Complete solutions for networking

Eltex.ACS

Operation manual

Version 1.12

Release number	Issue date	Revisions
1.12	07 May 2019	Changed: 1.1 Structure 3 Eltex.ACS installation and configuration 4 ACS-BOX installation 5 ELTEX.EMS.ACS installation 6 Server reservation and backup
1.11	14 November 2018	Changed: 1.1 Structure 1.2 Communication organization chart 2.1 Eltex.ACS + MySQL + Eltex.ACS-NBI + Eltex.ACS-BOX configuration 3 ACS installation and configuration 4 ACS-BOX installation 9 License
1.9	17 February 2016	Added exposure ConnectionRequest parameters during device initialization. Increased wait time, while the long answer on a request from the device settings. Developed nbi search function by device IP address. Changed the names of functions in the NBI. Developed the application management mechanisms on the basis of subscriptions for NBI. Fixed problem with NBI command to perform for device behind the NAT.
1.6.5	28 October 2014	NBI support added General optimizations
1.6.2	13 December 2013	Added SSL support for HTTPS. Revision of data model directories for new devices (TC, TAU, RG). Improved load capacity and stability. Reservation improvements.
1.6.1	21 June 2013	Full typing implementation in the soap protocol (for devices with permanent typing). Several standard data models added (including data models for correct operation with 'typed' devices) TR-098, TR-106, TR-104, TR-135. Operation logging improvements (+ ipping, direct commands). Device busy messages. License threshold exceeded messages. Server restart information in GUI.
1.6	17 May 2013	General log (including GUI): - informs to the general log; - commands to the general log; - errors to the general log; - log to GUI; - log to CLISH. TR-111 (STUN-Server) support: - UDPConnectionRequest; - Script, that gets parameters for STUN (NATDetected, UDPConnectionRequestAddres); - stream for parallel ConnectionRequest execution; - STUN parameter configuration (address, port, logins, etc.). Schedule for each firmware file: - support at the core; - CLI editor (optional). Personal authorization type identification, basic support. (optional).
Current Eltex.ACS version 1.12.350		

TARGET AUDIENCE

This operation manual is intended for technical personnel that performs server installation, configuration, monitoring, and maintenance. Qualified technical personnel should be familiar with the operation basics of TCP/IP & UDP/IP protocol stacks and Ethernet networks design concepts.

SYMBOLS

Symbol	Description
`Courier New`	Command entry examples, command execution results and program output are written in Courier New semibold.
<KEY>	Keyboard keys are written in upper-case and enclosed in angle brackets.

NOTES AND WARNINGS

Notes contain important information, tips, or recommendations on device operation and setup.

Warnings inform users about hazardous conditions which may cause injuries or device damage and may lead to the device malfunctioning or data loss.

CONTENTS

1 ELTEX.ACS SYSTEM GENERAL INFORMATION 6
1.1 Structure 6
1.2 Communication organization chart 8
1.3 Typical system operaion 9
2 HARDWARE REQUIREMETNS 12
3 *ELTEX.ACS* INSTALLATION AND CONFIGURATION 13
3.1 Server configuration, /etc/acsd.conf file description 13
3.2 STUN configuration, access to devices behind the NAT 16
4 ELTEX.ACS-BOX INSTALLATION 17
4.1 ELTEX.ACS-BOX connection 17
5 ELTEX.EMS.ACS INSTALLATION 19
5.1 Installation 19
5.2 Access ports configuration 19
5.3 Eltex.EMS.ACS time zone configuration 19
5.4 Connection to the management interface 20
6 SERVER RESERVATION AND BACKUP 21
6.1 ACS DB backup 21
6.2 EMS reservation 21
7 ACCESS PORTS REQUIRED FOR EXCHANGE 22
8 LICENSE 23
8.1 Licensing procedure description 23
8.2 License activation 23
8.3 License check 24
9 DESCRIPTION AND STRUCTURE OF THE ACS-CLI INTERFACE 25
9.1 General level commands 27
9.2 Commands available in all configuration modes (general commands) 27
9.3 Advanced settings ADVANCED level commands 28
9.4 Configuration settings. ACS-CONFIG level commands 31
9.5 Journalling settings ACS-JOURNAL level commands 32
9.6 Basic settings. SETTINGS level commands 33
9.7 Access settings. USERS level commands 34
9.8 Hardware class configuration. <HARDWARE CLASS> level commands 36
10 CPE — CPE PARAMETER CONFIGURATION MODE 41
10.1 Operations with parameter lists 41
10.2 Settings deletion 44
10.3 Direct commands 45
10.4 Edit configuration settings 48
10.5 View settings 50
10.6 Assign for deletion 51
11 FIRMWARE—CPE FIRMWARE UPDATE PARAMETERS' CONFIGURATION MODE 52
11.1 Add settings 52
11.2 Settings deletion 52
11.3 Edit settings 53
11.4 View settings 54
12 GROUP—GROUP CONFIGURATION MODE 56
12.1 Add settings 56
12.2 Batch transmission mode 58
12.3 Delete parameters 60
12.4 Direct commands 61
12.5 Group parameter editing – parameters deletion 63
12.6 Group parameter editing – parameters assigning 64
12.7 Generation for static groups 66
12.8 View settings 66
13 HARDWARE — HARDWARE PARAMETER CONFIGURATION MODE 67
13.1 Edit settings 67
13.2 View settings 67
14 PROFILE — PROFILE CONFIGURATION MODE 68
14.1 Add settings 68

14.2 Settings deletion ... 69
14.3 Change settings ... 70
14.4 View settings ... 72
15 SCHEDULE — FIRMWARE UPDATE SCHEDULE CONFIGURATION MODE ... 73
15.1 Edit settings ... 73
15.2 View settings ... 74
16 SERVICE — SERVICE CONFIGURATION MODE ... 75
16.1 Settings deletion ... 75
16.2 Edit settings ... 75
16.3 View settings ... 76
17 CONFIGURE PARAMETER OUTPUT MODE USING 'SHOW <PARAM> CONFIG' COMMAND ... 77
17.1 CPE parameter display mode configuration ... 77
17.2 Firmware update rule parameter display mode configuration ... 77
17.3 Hardware parameter display mode configuration ... 78
17.4 Profile parameter display mode configuration ... 78
17.5 Profile parameter display mode configuration ... 79
18 OPERATION WITH SCRIPTS ... 80
18.1 Scripts ... 80
18.2 Objects ... 80
19 ACS SERVER CONFIGURATION FOR ELTEX CPE ... 87
19.1 Server logon ... 87
19.2 NTE-RG configuration ... 87
19.3 NTE-RG rev.B Configuration ... 88
19.4 RG-14XX configuration for port registration on SIP server ... 88
19.5 Configure RG-14XX to perform inbound and outbound calls ... 89
19.6 TAU-8.IP configuration for port registration on SIP server ... 89
19.7 Configure TAU-8.IP to perform inbound and outbound calls ... 90
19.8 TAU-1.IP configuration for port registration on SIP server ... 90
19.9 Create firmware and assign it to CPE profile ... 91
19.10 Configure CPE firmware update schedule ... 92
20 FREQUENTLY ASKED QUESTIONS ... 93
APPENDIX A. DISTRIBUTION OF CLI ACS OPERATIONS BY ACCESS FLAGS ... 97
APPENDIX B SYNC AND NOSYNC MODES ... 113
APPENDIX C. INDIRECT ADDRESSING ... 115
APPENDIX D. AUTOMATIC UPDATE OF CPE LOGIN/PASSWORD FOR ACS SERVER CONNECTION. ... 117

1 ELTEX.ACS SYSTEM GENERAL INFORMATION

Structure

*** TR-069** (technical report 069) – one of specifications that describes **CWMP** (CPE WAN management protocol), where:

- **CPE** (customer premises equipment) – equipment that installed on the subscriber/client side;
- **WAN** (wide area network) – provider/operator network through which the connection between devices is establishing.

CWMP is intended for remote management of subscriber equipment using global network. Device management through this protocol involves deploying an **ACS** (Auto-Configuration Server) on provider network of a server or a cluster of dedicated servers.

*** Eltex.ACS** – **Eltex** company autoconfiguration server (**ACS** – Auto Configuration Server). The server supports **CWMP** and a lot of its specifications including **TR-069**. The **Eltex.ACS** is deployed on the operator/provider network, after which the devices can be routed to the server and the necessary configurations can be performed.

The **Eltex.ACS** allows you to solve many tasks related to the management of devices on the network, including:

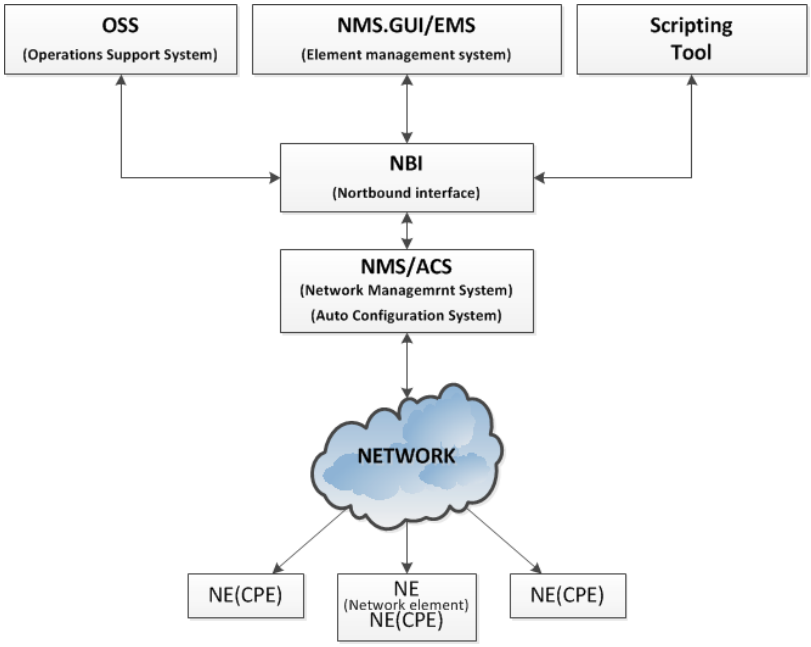
- drivers and microprogramms update;
- equipment and lines state diagnostics;
- changing necessary settings, reboot, etc.

*** Northbound Interface (NBI)** – Eltex.ACS API based on RESTful with JSON standard usage. The protocol is used to exchange arbitrary messages in JSON format. Using defined and prepared structures, you can send a request to the **Eltex.ACS**, which transmits them into corresponding requests to devices and organizes interaction with subscriber equipment. The devices also respond to requests to the server, and the server converts the responses back to JSON format and sends the response to the requested side via **NBI**. This interface is designed to integrate the **Eltex.ACS** into other systems that operator/provider uses to manage the existing infrastructure.

*** Eltex.ACS-BOX** — web interface for **Eltex.ACS** management that interacting via **NBI**.

***EMS** is a graphical interface that runs on **Java** and allows you to interact via **SNMP**. For the convenience of the **Eltex.ACS** was implemented to connect it to **EMS**, this bundle is called **Eltex.EMS.ACS**.

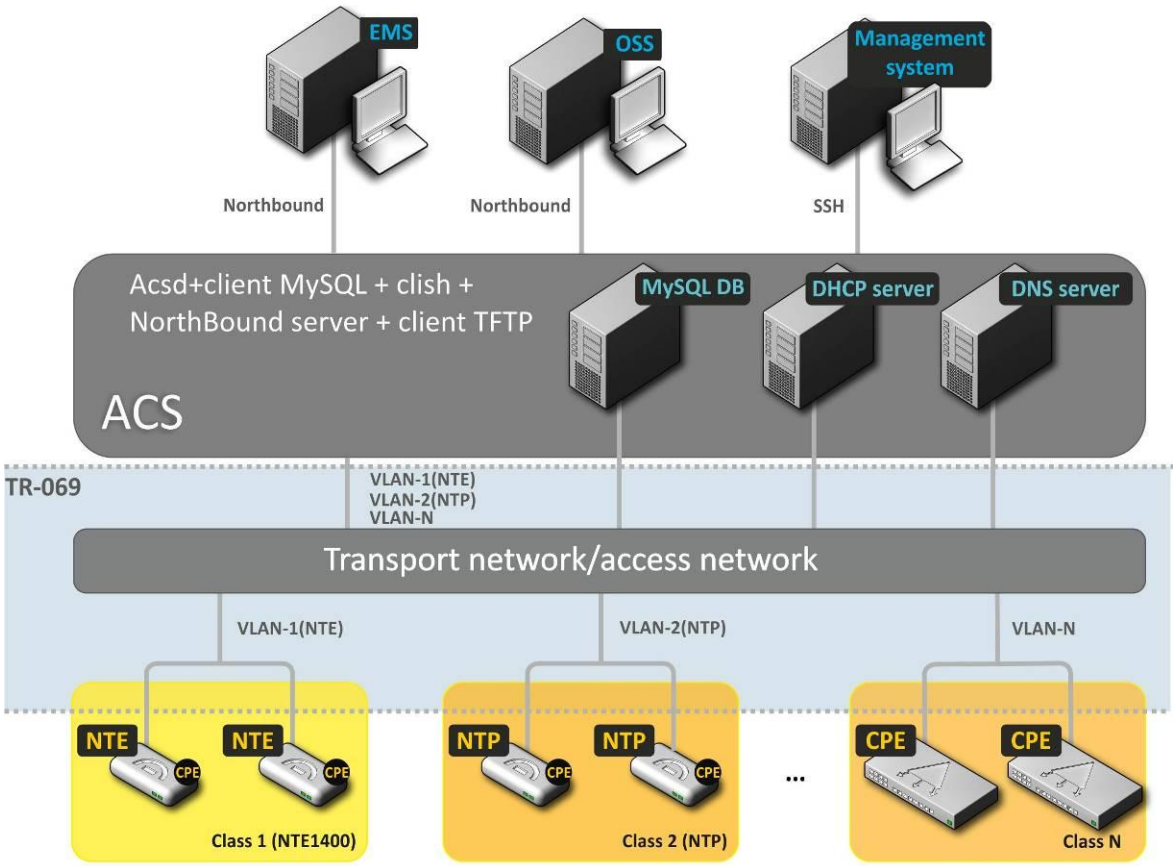
The general scheme of OSS-EMS-ACS to manage the CPE is shown in the following figure:



Eltex.ACS includes:

- **acsd** — the main executable file of the server (core of system), service that interacting with device;
- **DB (database)** — MySQL-based storage. DB stores server settings, individual and group profile device configurations, firmware update settings, journals and logs, various device classes data models, and other data essential for server operation;
- **NorthBound** — an API interface intended for **Eltex.ACS** integration with other systems;
- **acs-cli** - command line interace for server configuration.
- **Eltex.ACS-BOX** — main web interface for **Eltex.ACS** management that interacting via **NBI**.

Communication organization chart

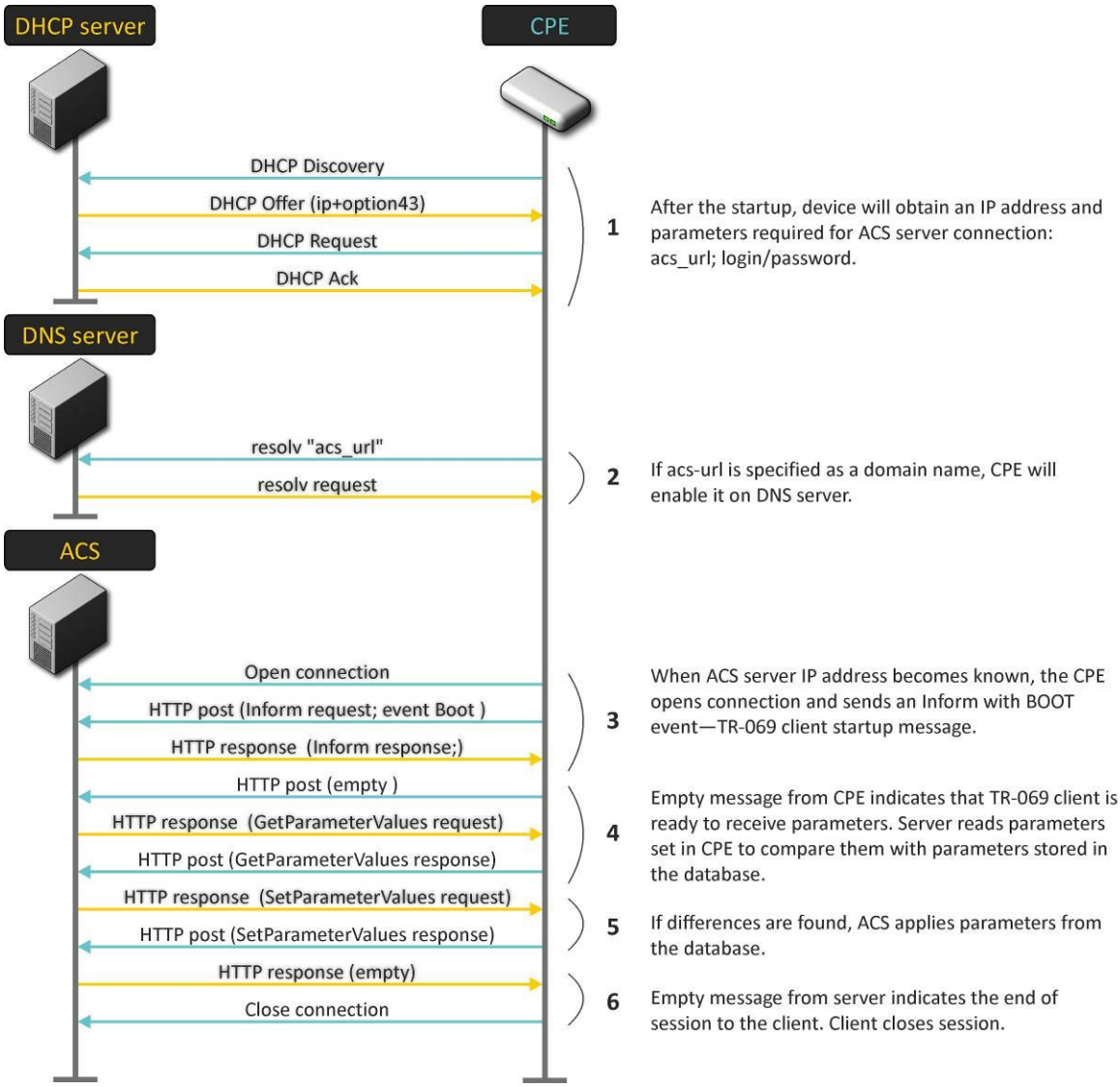


OSS - any centralized and automated operation system.

MySQL database, **DHCP**, and **DNS** are independent from the rest of the **Eltex.ACS**, and may be located on the same system as **Eltex.ACS**, or on standalone systems.

Typical system operaion

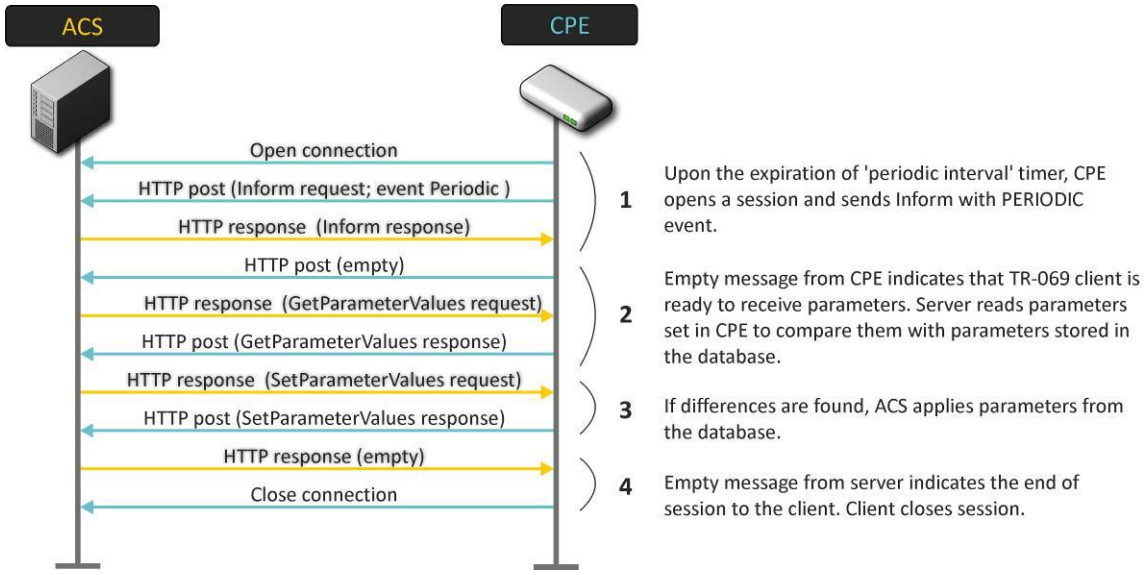
CPE startup



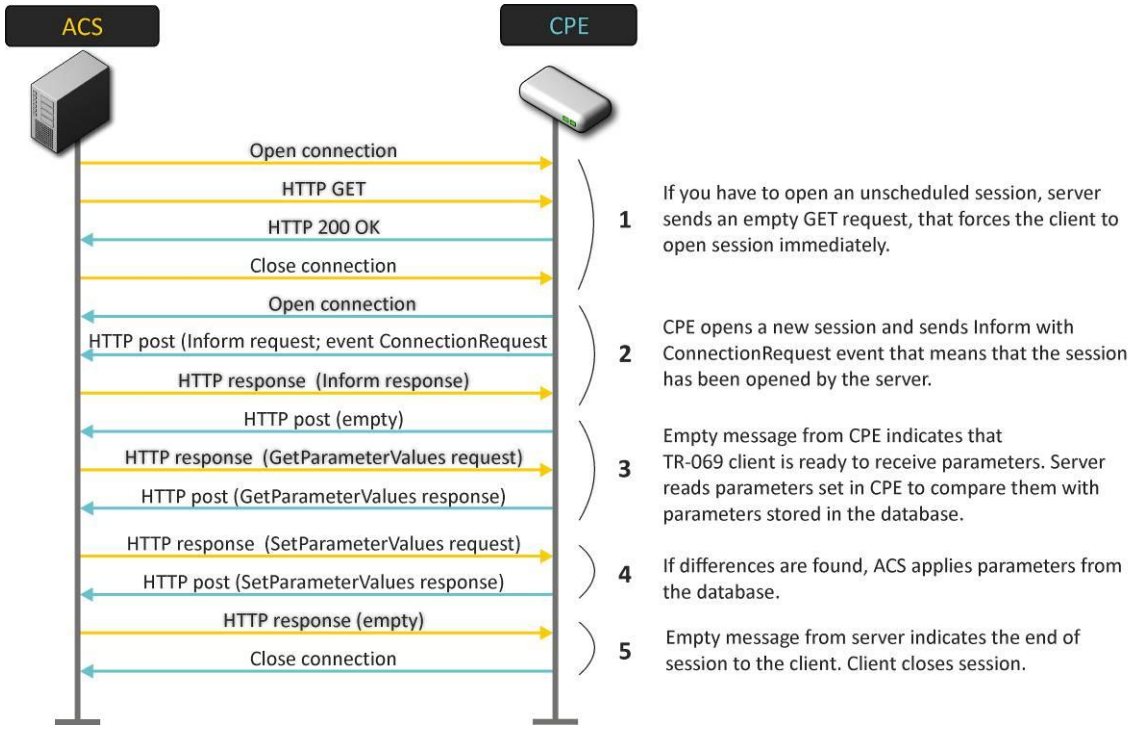
If CPE starts up with the necessary configuration present, Step 5 will be skipped. Server will compare all parameters, and will end the session, if they are found to be equal.

Periodic CPE synchronization with ACS

Periodic CPE and ACS server synchronization will be performed on the inform with the Periodic event. Inform sending frequency is defined by the 'periodic interval' parameter. This interval is configured by the ACS server, as well as other parameters.

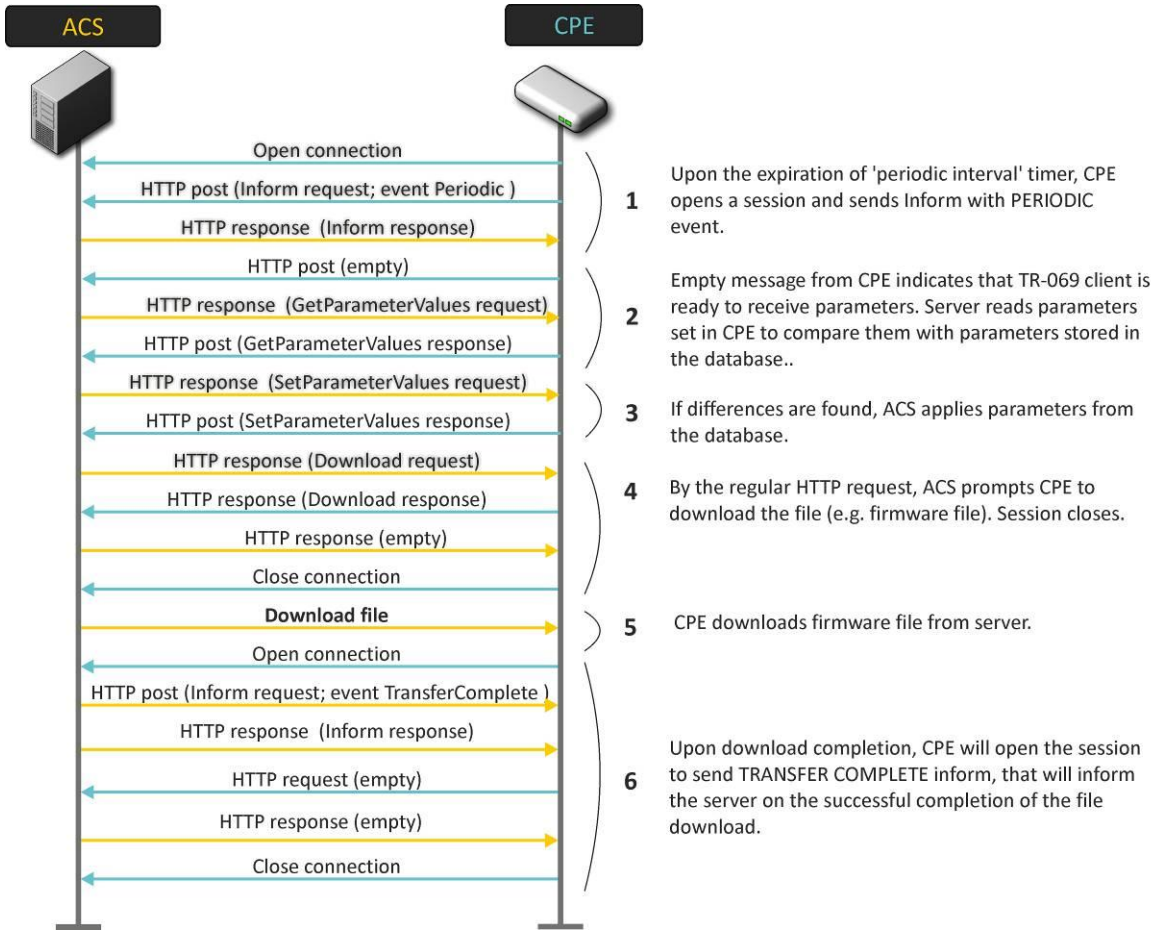


Unscheduled CPE synchronization with ACS



CPE firmware update

Firmware update may be performed either on the Periodic event (see example below), or on the Connection request event.



2 HARDWARE REQUIREMETNS

Typical Eltex.ACS + MySQL + Eltex.ACS-NBI + Eltex.ACS-BOX configuration (total 50 thousand ONT/CPE):	
CPU:	AMD Ryzen 5 1600X / Intel® Xeon® E5-2680
RAM:	16GB
DSS:	200 GB, 20 kIOPS

3 *ELTEX.ACS* INSTALLATION AND CONFIGURATION

ATTENTION: before starting the installation, make sure that the software repositories are available and that you have stable access to the Internet, and also make sure that the access to the network through the ports in section 7 is open.

Download the desired image at the following address:

ftp://ftp.eltex.org/ELTEX.ACS
Login: acsguest
Password: acsacs

1. Browse to the folder with last version of product.
2. Upload all packets for operation with USB keys on the server.
3. Navigate to the folder with relevant OS version and upload all packets to the server.

Installation should be performed with root (#) privileges.

4. Perform following commands on server in folder, where packets were uploaded:

```
# sudo apt-get update
# sudo apt-get install mysql-server
# sudo dpkg -i libzdb11_3.0.4_*.deb
# sudo dpkg -i libzdb-dev_3.0.4_*.deb
# sudo apt-get -yf install
# sudo dpkg -i libhal1_*
# sudo dpkg -i eltex-acs-db_1.12.350_*.deb
# sudo dpkg -i eltex-acs_1.12.350_*.deb
# sudo apt-get -yf install
```

5. After **eltex-acs-db** packet startup and installation you should enter login and password for the access to the DB.
6. After **eltex-acs** packet installation you should add the service to the startup of the system. For this execute following command:

```
# sudo update-rc.d acsd defaults
```

The user on whose behalf the work will be conducted must be in the acsd group. You can add the user into the group manually using the 'sudo usermod -a -G acsd <username>' command. After the addition, you need to re-authenticate the Ubuntu user (in some cases a system restart is required).

To add the user to the group use the following command:

```
usermod -a -G acsd <username>
```

Server configuration, /etc/acsd.conf file description

ATTENTION: Before starting the configuration, you must make sure that the network access by ports from section 7 is open, or to open access on the ports that you intend to use.

/etc/acsd.conf file is designed for more flexible **ACS core** configuration. It has the following parameters(settings listed above are default):

```
 [acsd]
# acsd process configuration section

# list of interfaces that will be used for server operation
# delimiter - semicolons without space (default 0.0.0.0, ie all addresses)
# if stun is used, the listening address must be given explicitly.
# if there are several addresses, only the first one will be used for outgoing stun requests.
listen_address = 0.0.0.0

#  TCP port for operation with CPE
listen_port = 9595

# list of interfaces that will support CPE session encryption .
#  certificate and key should be merged in one file and placed in /var/acsd/cert/cert_pkey.pem file
listen_address_ssl = 0.0.0.0

#  port for operation in ssl mode
listen_port_ssl = 9596

#  udp port for clish and Northbound operation
command_port = 9594

#  acsd operation directory
datadir = /var/acsd/

#  log storage directory
logdir = /var/log/eltex-acs/

#  log file size, permitted postfixes: b, k, m, g
log_size = 2M

#  each_cpe, individual device log, db log is stored in base
#  log_options = each_cpe; db

#  ConReq executing delay
conreq_nodelay = true

#  ending the session timeout on loss of communication with the CPE
cpe_recv_timeout = 40

# id classes, which included typing parameters, by default only for class UNKNOWN
dm_types_hwclass = UNKNOWN; TC

# number of records in a log, when this value is achieved, 20% of old records will be removed
journal_size =

# enable cookies
use_cookies = false

# autoflush interval for cookies
cookie_expire_time = 20


# enabling of a mechanism for requesting additional parameters through modules
plugin_modules = true
simultaneous_sync = true

# enable ConnectionRequest over UDP
#  accordingly, the external address must be determined and transmitted to the server from the device
use_stun = false

# setting parameters on the device to enable STUN
stun_enable = true

#  STUN-server address - at this address the device will connect to the STUN-server and determine its external address
stun_server_address = 192.168.0.10

#  STUN server port
stun_server_port = 3478
```

```
# login
stun_username =

# password
stun_password =

# STUN maximum keep alive period
stun_maximum_keep_alive_period = -1
# STUN minimum keep alive period
stun_minimum_keep_alive_period = 120

# limit of events on connection requests
udp_connection_request_address_notification_limit = 10

# ======= ACSD parameters ===========
# number of working threads
thread_workers = 8

# not used
# worker_maxclients = 700

# limit for connection processing. It is not recommended to set more than 32.
thread_pool_limit = 32

# number of simultaneous device sessions. The minimal value is 400.
db_pool_connection_limit = 400

[db]

# user name for database operations
username = acs

# password for database operations
password = password

# path to socket for local DB connection
socket = /var/run/mysqld/mysqld.sock

# DB host for remote connection
#host = localhost

# DB port
#port = 3306

[cli]


# admin password
passcode = 21232f297a57a5a743894a0e4a801fc3

# automatic login to the cli 1- enabled / 0 – disabled
autologin = 1

# login used for autologin
username = admin

# password used for autologin
password = admin
```

When modifications are made to the **/etc/acsd.conf** configuration file, you should restart **acsd**:

```
service acsd restart
```

STUN configuration, access to devices behind the NAT

ATTENTION: Before starting the configuration, you must make sure that the network access by ports from section 7 is open, or to open access on the ports that you intend to use.

1. Install a **stun** package:

```
# apt-get install stun
```

2. Configure the stun in **/etc/default/stun** file:

Configure the parameters **START_DAEMON**, **PRIMARY_IP**, **SECONDARY_IP** in **/etc/default/stun** for your network, e.g.:

START_DAEMON=true
PRIMARY_IP=192.168.0.10
SECONDARY_IP=127.0.0.1

Other parameters could be left with the default values.

3. Configure the **acsd** for STUN operation in **/etc/acsd.conf**:

Set a explicit IP address. When specifying multiple listening addresses, it is important to know that only the first one of listed will be used:

```
listen_address = 192.168.0.10
listen_address_ssl = 192.168.0.10
```

Enable STUN:

```
use_stun = true
# the same address as for PRIMARY_IP in /etc/defaults/stun
stun_server_address = 192.168.0.10
```

Start **stund**:

```
# service stun start
```

Now you can enable **STUN** on the device. Specify the **STUN** server address in the **stun_server_address** parameter.

4 ELTEX.ACS-BOX INSTALLATION

ATTENTION: Before starting the configuration, you must make sure that the network access by ports from section 7 is open, or to open access on the ports that you intend to use.

1. Start of installation, preparation of the environment:

```
# sudo apt-get update
# sudo apt-get -y remove openjdk*
# sudo apt-get -y install tomcat6 python-software-properties mongodb

# sudo add-apt-repository ppa:webupd8team/java
# sudo apt-get -y update
# sudo apt-get -y install oracle-java8-installer
# sudo echo "JAVA_HOME=/usr/lib/jvm/java-8-oracle" >> /etc/default/tomcat6

# sudo apt-get -yf install
```

2. Package installation

   Download the packages from ftp://ftp.eltex.org/ELTEX.ACS-BOX/eltex-acs-box_3.1-91
   Username: acs
   Password: acsacs

```
# sudo dpkg -i eltex-acs-nbi2_1.0.9-126_all.deb
```

   Choose **2**, **Use login-password authentication**.

   After packages installation change the credentials of the database connection (DB) **/etc/eltex-acs-nbi/nbi-acs.cfg** on those that are listed in the **/etc/acsd.conf** file.

3. Installation completion:

```
# sudo dpkg -i eltex-acs-box_3.1-91-common.deb
```

   Set the password for **admin** user or leave the default (**password**).

4. Update

   Updating is the same as installation:

```
# sudo dpkg -i eltex-acs-nbi_*
# sudo dpkg -i eltex-acs-box_*
```

ELTEX.ACS-BOX connection

Launch the web browser and enter http://acs-eltex.local:8080/acs-box in the address bar.

By default, consoles are connected to acs-eltex.local. If any other server or name is used, it is necessary to configure all devices to this address.

After connecting, a login dialog box will appear, where you need to enter the following username and password:

Login: **admin**
Password: **password**

After logging in, the device list page in the **STB** section will be displayed (by default).

If the device is already configured to interact with the server and exchange with it, the device name will appear in the list. Information on setting up a device for interacting with an ACS server can be obtained on our company's website in the 'Documents and Files' section (ACS Server Configuration Guide for CPE operation): https://eltex-co.com/catalog/eltex-acs-en.php?sphrase_id=176888.

5 ELTEX.EMS.ACS INSTALLATION

Installation

1. Download files:

   ems-prepare-deb.sh
   eltex-ems-db-3.7.0-113_all.deb
   eltex-ems-3.7.0-113_all.deb

   at the following address:

   ftp://ftp.eltex.org/ELTEX.EMS.ACS/3.7.0
   Login: acsguest
   Password: acsacs

2. Copy to the server user's home directory for installation (requires Internet access):

```
# sh ./ems-prepare-deb.sh
```

3. Use the following command to perform GUI installation/update:

```
# sudo dpkg -i eltex-ems-db-3.7.0-113_all.deb
# sudo dpkg -i eltex-ems-3.7.0-113_all.deb
# sudo apt-get install -fy
```

Access ports configuration

After installation, it is necessary to open access on the operational network to port **8080**.

An example command for iptables and the eth0 network interface is shown below. Outgoing calls must be allowed (by default for iptables, outgoing are allowed).

```
# iptables -A INPUT -i eth0 -p tcp --dport 8080 -j ACCEPT
```

Eltex.EMS.ACS time zone configuration

1. Download the archive located at:

https://cloud.eltex-co.com/index.php/s/mg5McNCdj6NdZ9P/download

2. Upload the archive to the server.

3. Extract the contents and execute:

```
java -jar tzupdater-2.2.0/tzupdater.jar -u
```

ATTENTION: You must make sure that the exchange with the www.iana.org domain on port 80 on your network is allowed, otherwise the installation will not be performed.

4. Restart **tomcat6**, **eltex-ems** services:

```
# sudo services tomcat6 restart
```

```
# sudo services eltex-ems restart
```

For more details the time zone and installing a package described at:

https://www.oracle.com/technetwork/java/javase/documentation/timezones-137583.html

Connection to the management interface

You can connect to the **ACS-GUI** graphical interface (**EMS-ACS**) by downloading the connection settings file through the browser. To do this, enter in the address bar of the browser:

```
http://<server address>:8080/ems/jws
```

Open the downloaded file using **JAVA.** During the boot process, an authentication input window will appear, where you need to enter:

User: admin
Password: <no password>

6 SERVER RESERVATION AND BACKUP

ACS DB backup

Every day at 6:25 (default settings cron) launches the script backup system: **/etc/cron.daily/acsd-backupdb**. As a result, in the directory **/var/acsd/backup/** create files **databases.sql.bz2 (.n)** - is compressed database dumps **acsmain, acscmds, acscache, acsinf**.

To restore the DB execute the following commands in **Linux** console:

```
# service acsd stop
# bunzip2 -c /var/acsd/backup/databases.sql.bz2 | mysql –u<USER> -p<PASS>
# service acsd start
```

If the Eltex.ACS files were also damaged, then the recovery is performed by reinstalling. For more details, see the section3.

EMS reservation

EMS reservation is implemented in the service and runs daily at 03:10.

Compressed database files are placed in the **/var/ems-backup/main** directory with names **eltex-ems-backup <DATE> -. <TIME> .gz**

To restore EMS database you should runthe script with the database parameter, for example:

```
cd /var/ems-backup/main
./ems-restore.sh eltex-ems-backup.<DATE>--<TIME>.gz
```

7 ACCESS PORTS REQUIRED FOR EXCHANGE

After installing the system to open the following ports specified in the following table:

Service	Port	Description
Web access	tcp: 8080	Connection and interaction with ACS-BOX. Connection to EMS.ACS.
EMS	tcp: 9310, 9340	Interaction with Eltex.EMS.ACS
ACS	tcp: 9595, 9596, 9998[1]	Devices interactions with Eltex.ACS.
STUN	udp: 3478, 3479	Connecting devices to the **STUN** server, for its external address determination.

[1] Port opens to the CPE side session initialization

8 LICENSE

Licensing procedure description

After installation, Eltex.ACS operates in demo mode — the total number of devices serviced by the server is limited — **100 devices maximum**. Next clients will register on the server, but you will not be able to connect to them to configure any parameters. When you try to interact with these devices information about the excess of the license threshold will be displayed in the log (the acsd log file).

Licensing includes two components:

Hardware key (ruToken / eToken)
license.bank file

Both components can be purchased in ELTEX commercial department (for STB-class devices (IPTV) preferential purchase is available). All the details can be found in ELTEX commercial department.

License activation

*** eToken:**

1. Download the **pkiclient_5.00.28-2_amd64.deb** package from FTP:

   ftp://ftp.eltex.org/acs/pkiclient_5.00.28-2_amd64.deb
   Login: acsguest
   Password: acsacs

2. Install the package and addons using following commands:

```
# sudo dpkg –i pkiclient_*
# sudo apt-get –y install opensc
```

3. Copy the**license.bank** file in directory**/var/acsd/license/**
4. Open the file using the command:

```
# sudo chmod +r /var/acsd/license/license.bank
```

5. Connect the **eToken** directly to the server and restart **acsd**:

```
# sudo service acsd restart
```

*** ruToken**[1]:

Acsd version shouldn't be less than 1.10.132.r47094.

1. Download the**librtpkcs11ecp_1.5.3.0-1_amd64.deb** package from FTP:

   ftp://ftp.eltex.org/acs/librtpkcs11ecp_1.5.3.0-1_amd64.deb
   Login: acsguest
   Password: acsacs

[1] Acsd version shouldn't be less than 1.10.131.r47079

2. Install the package and addons using following commands:

```
# sudo dpkg –i librtpkcs11ecp_*
# sudo apt-get –y install opensc
```

3. Copy the**license.bank** file in directory**/var/acsd/license/**

4. Open the file using the command:

```
# sudo chmod +r /var/acsd/license/license.bank
```

5. Connect the ruToken directly to the server and restart acsd:

License check

You can check license activation using acs-cli by running the following command:

```
# (acs)statistics
```

9 DESCRIPTION AND STRUCTURE OF THE ACS-CLI INTERFACE

Local connection from the server to the console is performed by the commands:

```
$ acs-cli
$ acscli-shell
```

When installing the ACS kernel, the special user ***acscli*** is added with which you can work remotely with the server over SSH.

A remote connection to the server's CLI interface is performing when establishing an SSH connection, specifying the ***acscli*** username and the password previously specified.

In both cases, you need to authenticate with the CLI interface. In the default configuration, the *admin* user is created with the password *admin*. You can add new users in the acs-users section.

The figure 1 shows the interrelation of command modes that are accessed from the **'acs'** mode.

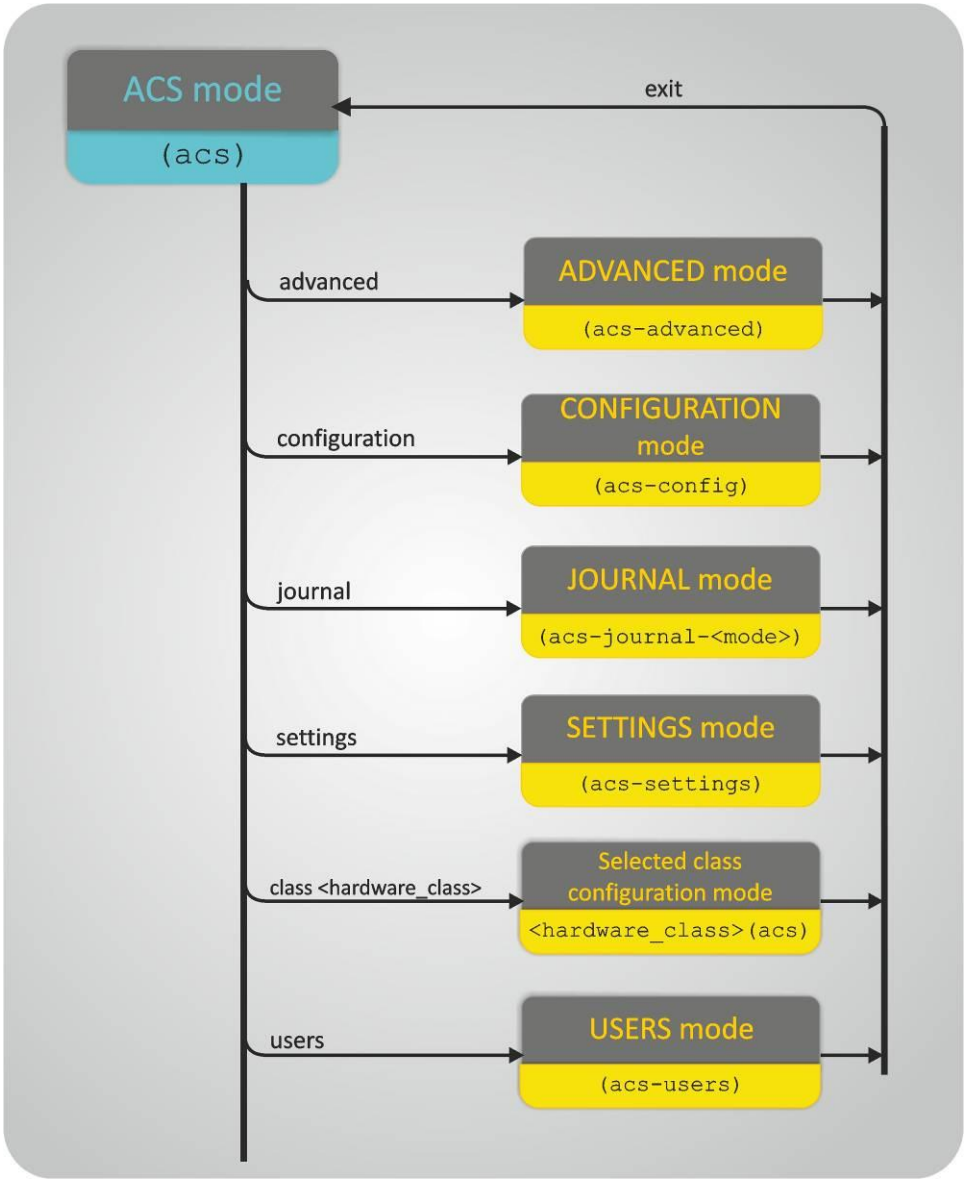


Figure 1 - Hierarchy of command modes of the 'acs' block

Fig. 2 shows the relation of the command modes available from the **'<hardware class>(acs)'** mode.

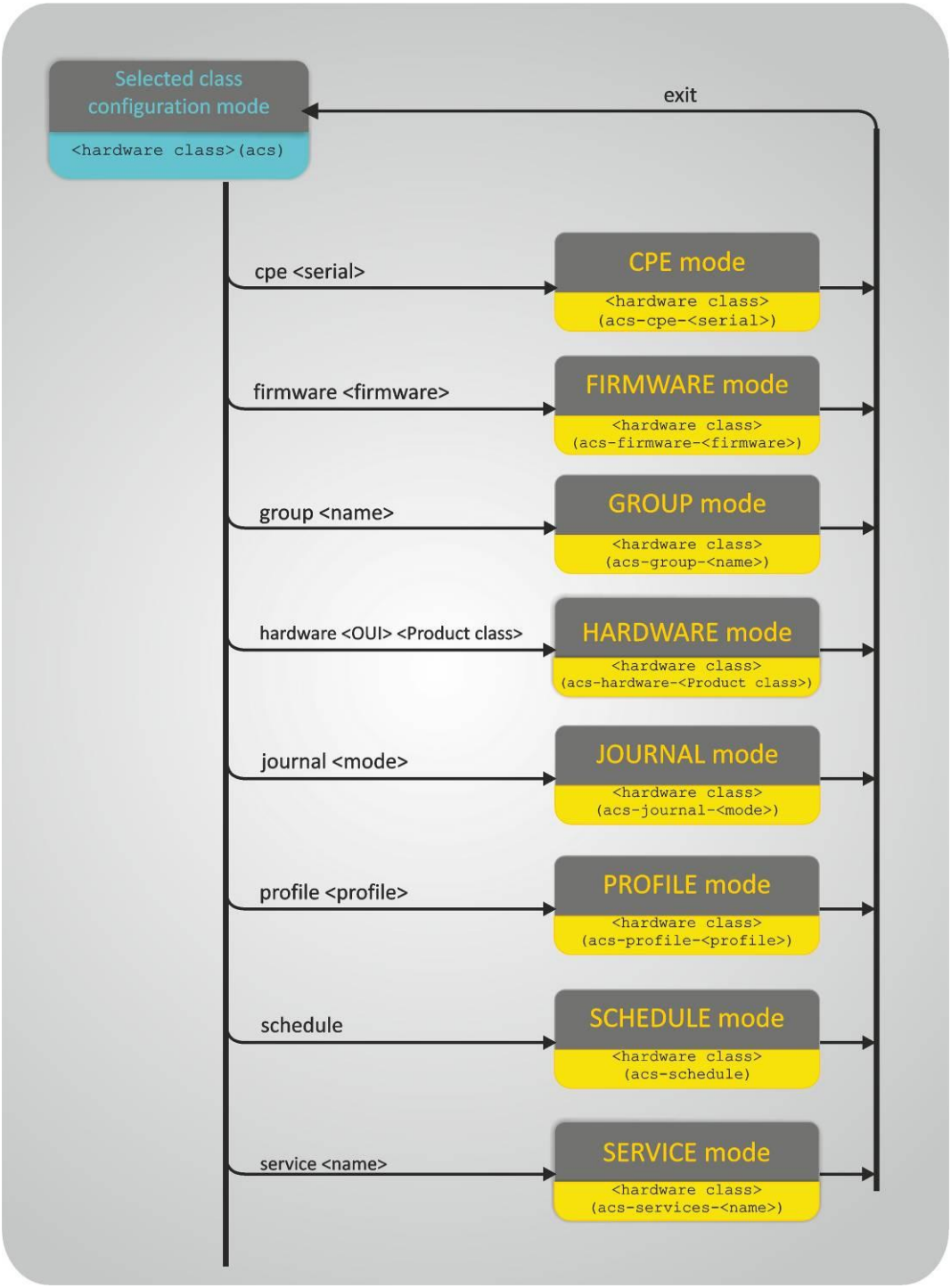


Fig. 2 — Command mode hierarchy of the '<hardware class>(acs)' block

Special aspects of command execution:

Some commands require entry confirmation:

```
(acs-config)default
Are you sure? (y/N):
```

If **y** key is pressed, the command will be executed;
If any other key is pressed, the command will not be executed.

Some commands require action selection (commit/rollback/cancel):

```
[Commit(c)/Rollback(r)/Cancel(Esc)]:
```

Commit — save changes to the database, enter **C** to execute;
Rollback — roll back changes, enter **R** to execute;
Cancel — cancellation, press <ESC> to execute.

Some commands require action selection (apply/ignore/cancel):

```
Attention! You must to restart acsd process after confirming changes in database! Do you
want to apply it? Your answer [Apply(a)/Ignore(i)/Cancel(Esc)]: :
```

Apply — save changes to the database and send acsd restart command, enter **a** to execute;
Ignore — save changes to the database, enter **i** to execute;
Cancel — cancellation, press <ESC> to execute.

The list, that contains more than 50 items, will be displayed in parts:

```
More? Enter - next line; Space - next page; Q - quit; R - show the rest.
```

When **<ENTER>** key is pressed, the next list line will be shown.
When **<SPACE>** key is pressed, the next list page will be shown (50 items max.)
When **R** character is entered, the list will be displayed to the last item without pagination.
When **Q** character is entered, the list display will be aborted.

General level commands

To proceed to configuration of system parameters, execute the **acs** command.

Table 11.1 — acs general level commands

Command	Parameter	Value	Action
advanced			Enter the advanced settings configuration section
class	<Hardware class>	NTE1400, NTE1400REVB, NTP, TAU, RG, TC, UNKNOWN	Enter the configuration section for the selected equipment class
configuration			Enter the ACS configuration operation mode
exit			Exit to the server console terminal
journal			Enter the ACS journal configuration and viewing section
logout			End user session
settings			Enter the ACS parameter configuration section
statistics			Show ACS statistics
sysinfo			Show system information
users			Enter the users, roles, and access rights configuration section
version			Show the ACS package build version
find cpe by	<ip>/<serial>		Search for CPE information by its serial number or IP address

Commands available in all configuration modes (general commands)

Table 2 — Basic commands available in all configuration modes

Command	Action
exit	Exit from any configuration mode to the upper level in CLI command hierarchy
logout	End the current user session

commit	Apply changes made to the current configuration
rollback	Discard all changes made to the current configuration
top	Go back to the root menu

Table 3 — Command parameters' system

Symbol	Parameter type
<Value>	Mandatory parameter, specified in any events
[Value]	Optional parameter, defined when necessary

Advanced settings ADVANCED level commands

ADVANCED mode is available from the global ACS mode.

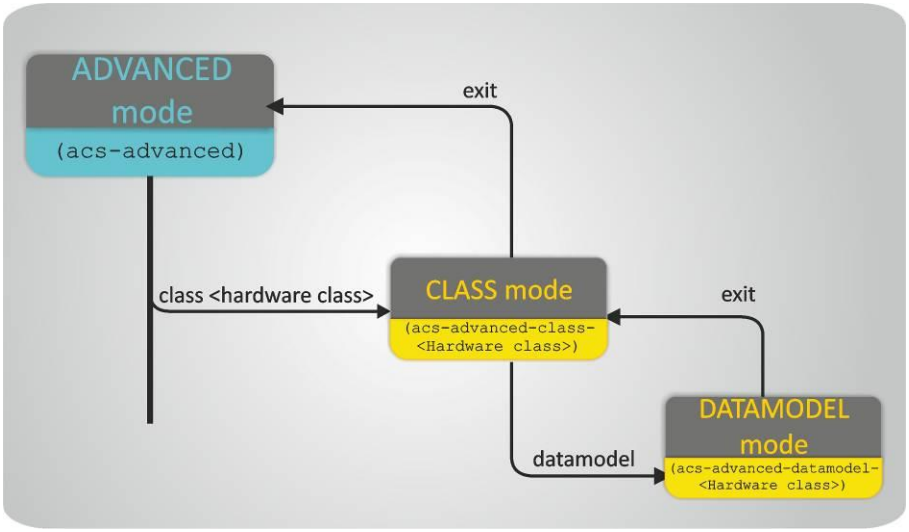


Fig. 3 — Command mode hierarchy of the '(acs-advanced)' block

After implementation of any changes into the ADVANCED section, you should restart acsd.

To proceed to configuration of settings, execute the **advanced** command.

Table 4 — acs-advanced level commands

Command	Parameter	Value	Action
add class	<Name> [Manufacturer]	String, 32 characters max. String, 32 characters max.	Add hardware class
add oui	<OUI> <OUI>	String, 32 characters max. String, 32 characters max.	Add a new binding (mapping) into the OUI-mapping table
class	<Hardware class>	Existing hardware class	Go to the hardware class parameter configuration
delete class	<Class>	Existing hardware class	Remove hardware class
delete oui	<OUI>	String, 32 characters max.	Remove binding (mapping) from the OUI-mapping table
show class hardware	<Hardware class>	Existing hardware class	Show hardware class specifications
show class list			Show the list of existing hardware classes

Command	Parameter	Value	Action
show oui list			Show OUI list
show oui mapped	<OUI>	Existing OUI	Show OUI-mapping table

Hardware class settings ADVANCED-CLASS level commands

To proceed to configuration of hardware class settings in the ADVANCED mode, execute the **class <Hardware class>** command, where <Hardware class> is an existing hardware class.

After implementation of any changes into the ADVANCED section, you should restart acsd.

Table 5 — acs-advanced-class-<Hardware class> level commands

Command	Parameter	Value	Action
add hardware	<OUI> <Product class>	Existing OUI value Equipment model	Add a new device class (hardware type)
datamodel			Go to equipment model parameters configuration
delete hardware	<OUI> <Product class>	Existing OUI value Existing equipment model	Remove device class
move hardware	<OUI> <Product class> <Hardware class>	OUI value Existing equipment model Destination class name	Move the model into a new class
set manufacturer	<Manufacturer>		Assign default manufacturer for a class
show hardware			Show the list of manufacturer id's and device classes

Device data model description ADVANCED-DATAMODEL level commands

To proceed to configuration of hardware class settings in the ADVANCED mode, execute the **datamodel** command.

After implementation of any changes into the ADVANCED section, you should restart acsd.

Table 6 — acs-advanced-class-<Hardware class> level commands

Command	Parameter	Value	Action
add parameter full	<Name>	String, 240 characters max.	Add a new parameter with the detailed specifications to the model description
	<Type>	String, 64 characters max.	
	<Writable property>	readonly/writable	
	<Min>		
	<Max>		
	<Length>	1..2147483647	
	<Default value>	1..240	
	<Version>	String, 32 characters max.	
	<Tr name>	Parameter TR name, 240 characters max.	
	[Flag]	Default/is_key/ no_sync/is_pass	
add parameter short	<Name>	String, 240 characters max.	Add a new parameter with the brief specifications to the model description
	<Type>	String, 64 characters max.	
	<Writable property>	readonly/ writable	
clear model			Remove all model description parameters
delete object	<Object name>	Object name from the list of existing objects	Remove the object with all assigned model parameters
delete parameter	<Name>	Parameter name	Remove the selected parameter from the model parameter list
set mode	<Field>	1..10	Configure parameter display mode *<Field >* — displayed fields: 1: parameter name; 2: type; 3: min. value; 4: max. value; 5: field length; 6: version; 7: default value; 8: read/write; 9: TR parameter name; 10: flag.
	[Mode]	on/off [on]	
	[Length]	1..250 [0]	
set parameter flag	<Name>	String, 240 characters max.	Set the indicator flag for the model parameter
	<Flag>	is_key/no_sync/ is_pass	
set parameter no flag	<Name>	String, 240 characters max.	Set the indicator flag for the model parameter
	<Flag>	is_key/no_sync/ is_pass	
set parameter pattern	<Name>	Parameter name	Edit parameter specification in the

Command	Parameter	Value	Action
			model description
	<Min>	Minimum parameter value	
	<Max>	Maximum parameter value	
	<Length>	1..2147483647	
	<Default value>	Default value, 240 characters max	
set parameter trname	<Name>	String, 240 characters max.	Modify parameter TR name in the model description
	<Tr name>	String, 240 characters max.	
set parameter type	<Name>	Parameter name	Modify parameter type in the model description
	<Type>	String, 64 characters max.	
	<Writable property>	readonly/ writable	
set parameter version	<Name>	Parameter name	Modify parameter version in the model description
	<Version>	Parameter version, 32 characters max.	
show list			Show the list of model description parameters
show mode			Show information display parameters
show model all			Show the list of model parameters
show model parameter	<Name>	Parameter name	Show the model description parameter specification

Configuration settings. ACS-CONFIG level commands

To proceed to configuration settings in the global ACS mode, execute the **configuration** command.

Table 7 — acs-config level commands

Command	Parameter	Value	Action
backup	<File>	Path to the configuration file upload location	Upload configuration file to the TFTP server
	<IP>	TFTP server IP address	
default			Reset ACS configuration to defaults, delete all loaded files
restore	<File>	Path to the configuration file	Download ACS server configuration from the TFTP server
	<IP>	TFTP server IP address	

Journalling settings ACS-JOURNAL level commands

To proceed to configuration settings in the global ACS mode, execute the **journal <view mode>** command, where <view mode> is a journal viewing mode (message type filter).

Table 8 — acs-log-<view mode> level commands

Command	Parameter	Value	Action
clear journal period	[from= Date from] [to= Date to]	Time and date in the following format: YYYY-MM-DD hh:mm Time and date in the following format: YYYY-MM-DD hh:mm	Delete journal entries for the selected period. The journal will be cleared entirely by default
set mode	<Field> [Mode] [Length]	1..18 on/off [on] 1..250 [0]	Configure the displayed parameters list. <*Field* > — displayed fields: 1: message number; 2: device serial number; 3: device event; 4: current date and time; 5: manufacturer; 6: manufacturer ID; 7: device model; 8: MaxEnvelopes; 9: restart counter; 10: general device information; 11: specification version; 12: hardware version; 13: firmware version; 14: initialization code; 15: connection request link; 16: device state; 17: interface; 18: Device IP address.
show entry	<ID>	1..2147483647	Show detailed information on the selected journal entry
show journal all			Show the list of journal entries
show journal last all			Show the list of the last journal entries
show journal last period	<Date from> <Date to>	Time and date in the following format: YYYY-MM-DD hh:mm Time and date in the following format: YYYY-MM-DD hh:mm	Show the list of the last journal entries for the specific period
show journal last serial	<Serial>	64 characters max.	Show the last journal entry related to the device with the specific serial number
show journal period	<Date from> <Date to>	Time and date in the following format: YYYY-MM-DD hh:mm Time and date in the following format: YYYY-MM-DD hh:mm	Show the list of journal entries for the specific period

Command	Parameter	Value	Action
show journal serial	<Serial>	64 characters max.	Show the list of journal entries related to the device with the specific serial number
view	<View mode>	full informs unauthorized errors	Change the journal display mode (message type filter)

Basic settings. SETTINGS level commands

To proceed to the basic settings in the global ACS mode, execute the **settings** command.

Table 9 — acs-settings level commands

Command	Parameter	Value	Action
add loglevel	<loglevel>	info/ error/ debug/ session/ params/ soap	Define the additional logging level
delete authorize type	<Interface>	Interface IP address	Delete authorization type data for the specific interface
delete authorize user	<Username> [Interface]	Default ACS login (64 characters max.) Server interface, default all	Remove default authorization parameters for the specific interface
set authorize type	<Interface> <Type>	Interface IP address auto/digest/ basic/none	Specify the authorization type for the interface
set authorize user	<Username> <Password> [Description] [Interface]	Default ACS login, 64 characters max. Default password, 64 characters max. String, 250 characters max. Server interface, default all	Define the default authorization settings for the ACS server
set cli_timeout	<Timeout>,sec	1..7200	Define the CLI session timeout in minutes
set loglevel	<loglevel>	disabled/ all/ info/ error/ debug/ session/ params/ soap	Define the logging level
set logoption	<Option name> <Option value>	Logging module option name enabled/disabled	Enable individual logging for each CPE
set password_mode	<Mode>	unsafe/ auto/ safe	Define the ACS server operation mode with passwords, assigned in CPE parameters

Command	Parameter	Value	Action
show logoption each_cpe	<Option name>	Logging module options	Show individual logging parameters
show settings authorize			Show the current ACS authorization parameters
show settings main			Show the current basic ACS parameters
set script	<svript_name>	1..250 [0]	Define the general script for all classes. For detailed command description, see Paragraph **18 Operation with scripts**
show file script all			View the server script list
set use_unknown	<Mode>	disable/enable	This parameter defines the UNKNOWN class utilization ('enable'—the class is enabled by default). All unknown devices will be added to this class
set cpe_auto_create	<Mode>	off/ goodpass/ all	CPE auto-creation mode (off—denied, goodpass—only authenticated, all—allowed for all)
cmd_timeout	<Timeout>,sec	1..7200	Time which should pass for the non-executed cpe command to enter the error state,sec (120s by default)

Access settings. USERS level commands

When a new user is created, you must fill in the name and password fields as they are required for each log-in (authorization).

USERS mode is designed for user and role management and available from the global ACS mode.

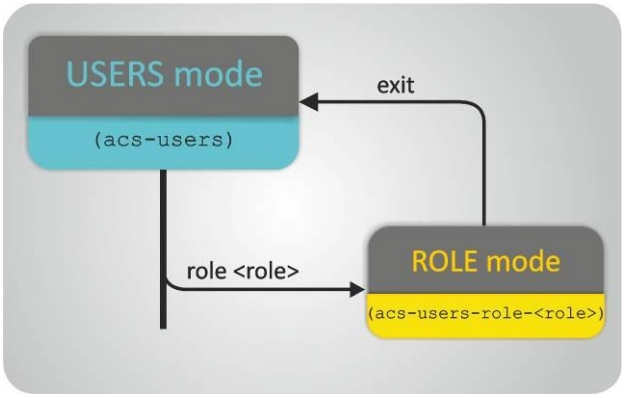


Fig. 4 — Command mode hierarchy of the '(acs-users)' block

To proceed to the basic settings in the global ACS mode, execute the **users** command.

Table 10 — acs-users level commands

Command	Parameter	Value	Action
add role	<Name> [Description]	String, 64 characters max. String, 250 characters max., description is not specified by default - []	Add a new role
add user	<Login> <Password>	Default ACS user name, 64 characters max. Default password, 64 characters max.	Add a new user

Command	Parameter	Value	Action
	<Role>	Role from the current role list	
delete user	<User>	User name in the current user list	Remove the specific user
delete role	<User>	User name from the current name list	Remove the specific role
role	<Role>	Role from the current role list	Go to role configuration
set mode	<Field> [Mode] [Length]	1..4 on/off 1..250 [0]	Configure the displayed parameters list
set password	<User> <Password>	User name from the current name list String, 250 characters max.	Specify a password for a user
set role	<User> <Role>	User name from the current name list Role from the current role list	Specify a role for a user
show user all			Show the list of existing system users
show mode			Show information display parameters
show role all			Show the list of existing roles
show role config	<Role>	Role from the current role list	Show the role settings

Role settings. USERS-ROLE level commands

Role mechanism is used as a basic principle of rights' distribution. Role is a logical entity that contains the list of specific parameters and defines the list of permitted actions. Roles are configured by the administrator according to the operator job duties.

To proceed to the basic settings in the USERS mode, execute the **role <role>** command, where <role> is a role from the existing role list**.**

Table 11 — acs-users-role-<role> level commands

Command	Parameter	Value	Action
add permission action	<Permission name>	The list of ready names	Add an action permission. For detailed description, see APPENDIX A. Distribution of CLI ACS operations by access flags
add permission class	<Hardware class>	Existing hardware class	Add the permission for hardware class usage
add permission pattern	<Permissions>	String, 3000 characters max.	Add an action permission (permission list delimited by '|')
clear permission action			Delete all action permissions
clear permission all			Delete all permissions
clear permission class			Delete all equipment model utilization permissions
delete permission class	<Hardware class>	Existing hardware class	Delete the specific hardware class utilization permission
delete permission pattern	<Permissions>	String, 3000 characters max.	Delete the specific template action permission (permission list delimited by '|')
delete permission simple	<Permission name>	The list of ready names	Delete the specific action permission
set description	<Description>	String, 250 characters max.	Define the role description
show config			Show the role settings

Hardware class configuration. <HARDWARE CLASS> level commands

To proceed to configuration of the specific class hardware in the global ACS mode, execute the **class <hardware class>** command, where <hardware class> is an existing hardware class.

Table 12 — <hardware class>(acs) level commands

Command	Parameter	Value	Action
add cpe	<Serial> <OUI> <Product class>	Device serial number String, 32 characters max. String, 64 characters max.	Add a device by its serial number, organization ID and model
move cpe config	<Serial> <Serial>	Old CPE serial number New CPE serial number	Copy configuration from one CPE to another, and delete the first CPE from the database
add firmware	<Firmware name>	Name of the firmware update rule, 64 characters max.	Add the firmware update rule for the hardware class
add group dynamic	<Name>	String, 128 characters max.	Add a new dynamic group
add group static	<Name>	String, 128 characters max.	Add a new static group
add hardware	<OUI> <Product class> [Manufacturer]	String, 32 characters max. String, 64 characters max. String, 32 characters max.	Add a hardware by organization ID, model and manufacturer (is not defined by default)
add private	<Param name>	String, 240 characters max.	Add a relationship between the long and short parameters.

Command	Parameter	Value	Action
	<Private name> [Check mode]	String, 240 characters max. nocheck/check	
add profile	<Name> [Base profile]	String, 64 characters max. Base profile name, not defined by default [-]	Add a new device profile
add service	<Name> <Description>	String, 64 characters max. String, 250 characters max.	Add a new device service
copy file firmware	<IP> <File>	IP address in XXX.XXX.XXX.XXX format String, 250 characters max.	Copy firmware file from the TFTP server to the selected folder
cpe	<Serial>		Go to CPE section. For detailed section command description, see Paragraph **10 CPE — CPE parameter configuration mode**
delete cpe	<Serial>		Remove the device by its serial number
delete file firmware	<File>	String, 250 characters max.	Remove the firmware file
delete firmware	<Firmware name>	Name of the firmware update rule, 64 characters max.	Remove the firmware update rule
delete group	<Name>	128 characters max.	Remove the specific group
delete hardware	<OUI> <Product class>	String, 32 characters max. String, 64 characters max.	Remove the specific hardware
delete private index	<Param name> <Index>	Name of the short parameter, 250 characters max. 1..250	Delete the specific short parameter by its index
delete private name	<Private name>	Name of the long parameter, 250 characters max	Delete the short parameter permanently or remove one of the related long parameters
delete private param	<Param name>	Name of the short parameter, 250 characters max.	Delete the specific short parameter
delete profile	<Profile>	Existing profile name	Remove the specific profile
delete service	<Name>	Existing service name	Remove the specific service
firmware	<Firmware>	Existing firmware update rule name	Enter the configuration section for the selected firmware update rule. For detailed section command description, see Paragraph **11 FIRMWARE—CPE firmware update parameters' configuration mode**
group	<Name>	Existing group name	Enter the configuration section for the specific group. For detailed section command description, see Paragraph **12 GROUP—**

Command	Parameter	Value	Action
			group configuration mode
hardware	<OUI> <Product class>	Existing manufacturer ID Existing equipment model	Enter the specific hardware section. For detailed section command description, see Paragraph **13 HARDWARE — hardware parameter configuration mode**
journal errors			Enter the specific journal configuration section
journal full			Enter the specific journal configuration section
journal informs			Enter the specific journal configuration section
journal unauthorized			Enter the specific journal configuration section
profile	<Profile>	Existing profile number	Enter the specific profile configuration section. For detailed section command description, see Paragraph **14** PROFILE — profile configuration mode
schedule			Enter the schedule configuration section. For detailed section command description, see Paragraph **15 SCHEDULE — firmware update schedule configuration mode**
service	<Name>	Existing service name	Go to configuration of the specific service. For detailed section command description, see Paragraph **16** SERVICE — service configuration mode
set cpe mode	<Field> [Mode] [Length]	1..24 on/off [on] 1..250 [0]	Configure the parameter display mode for the device configuration viewing. For detailed command description, see Paragraph **17** Configure parameter output mode using 'show <param> config' command
set firmware mode	<Field> [Mode] [Length]	1..9 on/off [on] 1..250 [0]	Configure the parameter display mode for the firmware update rule configuration viewing. For detailed command description, see Paragraph **17** Configure parameter output mode using 'show <param> config' command
set hardware mode	<Field> [Mode] [Length]	1..32 on/off [on] 1..250 [0]	Configure the parameter display mode for the hardware configuration viewing For detailed command description, see Paragraph **17** Configure parameter output mode using 'show <param> config' command
set profile mode	<Field> [Mode] [Length]	1..14 on/off [on] 1..250 [0]	Configure the parameter display mode for the profile configuration viewing. For detailed command description, see Paragraph **17** Configure parameter output mode using

Command	Parameter	Value	Action
			'show <param> config' command
set service mode	<Field> [Mode] [Length]	1..4 on/off [on] 1..250 [0]	Configure the parameter display mode for the service configuration viewing. For detailed command description, see Paragraph **17** Configure parameter output mode using 'show <param> config' command
set common flag boot_load_tree	<mode>	enable/disable	Enable parameter tree reading from cpe on its first connection to server after the startup. For detailed description, see Paragraph **0 Configuration of parameter tree reading on** startup.
set common script	<script name>	1..250 [0]	Define the general script for all classes. For detailed description, see Paragraph **18** Operation with scripts
show common settings			Show general class settings
show cpe all			View full device list
show cpe config	<Serial>	Device serial number, 64 characters max.	Basic configuration, flags, firmware update modes, subscriber settings
show cpe full	<Serial>	Device serial number, 64 characters max.	Basic configuration, flags, firmware update modes, subscriber settings, assigned profiles, full list of specifications
show cpe include	<Search value>	String, 250 characters max.	View filtered device list
show cpe mode			Show parameter display settings for the device configuration viewing
show cpe not-provisioned			Show the list of devices w/o individual settings
show cpe property	<Serial>	Device serial number, 64 characters max.	Show device 'property' rules
show cpe service assigned	<Serial>	Device serial number, 64 characters max.	Show the list of services assigned to the specific device
show cpe sort	<Field name>	default/ serial/ Product/ Firmware/ Profile/ lastcontact/ editor	Show the list of devices filtered by the specific field
show cpe updated	<Editor>	Undef/auto/ cli/nbi/gui	Show the list of devices with parameter editor filter
show file firmware all			Show firmware file list
show firmware all			Show firmware update rule list
show firmware config	<Firmware>	Name of the existing firmware update rule	Show the configuration of the specific firmware update rule
show firmware link hardware	<Firmware>	Name of the existing firmware update rule	Show the list of hardware that uses the specific firmware update rule
show firmware link profile	<Firmware>	Name of the existing firmware update rule	Show the list of profiles that use the specific firmware update rule

Command	Parameter	Value	Action
show firmware mode			Show parameter display settings for the firmware update rule configuration viewing
show group all			Show the list of groups
show group cpe	<Name>	Name of the existing group	Show the list of devices that belong to the specific group
show hardware all			Show the list of hardware
show hardware config	<OUI> <Product class>	Existing manufacturer ID Existing equipment model	Show the configuration of the specific hardware
show hardware mode			Show parameter display settings for the hardware configuration viewing
show private all			Show the list of parameters
show private param	<Param name>	Name of the short parameter	Show the specific short parameter
show profile all			Show the list of profiles
show profile config	<Profile>	Existing profile name	Show the configuration of the specific profile
show profile cpe using	<Profile>	Name of the existing profile	Show the list of devices that use the specific profile
show profile mode			Show parameter display settings for the profile configuration viewing.
show profile property	<Profile>	Existing profile name	Show the profile specifications
show profile struct			Show the sorted list of profiles
show service all			Show the list of existing services
show service config	<Name>	Name of the existing service	Show the configuration of the specific service
show service mode			Show parameter display settings for the service configuration viewing
show service property	<Name>	Existing service name	Show parameters of the specific service

10 CPE — CPE PARAMETER CONFIGURATION MODE

If you set up actions should be prioritizing the use (in order of priority):

1: Individual parameters;
2: Group parameters;
3: Profile parameters.

That is, the intersecting parameters are replaced by their more urgent.

Operations with parameter lists

Switching into the parameter list operation mode

Description: *If you need to write/read multiple parameters simultaneously, you can create a get/set parameters list.*

There are two types of parameter lists:

get list — list of parameters which values should be read from CPE;
set list — list of parameters which should be written to CPE.

Command*:* **batch**
Command syntax: batch

Parameters: there are no parameters for this command.

Example: **NTP(acs-cpe-'TG17000324') batch**

Execution result:
```
NTP(acs-cpe-'TG17000324'-batch)
```

Meaning: You have entered the batch transmission mode with the ACS database transaction confirmation.

Add command into 'get' queue

Description: *Add the specific command into the 'get' queue.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **add get** <Name> [Check mode]

Parameters: *<Name>* command name;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Add command into 'set' queue

Description: *Add the specific command into the 'set' queue.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **add set** <Name> [Check mode]

Parameters: *<Name>* command name;

[Check mode] device parameters conformance check mode:

Nocheck — do not perform the conformance check;

Check — perform the conformance check (default value).

Clear 'get' queue

Description: *This command clears the 'get' queue.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **clear batch get**

Parameters: there are no parameters for this command.

Clear 'set' queue

Description: *This command clears the 'set' queue.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **clear batch set**

Parameters: there are no parameters for this command.

Remove command from 'get' queue by its name

Description: *Remove the command from the 'get' queue by its name.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **delete get name** <Name>

Parameters: *<Name>* command name.

Remove command from 'get' queue by its number

Description: *Remove the command from the 'get' queue by its number.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **delete get index** <Index>

Parameters: *<Index>* command number.

Remove command from 'set' queue by its name

Description: *Remove the command from the 'set' queue by its name.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **delete set name** <Name>

Parameters: *<Name>* command name.

Remove command from 'set' queue by its number

Description: *Remove the command from the 'set' queue by its number.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **delete set index** <Index>

Parameters: *<Index>* command number.

Clear 'get' queue and remove header

Description: *Clear the 'get' queue and remove the header.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **reset batch get**

Parameters: there are no parameters for this command.

Clear 'set' queue and remove header

Description: *Clear the 'set' queue and remove the header.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **reset batch set**

Parameters: there are no parameters for this command.

Remove 'get' queue header

Description: *Remove the 'get' queue header.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **send batch get**

Parameters: there are no parameters for this command.

Remove 'set' queue header

Description: *Remove the 'set' queue header.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **send batch set**

Parameters: there are no parameters for this command.

View 'get' queue

Description: *Show contents of the 'get' queue.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **show batch get**

Parameters: there are no parameters for this command.

View 'set' queue

Description: *Show contents of the 'set' queue.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **show batch set**

Parameters: there are no parameters for this command.

Assign 'get' queue header

Description: *Assign the 'get' queue header.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **title batch get** <Title>

Parameters: *<Title>* queue header, 250 characters max.

Assign 'set' queue header

Description: *Assign the 'set' queue header.*

Command syntax: <hardware class>(acs-cpe-<serial>-batch) **title batch set** <Title>

Parameters: *<Title>* queue header, 250 characters max.

Settings deletion

Reset CPE parameters

Description: *This command will delete all existing CPE parameters.*

Command syntax: <hardware class>(acs-cpe-<serial>) **clear property**

Parameters: there are no parameters for this command.

Reset CPE services

Description: *This command will delete all existing CPE services.*

Command syntax: <hardware class>(acs-cpe-<serial>) **clear service**

Parameters: there are no parameters for this command.

Remove firmware downgrade permission

Description: *This command will remove the permission flag for CPE firmware downgrade.*

Command syntax: <hardware class>(acs-cpe-<serial>) **delete flag downgrade**

Parameters: there are no parameters for this command.

Remove the safe mode upgrade permission

Description: *This command will remove the permission flag for CPE firmware safe mode upgrade.*

Command syntax: <hardware class>(acs-cpe-<serial>) **delete flag safe_upgrade**

Parameters: there are no parameters for this command.

Delete personal firmware

Description: *This command will remove the update rule for the firmware that is assigned as personal for CPE.*

Command syntax: <hardware class>(acs-cpe-<serial>) **delete personal firmware**

Parameters: there are no parameters for this command.

Delete short parameters

Description: *This command will delete the specific short CPE parameter.*

Command syntax: <hardware class>(acs-cpe-<serial>) **delete private** <Name>

Parameters: *<Name>* parameter name.

Delete parameters

Description: *This command will delete the specific CPE parameter.*

Command syntax: <hardware class>(acs-cpe-<serial>) **delete property** < Name>

Parameters: *<Name>* parameter name.

Delete services

Description: *This command will delete the specific CPE service.*

Command syntax: <hardware class>(acs-cpe-<serial>) **delete service** <Name> [Instance]

Parameters: *<Name>* service name;

[Instance] mapping parameter, may take values in the range 0..100, not specified by default.

Direct commands

Upload file to CPE

Description: *This command will upload the specific file type to the device.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct download** <File type> <Filename>

Parameters: *<File type>* uploaded file type: *firmware, config*;

<Filename> link or path to the file, 250 characters max.

Get CPE configuration parameters TR attribute value

Description: *This command will get the CPE configuration parameters' TR attributes.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct get attribute**<Name> [Check mode]

Parameters: *<Name>* configuration parameter name, 240 characters max;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Get TR parameter value from CPE

Description: *This command will get TR parameter(s) from CPE.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct get parameter** <Name> [Check mode]

Parameters: *<Name>* parameter name, 240 characters max;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Get CPE RPC methods

Description: *This command will get existing CPE RPC methods.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct get rpc methods**

Parameters: there are no parameters for this command.

Get CPE status

Description: *This command will get CPE status.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct get state services**

Parameters: there are no parameters for this command.

Echo test

Description: *This command will send the simple PING command from CPE to the specific host.*

Command syntax: <hardware class>(acs-cpe-<serial>)
direct ipping <Host> <Interface> <Num of reps> <Timeout> <Data block size> [DSCP]

Parameters: *<Host>* IP address or domain name of the host to be tested;

<Interface> WAN or LAN interface, that will be used for echo test;

<Num of reps> number of echo test repetitions before results are shown;

<Timeout> timeout in milliseconds, from 1ms to 2147483647 ms;

<Data block size> size of data block sent in each echo test, 65535bytes max;

[DSCP] DSCP field value in echo test packets, from 0 to 63, default value is 0.

Restart

Description: *This command will force CPE to restart.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct reboot**

Parameters: there are no parameters for this command.

Reconfiguration

Description: *This command will open an unscheduled session with CPE.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct reconfigure**

Parameters: there are no parameters for this command.

Wake-on-LAN

Description: *This command will turn on the CPE remotely with the special byte sequence.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct wakeonlan**

Parameters: there are no parameters for this command.

Set TR attribute for CPE configuration parameter

Command syntax: <hardware class>(acs-cpe-<serial>)**direct set attribute** <Name> <Attribute name> <Attribute value> [Check mode]

Parameters: *<Name>* configuration parameter name, 240 characters max;

<Attribute name> attribute name: notification, access;

<*Attribute value*> attribute value:

for notification—0, 1, 2;
for access—none, subscriber;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Set TR parameter value in CPE configuration

Command syntax: <hardware class>(acs-cpe-<serial>) **direct set parameter** <Name> <Value> [Check mode]

Parameters: <*Name*> configuration parameter name, 240 characters max;

<*Value*> parameter value;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Restore factory settings

Description: *This command will restore CPE factory settings.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct setfactdef**

Parameters: there are no parameters for this command.

Set cpe passwords

Description: *This command will send passwords to CPE.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct setpass**

Parameters: there are no parameters for this command.

Synchronization

Description: *This command will synchronize the CPE.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct sync**

Parameters: there are no parameters for this command.

Firmware Update

Description: *This command will update CPE firmware.*

Command syntax: <hardware class>(acs-cpe-<serial>) **direct upgrade**

Parameters: there are no parameters for this command.

Edit configuration settings

Assign name to configuration

Command syntax: <hardware class>(acs-cpe-<serial>) **set configname** <Config name>

Parameters: *<Config name>* configuration name, 128 characters max.

Assign customer ID

Command syntax: <hardware class>(acs-cpe-<serial>) **set customer** <Customer ID>

Parameters: *<Customer ID>* customer ID, 64 characters max.

Firmware downgrade permission

Description: *This command will enable CPE firmware downgrade to previous versions.*

Command syntax: <hardware class>(acs-cpe-<serial>) **set flag downgrade** [Downgrade]

Parameters: *[Downgrade]* status:

Enable—allow downgrading to previous versions (default value);
Disable—deny downgrading to previous versions.

Loyalty mode configuration for firmware update

Description: *This* command *will enable safe CPE firmware update mode. Firmware will be updated only when Inform with BOOT event is received, i.e. on device startup.*

Command syntax: <hardware class>(acs-cpe-<serial>) **set flag safe_upgrade** [Safe mode]

Parameters: *[Safe mode]* status:

Enable—enable safe update (default value);
Disable—disable safe update.

Configuration of parameter tree reading on startup.

Description: *This* command *will enable parameter tree reading on first CPE connection to server after the startup (when Inform with BOOT event is received).*

Command syntax: <hardware class>(acs-cpe-<serial>) **set flag safe_ set flag boot_load_tree** [Boot_tree]

Parameters: *[Boot_tree]* status:

Enable—enable (default value);
Disable—disable.

Set server authentication password

Description: *This command will set the user name, that CPE will use in order to access the server.*

Command*:* <hardware class>(acs-cpe-<serial>) **set password**

Command syntax: set password <Password>

Parameters: *<Password>* password for device authorization on server, 64 characters max.

Set individual firmware update rule

Description: *This command will assign the specific firmware update rule as an individual rule for a device.*

Command syntax: <hardware class>(acs-cpe-<serial>) **set personal firmware** <Firmware>

Parameters: *<Firmware>* name of firmware update rule from the list of existing rules.

Assign subscriber parameters

Command syntax: <hardware class>(acs-cpe-<serial>) **set private** <Name> <Value>

Parameters: *<Name>* subscriber parameter name from the list of existing parameters;

<Value> parameter value.

Assign configuration profile

Command syntax: <hardware class>(acs-cpe-<serial>) **set profile** <Profile>

Parameters: *<Profile>* profile name.

Assign configuration properties

Command syntax: <hardware class>(acs-cpe-<serial>) **set property** <Prop name> <Prop value> [Check mode]

Parameters: *<Prop name>* property name, 250 characters max;

<Prop value> property value, 250 characters max;

[Check mode] device data model conformance check mode:

nocheck—do not perform the conformance check
check — perform the conformance check (default value).

Assign services

Command syntax: <hardware class>(acs-cpe-<serial>) **set service** <Service name> [Instance]

Parameters: *<service name>* service name;

[instance] mapping parameter, may take values in the range 0..100, default value is 1.

Assign subscriber ID

Command syntax: <hardware class>(acs-cpe-<serial>) **set subscriber** <Subscriber ID>

Parameters: *<Subscriber ID>* subscriber ID.

Set server authentication username

Description: *This command will set the password, that CPE will use in order to access the server.*

Command syntax: <hardware class>(acs-cpe-<serial>) **set username** <Username>

Parameters: *<Username>* username for device authorization on server, 64 characters max.

View settings

View configuration

Command syntax: <hardware class>(acs-cpe-<serial>) **show config**

Parameters: there are no parameters for this command.

Example of command execution:

```
Information about CPE '020050005598':

        ID = 5739
     Serial = "1"
     Profile = "0: Default NTP-RG"
  Product class = "NTP-2"
    Firmware = ""
       URL = ""
   Config name = ""
  Last contact = "NULL"
  Cfg Upd time = "NULL"
   Cfg Upd res = ""
  Sfw Upd time = "2012-11-22 17:48:53"
   Sfw Upd res = ""
    Hardware = ""
   Cfg version = ""
    Username = ""
    Password = ""
    Authtype = 0
   Customer ID = ""
  Con req user = "admin"
  Con req pass = "admin"
    Cfg force = 0
   Subscriber = ""
     Edit by = "gui"
   Auth result = "-"

User property:

  voice1_enable = "-"
  voice1_number = "-"
 voice1_password = "-"
  voice2_enable = "-"
  voice2_number = "-"
 voice2_password = "-"
    sip_proxy = "-"
    ppp_login = "-"
   ppp_password = "-"
 internet_vlanid = "-"

Flags:

     Personal firmware: none
Firmware upgrade safe-mode: none
    Firmware downgrade: none
```

View full CPE configuration

Command syntax: <hardware class>(acs-cpe-<serial>) **show full**

Parameters: there are no parameters for this command.

View list of configuration properties assigned for CPE

Command syntax: <hardware class>(acs-cpe-<serial>) **show property**

Parameters: there are no parameters for this command.

View results of file uploads to CPE

Command syntax: <hardware class>(acs-cpe-<serial>) **show result download**

Parameters: there are no parameters for this command.

View results of echo tests for CPE

Command syntax: <hardware class>(acs-cpe-<serial>) **show result ping**

Parameters: there are no parameters for this command.

Example of command execution:

```
ELTEX_NTP(acs-cpe-'ELTX08000034')show results ping
CPE 'ELTX08000034', ip 'ya.ru': success 1, failure 0; resp. time average 67, min 67, max 67
Start: 2012-01-30 11:05:20;end: 2012-01-30 11:05:21.
```

View list of assigned services

Command*:* <hardware class>(acs-cpe-<serial>) **show service**

Command syntax: show service

Parameters: there are no parameters for this command.

Assign for deletion

Delete parameters

Command syntax: <hardware class>(acs-cpe-<serial>) **unset property** <Name> [Check mode]

Parameters: *<Name>* parameter name;

[Check mode] name conformance check mode:

nocheck — do not perform the conformance check;
check — perform the conformance check (default value).

11 FIRMWARE—CPE FIRMWARE UPDATE PARAMETERS' CONFIGURATION MODE

For command execution examples, see NTP class

Add settings

Link firmware update rule to equipment configuration profiles

Description: *This command will link all configuration profiles to the firmware update rule.*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **add link all profile**

Parameters: there are no parameters for this command.

Add new equipment model

Description: *This command will add a new equipment model to the specific firmware update rule.*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **add link hardware** <OUI> <Product class>

Parameters: *<OUI>* organization unique identifier;

<Product class> model name;

Example: NTP(acs-firmware-'test') **add link hardware A8F94B NTP-RG-1402G-W**

Meaning: *The equipment—model NTP-RG-1402G-W, manufacturer ID A8F94B—has been added for the current firmware update rule.*

When the file is assigned to the firmware update rule, the model list fills in automatically (only for NTE1400, NTE1400REVB, and NTP classes).

Assign equipment configuration profile to firmware update rule

Description: *This command will assign configuration profile for the specific firmware update rule.*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **add link profile** <Profile name>

Parameters: *<Profile name>* configuration profile name, 250 characters max;

Example: **add link profile NTP-2**

Meaning: *'NTP-2'* configuration profile has been assigned to the firmware update rule.

Settings deletion

Remove firmware update rule from the specific CPE device

Description: *This command will remove firmware update rule from the specific CPE device.*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **delete personal cpe** <Serial>

Parameters: *<Serial>* device serial number.

Remove device model from firmware update rule

Description: *This command will remove the device model from the list of the specific firmware update rule.*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **delete link hardware** <OUI> <Product class>

Parameters: *<OUI>* organization unique identifier;

<Product class> model name.

Remove update configuration profile from firmware update rule

Description: *This command will remove CPE configuration profile from the firmware update rule.*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **delete profile** <Profile name>

Parameters: *<Profile name>* configuration profile name.

Example: <hardware class>(acs-firmware-<firm_name>) **delete profile NTP-2**

Meaning: *'NTP-2'* configuration profile has been deleted from the firmware update rule.

Edit settings

Configure firmware update rule using firmware file header data

Command syntax: <hardware class>(acs-firmware-<firm_name>) **set file firmware** <Filename>

Parameters: *<Filename>* firmware file name, 250 characters max.

Configure firmware update profile

Command syntax: <hardware class>(acs-firmware-<firm_name>) **set filename**<Filename>

Parameters: *<Filename>* firmware file name, 250 characters max.

***'set filename'* command is used, when firmware file does not contain CPE version information or hardware list.**

Enable firmware downgrade

Description: *This command will enable CPE firmware downgrade to previous versions (with lower build number).*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **set flag downgrade** [Downgrade]

Parameters: *[Downgrade]* update status:

Enable—allow downgrading to previous versions (default value);
Disable—deny downgrading to previous versions.

Enable loyalty mode for firmware update

Description: *This command will enable safe CPE firmware update mode. Firmware will be updated only when Inform with BOOT event is received, i.e. on device startup.*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **set flag safe_upgrade** [Safe mode]

Parameters: *[Safe mode]* update status:

Enable—enable safe update (default value);
Disable—disable safe update.

Define current firmware update rule as exclusive rule for device

Description: *This command will assign the current firmware update rule as an exclusive rule for a device with the specific serial number.*

Command syntax: <hardware class>(acs-firmware-<firm_name>) **set personal cpe** <Serial>

Parameters: *<Serial>* device serial number.

Enable firmware update rule

Command syntax: <hardware class>(acs-firmware-<firm_name>) **set schedule** <Schedule mode>

Parameters: *<Schedule mode>* schedule mode:

Enabled—enabled—CPE firmware will be updated on schedule;
Disabled—disabled—firmware update will not be performed.

Specify external HTTP server URL

Command syntax: <hardware class>(acs-firmware-<firm_name>) **set url** <URL>

Parameters: *<URL>* external HTTP server URL, 250 characters max.

Assign firmware version to file specified in this profile

Command syntax: <hardware class>(acs-firmware-<firm_name>) **set version** <version>

Parameters: *<version>* firmware version, 32 characters max.

View settings

Display specific profile information

Command syntax: <hardware class>(acs-firmware-<firm_name>) **show config**

Parameters: there are no parameters for this command.

Display list of models, that the current profile will be applied to

Command syntax: <hardware class>(acs-firmware-<firm_name>) **show link hardware**

Parameters: there are no parameters for this command.

Display list of assigned configuration profiles for firmware update rule

Command syntax: <hardware class>(acs-firmware-<firm_name>) **show link profile**

Parameters: there are no parameters for this command.

12 GROUP—GROUP CONFIGURATION MODE

There are two types of groups:

static — group content changes only when commands are executed **generate cpe by filter** and **add cpe**;
dynamic — group content changes without the additional commands.

Add settings

Add CPE to device group list Only for static groups

Description: *This command will add a new device into static device group (the command is not used in dynamic groups).*

Command syntax: <hardware class>(acs-cpe groups-'<gr_name>/static') **add cpe** <Serial>

Parameters: *<Serial>* device serial number.

Add filters

Add 'from' date filter for the specific field

Description: *This command adds the filter that will show records of the specific type, generated after the defined date.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>)

add filter datetime from <Field> <Date from>

Parameters: *< Field >* field name: lastcontact, cfg_updtime, sfw_updtime

<Date from> date and time, that will be used as a starting point, in YYYY-MM-DD hh:mm format.

Add 'during' date filter for the specific field

Description: *This command adds the filter that will show records of the specific type, generated in the defined period of time.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>)

add filter datetime from <Field> <Date from> <Date to>

Parameters: *< Field >* field name: lastcontact, cfg_updtime, sfw_updtime

<Date from> date and time, that will be used as a starting point, in YYYY-MM-DD hh:mm format

<Date to> date and time, that will be used as an ending point, in YYYY-MM-DD hh:mm format.

Add 'long ago' time filter for the specific field

Description: *This command adds the filter that will show time records of the specific type, which value is greater than the defined value.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **add filter datetime older** <Field> <Minutes>

Parameters: *< Field >* field name: lastcontact, cfg_updtime, sfw_updtime

<Minutes> time, 1..2147483647 minutes.

Add 'recently' time filter for the specific field

Description: *This command adds the filter that will show time records of the specific type, which value is less than the defined value.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **add filter datetime under** <Field> <Minutes>

Parameters: *< Field >* field name: lastcontact, cfg_updtime, sfw_updtime

<Minutes> time, 1..2147483647 minutes

Add 'until' date filter for the specific field

Description: *This command adds the filter that will show records of the specific type, generated before the defined date.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **add filter datetime until** <Field> <Date to>

Parameters: *<Field >* field name: lastcontact, cfg_updtime, sfw_updtime

<Date to> date and time, that will be used as an ending point for reporting, in YYYY-MM-DD hh:mm format.

Add specific expression filter

Description: *This command will add a specific expression filter.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **add filter expression** <Expression>

Parameters: *<Expression> expression string, 1024 characters max.*

Add 'Editor' field value filter

Description: *This command will add a specific field value filter.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **add filter value editor** <Value>

Parameters: *<Value> field value: undef, auto, cli, nbi, gui*

Add template mask value filter

Description: *This command adds the filter that will show records of the specific type by their mask value.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **add filter wildcard** <Field> <Mask>

Parameters: *<Field >* field name: serial, product_class, profile, url, version, hardware, username, customerid, conrequser, configname, subscriber.

<Mask> regular expression, 250 characters max.

Expression syntax: https://en.wikipedia.org/wiki/Regular_expression

Batch transmission mode

Enter the batch transmission mode

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **batch**

Parameters: there are no parameters for this command.

Example: **NTP(acs-cpe groups-'dyn/dynamic') batch**

Execution result:
```
NTP(acs-group-'dyn'-batch)
```

Meaning: You have entered the batch transmission mode with the ACS database transaction confirmation.

Add command into 'get' queue

Description: *Add the specific command into the 'get' queue.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **add get** <Name> [Check mode]

Parameters: *<Name>* command name;

[Check mode] device model parameters conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Add command into 'set' queue

Description: *Add the specific command into the 'set' queue.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **add set** <Name> [Check mode]

Parameters: *<Name>* command name;

[Check mode] device parameters conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Clear 'get' queue

Description: *This command clears the 'get' queue.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **clear batch get**

Parameters: there are no parameters for this command.

Clear 'set' queue

Description: *This command clears the 'set' queue.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **clear batch set**

Parameters: there are no parameters for this command.

Remove command from 'get' queue by its name

Description: *Remove the command from the 'get' queue by its name.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **delete get name** <Name>

Parameters: *<Name>* command name.

Remove command from 'get' queue by its number

Description: *Remove the command from the 'get' queue by its number.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **delete get index** <Index>

Parameters: *<Index>* command number.

Remove command from 'set' queue by its name

Description: *Remove the command from the 'set' queue by its name.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **delete set name** <Name>

Parameters: *<Name>* command name.

Remove command from 'set' queue by its number

Description: *Remove the command from the 'set' queue by its number.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **delete set index** <Index>

Parameters: *<Index>* command number.

Clear 'get' queue and remove header

Description: *Clear the 'get' queue and remove the header.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **reset batch get**

Parameters: there are no parameters for this command.

Clear 'set' queue and remove header

Description: *Clear the 'set' queue and remove the header.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **reset batch set**

Parameters: there are no parameters for this command.

Execute 'get' queue commands

Description: *Execute 'get' queue commands.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **send batch get**

Parameters: there are no parameters for this command.

Execute 'set' queue commands

Description: *Execute 'set' queue commands.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **send batch set**

Parameters: there are no parameters for this command.

Assign 'get' queue header

Description: *Assign the 'get' queue header.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **title batch get** <Title>

Parameters: *<Title>* queue header, 250 characters max.

Assign 'set' queue header

Description: *Assign the 'set' queue header.*

Command syntax: <hardware class>(acs-group-<gr_name>-batch) **title batch set** <Title>

Parameters: *<Title>* queue header, 250 characters max.

Delete parameters

Delete all CPEs from group

Description: *This command will clear the group (command is valid only for static groups).*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **clear cpe**

Parameters: there are no parameters for this command.

Delete all filters for group

Description: *This command will delete all filters assigned to a group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **clear filter**

Parameters: there are no parameters for this command.

Delete CPE from group

Description: *This command will delete CPE with the specific serial number from the group (command is valid only for static groups).*

Command syntax: <hardware class>(acs-cpe groups-'<gr_name>/static') **delete cpe** <Serial>

Parameters: *<Serial>* device serial number.

Delete filter from group

Description: *This command will delete the filter with the specific number from the group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>)
delete filter index <Index>

Parameters: *<Index>* filter number.

Direct commands

Upload file to CPE

Description: *This command will upload the specific file type to group devices.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>)

direct download [*vendorspecific***]** <File type> <Filename>

Parameters: *<File type>* uploaded file type: *firmware, config.* If *vendorspecific* is present, then it is arbitrary

<Filename> link or path to the file, 250 characters max.

Assign TR attribute to CPE parameter

Description: *This command will assign TR attribute to CPE parameter.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct get attribute**<Name> [Check mode]

Parameters: *<Name>* configuration parameter name, 240 characters max;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Assign TR parameter to CPE

Description: *This command will assign TR parameter to group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct get parameter** <Name> [Check mode]

Parameters: *<Name>* configuration parameter name, 240 characters max;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Get CPE service status

Description: *This command will get service status from all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct get state services**

Parameters: there are no parameters for this command.

Group restart

Description: *This command will force-restart all devices in a group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct reboot**

Parameters: there are no parameters for this command.

Group reconfiguration

Description: *This command will open an unscheduled TR session with all devices in a group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct reconfigure**

Parameters: there are no parameters for this command.

Group Wake-on-LAN

Description: *This command will turn on the CPE groups remotely with the special byte sequence.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct wakeonlan**

Parameters: there are no parameters for this command.

Assign TR attribute of CPE parameter

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct set attribute** <Name> <Attribute name> <Attribute value> [Check mode]

Parameters:

<Name> configuration parameter name, 240 characters max;

<Attribute name> attribute name: notification, access;

<Attribute value> attribute value:

for notification—0, 1, 2;
for access—none, subscriber;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Set TR parameter value of CPE configuration

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct set parameter** <Name> <Value> [Check mode]

Parameters:

<Name> configuration parameter name, 240 characters max;

<Value> parameter value;

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Group restoration of factory settings

Description: *This command will restore factory settings for all devices in a group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct setfactdef**

Parameters: there are no parameters for this command.

Set passwords

Description: *This command will send passwords to group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct setpass**

Parameters: there are no parameters for this command.

Group firmware update

Description: *This command will update firmware for all devices in a group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **direct upgrade**

Parameters: there are no parameters for this command.

Group parameter editing – parameters deletion

Group specification deletion

Description: *This command will delete all host parameters of group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit clear property**

Parameters: there are no parameters for this command.

Group service deletion

Description: *This command will delete all services of group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit clear service**

Parameters: there are no parameters for this command.

Remove firmware downgrade flag

Description: *This command will remove firmware downgrade permission flag for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit delete flag downgrade**

Parameters: there are no parameters for this command.

Remove safe mode update flag

Description: *This command will remove the safe mode firmware update permission flag for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit delete flag safe_upgrade**

Parameters: there are no parameters for this command.

Remove individual firmware update rule

Description: *This command will remove the firmware update rule assigned as an individual rule for each CPE in a group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit delete personal firmware**

Parameters: there are no parameters for this command.

Delete short parameters

Description: *This command will delete the specific short parameters for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit delete private** <Name>

Parameters: *<Name>* parameter name.

Delete host parameters

Description: *This command will delete host parameters for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit delete property** <Name>

Parameters: *<Name>* parameter name.

Remove service

Description: *This command will delete the specific service for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit delete service** <Service name> [Instance]

Parameters: *<Service name>* parameter name;

[Instance] mapping parameter, may take values in the range 0..100, not specified by default.

Group parameter editing – parameters assigning

Assign name to configuration

Description: *This command will assign the configuration name for the CPE group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit set configname** <Config name>

Parameters: *<Config name>* configuration name, 128 characters max.

Define firmware downgrade permission

Description: *This command will define the firmware downgrade permission (denial) for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit set flag downgrade** [Downgrade]

Parameters: *[Downgrade]* allow downgrading to previous versions:

Enable—enable (default value);
Disable—disable.

Define safe mode firmware update permission

Description: *This command will define the safe mode firmware update permission (denial) for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit set flag safe_upgrade** [Safe mode]

Parameters: *[Safe mode]* allow safe mode firmware update:

Enable—enable (default value);
Disable—disable.

Set individual firmware update rule

Description: *This command will assign the firmware update rule that will be used as an individual rule for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit set personal firmware** <Firmware>

Parameters: *<Firmware>* existing firmware update rule name.

Assign short parameters

Description: *This command will assign short parameters to all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit set private** <Name> <Value>

Parameters: *<Name>* parameter name;

<Value> parameter value.

Assign profile

Description: *This command will assign profile for all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit set profile** <Profile>

Parameters: *<Profile>* existing profile name.

Assign host parameters

Description: *This command will assign host parameters to all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit set property** <Name> <Value> [Check mode]

Parameters: *<Name>* host parameter name;

<Value> host parameter value.

[Check mode] conformance check mode:

Nocheck — do not perform the conformance check;
Check — perform the conformance check (default value).

Assign service

Description: *This command will assign the specific service to all group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **edit set service** <Service name> [Instance]

Parameters: *<Service name>* parameter name;

[Instance] mapping parameter, may take values in the range 0..100, not specified by default.

Generation for static groups

Generation for static CPE list with filters

Description: *This command will generate parameter-filtered CPE list.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **generate cpe by filter**

Parameters: there are no parameters for this command.

View settings

View command execution status

Description: *This command will show group command execution status.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **show command result** <Result ID> [State]

Parameters: *<Result ID>* *result* identifier*;*

[State] filter by command status:

All—all commands (default value);
Waiting—waiting state;
Inprogress—execution in progress;
Done—execution finished;
Error—execution error.

View list of given commands

Description: *This command will show the list of commands given to CPE group.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **show command sent**

Parameters: there are no parameters for this command.

View group content

Description: *This command will show the list of group CPEs.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **show cpe**

Parameters: there are no parameters for this command.

View group filters

Description: *This command will show group filters.*

Command syntax: <hardware class>(acs-cpe groups-<gr_name>/<gr_type>) **show filter**

Parameters: there are no parameters for this command.

13 HARDWARE — HARDWARE PARAMETER CONFIGURATION MODE

Service section. This menu allows to configure the hardware list.

When new models are added, their parameters are filled in automatically.

Edit settings

Assign manufacturer

Command syntax: <hardware class>(acs-hardware-<product_class>) **set manufacturer** <Manufacturer OUI>

Parameters: *<Manufacturer OUI>* – manufacturer's unique identifier, 64 characters max;

Example: **NTP(acs-hardware-'NTP-RG-1402GCB-W2')set manufacturer Eltex**

Meaning: Manufacturer *'Eltex'* has been assigned for the equipment model NTP-RG-1402GCB-W2.

Assign default profile

Command syntax: <hardware class>(acs-hardware-<product_class>) **set profile** <Hardware version>

Parameters: *< Profile name >* configuration profile name, 250 characters max;

Example: **NTP(acs-hardware-'NTP-RG-1402GCB-W2')set profile test**

Meaning: Default profile 'test' has been assigned for the equipment model NTP-RG-1402GCB-W2

Assign hardware version

Command syntax: <hardware class>(acs-hardware-<product_class>) **set version** <Hardware version>

Parameters: *<Hardware version>* – hardware version, 64 characters max;

Example: **NTP(acs-hardware-'NTP-RG-1402GCB-W2')set version 1v3**

Meaning: Hardware version 1.3 has been assigned for the equipment model NTP-RG-1402GCB-W2

View settings

View hardware information

Command syntax: <hardware class>(acs-hardware-<product_class>) **show config**

Parameters: there are no parameters for this command.

Command execution result:

```
Information about hardware model 'NTP-RG-1402GCB-W2':
        ID = 3711
 Manufacturer OUI = "333333"
   Product class = "NTP-RG-1402GCB-W2"
   Manufacturer = "Eltex"
 Hardware version = "1v3"
  Default profile = "test"
```

14 PROFILE — PROFILE CONFIGURATION MODE

Add settings

Restrict firmware version compatibility

Description: *This command will restrict the compatibility of the current profile with specific versions of CPE firmware. If the firmware version of CPE with assigned profile differs from the permitted one, this profile will not work. If the restriction is disabled, profile will work with any CPE version.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **add constraint version** <Version mask>

Parameters: *<Version mask>* restriction mask, ' * ' character at the end of the mask expression represents any quantity of any characters, '?' character strictly represents 1 character anywhere in mask expression.

Example:

```
ELTEX_NTP(acs-profile-'auto_140xx')add constraint version "5.?.1*"
Ok
ELTEX_NTP(acs-profile-'auto_140xx')show config
Information about device profile 'auto_140xx':
…..

Compatibility constraints on versions:

  1: 2.?.1*
```

Meaning: Restricted profile is compatible only with CPE which firmware version conforms to the mask '2.?.1*', e.g. 2.6.105

Restrict equipment model compatibility

Description: *This command will restrict the compatibility of the current profile with different models of equipment. If the model of CPE with assigned profile differs from the permitted one, this profile will not work. If the restriction is disabled, profile will work with any CPE version.*

Command syntax: <hardware class>(acs-profile-<prof_name>)
add constraint hardware <OUI> <Product class>

Parameters: *<OUI>* manufacturer ID;

<Product class> equipment model;

Example:

```
ELTEX_NTP(acs-profile-'auto_140xx') add constraint hardware A8F94BNTP-RG-1402G
Ok
ELTEX_NTP(acs-profile-'auto_140xx')show config
Information about device profile 'auto_140xx':
…..

Compatibility constraints on hardware:

  1: NTP-RG-1402G
```

Meaning: Profile with such restriction is compatible with NTP-RG-1402G only.

Link profile to firmware update rule

Description: *This command will add a firmware update rule to the current profile.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **add link firmware** <Firmware name>

Parameters: *<Firmware name>* name of the firmware update rule, that should be added to this profile. The list of available firmware update rules will appear, when the command is executed.

Settings deletion

Delete all parameters from profile

Description: *This command will delete all parameters from the profile.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **clear property**

Parameters: there are no parameters for this command.

Delete basic profile settings

Description: *This command will delete basic profile settings.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete base**

Parameters: there are no parameters for this command.

Remove all compatibility restrictions

Description: *This command will remove restriction and enable compatibility of the current profile with different models of equipment.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete constraint hardware all**

Parameters: there are no parameters for this command.

Remove index compatibility restrictions

Description: *This command will remove restriction and enable compatibility of the current profile with equipment models by their indices.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete constraint hardware index** <Index>

Parameters: *<Index>* compatibility restriction index

Remove parameter compatibility restrictions

Description: *This command will remove restriction and enable compatibility of the current profile with equipment models by their parameter.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete constraint hardware value** <OUI> <Product class>

Parameters: *<OUI>* manufacturer ID;

<*Product class*> equipment model.

Remove firmware version compatibility restrictions

Description: *This command will remove all restrictions and enable compatibility of the current profile with specific versions of CPE firmware.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete constraint version all**

Parameters: there are no parameters for this command.

Remove firmware version compatibility restrictions using indices

Description: *This command will remove restrictions and enable compatibility of the current profile with specific versions of CPE firmware by their indices.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete constraint version index** <Index>

Parameters: <*Index*> compatibility restriction index

Remove firmware version compatibility restrictions using masks

Description: *This command will remove restrictions and enable compatibility of the current profile with specific versions of CPE firmware by their masks.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete constraint version value** <Version mask>

Parameters: <*Version mask*> name of the created mask

Unlink the profile from the firmware update rule

Description: *This command will remove a firmware update rule from the current profile.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete link firmware** <Firmware name>

Parameters: <*Firmware name*> name of the firmware update rule, that should be removed from this profile. The list of available firmware update rules will appear, when the command is executed.

Remove profile specifications

Description: *This command will remove the profile specification by its name.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **delete property** <Name>

Parameters: <*Name*> parameter name.

Change settings

Assign basic profile

Description: *This command will assign basic settings to the profile.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **set base** <Base profile>

Parameters: *<Base profile>* name of the profile, that should be used as a basic profile.

Assign profile description

Description: *This command will assign a description to the profile.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **set description** <Description>

Parameters: *<Description>* text description of the profile, 250 characters max.

Assign script to profile

Description: *This command will assign a script to the profile.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **set script** <name>

Parameters: *<name>* script name, should be in '/var/acsd/scripts/' directory

Show script list

Description: *This command will show the list of scripts, that could be assigned to the profile.*

Command: <hardware class>(acs-profile-<prof_name>) **show file script all**

Command syntax: there are no parameters for this command.

Assign exchange interval for CPE and ACS server

Description: *This command will assign the* exchange interval for CPE and ACS server.

Command syntax: <hardware class>(acs-profile-<prof_name>) **set inform_interval** <Inform interval>

Parameters: *<Inform interval>* exchange interval for CPE and ACS server, may take values from 60 to 2147483647 seconds.

Modify/add configuration profile settings

Description: *This command will modify/add device profile settings.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **set property** <Name> <Value> [Check mode]

Parameters: *<Name>* profile property name, 240 characters max;

<Value> profile property parameter value, 250 characters max;

[Check mode] conformance check mode:

nocheck—do not perform the conformance check
check—perform the conformance check (default value)

Example:

```
NTP(acs-profile-‘0’)set property
"InternetGatewayDevice.X_BROADCOM_COM_IGMPCfg.AvailableGroupsEntity.1.DestinationIPStart""
233.7.70.1"
```

View settings

View profile information

Description: *This command will show the specific profile information.*

Command: <hardware class>(acs-profile-<prof_name>) **show config**

Command syntax: there are no parameters for this command.

View CPE list

Description: *This command will show the list of CPEs, that the current profile is assigned to.*

Command syntax: <hardware class>(acs-profile-<prof_name>) **show cpe using**

Parameters: *<Name>* profile name.

Example: **show cpe 2**

Command execution result:

```
List of hosts by profile '2':
  ##  Serial    Profile   Product class Firmware   Last contact
  1: ELTX080001d6 2      NTP-RG-1402G-W 2.2.1928   2011-09-27 10:03:21
  2: ELTX08000022 2      NTP-RG-1402G-W 2.2.1928   2011-09-27 10:04:36
  3: ELTX080001bd 2      NTP-RG-1402G-W 2.2.1928   2011-09-27 10:04:19
  4: ELTX080001d2 2      NTP-RG-1402G-W 2.2.1928   2011-09-27 10:04:46
  5: ELTX08001ec8 2      NTP-RG-1402G-W 2.2.1928   2011-09-27 10:03:29
  6: ELTX080001d5 2      NTP-RG-1402G-W 2.2.1928   2011-09-27 10:04:46
  7: ELTX08000182 2      NTP-RG-1402G-W 2.2.1909   2011-09-27 10:04:45
```

Meaning: You will see the list of CPEs with assigned profile '2', that includes 7 devices.

View profile settings list

Description: *This command will show the profile settings list.*

Command: <hardware class>(acs-profile-<prof_name>) **show property**

Command syntax: show property

Parameters: there are no parameters for this command.

15 SCHEDULE — FIRMWARE UPDATE SCHEDULE CONFIGURATION MODE

This section allows to define a schedule. If 3 following conditions are fulfilled: 'time', 'day of the week', 'date', the firmware update will be performed.

For correct schedule operation, check the system time settings on the server.

Edit settings

Adjust daily schedule

Description: *This command allows to set/adjust the daily schedule.*

Command syntax: <hardware class>(acs-schedule) **set daily** <Time from> <Time to>

Parameters: *<Time from>* activation time in HH:MM format;

<Time to> deactivation time in HH:MM format.

Example: **set daily 09:00 19:00**

Execution result: `Ok`

Meaning: Daily schedule is assigned—active from 9:00 to 19:00.

Define default schedule settings

Description: *This command will define daily/weekly/periodic schedule settings.*

Command syntax: <hardware class>(acs-schedule) set default <Type>

Parameters: *<Type> schedule type:*

daily;
period;
weekly;

Default schedule parameters:

```
1: daily   00:00   23:59
2: period   2011-06-01 2099-12-31
3: weekly   1     7
```

Adjust activity period

Description: *This command allows to set/adjust the periodic schedule.*

Command syntax: <hardware class>(acs-schedule) **set period** <Time from> <Time to>

Parameters: *< Day from>* activation date, in YYYY-MM-DD format;

< Day to> deactivation date, in YYYY-MM-DD format.

Example: **set period 2011-09-01 2012-03-01**

Execution result: `Ok`

Meaning: Schedule is assigned—active from 1 September 2011 to 1 March 2012.

Adjust weekly schedule

Description: *This command allows to set/adjust the weekly schedule.*

Command syntax: <hardware class>(acs-schedule) **set weekly** <Time from> <Time to>

Parameters: < *Day from*> activation day, in the range from 1 to 7 (1—monday, 2—tuesday, 3—wednesday, 4—thursday, 5—friday, 6—saturday, 7—sunday);

< *Day to*> deactivation day, from 1 to 7.

Example: **set weekly 1 5**

Execution result:

```
Ok
```

Meaning: Weekly schedule is assigned—active from Monday to Friday.

View settings

Show firmware update schedule settings

Description: *This command will show assigned ACS schedules.*

Command syntax: <hardware class>(acs-schedule) **show config**

Parameters: there are no parameters for this command.

Settings deletion

Reset CPE services

Description: *This command will delete all service parameters.*

Command syntax: <hardware class>(acs-service-<serv_name>) **clear property**

Parameters: there are no parameters for this command.

Delete service parameters

Description: *This command will delete the specific service parameter.*

Command syntax: <hardware class>(acs-service-<serv_name>) **delete property** <Name>

Parameters: *<Name>* service parameter name, 240 characters max.

Edit settings

Edit service description

Description: *This command will allow you to edit the profile description*

Command syntax: <hardware class>(acs-service-<serv_name>) **set description** <Description>

Parameters: *<Description>* service description, 250 characters max.

Edit service parameters

Description: *This command will allow you to add new/modify service parameters. Service is a set of rules without strict linking to the object instance.*

Command syntax: <hardware class>(acs-service-<serv_name>) **set property** <Name> <Value> [Is param] [Check mode]

Parameters:

<Name> service parameter name, 240 characters max.

<Value> profile parameter value, 250 characters max.

[Is param] link parameter, *true/false.* Set value

'True' value means that parameters are unique for each CPE; when assigning service to CPE, this parameter should be defined in CPE customization setting, otherwise they will not be applied;

[Check mode] device data model conformance check mode:

nocheck — do not perform the conformance check;
check — perform the conformance check (default value).

Example:

```
NTP(acs-service-'test')set property
"InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANIPConnection.{i}.DefaultGatewa
y" "5.5.5.5" false
```

View settings

Show service configuration

Description: *This command will show the service configuration.*

Command syntax: <hardware class>(acs-service-<serv_name>) **show config**

Parameters: there are no parameters for this command.

Show service settings

Description: *This command will show the service parameters.*

Command syntax: <hardware class>(acs-service-<serv_name>) **show property**

Parameters: there are no parameters for this command.

17 CONFIGURE PARAMETER OUTPUT MODE USING 'SHOW <PARAM> CONFIG' COMMAND

CPE parameter display mode configuration

Description: *This command will allow you to select fields to be displayed, while viewing the list using the 'CPE show list' command.*

Command syntax: <hardware class>(acs) **set cpe mode** <Field num> [Mode] [Length]

Parameters: *<Field num>* field number, may take values in the range 1 ..22:

1—Device identifier;
2—CPE serial number;
3—CPE profile name;
4—Hardware name (device model);
5—Firmware version;
6—Address;
7—Configuration name;
8—The last known device connection date and time;
9—The last known configuration update date and time;
10—Configuration update status;
11—The last known firmware update date and time;
12—Firmware update status;
13—Hardware version;
14—Configuration version;
15—Username;
16—Password;
17—Authorization type;
18—User identifier;
19—Connection request counter (username);
20—Connection request counter (password);
21—Configuration intrusions;
22—Subscriber;
23—Name of the user that performed the editing;
24—Authorization result;

[Mode] field display mode:

On—field is visible (default value);
Off—field is hidden;

[Length] field length, from 1 to 250 characters.

Firmware update rule parameter display mode configuration

Description: *This command will allow you to select fields to be displayed, while viewing the firmware update rule configuration.*

Command syntax: <hardware class>(acs) **set firmware mode** <Field> [Mode] [Length]

Parameters: *<Field num>* field number, field numbers correspond to the following parameters:

1—Firmware update rule name;
2—Firmware version;
3—URL;
4—Firmware file name;
5—Schedule;
6—Loyalty mode for firmware update;
7—Firmware downgrade mode;
8—Hardware (equipment);
9—Profiles;

[Mode] field display mode:

On—field is visible (default value);
Off—field is hidden;

[Length] field length, from 1 to 250 characters.

Hardware parameter display mode configuration

Description: *This command will allow you to select fields to be displayed, while viewing the hardware configuration.*

Command syntax: <hardware class>(acs) **set hardware** <Field num> [Mode] [Length]

Parameters: *<Field num>* field number, field numbers correspond to the following parameters:

1—Equipment identifier;
2—Manufacturer's unique identifier;
3—Product class;
4—Manufacturer;
5—Hardware version;
6—Default profile for this class

[Mode] field display mode:

On—field is visible (default value);
Off—field is hidden;

[Length] field length, from 1 to 250 characters.

0—length is not defined.

Profile parameter display mode configuration

Description: *This command will allow you to select fields to be displayed, while viewing the profile configuration.*

Command syntax: <hardware class>(acs) **set profile mode** <Field num> [Mode] [Length]

Parameters: *<Field num>* field number, field numbers correspond to the following parameters:

1—Profile name;
2—Inform interval (exchange interval for CPE and ACS server);
3—Full statistics per day;
4—Store statistics;

5—Store journal;
6—Store parameter value;
7—Store parameter interval value;
8—Store parameters on change;
9—Store parameters on startup;
10—Basic profile;
11—Description;
12—Assigned firmware profiles;
13—Equipment model compatibility;
14—Firmware version compatibility;

[Mode] field display mode:

On—field is visible (default value);
Off—field is hidden;

[Length] field length, from 1 to 250 characters.

Profile parameter display mode configuration

Description: *This command will allow you to select fields to be displayed, while viewing the profile configuration.*

Command syntax: <hardware class>(acs) **set service mode** <Field num> [Mode] [Length]

Parameters: *<Field num>* field number, field numbers correspond to the following parameters:

1—Service identifier;
2—Service name;
3—Service description;
4—Linking parameter;

[Mode] field display mode:

On—field is visible (default value);
Off—field is hidden;

[Length] field length, from 1 to 250 characters.

18 OPERATION WITH SCRIPTS

Scripts

ACS server may be configured with scripts written in JavaScript. To configure CPE with the script, you have to assign it to the profile using the following command: `set script <script from /var/acsd/scripts/ directory>`. The script will be applied to all CPEs with the specific profile.

Scripts are stored in the following directory: */var/acsd/scripts/*

```
NTE1400(acs-profile-'SIP_nte')set script test.js
```

Functions

logger(log)

logger(log) function allows to record informational messages into the acsd log.

```
logger ('My message');      // Message "My message" will be recorded into the log.
logger ('info', 'Info message'); // Message "info message" will be recorded into the log
with                                    the info logging level.
```

exec

Exec function allows to execute an additional script from the current script. Both scripts should be in /var/acsd/scripts/ directory.

```
exec('db.js');                        //execute script db.js
```

Objects

db

db — an object, that enables access to the server database. Query() method allows to perform SELECT statement, Update() method—INSERT/UPDATE/DELETE statements.

```
var profile_id = 34
var query1 = "select id from deviceprofilebean where name=" + profile_id
var result = db.Query(query1);

var prop_name = 'InternetGatewayDevice.X_ELTEX_Config.pbx.fxs1.phone'
var prop_value = '12345678'
var ont_id = 10
var query2 = "INSERT INTO hostpropertybean (name,value,hostid) VALUES ('" + prop_name +
"', '" + prop_value + "', " + ont_id + ");" ;
var result = db.Update(query2);
```

cpeflags

cpeflags — an object, that enables access to the 'acsmain.hostflagbean' database flag table.

Object ***cpeflags.<name>*** represent the flag and contains one or all of the following properties: *int_value*, *str_value*, *tr_name*.

getAt(index) — return object flag by its array index.
setAt(index, object) — set the flag using the specific index.

Flags used by acsd:

fw_personal_id — personal firmware update rule identifier, int_value integer;
fw_boot_only — enable firmware update on startup only, int_value boolean;
fw_downgrade — enable firmware downgrade, int_value boolean;
noautoconfig — disable configuration of the following rules: property, int_value boolean.

Examples:

```
if (!cpeflags.autoconfig.hasOwnProperty('int_value')) {    //hasOwnProperty method allows
to determine the availability of the selected property for the object; in fact, this
method indicates flag presence/absence.
  cpeflags.autoconfig.int_value = 1; //if the flag is absent, it will be created with
int_value=1
}
else if (cpeflags.autoconfig.int_value == 1)  // if int_value=1, specify other values
{
  cpeflags.autoconfig.int_value = 2;
  cpeflags.autoconfig.str_value = 'zero';
  cpeflags.autoconfig.tr_name =
'InternetGatewayDevice.ManagementServer.PeriodicInformInterval';
}

log("cpeflags.autoconfig = " + cpeflags.autoconfig + ", int_value = " +
cpeflags.autoconfig.int_value +
  ", str_value = " + cpeflags.autoconfig. + ", tr_name = " + cpeflags.autoconfig.tr_name);

try {
  var flag0 = cpeflags.autoconfig.getAt(0); // assign object flag cpeflags.autoconfig to
the variable flag0.
  log('autoconfig[0]: name = ' + flag0.name + ', int_value = ' + flag0.int_value + ',
str_value = ' + flag0.str_value + ', tr_name = ' + flag0.tr_name);
} catch (e) {
  log('error', 'error on reading flag with index: ' + e.message);
}

var tmp = cpeflags.autoconfig.getAt(1); // assign the first object flag
cpeflags.autoconfig to the variable tmp.

if (tmp != null) { // if there are some data present in tmp, show the log
  log('aquired autoconfig[1]: int_value = ' + tmp.int_value + ', str_value = ' +
tmp.str_value + ', tr_name = ' + tmp.tr_name);
  exit();
}
// else assign autoconfig flag with the index 1
try {
  var flag1 = {};
  flag1.int_value = 4;
  flag1.str_value = 'sample';
  flag1.tr_name = 'noname';
  cpeflags.autoconfig.setAt(1, flag1);
} catch (e) {
  log('error aquiring flag at index 1: ' + e.message);
}
```

cpe

cpe — an object, that enables access to all inform structures and RPC methods, defined in the TR-069 protocol description.

cpe object methods (for description of properties' parameters, see the TR -069 protocol description):

GetRPCMethods () — return the array of methods supported by CPE.

```
// this block returns the list of supported methods into the server log.
var methods = cpe.GetRPCMethods ();
for (i = 0; i < methods.length; i++) {
    logger ('Method: '+methods[i]);
}
```

Download (object_of_parameter) — file download command (configuration, firmware). Object parameters have the following properties*: CommandKey, FileType, URL, Username, Password, FileSize, TargetFileName, DelaySeconds, SuccessURL, FailureURL.*

Returns an object with the following properties: *Status, StartTime* and *CompleteTime*.

```
var dlcmd = {};
 dlcmd.CommandKey = 'acsd-js-dl';
 dlcmd.FileType = '3 Vendor Configuration File';
 dlcmd.FileType = '3 Vendor Configuration File';
 dlcmd.Username = 'testuser';
 dlcmd.Password = 'testpass';
 dlcmd.FileSize = 0;
 dlcmd.TargetFileName = ' config.txt';
 dlcmd.DelaySeconds = 0;
 dlcmd.SuccessURL = 'http://ya.ru';
 .FailureURL = 'http://yandex.ru';
cpe.Download(dlcmd);
log('Download response: Status ' + response.Status + ', StartTime ' + response.StartTime +
'; CompleteTime ' + response.CompleteTime);
```

Upload (object_of_parameters) — file upload command (from CPE). Object parameters have the following properties: CommandKey, FileType, URL, Username, Password, DelaySeconds.

Returns an object with the following properties: *Status, StartTime* and *CompleteTime*.

```
var array_parameters = {};
  array_parameters.CommandKey = 'acsd-js-dl';
  array_parameters.FileType = '3 Vendor Configuration File';
  array_parameters.URL = 'http://10.255.240.200/test/config.txt';
  array_parameters.Username = 'testuser';
  array_parameters.Password = 'testpass';
  array_parameters.DelaySeconds = 0;
cpe.Download(array_parameters);
 log('Upload response: Status ' + response.Status + ', StartTime ' + response.StartTime +
'; CompleteTime ' + response.CompleteTime);
```

GetParameterValues (object_of_parameters_names) — get the parameter list from CPE**.**

Returns an object with the following properties: *Name* and *Value*.

```
var arr = new Array ();
arr [0] = 'InternetGatewayDevice.DeviceSummary';
var response = cpe.GetParameterValues (arr);
logger (response[0].Name+'='+response[0].Value);
```

SetParameterValues (object_of_parameters) — parameter setting method. The object list with Name and Value properties is used as method parameters.

If error occurs during the method execution, the exception will be thrown. Exception will not be thrown when method has been successfully executed. This method returns no result.

```
var parameters = new Array ();
parameters[0] = {Name: 'InternetGatewayDevice.IPPingDiagnostics.Host', Value:
'192.168.0.1'};
parameters[1] = {Name: 'InternetGatewayDevice.IPPingDiagnostics.NumberOfRepetitions',
Value: '2'};
parameters[2] = {Name: 'InternetGatewayDevice.IPPingDiagnostics.NumberOfRepetitions',
Value: '2'};
cpe.SetParameterValues (parameters, "commandKey");
```

AddObject (tree_object_name, parameterKey) — add object.

Returns an object with the following properties: InstanceNumber and Status.

```
var response = cpe.AddObject
('InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.', 'acsd-js-addobj')
logger(' Instance new obj = ' + response.InstanceNumber + ', ' + response.Status);
```

DeleteObject (object_name, parameterKey) — delete object.

Returns an object with the following property: Status.

```
var response = cpe.DeleteObject
('InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.3.', 'acsd-js-delobj')
logger(' del obj result = ' + response.Status);
```

SetParameterAttributes (array_of_parameters) — set parameter attributes.

Returns no result.

```
var parameters = new Array();
parameters[0] = {};
parameters[0].Name = ' InternetGatewayDevice.ManagementServer.PeriodicInformInterval’
parameters[0].Notification = 1;
parameters[0].NotificationChange = true;
parameters[0].AccessListChange = true;
parameters[0].AccessList = [ 'subscriber' ];
cpe.SetParameterAttributes (parameters);
```

GetParameterAttributes (array_of_parameters) — get parameter attributes.

Returns an object with the following properties: Name, Notification and AccessList.

```
var arr=new Array();
arr [0]='InternetGatewayDevice.ManagementServer.PeriodicInformEnable';
var arr = cpe.GetParameterAttributes (arr);
logger(responce[0].Name + ' notify = ' + responce[0].Notification + ' Access = ' +
responce[0].AccessList)
```

GetParameterNames (parameter_names_array, NextLevel) — get writable parameter field.

Returns an object with the following properties: Name and Writable.

```
var response = cpe.GetParameterNames('InternetGatewayDevice.', false);
for (var y = 0; y < response.length; y++)
log('Name = ' + response[y].Name + ' writable = ' + response[y].Writable)
```

Reboot (commandKey) — CPE restart command.

```
cpe.Reboot ("commandKey");
```

FactoryReset — command will reset CPE configuration to factory defaults.

```
cpe.FactoryReset ();
```

Test.js script example:

```
/* sample acsd script */
log('javascript from CPE session');
logger('openacs-like log function');
logger('soap', 'soap level message, turn ed off by default');
log('info', 'info level message');

var cond = cpe.Inform.CurrentTime instanceof Date;
log('curtime instanceof Date: ' + cond);


/* traverse cpe */
log('cpe.Inform.MaxEnvelopes type is ' + typeof cpe.Inform.MaxEnvelopes);
log('cpe.Inform.CurrentTime type is ' + typeof cpe.Inform.CurrentTime);

var d = 'Inform:\n';
for (let prop in cpe.Inform) {
  if (typeof cpe.Inform[prop] == "number" || cpe.Inform[prop] instanceof Date)
    d += prop + ' = ' + cpe.Inform[prop] + '\n';
}

d += '\nDeviceId:\n';
for (let prop in cpe.Inform.DeviceId)
  d += prop + ' = ' + cpe.Inform.DeviceId[prop] + '\n';
d += 'Prototype: ' + cpe.Inform.DeviceId.__proto__ + '\n';

var SESSION_TRANSFER_COMPLETE = false;
var SESSION_BOOT = false;
d += '\nEvents:\n';
for (let i = 0; i < cpe.Inform.Event.length; i++) {
  d += cpe.Inform.Event[i].EventCode + ' ' + cpe.Inform.Event[i].CommandKey + '\n';

  if (cpe.Inform.Event[i].EventCode.search('Download') != -1 ||
    cpe.Inform.Event[i].EventCode.search('7 TRANSFER COMPLETE') != -1)
    SESSION_TRANSFER_COMPLETE = true;

  if (cpe.Inform.Event[i].EventCode.search('1 BOOT') != -1)
    SESSION_BOOT = true;
}

d += '\nParameters:\n';
for (let i = 0; i < cpe.Inform.ParameterList.length; i++)
  d += ' ' + cpe.Inform.ParameterList[i].Name + ' = ' + cpe.Inform.ParameterList[i].Value
+ '\n';

d += '\nCPE-supported RPC methods:\n';
var meth = cpe.GetRPCMethods();
for (let m in meth)
  d += ' ' + meth[m] + '\n';

log(d);

d = 'GPN:\n';
try {
  /* if there is nothing contained in path, null will be returned */
  /* var names = cpe.GetParameterNames('InternetGatewayDevice.DeviceInfo.', false); */
  var names =
cpe.GetParameterNames('InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.'
, true);
  for (let i = 0; names && i < names.length; i++)
    d += names[i].Name + ' : ' + names[i].Writable + '\n';
  log(d);
} catch (e) {
  ' + names[i].Writable + '\n'; ' + e.message);
}

d = 'GPV:\n';
try {
  let values = [ 'InternetGatewayDevice.DeviceInfo.ModelName',
```

```
          'InternetGatewayDevice.DeviceInfo.Description',
          'InternetGatewayDevice.DeviceInfo.UpTime' ];

  let response = cpe.GetParameterValues(values);
  for (let i = 0; i < response.length; i++)
    d += response[i].Name + ' = ' + response[i].Value + '\n';
  log(d);
} catch (e) {
  log('error', 'error on GPV: ' + e.message);
}

d = 'SPV: ';
try {
  let values = [ { Name: 'InternetGatewayDevice.ManagementServer.PeriodicInformInterval',
           Value: '1600' } ];
  let response = cpe.SetParameterValues(values, 'acsd-js-paramkey');
  d += response;
  log(d);
} catch (e) {
  log('error', 'error on SPV: ' + e.message);
}

d = 'GPA:\n';
try {
  let names = [ 'InternetGatewayDevice.ManagementServer.PeriodicInformInterval',
         'InternetGatewayDevice.DeviceInfo.ModelName',
         'InternetGatewayDevice.DeviceInfo.Description',
         'InternetGatewayDevice.DeviceInfo.UpTime' ];

  let response = cpe.GetParameterAttributes(names);

  for (let i = 0; i < response.length; i++) {
    d += response[i].Name + ' notify ' + response[i].Notification;
    if (response[i].hasOwnProperty('AccessList')) /* access list might be undefined */
      d += ', access list ' + response[i].AccessList;
    d += '\n';
  }
  log(d);
} catch (e) {
  log('error', 'error on GPA: ' + e.message);
}

try {
  let names = [];
  names[0] = {};
  names[0].Name = 'InternetGatewayDevice.ManagementServer.PeriodicInformInterval';
  names[0].NotificationChange = true;
  names[0].Notification = 1;
  names[0].AccessListChange = true;
  names[0].AccessList = [ 'subscriber' ];

  cpe.SetParameterAttributes(names);
  log('SPA done');
} catch (e) {
  log('error', 'error on SPA: ' + e.message);
}
try {
  let dlcmd = {};
  dlcmd.CommandKey = 'acsd-js-dl';
  dlcmd.FileType = '3 Vendor Configuration File';
  dlcmd.URL = 'http://eltex.loc/acsd-ntp.conf';
  dlcmd.Username = 'testuser';
  dlcmd.Password = 'testpass';
  dlcmd.FileSize = 1334;
  dlcmd.TargetFileName = 'shit';
  dlcmd.DelaySeconds = 2;
  dlcmd.SuccessURL = 'http://ya.ru';
  .FailureURL = 'http://yandex.ru';
  let response;

  log('BOOT flag ' + SESSION_BOOT + ', Transfer Complete flag ' +
SESSION_TRANSFER_COMPLETE);
```

```
  if (!SESSION_TRANSFER_COMPLETE || SESSION_BOOT) {
    response = cpe.Download(dlcmd);
    log('Download response: Status ' + response.Status + ', StartTime ' +
response.StartTime + '; CompleteTime ' + response.CompleteTime);
  }

  if (SESSION_TRANSFER_COMPLETE && cpe.Fault.Code != 0) {
    let tc_error = 'error on transfer complete: code ' + cpe.Fault.Code;

    if (cpe.Fault.String.length)
      tc_error += ': ' + cpe.Fault.String;

    log('error', tc_error);
  }
} catch (e) {
  log('error', 'error on Download: ' + e.message);
}
exit();
```

19 ACS SERVER CONFIGURATION FOR ELTEX CPE

Server logon

```
acs@acs-desktop:~$ acs-cli

****************************************
*     CLISH (see-lish)         *
*                    *
*   WARNING: Authorised Access Only   *
****************************************

Enter username (Esc - Cancel): admin
Enter password: *****

Welcome, it is Wed Feb 8 16:06:58 NOVT 2012
 (acs)
```

NTE-RG configuration

Configure **NTE-RG-1402G** with the serial number **02002B015390** in order to start PPP session with login **ppp_login**, password **ppp_pass**, and register a single FXS port on SIP server **voice.ru** with the phone number **200000**, password **sip_password**.

```
(acs)class NTE1400
NTE1400(acs)add profile Test_profile
NTE1400(acs)profile Test_profile
NTE1400(acs-profile-'Test_profile')
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.enablesip"            "1"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.useproxy"             "1"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.proxyip"          "voice.ru"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.outbound"             "1"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.dial_timeout"         "4"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.registration"         "1"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.registrarip"      "voice.ru"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.rri"                 "30"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.expires"             "600"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.authentication"       "1"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.ringback"             "1"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.rb_timeout"           "60"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.domain"           "voice.ru"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.hangup_timeout"       "30"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.busy_timeout"         "30"
 set property    "InternetGatewayDevice.X_ELTEX_Config.pbx.sip.domain_to_reg"        "1"
NTE1400(acs-profile-'Test_profile')exit
NTE1400(acs) cpe 02002B015390
NTE1400(acs-cpe-'02002B015390') set profile Test_profile
NTE1400(acs-cpe-'02002B015390')
 set property   "InternetGatewayDevice.X_ELTEX_Config.pbx.fxs1.phone"            "200000"
 set property   "InternetGatewayDevice.X_ELTEX_Config.pbx.fxs1.username"         "200000"
 set property   "InternetGatewayDevice.X_ELTEX_Config.pbx.fxs1.auth_name"        "200000"
 set property   "InternetGatewayDevice.X_ELTEX_Config.pbx.fxs1.auth_pass"        "sip_password"
 set property   "InternetGatewayDevice.X_ELTEX_Config.network.vlanW.1.username"  " ppp_login "
 set property   "InternetGatewayDevice.X_ELTEX_Config.network.vlanW.1.password"  " ppp_pass "
NTE1400(acs-cpe-'02002B015390') commit
```

NTE-RG rev.B Configuration

Configure **NTE-RG-1402G-W rev.B** with the serial number **020056000289** in order to start PPP session with login **ppp_login**, password **ppp_pass**, and register a single FXS port on SIP server **voice.ru** with the phone number **200000**, password **sip_password** and allow the user to modify PPP and SIP authorization settings.

```
(acs)class NTE14REVB
NTE14REVB(acs)add profile Test_profile
NTE14REVB(acs)profile Test_profile
NTE14REVB(acs-profile-'Test_profile')
set                                                                                    "Enabled
property  "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.1.Enable"     "
set       "InternetGatewayDevice.WANDevice.6.WANConnectionDevice.1.WANPPPConnection.1.
property  Enable"                                                                        "1"
set       "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.SIP.OutboundPr  "voice.
property  oxy"                                                                           ru"
set       "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.SIP.RegistrarS  "voice.
property  erver"                                                                         ru"
set       "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.SIP.ProxyServe  "voice.
property  r"                                                                             ru"
NTE14REVB(acs-profile-'Test_profile')exit
NTE14REVB(acs) cpe 020056000289
NTE14REVB(acs-cpe-020056000289)set profile Test_profile
NTE14REVB(acs-cpe-020056000289)
 set          "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.        "200000"
 property     Line.1.CallingFeatures.CallerIDName"
 set          "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.        "200000"
 property     Line.1.DirectoryNumber"
 set          "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.        "200000"
 property     Line.1.SIP.AuthUserName"
 set          "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.        "200000"
 property     Line.1.SIP.URI"
 set          "InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.     "sip_password"
 property     Line.1.SIP.AuthPassword"
 set          "InternetGatewayDevice.WANDevice.6.WANConnectionDevice.1.WANPP      "ppp_login"
 property     PConnection.1.Username"
 set          "InternetGatewayDevice.WANDevice.6.WANConnectionDevice.1.WANPP      "ppp_pass"
 property     PConnection.1.Password"
 set          "InternetGatewayDevice.WANDevice.6.WANConnectionDevice.1.WANPP          "1"
 property     PConnection.1.X_ELTEX_COM_UserDefinedAuthData"
 set          "InternetGatewayDevice.Services.VoiceService.1.X_ELTEX_COM_Use          "1"
 property     rDefinedData"
NTE14REVB(acs-cpe-020056000289) commit
```

RG-14XX configuration for port registration on SIP server

Configure **RG-14XX** with the serial number **VI0E000050** in order to obtain an address on WAN port via DHCP and register a single FXS port on SIP server voice.ru with the phone number **10000**, login= **10000**, password= **sip_password**.

```
(acs)class RG
RG(acs)add profile Test_profile
RG(acs)profile Test_profile
RG(acs-profile-'Test_profile')
set property      "InternetGatewayDevice.Services.VoiceService.1.VoiceP        "voice.ru"
                  rofile.1.SIP.ProxyServer"
set property      "InternetGatewayDevice.Services.VoiceService.1.VoiceP        "voice.ru"
                  rofile.1.SIP.RegistrarServer"
set property      "InternetGatewayDevice.Services.VoiceService.1.VoiceP        "voice.ru"
                  rofile.1.SIP.UserAgentDomain"
set property      "InternetGatewayDevice.Services.VoiceService.1.VoiceP           "1"
                  rofile.1.SIP.X_UseUserAgentDomainForRegister"
set property      "InternetGatewayDevice.Services.VoiceService.1.VoiceP         "1800"
                  rofile.1.SIP.RegisterExpires"
set property      "InternetGatewayDevice.Services.VoiceService.1.VoiceP           "1"
```

```
                  rofile.1.SIP.X_OutboundMode"
RG(acs-profile-'Test_profile')exit
RG(acs)cpe VI0E000050
RG(acs-cpe-'VI0E000050') set profile Test_profile
RG(acs-cpe-'VI0E000050')
set property    "InternetGatewayDevice.Services.VoiceService.1.VoiceP            "10000"
                rofile.1.Line.1.DirectoryNumber"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoiceP            "10000"
                rofile.1.Line.1.SIP.AuthUserName"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoiceP        "sip_password"
                rofile.1.Line.1.SIP.AuthPassword"
RG(acs-cpe-'VI0E000050') commit
```

Configure RG-14XX to perform inbound and outbound calls

Configure **RG-14XX** with the serial number **VI0E000050** in order to obtain an address on WAN port via DHCP and receive calls according to the routing plan between two ports with phone numbers **10000** and **10001**, and to perform calls to external subscribers via address **192.168.0.5**.

```
(acs)class RG
RG(acs)add profile Test_profile
RG(acs)profile Test_profile
RG(acs-profile-'Test_profile')
set property     "InternetGatewayDevice.Services.VoiceService.1              "S5, L30
                 .VoiceProfile.1.DigitMap"                        (1000x@{local}|x.@192.168.0.5)"
RG(acs-profile-'Test_profile')exit
RG(acs)cpe VI0E000050
RG(acs-cpe-'VI0E000050') set profile Test_profile
RG(acs-cpe-'VI0E000050')
set property     "InternetGatewayDevice.Services.VoiceService.1.Voi             "10000"
                 ceProfile.1.Line.1.DirectoryNumber"
set property     "InternetGatewayDevice.Services.VoiceService.1.Voi             "10001"
                 ceProfile.1.Line.2.DirectoryNumber"
RG(acs-cpe-'VI0E000050') commit
```

TAU-8.IP configuration for port registration on SIP server

Configure **TAU-8.IP** with the serial number **VI09000141** in order to obtain an address on WAN port via DHCP and register a single FXS port on SIP server **ngn–sip.sinor.ru** with the phone number **10000**, login= **10000**, password= **sip_password**.

```
(acs)class TAU
TAU(acs)add profile Test_profile
TAU(acs)profile Test_profile
TAU(acs-profile-'Test_profile')
set property    "InternetGatewayDevice.WANDevice.1.WANConnectionDevice.           "DHCP"
                1.WANIPConnection.1.AddressingType"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoicePro         "voice.ru"
                file.1.SIP.ProxyServer"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoicePro         "voice.ru"
                file.1.SIP.RegistrarServer"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoicePro         "voice.ru"
                file.1.SIP.UserAgentDomain"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoicePro            "1"
                file.1.SIP.X_UseUserAgentDomainForRegister"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoicePro          "1800"
                file.1.SIP.RegisterExpires"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoicePro            "1"
                file.1.SIP.X_OutboundMode"
TAU(acs-profile-'Test_profile')exit
TAU(acs)cpe VI09000141
TAU(acs-cpe-'VI0E9000141') set profile Test_profile
TAU(acs-cpe-'VI09000141')
set property    "InternetGatewayDevice.Services.VoiceService.1.VoiceP            "10000"
```

```
                rofile.1.Line.1.DirectoryNumber"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoiceP          "10000"
                rofile.1.Line.1.SIP.AuthUserName"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoiceP       "sip_password"
                rofile.1.Line.1.SIP.AuthPassword"
TAU(acs-cpe-'VI09000141') commit
```

Configure TAU-8.IP to perform inbound and outbound calls

Configure **TAU-8.IP** with the serial number **VI09000141** in order to obtain an address on WAN port via DHCP and receive calls according to the **dialplan** between two ports with phone numbers **10000** and **10001**, and to perform calls to external subscribers via address **192.168.0.5**.

```
(acs)class TAU
TAU(acs)add profile Test_profile
TAU(acs)profile Test_profile
TAU(acs-profile-'Test_profile')

set property    "InternetGatewayDevice.WANDevice.1.WANConnectionDev            "DHCP"
                ice.1.WANIPConnection.1.AddressingType"


set property    "InternetGatewayDevice.Services.VoiceService.1.Voic          "S5, L30
                eProfile.1.DigitMap"                                  (1000x@{local}|x.@192.168.
                                                                            0.5)"

TAU(acs-profile-'Test_profile')exit
TAU(acs)cpe VI09000141
TAU(acs-cpe-'VI09000141') set profile Test_profile
TAU(acs-cpe-'VI09000141')
set property    "InternetGatewayDevice.Services.VoiceService.1.VoiceP          "10000"
                rofile.1.Line.1.DirectoryNumber"
set property    "InternetGatewayDevice.Services.VoiceService.1.VoiceP          "10001"
                rofile.1.Line.2.DirectoryNumber"
TAU(acs-cpe-'VI09000141') commit
```

TAU-1.IP configuration for port registration on SIP server

Configure **TAU-1.IP** with the serial number **VI01000231** in order to obtain an address on WAN port via DHCP and register a single FXS port on SIP server **ngn–sip.sinor.ru** with the phone number **10000**, login= **10000**, password= **sip_password**.

```
(acs)class TAU
TAU(acs)add profile Test_profile
TAU(acs)profile Test_profile
TAU(acs-profile-'Test_profile')

set property        "InternetGatewayDevice.VoiceService.1.VoiceProfile.1.S    "voice.ru"
                    IP.ProxyServer"
set property        "InternetGatewayDevice.VoiceService.1.VoiceProfile.1.S    "voice.ru"
                    IP.RegistrarServer"
set property        "InternetGatewayDevice.X_ELTEXInfo.SIPOptions.SIPDomai    "voice.ru"
                    n"
set property        "InternetGatewayDevice.X_ELTEXInfo.SIPOptions.UseDomai        "1"
                    nToRegister"
set property        "InternetGatewayDevice.X_ELTEXInfo.SIPOptions.Outbound        "1"
                    Enable"
TAU(acs-profile-'Test_profile')exit
TAU(acs)cpe VI01000231
TAU(acs-cpe-'VI01000231') set profile c
TAU(acs-cpe-'VI01000231')
set property       "InternetGatewayDevice.X_ELTEXInfo.LineConfig.PhoneNu      "10000"
                   mber"
```

Create firmware and assign it to CPE profile

```
NTE1400(acs) copy file firmware 192.168.15.158 nte_rg_14xx_5.2.0.tgz
Copying file from host '192.168.15.158', remote path 'nte_rg_14xx_5.2.0.tgz' done.
NTE1400(acs)add firmware 1
Ok
NTE1400(acs)firmware 1
NTE1400(acs-firmware-'1') set file firmware nte_rg_14xx_5.2.0.tgz
Set version '5.2.0': ok.
Set filename 'nte_rg_14xx_5.2.0.tgz': ok.
Add hardware 'NTE-RG-1400F': Ok.
Add hardware 'NTE-RG-1400F-W': Ok.
Add hardware 'NTE-RG-1400FC': Ok.
Add hardware 'NTE-RG-1400FC-W': Ok.
Add hardware 'NTE-RG-1400G': Ok.
Add hardware 'NTE-RG-1400G-W': Ok.
Add hardware 'NTE-RG-1400GC': Ok.
Add hardware 'NTE-RG-1400GC-W': Ok.
Add hardware 'NTE-RG-1402F': Ok.
Add hardware 'NTE-RG-1402F-W': Ok.
Add hardware 'NTE-RG-1402FC': Ok.
Add hardware 'NTE-RG-1402FC-W': Ok.
Add hardware 'NTE-RG-1402G': Ok.
Add hardware 'NTE-RG-1402G-W': Ok.
Add hardware 'NTE-RG-1402GC': Ok.
Add hardware 'NTE-RG-1402GC-W': Ok.
NTE1400(acs-firmware-'1')add link profile NTE-RG1402
Ok
NTE1400(acs-firmware-'1')set flag safe_upgrade enable // Enable loyalty mode. Firmware will be updated only when BOOT Inform is received
Ok
NTE1400(acs-firmware-'1')show config
Information about firmware:

               Index = 1
          Version = '5.2.0'
          URL = ''
          Filename = 'nte_rg_14xx_5.2.0.tgz'
          Schedule = enabled
          Safe-mode = enabled
         Downgrade= disabled

Profiles for firmware 1

 ##  Name  Inform interval Day skeep stats Base profile  Description
 1: NTE-RG1402   3600              0
No CPE with firmware 1!
NTE1400(acs-firmware-'1')exit
```

Configure CPE firmware update schedule

```
NTE1400(acs)schedule
NTE1400(acs-schedule)set daily 00:00 05:00 //from 00:00 to 05:00 (from the midnight to 5am).
Ok
NTE1400(acs-schedule)set weekly 1 5  // from Monday to Friday
Ok
NTE1400(acs-schedule)show

 NTE1400(acs-schedule)show

  1: daily   00:00   05:00
  2: period   2011-06-01 2099-12-31
  3: weekly   1     5
NTE1400(acs-schedule)commit
Transaction was commited.
NTE1400(acs-schedule)
NTE1400(acs-schedule)exit
NTE1400(acs)
```

20 FREQUENTLY ASKED QUESTIONS

What is the device class (NTE1400, NTP, etc.) on the ACS server?

Class is the group of devices, that conform to the following conditions: *OUI* (manufaturer's identifier) and *ProductClass*(device model).

OUI and *ProductClass* parameters are present in each CPE session with the server. If the *OUI+ ProductClass* link sent from CPE is not found in the server configuration, the CPE will appear under the UNKNOWN class.

I already have a device manufactured by XXX company. How can I establish connection to a server?

You should configure **acs_url** address http://<url/ip>:<port>, and also the server login and password on the CPE (factory settings: acs/acsacs, port 9595).

When a new device appears under the UNKNOWN class, you may proceed to its configuration.

```
  ##  ID  Serial              Profile      Product class Firmware   Last contact         Edit by

  1: 53  020222111015394  0: Default..   265          1.2.5      2013-02-13 12:52:23 auto
```

To create a separate class for a new device, you should create a new class in the 'advanced' section using the following command:
`(acs-advanced) add class <class_name>`
and move there a new device model from the UNKNOWN class
`(acs-advanced-class-'UNKNOWN')move hardware <OUI>< Product class > <class_name>.`

You can view OUI in the journal using the following command: `UNKNOWN(acs-journal-full mode)show journal all:`

```
  ##   ID        SerialNumber     Events Manufacturer  OUI      CurrentTime
Softwar..    ConnectionRequestURL

  1: 2149248463   00265A9487C  1BOOT  Dlink         00265A   2012-12-06 17:30:09
0.02.72.1.1   http://10.255.240.82:9998
  2: 2149248465   00265A9237R  1BOOT  Dlink         00265A   2012-12-06 17:30:14
0.02.72.1.1   http://10.255.240.86:9998
  3: 2149248466   00265A9230C  1BOOT  Huawei       123456   2012-12-06 17:30:23
123.123.123  http://11.255.240.87:9998
```

For detailed configuration of third-party devices, see the **'Operation Manual Appendix *ACS Server Configuration Manual. Operations with CPE*'**.

What is the OUI mapping? Why do I need it, and what does it do?

OUI—manufacturer ID.

A manufacturer can have multiple OUIs (it is the upper part of the device MAC address). To simplify the server operation logic, all manufacturer OUIs are mapped into a single root OUI. In future, the root OUI will be used for class operations.

```
(acs-advanced)add oui 00265A 00265A
Ok
(acs-advanced)add oui 00265B 00265A
Ok
(acs-advanced)add oui 00265C 00265A
Ok
```

```
(acs-advanced)add oui 00265D 00265A
Ok
(acs-advanced)show oui mapped 00265A
List for mapped OUI "00265A":
  1: "00265A"
  2: "00265B"
  3: "00265C"
  4: "00265D"
```

In which modes the server can process passwords sent to the CPE (e.g. sip/ppp passwords)?

There are three password processing modes:

- **UNSAFE** mode:

  Passwords are unconditionally sent in each PERIODIC session.
  Enter the mode:

```
(acs-settings)set password_mode unsafe
```

- **AUTO** mode:

  Passwords are sent in BOOT session only (BOOT session occurs during the device startup).
  Enter the mode:

```
(acs-settings)set password_mode auto
```

- **SAFE** mode:

  Passwords are sent only when operator executes the 'direct setpass' command (e.g. **NTE1400(acs-cpe-'020022000210')direct setpass**
  Enter the mode:

```
(acs-settings)set password_mode safe
```

I cannot figure out, where can I configure logins and passwords and what is their purpose?

In the current CLI, there are several sections, where you can configure different logins and passwords for server's operations with the CPE.

1. CPE authorization on ACS server

In each session, CPE passes authorization while connecting to the ACS server.

One of the following login/password links is used for CPE authorization on the server:

a) EITHER links that are defined for the specific server eth interface:

```
(acs-settings) set authorize user <Username> <Password> [Description] [Interface])
```

In this case, the specified login/password link is a general link for all CPEs, that operate with the server via this interface. By default, acs/acsacs link is used for all interfaces.

b) OR individual links for each CPE:

```
(acs-cpe-'<serial>')set username <Username>
(acs-cpe-'<serial>')set password <Password>
```

If individual login and password are specified, CPE may authorize on the server either using this link or using link from the item a).

2. ACS server authorization on CPE

When server commands are executed on the CPE (direct commands), the server passes authorization while connecting to the CPE.

In this case, the server will use the following passwords for authorization:

```
<hardware class>(acs-cpe-<serial>)show config
Information about CPE -<serial>):
…..
   Con req user = "admin"
   Con req pass = "admin"
……
```

In the current implementation, these passwords cannot be changed; default values: login — **admin**, password — **admin**.

During the first CPE connection to the server (BOOT session), the server establishes registration link on CPE (using 'admin/admin' values) and works smoothly with the device from then on. It does not matter, what *'Con req'* login and password are configured on CPE, the server will gain access to it after the first session with the CPE server.

3. Authorization of subscriber services

There is another login/password category—subscriber logins/passwords. These links are specified to enable the operator services (sip, ppp, etc.) on CPE, and may be specified either with long parameters or short parameters (if available).

```
There is another login/password category—subscriber logins/passwords. These links are
specified to enable the operator services (sip, ppp, etc.) on CPE, and may be specified
either with long parameters or short parameters (if available).
Ok
NTE1400(acs-cpe-'020022000210')set property
"InternetGatewayDevice.WANDevice.5.WANConnectionDevice.1.WANPPPConnection.1.Password"
"1AS45deR"
Ok
NTE1400(acs-cpe-'020022000210')set private ppp_password 1234
CPE property has been updated: 'InternetGatewayDevice.X_ELTEX_Service.Internet.password' =
'1234'.
Ok
NTE1400(acs-cpe-'020022000210')set private voice1_password 1wd2eedd
CPE property has been updated: 'InternetGatewayDevice.X_ELTEX_Config.pbx.fxs1.auth_pass' =
'1wd2eedd'.
Ok
```

Which classes and OUI/ProductClass are supported by the server during the installation?

In the current release (1.6.2), the following classes are registered during the server installation and mysql database creation with the **create_acs_bases.sh** script.

Class	OUI	ProductClass	
NTP	**A8F94B**	NTP-2; NTP-2C; NTP-RG-1400G; NTP-RG-1400G-W; NTP-RG-1400G-W2; NTP-RG-1402G; NTP-RG-1402G-W; NTP-RG-1402G-W2; NTP-RG-1400GC; NTP-RG-1400GC-W; NTP-RG-1400GC-W2; NTP-RG-1402GC;	NTP-RG-1402GB-W; NTP-RG-1402GB-W2; NTP-RG-1402GCB; NTP-RG-1402GCB-W; NTP-RG-1402GCB-W2; NTP-RG-1402G-W rev.C NTP-RG-1402GC-W rev.B NTP-RG-1402G-W rev.B

		NTP-RG-1402GC-W; NTP-RG-1402GC-W2; NTP-RG-1402GB;		
NTE1400	**A8F94B**	NTE-RG-1402F; NTE-RG-1402G; NTE-RG-1402F-W; NTE-RG-1402G-W; NTE-RG-1402FC; NTE-RG-1402GC;	NTE-RG-1402FC-W; NTE-RG-1402GC-W; NTE-RG-1400F; NTE-RG-1400G; NTE-RG-1400F-W;	NTE-RG-1400G-W; NTE-RG-1400FC; NTE-RG-1400GC; NTE-RG-1400FC-W; NTE-RG-1400GC-W
NTE1400REVB	**A8F94B**	NTE-RG-1402G-W rev.C NTE-RG-1402GC-W rev.B NTE-RG-1402G-W rev.B		
RG	**A8F94B**	RG-1402G RG-1402G-W RG-1402GF RG-1402GF-W RG-1404G RG-1404F-W RG-1404G-W RG-1404GF-W	RG-1404GF RG-2404G-W RG-2404F-W RG-2404G RG-2402G RG-2402G-W RG-2402GF RG-4402G-W	
TC	**A8F94B**	TC-10 TC-10-W	TC-20 TC-20-W	
TAU	**A8F94B**	TAU-104.IP; TAU-104F.IP;	TAU-1E.IP; TAU-1EP.IP;	TAU-8.IP; TAU-8.IP-W TAU-32M.IP TAU-32M_revB TAU-36.IP TAU-72.IP TAU-1M.IP TAU-2M.IP
UNKNOWN		All unknown device types are automatically enlisted in this class.		

APPENDIX A. DISTRIBUTION OF CLI ACS OPERATIONS BY ACCESS FLAGS

1. 'General' commands and rules

Commands and rules, that are present in all section, where they fit.

Command	Access	Description
logout	always available	change user
exit	always available	return to the previous level
top	always available	return to root
commit	all editing rights	transaction confirmation
rollback	all editing rights	transaction rollback
—	cli-view-pass	show passwords

2. Root section commands

Command	Access	Description
class <class name>	access any internal function of a section	hardware class configuration section
log	access any internal function of a section	root journal (journal records, not linked to any class)
settings	access any internal function of a section	settings
configuration	access any internal function of a section	operations with configuration
advanced	access any internal function of a section	data model and hardware class model configuration section
users	access any internal function of a section	user and role management section
version	always available	show firmware version
statistics	always available	show ACS statistics
sysinfo	always available	show system and mysql data
find cpe by ip/serial	cpe-view-config + journal-view-root + journal-view-class	search information on CPE

3. CLASS section commands

Command	Access	Description
cpe <serial>	access any internal function of a section	enter CPE operations section
add cpe <serial> <OUI> <product class>	cpe-edit-create	add CPE
delete cpe <serial>	cpe-edit-create	remove CPE
move cpe config <src serial> <dst serial>	cpe-edit-create + cpe-edit-config + cpe-edit-subscriber + cpe-edit-profile + cpe-edit-property + cpe-edit-service + cpe-edit-personal-firmware + cpe-edit-personal-options + groups-edit-static	move configuration to another CPE
show cpe mode	cli-manage-showmode	show viewing settings
set cpe mode <field index> <mode> <length>	cli-manage-showmode	edit viewing settings
show cpe config <serial>	cpe-view-config	show CPE configuration
show cpe full <serial>	cpe-view-config + cpe-view-property	show full CPE information
show cpe all	cpe-view-config	show full CPE list
show cpe sort <field name>	cpe-view-config	show sorted CPE list
show cpe include <search value>	cpe-view-config	show filtered CPE list
show cpe not-provisioned	cpe-view-config	show CPEs with unassigned properties
show cpe updated <editor name>	cpe-view-config	show CPE filtered by source of last changes
show cpe service assigned <serial>	cpe-view-config	show list of assigned services
show cpe property <serial>	cpe-view-property	show list of all properties for CPE
profile <name>	access any internal function of a section	enter the profile operations section
add profile <name>	profile-edit-create	add profile
delete profile <name>	profile-edit-create	remove profile
show profile mode	cli-manage-showmode	show viewing settings
set profile mode <field index> <mode> <length>	cli-manage-showmode	edit viewing settings
show profile config <name>	profile-view-config	show profile configuration
show profile all	profile-view-config	show all profiles
show profile struct	profile-view-config	show all profiles in hierarchical structure

show profile cpe using <name>	cpe-view-config	show list of CPEs that use this profile
show profile property <name>	profile-view-property	show list of all properties for profile
hardware <OUI> <product class>	access any internal function of a section	enter the hardware model operations section
add hardware <OUI> <product class> [<manufacrturer>]	hardware-edit-create	add hardware model
delete hardware <OUI> <product class>	hardware-edit-create	remove hardware model
show hardware mode	cli-manage-showmode	show viewing settings
set hardware mode <field index> <mode> <length>	cli-manage-showmode	edit viewing settings
show hardware config <OUI> <product class>	hardware-view-config	show hardware model configuration
show hardware all	hardware-view-config	show list of all hardware models
firmware <name>	access any internal function of a section	firmware configuration section
add firmware <name>	firmware-edit-create	add firmware
delete firmware <name>	firmware-edit-create	remove firmware
show firmware mode	cli-manage-showmode	show viewing settings
set firmware mode <field index> <mode> <length>	cli-manage-showmode	edit viewing settings
show firmware config <name>	firmware-view-config	show firmware configuration
show firmware all	firmware-view-config	show list of all firmware
show firmware link profile <name>	firmware-view-links	show list of links with profiles
show firmware link hardware <name>	firmware-view-links	show list of links with hardware models
show file firmware all	file-view-firmware	show firmware file list
copy file firmware <ip> <filename>	file-copy-firmware	copy firmware file
delete file firmware <filename or mask>	file-delete-firmware	remove firmware file(s)
schedule	firmware-manage-schedule	enter firmware update schedule configuration subsection
service <name>	access any internal function of a section	service configuration section
add service <name>	service-edit-create	add service
delete service <name>	service-edit-create	remove service
show service mode	cli-manage-showmode	show viewing settings

set service mode <field index> <mode> <length>	cli-manage-showmode	edit viewing settings
show service config <name>	service-view-config	show service configuration
show service all	service-view-config	show list of all services
show service property <name>	service-view-property	show service property
group <name>	access any internal function of a section	CPE group configuration section
add group dynamic <name>	groups-edit-dynamic	add dynamic group
add group static <name>	groups-edit-static	add static group
delete group <name>	groups-edit-dynamic, groups-edit-static (depending on group type)	remove group
show group all	groups-view-config	show list of all groups
show group struct	groups-view-config	show group structure
show group cpe <name>	groups-view-cpe	show CPE list in a group
show private all	privates-view	show list of all private parameters
show private param <param name>	privates-view	show correspondence of private properties to specific private parameter
add private <param name> <private name> [nocheck]	privates-edit	add new private property value of the private parameter
delete private index <param name> <index>	privates-edit	remove private property value of the private parameter by index
delete private name <private name>	privates-edit	remove private property
delete private param <param name>	privates-edit	remove private parameter
journal informs	access any internal function of a section	go to journal (inform section)
journal unauthorized	access any internal function of a section	go to journal (unauthorized CPE section)
journal errors	access any internal function of a section	go to journal (error message section)
journal full	access any internal function of a section	go to journal (full view mode)
show common settings	class-view-common	show general class settings
set common flag boot_load_tree <value>	class-edit-flags	edit class flag
set common script <value>	class-edit-flags	specify class script
delete common flag boot_load_tree	class-edit-flags	edit class flag

delete common script	class-edit-flags	remove specified script from class
show file script all	file-view-script	show script list

3.1.CPE <serial> subsection command

Command	Access	Description
show config	cpe-view-config	show CPE configuration
show full	cpe-view-config + cpe-view-property	show full CPE information
show service assigned	cpe-view-config	show list of assigned services
set configname <value>	cpe-edit-config	edit CPE
set customer <value>	cpe-edit-config	edit CPE
set profile <value>	cpe-edit-profile	edit CPE
set username <value>	cpe-edit-authorization	edit CPE
set password <value>	cpe-edit-authorization	edit CPE
set subscriber <subscriber id>	cpe-edit-subscriber	edit CPE
delete subscriber	cpe-edit-subscriber	edit CPE
show property	cpe-view-property	show CPE property
set property <property name> <property value> [nocheck]	cpe-edit-property	assign property
unset property <property name> [nocheck]	cpe-edit-property	assign property removal in CPE
delete property <property name>	cpe-edit-property	remove property information
clear property	cpe-edit-property	clear all property information
set private <name> <value>	cpe-edit-property	assign short property
unset private <name> <value>	cpe-edit-property	assign short property removal in CPE
delete private <name>	cpe-edit-property	remove short property information
set service <service> <instance>	cpe-edit-service	assign service
delete service <service> <instance>	cpe-edit-service	unlink service
clear service	cpe-edit-service	unlink all services
set personal firmware <name>	cpe-edit-personal-firmware	assign personal firmware
delete personal firmware	cpe-edit-personal-firmware	remove personal firmware
set flag safe_upgrade <mode>	cpe-edit-personal-options	set safe_upgrade flag
delete flag safe_upgrade	cpe-edit-personal-options	remove safe_upgrade flag
set flag downgrade <value>	cpe-edit-personal-options	set downgrade flag
delete flag downgrade	cpe-edit-personal-options	remove downgrade flag
direct reboot	cpe-command-reboot	send command
direct setfactdef	cpe-command-setfactdef	send command

direct setpass	cpe-command-setpass	send command
direct reconfigure	cpe-command-reconfigure	send command
direct sync	cpe-command-sync	send command
direct upgrade	cpe-command-upgrade	send command
direct set parameter value <name> <value> [nocheck]	cpe-command-set-parameter	send command
direct set parameter attribute <name> <attribute name> <attribute value> [nocheck]	cpe-command-set-parameter	send command
direct get parameter names <name> [<nextlevel>] [nocheck]	cpe-command-get-parameter	send command
direct get parameter value <name> [nocheck]	cpe-command-get-parameter	send command
direct get parameter attribute <name> [nocheck]	cpe-command-get-parameter	send command
direct get state services	cpe-command-get-state-services	send command
direct get rpc methods	cpe-command-get-rpc-methods	send command
direct download config <filename>	cpe-command-download	send command
direct download firmware <filename>	cpe-command-download	send command
direct download vendorspecific <filetype> <filename>	cpe-command-download	send command
direct ipping <...>	cpe-command-ipping	send ip ping diagnostic command
show result download	cpe-command-download	show result
show result ipping	cpe-command-ipping	show result
show file firmware all	file-view-firmware	show firmware file list
show file config all	file-view-config	show configuration file list
batch	cpe-command-set-parameter, cpe-command-get-parameter	enter the batch mode

3.1.1. BATCH subsection commands

Command	Access	Description
add set <name> <value> [nocheck]	cpe-command-set-parameter	add command to set queue
delete set name <name>	cpe-command-set-parameter	remove command from set queue by its name
delete set index <index>	cpe-command-set-parameter	remove command from set queue by its index
add get <name> <value>	cpe-command-get-	add command to get queue

[nocheck]	parameter	
delete get name <name>	cpe-command-get-parameter	remove command from get queue by its name
delete get index <index>	cpe-command-get-parameter	remove command from get queue by its index
title batch set <title>	cpe-command-set-parameter	assign set queue header
title batch get <title>	cpe-command-get-parameter	assign get queue header
show batch set	cpe-command-set-parameter	show set queue commands
show batch get	cpe-command-get-parameter	show get queue commands
clear batch set	cpe-command-set-parameter	clear set queue commands
clear batch get	cpe-command-get-parameter	clear get queue commands
reset batch set	cpe-command-set-parameter	clear set queue commands + remove header
reset batch get	cpe-command-get-parameter	clear get queue commands + remove header
send batch set	cpe-command-set-parameter	remove set queue header
send batch get	cpe-command-get-parameter	remove get queue header
show result <result id> [<state>]	cpe-command-set-parameter, cpe-command-get-parameter	show batch sending result

3.2. PROFILE <name> subsection commands

Command	Access	Description
show config	profile-view-config	show profile configuration
show cpe using	cpe-view-config	show list of CPEs that use this profile
set inform_interval <value>	profile-edit-config	edit profile
set description <value>	profile-edit-config	edit profile
set base <value>	profile-edit-base	assign basic profile
delete base	profile-edit-base	assign absent basic profile
set script <scriptname>	profile-edit-config	assign script
delete script <scriptname>	profile-edit-config	remove script
show file script all	profile-edit-config	Show script list
show property	profile-edit-property	show property
set property <property name> <property value>	profile-edit-property	assign property

[nocheck]		
unset property <property name> [nocheck]	profile-edit-property	assign property removal
delete property <property name>	profile-edit-property	remove property information
clear property	profile-edit-property	clear all property information
add link firmware <firmware name>	firmware-edit-link-profile	assign firmware link
delete link firmware <firmware name>	firmware-edit-link-profile	remove firmware link
add constraint hardware <oui> <product class>	profile-edit-constraint-hardware	add hardware compatibility
delete constraint hardware value <oui> <product class>	profile-edit-constraint-hardware	remove hardware compatibility by value
delete constraint hardware index <index>	profile-edit-constraint-hardware	remove hardware compatibility by index
delete constraint hardware all	profile-edit-constraint-hardware	remove all hardware compatibilities
add constraint version <mask>	profile-edit-constraint-version	add firmware version compatibility
delete constraint version value <mask>	profile-edit-constraint-version	remove firmware version compatibility by value
delete constraint version index <index>	profile-edit-constraint-version	remove firmware version compatibility by index
delete constraint version index <index>	profile-edit-constraint-version	remove all firmware version compatibilities

3.3. HARDWARE <OUI> <Product Class> subsection commands

Command	Access	Description
show config	hardware-view-config	show hardware model configuration
set manufacturer <value>	hardware-edit-config	edit hardware model
set version <value>	hardware-edit-config	edit hardware model
set profile <value>	hardware-edit-profile	edit hardware model

3.4. FIRMWARE <name> subsection commands

Command	Access	Description
show config	firmware-view-config	show firmware configuration
show link profile	firmware-view-links	show profile links
show link hardware	firmware-view-links	show hardware model links
set version <value>	firmware-edit-config	configure firmware
set url <value>	firmware-edit-config	configure firmware
set filename <value>	firmware-edit-config	configure firmware
set schedule <value>	firmware-edit-schedule	configure firmware

set file firmware <filename>	file-parse-firmware + firmware-edit-config + firmware-edit-link-hardware	configure firmware using firmware header data
add link profile <profile name>	firmware-edit-link-profile	add profile link
add link all profile	firmware-edit-link-profile	add links to all profiles
add link hardware <OUI> <product class>	firmware-edit-link-hardware	add hardware model link
delete link profile	firmware-edit-link-profile	remove link from profile
delete link hardware	firmware-edit-link-hardware	remove link from hardware model
set personal cpe <serial>	cpe-edit-personal-firmware	assign this firmware to CPE as personal
delete personal cpe <serial>	cpe-edit-personal-firmware	remove this firmware from CPE as personal
set flag safe_upgrade <value>	firmware-edit-flags	set flag
set flag downgrade <value>	firmware-edit-flags	set flag

3.5. SCHEDULE subsection commands

Command	Access	Description
show config	firmware-manage-schedule	show schedule configuration
set daily <time from> <time to>	firmware-manage-schedule	define value
set weekly <day from> <day to>	firmware-manage-schedule	define value
set period <date from> <date to>	firmware-manage-schedule	define value
set default daily	firmware-manage-schedule	define default value
set default weekly	firmware-manage-schedule	define default value
set default period	firmware-manage-schedule	define default value

3.6. SERVICE <name> subsection commands

Command	Access	Description
show config	service-view-config	show service configuration
set description <value>	service-edit-config	edit description
show property	service-edit-property	show property list
set property <property name> <property value> [nocheck]	service-edit-property	assign property
unset property <property name> [nocheck]	service-edit-property	assign property removal
delete property <property name>	service-edit-property	remove property information
clear property	service-edit-property	clear all property information

3.7. GROUP <name> subsection commands

Command	Access	Description
show filter	groups-view-config	show filter conditions list
show cpe	groups-view-cpe	show CPE list in a group
add cpe <serial>	groups-edit-static	add CPE to static group list
delete cpe <serial>	groups-edit-static	remove CPE from static group list
clear cpe	groups-edit-static	clear static group list
generate cpe by filter [add/replace]	groups-edit-static	generate CPE static group list by filter conditions
add filter wildcard <field> <mask>	groups-edit-dynamic, groups-edit-static	add filter condition
add filter datetime interval <field> <date from> <date to>	groups-edit-dynamic, groups-edit-static	add filter condition
add filter datetime from <field> <date from>	groups-edit-dynamic, groups-edit-static	add filter condition
add filter datetime until <field> <date to>	groups-edit-dynamic, groups-edit-static	add filter condition
add filter datetime older <field> <minutes>	groups-edit-dynamic, groups-edit-static	add filter condition
add filter datetime under <field> <minutes>	groups-edit-dynamic, groups-edit-static	add filter condition
add filter value editor <value>	groups-edit-dynamic, groups-edit-static	add filter condition
add filter expression <expression>	groups-edit-dynamic, groups-edit-static	add filter condition
delete filter index <index>	groups-edit-dynamic, groups-edit-static	remove filter condition
clear filter	groups-edit-dynamic, groups-edit-static	add all filter conditions
set parent <group>	groups-edit-dynamic, groups-edit-static	define parent group
add child <group>	groups-edit-dynamic, groups-edit-static	add child group
delete parent	groups-edit-dynamic, groups-edit-static	remove parent group
delete child <group>	groups-edit-dynamic, groups-edit-static	remove child group
edit set configname <value>	groups-operate-cpe + cpe-edit-config	group editing
edit set profile <value>	groups-operate-cpe + cpe-edit-profile	group editing
edit set property <property name> <property value> [nocheck]	groups-operate-cpe + cpe-edit-property	group editing
edit unset property <property name> [nocheck]	groups-operate-cpe + cpe-edit-property	group editing

edit delete property <property name>	groups-operate-cpe + cpe-edit-property	group editing
edit clear property	groups-operate-cpe + cpe-edit-property	group editing
edit set private <private name> <private value>	groups-operate-cpe + cpe-edit-property	group editing
edit unset private <private name> <private value>	groups-operate-cpe + cpe-edit-property	group editing
edit delete private <private name>	groups-operate-cpe + cpe-edit-property	group editing
edit set service <name> <instance>	groups-operate-cpe + cpe-edit-service	group editing
edit delete service <name> <instance>	groups-operate-cpe + cpe-edit-service	group editing
edit clear service	groups-operate-cpe + cpe-edit-service	group editing
edit set personal firmware <name>	groups-operate-cpe + cpe-edit-personal-firmware	group editing
edit delete personal firmware	groups-operate-cpe + cpe-edit-personal-firmware	group editing
edit set flag safe_upgrade <value>	groups-operate-cpe + cpe-edit-personal-options	group editing
edit delete flag safe_upgrade	groups-operate-cpe + cpe-edit-personal-options	group editing
edit set flag downgrade <value>	groups-operate-cpe + cpe-edit-personal-options	group editing
edit delete flag downgrade	groups-operate-cpe + cpe-edit-personal-options	group editing
direct reboot	groups-operate-cpe + cpe-command-reboot	send command
direct setfactdef	groups-operate-cpe + cpe-command-setfactdef	send command
direct setpass	groups-operate-cpe + cpe-command-setpass	send command
direct upgrade	groups-operate-cpe + cpe-command-upgrade	send command
direct reconfigure	groups-operate-cpe + cpe-command-reconfigure	send command
direct set parameter value <name> <value> [nocheck]	groups-operate-cpe + cpe-command-set-parameter	send command
direct set parameter attribute <name> <attribute name> <attribute value> [nocheck]	groups-operate-cpe + cpe-command-set-parameter	send command
direct get parameter names <name> [<nextlevel>] [nocheck]	groups-operate-cpe + cpe-command-get-parameter	send command
direct get parameter value <name> [nocheck]	groups-operate-cpe + cpe-command-get-parameter	send command

direct get parameter attribute <name> [nocheck]	groups-operate-cpe + cpe-command-get-parameter	send command
direct get state services	groups-operate-cpe + cpe-command-get-state-services	send command
direct download config <filename>	groups-operate-cpe + cpe-command-download	send command
direct download firmware <filename>	groups-operate-cpe + cpe-command-download	send command
direct download vendorspecific <filetype> <filename>	groups-operate-cpe + cpe-command-download	send command
show command sent	access to batch sending of any command	show list of sent commands for a group
show command result <result id> [<state>]	access to batch sending of any command	show result of command execution (filtered by state)
batch	groups-operate-cpe + (cpe-command-set-parameter, cpe-command-get-parameter)	enter the batch mode

4. LOG section commands

Command	Access	Description
show mode	cli-manage-showmode	show viewing settings
set mode <field index> <mode> <length>	cli-manage-showmode	edit viewing settings
show journal all	journal-view-root (for the root journal), journal-view-class (for the class journal)	show entire journal (within the current viewing mode)
show journal period <date from> <date to>	journal-view-root (for the root journal), journal-view-class (for the class journal)	show records within the selected period
show journal serial <serial>	journal-view-root (for the root journal), journal-view-class (for the class journal)	show records by serial number
show journal last all	journal-view-root (for the root journal), journal-view-class (for the class journal)	show last records for different serial numbers
show journal last period <date from> <date to>	journal-view-root (for the root journal), journal-view-class (for the class journal)	show last records for different serial numbers within the selected period
show journal last serial <serial>	journal-view-root (for the root journal), journal-view-class (for the class journal)	show last record for serial number
show entry <id>	journal-view-root (for the root journal), journal-view-class (for the class journal)	show record by id
view <view name>	journal-view-root (for the root journal), journal-view-class (for the class journal)	change viewing mode
clear journal period <date from> <date to>	journal-edit-root (for the root journal), journal-edit-class (for the class journal)	clear journal for the period

5. SETTINGS section commands

Command	Access	Description
show settings main	settings-view-main	show basic settings
show settings authorize	settings-view-main	show authorization settings
settings-view-main	settings-view-log	show logoption
set password_mode <value>	settings-edit-main	edit settings
set cpe_auto_create <value>	settings-edit-main	edit settings
set use_unknown <value>	settings-edit-main	edit settings
set cli_timeout <value>	settings-edit-main	edit settings
set cmd_timeout <value>	settings-edit-main	edit settings
set authorize user <login> <pass> [<descr>] [<interface>]	settings-edit-authorize	authorization settings
delete authorize user <login> [<interface>]	settings-edit-authorize	authorization settings
set authorize type <interface> <type>	settings-edit-authorize	authorization type settings
delete authorize type <interface>	settings-edit-authorize	authorization type settings
add loglevel <value>	settings-edit-log	edit settings
set loglevel <value>	settings-edit-log	edit settings
set logoption <option name> <value>	settings-edit-log	edit settings

6. CONFIGURATION section commands

Command	Access	Description
backup <file> <ip>	config-backup	backup
restore <file> <ip>	config-restore	restore
default	config-default	reset to defaults
dump script <file>	config-dump	dump

7. ADVANCED section commands

Command	Access	Description
class <name>	access any internal function of a section	go to subsection for the specific class operations
show class all	advanced-view-class-config	show class list
show class hardware <name>	advanced-view-class-config	show product class list for class
show oui all	advanced-view-oui	show all OUIs
show oui mapped <mapped oui>	advanced-view-oui	show matching OUI to the specified mapped OUI
add class <name> <manufacturer>	advanced-edit-class-create	add class
delete class <name>	advanced-edit-class-create	remove class
add oui <oui> <mapped oui>	advanced-edit-oui	add OUI
delete oui <oui>	advanced-edit-oui	remove OUI

7.1. CLASS <class name> subsection commands

Command	Access	Description
datamodel	access any internal function of a section	go to subsection for data model operations
show hardware	advanced-view-class-config	show product class list
set manufacturer <value>	advanced-edit-class-create	define value
add hardware <OUI> <product class>	advanced-edit-class-hardware	add product class
delete hardware <OUI> <product class>	advanced-edit-class-hardware	remove product class
move hardware <OUI> <product class> <class name>	advanced-edit-class-move	move product class and hardware model to another class

7.1.1. DATAMODEL <class name> subsection commands

Command	Access	Description
show mode	cli-manage-showmode	show viewing settings
set mode <field index> <mode> <length>	cli-manage-showmode	edit viewing settings
show model all	advanced-view-class-datamodel	show list of all parameters in data model
show model parameter <name>	advanced-view-class-datamodel	show specific parameter
add parameter full <...>	advanced-edit-class-datamodel	add parameter with full initial data set
add parameter short <name> <type> <writable>	advanced-edit-class-datamodel	add parameter with short initial data set
set parameter type <name> <type> <writable>	advanced-edit-class-datamodel	edit parameter

set parameter type <name> <type> <writable>	advanced-edit-class-datamodel	edit parameter
set parameter version <name> <version>	advanced-edit-class-datamodel	edit parameter
set parameter trname <name> <trname>	advanced-edit-class-datamodel	edit parameter
set parameter flag <name> <flag>	advanced-edit-class-datamodel	edit parameter
set parameter no flag <name> <flag>	advanced-edit-class-datamodel	edit parameter
delete parameter <name>	advanced-edit-class-datamodel	remove parameter
delete object <name>	advanced-edit-class-datamodel	remove object (parameter group)
clear model	advanced-edit-class-datamodel	clear data model

8. USERS section commands

Command	Access	Description
role <role name>	access any internal function of a section	go to subsection for the specific role operations
show mode	cli-manage-showmode	show viewing settings
set mode <field index> <mode> <length>	cli-manage-showmode	edit viewing settings
show user all	authorize-view-users	show user list
show role all	authorize-view-roles	show role list
show role config <role name>	authorize-view-roles	show specific role configuration
add role <role name> <description>	authorize-edit-roles	add role
delete role <role name>	authorize-edit-roles	remove role
add user <login> <password> <role name>	authorize-edit-users	add user
delete user <login>	authorize-edit-users	delete user
set user password <login> <password>	authorize-edit-users	configure user
set user role <login> <role name>	authorize-edit-users	configure user

8.1. ROLE <role name> subsection commands

Command	Access	Description
show config	authorize-view-roles	show role configuration
set description	authorize-edit-roles	change description
add permission pattern <pattern>	authorize-edit-roles	allow access to actions with string template
add permission action <permission>	authorize-edit-roles	allow access to action
add permission class <class name>	authorize-edit-roles	allow access to class
delete permission pattern <pattern>	authorize-edit-roles	deny access to actions with string template
delete permission action <permission>	authorize-edit-roles	deny access to action
delete permission class <class name>	authorize-edit-roles	deny access to class
clear permission all	authorize-edit-roles	disable all rights
clear permission action	authorize-edit-roles	disable all actions
clear permission class	authorize-edit-roles	disable access to all classes
copy permission all <role>	authorize-edit-roles	copy all rights from another role to this one
copy permission action <role>	authorize-edit-roles	copy actions from another role to this one
copy permission class <role>	authorize-edit-roles	copy access to classes from another role to this one
set full permission all	authorize-edit-roles	set full access by actions and classes
set full permission action	authorize-edit-roles	set full access by actions
set full permission class	authorize-edit-roles	set full access by classes

APPENDIX B SYNC AND NOSYNC MODES

SYNC mode

In this mode during operations with CPE, if the number of object instances in the server configuration doesn't match the actual number of object instances in the device, the server will remove redundant instances (from the end of the list) or add missing ones. Only then the parameter matching and setting will begin.

It means, that full configuration should be present on the server, including parameters set by default in CPE, otherwise non-specified instances will be removed from CPE.

Example:

Configuration on server:

```
InternetGatewayDevice.Layer2Bridging.Bridge.1.BridgeName = other1
InternetGatewayDevice.Layer2Bridging.Bridge.4.BridgeName = other2
```

Configuration on device:

```
InternetGatewayDevice.Layer2Bridging.Bridge.3.BridgeName = VoIP
InternetGatewayDevice.Layer2Bridging.Bridge.4.BridgeName = STB
InternetGatewayDevice.Layer2Bridging.Bridge.7.BridgeName = PPPoE
InternetGatewayDevice.Layer2Bridging.Bridge.8.BridgeName = Multicast
```

Configuration on device after synchronization with server:

```
InternetGatewayDevice.Layer2Bridging.Bridge.3.BridgeName = other1
InternetGatewayDevice.Layer2Bridging.Bridge.4.BridgeName = other2
```

In this example, instances 7 and 8 of the Bridge object were deleted by the server, and parameters of instances 3 and 4 were adjusted to match the server values.

NOSYNC mode

In this mode, object instances are neither added or deleted during operations with CPE. If the object with the specified instance is found on the device, this parameter will be set from the server configuration; if there is no object with the specified instance, no changes will be made.

It means, that it is sufficient to have parameters on server that differ from the default device parameters.

Example:

Configuration on server:

```
InternetGatewayDevice.Layer2Bridging.Bridge.1.BridgeName = other1
InternetGatewayDevice.Layer2Bridging.Bridge.4.BridgeName = other2
```

Configuration on device:

```
InternetGatewayDevice.Layer2Bridging.Bridge.3.BridgeName = VoIP
InternetGatewayDevice.Layer2Bridging.Bridge.4.BridgeName = STB
InternetGatewayDevice.Layer2Bridging.Bridge.7.BridgeName = PPPoE
InternetGatewayDevice.Layer2Bridging.Bridge.8.BridgeName = Multicast
```

Configuration on device after synchronization with server:

```
InternetGatewayDevice.Layer2Bridging.Bridge.3.BridgeName = VoIP
InternetGatewayDevice.Layer2Bridging.Bridge.4.BridgeName = other2
InternetGatewayDevice.Layer2Bridging.Bridge.7.BridgeName = PPPoE
InternetGatewayDevice.Layer2Bridging.Bridge.8.BridgeName = Multicast
```

As the device and server have only one matching instance = 4, only the value of this instance has been changed in the device configuration.

Default operation mode with the specific object instances depends on the class data model.

Data model is the full list of objects and parameters, enabled for this device class, stored in the ACS server database.

Objects marked with the flag 2 are in NOSYNC mode by default, others are in SYNC mode.

List of objects in NTP class data model operating in the NOSYNC mode, i.e. marked with the flag 2:

```
InternetGatewayDevice.LANDevice.{i}.
InternetGatewayDevice.Services.VoiceService.{i}.
InternetGatewayDevice.Services.VoiceService.{i}.VoiceProfile.{i}.
InternetGatewayDevice.WANDevice.{i}.
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.
```

Data model is created by the script for each class. By modifying the script, you can modify the model itself.

The list of scripts, responsible for the class data models:

```
/usr/share/eltex-acs/data_model_01_NTP_dump.sql
/usr/share/eltex-acs/data_model_02_NTE1400_dump.sql
/usr/share/eltex-acs/data_model_03_TAU_dump.sql
```

There are the following records at the end of the script:

```
UPDATE datamodelbean set flag = 2 where hwc_id = 1 AND
       name = 'InternetGatewayDevice.Services.VoiceService.{i}.';
```

This record sets flag 2 for the following object: InternetGatewayDevice.Services.VoiceService.{i}.

To apply the modified data model you should execute the following command:

```
mysql -uuser -ppassword < /usr/share/acs/data_model_*.sql
```

and restart acsd.

Enable NOSYNC mode for object instance without modifications of class data model

When creating property in ACS server CLI, add '#' character before the instance to operate with it in NOSYNC mode.

Example:

Standard case of assigning property rule

```
(acs-cpe- 'ELTX06002656') set property
"InternetGatewayDevice.Layer2Bridging.Bridge.3.BridgeName" "VoIP"
```

Enable NOSYNC mode for the object instance InternetGatewayDevice.Layer2Bridging.Bridge.

```
 (acs-cpe- 'ELTX06002656') set property
"InternetGatewayDevice.Layer2Bridging.Bridge.#3.BridgeName" "VoIP"
```

APPENDIX C. INDIRECT ADDRESSING

Given that TR-069 creates configuration objects with arbitrary indices during the operation with terminal devices, it is increasingly difficult to work with the current device configuration.

By using the indirect addressing mechanism, it is not necessary to know the instance in order to operate the specific object instance, it is sufficient to know only one of the unique parameters of this instance.

In the property rule, you may specify the following records instead of the unknown instance:

```
.%+KeyField=KeyValue%
.%KeyField=KeyValue%
.%+=KeyValue%
.%=KeyValue%
```

where:
‘%’—masking character;
‘+’—permission to add an object, if it is absent. By default (when , '+' is missing), it is forbidden to add new objects;
KeyField—key paramete;
KeyValue—key parameter value.

KeyField parameter may not be given explicitly, in this case its value will be taken from the data model. Key data model parameters are market with the **flag 1**.

The list of key parameters for ELTEX_NTP class:

```
InternetGatewayDevice.Layer2Bridging.Bridge.{i}.BridgeName
InternetGatewayDevice.Layer2Bridging.Filter.{i}.FilterKey
InternetGatewayDevice.Layer2Bridging.Marking.{i}.MarkingKey
InternetGatewayDevice.Layer2Bridging.AvailableInterface.{i}.AvailableInterfaceKey
InternetGatewayDevice.QueueManagement.Classification.{i}.ClassificationKey
InternetGatewayDevice.QueueManagement.Policer.{i}.PolicerKey
InternetGatewayDevice.QueueManagement.Queue.{i}.QueueKey
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.IPInterface.{i}.X_BROADCOM_COM
_IfName
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.IPInterface.{i}.X_BROADCOM_COM
_FirewallException.{i}.FilterName
InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.IPInterface.{i}.X_BROADCOM_COM
_IpFilterCfg.{i}.FilterName
InternetGatewayDevice.LANDevice.{i}.LANEthernetInterfaceConfig.{i}.X_BROADCOM_COM_IfName
InternetGatewayDevice.LANDevice.{i}.LANUSBInterfaceConfig.{i}.X_BROADCOM_COM_IfName
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.BSSID
InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.PreSharedKey.{i}.PreSharedKey
InternetGatewayDevice.LANDevice.{i}.X_BROADCOM_COM_LANEponInterfaceConfig.{i}.IfName
InternetGatewayDevice.LANDevice.{i}.Hosts.Host.{i}.IPAddress;
InternetGatewayDevice.WANDevice.{i}.X_BROADCOM_COM_XTM_Interface_Stats.{i}.Port';
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.X_BROADCOM
_COM_IfName
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.X_BROADCOM
_COM_PortTriggering.{i}.Name
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.X_BROADCOM
_COM_FirewallException.{i}.FilterName
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.X_BROADCO
M_COM_IfName
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.X_BROADCO
M_COM_PortTriggering.{i}.Name
InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.X_BROADCO
M_COM_FirewallException.{i}.FilterName
InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}.X_BROADCOM_COM_PolicyRoutingName
InternetGatewayDevice.Layer3Forwarding.ForwardingDyn.{i}.X_BROADCOM_COM_PolicyRoutingName
InternetGatewayDevice.Services.StorageService.{i}.UserAccount.{i}.Username
InternetGatewayDevice.Services.VoiceService.{i}.VoiceProfile.{i}.Line.{i}.Codec.List.{i}.E
ntryID
```

Modification of the key data model parameters is performed by analogy to SYNC-NOSYNC modifications—by editing the data model scripts.

Assign key parameter in data model

Example

```
UPDATE datamodelbean set flag = 1 where hwc_id = 1 AND
 name = 'InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}.BSSID';
```

For InternetGatewayDevice.LANDevice.{i}.WLANConfiguration.{i}. object, the key parameter is *'BSSID'* by default.

Examples of indirect addressing in server operations

Example 1

```
(acs-cpe-'ELTX06002656')set property
"InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.%DirectoryNumber=123456
7%.AuthUserName" "tester"
```

Interpretation

Interpretations of this rule are as follows:

1) Do not use SYNC mode for the specific instance.
2) Identify object instance—{i}, which parameter value *InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.{i}.DirectoryNumber* is equal to '1234567'. Sought-for instance {i} will be used in the action 3. If this object is not found, stop processing this rule.
3) For the *InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.{i}. AuthUserName* parameter, assign the *'tester' value.*

Example 2

```
(acs-cpe-'ELTX06002656')set property
"InternetGatewayDevice.Layer2Bridging.Bridge.%+BridgeName=Bridge1%.VLANID" "1100"
```

Interpretation

Interpretations of this rule are as follows:

1) Use SYNC mode for the specific instance.
2) Identify object instance—{i}, which parameter value *InternetGatewayDevice.Layer2Bridging.Bridge.{i}. BridgeName* is equal to *'Bridge1'*.
   Sought-for instance {i} will be used in the action 3. If this object is not found, add new object with the specified BridgeName value.
3) Define value '1100' for the 'InternetGatewayDevice.Layer2Bridging.Bridge.{i}.VLANID' parameter.

APPENDIX D. AUTOMATIC UPDATE OF CPE LOGIN/PASSWORD FOR ACS SERVER CONNECTION.

To prevent problems with server login/password updates, there is a mechanism, that monitors changes of the *username* and *password* on the server and automatically updates this data on the device during the next session.

Operating principles of authorization data automatic update mechanism

1. Device can always establish connection using the default *username* and *password* values. Default values are set as general for ACS server in (acs-settings) section.
2. When setting personal passwords via CLI (in (acs-cpe-'xxxxxxxx') section) or NBI, the password will be changed on the device during the next session.
3. Until the device is updated with a new *username* and *password*, previous authorization data will remain valid.
4. If personal *username* and *password* are not defined, the default ones will always be valid.

TECHNICAL SUPPORT

Contact Eltex Service Centre to receive technical support regarding our products:

29v Okruzhnaya st., Novosibirsk, Russian Federation, 630020
Phone number: +7(383) 274-47-87
+7(383) 272-83-31
E-mail: **techsupp@eltex.nsk.ru**

Visit Eltex official website to get the relevant technical documentation and software, benefit from our knowledge base, send us online request or consult a Service Centre Specialist in our technical forum.

https://eltex-co.com/
https://eltex-co.ru/forum
https://eltex-co.com/support/knowledge
https://eltex-co.com/support/downloads/