Complete solutions for networking

Eltex.ACS.GUI

Operation Manual, version 1.1 (13 December 2013)

Autoconfiguration system

Document version	Issue date	Revisions
Version 1.1	13 December 2013	Second issue
Version 1.0	21 May 2013	First issue

NOTES AND WARNINGS

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

Warnings are used to inform the user about situations harmful for the device and the user alike, which could cause malfunction or data loss.

HARDWARE AND SOFTWARE REQUIREMENTS:

Minimum system requirements for operator station:

- CPU Pentium E5700 3.0GHz;
- 2GB RAM;
- 80GB HDD;
- Ethernet 100/1000Mbit/s network adapter;
- 1366x768 or higher-resolution display;
- MS Windows /XP/2000/Vista/7 or Linux operating system;
  - Java JRE VM (SUN JRE 6.18 or later);
  - Web browser with java plugin support.

Minimum system requirements for server:

- CPU Intel Core 2 Duo E7500 3GHz;
- 4GB RAM;
- 500GB HDD;
- Ethernet 100/1000Mbit/s network adapter;
- Ubuntu or Debian operating system.

CONTENTS

1 ABSTRACT

This manual describes monitoring, firmware update, configuration management and diagnostics of subscriber terminals (CPE), registered on Eltex.ACS server.

2 DESCRIPTION

Eltex.ACS system was designed for establishing a unified subscriber terminal (CPE) configuration and monitoring system. The system is equipped with the graphical user interface (GUI) and allows to facilitate the following equipment operations:

- Activation and reconfiguration of any network devices that support TR-069.
- Subscriber device status monitoring.
- Troubleshooting.
- User configuration backup.
- Centrally-controlled firmware update.

3 INSTALLATION AND CONFIGURATION

Eltex.ACS.GUI system employs 'client-server' architecture. Access server can be represented by any computer that have sufficient performance to process multiple requests (server requirements depend on the quantity of network devices and the number of workstations for technical personnel). The system uses Linux operating system. Server operates on Java virtual machine.

MySQL database serves as a storage, thus no license purchase is required. Web access to system functions is provided via Apache Tomcat also without any licensing.

To create a workstation (in order to launch the graphic client application) you will need a PC without any special requirements. PC should come with pre-installed up-to-date Windows OS (Windows 2000, XP, Vista, 7, 8) or Linux OS with graphics subsystem. Java virtual machine (SUN JRE 6.18 or later) and a web browser with java plugin support are mandatory. IE, Firefox, Opera, Google Chrome.

We recommend to install the server part on the multiprocessor computer running Linux OS. In this case, you will be able to install ACS core, ACS.GUI, and MySQL on a single PC. For installation manual, see 'Eltex.ACS Operation Manual'.

4 CONSOLE APPLICATION APPEARANCE AND FEATURES

User interface window is divided into three parts.

Fig. 1. ACS.GUI user interface window

1. **Control panel and control menu** — administration toolbar.

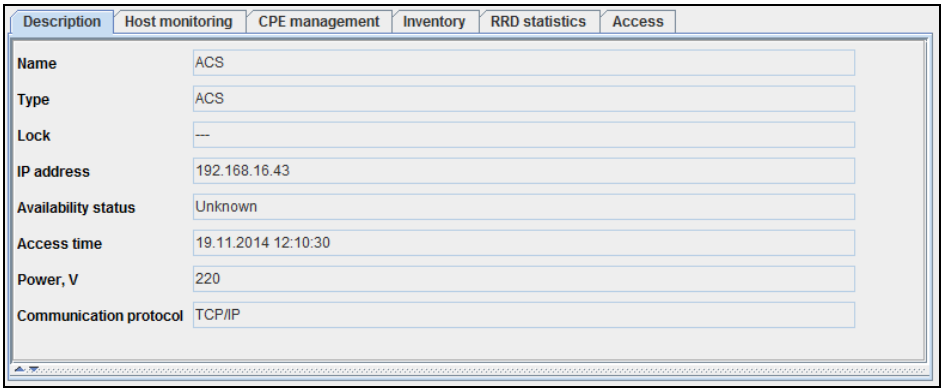
2. **Settings field** allows you to view and edit CPE and ACS server settings. Settings field contains tabs that are used as selectors for different groups of editable parameters. Some parameters are read-only, others are editable. If the user has sufficient rights for editing of current parameters, *'Edit'* button will become active. Otherwise, the button will be inactive, and the action is unavailable.

3. **Task field** allows you to view the 'group edit' command status.

5 OPERATIONS WITH ACS

5.1. DESCRIPTION

This window contains general device parameters.



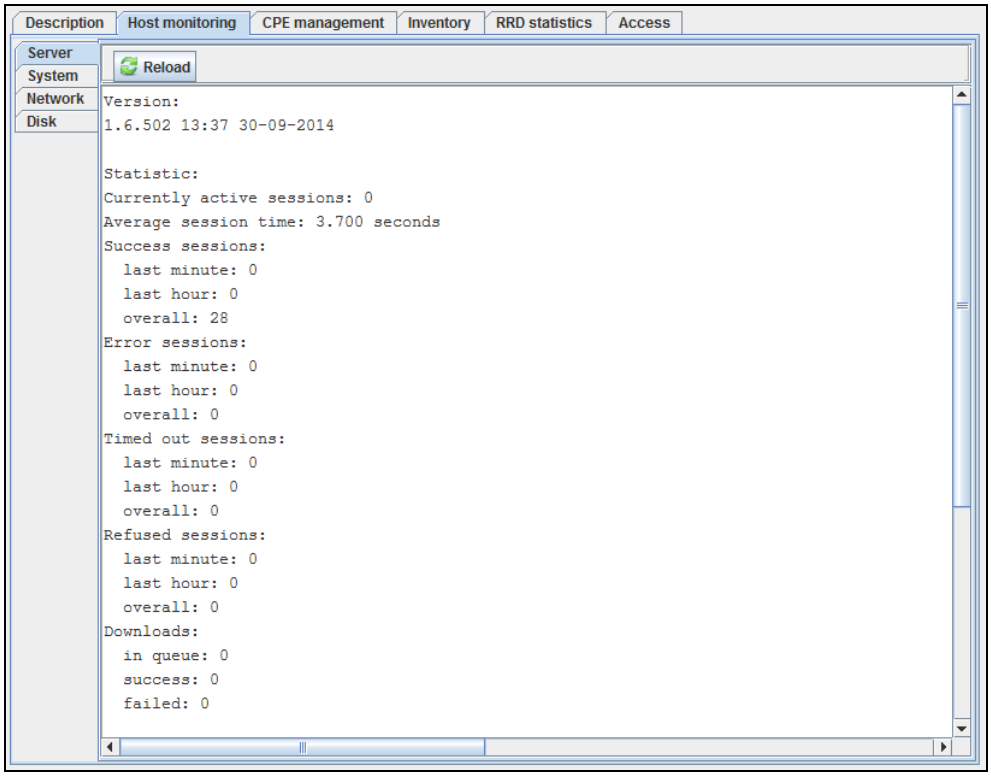
- *Name* — device name, specified during the object creation.
- *Type* — device type, specified during the object creation.
- *Lock* — identifies that the object is already being configured by another user. The name of the user, that has locked the object, will be shown in the field.
- *IP address* — ACS server IP address.
- *Availability status* — not used.
- *Access time* — not used.
- *Power, V* — ACS server power supply voltage. ACS server should be powered by the 220V AC electrical network.
- *Communication protocol* — communication protocol type.

The windows contains input field that you can use for adding the object description.

5.2. HOST MONITORING

The host monitoring menu contains the data collected during the ACS server polling.

5.2.1. SERVER



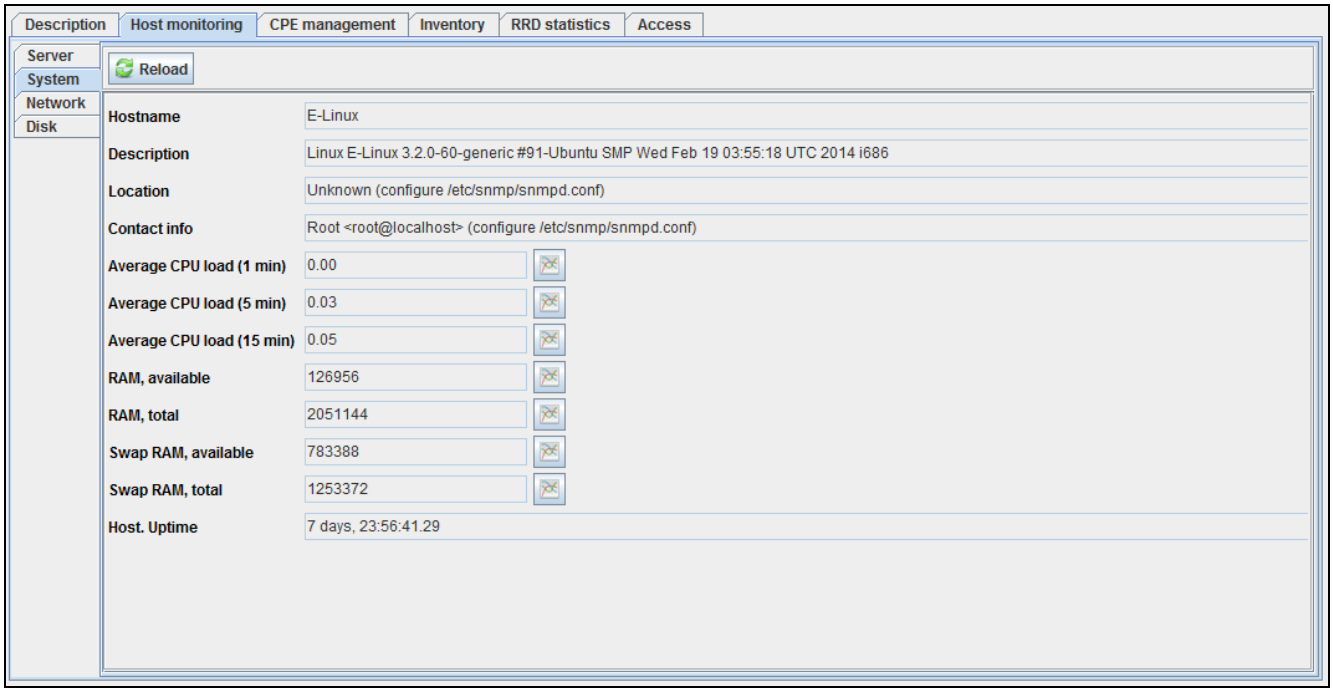
- *Version* — information on the ACS server firmware version and build date.
  Statistics:
- *Currently active sessions* — number of currently active sessions between the server and CPE.
- *Average session time* — quantity-weighted average time from the session initialization to the session end for each CPE.

  Number of sessions for the last minute/hour/total:
  - *Success sessions* — successful sessions.
  - *Error sessions* — error sessions.
  - *Timed out sessions* — sessions ended with a timeout.
  - *Refused sessions* — refused sessions (may occur during the intensive load).
  - *Downloads* — file downloads (firmware, configuration files) from the server to CPE (queued/successful/total).
- *Acsd licensed CPEs* — number of CPEs in a system, allowed by the license.
- *License description* — license parameters.
- *Memory usage* — used memory volume.
- *Cpu time (user mode)* — CPU utilization time in user mode.
- *Acsd process running for* — acsd process running time.
- *System information* — system information.

- *Number of records in DB tables:*
  - *Informs* — informs
  - *Hosts* — hosts
  - *Command queue* — command queues
  - *IP ping* — echo tests

Click *'Reload' button to refresh the information in the tab.*

5.2.2. SYSTEM



You can view these settings in the control panel — *'Information/System components information/System'* menu.

- — server network name.
- *Description* — server operating system information.
- *Location* — server location information.
- *Contact info* — server operator contact information.
- *Average CPU load 1 min/5 min/15 min* — CPU load in percentage for the last minute/5 minutes/15 minutes.
- *RAM, available* — free RAM volume, KB.
- *RAM, total* — total RAM volume, KB.
- *Swap RAM, available* — free volume in the RAM swap file, KB.
- *Swap RAM, total* — total volume of the RAM swap file, KB.
- *Host. Uptime* — device operation time since the last reboot.

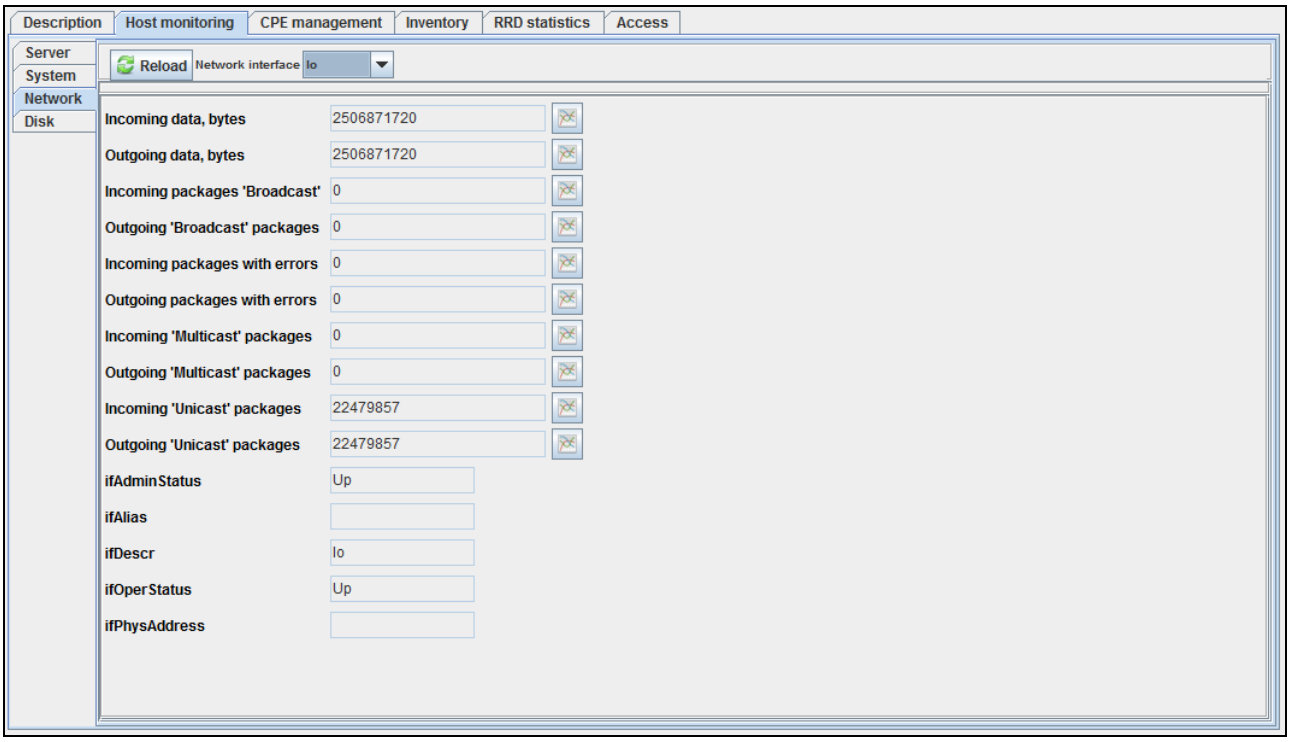
Click  button to proceed to *RRD Statistics* tab, to add new parameter monitoring task or to view the statistics for the previously assigned task (for detailed information, see chapter **5.5 RRD statistics**).

Click *'Reload'* button to refresh the information in the tab.

5.2.3. NETWORK

This tab shows the statistics of packet transmission through interfaces.

You can view these settings in the control panel — *'Information/System components information/Network'* menu.



To select the interface, use the *'Network interface'* drop-down menu.

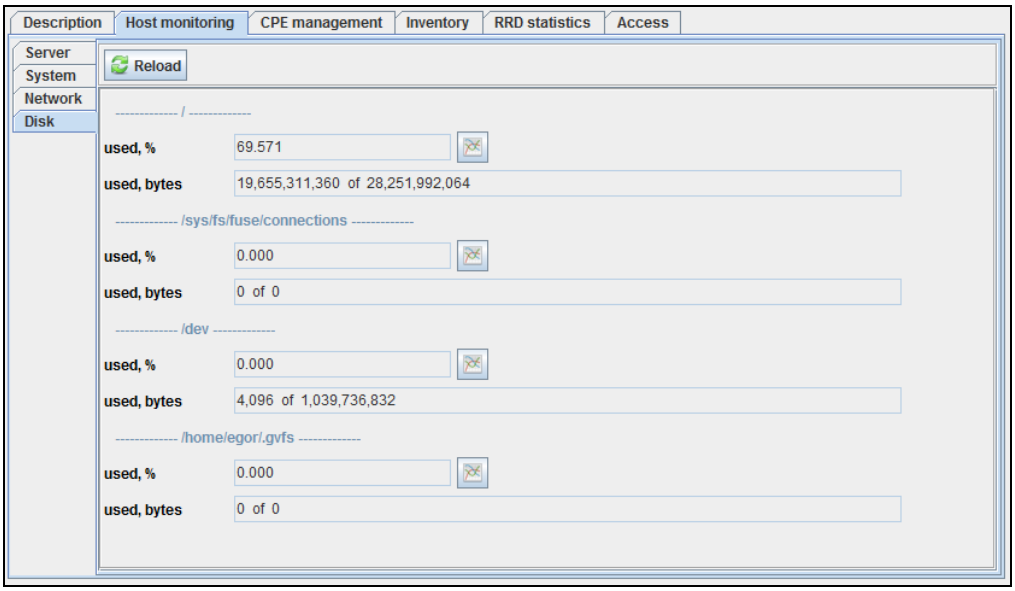
- *Incoming data, bytes* — amount of data received to the interface, in bytes.
- *Outgoing data, bytes* — amount of data sent via the interface, in bytes.
- *Incoming Broadcast packages* — amount of broadcast packets received to the interface.
- *Outgoing Broadcast packages* — amount of broadcast packets sent from the interface.
- *Incoming packages with errors* — amount of received packets with errors.
- *Outgoing packages with errors* — amount of sent packets with errors.
- *Incoming Multicast packages* — amount of multicast packets received to the interface.
- *Outgoing Multicast packages* — amount of multicast packets sent from the interface.
- *Incoming Unicast packages* — amount of unicast packets received to the interface.
- *Outgoing Unicast packages* — amount of unicast packets sent via the interface.
- *ifAdminStatus* — interface administrative status.
- *ifAlias* — name of the interface.
- *ifDescr* — description of the interface.
- ifOperStatus — interface status (up/down).
- *if PhysAddress* — interface physical address.

Click  button to proceed to *RRD Statistics* tab, to add new parameter monitoring task or to view the statistics for the previously assigned task (for detailed information, see chapter **5.5 RRD statistics**).

Click *'Reload' button to refresh the information in the tab.*

5.2.4. DISK

This tab contains the basic information on used disk space for a server.



You can view these settings in the control panel — *'Information/System components information/Disks'* menu.

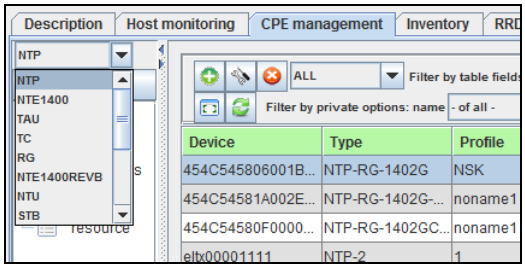
Click  button to proceed to *RRD Statistics* tab, to add new parameter monitoring task or to view the statistics for the previously assigned task (for detailed information, see chapter **5.5 RRD statistics**).

Click *'Reload'* button to refresh the information in the tab.

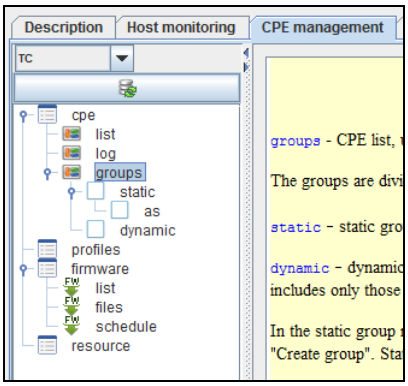
5.3. DEVICE OPERATIONS

5.3.1. SELECT OPERATION CLASS

Class (NTE1400, NTP, TAU, etc.) — group of devices that conform to the following conditions: OUI (manufacturer identifier) and ProductClass (device model identifier). Devices with unregistered OUI+ProductClass link (section **5.4.2 Classes**) will appear under the UNKNOWN class.



5.3.2. SECTION TREE



Navigation for each device class is performed using the navigation tree with the following structure:

- ***cpe*** — *list of client devices* (CPE (Customer Premises Equipment) is equipment located at the customer's premises). The list contains the following subsections:
  - *list* — table that lists all devices located in this class.
  - *journal* — list of messages that contains CPE references to the server (informs), CPE parameter setting errors, and commands from the operator to CPE and the respective replies.
  - *groups* — operations with CPE logical groups:
    - *static* — static groups. List of CPEs in a group is edited by the operator and cannot be changed in his absence. The static group may have several nesting levels. To create the nested group, right-click the specific group and select the 'Create group' menu item. Parent static group includes CPE devices from child static groups.
    - *dynamic* — dynamic groups. List of CPEs in a group depends on the group formation rules. In each moment of time, the group includes only those devices that conform to the rules.

- ***profiles*** — configure lists of CPE configuration rules.

- ***firmware*** — this tab controls the subscriber device firmware update process and contains three sections:
  - *list* — deals with CPE firmware update rules. Here you can specify the file for firmware update rules and link the rule to CPE profiles and models.
  - *files* — contains the list of firmware files registered in the system and allows to perform 'add' and 'remove' operations.

- *schedule* — deals with the firmware update schedule configuration. By default, the *'Global for all'* record is enabled for all rules. You can create an individual schedule for each update rule, if necessary.

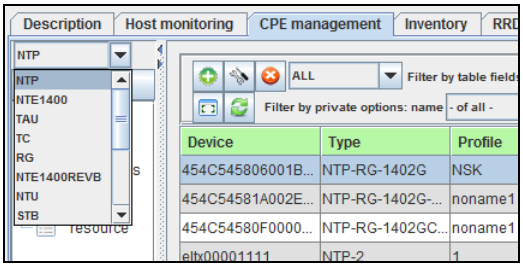
- ***resource***[1] — contains the list of other files registered in the system and allows to perform 'add' and 'remove' operations.

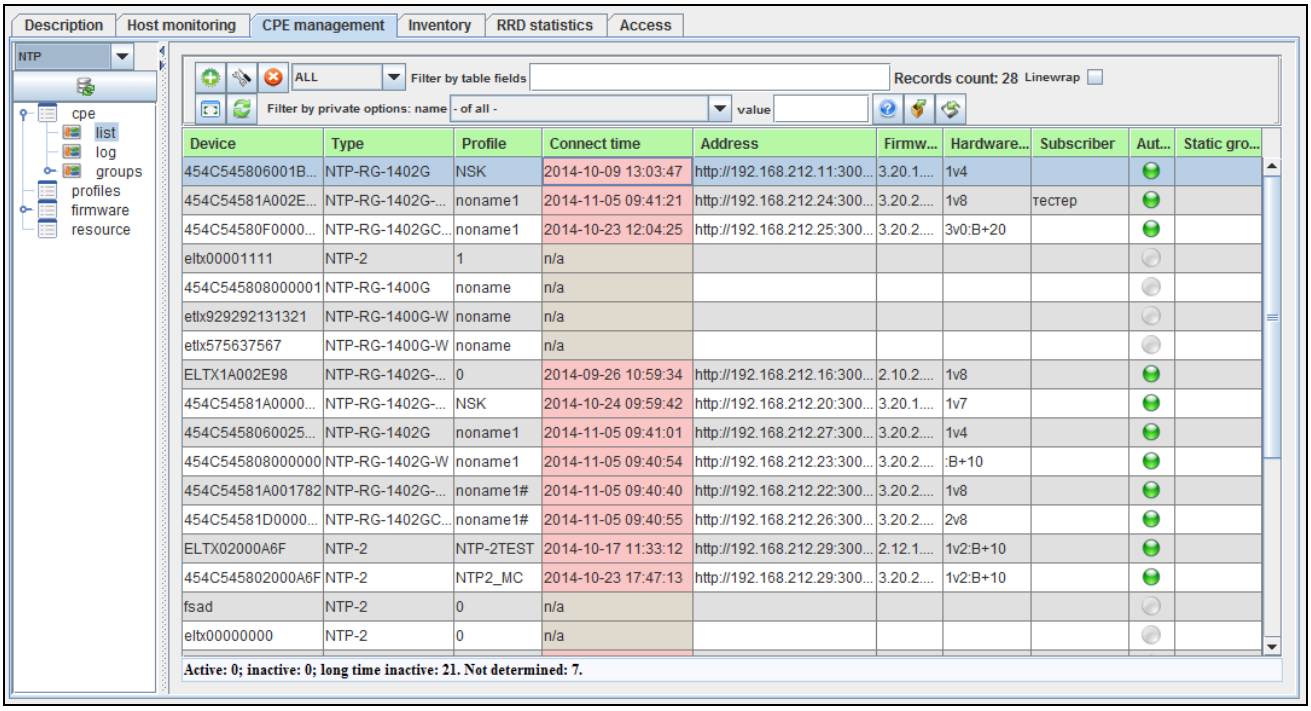
5.3.3. CPE — DEVICE OPERATIONS

5.3.3.1. CPE-LIST. DEVICE LIST

List of the current class devices is shown in the *'Device operations'* tab located in the *'cpe/list'* tree branch.

Devices with the data in 'Connect time' column highlighted green have connected to the server recently and are available at the moment (highlighted yellow — connected a long time ago, red — a long while ago). You can configure time intervals for state transitions in the *'Administration/Server configuration/System modules'* menu.





[1] Only for TC class

5.3.3.1.1. OPERATIONS WITH THE LIST CPE

Select the *'Linewrap'* checkbox to show the full information in the *'Address'*, *'Subscriber'*, *'Static groups'* columns regardless of their width.

CPE authorization status

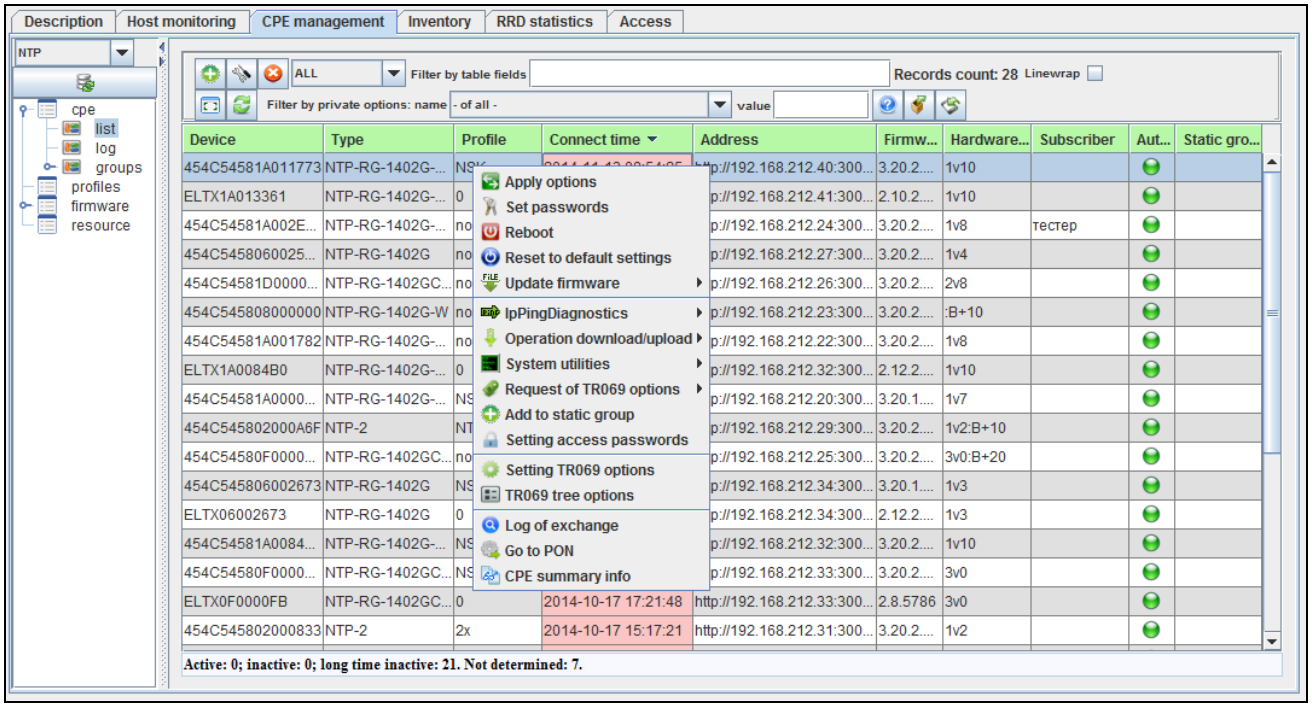
— authorization is not defined.
— authorization success.
— authorization error.

CPE table search operating principles

- Table record is reproduced as a string depending on the visibility of fields (columns).
- The search is conducted in the search field by words delimited with spaces.
- The filter is not case-sensitive.
- Criteria are joined according to the logical '**AND**' principle; records that satisfy all criteria will be shown in search results.
- Other special characters (except for the space) will not be used when entered in the search field.

To change the columns displayed in the table, click the  (*'Change table fields'*) button.

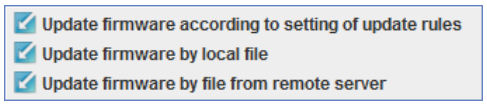
Click the right mouse button on the CPE row in the general list to show the device control menu:



- ***Apply options*** — CPE options synchronization command, common operation that should be performed after implementing any changes to CPE settings on the server. New settings will be applied during the next periodic session or may be applied with the current command.
- ***Set passwords***[1] — command allows you to set CPE passwords. If you change passwords for services (PPP, SIP, etc.), execute the command to apply them for CPE.
- ***Reboot*** — reboot the CPE.
- ***Reset to default settings*** — reset the terminal to factory settings.

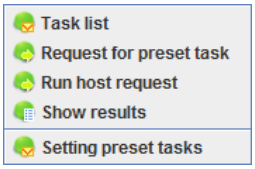
[1] Command is used in the *safe* server operation mode.
In the *unsafe* server operation mode (set by default), this button is not used.
For detailed description, see chapter **14. Frequently asked questions** of the **Eltex.ACS**

- ***Update firmware:***
  - *Update firmware according to setting of update rules* — update CPE firmware immediately (using the firmware update rule assigned to the profile or individual update rule (always has a higher priority) while ignoring *'Firmware downgrade'* and *'Update on restart only'* checkboxes, and firmware update schedule.

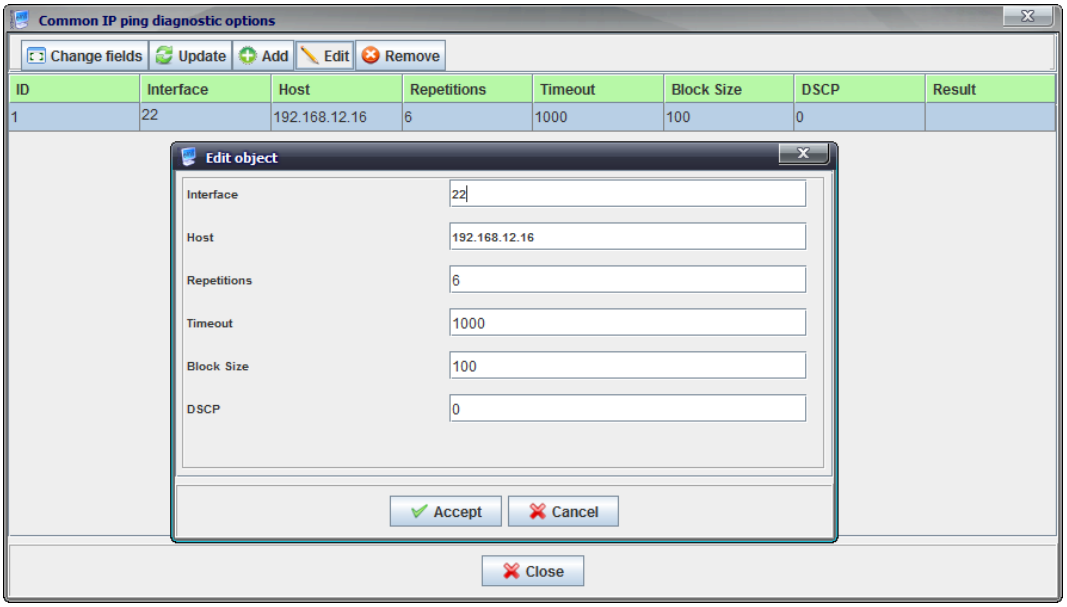


  - *Update firmware by local file* — update firmware from the local firmware file located in this class (firmware-list).
  - *Update firmware by file from remote server* — update firmware from the firmware file located on the remote server, full path to the file should be provided.

- ***IPPingDiagnostics*** — function that performs echo test from CPE to the arbitrary LAN/WAN host:



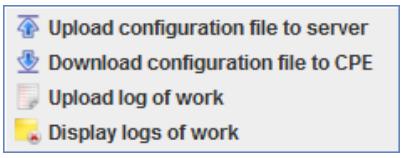
  - *Task list* — view and edit the list of *IPPing* diagnostics tasks that has been executed from CPE by the operator during the current session.
    - *Request for preset task* — execute *IPPing* diagnostics using the preset task.
    - *Run host request* — request *IPPing* by specifying only the host address for the echo test. The interface used for the test will be selected automatically by CPE from the route table.
    - *Show results* — show all *IPPing* diagnostics results for CPE textually.
    - *Setting preset tasks* — configure tasks that will be executed with *'Request for preset task'* command.



    - *Interface* — device interface that will be used by a test. If the parameter is not specified, the test should be performed through the interface selected from the route table on the device.
    - *Host* — echo test server domain or IP address.
    - *Replications* — number of ping queries in the test.
    - *Timeout* — query response time that will invalidate the query, if exceeded.
    - *Block size* — data block size for a single query.
    - *DSCP* — corresponding field in the query.

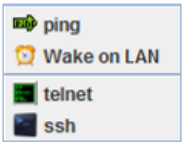
- ***Download/upload operations*** — manage device log and configuration file download/upload operations:

- *Upload configuration file to server* — upload the configuration from CPE to the server as a single file. CPE configuration file will be created in the following directory: */root catalog/config/<class>/<serial number>/* name in *YYYY-MM-DD_HH-MM-SS* format.



- *Download configuration file to CPE* — download CPE configuration to the server as a single file. File list from the directory */root catalog/config/<class>/<serial number>/* is shown.
- *Refresh graphics resources*[1] — execute the command to refresh the graphics resources — logo, boot screen.
- *Upload log file* — execute the command to upload CPE log file to the server.

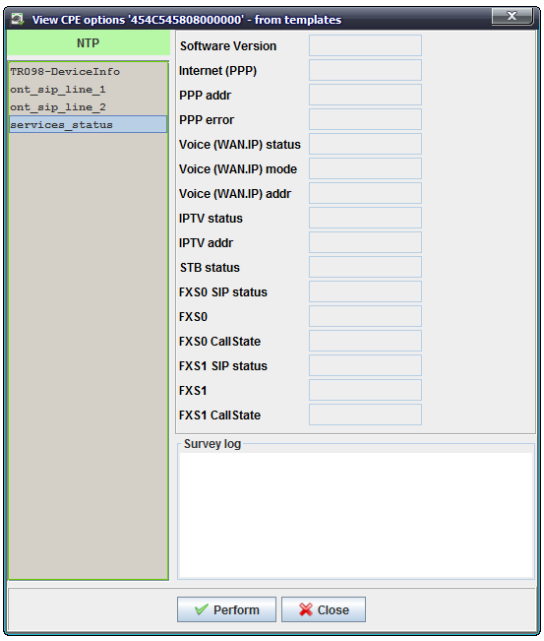
- ***System utilities*** — contains the bundle of utilities:



  - *ping — echo test (sends the simple 'ping' command) from the server to device.*
  - *Wake on LAN*[1] — power on the device via WOL protocol.
  - *telnet*[2] — connect to device via Telnet protocol.
  - *ssh*[2] — connect to device via SSH protocol.

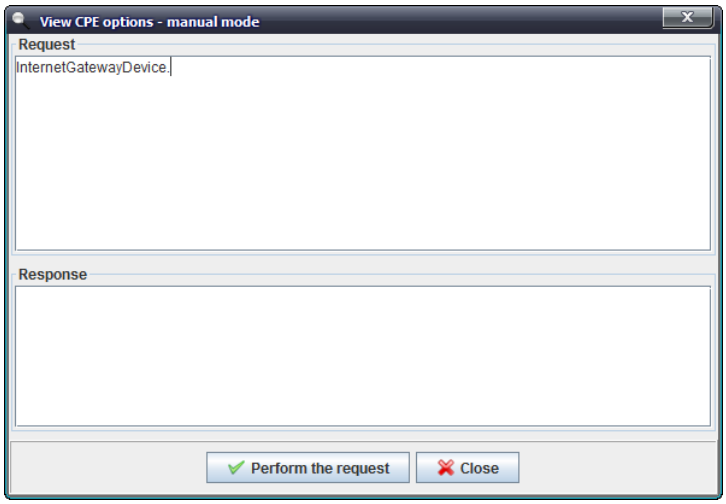
- ***Request TR069 parameters*** — commands that allow to receive the data:

  - *Templates* — request monitoring parameters from CPE using the preconfigured template. You can flexibly adapt these templates to any class by editing the system files.

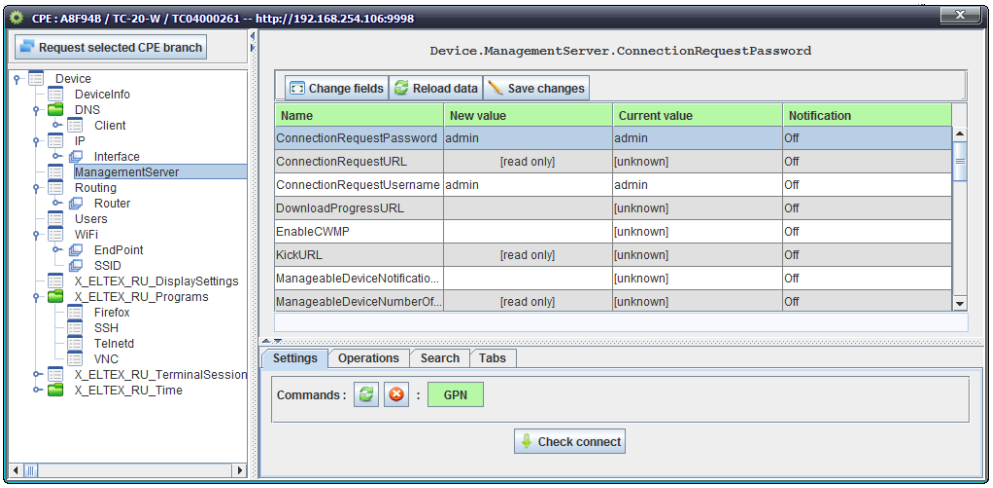


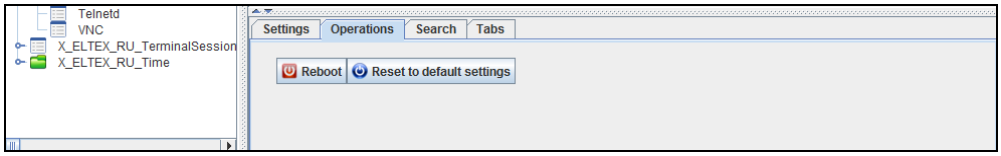
  - *Random request* — request parameters from CPE by defining the full parameter name in the request.

[1] Only for TC class
[2] Not supported in the current firmware version.



- ***Add to static group*** — add the current device into the existing static group.
- ***Configure access passwords*** — configure access passwords from the server to CPE and from CPE to the server.
- ***TR069 parameter configuration*** — configure and view TR-069 parameters, set the profile and checkboxes.
- ***TR069 parameter tree*** — configure and view TR-069 parameters as a tree. Tree operations are performed in the real time, all changes implemented by the operator into the CPE configuration will not be saved in the server database.



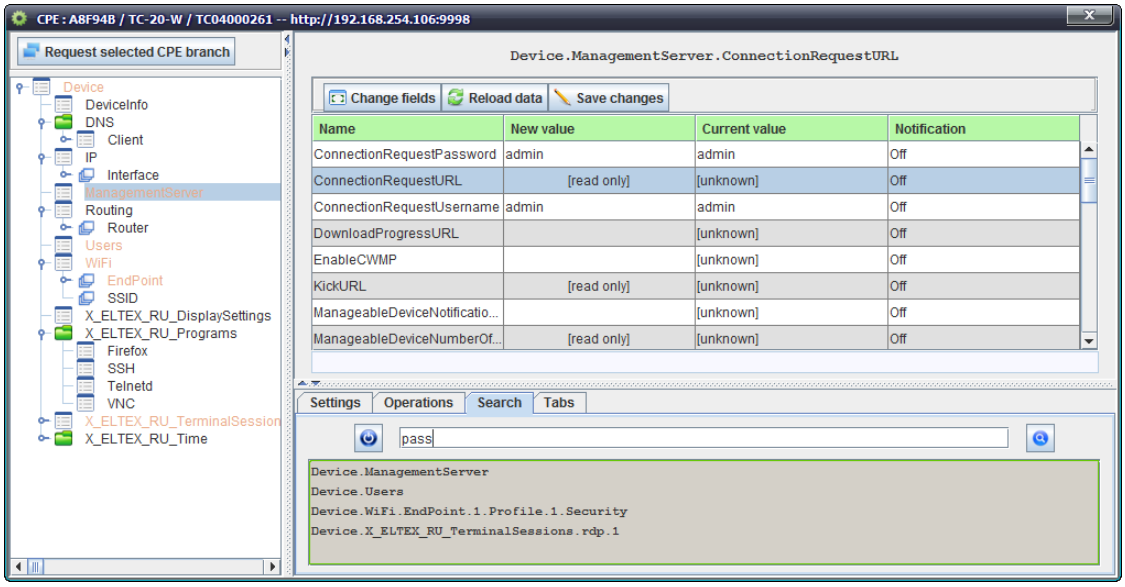


To request the parameter value (parameter branch) from the device, click the *'Request selected CPE branch'* button.

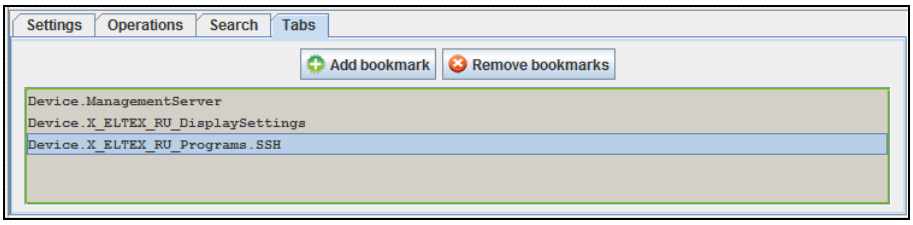
Click the *'Reload data'* button to display the parameter values stored in the server cache.

Commands sent to CPE during tree operations are displayed in the *'Commands'* field. To change the parameter value, enter a new value into the *'New value'* column and click *'Save changes'* button. To add and remove objects, right-click the specific object in a tree.

You can search for the parameter by its name:



Also, you can add the bookmark for any branch, so you can quickly navigate to it at a later time.

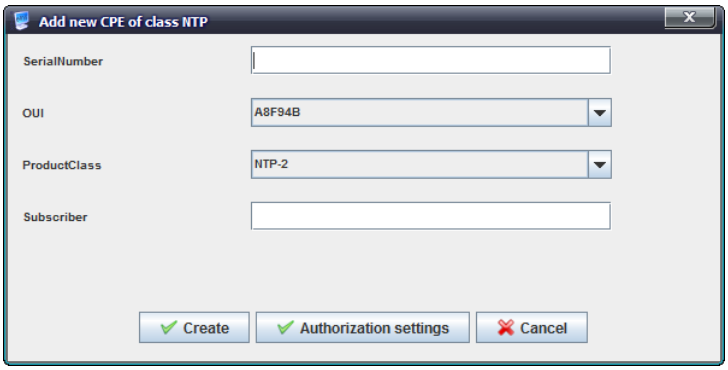


– ***Exchange log*** — go to operation log filtered by the serial number of the current CPE.

5.3.3.1.2. ADDING DEVICES

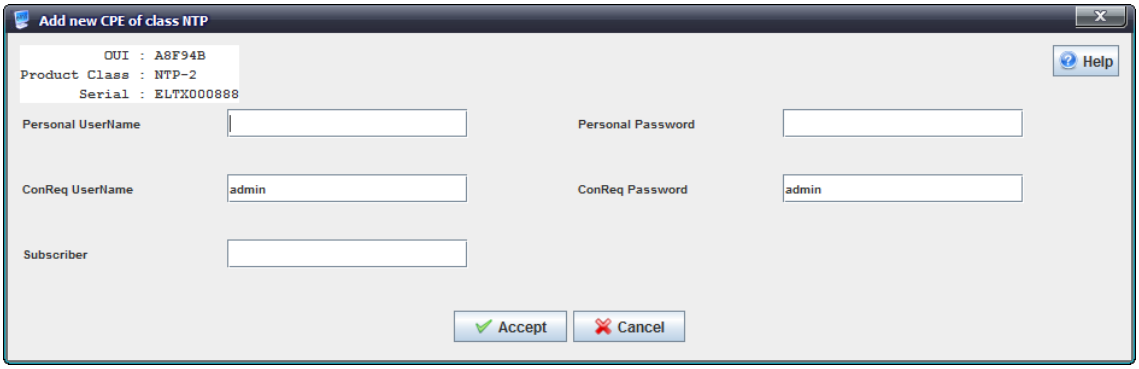
New devices appear in the list automatically. Also, you can create a device that is not yet connected to the server.

To add such a device, click the  button and enter its serial number (*'SerialNumber'*), specify the manufacturer's identifier *('OUI'),* model *('ProductClass')*, and click *'Create' or 'Authorization settings'* button.



If you click the *'Create'* button, the password configuration window will not be shown, and you will be taken to the private CPE options page.

Click the *'Authorization settings'* button to open the password configuration window.



'Personal UserName' and '*Personal Password*' — login and password that will be used by CPE for establishing connection to the server. Login, if it is used, should be unique for each CPE.

If these fields are empty, use the general login and password, defined on the server. For CPE, these parameters are configured manually or using DHCP Option 43.

'ConReq UserName' and '*ConReq Password*' — login and password that will be used by the server for establishing connection to CPE.

These fields are mandatory. If parameter values does not match on the server and CPE, you will not be able to send any commands to CPE.

'Subscriber' — subscriber description.

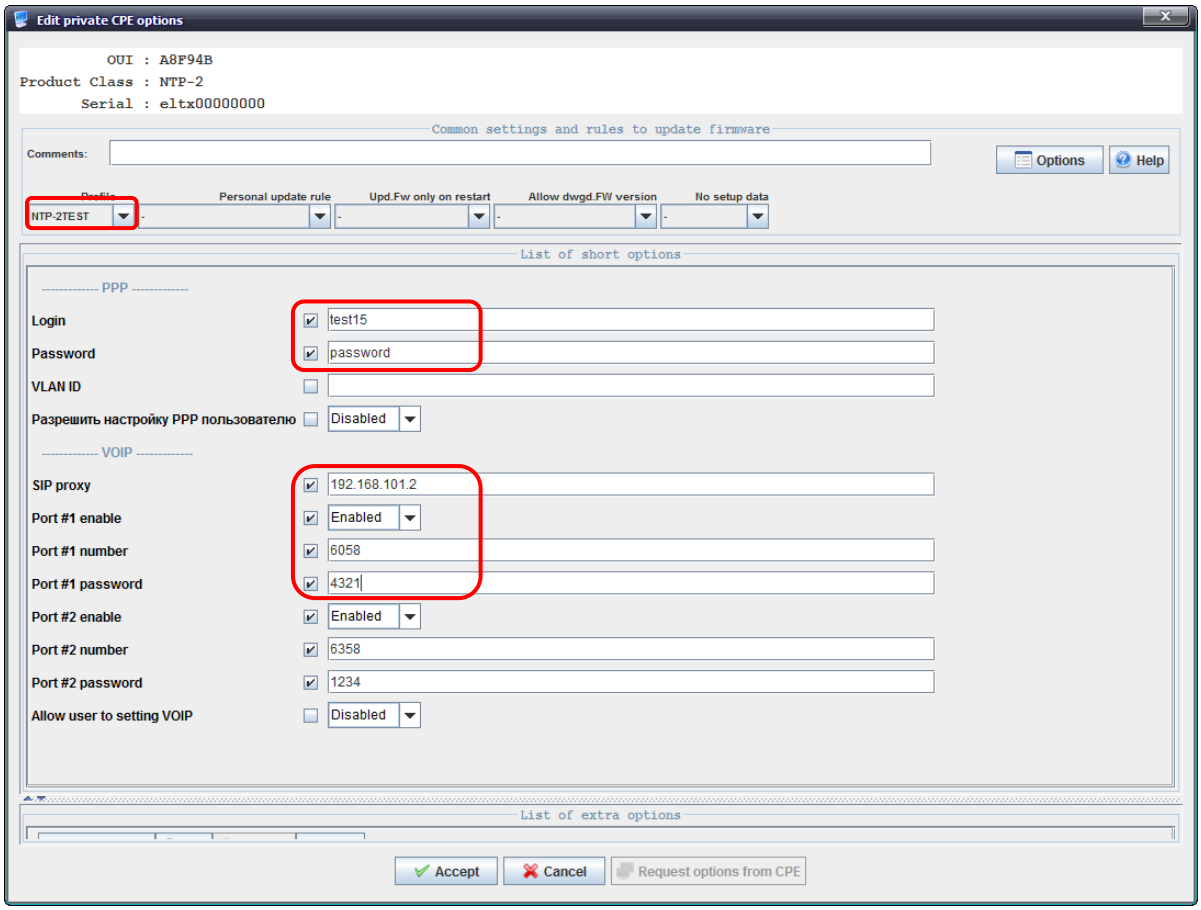
If you click the *'Accept'* button, you will be taken to the private CPE options page.

To import the devices from the CSV file, use the *'Devices/Import from CSV file'* main menu or click  button in the device list window.

To export the device list, click  button.

5.3.3.1.3. ASSIGNING PRIVATE CPE OPTIONS

In the general list, double click the row with CPE to show the private options menu.



General list of options is extracted from the profile (the figure shows *SZT-WF* profile).

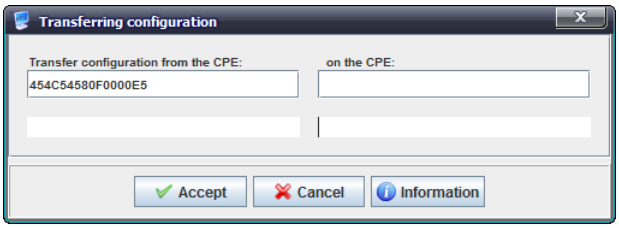
To configure private options, use *'List of short options'* and/or *'List of extra options'* sections.

- *Personal update rule* — set the personal update rule for the current CPE. This rule has a higher priority than the rule assigned to the 'Profile'.
- *'Upd. only on restart'* and *'Allow dwgd. FW version'* — duplicate the firmware update rule checkboxes, but have a higher priority.
- *No setup data* — if the flag is set, there are no profile rules or private parameters for this CPE.
- *'Full configuration'* — click this button to access the resulting list of rules that will be applied to CPE.
- *Request options from CPE* — click this button to request options from the short options list, results will be displayed in the corresponding list fields.

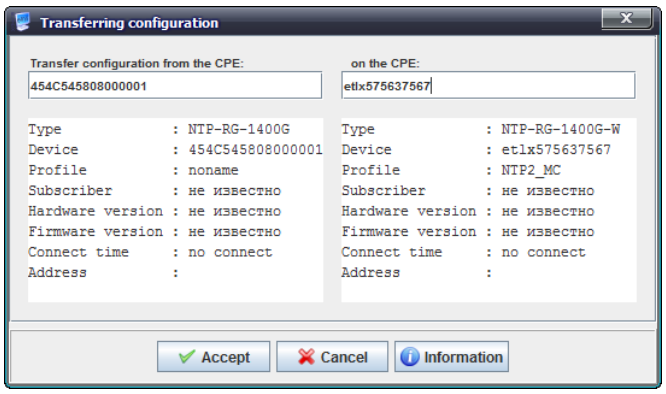
5.3.3.1.4. TRANSFERRING CPE CONFIGURATION

If you are planning to replace the CPE, you may transfer its configuration (profile, flags, and private options) to another CPE; by doing so, the configuration source CPE will be removed from the server as non-relevant.

To access the transfer menu, click the  button.

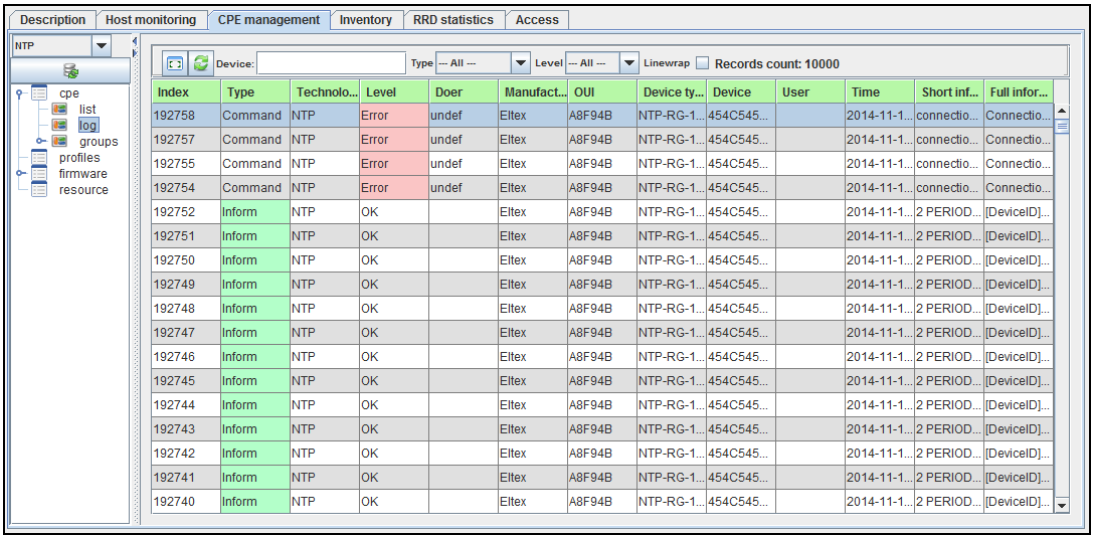


Click the *'Information'* button to show the brief information on CPE configuration, subject to transfer.



5.3.3.2. CPE-LOG. CPE DATA EXCHANGE LOG

Show the list of messages and errors, that were received during the CPE operation: list of messages that contains CPE references to the server (informs), CPE parameter setting errors, and commands from the operator to CPE and the respective replies.

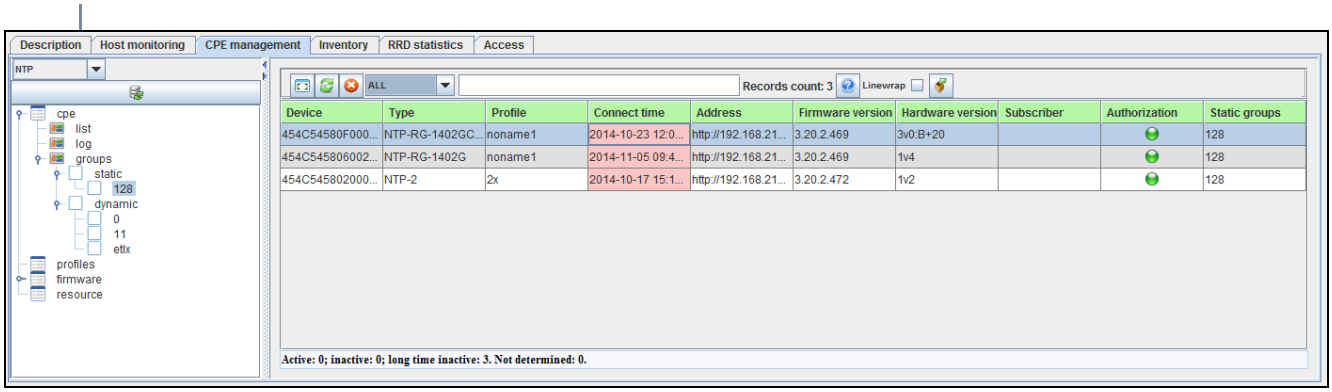


Data highlighted in red calls for special attention, they represent errors in the CPE configuration or authentication process on the server. Double-click the record to open its detailed description.

5.3.3.3. CPE-GROUPS. CPE GROUPS

5.3.3.3.1. STATIC

List of CPEs in a static group is edited by the operator and cannot be changed in his absence. It may have several nesting levels. To create the nested group, right-click the specific group and select the 'Create group' menu item. Parent static group includes CPE devices from child static groups. To add CPE into the static group, use the *'Add to static menu'* context menu command in the general CPE list.

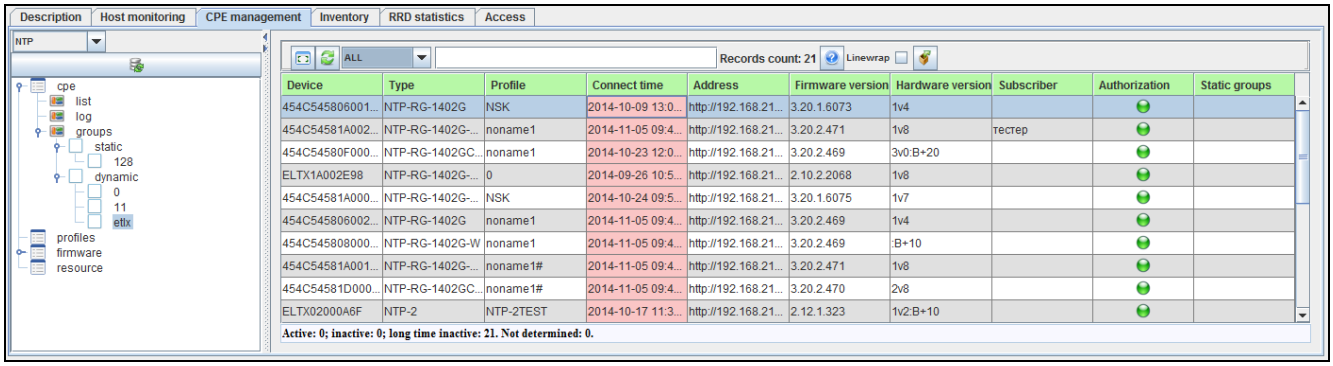


Click the right mouse button on the CPE row in the general list to show the device control menu.

To get help with the context menu commands, see Paragraph **5.3.3.1.1 Operations with the list CPE 5.3.3.1 CPE-list. Device list.**

5.3.3.3.2. DYNAMIC

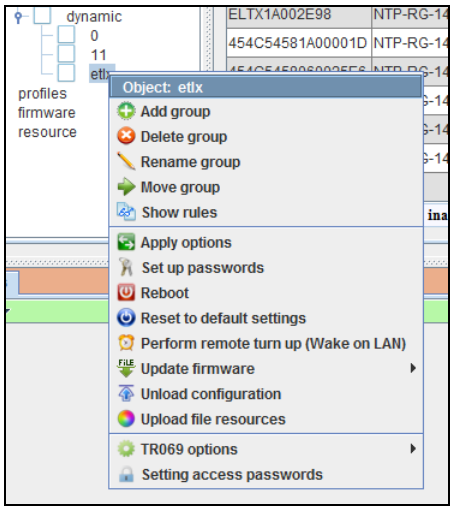
List of CPEs in a dynamic group depends on the group formation rules. In each moment of time, the group includes only those devices that conform to the rules.



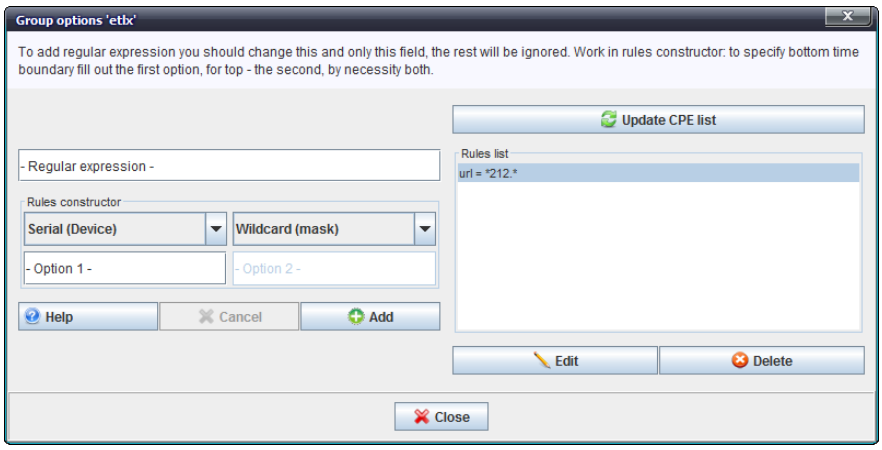
Click the right mouse button on the CPE row in the general list to show the device control menu.

To get help with the context menu commands, see Paragraph **5.3.3.1.1 Operations with the list CPE 5.3.3.1 CPE-list. Device list.**

For dynamic groups, you may edit filtering rules that define CPE inherence to the group.



To show the editor, use the context menu by right-clicking the group name in a tree.



5.3.3.3.3. REGULAR EXPRESSION OPERATION PRINCIPLES

1. Multiple regular expressions are joined together according to **'OR'** principle.

   Permitted expression formats:

   - **param = value**, in this case, 'value' is a number.
   - **param = 'value'** (equal to 'like' in sql), in this case, 'value' is a string mask.
   - **param = 'value'** (equal to 'regexp' in sql), in this case, 'value' is a regular expression.

2. You may define multiple constructions in a single expression,for example:

   param1 = value1 or param1 = value2 and param2~'value3'.

3. String parameter values (including the date) should be enclosed in the single quotes — **'value'.**

4. Numeric parameter value should be defined in expression without quotes.

5. You may use the following special characters during the creation of the 'value' (param = 'value') mask:

   - **?** — any character.
   - ***** — any characters.

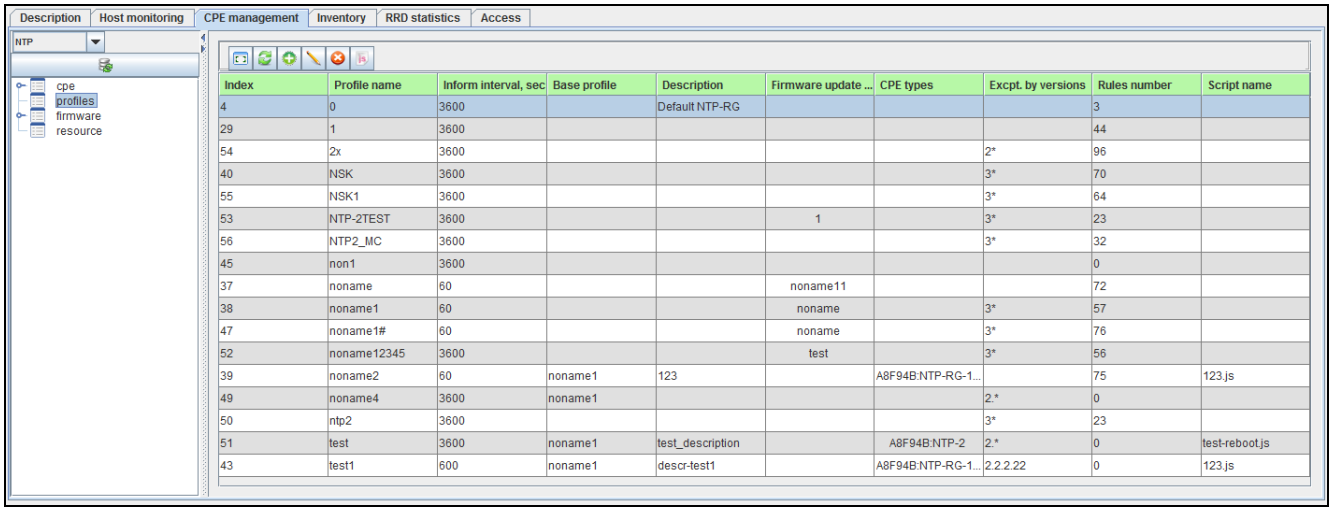
5.3.3.3.4. RULE CONSTRUCTOR OPERATIONS

1. Constructor rule of the same type are joined together according to **'OR'** principle.
2. Constructor rule of different types are joined together according to **'AND'** principle.
3. **Wildcard** comparison type is used for *Serial, Product Class, Profile, URL*, *Version, Hardware, Username, Customer ID, Conrequser, Config name, Subscriber* parameters.
4. Special characters during the Wildcard rule creation are used in the same way as for (param = value) regular expression mask.
5. Value comparison type is used for the Editor parameter only, permitted values are listed in the tooltip.
6. **Datetime** and **Current** comparison types are used for *Last contact, Config* update *time, Software update time* parameters.
7. For the **Datetime** comparison type, you should specify the time in 'YYYY-MM-DD hh:mm' format.
8. For the **Current** comparison type, you should specify the time in minutes.

5.3.4. PROFILES. CONFIGURATION PROFILES

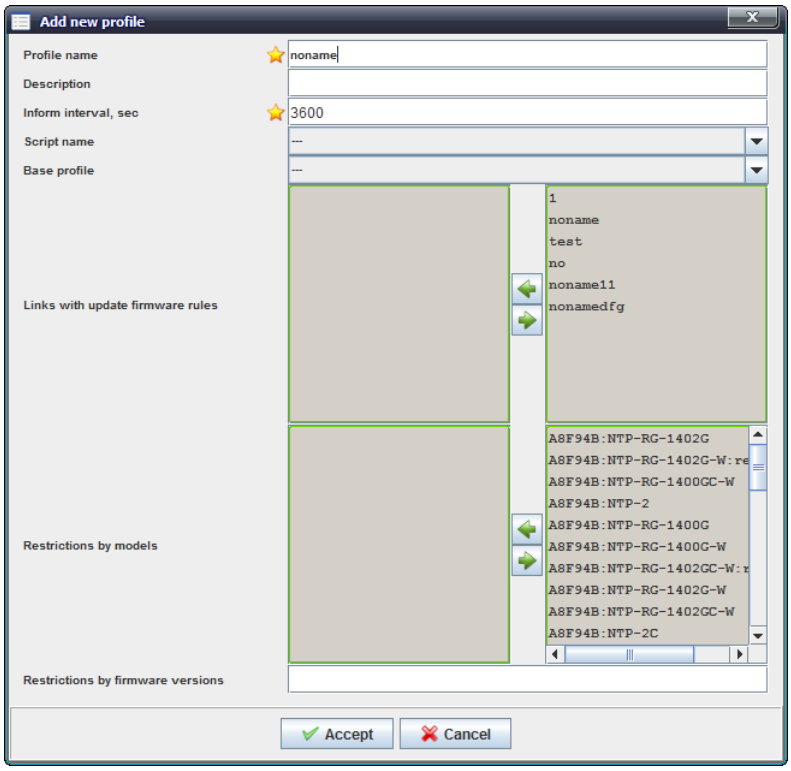
This section allows you to create profiles to be assigned to CPE afterwards.

Profile is the set of configuration rules (property) that is common for the array of devices.

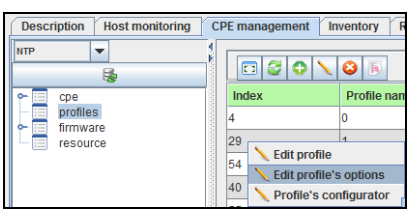


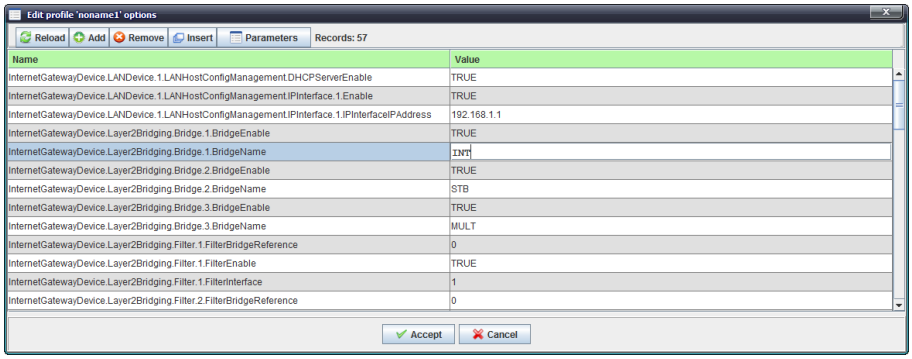
Index	Profile name	Inform interval, sec	Base profile	Description	Firmware update ...	CPE types	Excpt. by versions	Rules number	Script name
4	0	3600		Default NTP-RG				3	
29	1	3600						44	
54	2x	3600					2*	96	
40	NSK	3600					3*	70	
55	NSK1	3600					3*	64	
53	NTP-2TEST	3600			1		3*	23	
56	NTP2_MC	3600					3*	32	
45	non1	3600						0	
37	noname	60			noname11			72	
38	noname1	60			noname		3*	57	
47	noname1#	60			noname		3*	76	
52	noname12345	3600			test		3*	56	
39	noname2	60	noname1	123		A8F94B:NTP-RG-1...		75	123.js
49	noname4	3600	noname1				2.*	0	
50	ntp2	3600					3*	23	
51	test	3600	noname1	test_description		A8F94B:NTP-2	2.*	0	test-reboot.js
43	test1	600	noname1	descr-test1		A8F94B:NTP-RG-1...	2.2.2.22	0	123.js

To add a new profile, click the  button.



- *Profile name* — name of the profile.
- *Description* — arbitrary text description.
- *Inform interval, sec* — time which should pass before CPE will establish access to the server for data synchronization.
- *Script name* — .js script which will be used for the current profile (**8 Operation with scripts**);
- *Base profile* — select the base profile from the drop-down list.
- *Links with update firmware rules* — specify the update rule for profile. Select rules to be used in the current profile from the right list and move them to the left list.
- *Restrictions by models* — no restrictions are applied by default (when the left list is empty). When you move the record from the general list on the right to the left list, selected restriction will be applied. If the profile with this restriction is assigned to CPE which model is absent from the left list, this profile will not work for such CPE.
- *Restrictions by firmware versions* — no restrictions are applied by default (empty field). You can specify a restriction using the mask, where '*' represents any quantity of any characters, and '?' represents any single character. **Example:** "5.3.*; 6.?.??" If CPE firmware version falls outside the scope of the selected masks, the profile will not work for such CPE.



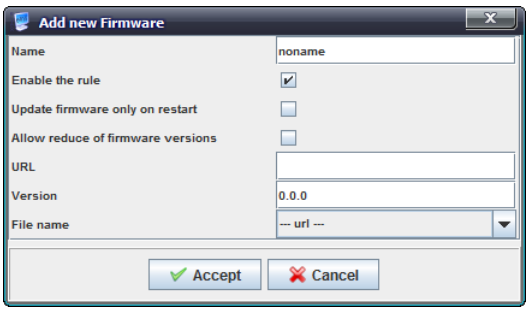


5.3.5. FIRMWARE. FIRMWARE UPDATE RULES

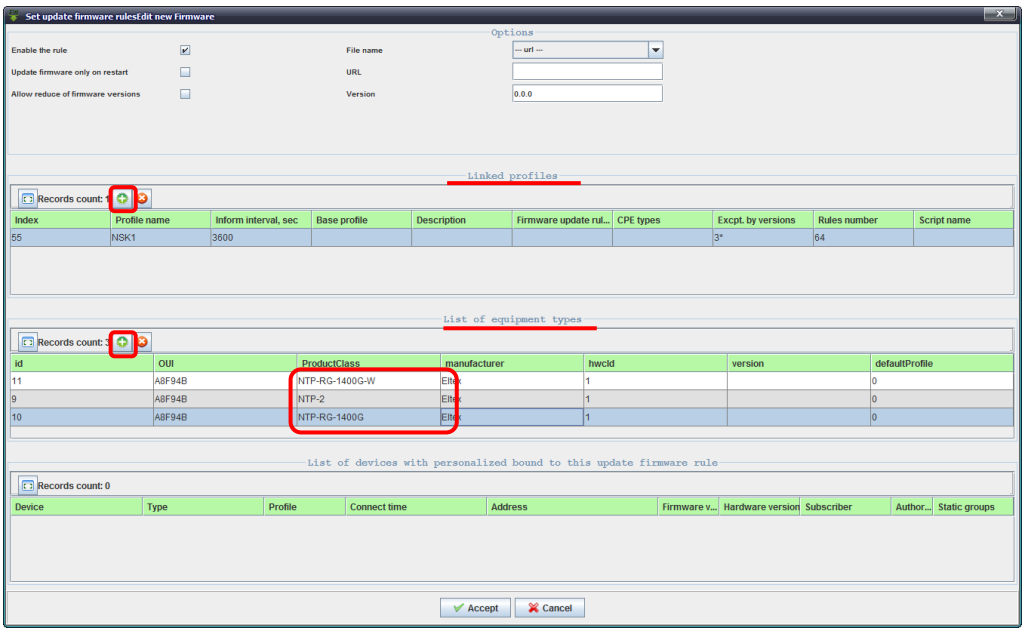
List of the firmware update rules for the current class is shown in the *'Device operations'* tab located in the *'firmware/list'* tree branch.

Adding new firmware update rule

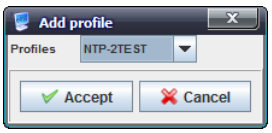
To create the firmware update rule, click the  button. In the opened window, select the file in the *'File'* field from the list of uploaded files in the '*firmware-files*' tree branch, define the name and version, and click *'Apply'* button.

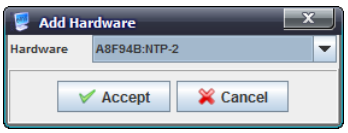


Detailed configuration window will open:



You should specify links to the required profiles and CPE types. To do this, use  buttons to add records from the drop-down lists in the respective sections.





Profile list defines profiles for options that the current firmware update rule will be linked to.

Equipment type list defines device models that the update will be applied to.

'Enable the rule' checkbox — enables/disable the selected update rule.

'Update firmware only on restart' checkbox — when selected, firmware will be updated on the next device reboot (on the inform with the BOOT event).

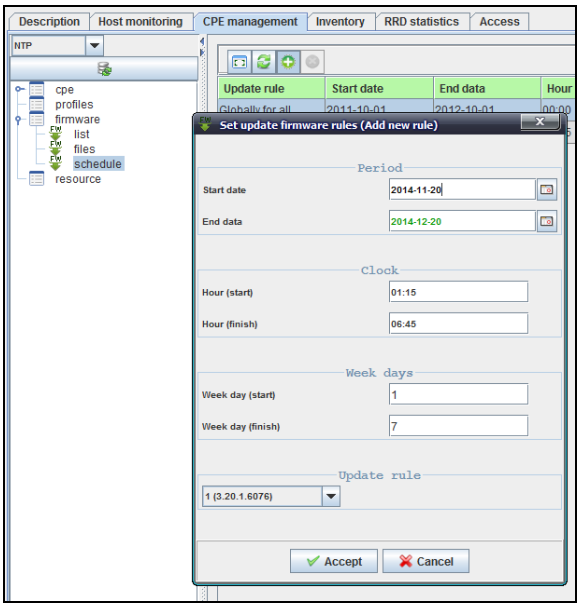
'Allow reduce of firmware versions' checkbox — when unselected, firmware on CPE will be updated only when the firmware version on server is higher than the actual device firmware version. When the checkbox is selected, this restriction will be lifted.

In the *'cpe list'* list you will find the list of devices that have the current update rule assigned as a personal rule.

Firmware update schedule configuration

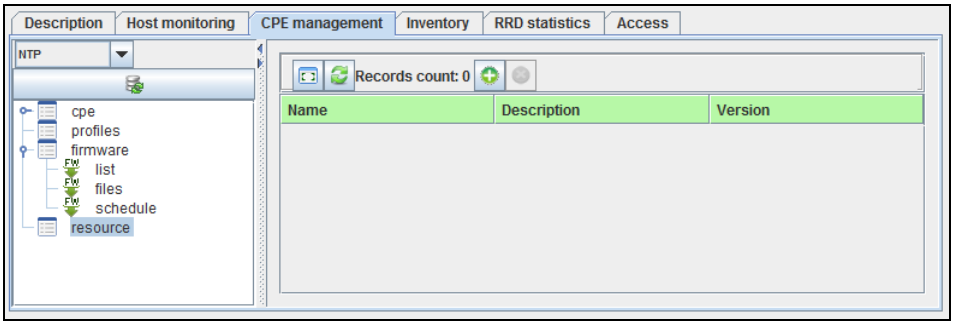
The firmware update schedule list is shown in the *'Device operations'* tab located in the *'firmware/schedule'* tree branch.

By default, the *'Global for all'* record is enabled for all records. You can create daily/weekly periodic schedule for each update rule, if necessary.



5.3.6. RESOURCE. THE LIST OF REGISTERED FILES[1]

The list of other files (except for the firmware files) registered in the system is shown in the *'Device operations'* tab located in the *'resource'* tree branch. Also, you can perform 'add' and 'remove' operations in this tab.



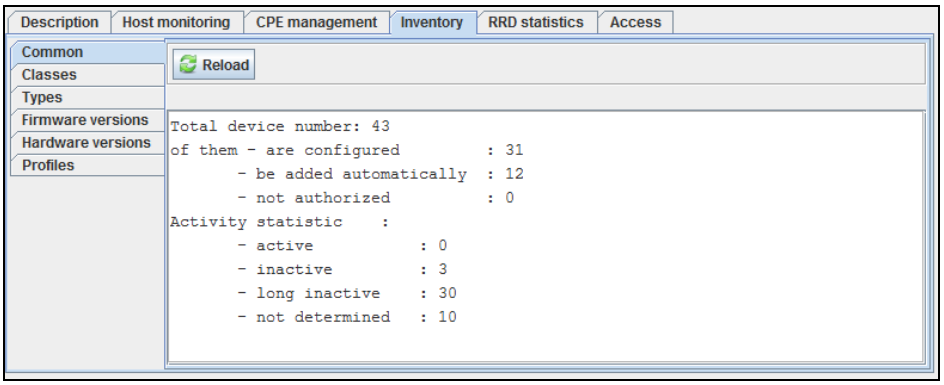
Click  button to refresh the information in the tab.

[1] Only for TC class

5.4. INVENTORY

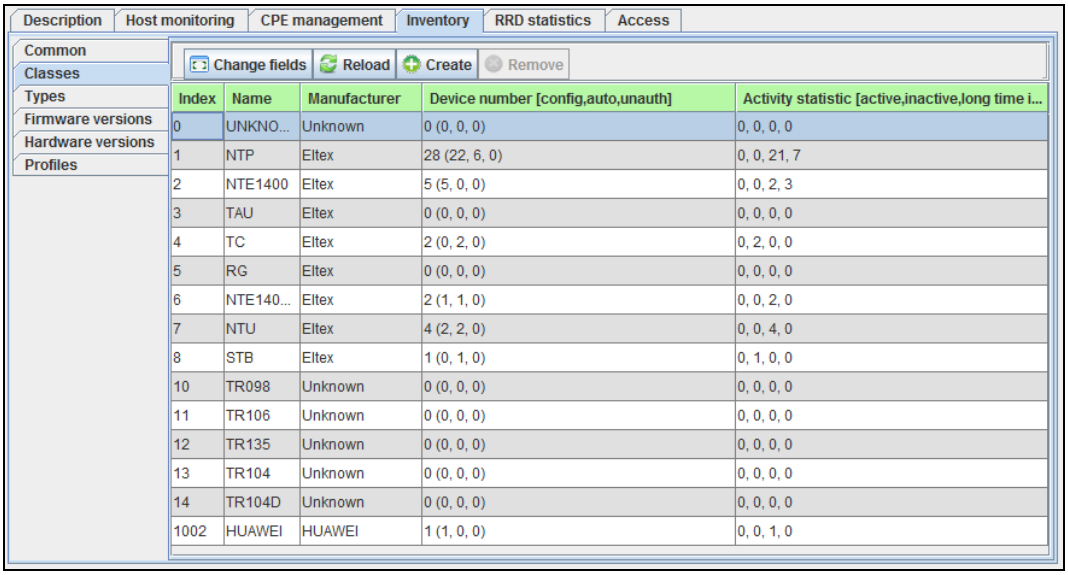
Use the *'Inventory'* tab to view the general statistics for devices on the server sorted by different criteria.

5.4.1. COMMON



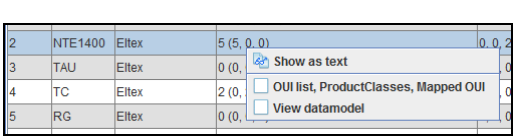
- *Total device number* — total device quantity in the system.
- *of them* — particular configuration statistics:
  - *are configured* — CPE parameters has been edited by the operator.
  - *be added automatically.*
  - *not authorized* — not authorized on server.
- *Activity statistics* — particular activity statistics:
- *active.*
- *inactive.*
- *long inactive.*
- *not determined.*

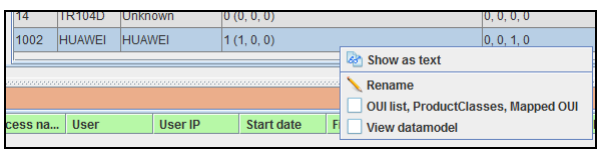
5.4.2. CLASSES



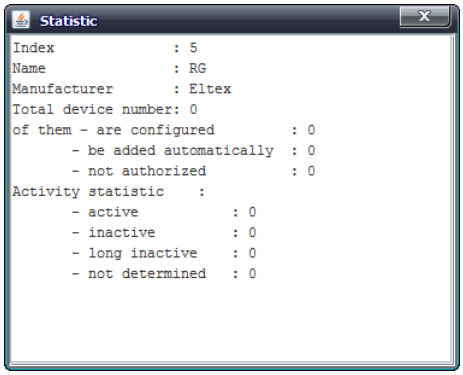
Use this tab to view the list of the existing device classes, manufacturer information, number of devices of each class and activity statistics.

Right-click the table row to access the class operations menu.

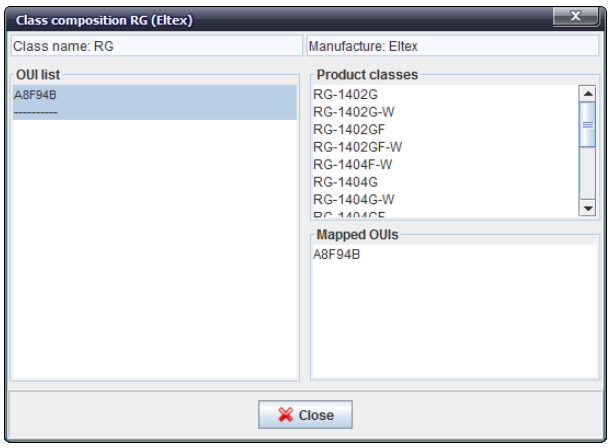




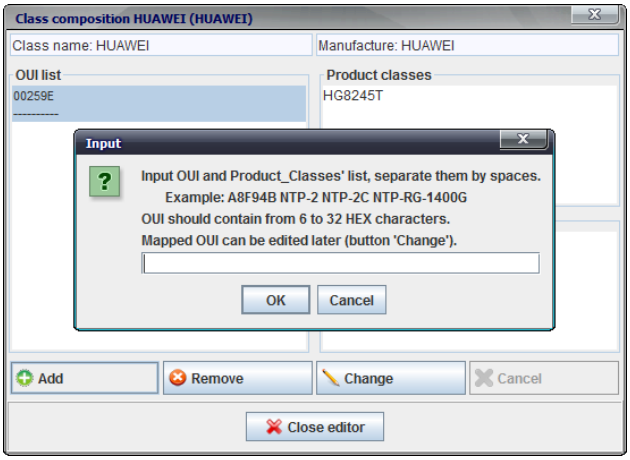
- *Show as text* — show device class statistics in the simplified text view (also, you can access the menu by double-clicking the table row).



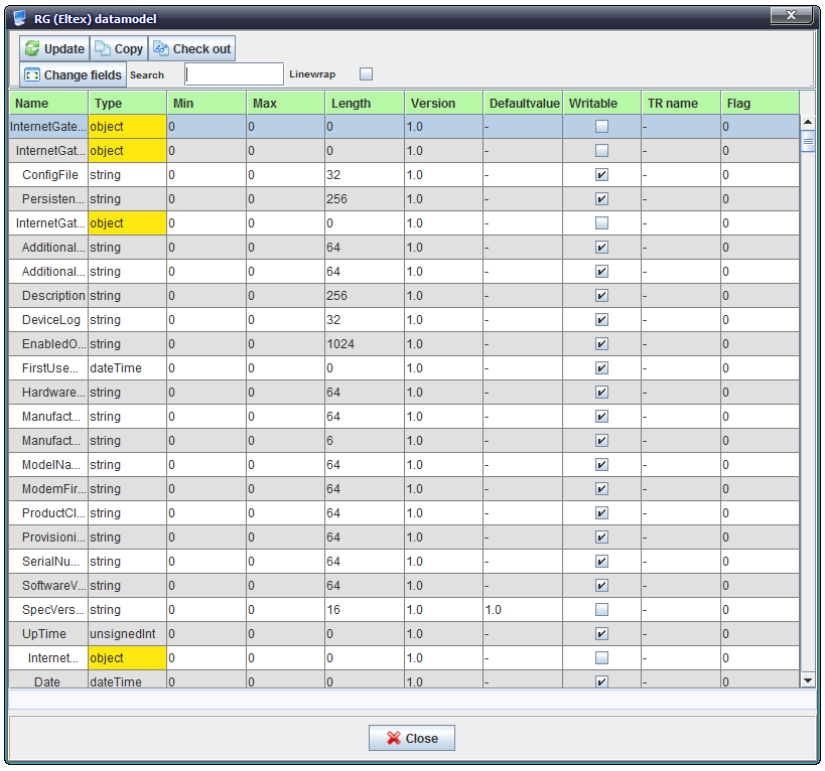
– *Rename* — rename the class. This operation is available only to those classes that has been created manually by the operator.
– *OUI list, ProductClasses, Mapped OUI* — view and edit (for classes that has been created manually by the operator) class contents.



To create/edit parameters of a manually created class, click the respective buttons and enter *OUI* and *ProductClass* delimited by spaces in the opened window.

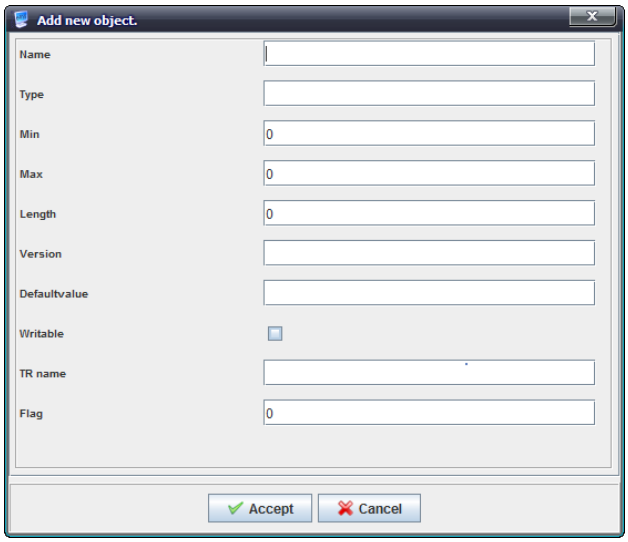


– *View datamodel* — view and edit (for classes that has been created manually by the operator) datamodel.

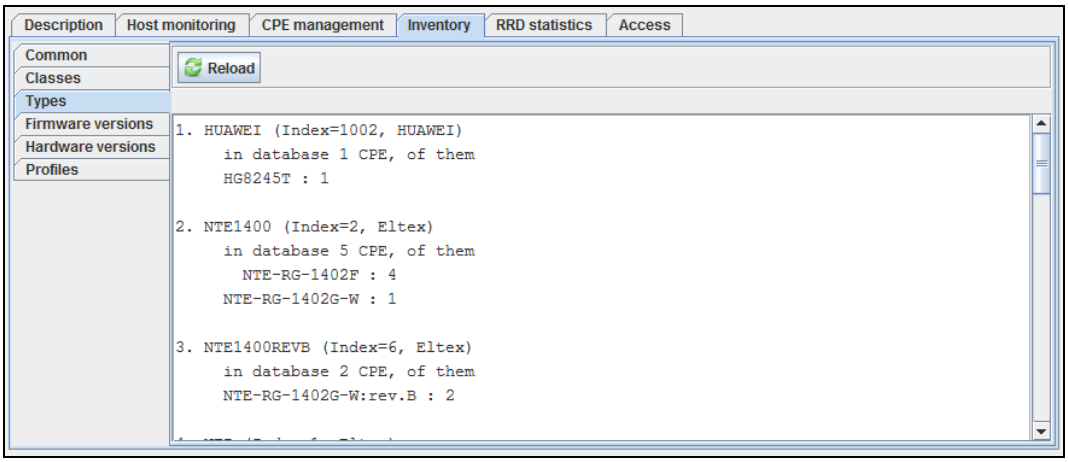


You can edit the following parameters:

- *Name* — parameter name.
- *Type* — type.
- *Min* — minimum value of a parameter.
- *Max* — maximum value of a parameter.
- *Length* — field length.
- *Version* — version.
- *Default value* — default value.
- *Writable* — read/write.
- *TR name* — TR parameter name.
- *Flag* — flag.

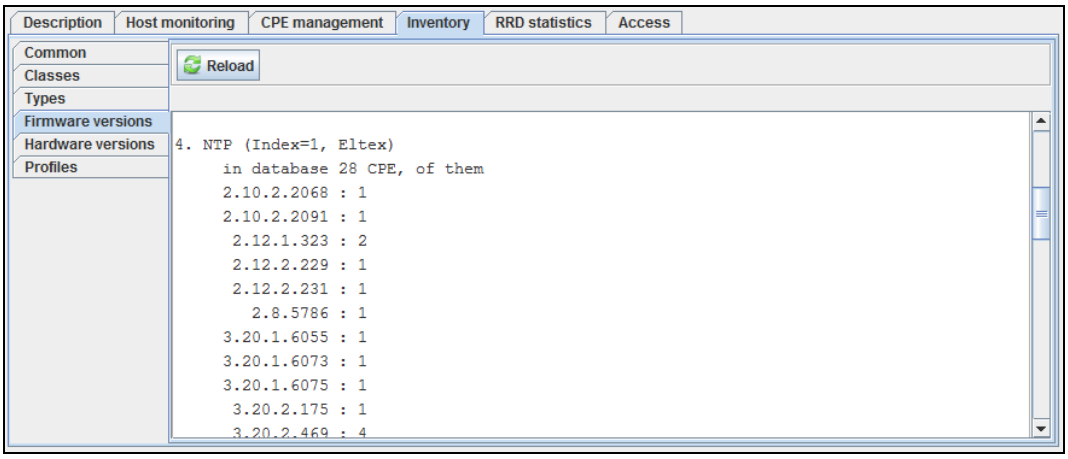


5.4.3. TYPES



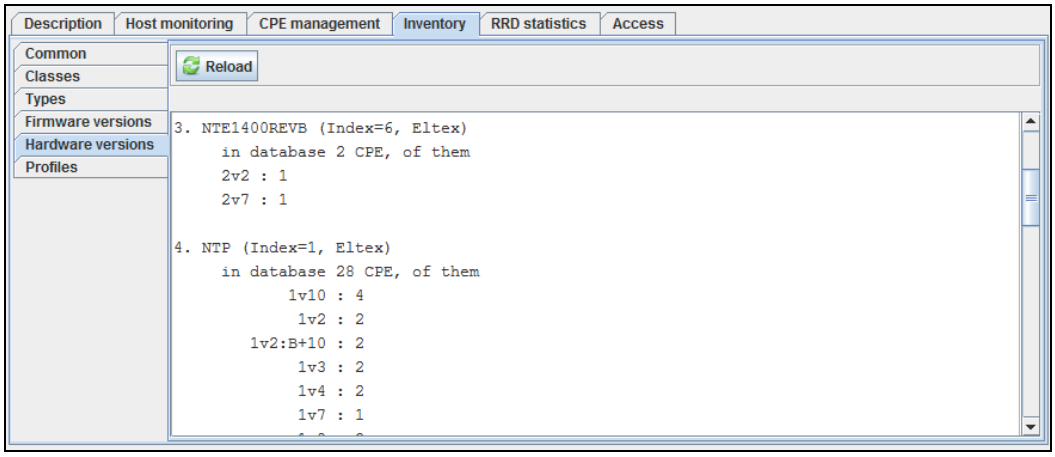
This tab shows the list of devices by classes and the quantity of devices of each type.

5.4.4. FIRMWARE VERSIONS



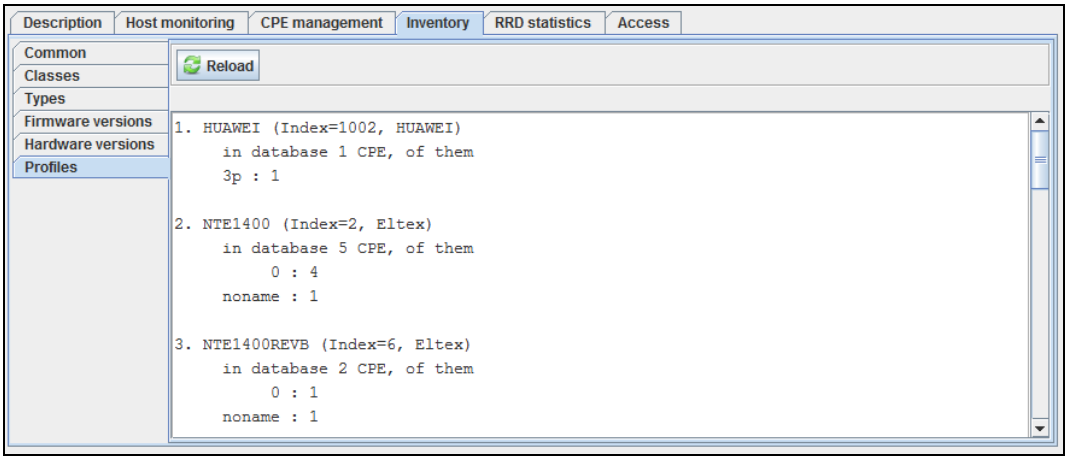
This tab lists the information on the existing firmware versions on the network and number of devices with such firmware version sorted by classes.

5.4.5. HARDWARE VERSIONS



This tab lists the information on the existing hardware versions and number of devices with such hardware version sorted by classes.

5.4.6. PROFILES

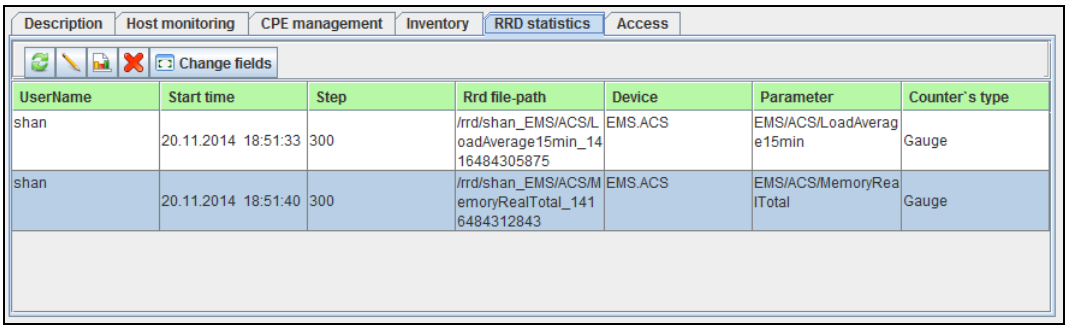


This tab lists the information on the existing profiles and number of devices with such profiles sorted by classes.

5.5. RRD STATISTICS

This menu allows you to configure the collection of network interface load statistics. The data is output in graphics/tabular format.

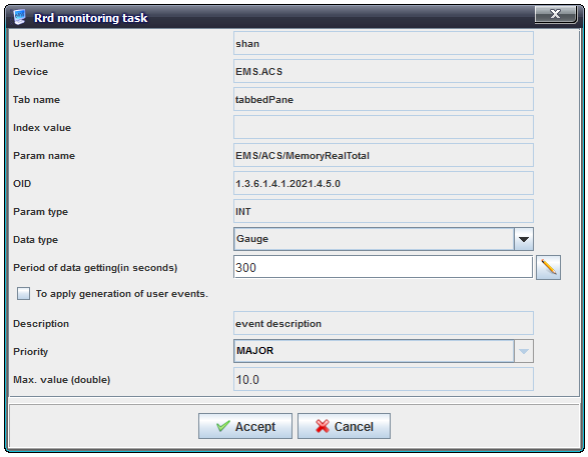
Round-robin Database (RRD) is a database, where the amount of stored data remains constant over time. As the number of records remains constant, they are used in cycles when data saving is performed. As a rule, such databases are used for storing information that is rewritten in regular periods of time.

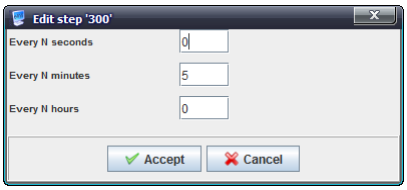


UserName	Start time	Step	Rrd file-path	Device	Parameter	Counter`s type
shan	20.11.2014 18:51:33	300	/rrd/shan_EMS/ACS/LoadAverage15min_1416484305875	EMS.ACS	EMS/ACS/LoadAverage15min	Gauge
shan	20.11.2014 18:51:40	300	/rrd/shan_EMS/ACS/MemoryRealTotal_1416484312843	EMS.ACS	EMS/ACS/MemoryRealTotal	Gauge

For adding monitoring tasks, use *Host monitoring* section, *System, Network* and *Disk* tab. Settings available for monitoring are marked with  button located to the right of the entry field. Click this button to open Add Task dialog window or go to the record with existing task.

Task editing is performed by clicking  .

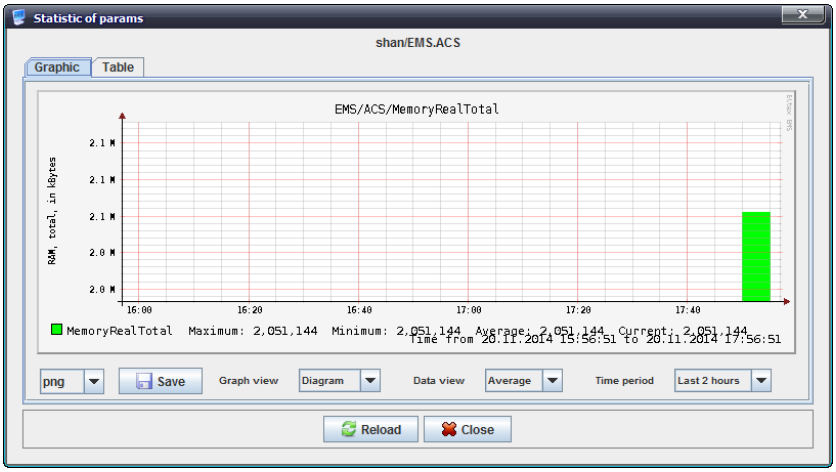




You can edit the following settings:

- *Data type* — select the type of saved data: absolute or increment (difference between values).
- *Period of data getting (in seconds)* — set the polling period in seconds. If polling values are great, you can use the additional editing menu that is available by clicking the button to the right of the entry field. There you can set the polling period in hours, minutes and seconds. For example, every 1h 30min (0 sec), every 2h (0min, 0 sec), every 2min 30sec. At that, the value entered into the field will be automatically converted to seconds.
- *To apply generation of user events* — when checked, the following settings will become available:
  - *Event description* — arbitrary text description.
  - *Priority* — select the event priority from the drop-down list.
  - *Max. value (double)* — maximum value of the monitored parameter; if exceeded, user event will be generated with the defined priority.

Data gathered by the task are shown by clicking  .



The chart explicitly shows time dependence of the measured parameter. You can adjust the chart type (diagram or line chart), data type (average or maximum), and displayed time period (from the last two hours to a week) with the corresponding drop-down lists below the chart.

The table lists measured parameter values for each point of time according to the polling period.

You can save the resulting chart into a file — just select its extension and click *Save* button.

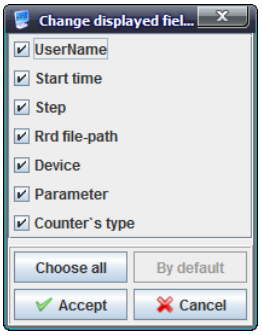
Available extensions:

- Bmp.
- Gif.
- Jpeg.
- Jpg.
- Png.
- Wbmp.

Click *Reload* button to refresh the information.

5.5.1. EVENT TABLE CONFIGURATION

Click *Change Fields* button to configure the set of fields for the event table.



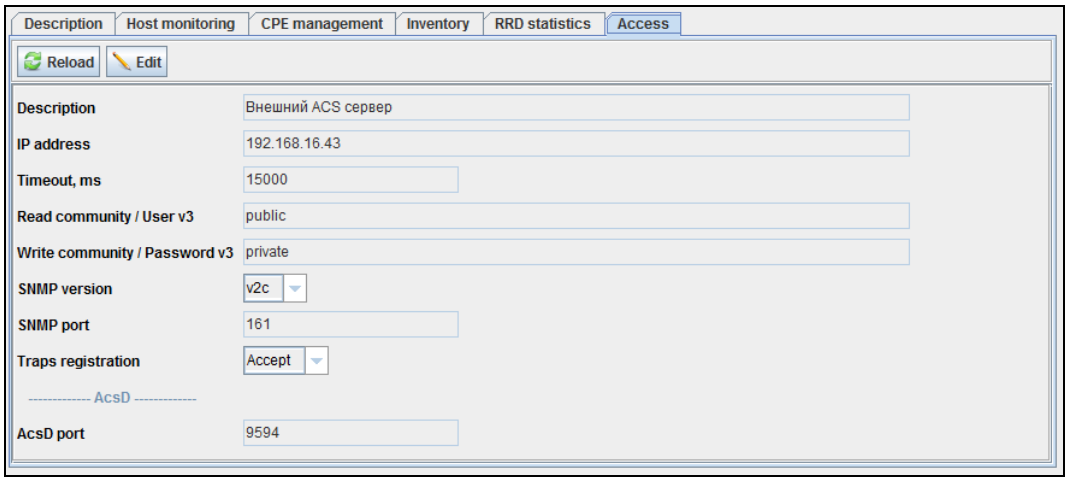
List of displayed fields:

- *User Name* — record identifier.
- *Start time* — record creation date.
- *Step* — priority of the occurred event.
- *Rrd file-path* — path to statistics output file.
- *Device* — name of the device, that statistics is gathered for.
- *Parameter* — monitored parameter.
- *Counter's type* — absolute or increment.

5.6. ACCESS

This tab contains general settings for data exchange between the ACS core and GUI. Click *Edit* button to make them available for editing. All settings are editable on this tab, except for the '*Icmp Ping delay, ms*'.

The following options are essential for gaining access: *Read Community, Write Community*. These settings should be confirmed by the network administrator or checked against the SNMP agent configuration file.



- *Description* — text description that facilitate identification by the user.
- *IP address* — device IP address.
- *Timeout, ms* — timeout of data exchange with the device.

We do not recommend setting the timeout value lower than 5000ms.

- *Read Community/User v3* — password for read access, for SNMP v3 — user login.
- *Write Community/Password v3* — password for write access, for SNMP v3 — user password.
- *SNMP version* — SNMP protocol version (supported versions: v2c, v3).
- *SNMP port* — device IP port number for data exchange via SNMP.
- *Trap registration* — trap registration mode:
  - *Accept* — system will generate traps received from devices.
  - *Block* — system will not generate or show traps.
- *AcsD port* — web service port number.

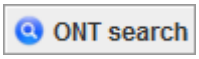
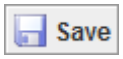
To accept or save changes made to parameters, click the corresponding buttons — 'Accept' or *'Save'*.

Click *'Reload' button to refresh the information in the tab.*

6 CONTROL PANEL

The control panel is located in the upper part of the interface. It allows to create and edit .js scripts, short parameter templates for CPE search and some system operations. Table 1 lists description of the control panel basic controls.

Table 1. Controls

Designation	Button name	Description
Shortcuts		
	*Synchronize*	Menu is not used.
	*ONT search*	
	*Save*	
	*Apply*	
Editing object tree		
	*Add*	Add object into current tree node. For detailed description, see Section **5 Operations with ACS**
	*Remove*	Remove current object or node.
	*Re-read*	Update the tree (the tree is completely re-read from the database)
Applet	**User application configuration**	
	*Authentication [locking]*	Block or unblock the applet with a password
	*Session data*	Show the window with the current user session information
	*Decor*	Applet theme configuration
	*View*	Configure the appearance of the elements
	*Pattern of utilities running*	Edit startup templates for ping, ssh, web, telnet utilities
	*Save applet settings*	Save current applet size and location on the screen
	*Exit*	Close the applet (terminate the current user session)
Devices	**Menu is not used.**	
Management	**Menu is not used.**	
OLT	**Menu is not used.**	
ONT	**Menu is not used.**	
ACS	**Edit scripts and search for devices in node**	
	*Search for subscriber's devices (CPE)*	CPE search in the current node. For detailed description, see chapter **9 DEVICE SEARCH**
	*Editor of embedded scripts (Java Script)*	Show the script editor
	*Setting common script of all classes*	Configure general script for all device classes

	Setting xml-templates of device option	Show CPE short parameter editor
	Setting xml-templates of device monitoring	Show CPE monitoring parameter editor
Events	**Menu is not used.**	
Utilities	**System utilities, duplicates the pop-up menu of the object tree**	
	*Run PING from the user's PC to the device*	Perform echo test from the user's PC to Eltex.ACS server
	*Run PING from the server to the device*	Perform echo test from GUI to device
Administration		
	Rights and users (Menu is not used.)	
	GUI behaviour (Menu is not used.)	
	Server configuration (Menu is not used.)	
	*SNMP traps receiving and processing*	Menu is not used.
	*Scheduled tasks (monitors)*	Menu is not used.
	*System modules settings*	View and edit module parameters
	*Administrator's workstation*	Go to Administrator Automated Workstation menu
	*EMS server restart*	Reboot the EMS server
	Device firmware (Menu is not used.)	
Information		
	*State of backup system*	View the state of the reservation system
	*Information about system components*	View the EMS system state
	*User actions log*	View user activity log
	*System notification of users*	Send messages to all users connected to the system at the moment
Help	**HELP INFORMATION**	
	*About*	Information about Eltex.EMS software and supported devices
	*License*	Information about used modules and effective license restrictions
	*Revision history*	Short changelog

7 ADMINISTRATION RIGHTS AND USERS CONFIGURING USERS AND ROLES[1]

7.1. PRINCIPLE OF USER RIGHTS' DISTRIBUTION

Role mechanism is used as a basic principle of rights' distribution. Role is a logical entity, that contains the following data:

- Role name
- Text description
- Idle time (seconds)
- List of permitted actions with objects:
  - Edit properties in a tree
  - Add an object into the tree
  - Remove an object from the tree
  - SNMP SET (modify and save)
  - SNMP RESTART (reboot device)
  - Firmware update (FW)
  - Configuration update
  - Edit privileges and roles (delegate system administrator privileges)
  - Edit ONT passwords
- List of permitted nodes and objects
- Alarm registration rights:
  - Info
  - Warning
  - Minor
  - Major
  - Critical

The system has one basic administrator role, named 'SuperUser'. This role is disabled for editing. It automatically has all rights for each object.

All other roles are configured by the administrator according to operator job duties and logical breakdown by devices or locations.

System user is a logical entity, that is designed for authorized logging into system. Each user has the following set of parameters:

- Name
- Description
- Role
- Password
- Account expiration date
- Email address
- Forwarding email messages to the user address
- User blocking

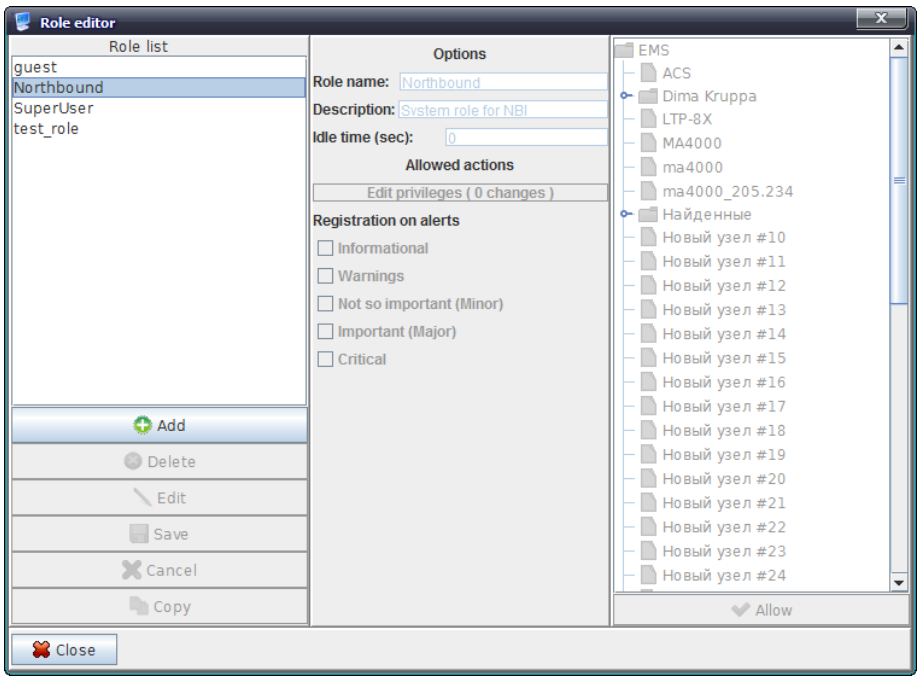
When creating a new user, you have to complete all available fields. The name and password are required for log in (authorization), the role describes the list of permitted actions, and the account expiration date defines the account lifetime and is checked upon each authorization attempt.

7.2. CONFIGURING ROLES

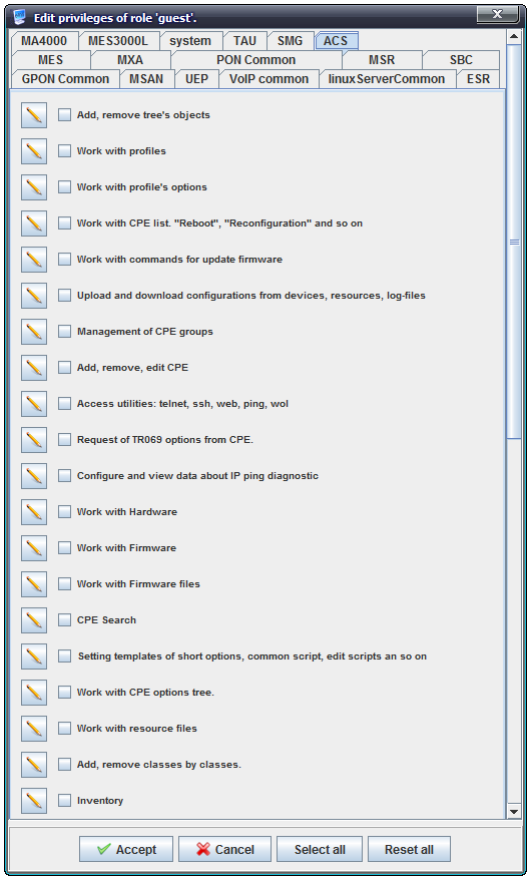
Configuration of roles and users is available for system users with *Edit rights and roles* rights. To add or edit roles, go to ***Administration/Rights and users/User role configuration*** menu item. When this menu item is

[1] Not used in the current firmware version.

selected, the application will give show the dialog window where you will be able to edit roles (except for the SuperUser system role), and also add or remove them.



There is a list of permissions for each role:



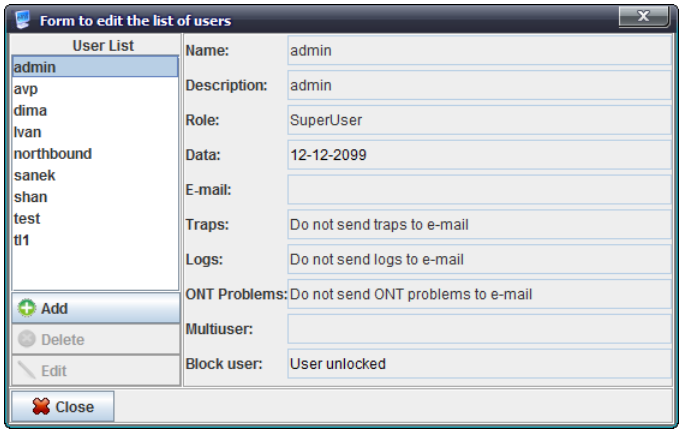
- *Add, remove tree objects*
- *Profile operations*
- *CPE list operations*
- *CPE list operations, 'Reboot', 'Reconfiguration', etc.*
- *Commands to upload and download configurations from devices*
- *Upload and download configurations from devices, resources, log-files*
- *CPE group management*
- *Add, remove, edit CPE*
- *Access utilities: telnet, ssh, web, ping, wol*
- *Actions for retrieving some data from ONT*
- *Configure and view IP ping diagnostics data*
- *Hardware operations*
- *Firmware operations*
- *Firmware file operations*
- *CPE search*
- *ACS configuration*
- *Operations with tree-like CPE parameters*
- *Resource file operations*
- *Inventory*
- *Scheduler*
- *Data model operations*

In addition to rules defined for each role, you have to specify the scope of effect for these rules. To do this, edit the role and select '*Enable*' checkbox against the respective nodes in the right part of the role configuration dialog window. If you enable access to a node for this role, all nested nodes and objects in this node will become available automatically. To enable full access to the tree, you should give permission to access the root node *RootNode*.

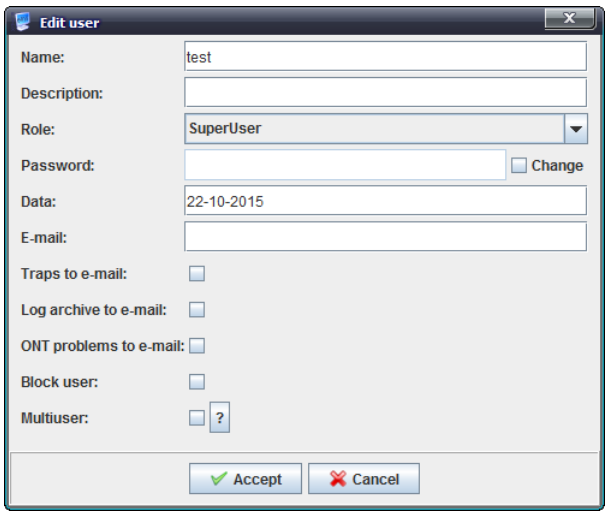
The application stores previously given permissions, and they are saved when these nodes being merged with the higher level nodes. Note this, when revoking permissions. Also note, that the application will not let you to delete the role, unless it belongs at least to one user.

7.3. CONFIGURE SYSTEM USERS

You have to enter your account name and password in order to login. When user authentication is completed, you will see the dialog window with the list of permitted actions and nodes or the login error message. ***You cannot operate the system without registration*** Configuration of user rights is performed by the system administrator (admin) or another user with the respective rights.



To add or edit users, go to ***Administration/Rights and users/System users configuration*** menu item. If you choose this menu item, the application will show the user edit dialog window. System user **admin** cannot be deleted or renamed. Also, you can't change its expiration date or password. You can define the following parameters for other users:



- *Name* — arbitrary name, up to 32 characters
- Description — arbitrary description, up to 64 characters
- Role — role, that defines access rights
- *Password* — arbitrary alphanumeric password
- *Date* — user account expiration date
- *E-mail* — e-mail address for sending alarm messages
- *Send traps by e-mail* — when checked, send e-mail messages to the defined address, otherwise — do not send.

- *Send logs by e-mail* — when checked, send e-mail messages to the defined address, otherwise — do not send.
- *Multiuser* — mode, that allows authorization of multiple users with the single login In this mode you can define approved IP addresses for the user. If user performs authorization from one of these addresses, the password will not be prompted. Addresses should be delimited with space or comma. Validation of addresses is not performed. Field size limit — 255 characters.

If the address list database doesn't exist, this mode considered to be disabled.

Password is stored encrypted in the database, thus the system administrator will not be able to acces this information.

***'Edit'* checkbox next to *'Password'* field allows you to change passwords. If you need to change the password (or to define it for the first time), select this checkbox and fill in the *'Password'* field. Otherwise, if you edit other parameters with this checkbox unselected, the password will not be changed. This feature allows the system administrator to avoid entering user password while changing other parameters of the account. Default password for 'admin' account — empty field.**

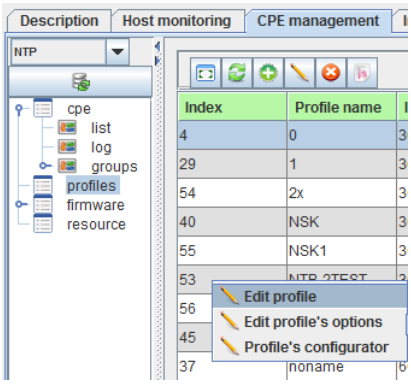
After the user account expires, the access to the system with this name will be blocked. System administrator can modify the expiration date or delete the account.

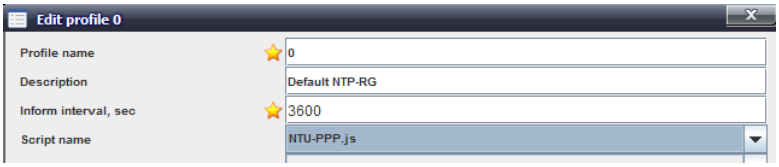
8 OPERATION WITH SCRIPTS

8.1. SCRIPTS

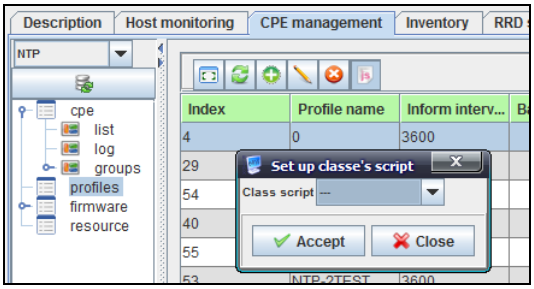
ACS server may be configured with scripts written in JavaScript. To configure CPE using the script, you have to assign it to:

– Profile:

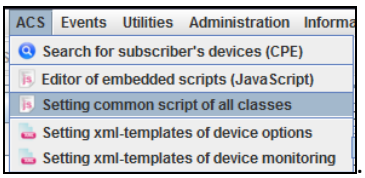




– Class:

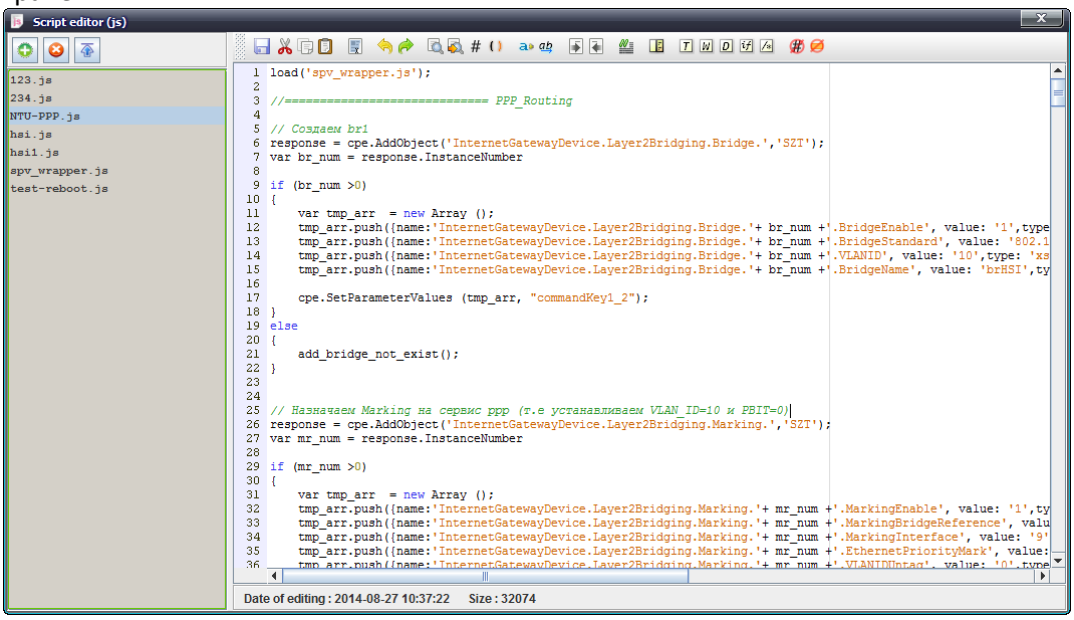


– All devices on server:



. .

To proceed to script creation and editing, use the *'ACS/Editor of embedded scripts (JavaScript)'* menu in the control panel.



8.2. FUNCTIONS

8.2.1. 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.
```

8.2.2. EXEC

Exec function allows to execute an additional script from the current script. Both scripts should be in <datadir>/scripts/ directory.

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

8.3. OBJECTS

8.3.1. 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);
```

8.3.2. 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 boolen;
- *fw_downgrade* — enable firmware downgrade, int_value boolen;
- *noautoconfig* — disable configuration of the following rules: property, int_value boolen.

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);
}
```

8.3.3. 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.URL = 'http://10.255.240.200/test/config.txt';
  dlcmd.Username = 'testuser';
  dlcmd.Password = 'testpass';
  dlcmd.FileSize = 0;
  dlcmd.TargetFileName = ' config.txt';
  dlcmd.DelaySeconds = 0;
  dlcmd.SuccessURL = 'http://ya.ru';
```

```
  dlcmd.FailureURL = 'http://yandex.ru';
cpe.Download(dlcmd);
log('Download response: Status ' + response.Status + ', StartTime ' +
response.StartTime + '; CompleteTime ' + response.CompleteTime);
```

Upload (object_of_parameters) — upload a file 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. 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);
```

SetParameterAttributes (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 writeable 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 reboot command.

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

FactoryReset — command will reset CPE configuration to factory defaults.

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

Script example

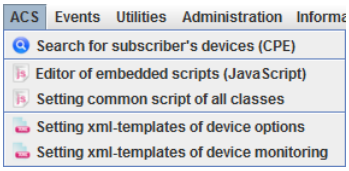
test.js:

```
/* 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';
```

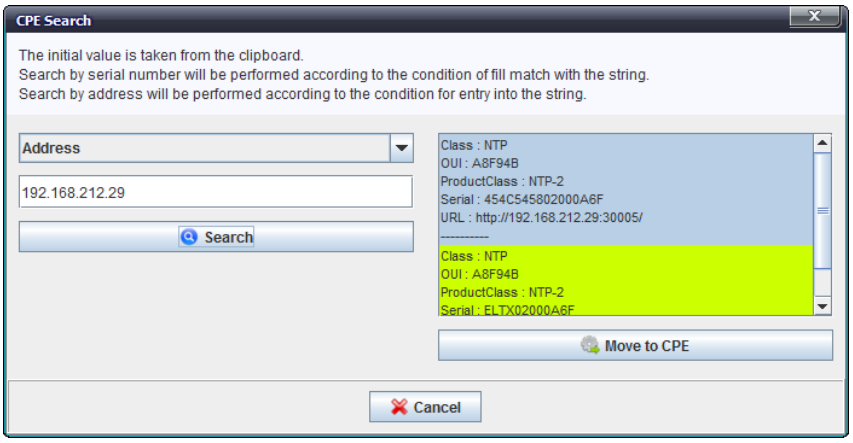
```
var SESSION_TRANSFER_COMPLETE = false;
var SESSION_BOOT = false;
d += '\nEvents:\n';
for (let i = 0; i < cpe.Inform.Event.length; i++) {
   d += cpe.Inform.Event[i].EventCode + ' ' + cpe.Inform.Event[i].CommandKey + '\n';

   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) {
   log('error', 'error on GPN: ' + 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';
   dlcmd.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();
```

9 DEVICE SEARCH



If you don't know which class the device belongs to, you may search for it by the serial number or IP address.



The result will contain CPE list matching the specified criteria accompanied by the short description. To go to CPE, double-click its records or click the *'Move to CPE'* button.

TECHNICAL SUPPORT

Contact Eltex Service Centre to receive technical support regarding our products:

29v Okruzhnaya ul., Novosibirsk, Russian Federation, 630020
Tel.:
+7(383) 274-47-88
+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.

http://eltex.nsk.ru/en/
http://eltex.nsk.ru/en/support/downloads/
http://eltex.nsk.ru/forum/index.php
http://eltex.nsk.ru/en/support/knowledge/