

# Command administration

## DESCRIPTION



## NOTICE

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). Mitel makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

## TRADEMARKS

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at [legal@mitel.com](mailto:legal@mitel.com) for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

© Copyright 2017, Mitel Networks Corporation

All rights reserved

## 1

## GENERAL

There are several methods of accessing the MX-ONE Service Node: a shell utility, locally or remotely through the network, or using the MX-ONE Service Node Manager. For full access to all configuration options the MD-shell can be used.

## 1.1

## USING MD-SHELL COMMANDS

MD-shell (mdsh) is a command interpreter similar to standard Linux shells like the "Bourne Again Shell" (bash). Compared to other shells MD-shell has two extra main features: it understands MML commands, and it has the concept of authority levels. MD-shell also has a number of minor features that differentiate it from the standard Linux shells. Service node users should have mdsh as their log-in shell, but can also start it from another shell by typing **mdsh**.

There are three fundamentally different types of commands in mdsh:

- Built-in commands

These commands are an integrated part of mdsh and executed by mdsh itself. An example of such a command is "cd". Help on these commands is available by typing "help <command>" in mdsh.

- MML commands

These commands are different in syntax from normal Linux commands. No help for these commands can be retrieved in mdsh. See the command description for respective command group. For more information on using MML commands, see chapter 3 MML commands on page 6.

- Linux commands

These commands are separate executable files in the Linux environment, outside of mdsh. The Linux commands can be divided in three groups

- Standard Linux commands and tools like "less". These commands and tools are not specific for the MX-ONE Service Node. They work just as they work on any Linux computer. Help on these commands is usually available by typing "man <command>".
- MX-ONE Service Node commands based on the MD\_COM framework. Examples of commands based on the MD\_COM framework are "alarm" and "extension". MD-shell interacts with the MD\_COM framework, and tab-completion is available also for the command line switches. Help on these commands is available by typing "<command> -?" (or "<command> --help").
- Old MX-ONE Service Node commands (not based on the MD\_COM framework). Examples of old commands are "trace" and "number\_initiate". (These old commands will probably be rewritten into new commands based on the MD\_COM framework in future releases.) Help on these commands is available by typing "<command> -help"

Typing "help" in mdsh will result in a list of all commands that are available for a user. Flow control statements like loops, "if" statements, "while" statements, pipes or redirection are not supported in mdsh. Another shell should be used if these features are needed (bash is recommended).

The following switches can be used on the mdsh command line:

**-c <command>**

Only execute command. The command is executed directly and mdsh is terminated. No prompt is received.

**-f**

Turn off confirmation of dangerous commands

**-genConf file, --genConf file**

Generate an example configuration file for mdsh

**-, -H, -h, -help, --help**

Print mdsh help text

A command line argument that is not one of the above switches, should be the name of a script file with mdsh commands. Further command line arguments are passed as arguments to the script file.

## 1.2 AUTHORITY LEVELS

All commands that can be executed inside MD-shell are assigned an authority level from 0 to 7 in the MD-shell configuration file `/etc/opt/eri_sn/mdsh.conf`. The users are of the system belong to groups and the groups are given authority levels in MD-shell configuration file `/etc/opt/eri_sn/mdsh.conf`.

Inside MD-shell a user can execute only commands that require an authority level lower than or equal to the authority level of the group that the user belongs to. When typing "help" with no arguments in the MD-shell, only the commands that the user is allowed to run (according to the authority level) are shown

## 1.3 DANGEROUS COMMANDS

In the MD-shell configuration file `/etc/opt/eri_sn/mdsh.conf` some commands are classified as dangerous. If a command is classified as dangerous a questions like

Are you sure?(Y/N):

is printed when the command has been entered. To continue executing the command type Y (or YES) and <Enter>. If the operator types N (or NO) and <Enter> the commands is aborted.

By setting the environment variable `$_FORCED` to true, these "are you sure" questions are suppressed.

## 1.4 COMMENTS

In MD-shell the "#" character starts a comment. The rest of the line from the "#" character is a comment.

## 1.5 INTERRUPTING A COMMAND

As usual in Linux commands can be interrupted by the CTRL and C key at the same time. No confirmation of the break (other than a new prompt) will normally be given.

## 2 MX-ONE SERVICE NODE COMMANDS AND BUILT-IN COMMANDS

### 2.1 COMMAND SYNTAX

A command consists of a command code which may be followed by one or more switches (parameters).

`command_code -parameter1 {[-parameter2 [-parameter3]] | [-parameter4]}`

- [ ] Indicates that this part is optional (may be omitted).
- { } Indicates a grouping of switches (often of alternatives).
- | Separates different alternatives.

### 2.2 CHARACTERS

The language used for entering Linux commands and built-in commands uses the English alphabet including "\_" (underscore).

#### 2.2.1 SPECIAL CHARACTERS

When entering commands, the following characters are also used:

- Precedes parameter name. Denotes sub-element in hardware equipment positions
- , Separates parameter values, for example 100,200, meaning the values 100 and 200, but not the values in between.
- .. Separates beginning and end values for a consecutive series of parameter values. Example: 100..200 means all values from 100 up to and including 200.

### 2.3 COMMAND VALIDITY AND DATA LOCK

Before doing the real work of the command the command line is checked both syntactically and semantically.

If the command is to change exchange data a "data lock" is acquired while doing the change to ensure that no other command can change the data at the same time. If another command holds the "data lock" the "data lock" cannot be acquired and the command is terminated with an error message.

## 3

## MML COMMANDS

The man-machine communication is executed by means of command and confirmation. A command is an instruction from an authorized maintenance operator, and a confirmation is the system's response to a command.

For every command the system checks the syntax and format of the entered characters.

## 3.1

## COMMAND SYNTAX

A command consists of a command code which may be followed by one or more parameters.

$$\text{Command code: parameter} \left\{ \begin{array}{l} \\ ; \end{array} \right\}$$

$$\text{Parameter name} = \left\{ \begin{array}{l} \text{All} \\ \text{Parameter value} \left[ \begin{array}{l} \& \text{parameter value} \\ \&\& \text{parameter value} \end{array} \right] \end{array} \right\}$$

The maximum number of characters for a command is 200, that is, two rows. All characters belong to the ACSII code set.

- [ ] Indicates that this part is optional (may be omitted).
- { } Indicates that one of the alternatives must be specified.
- ... Indicates that one or more values can be specified for this parameter, that is, the parameter may contain one or several of the following &, && or ALL

## 3.2

## CHARACTERS

The language used for man-machine communication in the system comprises the following characters.

## 3.2.1

## COMMAND CHARACTERS

- A-F** Capital letters.
- 0-F** Hexadecimal digits.

## 3.2.2

## EDITING CHARACTERS

During communication with the system, the following characters are also used:

- Used, for example., to denote sub-element in hardware equipment positions.
- ,
- & Separates parameter values, for example 100&200, meaning that the values 100 and 200 can be used, but not the values in between.

<b>&amp;&amp;</b>	Separates beginning and end values for a consecutive series of parameter values. Example: 100&&200 means that all values from 100 up to and including 200 can be used.
<b>:</b>	Separates command code and parameter part.
<b>;</b>	Terminates a command string.
<b>=</b>	Ties a value to a parameter.
<b>#</b>	Indicates beginning of comment. The rest of the line will be interpreted as a comment.
<b>H"</b>	Hexadecimal notation.
<b>O"</b>	Octal notation.
<b>B"</b>	Binary notation.
<b>"</b>	The next character is not to be checked.
<b>"</b>	Indicates beginning and end of a parameter value which consists of a text string.
<b>&amp;</b>	After the command string terminator (;) implies, that the command is executed in the background and the result printout is given, when the execution is completed.

### 3.3 COMMAND VALIDITY

Besides the syntax check, each command is further checked with respect to:

- validity of each character
- validity of command code
- whether the command is free
- if the command is to change exchange data, a check is performed to ensure that no other such command is in the process of being executed
- number of parameters
- whether parameter values conform with the type of command
- whether the parameter values are within the boundaries

### 3.4 COMMAND STRUCTURE

A command consists of five characters.

XXYYZ where

<b>XX</b>	Indicates the command group to which the command belongs.
<b>YY</b>	Indicates the function the command is to perform.
<b>Z</b>	Indicates the operation code.

The following operation codes exist:

<b>C</b>	Change
<b>E</b>	End
<b>I</b>	Initiate
<b>L</b>	Load
<b>P</b>	Print
<b>R</b>	Remove
<b>S</b>	Set, allocate
<b>T</b>	Transfer

## 3.5 COMMANDS THAT CHANGE EXCHANGE DATA

On certain occasions, e.g LIMs blocked, it is impossible to execute commands that are changing or copying exchange data. If the maintenance operator keys a command when this is the case, the system can answer with any of the following printouts:

### **NOT ACCEPTED**

followed by any of:

#### **LIM BLOCKED**

LIM is blocked

#### **RELOAD IS EXECUTING**

Reload is in progress

#### **RESTART IS EXECUTING**

Restart is in progress

#### **OTHER COMMAND THAT CHANGES PROGRAM CODE IS NOW EXECUTING**

Dump to backup media is in progress.

#### **OTHER COMMAND THAT CHANGES RELOAD DATA IS NOW EXECUTING**

Another command that is changing or copying exchange data is executing

The maintenance operator must wait to key a command in the above cases.

## 3.6 PRINTOUTS

### 3.6.1 GENERAL

The system has three different types of printouts:

- Verification printouts
- Procedure printouts
- Immediate response printouts

### 3.6.2 VERIFICATION PRINTOUTS

A verification printout consists of a command being echoed back to the terminal if the command is classified as dangerous to the system. For more information, see 1.3 Dangerous commands on page 4.

### 3.6.3 PROCEDURE PRINTOUTS

Procedure printouts are used to inform the maintenance operator whether a command has been executed, partly executed or not executed at all.

If the command has been accepted, the following printouts may occur;

#### **EXECUTED**

The command has been fully executed.

Error detected during execution:



**NOT ACCEPTED**

The command is erroneously specified. No action has been taken.

**PARTLY EXECUTED**

The action specified has been partly executed but detection of an error has interrupted the execution.

The messages NOT ACCEPTED and PARTLY EXECUTED will be followed by a printout specifying the error.

For reload data changing commands it is possible to have a sequence number printed out in addition to the printout. The sequence number is incremented with each command and is used for tracking purpose.

### 3.6.4

### IMMEDIATE RESPONSE PRINTOUTS

This kind of printouts are given as soon as the system has accepted a command. If the response time to printout is long, WAIT will be printed.

Immediate response printouts terminate with END.

## 4

## REMOTE MAINTENANCE

Remote maintenance of the MX-ONE Service Node is possible using a utility such as ssh. It provides encrypted authentication and communication.

In SuSE Enterprise Server, ssh is delivered in the OpenSSH package. See appropriate SuSE Linux Reference Manual for details regarding usage.

## 5

## MX-ONE SERVICE NODE MANAGER

The MX-ONE Service Node Manager is used to maintain IP, fixed remote, and mobile extensions in the MX-ONE. It also includes a Command Line Interface (CLI) that can be used for other operations on the MX-ONE Service Node. The CLI gives users access to mdsh functionality from a web browser.