

## ***234- Global And Environmental Commands***

---

234.1 History Command	234-7330
234.2 Echo Command	234-7331
234.3 Telnet Command	234-7332
234.4 Telnet AI Command	234-7333
234.5 Exec Command	234-7334
234.6 Sleep Command	234-7336
234.7 Tree Command	234-7337
234.8 Write Command	234-7338
234.9 Logout Command	234-7339
234.10 Help Command	234-7340
234.11 Info Command	234-7341
234.12 Exit Command	234-7344
234.13 Back Command	234-7345
234.14 Delete Command	234-7346
234.15 In Command	234-7347
234.16 Configure Command	234-7348
234.17 Action Command	234-7350
234.18 Show Command	234-7351
234.19 Recursive Show Command	234-7352
234.20 Sample Command	234-7353
234.21 Poll Command	234-7354
234.22 Debug Command	234-7355
234.23 Sysfil Command	234-7356

## 234.1 History Command

### Command Description

*This command displays the last commands entered at the terminal.*

*A maximum of 20 commands can be stored. A command that is identical to the previous command is not stored.*

*Commands containing syntax and semantic errors are also stored in the history buffer. This way a user can correct a command without retyping the command completely.*

*Each command is preceded by a sequence number.*

*The sequence number does not increment when a command is not stored because it is identical to the previous command.*

*Example :*

```
esam># history
1 : history
esam># echo "first command"
first command
esam># echo "second command"
second command
esam># echo "second command"
second command
esam># history
1 : history
2 : echo "first command"
3 : echo "second command"
4 : history
esam>#
```

### User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*This command can be executed by all users.*

### Command Syntax

The command has the following syntax:

> history

## 234.2 Echo Command

### Command Description

*This command echoes a string to the standard output.*

### User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*This command can be accessed by any user.*

### Command Syntax

The command has the following syntax:

```
> echo [ [ no ] message ]
```

### Command Parameters

**Table 234.2-2 "Echo Command" Command Parameters**

Options	Type	Description
[no] (message)	Format: - a long message - length: x<=512	<i>optional parameter with default value: ""</i> the string to be echoed

# 234.3 Telnet Command

## Command Description

*This command establishes a telnet session to another server.*

*The telnet session will last until the connection is closed by the telnet server or when the operator enters the character **Control-^**.*

*The session inactivity-time protection will not put any messages on the screen when you are executing the telnet command. Your connection will be broken without warning if the inactivity timeout expires.*

## User Level

*The command can be accessed by operators with lanxAccess,lanxAccessext privileges.*

*This command can only be executed by users having read or write access to functions located in the SHub/IHub.*

*The command can only be executed from a telnet terminal, not from a craft terminal.*

## Command Syntax

The command has the following syntax:

> telnet (destination)

## Command Parameters

Table 234.3-2 "Telnet Command" Command Parameters

Options	Type	Description
(destination)	Format: shub Possible values: - shub : the service hub (shub or ihub)	<i>optional parameter</i> identification of the telnet server

## 234.4 Telnet AI Command

### Command Description

*This command establishes a telnet session to the Application Intelligence platform (core2&3 on NANT-E or FANT-F).*

*The telnet session will last until the connection is closed by the telnet server or when the operator exits from the telnet session.*

*The session inactivity-time protection will not put any messages on the screen when you are executing the telnet command. Your connection will be broken without warning if the inactivity timeout expires.*

### User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*The command can be executed from a craft terminal or a telnet terminal.*

### Command Syntax

The command has the following syntax:

```
> telnet-ai (destination) [ virtualplatform <AI::VirtualPlatform> ]
```

### Command Parameters

**Table 234.4-2 "Telnet AI Command" Command Parameters**

Options	Type	Description
(destination)	Format: ( nt-a   nt-b ) Possible values: - nt-a : the ai platform (core2-3) on NT-a - nt-b : the ai platform (core2-3) on NT-b	<i>optional parameter</i> identification of the telnet server
virtualplatform	Parameter type: <AI::VirtualPlatform> Format: - AI Virtual Platform Index within the Slot - range: [1]	<i>optional parameter</i> identification of the virtual platform

## 234.5 Exec Command

### Command Description

*This command executes CLI commands found in a text file. A file containing CLI commands is called a script file.*

*A script file must follow the same syntax rules as commands entered interactively. Only the tab and the question mark lose their special meaning in a script, so command expansion and single-line help are not available.*

*When initiating the execution of a script, the operator can choose how the system must behave in case of an error in a CLI command. By default the script will be aborted. Other command-line options are: ignore the error and proceed executing the script or prompt the operator. When prompted the operator can choose to either abort the script or ignore the error and proceed. When prompted due to an error, the operator may wish to open a separate CLI session to take some corrective action for the failed command and then revert to the session where the script is running to continue with the next command.*

*A command may be preceded by a minus '-' to prevent the script from being aborted in case of a semantic error; for example, when you try to delete a node instance that does not exist. In case a minus-sign precedes the command and the error is a semantic error, then the operator will not be prompted.*

*By default, a command script does not interact with the user. The execution of the script will be aborted if a command or filter requires interaction with the user. Interactivity can be enabled by specifying the **interactive** option with the **exec** command.*

*The **exec** command may not be used in a command script, so command scripts may not be nested.*

*Script files are typically stored in the **/var/cmd** directory.*

*A script file can be used to restore a configuration. This can be done by reloading an initial and working configuration script. Alternatively, the complete configuration of an existing system can be captured. A configuration script is produced by means of the command **info configure <configuration node> | tee <script name>** and the script is stored in file **/var/cmd/<script name>**. Subsequently, the retrieved file needs editing as the output is not guaranteed to be in line with the CLI sequences required to configure a new system. The configuration can then be restored on an empty system with the command **exec /var/cmd/<script name>**.*

*Warning: the **/var/cmd** is cleaned when the system resets, so the configuration file must be transferred to another system with **tftp** before resetting the system.*

*Note: saving and restoring the configuration will not work for areas in which resource identifiers are assigned dynamically by the system.*

### User Level

*The command can be accessed by operators with **allAccess,allAccessext** privileges.*

*This command can be executed by all users. However, the execution of the script will be aborted when the CLI parser tries to execute the first command in the script for which the user does not have authorization.*

### Command Syntax

The command has the following syntax:

```
> exec (filename) [ [ no ] interactive ] [ no at-replay | at-replay <Option::AtFailure> ]
```

### Command Parameters

Table 234.5-2 "Exec Command" Command Parameters

Options	Type	Description
(filename)	Format: - an absolute or relative path name (example : /var/log) - range: [-z-] - length: 1<=x<=128	<i>optional parameter</i> the name of the file with the script
[no] interactive	Parameter type: boolean	<i>optional parameter</i> interaction with the operator is allowed
[no] at-replay	Parameter type: <Option::AtFailure> Format: ( stop   continue   prompt ) Possible values: - stop : stop execution of the script - continue : continue with the execution of the script - prompt : prompt the operator what to do	<i>optional parameter with default value: "stop"</i> directs what to do if error is encountered in the script

# 234.6 Sleep Command

## Command Description

*This command pauses the execution for the specified number of seconds.*

*This command is typically used in scripts (see : the **exec** command).*

## User Level

*The command can be accessed by operators with **allAccess,allAccessext** privileges.*

*This command can be executed by any user.*

## Command Syntax

The command has the following syntax:

> sleep (time)

## Command Parameters

Table 234.6-2 "Sleep Command" Command Parameters

Options	Type	Description
(time)	Format: - the period to sleep - unit: sec - range: [1...100]	<i>optional parameter</i> the period to sleep



## 234.7 Tree Command

### Command Description

*This command displays the structure of a node and its subnodes.*

### User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*This command can be executed by any user on a node to which he has read or write access.*

*The output does not contain any information related to subnodes to which the user has no access.*

### Command Syntax

The command has the following syntax:

```
> tree [ [ no ] detail ]
```

### Command Parameters

Table 234.7-2 "Tree Command" Command Parameters

Options	Type	Description
[no] detail	Parameter type: boolean	<i>optional parameter</i> show the details

## 234.8 Write Command

### Command Description

*This command writes a message to the terminal of the addressee.*

*The message can be directed to a user, all user with the same user profile or to all users.*

*Messages are only delivered to users that are logged in. Users that are not logged in, will not receive the message.*

*The sender will also receive the message if it is one of the addressees.*

### User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*This command can be used by any user.*

### Command Syntax

The command has the following syntax:

> write to <Option::MessageDestination> (message)

### Command Parameters

**Table 234.8-2 "Write Command" Command Parameters**

Options	Type	Description
to	Parameter type: <Option::MessageDestination> Format: ( all   operator : <Sec::OperatorName>   profile : <Sec::ProfileName> ) Possible values: - all : all active sessions - operator : all sessions of the operator with that name - profile : all sessions of operators with that profile name Field type <Sec::OperatorName> - the name of an operator - range: [a-z0-9+] - length: 1<=x<=11 Field type <Sec::ProfileName> - the name of an operator profile - range: [a-z0-9] - length: 1<=x<=11	<i>optional parameter</i> the destination of the message
(message)	Format: - a message - length: x<=256	<i>optional parameter</i> the message

## 234.9 Logout Command

### Command Description

*This command logs the user out.*

### User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*This command can be executed by any user.*

### Command Syntax

The command has the following syntax:

> logout

## 234.10 Help Command

### Command Description

*This command gives the user detailed information on a subject.*

*By default the command gives information on the current node.*

### User Level

*The command can be accessed by operators with `forHelp`, `forHelpext` privileges.*

*This command can be executed by each user. A user will only be able to get help for nodes for which he has read or write access.*

### Command Syntax

The command has the following syntax:

> help

## 234.11 Info Command

### Command Description

*This command shows the configuration of the current node and all of its subnodes.*

*The user can select three different output formats:*

- *The default is hierarchical command format.*
- *An other format is the flat command format. The output in this format can be saved in a file with the **tee** filter and can be executed with the **exec** command. This way a saved configuration can be restored on an empty system.*
- *The other format is XML. The output in this format is much more suited as input for tools because it is less sensitive to further changes.*

*Example of the default format :*

```
esam>configure>system>security# info profile
configure system security
#-----
echo "system"
#-----
profile admin
  prompt "%n%d%c "
  terminal-timeout 30
  description "default administrator group"
  aaa write
  atm write
  alarm write
  dhcp write
  equipment write
  igmp write
  ip write
  ppoe write
  qos write
  software-management write
  transport write
  vlan write
  xdsl write
  security write
exit
profile read
  prompt "%n%d%c "
  security read
exit
#-----
```

*Example of the flat command format :*

```
esam>configure>system>security# info profile flat
configure system security profile admin prompt "%n%d%c " terminal-timeout 30
description "default administrator group" aaa write atm write alarm write dhcp write
equipment write igmp write ip write ppoe write qos write software-management write
transport write vlan write xdsl write security write
configure system security profile read prompt "%n%d%c " security read
```

*Example of the XML format :*

```
esam>configure>system>security# info profile read xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration-data display-level="normal">
  <hierarchy name="configure" type="static">
    <hierarchy name="system" type="static">
      <hierarchy name="security" type="static">
        <hierarchy name="profile" type="dynamic">
          <hier-id name="name" is-named="no" value="read" is-default="no"
type="Sec::ProfileName"/>
          <instance>
            <parameter name="prompt" is-default="no" type="Sec::Prompt">"%n%d%c
" </parameter>
            <parameter name="password-timeout" units="days" is-default="yes"
type="Sec::PasswordTimeout">0</parameter>
            <parameter name="terminal-timeout" units="minutes" is-default="yes"
type="Sec::InactivityTimeout">0</parameter>
```

```

        <parameter name="description" is-default="yes"
type="Sec::ProfileDesc"></parameter>
        <parameter name="aaa" is-default="yes" type="Sec::Aaa">none</parameter>
        <parameter name="atm" is-default="yes" type="Sec::Atm">none</parameter>
        <parameter name="alarm" is-default="yes"
type="Sec::Alarm">none</parameter>
        <parameter name="dhcp" is-default="yes" type="Sec::Dhcp">none</parameter>
        <parameter name="equipment" is-default="yes"
type="Sec::Equipment">none</parameter>
        <parameter name="igmp" is-default="yes" type="Sec::Igmp">none</parameter>
        <parameter name="ip" is-default="yes" type="Sec::Ip">none</parameter>
        <parameter name="ppoe" is-default="yes" type="Sec::Ppoe">none</parameter>
        <parameter name="qos" is-default="yes" type="Sec::Qos">none</parameter>
        <parameter name="software-management" is-default="yes"
type="Sec::Sw">none</parameter>
        <parameter name="transport" is-default="yes"
type="Sec::Transport">none</parameter>
        <parameter name="vlan" is-default="yes" type="Sec::Vlan">none</parameter>
        <parameter name="xdsl" is-default="yes" type="Sec::Xdsl">none</parameter>
        <parameter name="cluster" is-default="yes"
type="Sec::Cluster">none</parameter>
        <parameter name="security" is-default="no"
type="Sec::Security">read</parameter>
    </instance>
</hierarchy>
</hierarchy>
</hierarchy>
</hierarchy>
</configuration-data>
esam>configure>system>security#

```

The user can select two different output levels :

- The default is to show only the parameters which have not the default value. Static subnodes which contain (recursively) only default values are not shown. The XML format shows always all parameters.
- In the **detail** output level all nodes and all parameters shown. This output can not be used for replay.

Example of detailed information :

```

esam>configure>system>security# info profile read detail
configure system security
#-----
echo "system"
#-----
profile
  prompt "%n%d%c "
  no password-timeout
  no terminal-timeout
  no description
  no aaa
  no atm
  no alarm
  no dhcp
  no equipment
  no igmp
  no ip
  no ppoe
  no qos
  no software-management
  no transport
  no vlan
  no xdsl
  security read
exit
#-----
esam>configure>system>security#

```

## User Level

The command can be accessed by operators with readAccess,readAccessext privileges.

This command can be accessed in all nodes to which the user has read access. It is not available in **admin** and **show** nodes or their subnodes.

The output will only contain information for the nodes for which the user has read access.

## Command Syntax

The command has the following syntax:

```
> info [ [ no ] detail ] [ (format) ]
```

## Command Parameters

**Table 234.11-2 "Info Command" Command Parameters**

Options	Type	Description
[no] detail	Parameter type: boolean	<i>optional parameter</i> show the details
(format)	Format: ( hierarchical   flat   xml ) Possible values: - hierarchical : show the output in hierarchical command format - flat : show the output in flat command format - xml : show the output in XML format	<i>optional parameter</i> the format in which the command output is printed

# 234.12 Exit Command

## Command Description

*This command changes the current position to the parent node of the current position. This command always succeeds, even if the current position is the top node. The command has no effect in this case.*

*The option **all** changes the current position to the top node.*

## User Level

*The command can be accessed by operators with **allAccess,allAccessext** privileges.*

*This command can be executed by all users*

## Command Syntax

The command has the following syntax:

> exit [ [ no ] all ]

## Command Parameters

Table 234.12-2 "Exit Command" Command Parameters

Options	Type	Description
[no] all	Parameter type: boolean	<i>optional parameter</i> set the current position to the top node



## 234.13 Back Command

### Command Description

*This command changes the current position to the previous position.*

*The command will always succeed, even if the previous position no longer exists. The command changes the current position to the top node in that case.*

### User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*The command can be executed by all users.*

### Command Syntax

The command has the following syntax:

```
> back
```

# 234.14 Delete Command

## Command Description

*This command deletes a file from the directory **/var/cmd**. Such file is typically created with the **tee** filter.*  
*The command will always succeed, even if the specified file does not exist.*

## User Level

*The command can be accessed by operators with **allAccess,allAccessext** privileges.*  
*The command can be executed by all users.*

## Command Syntax

The command has the following syntax:

> delete-file (file)

## Command Parameters

Table 234.14-2 "Delete Command" Command Parameters

Options	Type	Description
(file)	Format: - a file name (example : errors.log) - range: [.0-z-] - length: 1<=x<=119	<i>optional parameter</i> the name of the file to be deleted

## 234.15 In Command

### Command Description

*This command can not be directly executed by the operator. It is the default command for all nodes when no node attributes are specified, except for all subnodes of **configure** and the leaf nodes of **show**.*

*This command changes the position in the command tree to the current node instance.*

### User Level

*The command can be accessed by operators with `readAccess`, `readAccessext` privileges.*

*This command can be executed by any user for the nodes to which he has read or write access.*

## 234.16 Configure Command

### Command Description

*This command can not be directly executed by the user. It is the default command for the **configuration** node and most of its subnodes.*

*This command serves four purposes :*

- *changing the current position in the command tree to a node instance.*
- *creating a node instance. The current position will be changed to the newly created node instance.*
- *modifying one or more attributes of a node instance. The current position in the command tree will be changed to the modified node instance.*
- *deleting a node instance.*

*The command changes the current position in the command tree to a node instance in case you specify an existing node instance and no parameters. Example : **configure equipment shelf 1/1**. The last character in the default prompt will be #.*

*The command creates a node instance in case the node is dynamic and the specified node instance does not exist. Example : **configure system security operator new\_user password plain:"hallo#"**. Parameter values must be supplied for all mandatory parameters. Parameter values may be supplied for optional parameters. The command changes the current position in the command tree to the newly created node instance. The last character in the default prompt will be \$.*

*The command modifies the attributes of a node instance in case the node instance exists and if the user supplies parameters. Example : **configure system security operator existing\_user description "a new description"***

*The command will delete a node instance when it is a dynamic node and the node name is preceded by **no**. Example : **configure system security no operator existing\_user**.*

*The command generates no output*

*Example 1 - change the current position to an existing static node :*

```
esam# configure system security
esam>configure>system>security#
```

*Example 2 - create a new dynamic node (note that one has to enter the mandatory parameters) :*

```
esam>configure>system>security# operator new profile read password plain:&secret&
esam>configure>system>security>operator>new$
```

*Example 3 - change an attribute :*

```
esam>configure>system>security>operator>new$ description "just created"
esam>configure>system>security>operator>new$
```

*Example 4 - change to an existing dynamic node (note that the prompt is different with example 2) :*

```
esam>configure>system>security>operator>new$ back
esam>configure>system>security# operator new
esam>configure>system>security>operator>new#
```

*Example 5 - delete a dynamic node*

```
esam>configure>system>security>operator>new# back
esam>configure>system>security# no operator new
esam>configure>system>security#
```

*Example 6 - go to the root*

```
esam>configure>system>security# exit all
esam>#
```

*Example 7 - request help for a non-existing static node, no parameters or subnodes will be shown*

```
esam>#configure igmp channel 1/1/6/1 ?  
<CR>                : execute command  
esam>#configure igmp channel 1/1/6/1
```

*Example 8 - go to a non-existing static node*

```
esam>#configure igmp channel 1/1/6/1  
Error : port not available  
esam>#
```

## User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*A user can change the current position to each node to which he has read access.*

*A user can create, modify and delete nodes to which he has write access.*

## 234.17 Action Command

### Command Description

*This command can not be directly executed by the user. It is the default command for the **admin** node and most of its subnodes.*

*This command serves two purposes :*

- *changing the current position in the command tree to a node instance.*
- *executing an action.*

*The command changes the current position in the command tree to a node instance in case you specify an existing node instance and no parameters. From the moment you specify a parameter it will try to execute the corresponding action.*

### User Level

*The command can be accessed by operators with `allAccess`, `allAccessExt` privileges.*

*A user can change the current position in the command tree to each node to which he has read access.*

*A user can execute the action in each node to which he has access.*

## 234.18 Show Command

### Command Description

*This command shows the runtime data and eventually some important configuration data.*

*This command can not be executed directly by the user. It is the default command for the leaf nodes of **show**. It shows the runtime data.*

*When show data in no detail mode, if the length of parameter is too long for a table cell, an asterisk will be used to omit some character to fit the table cell length.*

### User Level

*The command can be accessed by operators with readAccess,readAccessext privileges.*

*The user can execute this command for all nodes to which he has read access.*

### Command Parameters

**Table 234.18-2 "Show Command" Command Parameters**

Options	Type	Description
[no] detail	Parameter type: boolean	<i>optional parameter</i> show the details
[no] xml	Parameter type: boolean	<i>optional parameter</i> show the output in XML format

## 234.19 Recursive Show Command

### Command Description

*This command shows the runtime data and eventually some important configuration data.*

*This command can not be executed directly by the user. It is the default command for some leaf nodes of **show**. It shows the runtime data in a recursive way (similar to **info**).*

### User Level

*The command can be accessed by operators with `readAccess`,`readAccessext` privileges.*

*The user can execute this command for all nodes to which he has read access.*

### Command Parameters

Table 234.19-2 "Recursive Show Command" Command Parameters

Options	Type	Description
[no] detail	Parameter type: boolean	<i>optional parameter</i> show the details
[no] xml	Parameter type: boolean	<i>optional parameter</i> show the output in XML format



## 234.20 Sample Command

### Command Description

*This command executes a test and shows the test result.*

*This command can not be executed directly by the user. It is the default command for some leaf nodes of **show**. It shows the final test results.*

### User Level

*The command can be accessed by operators with readAccess,readAccessex privileges.*

*The user can execute this command for all nodes to which he has read access.*

### Command Parameters

Table 234.20-2 "Sample Command" Command Parameters

Options	Type	Description
[no] detail	Parameter type: boolean	<i>optional parameter</i> show the details
[no] xml	Parameter type: boolean	<i>optional parameter</i> show the output in XML format

## 234.21 Poll Command

### Command Description

*This command can not be directly executed by the user. It is the default command for some leaf nodes in **show**.*

*This command starts a test and displays at regular time intervals during the specified time period the intermediate results of the test.*

*The difference between **poll** and **sample** command is that, in **poll** command the intermediate results are shown while in **sample** command only the final results are shown.*

### User Level

*The command can be accessed by operators with writeAccess,writeAccessext privileges.*

### Command Parameters

**Table 234.21-2 "Poll Command" Command Parameters**

Options	Type	Description
[no] detail	Parameter type: boolean	<i>optional parameter</i> show the details
[no] xml	Parameter type: boolean	<i>optional parameter</i> show the output in XML format
[no] sample-interval	Parameter type: <Option::SampleInterval> Format: - the time between two samples - unit: sec - range: [1...60]	<i>optional parameter with default value: 1</i> the time between 2 samples
[no] number-of-samples	Parameter type: <Option::SampleCount> Format: ( unlimited   <Option::SampleCount> ) Possible values: - unlimited : sampling continues until stopped by the user Field type <Option::SampleCount> - the number of samples to be taken - range: [1...1000]	<i>optional parameter with default value: "unlimited"</i> the number of samples to be taken

## 234.22 Debug Command

### Command Description

*This command shows the runtime data and eventually some important configuration data.*

*This command can not be executed directly by the user. It is the default command for the leaf nodes of **debug**. It shows the runtime data.*

### User Level

*The command can be accessed by operators with **allAccess**,**allAccessext** privileges.*

*The user can execute this command for all nodes to which he has read access.*

### Command Parameters

Table 234.22-2 "Debug Command" Command Parameters

Options	Type	Description
[no] xml	Parameter type: boolean	<i>optional parameter</i> show the output in XML format

# 234.23 Sysfil Command

## Command Description

*This command shows the syslog files and its contents in CLI session.*

*It is the default command for the leaf nodes of **show system syslog list files and show system syslog file name filter options**. It shows the runtime data. Only xml option is supported for this handler type.*

## User Level

*The command can be accessed by operators with allAccess,allAccessext privileges.*

*The user can execute this command for show system syslog list/file nodes to which he has security read access.*

## Command Parameters

Table 234.23-2 "Sysfil Command" Command Parameters

Options	Type	Description
[no] xml	Parameter type: boolean	<i>optional parameter</i> show the output in XML format