	The Connections Workspace	7
4.2	Creating a New Connection	9
5	Working with Node Configurations	16
5.1	The Configuration Browser	16
5.2	Working with Parameter Groups and Structs	22
5.3	Working with Parameters	26
5.4	Validating a Configuration	30
5.5	Pushing Configuration Changes	31
6	Comparing Two Configurations	32
6.1	The Comparison Workspace	33
6.2	Performing a Comparison	37





1 About This Guide

This guide describes how to use the Ericsson NETCONF Browser to interact with live node configurations.

1.1 Target Groups

This document is intended for Operations and Maintenance (O&M) personnel who are using the Ericsson NETCONF Browser to work with node configurations in a live system environment.

1.1.1 Prerequisite Knowledge

Users of this document should familiar with the following topics:

- Microsoft Windows or Linux
- IP Networking
- Network Configuration Protocol (NETCONF)
- Ericsson Common Information Model (ECIM)
- Node Configuration Data
- Extensible Markup Language (XML)

1.2 How This Guide is Organized

This document is organized into the following major sections:

Table 1 Document Organization

Section	Description
About This Guide	Introduces the guide. This section describes the target groups, document structure, and typographic conventions.
Overview	Provides an overview of ENB.
Getting Started	Describes the installation of ENB.
Working with Node Connections	Provides tasks for working with node connections.



Section	Description
Working with Node Configurations	Provides tasks for working with node configurations.
Comparing Two Configurations	Describes how to compare two configurations in ENB.

1.3 Conventions Used in This Guide

Table 2 provides a list of typographic conventions that may be encountered in this document:

Table 2 Typographic Conventions

Convention	Description	Example
Bold	Menus, fields, and buttons, dialog boxes, and options on the screen.	After listening to your message, press 1 to reply.
Note	Text displayed outside of a paragraph that provides additional information about the topic.	Note: Selecting Delete permanently removes this contact from your address book.
Input	Text that you enter in the system.	I am writing you a text message.
Output	Text displayed by the system	Sorry, you have no messages.
Italics	File or document names	My Distribution List

1.4 Comments About the Documentation

Ericsson encourages you to provide feedback, comments, or suggestions so that we can improve the documentation to better meet your needs. With your comments, provide the following:

- Document title
- Document number and revision
- Page number

Please send your comments to your local Ericsson Support.



2 Overview

The Ericsson NETCONF Browser (ENB) is a tool that allows you to work with node configurations on remote systems using the NETCONF interface. The browser presents configuration data graphically, allowing you to navigate the configuration structure and perform add, modify, and delete operations on configuration elements.

ENB supports node configurations based on the proprietary Ericsson Common Information Model (ECIM). When working with ECIM compliant configurations, ENB is able to read and write configuration data. Configurations that are not ECIM compliant are presented in read-only mode.

2.1 Prerequisites

To fully interact with ECIM compliant nodes, ENB has the following prerequisites:

- COM 3.2+

If the node does not meet these prerequisites, ENB operates in read-only mode.

3 Getting Started

This section describes how to install, uninstall, and upgrade the Ericsson NETCONF Browser.

3.1 Prerequisites

The Ericsson NETCONF Browser is available for Microsoft Windows and Linux.

The minimum software requirements for installing ENB are as follows:

Microsoft Windows (32/64 bit)

Microsoft Windows 7 or 8

Linux (32/64 bit)

ENB supports the following Linux distributions:

- Fedora Linux 17+



- Debian 7+
- openSUSE 12.2+
- SLES 12+
- Ubuntu 12.04+

Other distributions may be supported if they contain the following dependencies:

- gtk2 2.18+
- NSS_3.14.3+
- libgio-2.0.so.0
- libdbus-1.so.3
- libcap.so.2
- libstdc++ shared object with GLIBCXX_3.4.15

Java

Java Runtime Environment (JRE) 7+

3.2 Working in Microsoft Windows

3.2.1 Installing ENB

For Windows, ENB is delivered as a self-installing executable.

To install ENB:

1. Move the ENB installation file to your Windows machine.
2. Run the installation file and follow the prompts in the Setup Wizard.

ENB is installed.

ENB requires a Java entry on the system PATH.

To verify that the Java executable is on the PATH, open a Command Prompt window and execute `java -version`.

If the command completes successfully, Java is on the PATH and no additional action is required. If the command does not complete, update the system PATH by following the instructions on <https://java.com/en/download/help/path.xml>.



3.2.2 Uninstalling ENB

Use **Programs and Features** on the Windows Control Panel to remove the ENB software.

To preserve your ENB settings between installations, the ENB uninstaller does not remove the %HOMEPATH%\ .enb directory containing logs, settings, and any local schema information downloaded from nodes. If required, completely remove ENB by deleting %HOMEPATH%\ .enb.

3.2.3 Upgrading ENB

ENB supports an in-place upgrade on Windows.

To upgrade ENB:

1. Move the new ENB installation file to your Windows machine.
2. Run the installation file and follow the prompts in the Setup Wizard.

ENB is upgraded.

3.3 Working in Linux

3.3.1 Installing ENB

For Linux, ENB is delivered as a gzip compressed tar archive. Two Linux distributions are available, one for 32 bit, and one for 64 bit platforms.

To install ENB:

1. On your Linux machine, create an installation folder for ENB.
2. Copy the correct ENB distribution file to the installation folder.
3. In the ENB installation folder, decompress the software package:

```
tar zxvf <ENB distribution file>
```

The ENB files are extracted to a subdirectory called enb_<version> and installation is complete.

ENB requires a Java entry on the system PATH.

To verify that the Java executable is on the PATH, execute **java -version**.

If the command completes successfully, Java is on the PATH and no additional action is required. If the command does not complete, update the system PATH by following the instructions on <https://java.com/en/download/help/path.xml>.



To run ENB, navigate to the ENB software directory and execute `./enb.sh`. If `enb.sh` lacks permission to execute, make the file executable as follows:

```
chmod u+x enb.sh
```

At first execution, ENB will create a hidden folder in the current user's home directory called `~/.enb`. This folder contains logs, settings, and any local schema information downloaded from nodes.

3.3.2 Uninstalling ENB

To uninstall ENB:

1. On your Linux machine, close the ENB application.
2. Navigate to the ENB installation folder.
3. Remove the `enb_<version>` subdirectory from the installation folder:

```
rm -rf enb_<version>
```

4. If required, completely remove ENB by deleting the `~/.enb` directory containing logs, settings, and any local schema information downloaded from nodes.

Note: Keeping this folder allows you to preserve your ENB settings between installations.

3.3.3 Upgrading ENB

To upgrade ENB:

1. On your Linux machine, copy the new distribution file to the ENB installation folder.
2. In the ENB installation folder, decompress the software package:

```
tar zxvf <ENB distribution file>
```

The ENB files are extracted to a subdirectory called `enb_<version>` and upgrade is complete.

Note: Multiple versions of ENB can be installed simultaneously.

To run ENB, navigate to the new ENB software directory and execute `./enb.sh`. If `enb.sh` lacks permission to execute, make the file executable as follows:

```
chmod u+x enb.sh
```

Logs, settings, and any local schema information is preserved between installations.



4 Working with Node Connections

A node connection defines the NETCONF interface for communication with a remote node. These connections allow ENB to download schema and configuration data to work with.

Working with node connections involves the following tasks:

- Creating a New Connection
- Editing a Connection
- Duplicating a Connection
- Deleting a Connection

4.1 The Connections Workspace

The **Connections** workspace allows you to work with NETCONF connections in ENB.

To access the **Connections** workspace, select the **Connections** tab in ENB. See Figure 1.

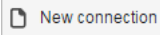
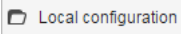
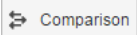



Note: This workspace is opened automatically when ENB is started.



Figure 1 The Connections Workspace

The following table describes the elements of the **Connections** workspace:

Table 3 Connections Workspace Elements

Element	Description
 New connection	Creates a new connection to a remote NETCONF interface.
 Local configuration	Opens a local configuration package that was previously saved from ENB.
 Comparison	Compares the contents of node configurations.
 Edit	Edits the node connection settings.
 Duplicate	Duplicates (copies) the node connection settings to a new connection.
 Delete	Deletes the connection.



4.1.1 Hot Keys

The following keyboard shortcuts are available in the **Connections** workspace:

Table 4 Connections Workspace Hot Key Combinations

Key Combination	Action
Alt+D	Duplicate the selected connection.
Alt+E	Edit the selected connection.
Ctrl+L	Load a local configuration.
Ctrl+N	Create a new connection.
Ctrl+X	Open the Comparison workspace.

4.2 Creating a New Connection

Creating a new connection allows you to define a new NETCONF interface to communicate with.

Creating a new connection involves the following tasks:

1. Setting up the node connection
2. If required, configuring schema retrieval
3. If required, applying a NETCONF filter
4. If required, setting up a tunnel end point

4.2.1 Setup the Node Connection

To establish a connection to the NETCONF interface on a node, ENB requires network information and access credentials.

To setup a node connection:

1. In the **Connections** workspace, click **New connection** or press **Ctrl+N**.

The **Connection** dialog box opens on the **NETCONF Interface** tab. See Figure 2.

Figure 2 Connection Dialog Box

The following table describes the elements of the **NETCONF Interface** tab:

Table 5 NETCONF Interface Elements

Element	Description
Name	The connection name.
Connect directly Connect through tunnel	Sets the type of network connection to the node. <ul style="list-style-type: none"> • Connect directly attempts to establish a direct connection between the client machine and the node. • Connect through tunnel routes the connection through a tunnel end point. When selected, the tunnel end point must be defined on the Tunnelling tab.



Element	Description
Use NETCONF subsystem	<p>Directly access the NETCONF subsystem over a SSH connection with the following syntax:</p> <pre>ssh -p 830 <user>@<host> -s netconf</pre> <p>Systems that use Common Operation and Maintenance (COM) 5.1+ may block incoming connections that attempt to access the NETCONF subsystem directly.</p>
Host	The IP address or hostname of the Operations, Administration and Maintenance (OAM) interface on the node.
NETCONF port	The communication port number.
User name	The account user name to access the node.
Password	The account password.
Fetch method	<p>The fetch method to retrieve information from the node.</p> <ul style="list-style-type: none"> • get retrieves configuration data and node state information. • get-config only retrieves configuration data.

2. Enter the required information. See Table 5.
3. Perform the following procedures, if required:
 - Configure Schema Retrieval
 - Apply a NETCONF filter
 - Setup a Tunnel End Point
4. Perform one of the following options:
 - a. Click **Save** to save the new connection to the **Connections** list.
 - b. Click **Fetch** to retrieve the configuration data without saving.

4.2.2 Configure Schema Retrieval

Configuration schemas are templates containing all possible classes, attributes, relationships, and constraints that are part of the configuration of a specific node.

ENB requires all elements in a node configuration to adhere to the constraints laid out in the associated schema. This governance ensures that the elements contained within a configuration are understood by the supported nodes.

ENB must have access to a configuration schema to perform the following operations:

- Add, modify, or delete configuration elements
- Display meta-model information
- Push a configuration to the node

ENB allows you to configure the schema retrieval options for each node connection.

To configure schema retrieval:

1. In the **Connection** dialog box, click the **Schema Retrieval** tab.

The **Schema Retrieval** tab opens. See Figure 3.

Figure 3 Schema Retrieval Tab

The following table describes the elements of the **Schema Retrieval** tab:

Table 6 Schema Retrieval Elements

Element	Description
Automatically retrieve meta-model information from the node	Default Retrieve schema information using automatic settings.



Element	Description
Retrieve meta-model information from local disk	Retrieve schema information from a local directory. The directory location must contain all required schema files.
Retrieve meta-model information from an external server	<p>Retrieve schema information from a directory on a remote server. The directory location must contain all required schema files.</p> <p>An external connection requires the following settings:</p> <ul style="list-style-type: none"> • Host • SSH port • User name • Password • Schema path • Connect directly / Connect through tunnel
Do not retrieve meta-model information (read-only mode)	Do not retrieve schema information. The node configuration is opened as read-only.

2. Enter the required information. See Table 6.

4.2.3 Apply a NETCONF Filter

ENB allows advanced users to specify a NETCONF filter that restricts the configuration data retrieved from the node.

To apply a NETCONF filter:

1. In the **Connection** dialog box, click the **NETCONF Filter** tab.

The **NETCONF Filter** tab opens. See Figure 4.



Figure 4 NETCONF Filter Tab

2. Input the required NETCONF filter. The following examples show sample filters.

This filter gets all parameters under the CSCF branch for a CSCF node:

```
<ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
  <managedElementId>jambala</managedElementId>
  <CscfFunction xmlns="urn:com:ericsson:ecim:CscfFunction">
    <applicationName>1</applicationName>
    <CSCF-Application xmlns="urn:com:ericsson:ecim:CSCF">
      <applicationName>CSCF</applicationName>
    </CSCF-Application>
  </CscfFunction>
</ManagedElement>
```

This filter gets a specific parameter under the CSCF branch:

```
<ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
  <managedElementId>jambala</managedElementId>
  <CscfFunction xmlns="urn:com:ericsson:ecim:CscfFunction">
    <applicationName>1</applicationName>
    <CSCF-Application xmlns="urn:com:ericsson:ecim:CSCF">
      <applicationName>CSCF</applicationName>
      <cscfAdministrativeState></cscfAdministrativeState>
    </CSCF-Application>
  </CscfFunction>
</ManagedElement>
```



4.2.4 Setup the Tunnel End Point

If required, ENB allows you to specify a secure tunnel end point for the node connection. A tunnel is required when the target node cannot be reached by ENB because it is on a different network.

Note: ENB only supports tunnel connections with a single hop between ENB and the target node.

To specify a tunnel end point:

1. In the **Connection** dialog box, click the **Tunneling** tab.

Note: The **Tunneling** tab only appears when **Connect through tunnel** is selected on the **NETCONF Interface** tab.

The **Tunneling** tab opens. See Figure 5.

Figure 5 Tunneling Tab

The following table describes the elements of the **Tunneling** tab:

Table 7 Tunneling Elements

Element	Description
Host	The IP address or hostname of the Operations, Administration and Maintenance (OAM) interface on the node.
SSH port	The Secure Shell (SSH) port number.
User name	The account user name to access the tunnel end point.
Password	The account password.



2. Enter the required information. See Table 7.

5 Working with Node Configurations

Node configurations are a collection of parameters groups and parameters that configure the software on a node. These elements (collectively called configuration data) are structured in accordance with a configuration schema that defines the legitimate structure of the configuration. ENB provides a dedicated workspace for configuration data. The configuration browser is described in Section 5.1 on page 16.

Working with node configurations involves the following tasks:

- Working with Parameter Groups and Structs
- Working with Parameters

5.1 The Configuration Browser

The configuration browser allows you to work with configuration data in ENB.

To access the configuration browser, select a connection to open in the **Connections** workspace.

Note: If the connection is not present, create a new one as described in Section 4.2 on page 9.

ENB establishes the connection and downloads schema and configuration data from the target node. After retrieving the required data, the node configuration is opened on a separate tab.

In the browser, configuration data is displayed as a cascading tree of configuration elements. Information is divided between two frames. The left frame shows a list of parameter groups and parameter structs. The right frame displays the parameters or struct members contained within a parameter group or parameter struct. Selecting a configuration element in the left frame displays its contents in the right frame.

Parameters or struct members are presented in a table with the following information:

- Name
- Value



See Figure 6.

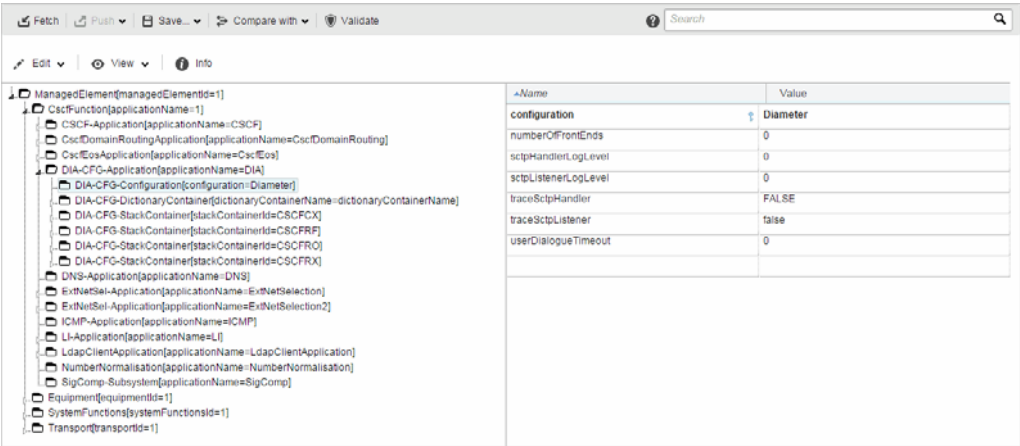


Figure 6 The Configuration Browser

The following table describes the elements of the configuration browser:

Table 8 Configuration Browser Elements

Element	Description
	Refreshes the current configuration with the latest configuration data from the node. All local changes are lost.
	Updates the node configuration by pushing local changes. Advanced Opens the Edit configuration dialog box, allowing you to edit the configuration changes in XML format before pushing them to the node. Use to open a local XML file containing configuration changes.
	Full configuration bundle Saves the current configuration, including all changes, to a ZIP archive. This file can be opened in ENB and used for comparison. Changes as template Saves all configuration changes as an XML snippet in a template file.



Table 8 Configuration Browser Elements

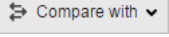
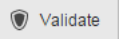



Element	Description
	<p>Compares the current configuration with another in ENB. For more information on comparing node configurations, refer to Section 6 on page 32.</p> <p>Use local file</p> <p>Compare the current configuration with a saved configuration.</p> <p>Advanced</p> <p>Opens the Comparison workspace.</p>
	<p>Validates the local configuration to ensure it conforms to the schema definition. For more information on validating node configurations, refer to Section 5.4 on page 30.</p>



Table 8 Configuration Browser Elements

Element	Description
 Edit ▼	<p>Add parameter group</p> <p>Adds a new parameter group to the configuration at the current location.</p> <p>Add parameter struct</p> <p>Adds a new parameter struct to the configuration at the current location.</p> <p>Delete parameter group</p> <p>Deletes the selected parameter group or parameter struct from the local configuration.</p> <p>Undelete parameter group</p> <p>Restores a deleted parameter group or parameter struct to the local configuration.</p> <p>Add parameter</p> <p>Adds a new parameter instance to the configuration at the current location.</p> <p>Delete parameter</p> <p>Deletes the selected parameter instance from the local configuration.</p> <p>Revert changed value</p> <p>Reverts configuration changes to the selected parameter instance.</p> <p>Revert all value changes</p> <p>Reverts all configuration changes in the selected parameter group.</p>
 View ▼	<p>Expand All</p> <p>Expands all elements in the configuration tree.</p> <p>Collapse All</p> <p>Collapses all elements in the configuration tree.</p>
 Info	<p>Displays information about the selected configuration element that is stored in the schema.</p>



5.1.1 Hot Keys

The following keyboard shortcuts are available in the configuration browser:

Table 9 Configuration Browser Hot Key Combinations


Key Combination	Action
Alt+C	Collapse all nodes in the configuration tree.
Alt+E	Open the Edit menu.
Alt+G	Add a parameter group instance at the selected location.
Alt+I	Display schema information for the selected element.
Alt+O	Expand all nodes in the configuration tree.
Alt+P	Add a parameter instance at the selected location.
Alt+S	Add a parameter struct instance at the selected location.
Ctrl+F	Search the configuration data. The following hot keys can be used to navigate the search results: <ul style="list-style-type: none">• F3 - Move to the next search hit.• Shift+F3 - Move to the previous search hit.
Ctrl+S	Save the configuration to a local file.
Ctrl+X	Open the Comparison workspace.
Ctrl+Z	Revert configuration changes to the selected element.
Ctrl+Shift+Z	Revert all configuration changes in the selected parameter group.

5.1.2 Color Use

When working with the configuration browser, colors are used to provide extra information about the configuration elements as follows:




Table 10 Color Use in the Configuration Browser

Color	Sample Representation	Description
Black	<code>bcfOperationalState</code>	Configuration elements that have been fetched from the node.
Bold Black	<code>applicationName</code>	Primary Keys.
Blue	<code>productIdentity[0]</code>	New configuration elements that were added to the local configuration.
Green	 <code>CscfFunction[applicationName=1]</code>	Parameter groups or parameter structs that contain configuration changes.
Red	<code>DNS-Application[applicationName=DNS]</code>	Deleted data references marking configuration elements that were removed from the local configuration.

5.1.3 Searching Node Configurations

The configuration browser includes a search functionality to locate specific configuration elements. By default, searches are performed against element names and values.


The search field accepts keywords and logic separators to narrow down the search results. A description of the search syntax and a list of advanced search options is available from the search help by clicking .

To perform a search:

In the configuration browser, click in the search bar or press **Ctrl+F**.

For a basic search, enter the name or value of the configuration element you are searching for.

Note: Search strings are not case-sensitive. All strings return a partial match unless they are surrounded by double quotes.

For a keyword search, enter a search string using one or more valid keywords and logical separators. Click  for a list of available options.



Note: Different keywords can be combined to refine the search pattern.

Some sample search queries, include:

- `trusted value="false"`

This query would search for parameters that have a name including `trusted` and a value of `false`.

- `trusted datatype=boolean`

This query would search for parameters that have a name including `trusted` and a data type of `boolean`.

- `name=trusted or value=::`

This query would search for parameters that have a name including `trusted` or a value including `::`.

- `not name=trusted`

This query would search for parameters or parameter groups with names that do not contain `trusted`.

- `(name=auth and iskey=true) or (name=e and value=900)`

This query would search for primary key parameters that have a name including `auth` or parameters that have a name including the letter `e` and a value of `900`.

All search hits for name or value are highlighted. Multiple hits are counted on the right side of the search bar. Cycle through multiple hits by pressing **Enter**, using the search hot keys, or the available arrows.

5.2 Working with Parameter Groups and Structs

Within the constraints of the schema, you can add or remove parameter groups and parameter structs.

5.2.1 Adding Parameters Groups

Where permitted by the configuration schema, the configuration browser can be used to add new parameter groups to a node configuration.

To add a new parameter group:

1. In the configuration browser, select a location in the configuration tree to add the new parameter group and click **Edit > Add parameter group** or press **Alt+G**.



If needed, you can perform a search for the required location as described in Section 5.1.3 on page 21.

The **Add New Parameter Group** dialog box opens. See Figure 7.

Figure 7 Add New Parameter Group Dialog Box

The following table describes the elements of the **Add New Parameter Group** dialog box:

Table 11 Add New Parameter Group Dialog Box Elements

Element	Description	Notes
Object Class	A list of valid parameter groups that can be added at this location. This list is derived from the schema definition and indicates what can be added at the selected level of the tree.	
Mandatory Parameters	A list of mandatory parameters in the selected object class. All mandatory parameters must be assigned a value that meets the schema constraints before the parameter group can be added.	Primary keys are identified by . Click next to any mandatory parameter to display the associated schema information.

- 2. Enter the required information. See Table 11.

ENB automatically validates the input for a new parameter group and displays the results, if applicable. Click next to any validation error for more information.

- 3. Click **OK**.

The parameter group is added to the local configuration data. This update must be pushed to the node before it can take effect. For more information on pushing configuration changes, see Section 5.5 on page 31.

5.2.2 Adding Parameter Structs

Where permitted by the configuration schema, the configuration browser can be used to add new parameter structs to a node configuration.

To add a new parameter struct:

1. In the configuration browser, select a location in the configuration tree to add the new parameter struct and click **Edit > Add parameter struct** or press **Alt+S**.

If needed, you can perform a search for the required location as described in Section 5.1.3 on page 21.

The **Add New Parameter Struct** dialog box opens. See Figure 8.

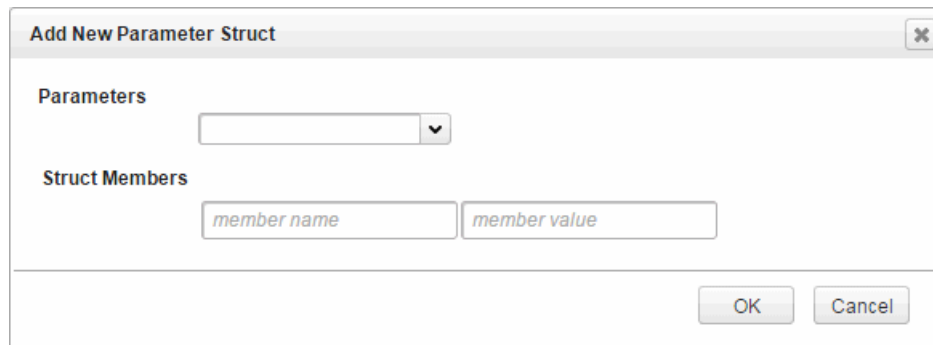
The dialog box is titled "Add New Parameter Struct" and has a close button (X) in the top right corner. It contains two main sections: "Parameters" and "Struct Members". The "Parameters" section has a dropdown menu. The "Struct Members" section has two input fields: "member name" and "member value". At the bottom right, there are "OK" and "Cancel" buttons.


Figure 8 Add New Parameter Struct Dialog Box

The following table describes the elements of the **Add New Parameter Struct** dialog box:

Table 12 Add New Parameter Struct Dialog Box Elements

Element	Description
Parameters	A list of valid parameter structs that can be added at this location. This list is derived from the schema definition and indicates what can be added at the selected level of the tree.
Struct Members	A list of members in the selected parameter struct. All struct members must be assigned a value in the corresponding field before the parameter struct can be added.

2. Enter the required information. See Table 11.

ENB automatically validates the input for a new parameter struct and displays the results, if applicable. Click  next to any validation error for more information.



3. Click **OK**.

The parameter struct is added to the local configuration data. This update must be pushed to the node before it can take effect. For more information on pushing configuration changes, see Section 5.5 on page 31.

5.2.3

Deleting Parameter Groups and Structs

The configuration browser can be used to delete parameter groups and parameter structs from a node configuration.

Caution!

Deleting local changes that have not been pushed to the node is a permanent operation.

Deleting configuration elements that have been fetched from a node replaces them with a deleted data reference marked in red. These data references can be pushed to the node, making them permanent, or undeleted to restore the original data.

To delete a parameter group or parameter struct:

- In the configuration browser, select a parameter group or parameter struct to delete and click **Edit > Delete parameter group** or press **Delete**.

If needed, you can perform a search as described in Section 5.1.3 on page 21.

The selected parameter group and all nested configuration elements are marked for deletion.

Note: Any local changes that are captured by the delete operation are permanently removed from the configuration.

This update must be pushed to the node before it can take effect. For more information on pushing configuration changes, see Section 5.5 on page 31.

Undeleting Parameter Group and Structs

If required, you can undelete a parameter group or parameter struct that is marked for deletion. This operation must be performed before the deletion is pushed to the node.

To undelete a parameter group or parameter struct:

- In the configuration browser, select a parameter group or parameter struct that is marked for deletion and click **Edit > Undelete parameter group** or press **Ctrl+Z**.

The selected parameter group and all nested configuration elements are restored.

5.3 Working with Parameters

Within the constraints of the schema, you can add, modify, or remove configuration parameters.

5.3.1 Adding Parameter Instances

Where permitted by the configuration schema, the configuration browser can be used to add new parameter instances to a node configuration.

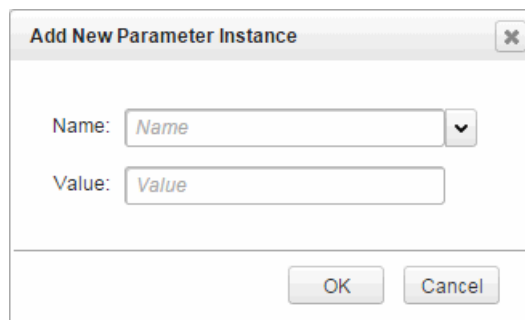
Parameter instances can be added using the **Add New Parameter Instance** dialog box, described here, or directly in the configuration browser using in-line editing. For more information on using in-line editing to add new parameter instances, refer to Section 5.3.1.1 on page 27.

To add a new parameter instance using the **Add New Parameter Instance** dialog box:

1. In the configuration browser, select a location in the configuration tree to add the new parameter instance and click **Edit > Add parameter** or press **Alt+P**.

If needed, you can perform a search for the required location as described in Section 5.1.3 on page 21.

The **Add New Parameter Instance** dialog box opens. See Figure 9.



The dialog box titled "Add New Parameter Instance" has a close button (X) in the top right corner. It contains two input fields: "Name:" with a text box containing "Name" and a dropdown arrow, and "Value:" with a text box containing "Value". At the bottom, there are "OK" and "Cancel" buttons.

Figure 9 Add New Parameter Instance Dialog Box


The following table describes the elements of the **Add New Parameter Instance** dialog box:



Table 13 Add New Parameter Instance Dialog Box Elements

Element	Description
Name	A list of valid parameter instances that can be added at this location. This list is derived from the schema definition and indicates what can be added at the selected level of the tree.
Value	<p>The parameter value. Where parameters have predefined values (such as true or false), the possible settings can be selected from a list.</p> <p>All parameters must be assigned a value before they can be added.</p>

2. Enter the required information. See Table 13.

ENB automatically validates the input for a new parameter instance and displays the results, if applicable. Click  next to any validation error for more information.

3. Click **OK**.

The parameter instance is added to the local configuration data. This update must be pushed to the node before it can take effect. For more information on pushing configuration changes, see Section 5.5 on page 31.

5.3.1.1

Adding Parameter Instances Using In-Line Editing

The configuration browser can be used to add new parameter instances directly to a parameter group using in-line editing.

To add parameter instance using in-line editing:

1. In the configuration browser, select a location in the configuration tree to add the new parameter instance.

If needed, you can perform a search for the required location as described in Section 5.1.3 on page 21.

The table in the right frame is populated with existing parameters in the selected parameter group. See Figure 10.

Name	Value
applicationName	CscfDomainRouting
cscfDomainRoutingActiveConfig	default
cscfDomainRoutingActiveMatchingTable	0
cscfDomainRoutingEnabled	false
cscfDomainRoutingSelectedConfig	default
cscfDomainRoutingSyncConfig	false
cscfDomainRoutingSyncState	synchronized

Figure 10 Parameters in a Parameter Group

2. Select the empty row at the bottom of the table and input the parameter name in the **Name** column.

Typing part of a name displays a list of matching parameters that can be added at this location. Click to display the complete list. See Figure 11.

The parameter instance is added to the local configuration data with a default value.

cscfDomainRoutingEnabled
cscfDomainRoutingSelectedConfig
cscfDomainRoutingSyncConfig
cscfDomainRoutingSyncState
cscfDomainRoutingActiveConfig
cscfDomainRoutingActiveMatchingTable

Figure 11 Parameter Autocomplete

3. If required, update the parameter's default value in the **Value** column. Where parameters have predefined values (such as true or false), the possible settings can be selected from a list by clicking .

ENB automatically validates the input for a new parameter instance and displays the results, if applicable.

This update must be pushed to the node before it can take effect. For more information on pushing configuration changes, see Section 5.5 on page 31.

5.3.2 Modifying Parameter Instances and Struct Members

The configuration browser can be used to modify the value of parameter instances and struct members using in-line editing.

To modify the value of a parameter instance or struct member:


1. In the configuration browser, select the parameter group or parameter struct containing elements to modify.



If needed, you can perform a search for the required location as described in Section 5.1.3 on page 21.

The table in the right frame is populated with the parameter instances or struct members.

2. Select the parameter instance or struct member to modify and input the new value in the **Value** column.

Where parameters have predefined values (such as true or false), the possible settings can be selected from a list by clicking .

ENB automatically validates the changes to parameter instances and struct members displaying the results, if applicable.

The parameter instance or struct member is updated with the new value.

This update must be pushed to the node before it can take effect. For more information on pushing configuration changes, see Section 5.5 on page 31.

5.3.3 Deleting Parameter Instances

The configuration browser can be used to delete parameter instances from a node configuration.

Caution!

Deleting local changes that have not been pushed to the node is a permanent operation.

Deleting configuration elements that have been fetched from a node replaces them with a deleted data reference marked in red. These data references can be pushed to the node, making them permanent, or reverted to restore the original data.

To delete a parameter instance:

- In the configuration browser, select a parameter instance to delete and click **Edit > Delete parameter** or press **Delete**.

If needed, you can perform a search as described in Section 5.1.3 on page 21.

The selected parameter instance is marked for deletion.

This update must be pushed to the node before it can take effect. For more information on pushing configuration changes, see Section 5.5 on page 31.



5.3.4 Reverting Configuration Changes

The configuration browser can be used to revert local changes to parameter instances that have not been pushed to the node.

Caution!

Reverting local changes is a permanent operation.

To revert configuration changes:

- In the configuration browser, select a parameter instance or struct member that is new, modified or deleted and click **Edit > Revert changed value** or press **Ctrl+Z**.

Use **Edit > Revert all value changes** or press **Ctrl+Shift+Z** to roll back all changes in the selected parameter group.

If needed, you can perform a search as described in Section 5.1.3 on page 21.

The selected changes are reverted.

5.4 Validating a Configuration

When ENB has access to the configuration schema, it can validate the local configuration to ensure it conforms to the schema definition. It is highly recommended to perform this check after any operation that impacts the configuration data.

ENB performs the following checks during schema validation:

Cardinality	Verifies that the number of instances of configuration elements within the local configuration is allowed by the schema.
Value Constraints	Verifies that the parameter values within the local configuration follow the type constraints and value patterns that are allowed by the schema.
Schematron Rules	Verifies that configuration elements within the local configuration obey any schematron rules laid out in the schema.



Issues discovered during the validation check are output to a plain text file that can be saved.

To validate a configuration:

- In the configuration browser, click **Validate**.

The local configuration is validated against the associated schema.

After validating the local configuration data, a **Validation report** opens.

5.5 Pushing Configuration Changes

All changes made to a node configuration using the configuration browser are local. Updates must be pushed back to the node before the changes can take effect.

Caution!

Pushing configuration changes is a permanent operation.

Multiple configuration changes can be pushed at the same time in a single operation.

To push configuration changes:

After updating the node configuration, click **Push**.

The configuration changes are pushed to the node.

ENB has an advanced push option that allows you to work with the raw XML data sent to the node. For more information on the advanced push, see Section 5.5.1 on page 31.

5.5.1 Advanced Push

All configuration changes pushed to the node are recorded in XML. ENB allows advanced users to work directly with the raw XML data.

To perform an advanced push:

1. In the configuration browser, click **Push > Advanced**.

The **NETCONF snippet** dialog box opens with the current changes recorded in XML format. See Figure 12.

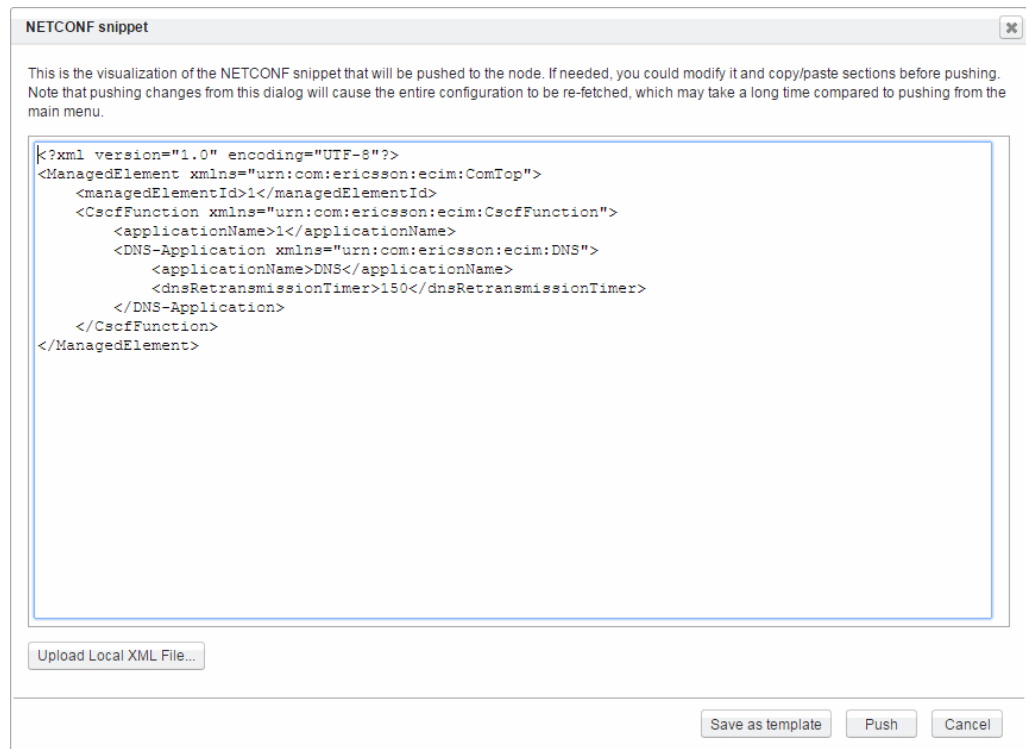


Figure 12 Edit Configuration Dialog Box

2. Perform one of the following operations:

- a. If required, update the raw XML and click **Push**.

The configuration changes are pushed to the node.

- b. Upload a local XML file with the required changes by clicking **Upload Local XML File**.

The configuration changes are loaded into the **Edit Configuration** dialog box and can be edited or pushed as required.

- c. Save the XML snippet as a template file by clicking **Save as template**.

6 Comparing Two Configurations

Using ENB you can compare the contents of two node configurations to identify the differences between them. Elements in both configurations are analyzed in a comparison and any differences in name, value, and location are identified.



Comparisons can be performed between two live nodes, two saved configurations, or between a live node and a saved configuration.

6.1 The Comparison Workspace

The Comparison workspace allows you to compare node configurations.

To access the **Comparison** workspace, perform one of the following steps:

- In the **Connections** workspace, click **Comparison** or press **Ctrl+X**.
- In the configuration browser, click **Compare with > Advanced** or press **Ctrl+X**.

The **Comparison** Workspace opens. See Figure 13.

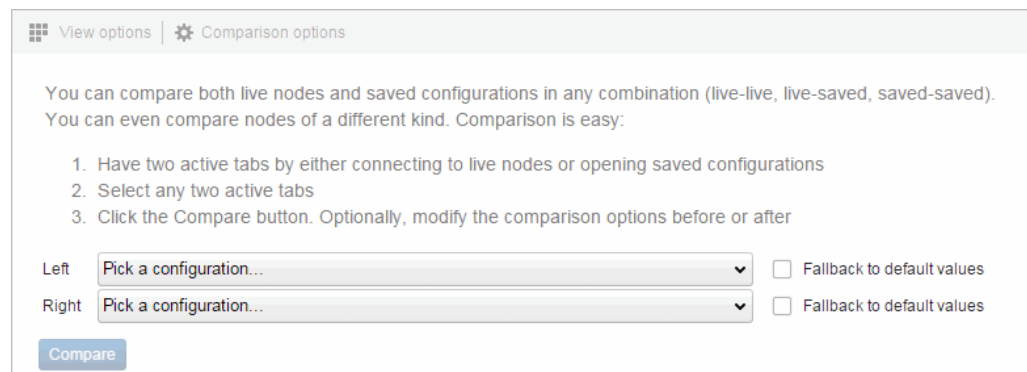


Figure 13 The Comparison Workspace

The workspace is divided into three areas as follows:

- | | |
|--------------------------------|---|
| View options | Allows you to control what information is initially displayed in the comparison report table by expanding or collapsing the list of configuration elements. |
| Comparison Options | Allows you to customize the comparison report by setting comparison options. For more information on comparison options, refer to Section 6.1.1 on page 33. |
| Configuration Selection | Allows you to select two open configurations to compare. For each configuration, you have the option to fallback to default values when a configuration element is defined in one configuration but is absent from the other. |

6.1.1 Customizing Comparison Options

ENB includes a set of options to customize the comparison report. These options help to restrict the overall number of differences identified in the report.

To set the comparison options:

1. In the **Comparison** workspace, select two configurations to compare.

Comparison options is enabled

2. Click **Comparison options**.

The comparison options are displayed, see Figure 14.

Scope

Both (default)

Left

Right

Read-only and case sensitivity

☐ Case insensitive

☐ Ignore read only parameters

☐ Automatically map differences in root elements

Managed Object Class (MOC)

Hold SHIFT or CTRL and click the left mouse button to select multiple MO classes. Use CTRL+A to select the complete list.

AuthenticationOrder
AuthorizationOrder
BcfNetworkInterface
BcfNwIfs
BrM
BrmBackup
BrmBackupInHousekeeping

Reset Apply

Figure 14 Comparison Options

The following table describes the different comparison options:



Table 14 Comparison Options

Element	Description
Scope	<p>Configures how ENB performs the comparison.</p> <ul style="list-style-type: none"> • Both (All Differences) Enabled by default, ENB examines all of the parameters and parameter groups within both configurations and reports all differences in the comparison. • Left (Differences Unique to the Left Config.) ENB examines all of the parameters and parameter groups within the configuration on the left and compares them against the configuration on the right. Only those differences unique to the left configuration are reported in the comparison. • Right (Differences Unique to the Right Config.) ENB examines all of the parameters and parameter groups within the configuration on the right and compares them against the configuration on the left. Only those differences unique to the right configuration are reported in the comparison.
Case insensitive	Ignores the letter case of parameter values.
Ignore read only parameters	Ignores differences in ReadOnly parameters.
Automatically map differences in root elements	If the root elements have the same name, but a different primary key, they are automatically considered a match in the comparison through mapping.
Managed Object Class (MOC)	This multi-select list box allows you to filter the comparison results and include only specific MOCs.

3. Enter the required information. See Table 14.

4. Click **Apply**.

ENB performs the comparison with the selection options and displays the comparison report.



6.1.2 The Comparison Report

Comparison results are output to a table that highlights the differences between both configurations. See Figure 15.

View options		Comparison options	
CSCF vIMS (CBA) 20150508-1537 MODIFIED.zip		CSCF vIMS (CBA)	
Name	Value	Name	Value
▼ ManagedElement[managedElementId=1] (12)		▼ ManagedElement[managedElementId=1] (12)	
▶ DIA-CFG-Application[applicationName=DIA] (6)		▶ DIA-CFG-Application[applicationName=DIA] (6)	
▼ SystemFunctions[systemFunctionId=1] (6)		▼ SystemFunctions[systemFunctionId=1] (6)	
▼ Fm[fmlId=2] (6)		▼ Fm[fmlId=1] (6)	
fmlId	2	fmlId	1
heartbeatInterval	55	heartbeatInterval	60
lastChanged	2015-02-13T19:11:42Z	lastChanged	2015-02-12T18:37:45Z
lastSequenceNo	1035668	lastSequenceNo	1035668
sumMinor	10	sumMinor	1
sumWarning	[missing]	sumWarning	5

Figure 15 Comparing Two Configurations

This configuration comparison report can be customized. For more information on the comparison options, refer to Section 6.1.1 on page 33.

When performing a comparison, ENB distinguishes between a Left Configuration and a Right Configuration. The Left Configuration serves as a starting point for the comparison. Data from the Left Configuration is compared against the Right Configuration. Both the Left and the Right Configurations can be a live node or a saved configuration.

The comparison report is presented as a cascading list of parameter groups, each group containing one or more parameter differences. The number of differences contained within a parameter group is printed next to it in parenthesis.

By default, this list of parameter groups is expanded to show all differences. Each level of the list can be collapsed to hide the enclosed changes. You can control the visibility of all elements simultaneously using **View options** at the top.






In the report, colored highlighting is used to designate types of differences. The following table describes color use in the configuration comparison report:

Table 15 Color Use in the Configuration Comparison Report

Color	Sample Representation	Description
Blue	<div></div>	A parameter instance with the same set of values exists in both configurations, but the values are in a different order.



Table 15 Color Use in the Configuration Comparison Report

Color	Sample Representation	Description
Green		Two parameter group instances with the same name, but different primary keys and different parameters are being compared through mapping. Parameter groups that can be mapped are identified by a  button in the configuration report. Click  to perform a mapping.
Orange		A parameter instance occupies the same place in both configurations, but has different values.
Yellow		A parameter instance in one configuration does not occupy the same location in the other configuration.

6.2 Performing a Comparison

In ENB, a configuration comparison can be initiated from the **Connections** workspace or from the configuration browser.

Compare from the Connections Workspace

1. In the **Connections** workspace, open two node configurations.
2. After the configurations have opened in separate tabs, click **Comparison** in the **Connections** workspace or press **Ctrl+X**.

The **Comparison** workspace opens on a new tab.

3. Enter the required information. See Section 6.1 on page 33.
4. Click **Compare**.

The selected configurations are compared and a configuration comparison report is generated with the results.

Compare from the Configuration Browser

In the configuration browser, select **Compare with** and one of the following options:



- **Use local file** opens a saved configuration for the comparison.
- **Advanced** or **Ctrl+X** opens the **Comparison** workspace.
- **<List of Open Configurations>** compare the current configuration with another open configuration.

When comparing configurations outside the **Comparison** workspace, the comparison is performed with default options. The comparison options can be updated from the comparison report by clicking **Comparison options**.