

# Ericsson Command-Line Interface User Guide

---

## USER GUIDE

**Copyright**

© Ericsson AB 2014, 2015. 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.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Related Information	1
1.2	Key Features of ECLI	1
<b>2</b>	<b>ECLI Modes</b>	<b>3</b>
<b>3</b>	<b>Start an ECLI Session</b>	<b>5</b>
<b>4</b>	<b>Use ECLI Online Features</b>	<b>7</b>
4.1	Static Help	7
4.2	Context-Sensitive Help	7
4.3	Auto-Completion	8
<b>5</b>	<b>Navigate in MIB</b>	<b>11</b>
<b>6</b>	<b>Display ME Status</b>	<b>13</b>
<b>7</b>	<b>Change ME Configuration</b>	<b>15</b>
<b>8</b>	<b>Execute MO Actions</b>	<b>17</b>





# 1 Introduction

This document provides an overview of the Ericsson Command-Line Interface (ECLI). The ECLI is a terminal-based CLI, which allows you to monitor and manage the Managed Element (ME). It is based on industry de facto standard patterns to make it easy for you to learn how to manage the system.

The ECLI enables you to interact with the Management Information Base (MIB) through common, generic-purpose commands.

## 1.1 Related Information

For a complete description about the ECLI, including descriptions of all commands, key combinations, error messages, character support, and examples, refer to Interwork Description *Ericsson Command-Line Interface*.

For information on the Managed Object Model (MOM), Managed Object Classes (MOCs), Managed Objects (MOs), and related concepts mentioned in this document, refer to *Managed Object Model User Guide*.

## 1.2 Key Features of ECLI

The key features of the ECLI are described in Table 1.

Table 1 Key Features of ECLI

Feature	Description
Access control	The result of the ECLI commands manipulating the MIB is subject to authentication and authorization. If you have no permission to access an MO instance or attribute, then operations behave as if the MO instance does not exist.
Auto-completion	By pressing the <b>Tab</b> key, all possible ECLI command completions are displayed and unique completions are added to the command line.
Context-sensitive help	By pressing the <b>?</b> key, a description of the ECLI command element is displayed.
ECLI modes	Two ECLI modes are supported. Exec mode is intended for observation and executing actions. Config mode is used for changing the ME configuration.
Model driven	The ECLI command elements and their properties are defined in the MOM as MOCs, attributes, and actions.
Navigation	The position in the MO tree can be changed. The position determines the context of the ECLI command.

*Table 1 Key Features of ECLI*

<b>Feature</b>	<b>Description</b>
Security	An ECLI session is running securely over SSH.
Transactions	Configuration changes are applied through atomic transactions. Thus, it is ensured that all or none of the operations are executed.



## 2 ECLI Modes

The ECLI provides the following two modes:

- Exec mode – Displays the status of the ME. In this mode, you enter commands to monitor the ME, display its configuration, and execute actions.
- Config mode – Used to change the ME configuration. In this mode, you start a configuration transaction to the MIB, enter commands to change the ME configuration, and commit the changes.

As shown in Figure 1, when you initiate an ECLI session, you always enter Exec mode, which is the default mode.

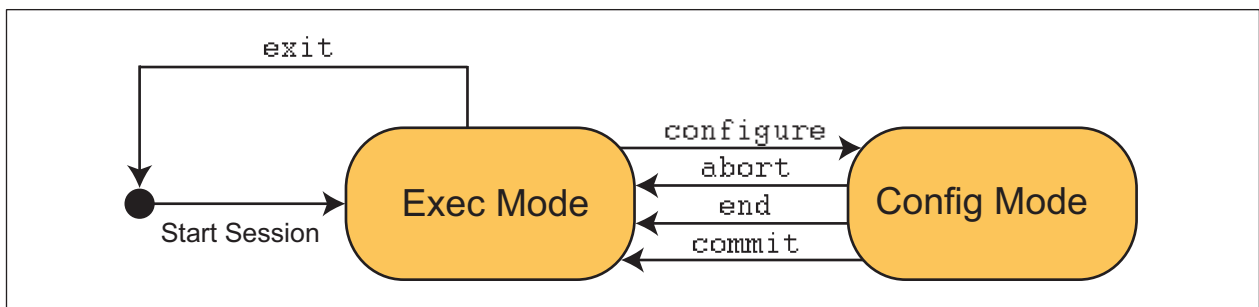


Figure 1 ECLI Modes

Some of the basic ECLI commands are described in Table 2.

Table 2 Basic ECLI Commands

Command	Description
<b>abort</b>	Discards the changes in the transaction, terminates the transaction, and returns to Exec mode.  When used with argument <b>-s</b> , the ECLI remains in Config mode and starts a new transaction instead of returning to Exec mode.
<b>commit</b>	Validates the transaction, commits the configuration changes, keeps the ECLI position, and returns to Exec mode.  When used with argument <b>-s</b> , the ECLI remains in Config mode and starts a new transaction instead of returning to Exec mode.
<b>configure</b>	Enters Config mode and initiates a new configuration transaction.
<b>end</b>	Returns from Config mode to Exec mode when there are no changes in the configuration transaction. Use command <b>abort</b> or <b>commit</b> if you wish to return to the Exec mode after entering configuration changes.
<b>exit</b>	Exits the ECLI session.





## 3 Start an ECLI Session

To start an ECLI session:

1. Use a terminal and start the SSH session.

Example of logon with OpenSSH client:

```
ssh <user>@<target_host> -p 22
```

The options are as follows:

- **<user>** – Username.
- **<target\_host>** – The OAM virtual IP address of the MO.
- **-p** (port number) – TCP port 22 is default.

Root user access is denied.

2. Wait for the session to start.





## 4 Use ECLI Online Features

The ECLI provides online and context-sensitive help. It enables you to access information and learn about the commands, the MIB, and the MOM without relying on the documentation library.

### 4.1 Static Help

Command `help` provides static help about the ECLI. The command is described in Table 3.

Table 3 Command Help

Command	Description
<code>help</code>	Displays a quick ECLI introduction and a summary of the main commands applicable to the ECLI mode (Exec or Config).

### 4.2 Context-Sensitive Help

By pressing the `?` key, you can request context-sensitive help on the ECLI commands and MOM elements (MOCs, attributes, and actions). The returned output depends on your input, the ECLI mode, and the ECLI position. Online help is only available for commands and model elements that can be auto-completed in the current ECLI mode.

The following examples show how to use the `?` key to trigger context-sensitive help:

```
>show?  
show      Command  
Display information  
  
>show ?
```



```
--moc          Option to select a specific child
                MOC under the current DN
--recursive    Display all information
--sort         Sort the MO instances in
                numerical/alphabetical order
--verbose      Display verbose information
-m            Option to select a specific child
                MOC under the current DN
-r            Display all information
-s            Sort the MO instances in
                numerical/alphabetical order
-v            Display verbose information
+ManagedElement The top-level class in the Common
                Information Model is Managed Element
                root Managed Object Class.
```

```
>show -v -?
```

```
--recursive    Display all information
-r            Display all information
```

```
(config-ManagedElement=NODE06ST)>userLabel?
```

```
userLabel      String [optional]
A freetext string for additional information to assist
Managed Element identification.
```

## 4.3 Auto-Completion

The **Tab** key can be used at any point on the command line to either auto-complete the line or (if more than one option for completion exists) display all the valid options matching what has been typed so far, for example:

```
>dn ManTab
```

```
>dn ManagedElement=NODE06ST
```

```
>dn ManagedElement=NODE06ST,Tab
```

```
CSCF-Application=CSCF
CscfDomainRoutingApplication=CscfDomainRouting
CscfEosApplication=CscfEos
DIA-CFG-Application=DIA
DNS-Application=DNS
ExtNetSel-Application
LdapClientApplication=LdapClientApplication
NumberNormalisation=NumberNormalisation
SigComp-Subsystem=SigComp
SystemFunctions=1
Transport=1
```

```
>dn ManagedElement=NODE06ST,
```

```
>dn ManagedElement=NODE06ST,cscfTab
```



```
CSCF-Application=CSCF  
CscfDomainRoutingApplication=CscfDomainRouting  
CscfEosApplication=CscfEos  
>dn ManagedElement=NODE06ST,cscf
```





## 5 Navigate in MIB

Navigation allows you to change the ECLI position without changing the configuration.

The commands described in Table 4 are used to navigate in the MIB.

*Table 4 Navigation Commands*

Command	Description
<code>&lt;RDN&gt;</code>	Changes the ECLI position to the Relative Distinguished Name (RDN), which is the address of an MO instance in relation to its parent MO.  Example: <code>&gt;ManagedElement=NODE06ST</code>
<code>dn</code>	Navigates to any position in the MIB.  Example: <code>&gt;dn ManagedElement=NODE06ST, SystemFunctions=1</code>
<code>back [-h]</code>	Navigates back to the previous position in the MO tree. With option <code>-h</code> , the 10 previous positions in the MO tree are listed without changing the current position.
<code>show-dn</code>	Displays your position in the MIB.
<code>top</code>	Changes the ECLI position to the root position in the MIB.
<code>up</code>	Changes the ECLI position to the parent MO.
<code>..</code>	Changes the ECLI position to the parent MO.  Command <code>..</code> can be part of the RDN navigation command, for example, <code>(SysM=1) &gt;.., Fm=1</code> .

**Note:** In Config mode, command `<RDN>` creates the corresponding MO if it does not exist.





## 6 Display ME Status

The commands described in Table 5 are used to display the ME status.

*Table 5 Commands to Display ME Status*

Command	Description
<b>show</b>	Displays the configuration and state information as MO properties.  Without argument, only those attributes are displayed that have a non-default value assigned.  With argument <b>-v</b> , all attributes are displayed.  With argument <b>-r</b> child MO instances are displayed recursively.
<b>show-table</b>	Displays configuration and state information in tabular format.
<b>show-config</b>	Displays the output in configuration format. The format is also a valid input for the ECLI. Thus, copy/paste or terminal input/output redirection allows configuration copy.





## 7 Change ME Configuration

You can change the ME configuration by explicitly creating MOs, updating MOs (setting and changing MO attributes), and deleting MOs.



### Attention!

Risk of data loss or data corruption.

Ensure to commit the changes to make them effective.

The commands described in Table 6 are used to change the ME configuration and can only be used in Config mode.

*Table 6 Commands to Change ME Configuration*

Command	Description
<b>&lt;RDN&gt;</b> Example: (config-Snmp=1) > <b>SnmpTargetV1=OSS_v1</b>	Creates an MO if it does not exist.
<b>&lt;attribute_name&gt;=&lt;attribute_value&gt;</b> Example: (config-SnmpTargetV1=OSS_v1) > <b>address=192.0.2.10</b>	Sets or changes a simple attribute value.
<b>no &lt;attribute_name&gt;</b> Example: (SnmpTargetV2C) > <b>no informRetryCount</b>	Deletes an attribute value.
<b>no &lt;RDN&gt;</b> Example: (config-Snmp=1) > <b>no SnmpTargetV3=OSS_v3</b>	Deletes an MO.





## 8 Execute MO Actions

An MO action is a command that you can use only when the ECLI is positioned at the proper MO. Actions are used to trigger MO-specific operations.



### Attention!

Risk of data loss or data corruption.

Unless stated otherwise, only execute actions in Exec mode to minimize the risk to misconfigure the system.

The command syntax to execute actions is described in Table 7.

Table 7 Command to Execute Actions

Command	Description
<pre>[.,] &lt;action_name&gt; [--&lt;action_parameter_name&gt; &lt;action_parameter_value&gt;] ...</pre> <p>Example: (Schema=2) <b>export --uri sftp://user@10.64.8</b>  <b>8.91/tmp/ --password userpwd</b></p>	Executes an action.

If `<action_name>` is conflicting with any command name, give the action request as `'.,<action_name>'` if the action must be executed from the current Distinguished Name.