ERICSSON

Operations Procedure Support

Operating Instructions

1/1543-CNA 403 3474 Uen D

Contents

1 Introduction

This chapter contains the following parts:

— Purpose

— Target Groups

— Prerequisites

— Related Documents

— Graphic Conventions

Purpose

This document describes the Operations Procedure Support (OPS) client directory structure, environment variables, data files and the OPS client configuration file ops.ini .

It contains descriptions and instruction for the standard maintenance and corrective maintenance procedures for the OPS system administrator.

Target Groups

This document is intended for OPS System Administrators.

1.3 Prerequisites

You must be having the following roles:

1. Ops_Operator
2. WinFIOL_Operator
3. Element_Manager_Operator
4. SystemAdministrator
5. SystemSecurityAdministrator
6. BscApplicationAdministrator
7. Scripting_Operator
8. Cmedit_Operator
9. FIELD_TECHNICIAN

Note: Other roles may be required depending on what the OPS script aims to do. For example: node roles for actions towards NEs. Refer to the User Management User Guide for instructions on creating a user/role and setting a password.

It is assumed that you are familiar with the following:

— OPS Graphical User Interface (GUI)

— LINUX terminology and standard procedures

Related Documents

S.No.	Doc Name	Available where?
1	Operations Procedure Support User Guide	This document is available in the Ericsson Network Manager, ENM, AOM 901 151library.
2	Operations Procedure Support Scripts Migration Guide	This document is available in the Ericsson Network Manager, ENM, AOM 901 151library.

Graphic Conventions

The following graphic conventions represent directory structures:

— A directory is represented by a box containing the directory name

— A file is represented by an ellipse containing the file name

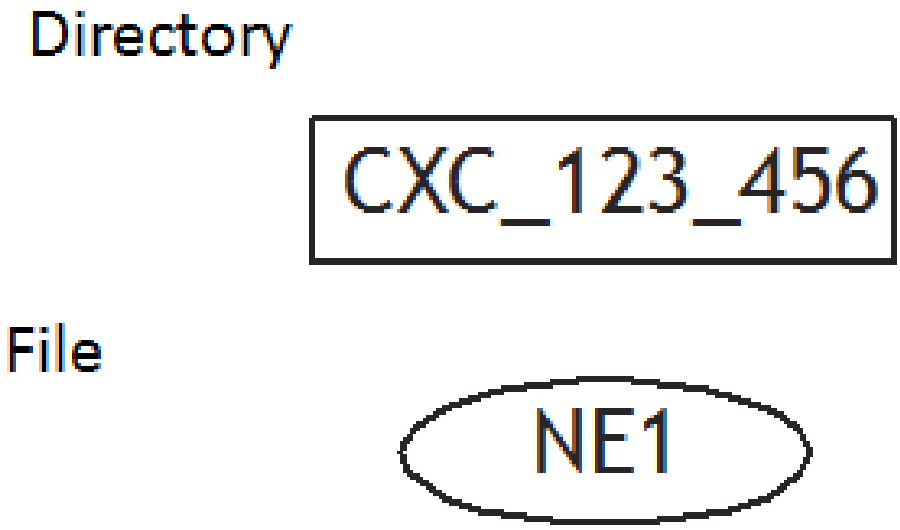

Figure 1 OPS Directory Structure Convention

2 OPS Client Reference

This chapter contains the following:

— Directory structure

— OPS GUI start script

— Parameter File

— Environment variables for Client (LINUX)

— Enabling and Disabling Trace

— OPS NUI Start Script

— OPS NUI Rest call

2.1 Directory Structure

When the OPS client is running, most of the information needed is read from opt/ops/ Directory on OPS VM:

— /opt/ops where the OPS executable code, configuration files, and dependent libraries are stored.

These directories are defined at start up of the OPS client through one of the following:

— Parameters to the OPS start script,

— In a parameter file read by the script

— By environment variables

The default values for the parameters are defined at installation of the OPS client.

/opt/ops Directory Structure

The structure of the opt/ops directory is shown in the following figure. The default location is /opt/ops. This is where the static data like scripts, executable code, language files, are stored. The directory is created by the installation procedure and there must be no need for the System Administrator to change any files in this directory. The OPS client is configured by providing information in configuration files named ops.ini. This file is stored in the etc directory (/opt/ops/ops_client/etc/ops.ini)

The figure does not show all details of the directory structure.

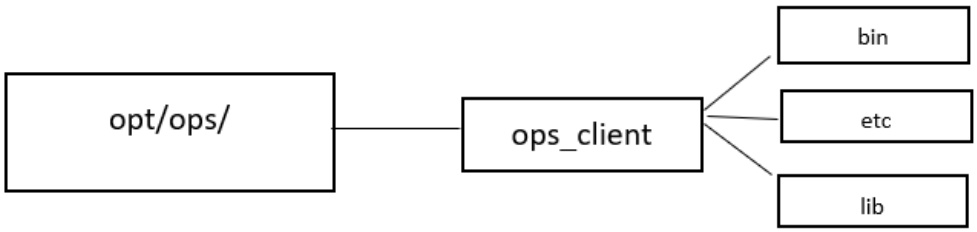

Figure 2 Directory Structure for ops_client

— bin

— ops_gui.sh

— ops_nui.sh

— etc

— logging.properties

— ops.ini

— lib

This directory contains Visibroker and client jar files

2.2 OPS GUI Start Script

Script to ops_gui exist in /opt/ops/ops_client/bin on any OPS VM. The OPS client is started by the following command:

/opt/ops/ops_client/bin/ops_gui

Note: User must be having the following roles:

1. Ops_Operator
2. WinFIOL_Operator
3. Element_manager_Operator roles.

The command line syntax of ops_gui is as follows:

ops_gui [-mail user1 [,user2 ... ,userN]] [-file command file name] [-es n] [-mode m] [-startlabe l] [-autoquit] [-routefile file] [-routemail user]

[-D varname=varvalue]-user <user> -pwd <password>

If a command option is not specified, the start script tries to get the value from an environment variable. A variable can be given a value in the parameter file in which case an existing value from the environment is overridden if no command option, parameter file setting or environment variable is defined, a default value, if defined in the script, is used. Some important options are:

See also Section 5.1 for information about environment variables that are used by the OPS client.

2.3 OPS NUI Start Script

The command line syntax is as follows:

ops_nui [-mail user1 [,user2 ... , userN]] [-es external system name] [-startlabel label] -file command file name [-routefile log file name]

[-routemail user] [-D varname=varvalue] [-loghistory nrOfLines][-sessionid sessionid]

-mail user1 [,user2 ... ,userN] Specifies users who receive mail with the execution result.

-es external system name An external system to which a connection is established before the command file is executed.

-startlabel label Specifies a label in the command file at which the execution starts.

-file command file name The command file to be executed (mandatory).

-routefile file Specifies a file in which all information displayed in the Command File Output area of the main window is logged. (This is equivalent to executing the @LOG command).

-routemail user Specifies a user who receives mail with all information displayed in the Command File Output area of the main window.

-D varname=varvalue Sets a variable to a value in the script that is to be interpreted (This is combined with the use of -file option).

-loghistory nrOfLines Specifies that the output from the script must be stored in a file. This option must be used if a user wants to switch from NUI-mode to GUI-mode without loosing the log history. The nrOfLines variable can be used to specify how much of the log history that must be stored in the GUI output window. This option is always activated if OPS is started from SHM job thread with nrOfLines set to 10000. This option is not activated if OPS is started from the job manager or the activity manager.

-sessionid sessionid OPS GUI to be launched with session id created by OPS NUI

2.4 Parameter File

After installation scripts/parameter files are created in bin directory containing environment variables . Files are present at below location /opt/ops/ops_client/ .

The configuration files contain the following data:

2.5 Environment variables for Client (LINUX)

CLASSPATH : Paths to Java jar files used by OPS client.

OPS_CORBA_HOST : Host name of OPS server to bind to.

OPS_HOME :Home directory for OPS client.

LD_LIBRARY_PATH: Path for run time library search.

2.6 Enabling and Disabling Trace

The clients (NUI and GUI) can be configured to provide detailed traces.

These can be useful for debugging the OPS Client.

The trace configuration file for the client is located under /opt/ericsson/nms_ops_client/etc/logging.properties.

To enable traces, the file needs to be edited. For more details about OPS logging, see OPS DCG Document.

Note: Loggers are attached to packages, where the level for each package is specified. The global level is used by default, so levels specified here simply act as an override of package-level settings.

The following logging variables can be set:

Table 1 Logging Parameters

Parameter	Default value	Comments and Recommendations
com.ericsson.level	OFF	Change this level to ALL to produce OPS trace.
.level	OFF	Default global log level. It is not recommended to change the default value because library activities, such as java.awt and other 3PP, are also controlled by this option. The amount of logging activity created by this option causes load on the system. It is recommended to set com.ericsson.level instead, which controls the log level.

com.ericsson.nms.ops.util.StdoutConsoleHandler.level	ALL	This parameter makes it possible to print trace information to standard output. If this parameter is set to OFF, nothing is displayed on the console, even if the level and com.ericsson.level are set to a non-OFF value.
java.util.logging.FileHandler.level	ALL	This parameter makes it possible to produce a log file. If this parameter is set to OFF, nothing is displayed in the log file, even if the level and com.ericsson.level are set to a non-OFF value.
java.util.logging.FileHandler.pattern	%h/ops_client_%u.log	%h: the home directory of the user running the OPS client. %t: the default temp directory of the system, which is /var/tmp/ in Solaris.
java.util.logging.FileHandler.limit	(50000)	The maximum size of a log file in bytes. After the limit has been reached, a new log file is started. By default, this parameter is commented out. There is no limit unless set by the user.

The following log levels are available:

Table 2 Available Log Levels

Log level	Short Description	Comment
OFF	Turns off logging	
SEVERE	Serious failures are logged.	SEVERE messages describe important events that prevent normal program execution.
WARNING	Indicates potential problems	WARNING messages describe events that are of interest to end users or system managers, or they indicate potential problems.
INFO	Information messages are also logged.	Typically, INFO messages are written to the console or its equivalent. So the INFO level must only be used for reasonably significant messages that make sense to end users and system administrators.
FINE	Provides basic tracing information	All of FINE, FINER, and FINEST are intended for relatively detailed tracing. The exact meaning of the three levels varies from subsystem to subsystem. In general, FINEST must be used for the most voluminous, detailed output, FINER for less detailed output (logging calls for entering, returning, or throwing an exception), and FINE for the lowest volume (and most important) messages. The FINE level must be used for information that is broadly interesting to developers who do not have a specialized interest in the specific subsystem. FINE messages can include things like minor (recoverable) failures.

		Issues indicating potential performance problems are also worth logging as FINE.
FINER	Provides more detailed tracing information	
FINEST	Detailed tracing information	
ALL	All messages are logged	

Note: All OPS GUIs must be shut down before the changes can take effect. The OPS GUIs share one JVM by default, and the JVM needs to be restarted for the changes to take effect.

To disable the trace, the OFF values from Table 1 need to be restored.

Tracing reduces the performance for OPS, turn it off when it is no longer needed.

The trace file size is unlimited by default, but it is possible to limit it by specifying java.util.logging.FileHandler.limit to the required maximum trace size in bytes.

3 OPS Server Reference

This chapter contains the following parts:

— General

— OPS Directory Structure

— Enabling and Disabling Trace

3.1 General

OPS is used for Man-Machine Language (MML) communication with all types of External Systems (ESs) that are supported by Winfiol. OPS includes the following functions:

— Handling of OPS scripts

— Presentation of delayed responses reports

— Command sessions and presentation of immediate responses

OPS Scripts

An OPS script can be created and executed by the user. The file can contain the following building blocks:

— OPS script commands

— Comments

— Blank lines

— MML commands

3.2 OPS Server Directory Structure

This chapter describes the directories related to OPS.

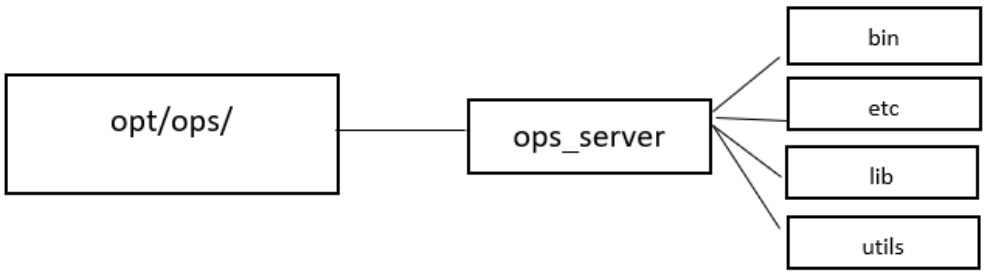

Figure 3 Directory Structure for ops_server

Software Directory - /opt/ops/ops_server

This directory contains static data, and has read only permission for users.

Software Directory

/opt/ops/ops_server

- bin
- ops_server.sh
- post-install.sh
- start_visibroker.sh
- etc
- log4j.xml
- logging.properties
- ops-config.properties
- lib

  This directory contains jar files
- utils
  - smoaxe.sh
  - pwAdmin.sh
  - getVersion.sh
  - bladeRunnerScript

/opt/ops/ops_server/bin	This directory contains the OPS startup scripts
/opt/ops/ops_server/bin/post_install.sh	OPS server startup script.

/opt/ops/ops_server/etc	This directory contains the reference configuration file.
/opt/ops/ops_server/etc/logging.properties	The OPS logging configuration reference file.
/opt/ops/ops_server/lib	This directory contains the jar files.
/opt/ops/ops_server/utils	This directory contains the script files for OPS utilities.

$HOME Directories

The $HOME directory path is the path to the user home dir. For each user of OPS, the directories shown in the Figure below must be created.

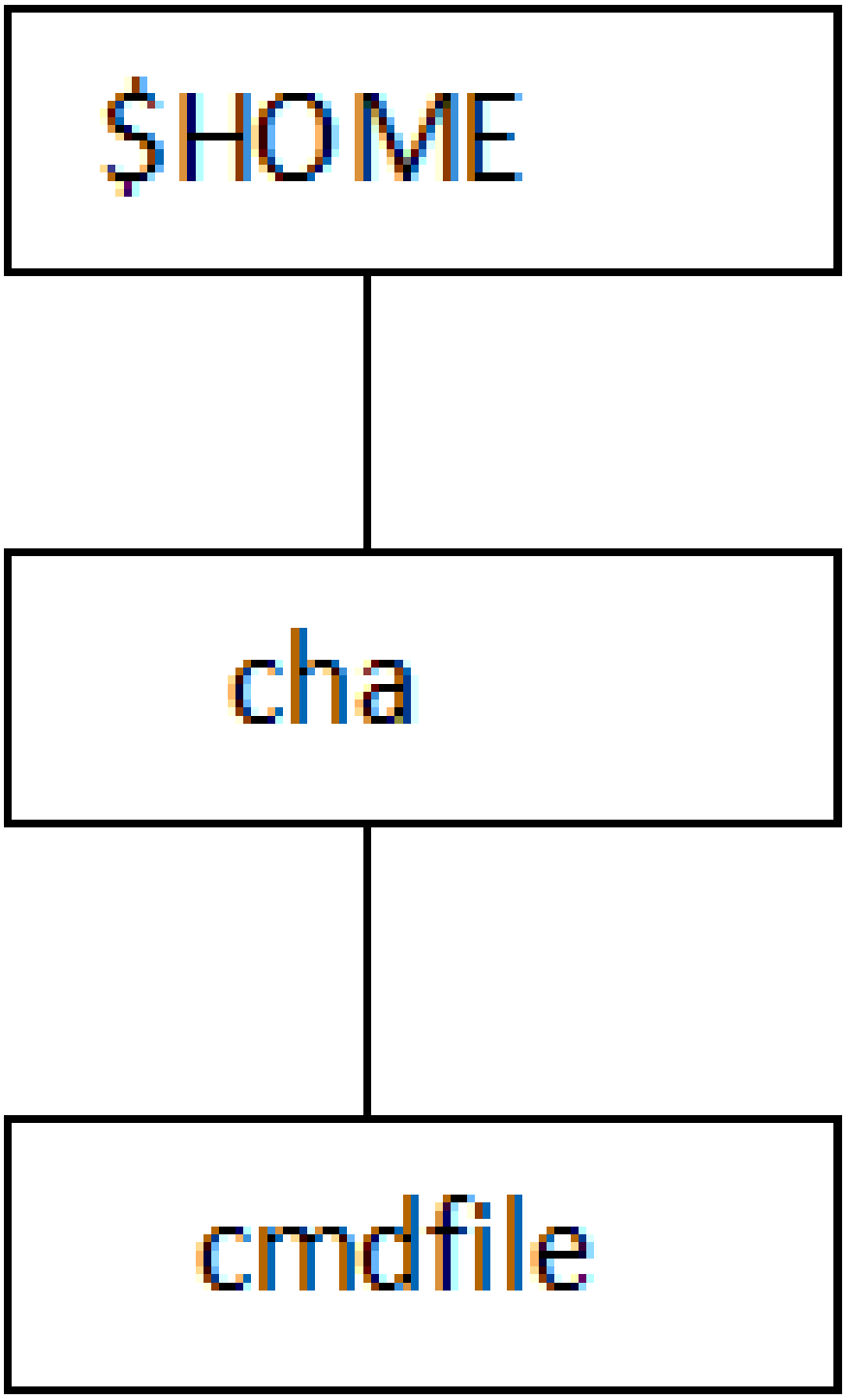

Figure 4 $HOME Directories

You can create your own directory structure for command files.

$HOME/cha/cmdfile

This directory contains command files that can be executed in OPS.

3.3 Enabling and Disabling Trace

It is possible to configure the server to provide detailed traces, which can be useful for debugging the OPS server. The trace configuration file for the client is located under

/opt/ops/ops_server/etc/logging.properties. In order to enable traces, the file needs to be edited. For more details about OPS logging, see OPS DCG Document

4 OPS Parser Reference

This chapter contains the following parts:

— General

— OPS Directory Structure

— Enabling and Disabling Trace

4.1 General

OPS Parser is used for parsing the OPS script commands . It uses Linux flex and bison utility to parser it successfully. Parser makes communication with winfiol in following scenario:

— Terminating the session.

— Sending the data towards the node.

— During connection request towards node

— Creating an asynchronous Session

4.2 OPS Directory Structure

This chapter describes the directories related to OPS.

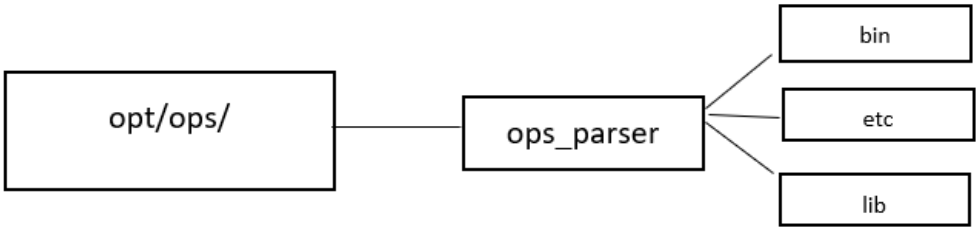

Figure 5 Directory Structure for ops_parser

Software Directory - /opt/ops/ops_parser

This directory contains static data, and has read only permission for users.

Software Directory

/opt/ops/ops_parser

- bin
- ops_parser.sh
- ops_parser
- post_install.sh
- etc
- logging.properties
- nls
- lib
  - Jar files
  - 3pp files
  - grpc files
  - boost files

4.3 Enabling and Disabling Trace

It is possible to configure the parser to provide detailed traces, which can be useful for debugging the OPS parser. The trace configuration file for the client is located under /opt/ops/ops_parser/etc/logging.properties. To enable traces, the file needs to be edited. For more details about OPS logging, see OPS DCG Document.

5 Configuration Information

This section includes information about the following items:

— Log and trace configuration

— OPS processes and script primitives

— Environment variables

Environment Variables for Server

CLASSPATH Path to Java classes used by OPS_SERVER.

PROCNAME Process name to be used.

LD_LIBRARY_PATH Path to dynamic C++ libraries.

OPS_CENTRALDIR Common store for OPS scripts.

Processes of CNA 403 3474

CNA 403 3474 includes three processes, ops_client, ops_server and ops_parser.

Ops_client Client process for the OPS

ops_server Daemon OPS server .

ops_parser Main process of OPS server. Started one per client session.

Application Log Configuration

The server has an application log contains major events for each session as well as general server events. The log is written to /var/opt/ericsson/nms_ops_server/logs. The log file has a maximum size of 150 MB. When this is reached, the file is rotated meaning opsserverlog0.log is renamed to opsserverlog1.log and a new opsserverlog0.log is started. If there is an old opsserverlog1.log present, it is deleted. Consequently, the log uses maximum 300 MB total in two files of maximum 150 MB each.

The application log can be configured by editing these properties in the configuration file (/opt/ops/ops_server/logging.properties):

Table 3 Application Log Configuration

Property	Description
ApplicationFileHandler.level	Set to APPLICATION if application logging is enabled or OFF if it is disabled. The default value is APPLICATION.

ApplicationFileHandler.limit	Maximum sizes for each log file in bytes. The default value is 150 MB (150,000,000 bytes).
ApplicationFileHandler.count	The desired number of rotated log files. The default value is 2.

Note: The server managed component (OPS_Server) needs to be restarted for the changes to take effect.

Trace Configuration

You can enable detailed traces for the server. This can be useful for debugging purposes. The trace ends up in

/var/tmp/opsserver_trace.log.

Note: Tracing reduces the performance of OPS. Turn OPS off when you are finished. The trace file size is unlimited by default. It is possible to set a limit to it.

The trace is configured in by editing these properties in the configuration file (/opt/ops/ops_server/etc/logging.properties):

Table 4 Trace Configuration

Property	Description
FileHandler.level	Set to ALL if tracing is enabled or OFF if it is disabled. The default value is OFF.
FileHandler.limit	Maximum sizes for each log file in bytes. The default is unlimited.

Note: The server managed component (OPS_Server) must be restarted for the changes to take effect.

Reference Configuration

Configuration files are stored in etc directory of client , server and parser.

/opt/ops/ops_client/etc/logging.properties /opt/ops/ops_server/etc/logging.properties /opt/ops/ops_parser/etc/logging.properties

6 OPS Configuration File

Configuration file ops.ini is used for user's private settings. The exact location of the configuration files depends on the hardware platform, in which system package OPS is used and what is specified in the installation process. The file is placed in /opt/ops/ops_client/etc directory.

The information in a configuration file can include the following:

— User interface configuration

— Customized Menu action commands which can be accessed by the user

— The locale that must be used

The file consists of a sequence of configuration items. Each configuration item is identified by a tag enclosed in brackets ({*}[]*) and consists of a sequence of parameters and reference lists, as follows :

[Module Configuration_Item_Name Configuration_ID]

PARAMETER_NAME=parameter_value REFERENCE_LIST=(Configurati
on_ID1 ... Configuration_IDn)

Note: The Configuration_ID part of the tag is omitted for some configuration items.

The text in a line after a # character is ignored.

7 Standard Maintenance Procedures

This section describes the standard maintenance procedures that the OPS administrator must perform occasionally or on a regular basis. The procedures are divided between the Client part and the Server part of OPS as follows:

Table 5 Maintaining the OPS Client

OPS Client
Modifying the OPS Configuration File
Starting OPS
Stopping OPS

Modifying the OPS Configuration File

The OPS administrator can modify the global OPS configuration file. See Chapter 5 for information about where to find the configuration file and the sub chapters in Chapter 5 for detailed information about the different configuration items.

Starting OPS

The OPS GUI is started by the command present in ops client directory.

/opt/ericsson/bin/ops_gui

Stopping OPS

OPS is stopped when all the process finish the execution and GUI is closed.

8 WinFIOL

Table 6 Winfiol Interface

Interface Name	Description
WinFIOL	WinFIOL serves as an interface between node and OPS. It provides various API's for the interaction. Create Session towards node Connect With Node Send Data to Node