Network Manager 3.0.0 (console version)

Technical documentation

Katarzyna Wladyszewska, Hadden Sp.J.

Network Manager 3.0.0 (console version): Technical documentation

by Katarzyna Wladyszewska

Published April 2010

Table of Contents

List of Tables

Chapter 1. Conventions

The following typographical conventions are used in this manual:

Table 1.1. The typographical conventions used in this manual

Font	What the font represents	Example
Italic	Environment variables.	The name is kept in environmental variable *$DAVIDPRIVDIR*...
Italic	Synopsis options.	[*-l,--log-facility log_facility*]
Bold	Names of programs and products.	**damcsud** is a part of **Operation Manager-a**.
`Computer`	Names of options and menus.	There is `Show tool bar` option in `View` menu.
`Computer`	Names of files and directories.	... reads its configuration file `.damadbudrc`.
`Computer`	Names of windows and dialog fields.	In `A sessions property` window, in `Sticking string` field, you can write...
`Computer`	Names of buttons.	Pressing `Apply` button lets you apply changes.
`Computer Bold`	Math formulas.	**`exp( -x ), when a = 0 1 / pow( a , a ) * pow( x , a ) * exp( -x + a ), when a > 0`**.
`Computer Bold`	Terms used in David system terminology.	**`SNMP Data`** - a kind of data...
`Computer Bold`	Contents of configurations files.	**`action`** **`{`** **`...`** **`}`**

Chapter 2. General information about David system

2.1. General

David systemis a network management system. It is a packet of applications (modules) that allows computer network to be monitored and managed in real-time through the Internet. There is only one condition that managed devices must meet. Each device must provide SNMP (Simple Network Management Protocol) service. SNMP is the most common management protocol in the Internet so that requirement shouldn't be difficult to meet. Here is the list of typical devices that can be monitored:

- IP routers,
- ATM switches,
- manageable ethernet switches,
- UPSes with a SNMP adapter,
- TV-SAT modems that allow IP devices to work in TV cable networks,
- computers.

One of the most important feature of **David system** is its architecture. It's built of high level configureable and independent from one another modules. This principle is the most essential rule of the project. In consequences, in th metter of speaking, the same modules may build different management system. Here are the main features of **David system**:

- general thinking in information flow controlling that come form high level independence of modules of the system,
- high level configureability of the system modules that allows a special configuration of **David system** to reach end-user expectations so close as it's only possible,
- the system scalability, so you can build up the system adding additional modules in very easy way; note that these modules needn't to be part of **David system** at all; adding another monitored devices to the system is a very easy procedure,
- using shell scripts in information processing is opportunity for modeling information and influence on processing it,
- all configuration files of **David system**, files with input/output data and log files are text files,

- using SNMPv1, SNMPv2C and SNMPv3 to communicate with monitored devices.

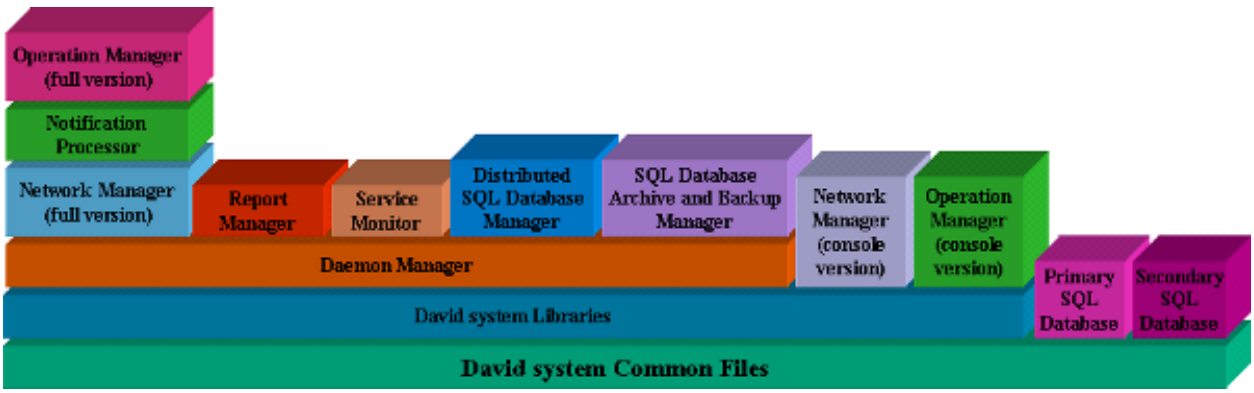
2.2. David system architecture

Table 2.1. David system products

Product	Description
David system Common Files	The product, during its installation, prepares the rudimentary directory tree for other products of **David system**. It also contains some essential and common files for all the products. Thus, this is a fundamental product of **David system** required by other its products.
Primary SQL Database	The product installs the primary SQL database of **David system**. Every single installation of **David system** must have only one the primary database.
Secondary SQL Database	The product installs the secondary SQL database of **David system**. Each installation of **David system** may have many secondary databases or none. It allows to distribute the SQL database of **David system** among many servers.
David system Libraries	This product provides libraries of **David system** required by its applications. Many other products of **David system** require that one.
Daemon Manager	It engages in running and terminating daemons of **David system** as well as monitoring of their work.
Network Manager (full version)	The product using SNMP protocol allows to visualise a topology of monitored networks and auto-discover devices in managed networks. The state of monitored devices also is visualized. The product also collects data from monitored devices using SNMP protocol and allows you to manage user accounts.
Network Manager (console version)	The product, through a graphic application, allows to visualize a topology of monitored networks and shows states of monitored resources. It allows you to control daemons monitoring devices as well as that ones gathering data. Currently, most of functions of that application is obtainable through web applications.
Notification Processor	The product chiefly engages in processing SNMP Trap notifications coming from monitored devices to management stations. The received messages can be formatted to the human readable forms, and then recorded as well. The processed notifications can be passed on to future processing.
Operation Manager (full version)	It can run specified actions on the basis of received data. Sophisticated estimation process depends on information coming from other products of **David system** and correlation of that information. It tries to build more intelligent and useful notifications then just simple reactions to incoming

Product	Description
	events. The graphic application displays notifications about received events and allows to play audio files as well as reading messages by an outer speach synthesizer.
Operation Manager (console version)	The product contains a graphic application displaying notifications about events and allowing to play audio files as well as reading messages by an outer speach synthesizer.
Report Manager	The product processes recorded SNMP Trap notifications, entries about pending operations and entries about state changes of monitored devices (ping objects, network interfaces and BGP peers), and generates raports on the basis of them. Raports can be viewed using a Web application.
Service Monitor	The product monitors selected network services on application level. In order to do this it monitors selected TCP ports of specified hosts. It checks both availability of ports and a correct reaction for a few selected network protocols (HTTP, SMTP, FTP). It also can verify correctness of work of selected services by verification of received data. Results of its work can be viewed as reports and graphs made available by a Web application.
SQL Database Archive and Backup Manager	It archives the SQL Database used by **David system** applications.
Distributed SQL Database Manager	It allows to devide the database of **David system** into one primary database and many secondary ones. Such step boosts performance of the system and decreases load of the servers where daemons of **David system** work. The migration takes place during the rutine work of the system. Such division may be altered many times.

Dependences between the **David system** products are shown on the following chart:.



David system functionality can be very large and it depends on particular configuration a lot. The most important features of **David system** are:

- discovering and visualization of monitored networks topology including visualization of states of

particular nodes;

- possibility of building control panels to monitored devices (they must support SNMP protocol), regardless of device providers;
- formatting and recording SNMP Traps sent by agents working on monitored devices;
- automatic reaction to specified SNMP Traps received from monitored devices;
- possibility of identification of an operator that has received an alert from the system about a problem;
- collecting data concerning parameters of monitored devices;
- automatic reaction to incorrect values of data that were found during data collecting;
- recording pending cases, processed by the system, which have been created as responses for events detected by the system in a monitored network;
- monitoring selected network services on application level.

Chapter 3. Terminology

3.1. Authorization process made by David system products

The modules of David system which need to do an authorization of message senders (i.e. **damsnmpdaud**, **dnmmsd**, **dgnsd**), use the library, that checks whether an IP address of a sender matches with any record found in the file `.known.host`. The library expects to find the file in a directory pointed by a variable *confdir* in the file `/etc/system-david.conf`.

Records in the file `.known.host` are regular expressions specifying acceptable IP addresses.

3.2. David system terminology used in the documentation

There is an explanation of some terms, that are used in David system and its documentation:

- **`massages (information)`** - data received by interfaces of **Operation Manager**, its data analysers and **Cases Database Unit** of the product.
- **`notifications`** - the term often is used in the products: **Notification Processor**, **Operation Manager** and **Report Manager**; There are mostly data, that a source are SNMP agents working on network monitored devices.
- **`events`** - the term often is used in the products: **Operation Manager** and **Report Manager**; and it describes a being, that a source is SNMP Trap or SNMP Data; an **`event`** is always a part of a **`case`**;
- **`cases`** - the term often is used in the products: **Operation Manager** and **Report Manager**; and it describes a group of events connected one another; one **`event`** at last must be included in a **`case`**;
- **`SNMP Trap`** - a kind of data of **Operation Manager** product, which a source are received responses from SNMP agents; SNMP Traps aren't answers on the requests sent by a management station, but they are sent by agents managing network interfaces and processed by **Notification Processor** product;
- **`SNMP Data`** - a kind of data of **Operation Manager** product, which a source are received responses from SNMP agents on request which a management station sent to them by **Network Manager**.

Chapter 4. Installation

4.1. The main configuration file of David system

The essential configuration file of David system in `/etc/david-system.conf`. It contains entries as pairs: key = value. Basically, except the entry `default_email_recipient`, there is no such need to modify any record in that file. All necessary modifications are made during installation processes of particular David system products. Below, there is a list of all entries along with their descriptions that may occur in this basic configuration file.

- `user` - a name of the user with which rights all daemons of David system works;
- `default_email_recipient` - the default e-mail address where messages from David system applications are sent;
- `bindir` - the directory containing David system applications (default: /usr/bin/david-system);
- `libdir` - the directory containing David system libraries (default: /usr/lib/david-system);
- `incdir` - the directory containing David system headers (default: /usr/include/david);
- `confdir` - the directory containing David system configuration files (default: /etc/david-system);
- `logdir` - the directory containing log files of David system applications (default: /var/log/david-system);
- `sharedir` - the directory containing various files (images, audio files, web files) of David system (default: /usr/share/david-system);
- `docdir` - the directory containing various files (images, audio files, web files) of David system (default: /usr/share/david-system);
- `vardir` - the directory containing archive files of David system SQL database (default: /var/lib/david-system);
- `is_sqldb_installed` - the flag that indicate whether the SQL database of David system has been installed or not.

4.2. Dedicated account for service of David system

There is no needs to run any David system module as superuser (usually an account `root` with UID equals 0). Even if some David system daemon requires root rights when starting, there is always possibility to specify, as one of the daemons starting arguments, a user that rights should be taken.

It is a good idea to add a new user to an operating system, under which control David system will work.

4.3. Directories of David system

This hierarchy depends on a particular configuration of David system. In the default system configuration, David system contains the following directories:

- `/usr/bin/david-system` - binaries and shell scripts;
- `/etc/david-system` - configuration files;
- `/usr/share/doc/david-system` - the documentation;
- `/usr/share/david-system` - graphic and audio files, web portal;
- `/usr/include/david` - David system header files;
- `/usr/lib/david-system` - David system libraries;
- `/var/log/david-system` - log files;
- `/var/lib/david-system` - archive files of the David system SQL database;

4.4. Configuration of syslogd daemon

David system modules use `syslog` subsystem available on UNIX platforms. Default configuration of the system modules causes that log messages are sent with local6 `facility`. It may be changed for every module during its startup. Its recommended to configure `syslogd daemon` to write all messages from David system modules into one place (one or more files with characteristic name i.e.: david.log).

Chapter 5. Network Manager requirements

The following requirements must be met by a management platform where **Network Manager** works:

- 512 MB of RAM and 1 GB of swap memory at least;
- at least 1.5 GHz CPU;
- installed compatible version of **David system Libraries**.

Chapter 6. Installation

6.1. Installation from RPM package

You must be `root` to install the product. Following steps must be taken in order to install the product:

- Install the product:

```
rpm -i david-xxx-nm-c-yyy.rpm
```

6.2. Installation from the script

You must be `root` to install the product. Following steps must be taken in order to install the product:

- Uncompress and unpack the archive:

```
gunzip david-xxx-nm-c-yyy.i386.tar.gz
tar xf david-xxx-nm-c-yyy.i386.tar
```

  The operations create david-xxx-nm-c-yyy.i386 directory in your current directory.

- Change your current directory to david-xxx-nm-c-yyy.i386:

```
cd david-xxx-nm-c-yyy.i386
```

- Read LICENSE file and CONTINUE THE INSTALLATION, ONLY WHEN YOU ACCEPT ALL CONDITIONS INCLUDED IN THE LICENSE.

- Run the installation script:

```
./install
```

Chapter 7. Network Manager (NM)

7.1. Functionality

Network Manager makes possible:

- auto discovering of devices included in monitored networks and adding them to the managed device database;
- visualization of monitored network topologies according to data gathered from managed devices through SNMP protocol;
- monitored device states visualization (states of network interfaces, BGP sessions);
- visualization of responses of monitored devices for `ICMP ECHO` packets sent by the management station (similary as `ping` command);
- creation objects that represent service, node etc., which state depends on states of other objects and services stored in the database;
- running directly from presented topology maps the most suitable control panels for selected devices and other graphic applications of David system;
- designing graphic interfaces to control network devices that supply SNMP protocol; appearance of each interface and its functionality depend on an end-user invention (every control panel is written as a separate file);
- real-time monitoring (considering speed of computer networks and polling interval) of devices work parameters and controlling its work, through generated application - control interface to a particular device;
- visualization of network device work parameters through a generated application - interface to a particular device.

7.2. Description

One of the assigments of **Network Manager** is discovered devices belonging to managed networks, built topology maps of monitored networks and made visualization of discovered device states.

The product can gather information about networks from their nodes through SNMP protocol, and next it visualises the collected data through a graphic application which is a user friendly interface to the monitored network database.

Network Manager, on the base of its configuration files, can generate applications that are graphic interfaces that monitor work of devices on which SNMP agents work. Such interfaces usually show that parameters of devices, which are important in a given situation. Many interfaces to a particular device may exist at the same time (stored in separate files). The panels can be created and modified very easily by end-users. Main edition technique used to panels creation is, like in a different kind of graphic editors, using a mouse as main designed tool.

7.3. Related articles

Network Management Map (xdnmm)

Network Nodes Viewer (xdnnv)

Network Node Views Editor (xdnnve)

Chapter 8. Network Management Map (xdnmm)

8.1. General

xdnmm application is **Network Management Map (xdnmm)** and it is a part of **Network Manager**. It's a graphic client of dnmmsd daemon that services **Netowrk Management Map**, i.e. it polls devices in managed networks using SNMP protocol and ICMP (ping). The client is the comfortable interface and it delivers a visualization and makes commands on objects of **Network Information Database** that is wholly managed by dnmmsd.

8.2. Synopsis

xdnmm can be run with the following options: [*-l,--log-facility log_facility*] [*-L,--log-level log_level*] [*-v,--version*] [*-h,--help*]

8.3. Options

Table 8.1. xdnmm options

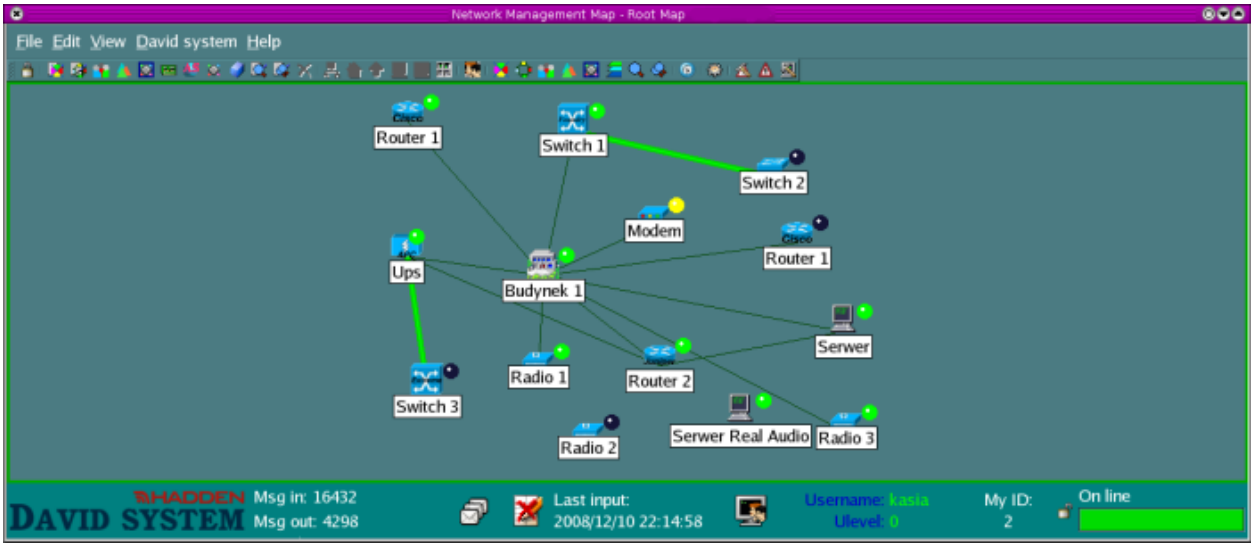
Option name	Description
-l,--log-facility log_facility	Choose log facility: daemon | user | local0 | ... | local7 (default: local6).
-L,--log-level log_level	Choose log level (on stderr and syslog) i.e. messages of selected level and more important levels will be logged: emerg | alert | crit | err | warning | notice | info | debug0 | ... | debug2 (default: warning).
-v,--version	Display version number on stderr and exit.
-h,--help	Display this help and exit.

8.4. Description

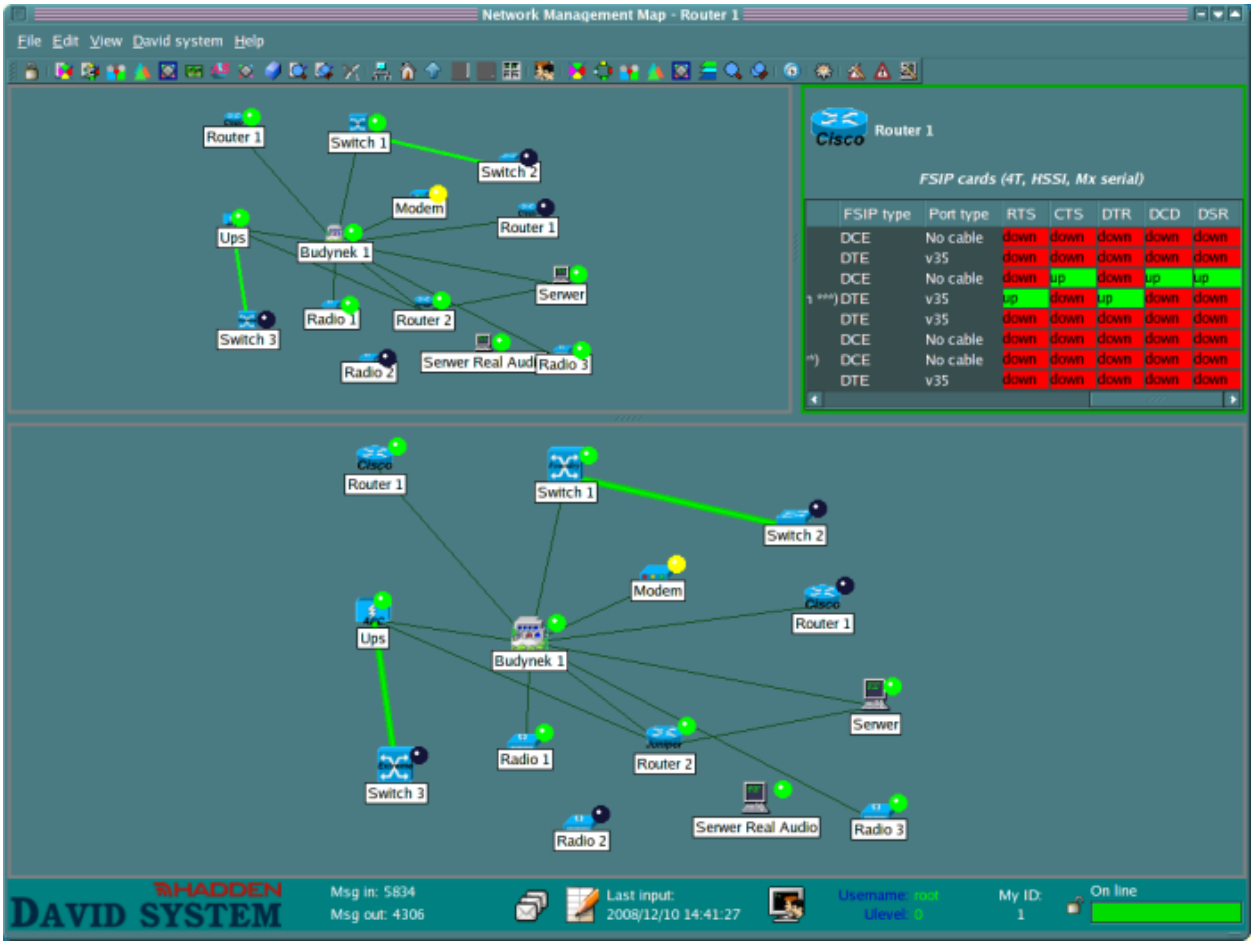
8.4.1. Starting up and terminating the aplication

xdnmm rads its configuration parameters from `.xdescrc` file during its startup. The parameters concern an appearance of the program and other working parameters. The application expects to find its configuration file in a directory which name is kept in the environmental variable **$DAVIDPRIVDIR**. When the file doesn't exist, the application will begin its work with its default settings.

8.4.2. Main window work



The application is a graphic client of dnmmsd server and in this connection it has have complete functionality after its connection process with the server. The connection will be realized when the application user identifies on the server passing his identifier and his password. After the successfully authentication, data of **Network Information Database** accessed through the server, are loaded. This process usually takes a little time. A progress of this process is visualized by the application. **xdnmm** has the complete functionality after ending of this process.



A main view of the application can be split into any number of horizontal and vertical panels. Each panel can include other object. Only one panel is always active. It is surrounded by a green frame.

At the bottom of the application view is the status bar. Two counters: incoming and outgoing messages are placed from the left side of the status bar.

There are also the buttons described below:

Table 8.2. xdnmm - the status bar buttons

Button	Description
	It fulfills some functions. Pressing the button in the state like that causes connection with the server when the application isn't connected with it yet, otherwise when you press the button, the application will be in the active mode to read only. It changes a button picture on the button below.
	When you press the button in this state, the application will be in the mode making possible modification of the Database items after successfully authentication process (you don't need to login as the same user). The button picture again comes back to the initial state.

Button	Description
	It allows you to connect with the server or login to the server again (the application is in the active mode).
	It shows if the application is locked.
	It shows if the application is unlocked.

Before the button that shows if the application is locked or not, three fields are placed. First and second field describes a logged in user's identifier and his level of rights. After them, there is the application identifier which is assigned its by the server. The last item is an indicator of a current connection status of the application with the server.

8.4.2.1. Main window buttons

On the toolbar are placed the buttons that allow you to control the application work. The first button from the left agree with option from `File` menu.

Table 8.3. xdnmm - the File menu buttons

Button	Description
	It allows you to lock access to the application. Then if you press any button of the keyboard or the mouse, or move the mouse, `Authorization` dialog will appear. In the dialog you should write a username and a password to unlock access to the application and login again to the server (you don't need to login as the same user).

In `File` menu also are options which aren't placed on the toolbar.

Table 8.4. xdnmm - File menu buttons which aren't placed on the toolbar

Button	Description
 - `Connect`	It allows you to connect with dnmmsd server to login.
	It lets you disconnect with the server (to logout).
	It lets you exit the application.

Next buttons agree with `Edit` menu. There are the buttons to edition of different objects and after pressing them, non-modal edition dialogs appear. In each case you can edit unlimited number of objects at the same time by suitable selecting of them.

Table 8.5. xdnmm - the Edit menu buttons

Button	Description
	It opens the window allowing to edit styles.
	It opens the window allowing to edit state groups.
	It opens the window allowing to edit communities.
	It opens the window allowing to edit the collection groups.
	It opens the window allowing to edit menaged items groups.
	It opens the dialog allowing to edit network interfaces of the current monitoring device. In this case you can't get a list of current available network interfaces in the Database because it doesn't exist.
	It opens the dialog, that allows you to edit BGP peers of the current monitoring device. In this case windows with current BGP peers in the Database don't exist. The device must be a `router` type.
	It opens the dialog, that allows you to edit managed items of the current monitoring device. In this case a dialog with all managed items doesn't exist.
	It lets you open a window to edit objects. You can choose objects to edition when you mark them on a current presented map. In this case you can also edit many windows at the same time.
	It opens a window that allows you to edit ping objects.
	It opens a window that lets you edit ping object group.
	It lets you delete selected item.

In `Edit` menu another options are placed which don't show on the toolbar.

Table 8.6. xdnmm - Edit menu options which aren't placed on the toolbar

Option	Description
`Reset to OK state`	It lets you change an item state on `OK` state. If an item is an object and includes others sub-objects, their state will be changed too with the exception of a state marked as `Not managed` or `Delete`.
`Manage`	It lets you to set a state of chosen items on `OK` state and begin a management of this items. If an item is an object of `Network` type, the process of its discovering (scan procedure) will be begun. If an object is `computer` or `router` type, its all network interfaces and some current BGP peers will be managed (i.e. their state on this device will be monitored).

Option	Description
`Manage but don't discover`	It complies with objects of `Network` type. Then a propagation service of item states will be run. Item states have an influence on these items but a scan procedure of a network won't be run. Non- scanned networks have a blue color.
`Don't manage`	It sets a selected object state on `Not managed` value.
`Move up`	It lets you move up selected objects to a map being higher in a map hierarchy, i.e. to an object including a current presented object.
`Discover (poll) node`	It allows you to discover a given device or poll it again if it is placed in the Database. In this way you can updated its configuration.
`Save positions of ping panels`	Save positions and sizes all panles, that show ping objects.

There are some buttons on the toolbar that agree with `View` menu.

Table 8.7. xdnmm - View menu buttons

Button	Description
	It lets you find an object being on the top of a hierarchy, i.e. `Root Map` object.
	It allows you to enter inside of an object selected as a home object. Currently it always is `Root Map` object.
	It lets you enter an object standing higher about one step in a hierarchy, i.e. a parent of a current presented object.
	It lets you look at an object inside. It's synonymous with double clicking on an object.
	It allows you to look at a selected item (`computer` or `router` type) using Network Nodes Viewer (xdnnv), if you find a suitable control panel for it.
	It lets you do a similar operation like before but a device name or an IP device you can pass as an argument of a command. In this connection a given device hasn't to exist in the Database.
	It opens a dialog presenting a list of logged in users on dnmmsd server.
	It lets you receive a list of all current defined styles in the Database.
	It lets you receive a list of all current defined state groups.
	It lets you receive a list of all current defined communities.
	It lets you receive a list of all current defined the collection groups in the Database.
	It lets you receive a list of all current defined managed item groups.
	It lets you edit a list of all defined layers in the Database. The layer unctionality isn't currently used in **Network Management Map**.

Button	Description
	It opens a dialog that presents ping objects.
	It shows you a list of current defined ping object groups in the Database.
	It lets you look at a waiting room in which new-discovering objects wait for adding them to a selected item (map). In this case you should select objects from the waiting room using the mouse and move them to a current presented object (map).
	It lets you configure working parameters of the application.

In `View` menu other options are additionally placed which don't show on the toolbar.

Table 8.8. xdnmm - the View menu buttons, that aren't placed on the toolbar

Button	Description
`Show tool bar`	Show or hide the tool bar.
`Show status bar`	Show or hide the status bar.
`Show deleted elements`	Show or hide elements marking as deleted.
`Show not managed elements`	Show or hide elements, that are in `Not managed` state.
`Split view horizontally`	It splits horizontally an active panel into two separated panels.
`Split view vertically`	It splits verically an active panel into two separated panels.
`Close view`	Close an active panel.

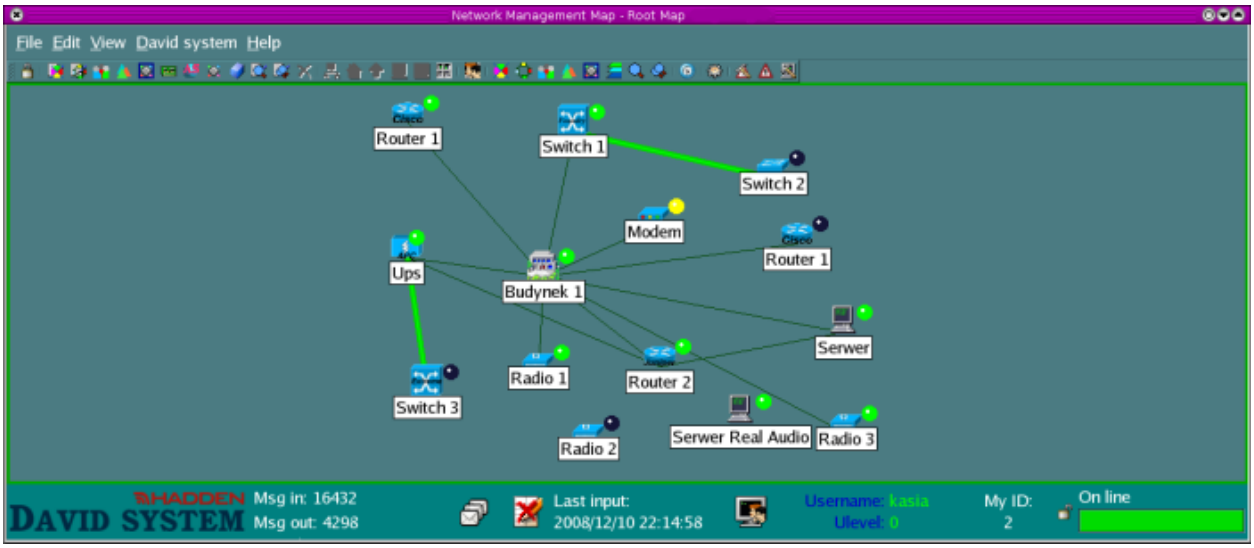
The last three options agree with `David system` menu and let you run the David system applications:

Table 8.9. xdnmm - the David system menu buttons

Button	Description
	Events Service Configurator (xdesc);
	Graphic Notifications Presenter (xdgnp);
	Network Node Views Editor (xdnnve).

Through `Help` menu you can find out the application version and its creation time.

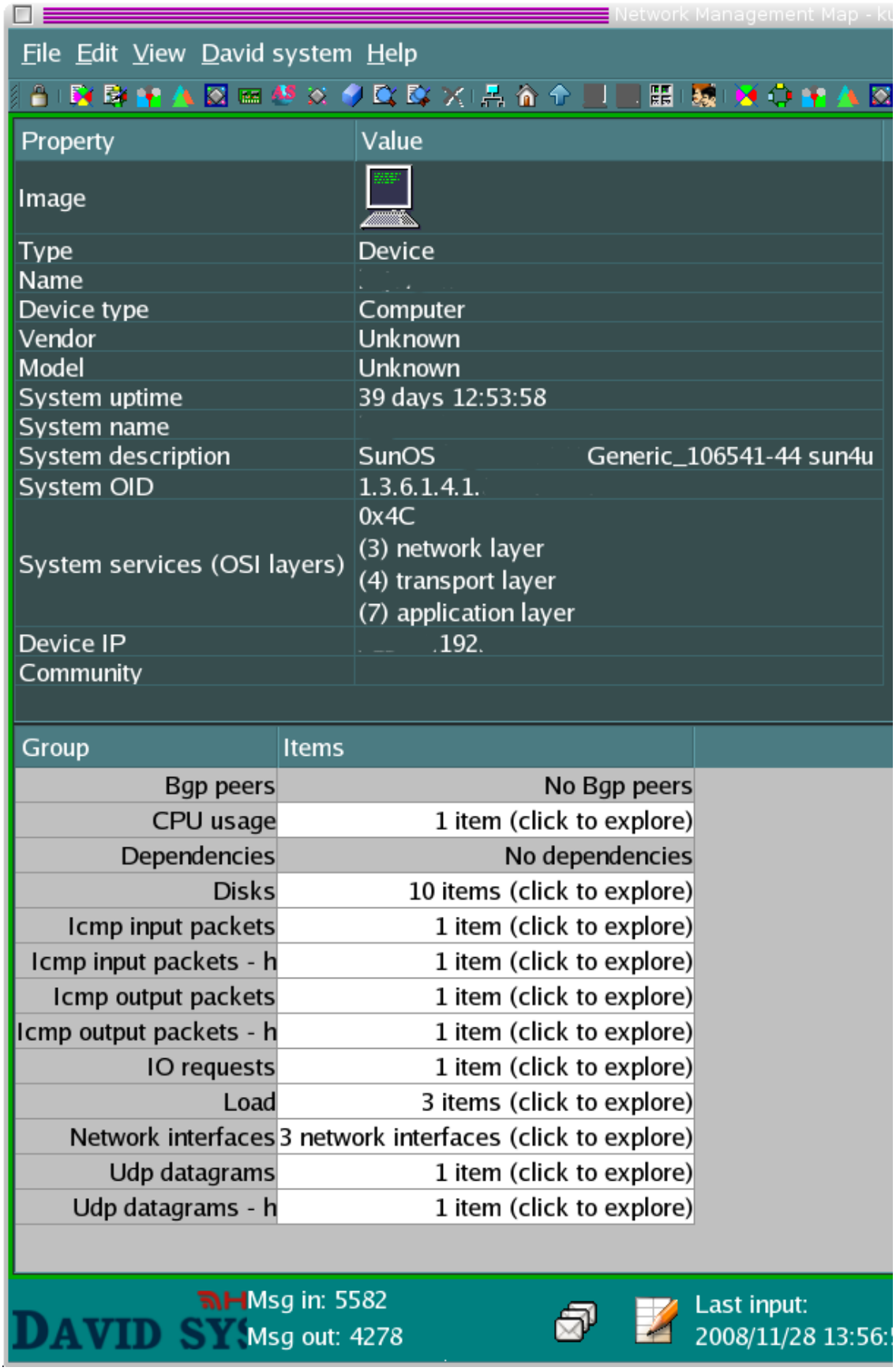
8.4.2.2. Maps and objects exploration



A main view of the application is inside of a presented object. In the Database at last one object always exists. It's a localization which is named `Root Map` and it stands on the top of an object hierarchy. Some objects can include other objects. In this way you can create hierarchy structure of Databse objects. You can see the object inside in very simple way, by double clicking on itself. The application allows to add different types of objects:

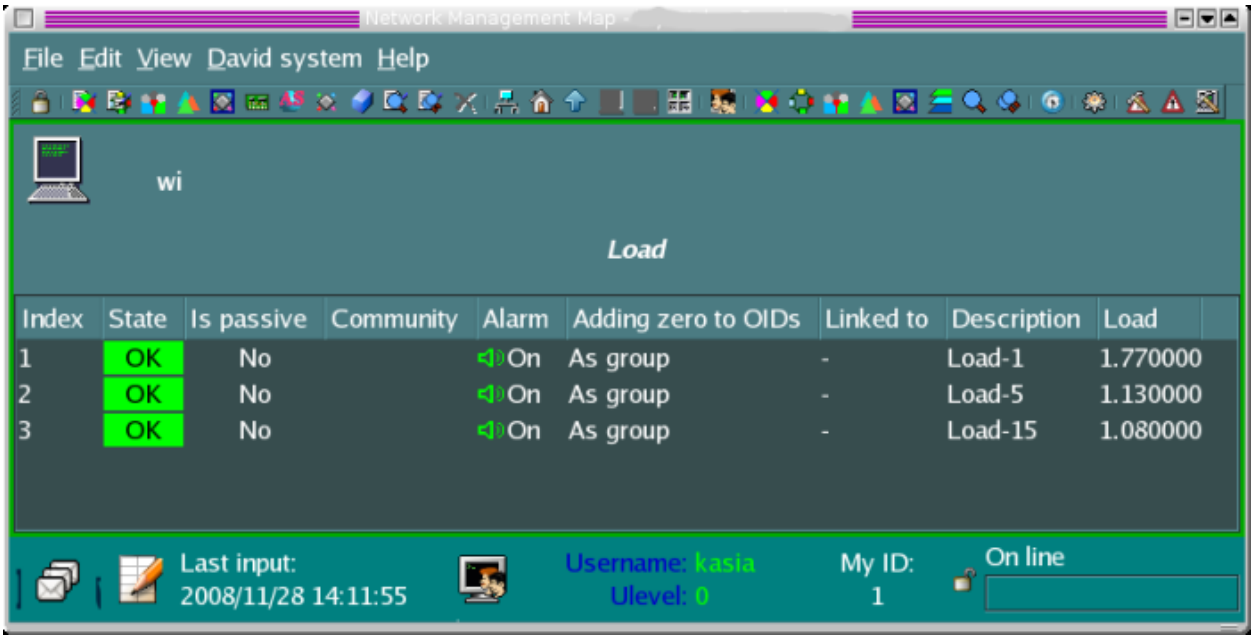
- objects of type `Location` can include some, other objects;
- objects of type `Device` are agreed with a phisical network devices;
- objects of type `Network` are agreed with discovered networks – added automatically;
- objects of type `Link` are connections between two objects of below types on the same map.

When the browsed objct is agree with real device (a type of `Device`), its inside shows a little information about the device and a list of network interfaces, BGP peers and managed items.

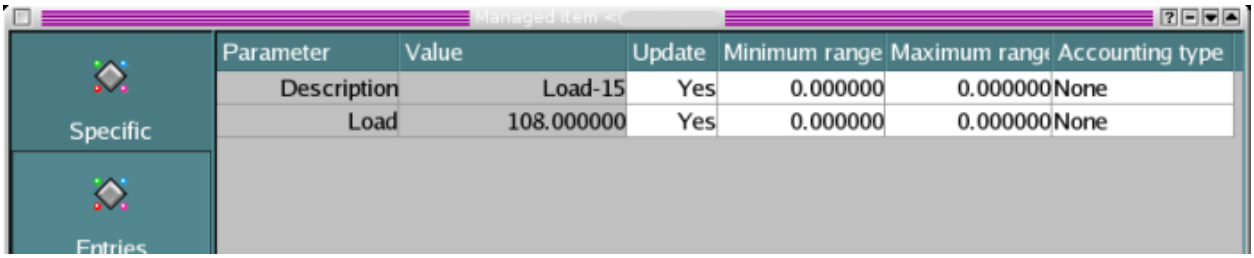

Network Management Map - ku
File Edit View David system Help
Property
Value
Image
Type
Device
Name
Device type
Computer
Vendor
Unknown
Model
Unknown
System uptime
39 days 12:53:58
System name
System description
SunOS Generic_106541-44 sun4u
System OID
1.3.6.1.4.1.
System services (OSI layers)
0x4C
(3) network layer
(4) transport layer
(7) application layer
Device IP
.192.
Community
Group
Items
Bgp peers
No Bgp peers
CPU usage
1 item (click to explore)
Dependencies
No dependencies
Disks
10 items (click to explore)
Icmp input packets
1 item (click to explore)
Icmp input packets - h
1 item (click to explore)
Icmp output packets
1 item (click to explore)
Icmp output packets - h
1 item (click to explore)
IO requests
1 item (click to explore)
Load
3 items (click to explore)
Network interfaces
3 network interfaces (click to explore)
Udp datagrams
1 item (click to explore)
Udp datagrams - h
1 item (click to explore)
Msg in: 5582
Msg out: 4278
Last input:
2008/11/28 13:56:

A type of `Device` includes a few subtypes: `Router`, `Computer`,, `Ethernet switch`, `UPS`, etc.

Clicking on a line of the list describing network interfaces or BGP peers opens their detailed list.



Clicking on a list describing one of managed item groups opens their detailed list. A view of the managed item is specific for a given group and it depends on its define.



The options described below allows you to move between hierarchical maps of objects.

Table 8.10. xdnmm - hierarchical maps of objects buttons

Button	Descriptions
	It lets you see an object which is a parent of a current presented one.
	It lets you see an object on the top of whole hierarchy.
	It gives you the same effect as below but its work will be changed in the future and it will point at a presented object from which **xdnmm** application default begins its work after its connecting with dnmmsd server.

8.5. A configuration of the Database items

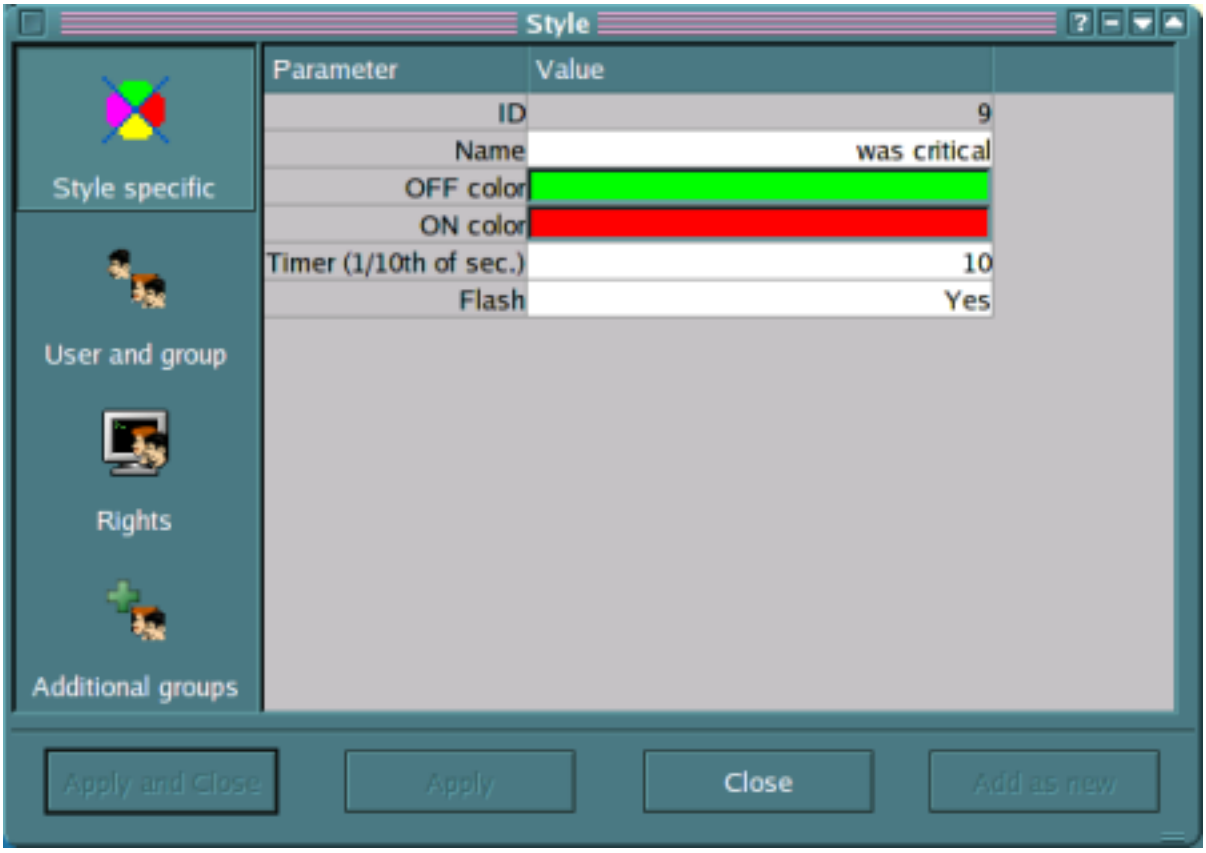
You can make a configuration of **Network Information Database** items using un-modal windows. Some types of items have got additional windows presenting a list of all available items of a given type in the Database.

Each of edit windows has from the left side two or more buttons that allow you to move between panels within a edit window. Beside a specific panel for a given item, at last one more panel exists and it describes access rights to an item or a whole list of items. All edit windows with an exception of two have four following buttons:

- `Apply and Close` - confirms changes and closes a window;
- `Apply` - confirms changes without closing a window;
- `Close` - closes a window;
- `Add as new` - adds a new item on the basis of current specified item parameters.

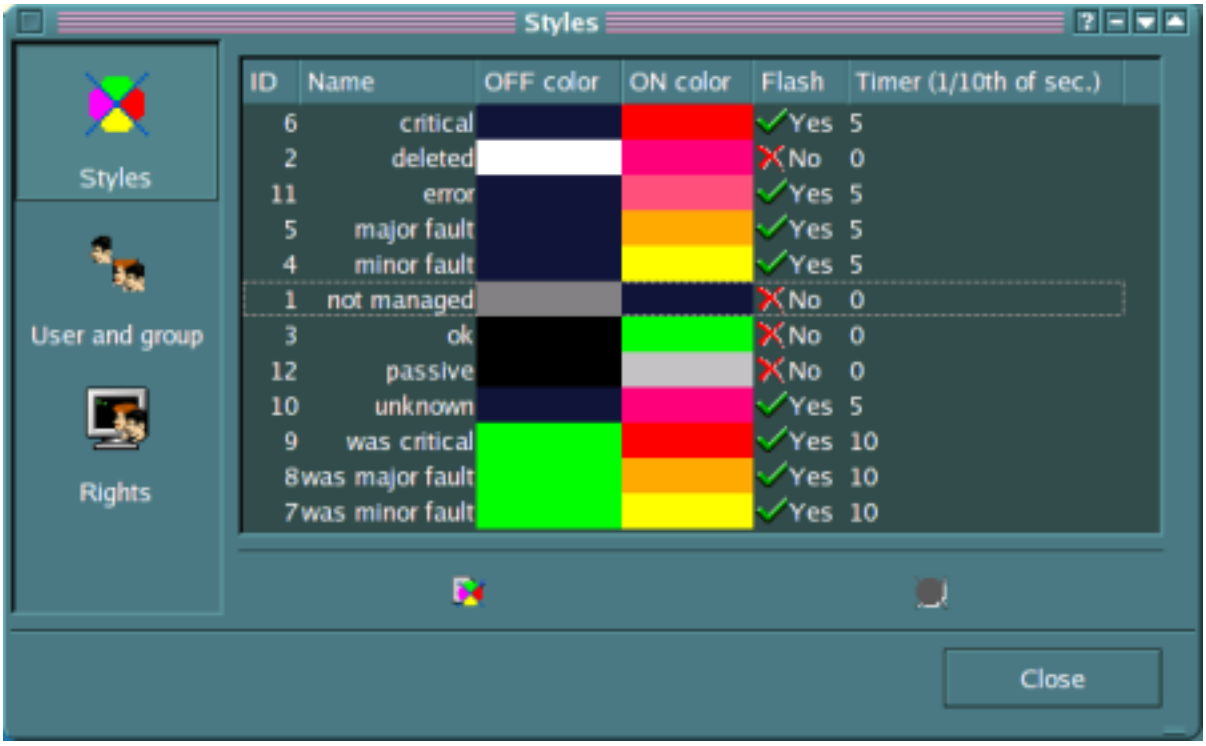
There isn't the last `Add as new` button in edit windows for network interfaces and BGP peers and managed items. Thereby you can't add this items to the Database because the system adds and backups the items itself on the basis of a managed device configuration.

8.5.1. Styles configuration



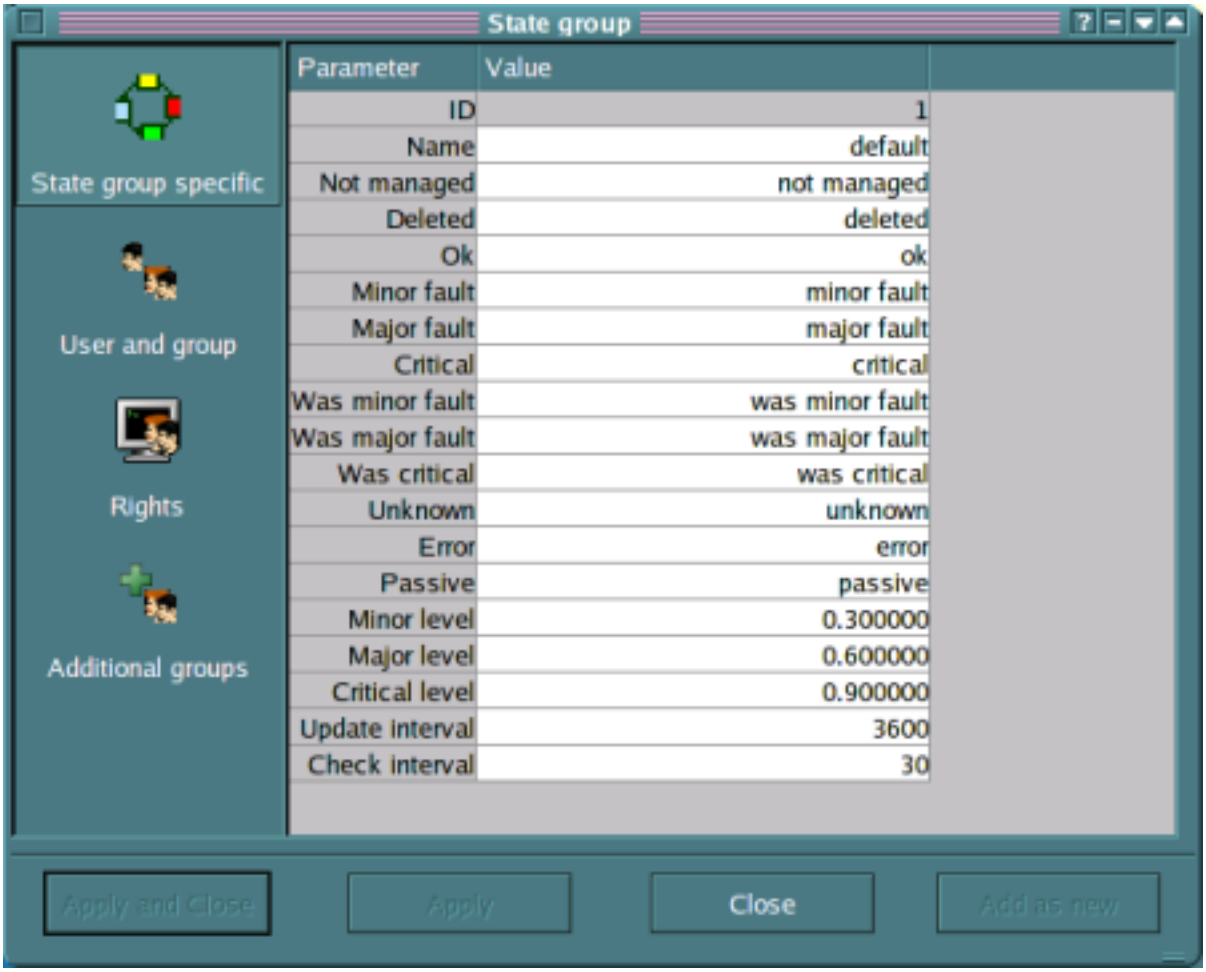
Styles are items of state groups. The states groups can accept the Database items as i.e.: network interfaces and objects. Each style includes an unique style identifier to others styles and it can include a name. It has two properties signified as `OFF color` and `ON color`. That colors mean states on/off and they create an effect of fleshing colors of an item. You can do the effect setting a timer value greater then zero and pressing `Flesz` option. Otherwise a continuous color will be used which is signed as `ON color`.

8.5.1.1. A list of available styles



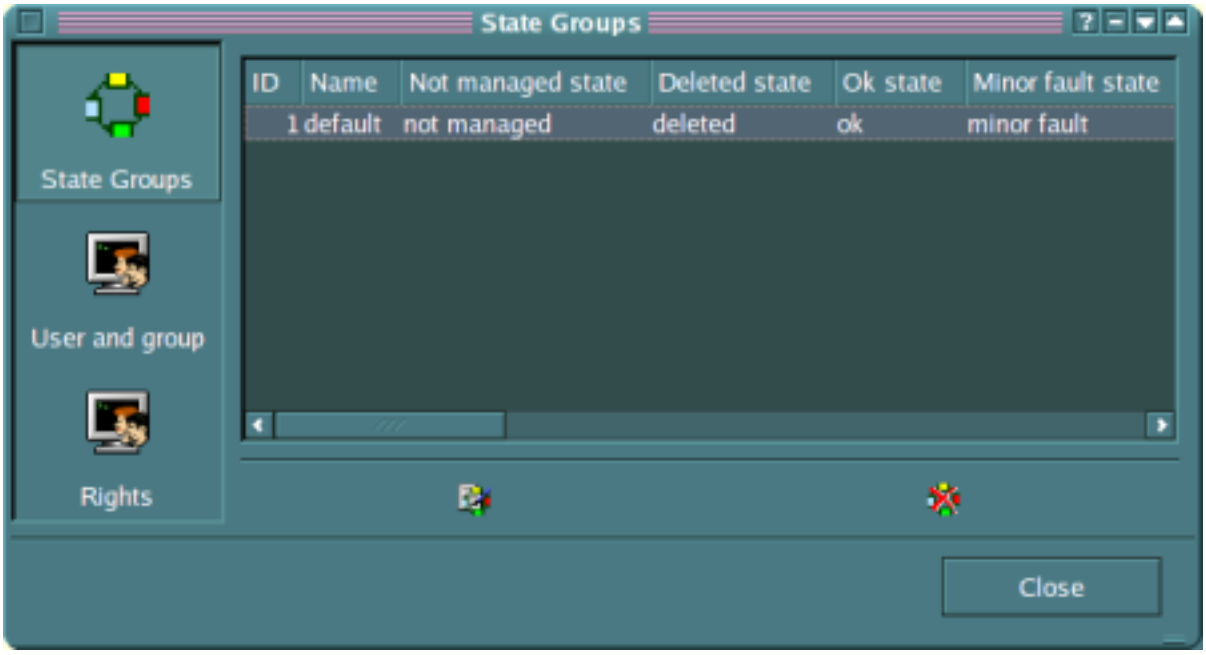
A list of styles presents styles sorted according to their names. There are properties of particular items in the columns. Selected items of a list can be edit in the Style window. You can open it pressing the button below the item list or choosing the window from the main view of the application. You can choose more then one selected item. The right button below the item list allows you to delete selected items of the item list.

8.5.2. A configuration of state groups



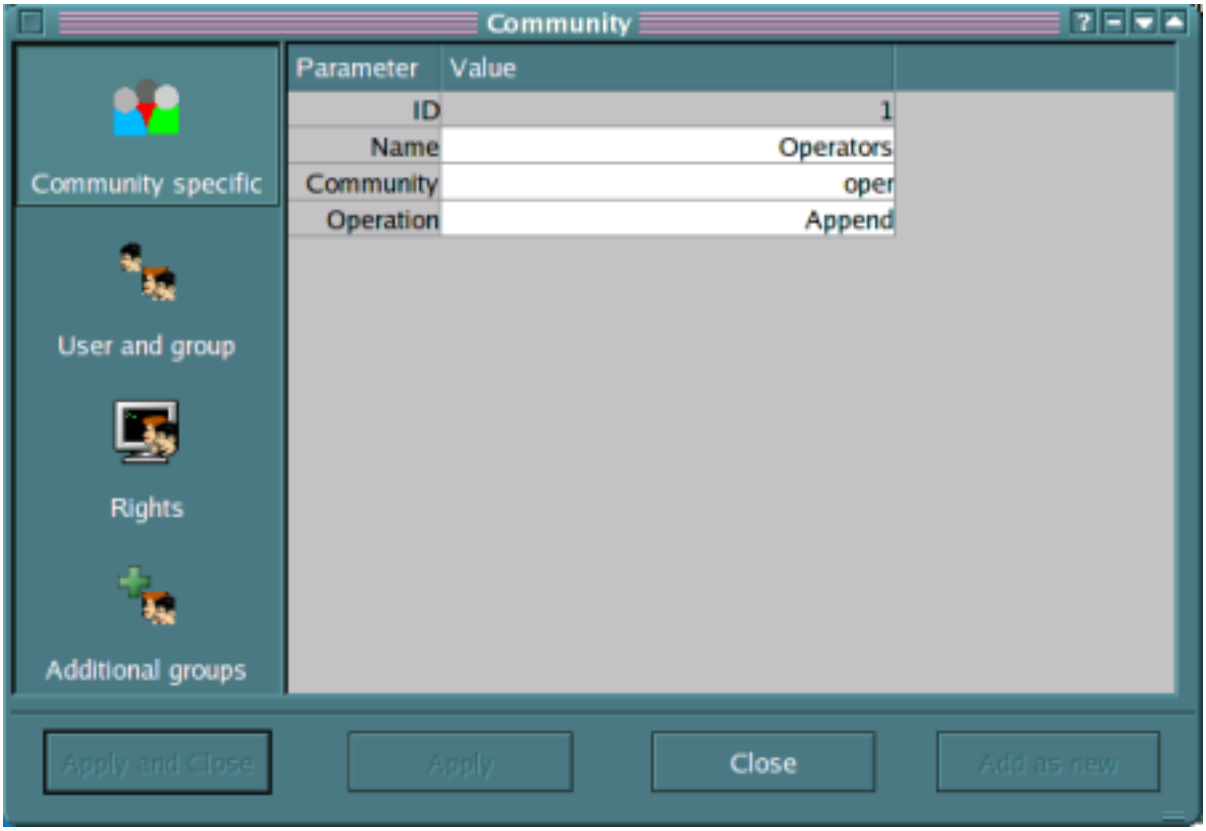
State groups specify a work of the Database items such as: network interfaces, BGP peers, objects and ping objects. Each of the items includes a field describing to which a state group a given item belongs. Each state group includes an unique identifier with relation to other state group and it can include a name. It also includes a list of styles that agree with particular states beginning with `Not managed` and finishing on `Passive`. A state group also have three fields defining levels of particular state items. All possible values belong to a rang <0;1>. An item state will be defined as `OK`, if a counting value of an item is below `Minor level`. Otherwise, if an item value is below `Major level`, an item state will be described as `Minor fault`. Otherwise, if an item value is below `Critical level`, an item state will be described as `Major fault`. Otherwise an item state will be `Critical`. An item value that presents physical device (`computer` or `router`) is counted on the basis of its included items such as: network interfaces, BGP peers and managed items. In other cases states of item children influence on an item state (i.e. `location`). In this way states of children propagate on higher levels of a map hierarchy through their parents. Additionally on states of items influnace item states from which the items are depended on (`Dependences`). The last two fields of a state group are `Update interval` and `Check interval`. The first one describes about what period of time (in seconds) a configuration of monitored devices will be actualized by SNMP protocol and second one - about what period of time an item state will be checked. In a natural way the first field is used for physical devices and second one for network interfaces, BGP peers, managed items and ping objects.

8.5.2.1. A list of available state groups



A list of state groups presents state groups sorted according to their names. In the columns, properties of particular items are showed. Selected items of a list can modified in the State Groups window. You can open the window pressing a left button below an item list, besides its opening in the main view of the application. There are more then one selected item. The right button below an item list allows you to delete selected items from a list.

8.5.3. Communities configuration



`Community` it is an element, that includes a string and a kind of the operation, that can be done with it. The operations are done during work on two `Community` elements. As a result of the operation is a string. Kinds of operations are presented in the table below:

Table 8.11. Meaning of Operation field in Community window

Value	Description
`Append`	It sticks a string of one `Community` onto second one and use `':'` mark as a separator.
`Replace`	It replaces a current string with a new one from the next `Community`.
`Don't inherit`	This `Community` is ignored during a string is calculated for children (i.e.: network interfaces).
`Cut`	It cuts out a string from a string inherited from its parent.

`Name` field allows you to write an item name, and then it is visible as its identifier, when you use an item.

`Community` items are used to control stream of information in many places of the system. They allow you to make a decision, who can be informed about any event or allow you to decide, that make any action or not, etc.

`Community` items can be set for such objects as: network interfaces, BGP peers, managed items, objects, ping objects.

A result string equivalent to a given object or ping object is equal a string of `Community` field, and `Community` item which is set for it. Calculating of a result string for network interfaces, BGP peers and managed items consists of two steps. First a string for a device, i.e. object which has a given item, is calculated, and then its own `Community` item is taken into consider. During the operations on these two `Communities` items, a sequence of items and values of their `Operation` fields figures much.

You have not to attribute `Community` items to any object. So, the result string can be empty for a given object.

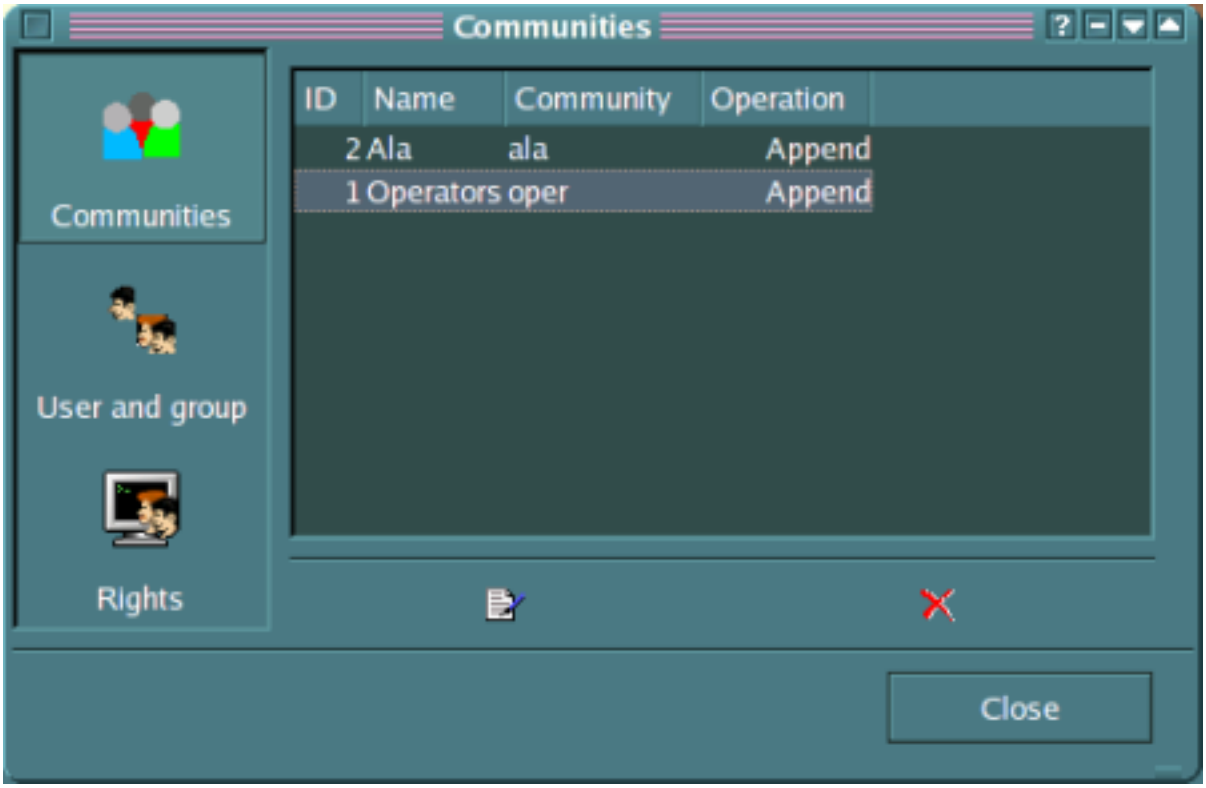
8.5.3.1. An example of using of Community items

The example shows a work result of two `Communities` for a network interface (i.e. `so-1/0/1`) for a given device (i.e. `juniper-gw`). The device `juniper-gw` has `Community` with a string `a:b:c`. The network interface `so-1/0/1` has `Community` with a string `c:d`. The work results of these two `Communities` are presented in the table below:

Table 8.12. The work results of two Communities depending on Operation field

	juniper-gw: a:b:c + append	**juniper-gw: a:b:c + replace**	**juniper-gw: a:b:c + don't inherit**	**juniper-gw: a:b:c + cut**
so-1/0/1: c:d + append	a:b:c:d	a:b:c:d	c:d	a:b:c:d
so-1/0/1: c:d + replace	c:d	c:d	c:d	c:d
so-1/0/1: c:d + don't inherit	c:d	c:d	c:d	c:d
so-1/0/1: c:d + cut	d	d	c:d	d

8.5.3.2. A list of available communities

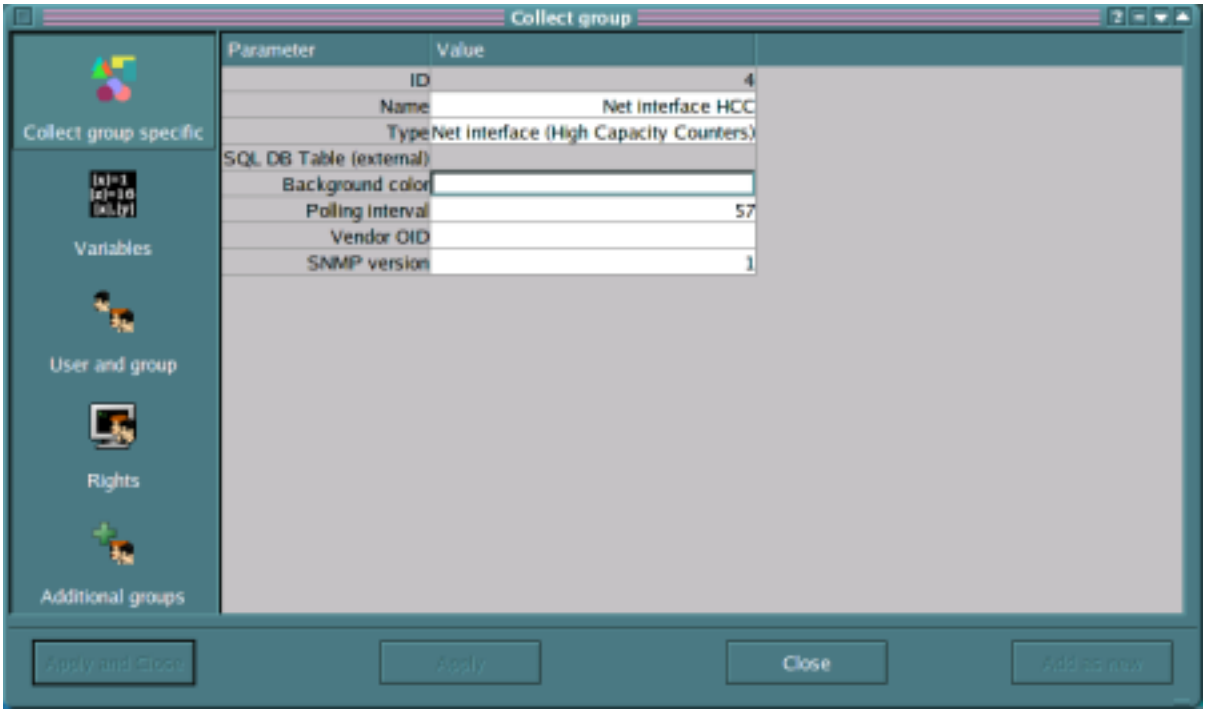


The list presents `communities` sorted into their names. Features characterizing particular items are shown in the columns. Selected items of the list can be edited in Community dialog. There are two buttons below the list:

Table 8.13. Communities dialog - description of the buttons

Button	Decription
	It opens `Community` dialog which also can be run from the main view of the application.
	It deletes selected items of the list. It can be more then one item.

8.5.4. Collection groups configuration



A collection group is a group of SNMP variables collecting same work parameters of a device. The variables have properties allowing to collect them and defining a method of their visualization. Each variable defines one work parameter of a device.

Collection on the base of their parameters are automatically allocated to devices and network interfaces. When we added to the system managed item groups, meaning of collection groups was declined. Now, thay are defining only collecting of traffic of network interfaces, so their another functions wear thin. Managed item groups characterize more general and flexible aprroach. For that's sake, we suggest, that defined groups won't be modified.

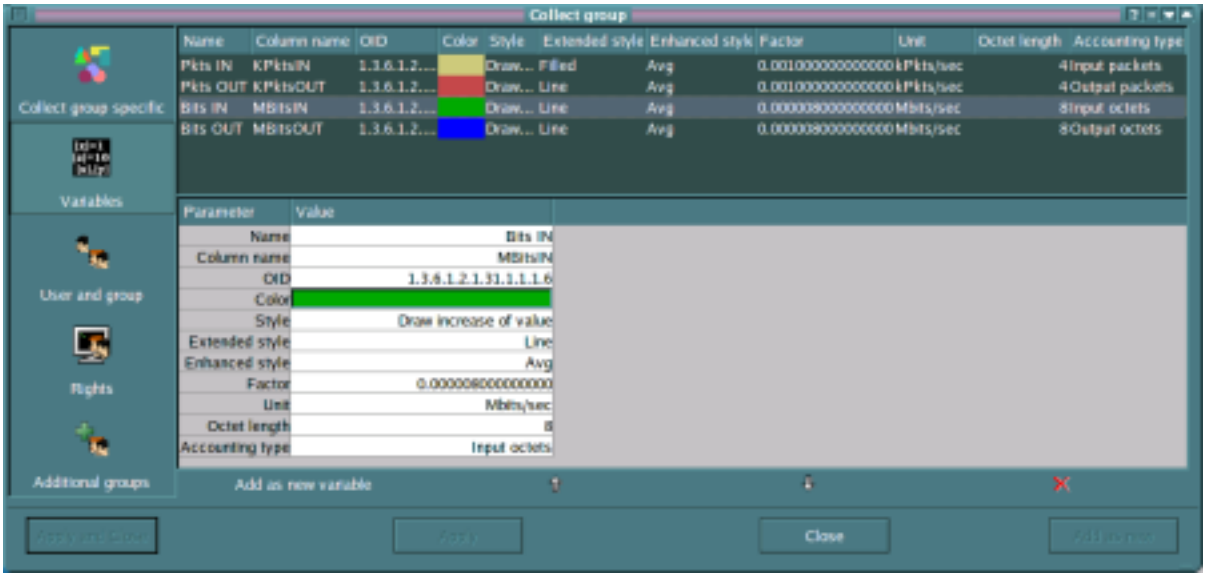
`Collection group specific` tab presents the particular features of collection groups. `Name` field is a text describing a given group and its identifier which is used in other parts of **David system**. `Type` describes a collection type. The following table characterizes them shortly:

Table 8.14. Types of the collection groups

Type	Description
`Net interface`	The collection is destined for network interfaces not-supporting 64-bit counters.
`Net interface (High Capacity Counters)`	The collection is destined for network interfaces supporting 64-bit counters.
`Object`	The collection is destined for an object.

Type	Description
External	The collection allows to vizualization external data. No real data are collected for the collcection, but if you define the collection, you can browse the data through the web application Collection Browser. The data are saved in the Database by a script run by UNIX CRON service.
Aggregated	The special collection is destined for pooling of collections which are collected for network interfaces. Only network interfaces, assigned to a given collection, provide data to it.

SQL DB Table (external) field is important for an External type collection. It includes SQL table name from which Collection Browser will be read data during a graph drawing. Background color field helps you to specify a background color for graphs of collected data. Polling interval field shows an interval, in seconds, of which another sample is collected. Vendor OID and SNMP version fields let you specify criterions, and on the base of them, the collection will be assigned to an item of the given device. Vendor OID field includes OID which is compared to OID variable, finding on a given device. The field is treated as the beginning of that OID. The vendor is ommited as a selection parameter if that field is empty. SNMP version field shows a minimum version of SNMP protocol, that a device has to provide. If these two fields satisfy the conditions, a collection is assigned to a device or its network interface.



Variables tab presents a list of SNMP variables defined for a given group. The view is split into two parts: the top one presents a list of variables, and the bottom one allows to edit particular items of the list. There are four buttons below, that allows to add a new item to the list, delete an existetd one and change an order of the item list.

Selecting a list item causes showing its fields into the edit fields, so any change you make there modifies the selected list item. Name field shows a variable name. It is visible during drawing of a graph and it explains meaning of a variable. Column name field - it is a column name in SQL table created automatically in the Database, where variable values are saved. OID field specifies OID, that a varible is

in evidence in SNMP protocol. In a case of collections for network interfaces, a number `ifIndex` of a given network interface is added to that field value at the end. `Color` field defines a variable color, that it is drawn on a graph during data visualization. `Style` field tell us, how the variable values will be treated and if it will be drown. The following table presents values of `Style` field and their meaning:

Table 8.15. Values of Style field for variables of colletion group

Value	Description
`Do not draw`	A variable is not draw on a graph.
`Draw value`	The variable values can be changed in any line, i.e. they can be increasing or decreasing.
`Draw increase of value`	The variable values can only increase until it reaches a maximum value and starts increasing again from zero. It is a characteristic feature of counters.

`Extended style` field shows, in which way a variable will be drawn. It can have three values, but `Bar` value is not taken account. `Line` value lets the variable draw as a line, while `Filled` value lets a graph fill at the bottom. `Enhanced style` field can have four values: `Min`, `Max`, `Min & Max` and `Avg`. The field has historical meaning, because `Avg` value is always used. `Factor` field is a number on which variable values are multiplied by during its drawing and further processing. Its value also tell us how many times you sholud multiply a variable value, that its quantity was expressed in units specified in `Unit` field. It can have the following values:

- bits/sec;
- kbits/sec;
- Mbits/sec;
- Gbits/sec;
- Bytes/sec;
- kBytes/sec;
- MBytes/sec;
- GBytes/sec;
- Pkts/sec;
- kPkts/sec;
- MPkts/sec;

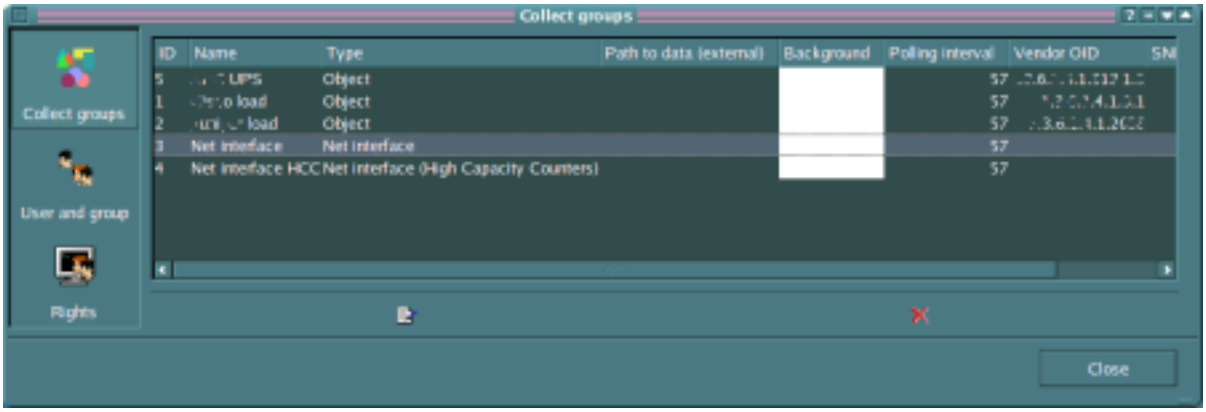
- GPkts/sec;
- Other.

`Octet length` field, if it is greater then 0, shows how many bytes (octets) a given variable has (i.e. if it is a value of 32 or 64 bits). The field `Accounting type` shows if the variable is accounted or not and, if it is, how its values should be treated. The field can get following values:

Table 8.16. Values of the field Accounting type for variables of colletion group

Value	Description
`None`	The variable is not accounted.
`Input octets`	Values of the variable are accounted as input octets (bytes).
`Output octets`	Values of the variable are accounted as output octets (bytes).
`Input packets`	Values of the variable are accounted as input packets.
`Output packets`	Values of the variable are accounted as output packets.

8.5.4.1. A list of available collection groups



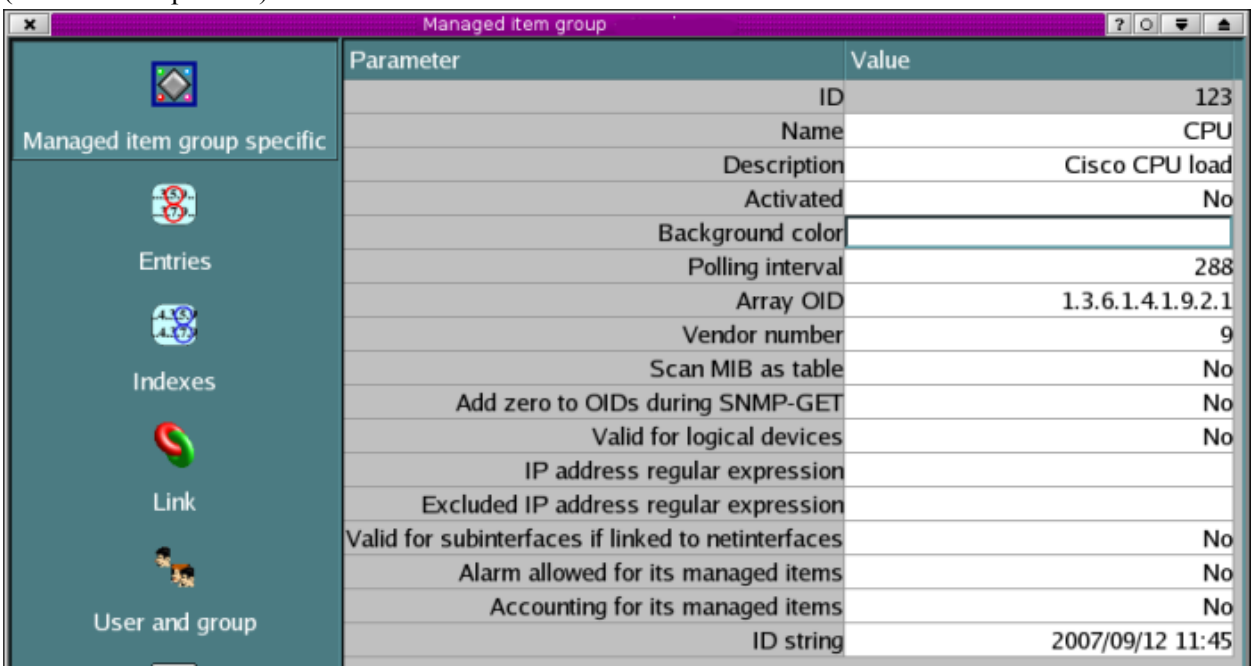
The list presents the collection groups, that are sorted into their names. Features characterized particular items are shown in the columns. Selected items of the list can be edited in Collect group dialog. There are two buttons at the bottom of the list.

Table 8.17. A description of the Collect groups buttons

Button	Description
	It opens `Collect group` dialog, that you can also run from a main view of the application.
	It let you delete selected items of the list. It can be more then one selected item.

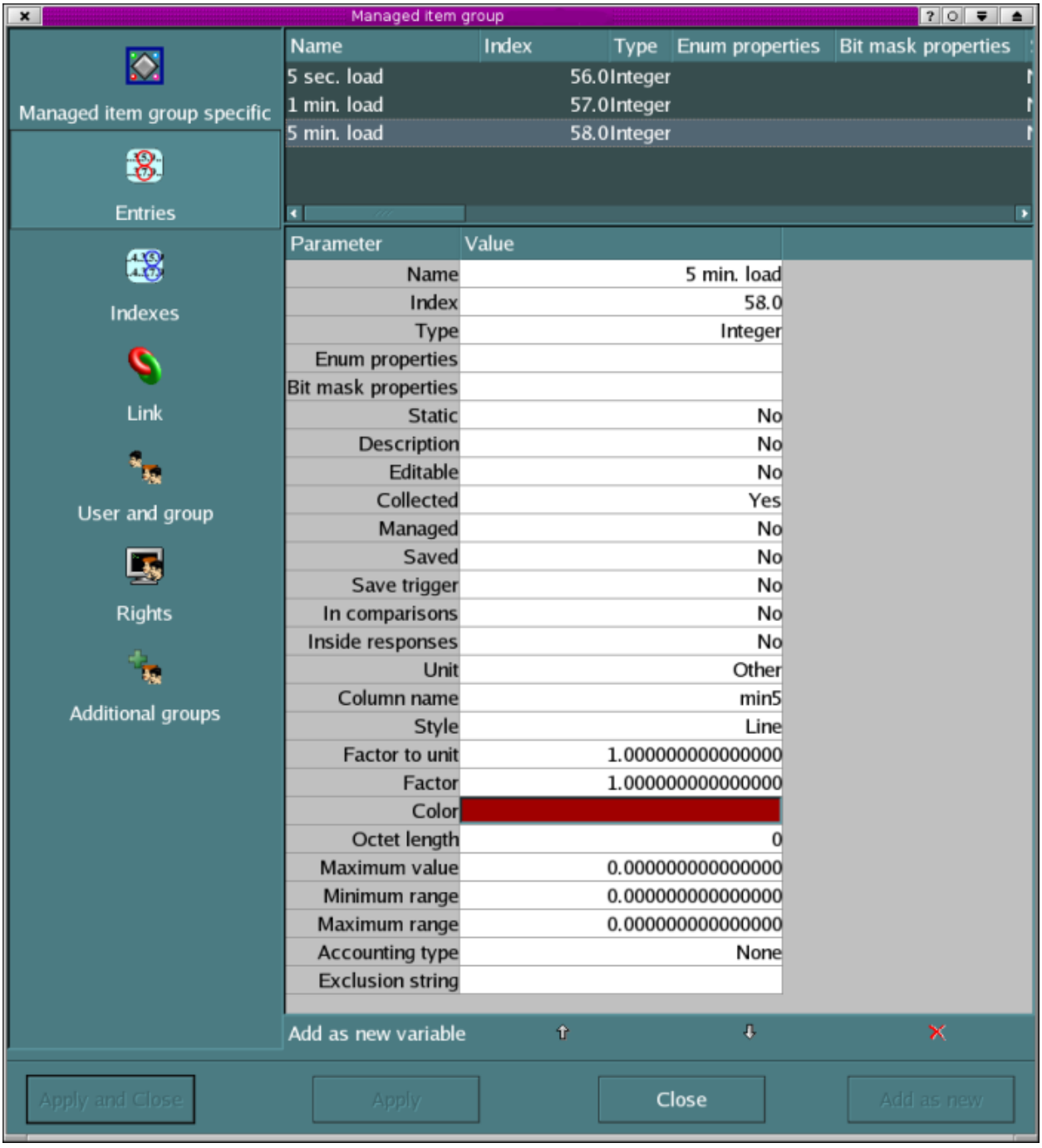
8.5.5. Managed item groups configuration

Managed item groups are definitions, that interpret particular items of tables which are in devices MIB and acessible through SNMP protocol. The definition allows to describe a few ways of data interpretation. A part of data from a given table can be collected, and another part of data can be monitored, while another one can be only gathered during a standard procedure of data update of a given device. The device is searched at an angle data specified in a definition of the group, during the update procedure. The finding data, that match the group definition, become the managed items. They can represent any work aspects of a device: serviced protocols (BGP, ISIS, OSPF, MPLS, VPN etc.), phisical device elements (ports, interfaces, power supplies, fans, proccesors etc.) and other entities acessible by SNMP protocol (i.e.: filters of packets).



`Managed item group specific` tab shows particular features of the managed item groups. `Name` field includes a short name of the group. It is visible during browsing of a collection or a content of a given device, so it should show clearly which aspect of device working it concerns. `Description` field allows to describe a content of the table which is scaned using the group. `Activated` field shows, if the group is taken into consideration during the data update. `Background color` field describes a background color of drawn graphs for data, collected on the strength of the group. `Polling interval` field shows an interval, in seconds, of which another collected data of the group are collected. `Array OID` field includes an OID of the MIB table which is scaned to search managed items. `Vendor number` field descrbes an identifier of device vendor, that will be scaned for the group. If a

value of Vendor number field has 0, the group is looked for on devices of all vendors. Scan MIB as table field shows, if a group definition relates to a table in MIB database or to its scalar values. In other words, sanning of MIB database is done according to searching of rows of a given table or ordinary salar values. Add zero to OIDs during SNMP-GET field allows to go around errors of SNMP implementation on same devices. The error shows rather seldom (some SNMP agents on measures SNMP to UPS of APC company), the problem is, that a request SNMP-GETNEXT for some MIB elements gives back the next element, which answers in error on a direct request SNMP-GET. If you want to get a correct answer, you should add '0' to its identifier. So, Add zero to OIDs during SNMP-GET field almost has always No value. Valid for logical devices field describes, if logical devices (logical routers of Juniper company) will be searched at an angle the group or not. Becaouse some MIB table are the same both phisical devices and logical devices configured on them, duplicating data is no use. IP address regular expression field allows to limit searching of the group for devices, that are polled through a given set of IP adresses. The empty field shows, that IP address doesn't decide about searching of the group on a device. Excluded IP address regular expression field lets you limit searching of groups for device, that are polled through excluding a given set of IP addresses. The empty field shows, that IP addres doesn't influance on decision to search group on a device. Valid for subinterfaces if linked to netinterfaces field shows, that managed items, finding on a scaned device and connected with its network interfaces, will be ignored, if a connected interface is a subinterface. There are some situations when the same data are replaied for all elements corresponding with subinterfaces of the same network interface. In this case monitoring of an element is sufficient, that is connected with a main network interface. Alarm allowed for its managed items field allows to describe, if for managed items, discovered by the group, will be run scripts alerting about CRITICAL state. Accounting for its managed items field shows, if given variables are accounted or not. If they are, the variables, that a value of Accounting field is different from None, are accounted. ID string field includes a string identyfing a given managed item group. For convenience, the field includes a date of creation of a given group. The field is only important during actualization of **Network Manager** product, because new, identyfied groups are added on the base of this field.



`Entries` tab presents a list of searched items of a given group during scanning of a device. If MIB table is scanned, each entry is agreed with a single column of the table. If scalar values are scanned, the connections between entries can be default, free. The panel is split into two parts: the top one shows a list of entries, and the bottom one lets to edit elements of the list. There are four buttons below the list, that allow to add a new element to the list, delete an existed item and change a sequence of the list.

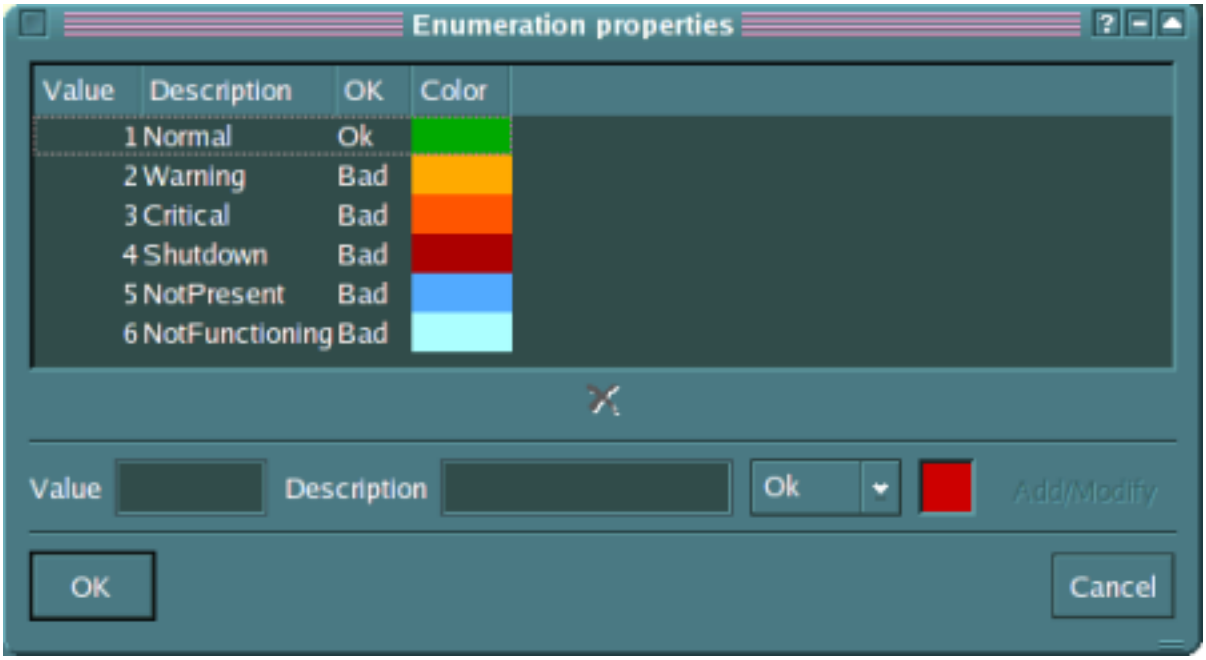
Selecting of the list item concerns, that its fields appear on edited fields and any change changes the list item. `Name` field includes an item name (a column). `Index` field specifies a column index of MIB table,

that the entry concerns. Indexes of all entries in a given group have to be different from themselves. An index lower then zero shows, that the item isn't taken from MIB table but is defined by a user. `Type` field describes type of values and shows in which way a value of MIB object can be interpreted. The below table presents values of `Type` field and their meaning:

Table 8.18. Values of Type field for entries of managed item group

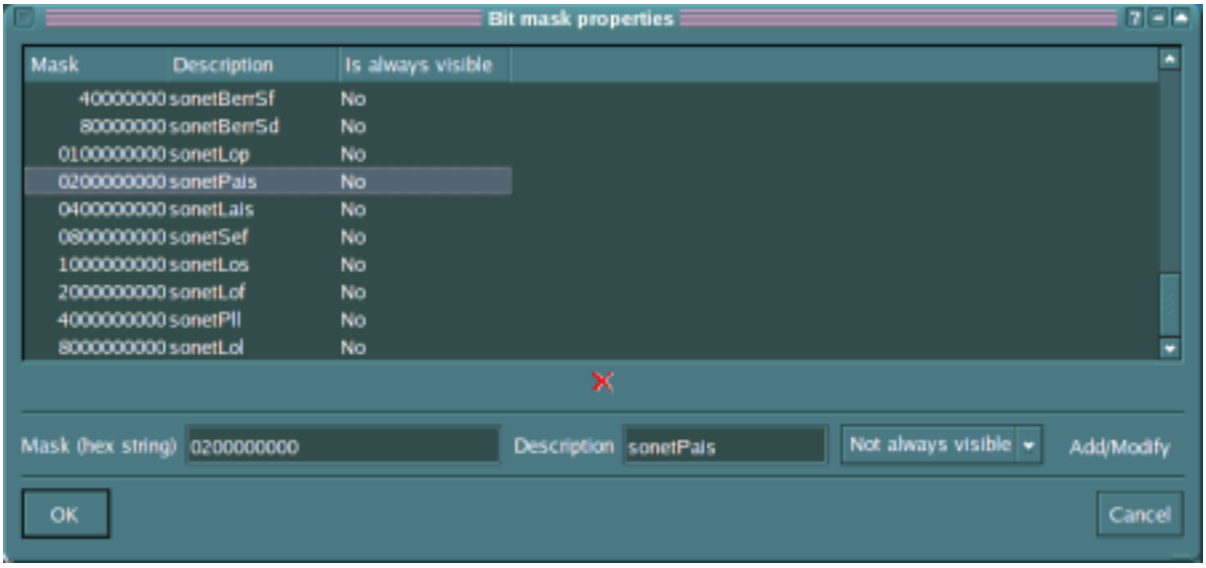
Value	Description
`String`	The fild is a string.
`Integer`	The field is a signed integer.
`Enumeration`	The field includes a set of the enumeration values.
`Administration state`	The same meaning as `Enumeration` field, but additionally with a tip about meaning of the field.
`Operational state`	The same meaning as `Enumeration` field, but additionally with a tip about meaning of the field.
`Unsigned integer`	The field is an unsigned integer.
`Value`	The field includes a real number.
`Counter`	The field includes a counter.
`Speed`	The field includes a number in bits/sec.
`Bool`	The field includes a bool number.
`Date and time`	The field includes date and time.
`Bit list`	The field is interpreted as a list of bits (i.e. hexadecimal string), where positions of bits '1' are printed out..
`Other`	The field includes other type of data and it is treated as a string.

`Enum properties` field includes a list of enumerated values, that an object of the MIB table can have. A set of the enumerated values isn't ignored only then `Type` field is set an `Enumeration`, `Administration state` or `Operational state` value. If you want to edit the list, you should click `Enum properties` field. The appeared dialog lets you edit particular items of the list of enumerated values.



`Enumeration properties` window lets you edit a list of enumerated values. Each value must be unique. Each entry corresponding to an enumerated value has `Description` field, that is a texted description of the value and it shows, that it is a positive or negative value. `Color` field specifies a color, that a value is drawn.

The field `Bit mask properties` of `Entries` tab includes a list of the bit masks, that can include values of MIB table object. Values of the filed are taken into consideration, when they exist, regardless of a velue of `Type` field. When you want to edit the list, you should click on `Bit mask properties` field. The open dialog lets you edit particular elements of the bit mask list.



`Bit mask properties` dialog lets edit a list of the bit masks. Each entry corresponding with bit mask in the form of hexadecimal includes `Description` field which is text description of a bit mask. The fild `Is always visible` describes, if a value of a given mask is shown always or only then,

when it shows.

`Static` field of `Entries` tab lets describe, if values of the MIB object are dynamic, i.e. thay are taken from MIB by SNMP protocol, or if thay are edited by a user. `Description` field describes, if values of the MIB object are treated as a description of the managed item. `Editable` field shows, if a value can be edited by a user, while `Collected` field shows, if values of the MIB object are collected by dedcd. `Managed` field shows, if the MIB object is monitored, in other words, SNMP requests concerning its value periodically are send by dnmmsd. `Saved` field describes if monitored requests would be saved, when values are changed. In other way values are only updated, their changes are ignored and aren't logged to logs. `Save trigger` field shows if the MIB object change involves writting down whole managed item to log. `In comparisons` field shows, if a given MIB object is taken into consideration during comparison of two managed items. If one field has `TRUE` value, only the values of the MIB objects mean during comparison of two managed items. In other coincidence, item indexes are compared with. Manged item comparison is taken during the update procedure of a device. Then, the managed items are updated, new one are added and the not-existed one are deleted. `Inside responses` field shows, if values of the MIB object are included in responses to requests of dnmmc. `Unit` field specifies units, in which the MIB object values are expressed. It is important i.e.: during drawing of graphs. The field can have the following values:

Table 8.19. The Unit field values for entries of the managed item group

Value	Description
`bits/sec`	It shows a number of bits in second.
`Pkts/sec`	It shows a number of packets in second.
`sec.`	It shows time.
`Other`	It shows other units.

`Column name` field is important if the column values are collected. It specifies a column name in SQL table, when data are gathered. `Style` field shows, how the MIB object values will be drawn. The below table shows possible values:

Table 8.20. Style field values for entries of the managed item group

Value	Description
`None`	No values are drawn.
`Line`	A linear graph.
`Filled`	A filled graph.
`Bar`	Bars.

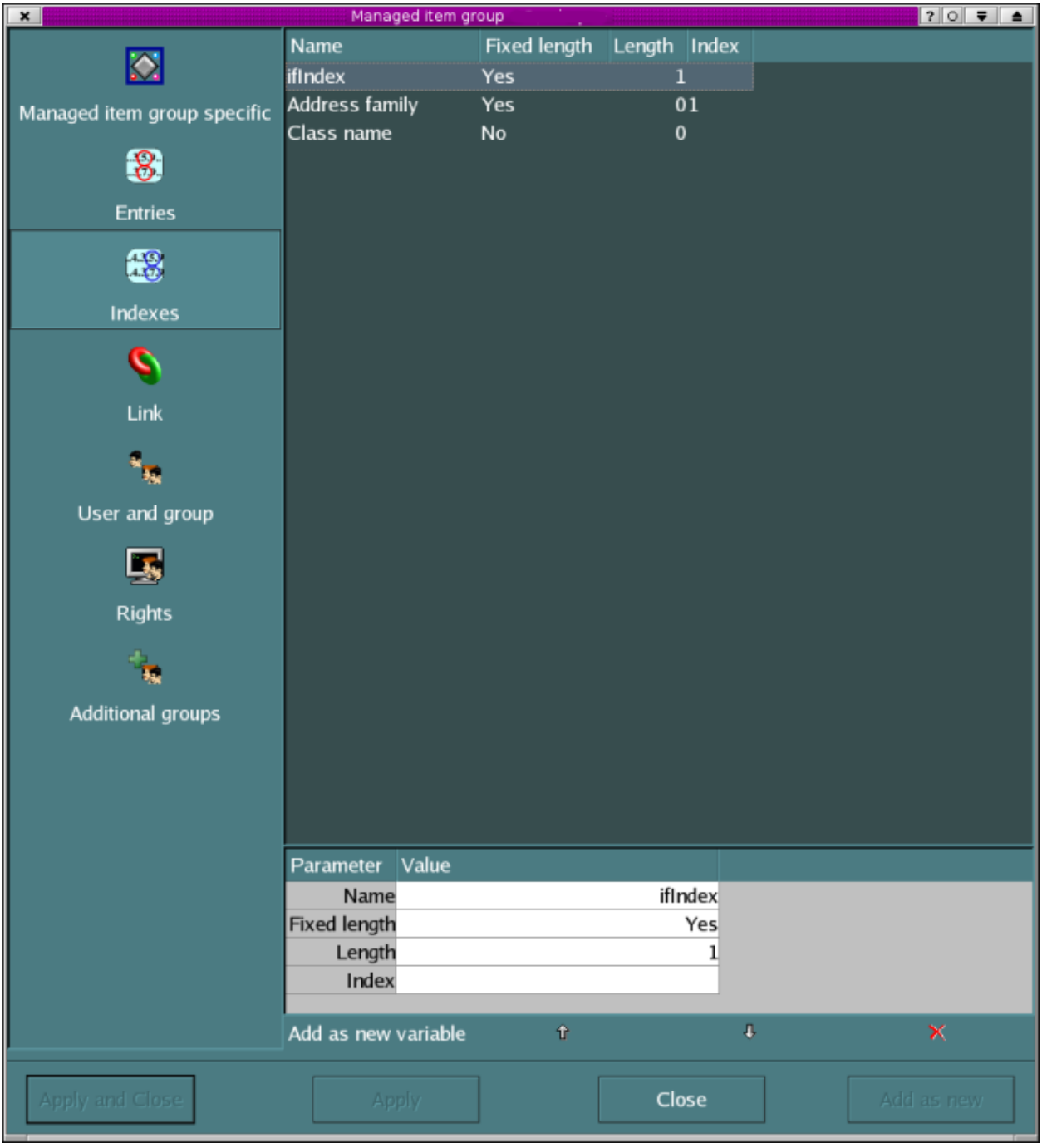
`Factor to unit` field shows a number which multiplied by a value of the column gives an amount

expressed in units that `Unit` field shows. `Factor` field shows a next mulitiplier. It is used during sending the data to **Operation Manager** product. `Color` field specifies a color which the column values are drawn. `Octet length` field, if it is greater then 0, shows how many bytes (octets) a given variable has (i.e. when it is a value of 32 or 64 bits). `Maximum value` field lets you set a maximum value of a given variable. `Minimum range` and `Maximum range` fields let you set a range of acceptable values for this variable. Values outside the range generate alarm situations. The field `Accounting type` shows if the column is accounted or not and, if it is, how its values should be treated. The field can have the following values:

Table 8.21. Values of the field Accounting type for entries of managed item group

Value	Description
`None`	The column is not accounted.
`Input octets`	Values of the column are accounted as input octets (bytes).
`Output octets`	Values of the column are accounted as output octets (bytes).
`Input packets`	Values of the column are accounted as input packets.
`Output packets`	Values of the column are accounted as output packets.

`Exclusion string` field - if it is not empty, it shows what kind of variable string excluded whole managed item from monitoring.

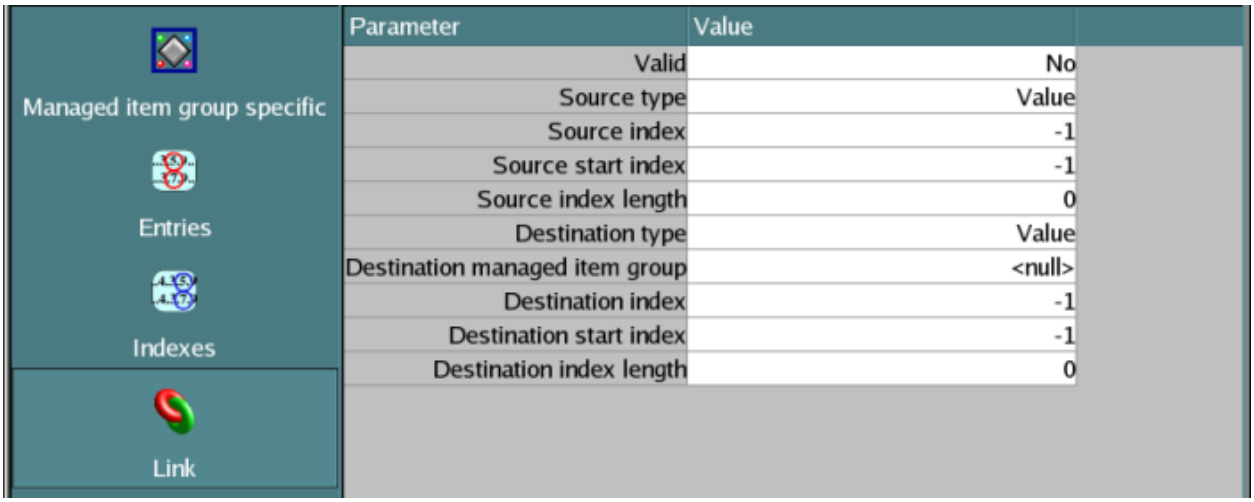


`Indexes` tab shows elements of indexes which managed items have and thay are a result a definition of the group. The tab is empty in mainly cases. You can use the tab, when you want to search MIB table and choose its a few rows. `Indexes` tab defines an item selection on the base of index values. It includes a list of a part of the index until the part being a filter. Each of the parts has a few properties, that define it.

The panel is split into two parts: the top one presents a list of indexes, and the bottom one allows to edit particular items of the list. There are four buttons below, that allows to add a new item to the list, delete an existetd one and change an order of the item list.

Selecting of a list item causes appearing its fields into the edit fields, so change of the list item is made by any change.

`Name` field lets you remember meaning of the item. `Fixed length` field show if the index part has fixed length, or not. If it has fixed length, `Length` field allows to pass the size. If `Index` field is specified, values of other fields are ignored, because both lenght and value can be set using `Index` field. A value of `Index` field is a filter of managed items. Only the manged items will be acepted, that an adequate index part is equal in `Index` field value. If the field is empty and in `Fixed length` field is selected `No`, a length of the index part is read from a next number included in index of the managed item.



`Link` tab allows to specify requirements on which a result managed item connects with other managed item or network interface. `Valid` field shows if a link is important or not. If not, another fields of the tab are ignored and no connection is created. Other fields are split into two groups. The first one concerns a source, i.e. proccesing, managed item, and the second one concerns an aim of connection. `Source type` field shows if an index of proccesing managed item is selected to search a connection or its value. If `Value` value is selected, `Source index` field includes a column index of MIB table, where the value is included. If a value of `Source index` field is `Index`, an index part of proccesing managed item is selected to search a connection. Then `Source start index` field includes an index of OID element, that is an item index while `Source index length` includes a length of a index part of the item, that will be used to comparison.

Analogical meaning have fields concerning a searched aim of the link. `Destination type` field can have one of three values:

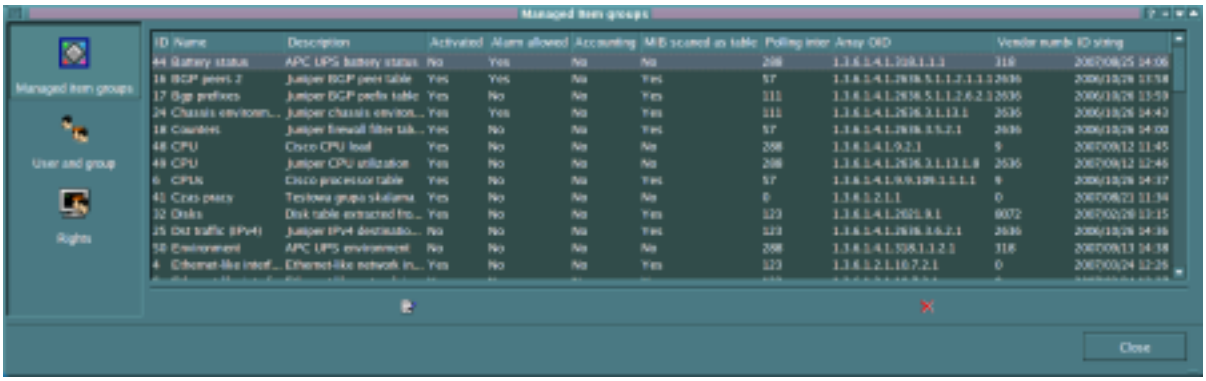
Table 8.22. Values of Destination type field for Link tab of managed item group

Value	Description
`IfIndeks`	A network interface is a searched item about adequate value of `ifIndex` field.
`Indeks`	A managed item is a searched item about adequate column value with `Destination index` index of MIB table.
`Value`	A managed item is a searched item about adequate part of index, that begins on

Value	Description
	`Destination start index` position and has a length `Destination index length`.

`Destination managed item group` field show a destination group of the connection, if a value of `Destination type` field is `Index` or `Value`.

8.5.5.1. A list of accesible managed item groups



The list presents managed item groups sorted according to their names. Features characterized particular items are shown in columns. Selected items of the list can be edited in Managed item group dialog. There are two buttons below the list.

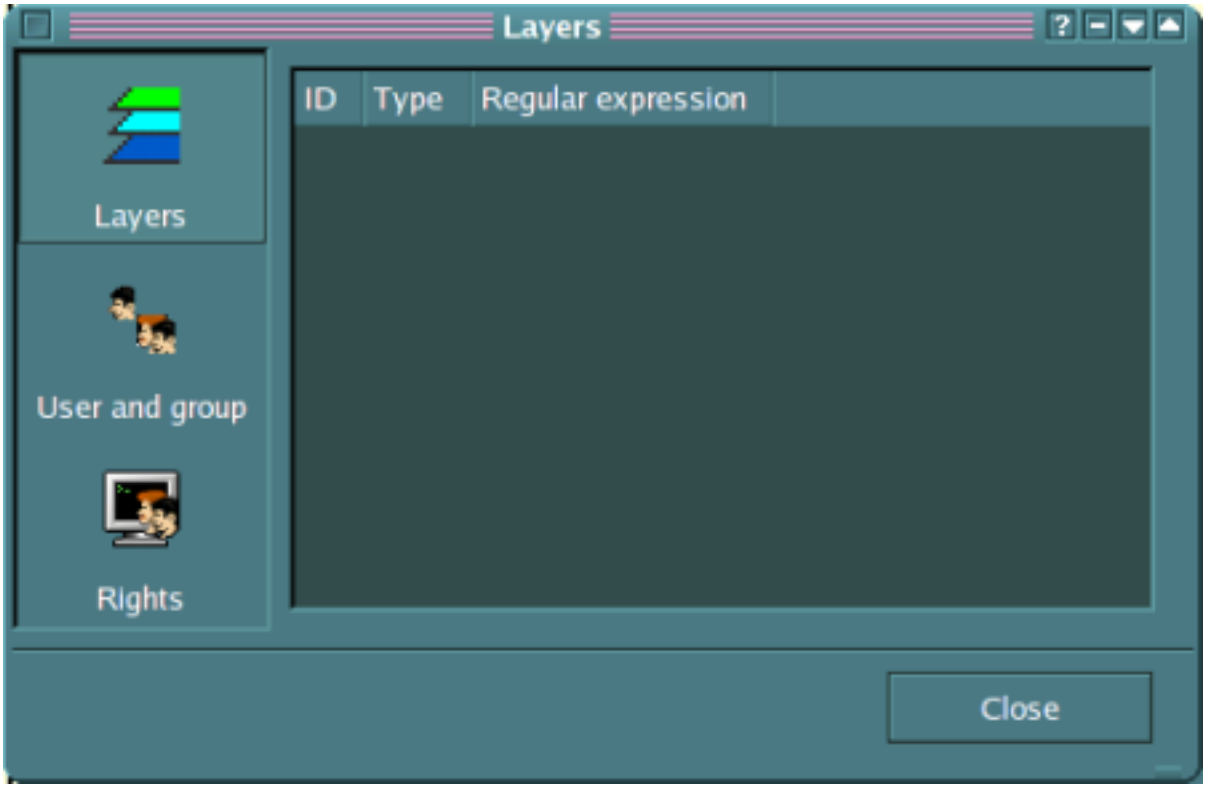
Table 8.23. The buttons of Managed item groups dialog

Button	Description
	It opens `Managed item group` dialog, that also can be opend in the main view of the application.
	It allows to delete selected items of the list. It can be selected more then one item.

8.5.6. Layers configuration

The functionality is currently inaccessible.

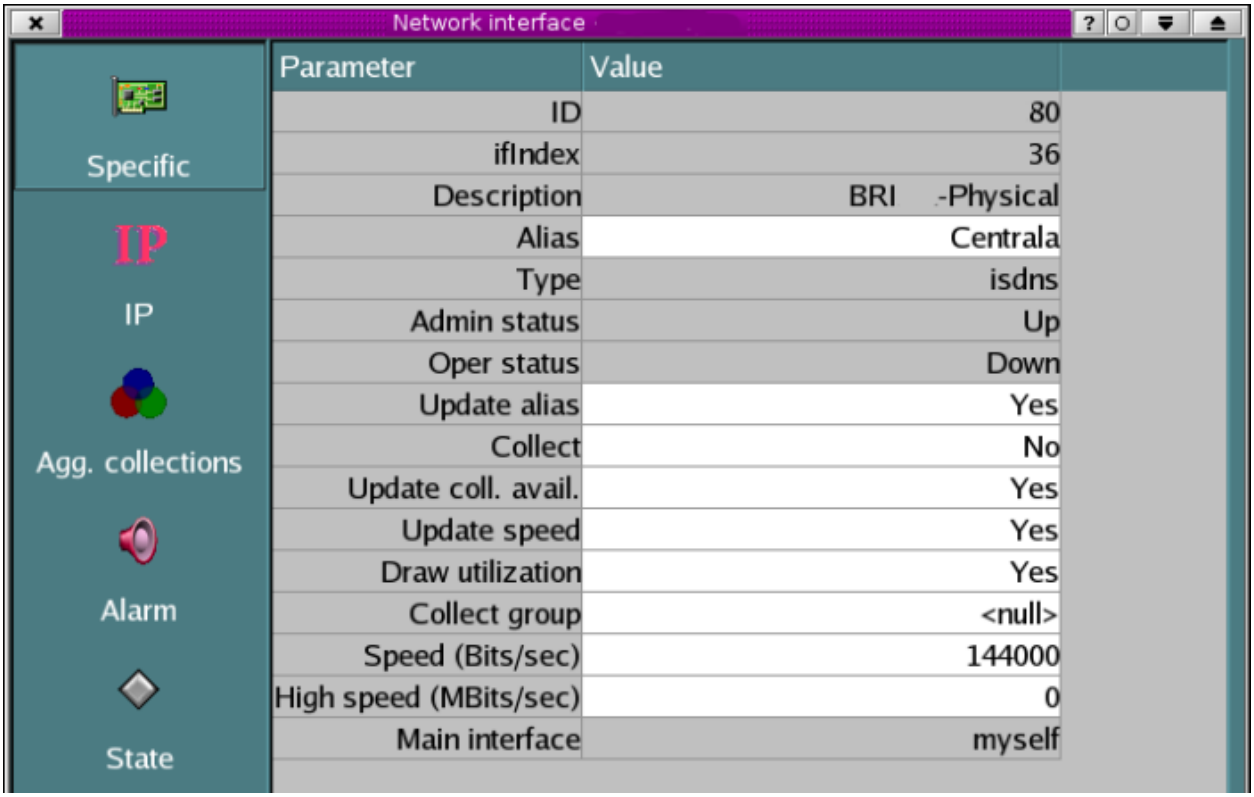
8.5.6.1. A list of available layers



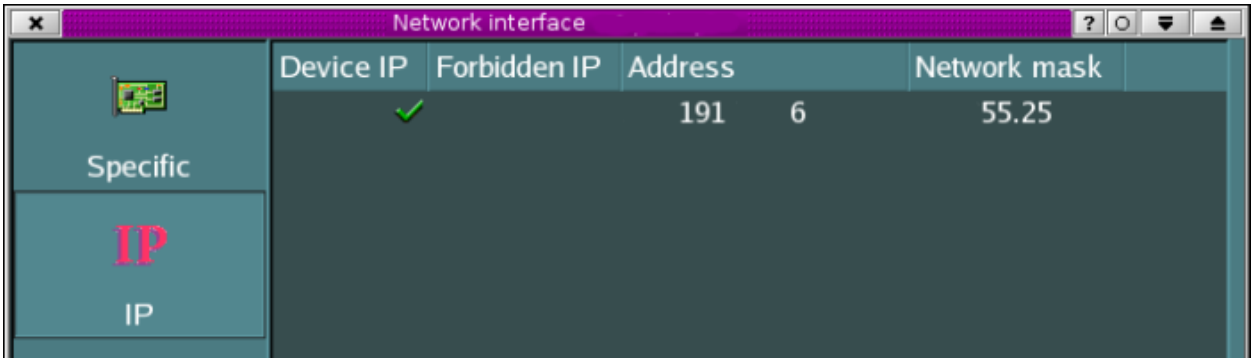
A list of accessible layers presents layers sorted according to their unique identifiers. In the columns properties of particular items are showed. There are a layer type and regular expression described a layer. Because the layer functionality isn't used currently, we finish its description.

8.5.7. Network interfaces configuration

Network interfaces always belong to physical devices discovered by dnmmsd server. They represent both physical and logical network interfaces that are presented at monitored devices. They have many features and properties resulting from both their equivalents on physical devices and David system functionality.

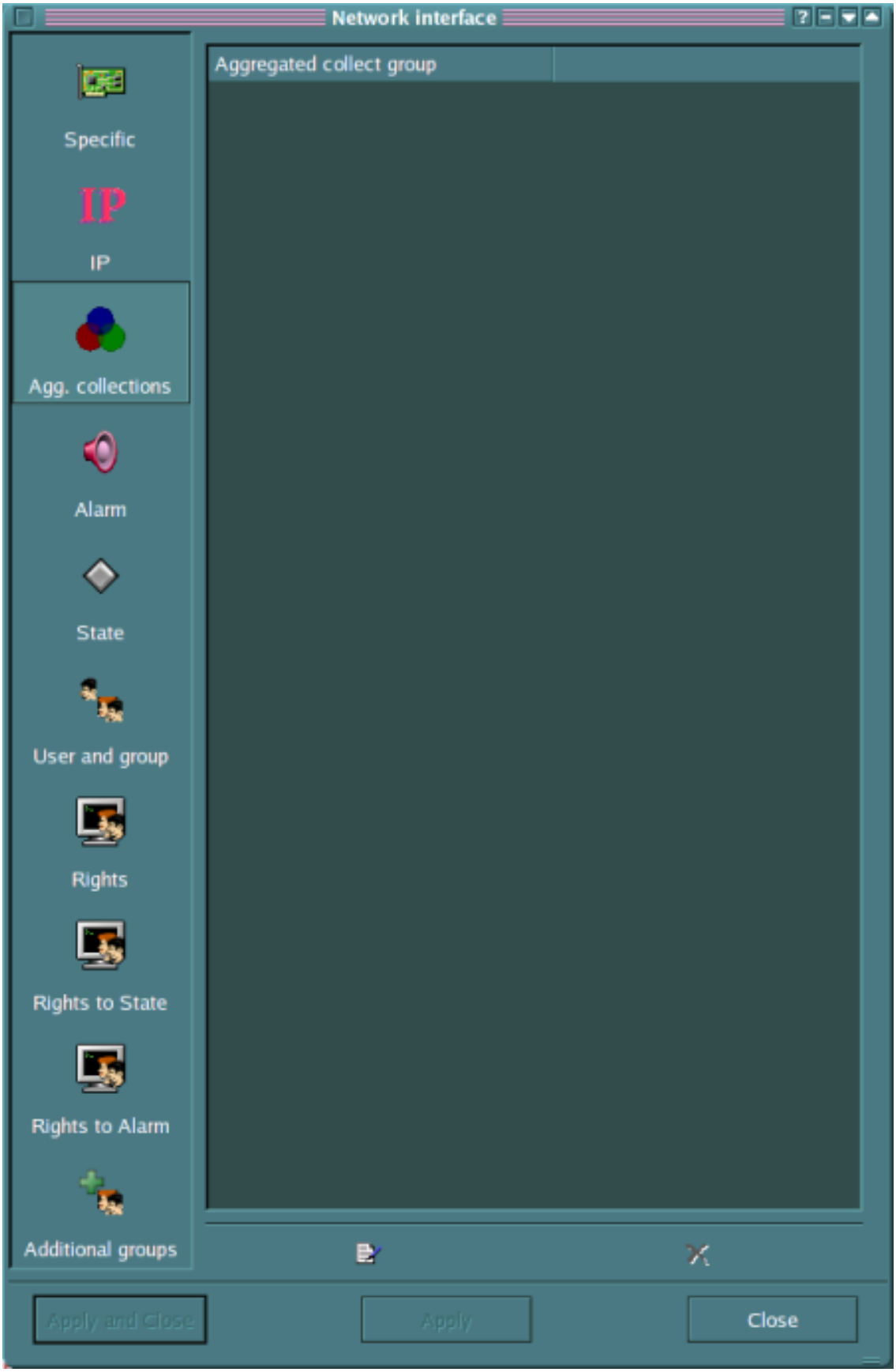


`Specyfic` tab includes fileds, that are characteristic for a network interface. The fields `ifIndex`, `Description` and `Alias` are accessible by SNMP protocol and they characterize an interface (`ifIndex` is an interface index in MIB of a device). `Alias` field can be set by a system user permanently. The field shouldn't be written at the next reading of device configuration, and in this case you should set `Update alias` option as `No`. `Type` field describes an interface type and `Admin status` and `Oper status` - a current operation and administration state (i.e.`UP`, `DOWN`, `UNKNOWN`). `Collect` field shows, if the interface is collected. A collection name is showed by `Collect group`. If `Update coll. avail.` field is set `Yes`, the collection is made automatically during updating of a device configuration. `Speed` and `High speed` fields show information about an interface speed. The value are updated automatically, when `Update speed` field has `Yes` value. `Main interface` field, for subinterface, shows physical interface, on which it is defined. For physical interface it shows itself, and then in the field description *myself* appears.



`IP` panel includes a list of IP addresses with masks of networks ascribed to the network interface.

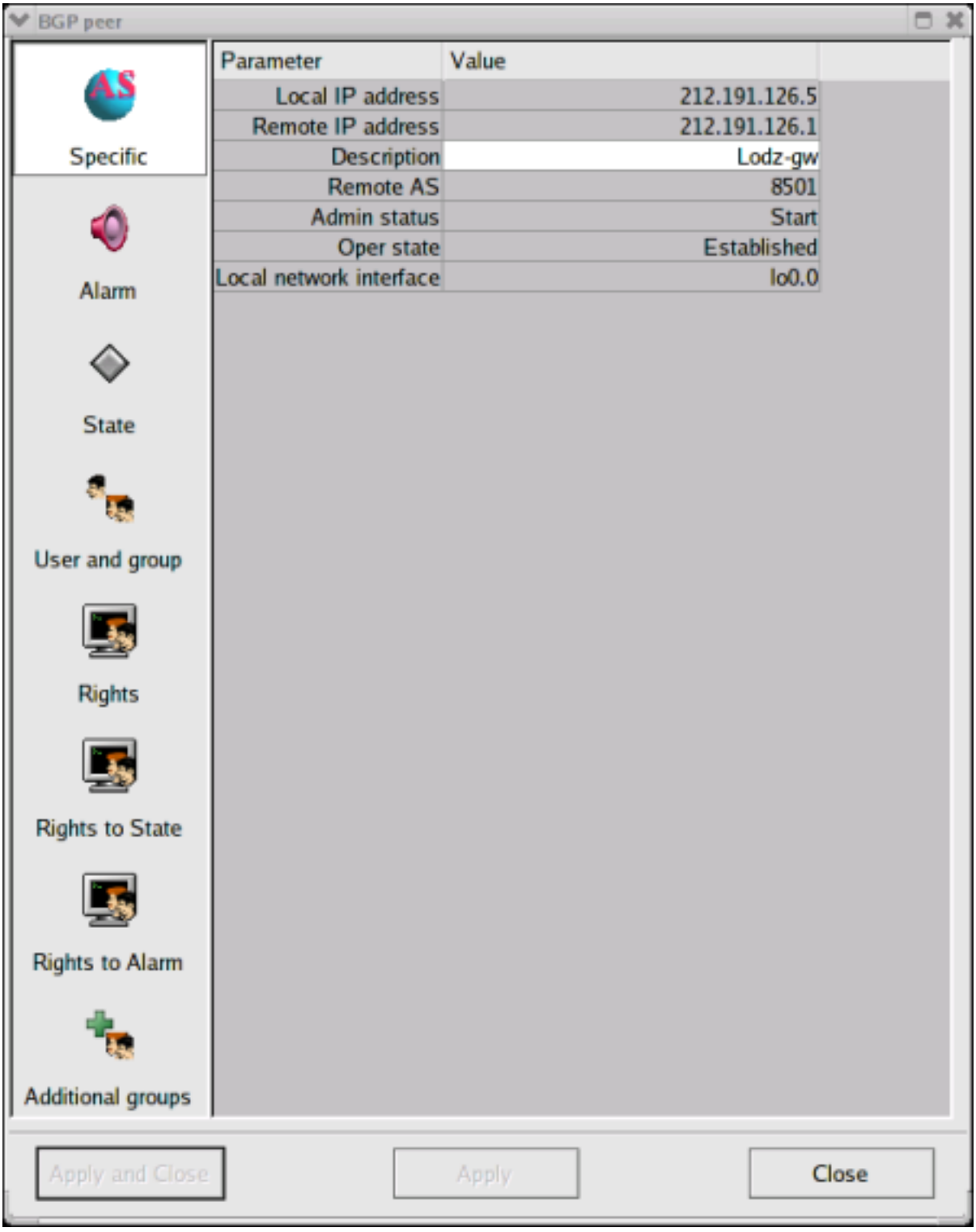
`Device IP` column shows, if a given address is treated as a device address or not. If yes, the device is polled by selected IP address. The device can have such address only one. `Forbidden IP` column shows addresses, that are never used during polling of the device. `Device IP` column shows, if a given address is treated as a device address. Then, the device is polled by a selected IP address. A davice may have at the most one the address. `Forbidden IP` column shows addresses, that are never used during polling of the device. If you want to change one of properties of a given address, you should click an address entry in a suitable column.


Network interface
Specific
IP
IP
Agg. collections
Alarm
State
User and group
Rights
Rights to State
Rights to Alarm
Additional groups
Aggregated collect group
Apply and Close
Apply
Close

`Agg. collections` tab shows a list of aggregate collections, to which an interface belongs to. The panel shows a list and its buttons, that allow to edit it. If you add a network interface to an aggregate collection, data collcectd for the interface increase suitable date of the collcection. So, it is a sum of them.

You can't see a list of all network interfaces available in **Network Information Database**. However you can see all interfaces of a given device, when you run a preview. To do this you should double click on a device symbol on the **Network Management Map** or select a device and choose suitable items of a menu or suitable button on the toolbar.

8.5.8. BGP peers configuration

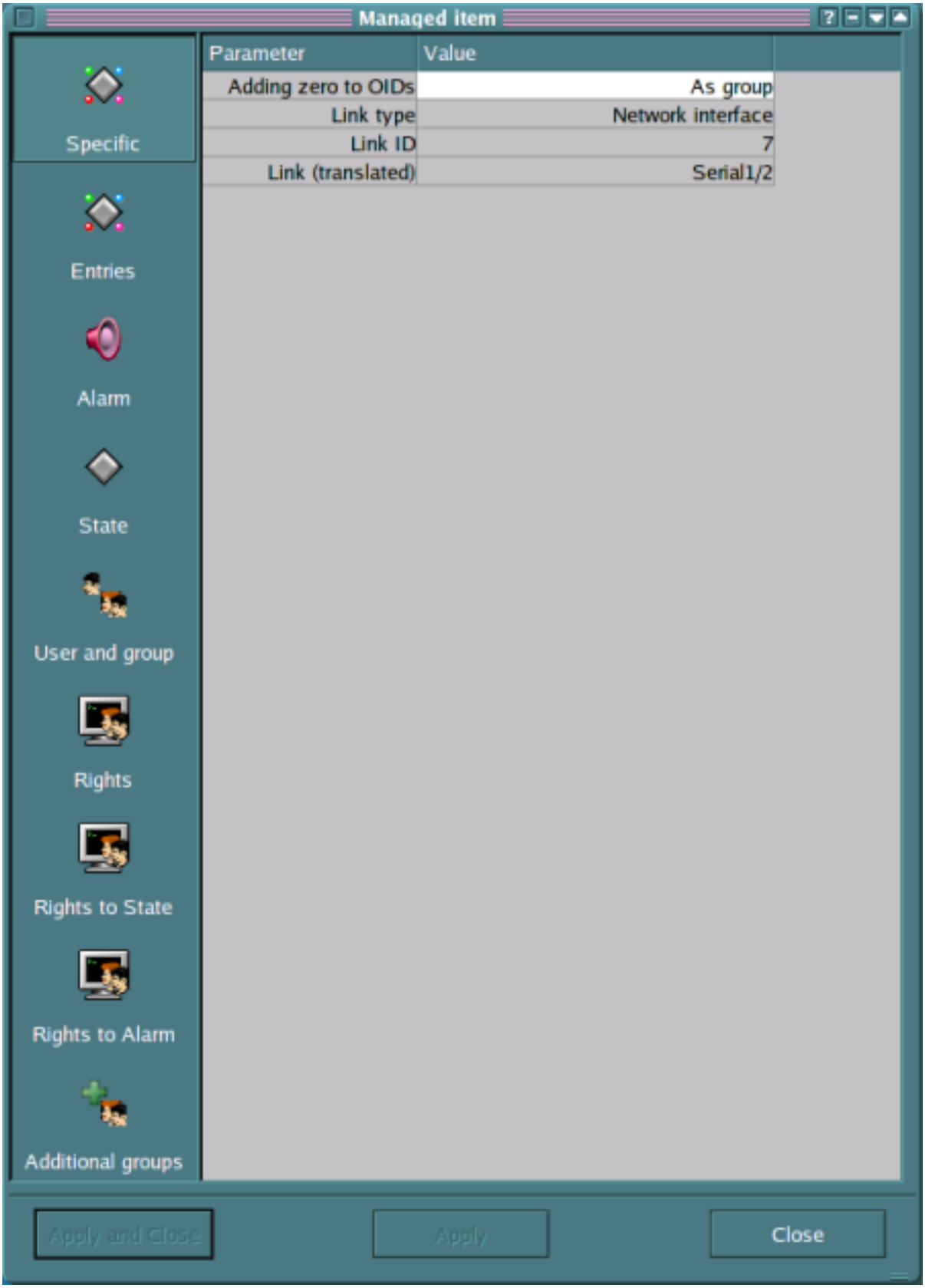


BGP peers always belong to physical routers discovered by dnmmsd server. They have many features and properties resulting from both their equivalents on physical devices and David system functionality. The field `Local IP address` shows a local IP address of defined BGP peer, and `Remote IP address` field - its remote address. `Description` field is set by a system user. `Remote AS` field

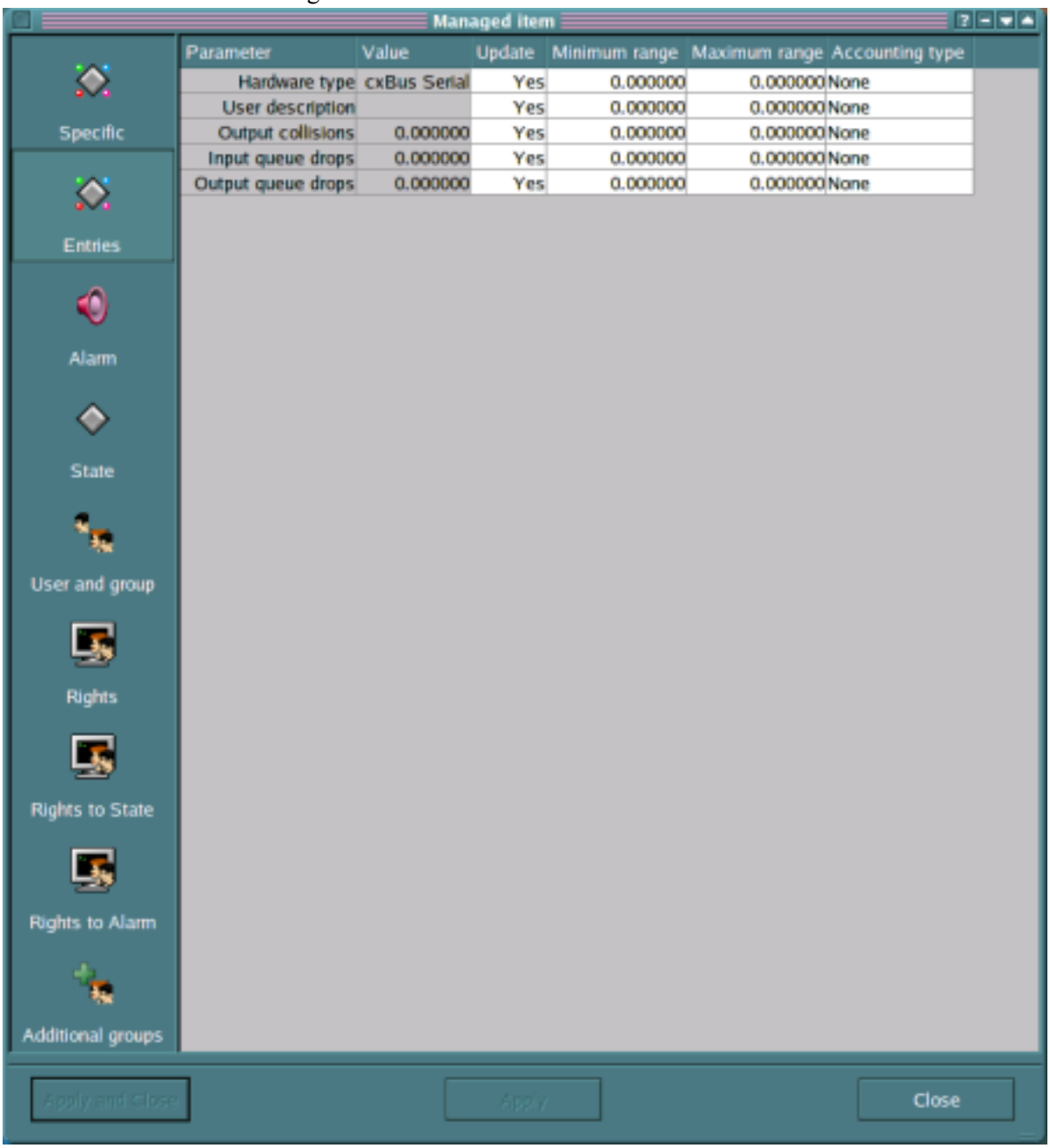
shows an AS number of a given peer while `Admin status` and `Oper status` a current operation and administration state (i.e. `ACTIVE`, `ESTABLISHED`, `UP`). `Local network interface` field shows a local network interface, that has IP address consistent with `Local IP address` field.

You can't see a list of all BGP peers available in **Network Information Database**. However you can see all BGP peers of a given router, when you run a preview. To do this, you should double click on device symbol on the **Network Management Map** or select a router and choose suitable items of a menu or suitable button on the toolbar.

8.5.9. Managed item configuration


Managed item
Specific
Entries
Alarm
State
User and group
Rights
Rights to State
Rights to Alarm
Additional groups
Parameter
Value
Adding zero to OIDs
As group
Link type
Network interface
Link ID
7
Link (translated)
Serial1/2
Apply and Close
Apply
Close

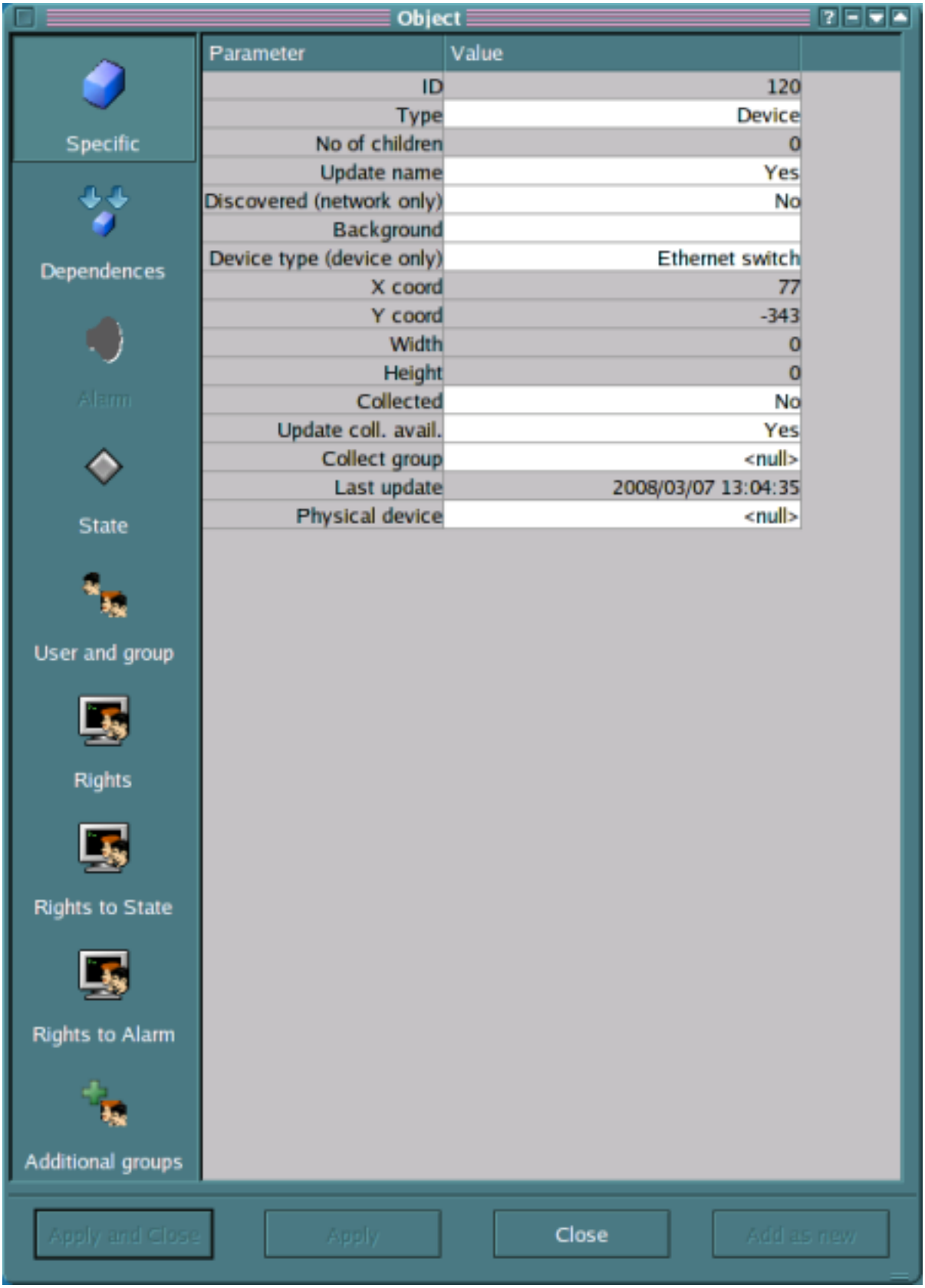
Managed items are results of scanning process of a physical device that concerns a given managed item group. Their properties and fields depend on a definition of origin group. `Specific` tab includes a few fields, common for all types of managed items. `Adding zero to OIDs` field shows if for the managed item is added '0' to its index during SNMP-GET operation (`On`), or not `Off`. A value which is used the most often is `As group`, i.e. a procedure consistent with the group definition. `Link type` field shows a link type if a given item is connected with other item, and `Link ID` field shows an identifier of a connected item. `Link (translated)` field is a resulted description of the item to which is connected a seclected managed item.



`Entries` tab shows a list of entries, that belong to the managed item. Definition of origin managed item

shows which one can be edited or not. The fields usually can be edited. Columns `Minimum range` and `Maximum range`, if they are different from each other, they overwrite ranges defined by the origin group. It lets you set a range of acceptable values for each item separately. `Accounting type` column lets you set type of accounting for each item. If it is `None`, this value is taken from a definition of a appropriate entry of the origin group.

8.5.10. Objects configuration


Object
Specific
Dependences
Alarm
State
User and group
Rights
Rights to State
Rights to Alarm
Additional groups
Parameter
Value
ID
120
Type
Device
No of children
0
Update name
Yes
Discovered (network only)
No
Background
Device type (device only)
Ethernet switch
X coord
77
Y coord
-343
Width
0
Height
0
Collected
No
Update coll. avail.
Yes
Collect group
<null>
Last update
2008/03/07 13:04:35
Physical device
<null>
Apply and Close
Apply
Close
Add as new

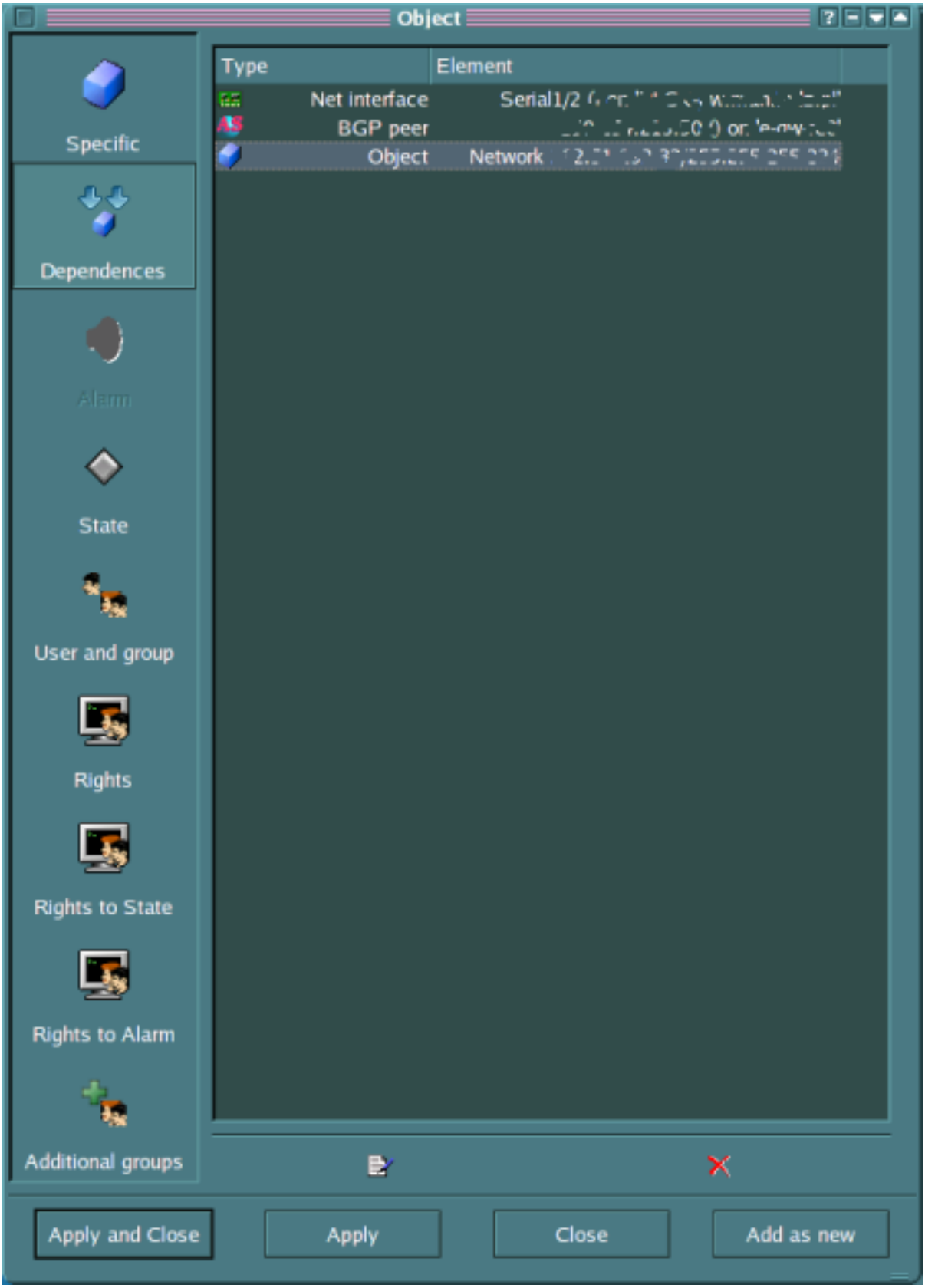
Objects can be both physical devices discovered by the system and symbols crated by a user for the system needs. The following types of objects are permitted: Network, Link, Device, Location. A user can create objects of Link or Location type. If you want to create an object of Location type, you should specify suitable fields to edition. If you want to define an object of Link type, you should select additionally two objects that are its ends. A new object always is added to a current presented map.

In Specific panel the field Type describes an object type. Update name shows if its name is updated duritng its update of the configruration, and No of children shows us how many children objects are included in the object. Discovered (network only) field concerns only objects of Network type and shows if a network is scaned to find new objects or not. Background field specifies a path to a local graphic file, that is a background of the object inside. The field Device type describes a device type, if the object has Device type. The field can have the following values:

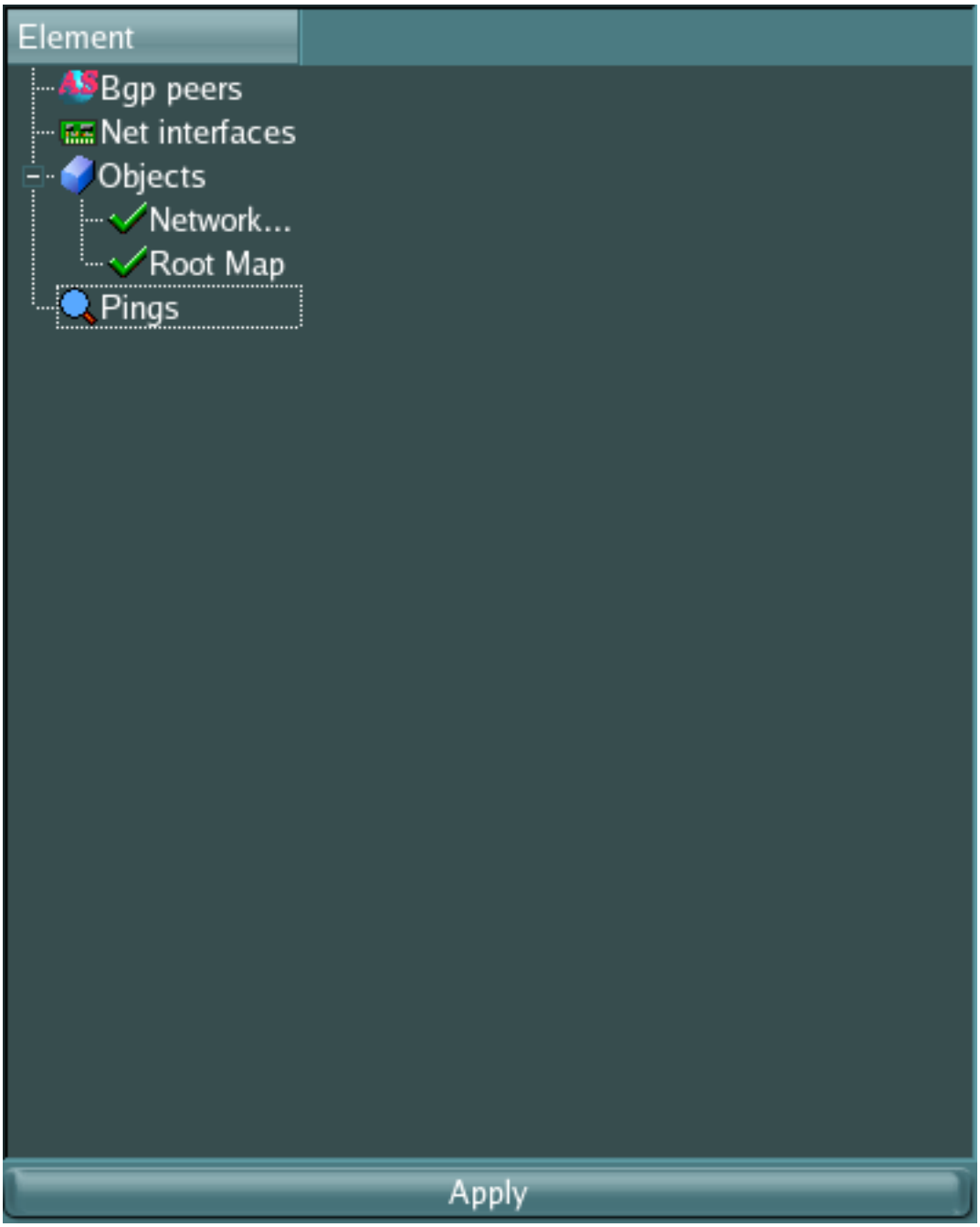
- Generic device;
- Computer;
- Router;
- ATM switch;
- Ethernet switch;
- Ups;
- Print server;
- Wireless device;
- Modem.

In the moment of discovering of a device by the system, it tries to find a correct value of the field, that next can be changed by a user.

The fields X coord and Y coord show an object location on a mother map. Collected field shows if collected data for the object are compatible with Collect group. Update coll. avail. field shows if during update of the configuration of a device, the best collection group will be searched for it every time. Last update field means for objects which reflect physical devices and the field show when the last update of a device had taken place using SNMP protocol. Physical device field shows an object which is a phisical device for a given object. Selecting in this field other object by a user is understood, that the edited object is set as a virtual object. The function is used when virtual routhers of Juniper company are monitored. In the case of Link object type you can use additionally two fields (Link from and Link to) describing objects that it connects.


Object
Specific
Dependences
Alarm
State
User and group
Rights
Rights to State
Rights to Alarm
Additional groups
Type
Element
Net interface
Serial1/2
BGP peer
Object
Network
Apply and Close
Apply
Close
Add as new

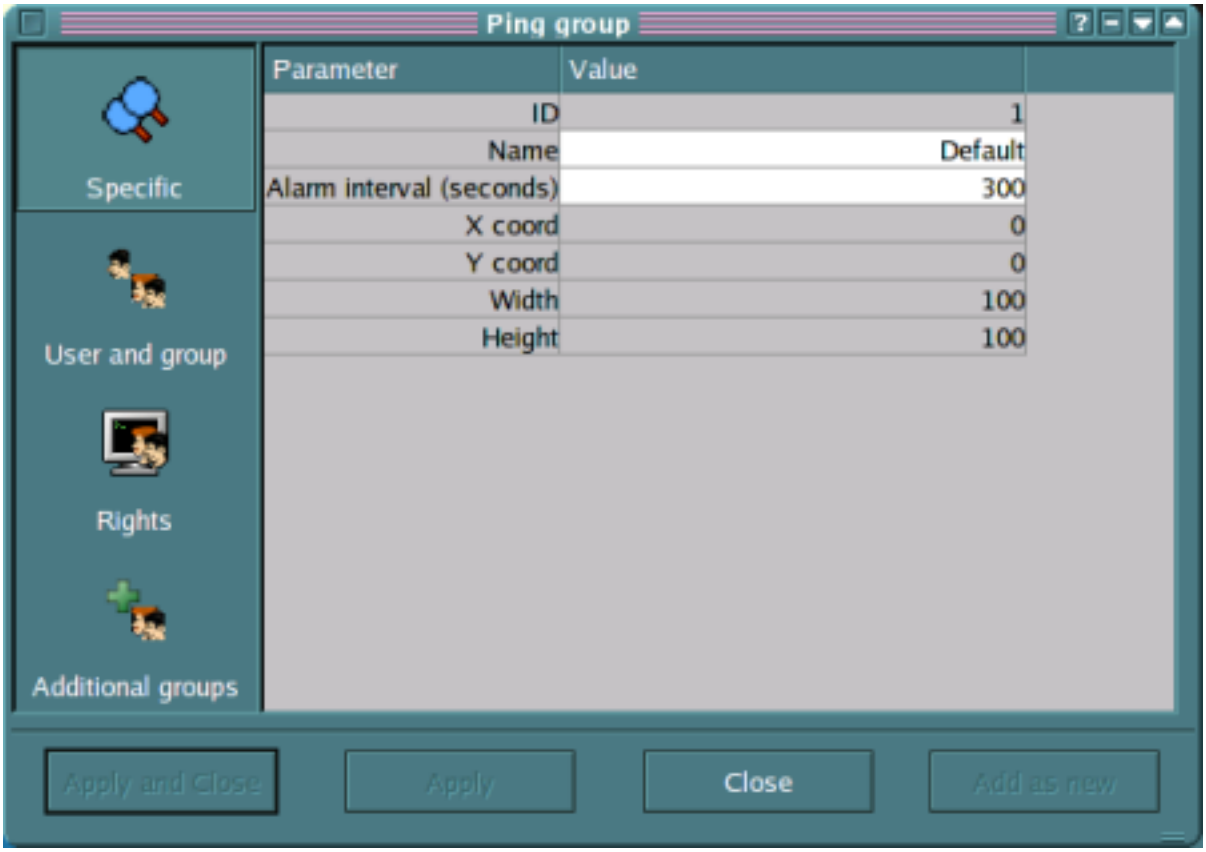
`Dependences` panel presents a list of items that their states influence on an object state. Besides the list, states of object children, network interfaces and managed items (if an object is a physical device) and BGP peers (if additionally the object is a router) influence on the object state. Two buttons are placed below the list. The first one helps you edit the list and second one lets you delete selected items. A list edition opens a dialog with the list of all current available items from `Network Information Database` that you can use in this place, and the dialog will be described below.



A list of items can be a multiple-choice or a single-choice. It is depended on a given context in which it was opened. An exploration of branches is depended on the same fact. If you want to apply that changes,

you should press the button `Apply`, otherwise you should press button `Escape` or click the mouse behind an area of the dialog.

8.5.11. A configuration of ping object groups



Ping object groups group objects in logical sets. Each group is displayed as a separate panel of objects. It defines any common properties for gathered objects. `Specific` tab has many fiels which meaning is described below:

- `Name` - a group name;
- `Alarm interval (seconds)` - it defines about what period of time alarms will be run for objects belongs to that group;
- `Number of packet` - a number of packets;
- `Packet sieze` - a size of packets;
- `Interval between packets (ms)` - it defines about what period of time single pakets in each burst (series);
- `Timeout (ms)` - timeout for a single packet;

- `Interval between series (sec)` - interval between next series of packets;
- `Log results` - it shows if response times will be logged or not;
- `Can send Traps` - it shows if SNMP Traps can be sent or not;
- `Bad status upper limit (%)` - upper limit of BAD status in percents;
- `Down status upper limit (%)` - upper limit of DOWN status in percents;
- `Ok limit for min (ms)` - a limit of acceptable minimum response times;
- `Ok limit for avg (ms)` - a limit of acceptable avarage response times;
- `Ok limit for max (ms)` - a limit of acceptable maximum response times;
- `X coord i Y coord` - X and Y coordinates of ping object panel;
- `Width i Height` - width and height of ping object panel.

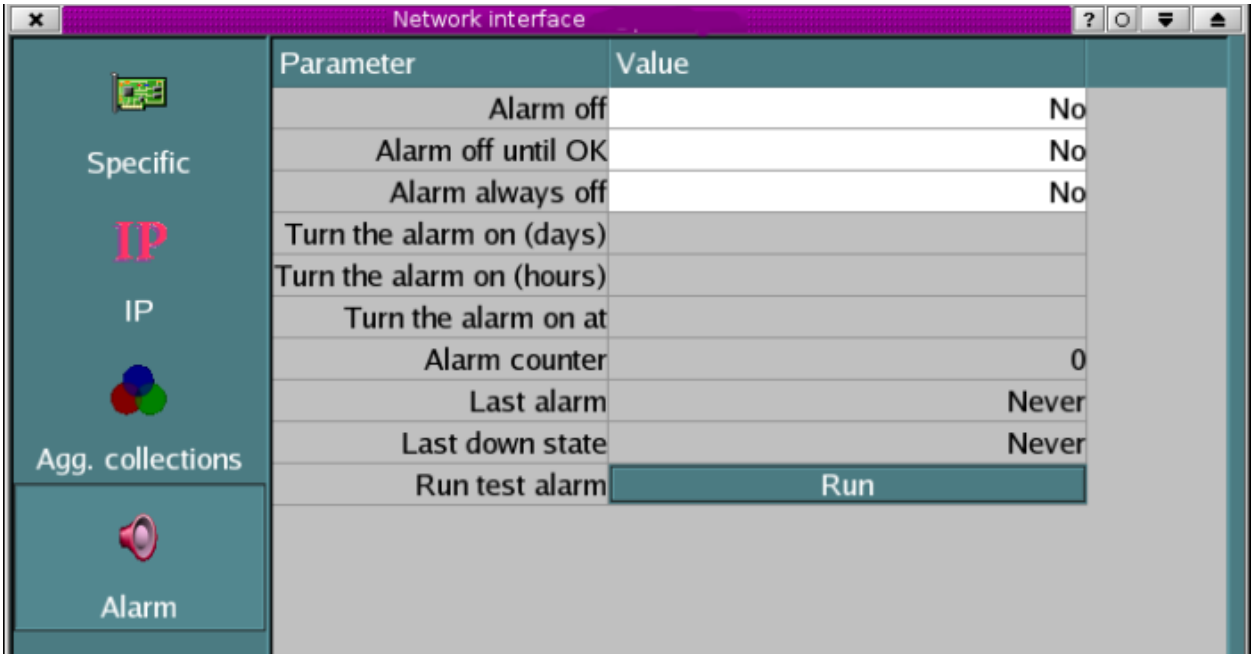
8.5.11.1. A list of available ping object groups

A list of ping object groups presents ping object groups sorted according to their names. Properties of items are shown in the columns. Selected items of a list can be edited in Ping groups window. Two buttons are placed below the list of items.

Table 8.24. A description of the Ping groups window buttons

Button	Description
	It opens `Ping group` window, that you can also open in the main view of the application.
	It allows you to delete selected items of the list. It can be more then one item.

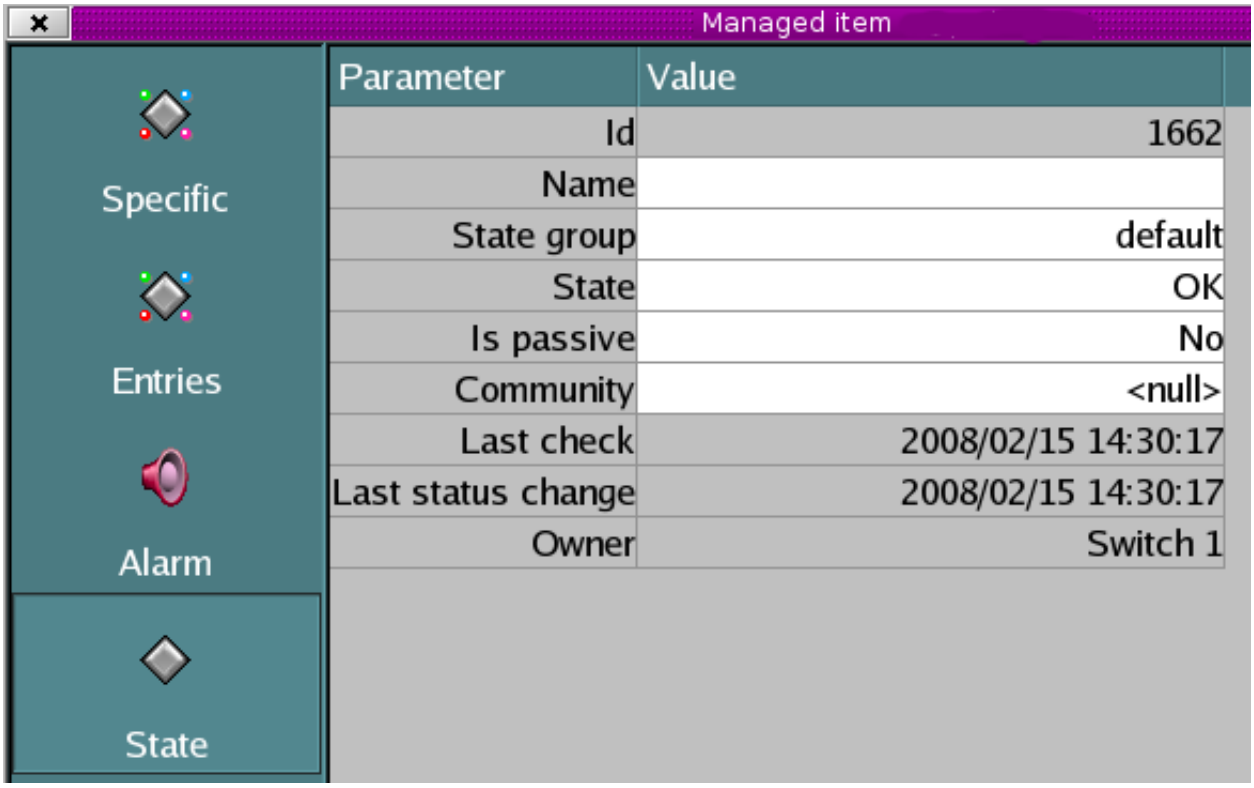
8.5.12. Item alarms configuration



A part of edited windows of different items has `Alarm` tab. There is data concerning alarms, i.e. programs, that are periodically run for items which state has `CRITICAL` value. The alarm programs are run by dnmmsd server. Frequency of running alarms is a property of a ping object group when the alarm is called for a ping object. For other items (network interfaces, BGP peers and managed items) the value is fixed and it is 5 minutes.

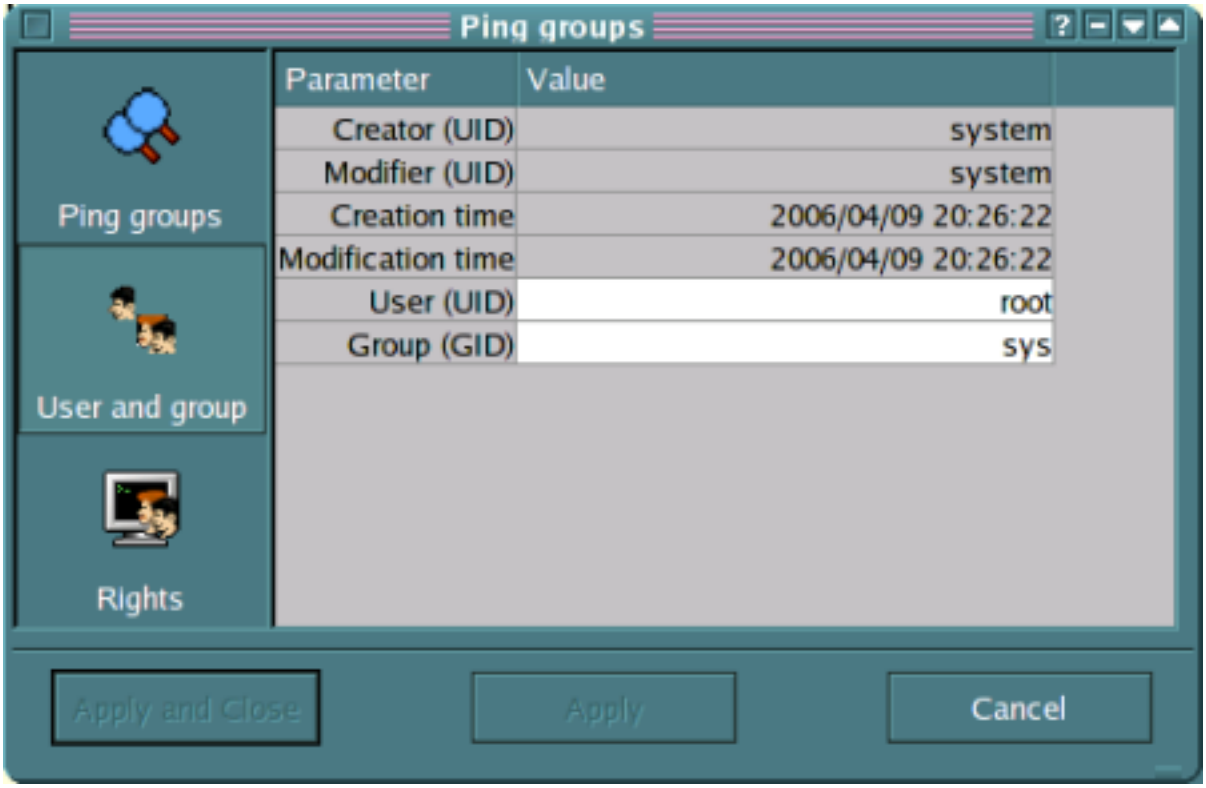
To turn off alarm for an object to a given moment, `Yes` value should be set for `Alarm off` field, and then in `Turn the alarm on (days)` field should be choosen a day and in `Turn the alarm on (hours)` field - an hour when the alarm is active again. `Turn the alarm on at` field shows additionally a day and an hour, when the alarm is activated. When `Alarm off until OK` field has `Yes` value, alarm will be active 15 minutes after last change of an object state from `CRITICAL` state. When `Alarm always off` field has `Yes` value, the alarm will be never run for the object. `Alarm counter` field shows a counter of run alarms for the item. `Last alarm` field shows last time of alarm for the item. `Last down state` filed shows last time of CRITICAL state for the item. `Run test alarm` field allows to run a test alarm for the item.

8.5.13. Item states configuration



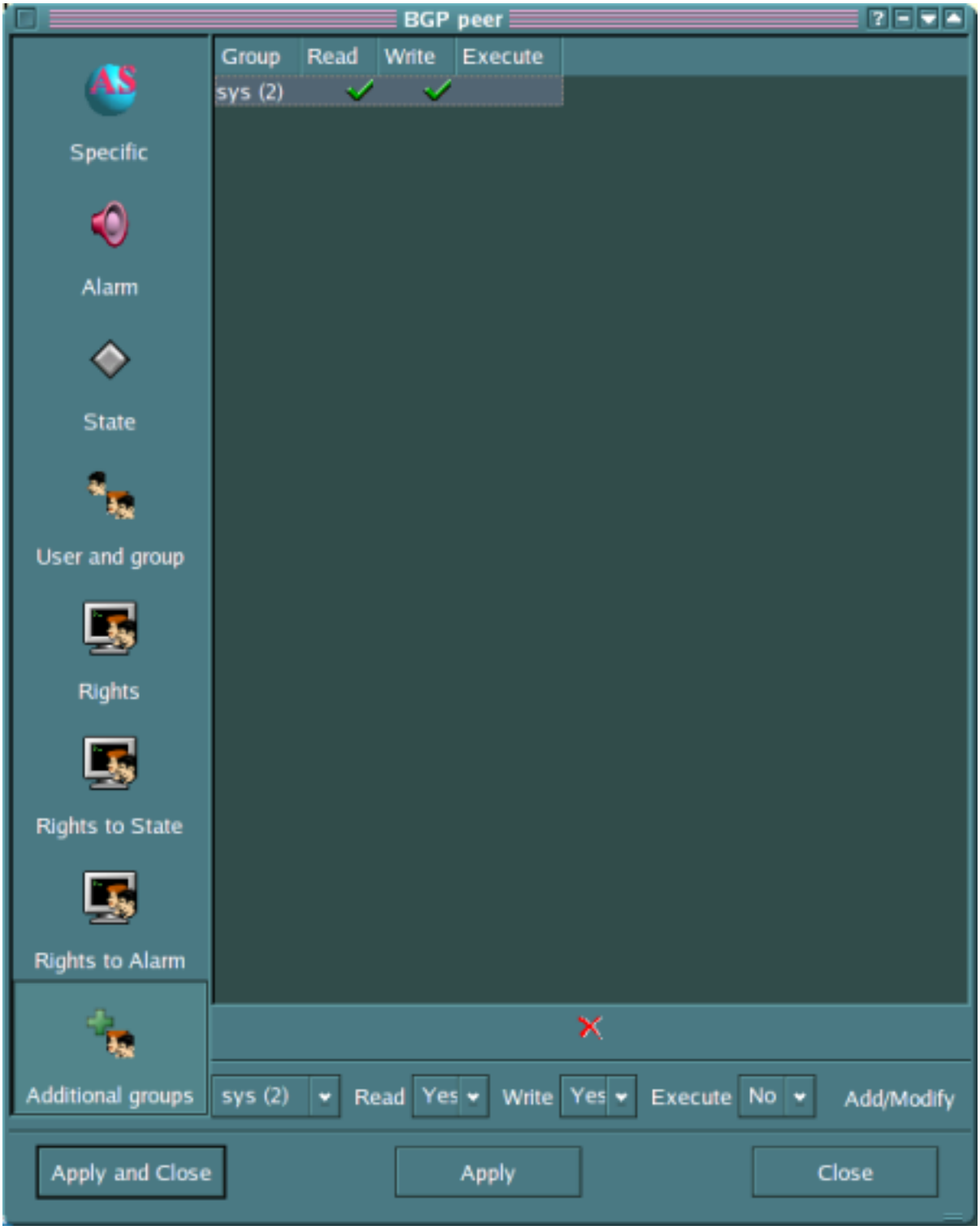
A part of edited windows of different items has `State` tab. It includes information about a state of the item and allows to specify some propeties, that are common for alomost all items. `Name` field includes an item name. For some types of items (i.e. BGP peers) the field is empty, because its value is unimportant. `State group` field shows a group of states to which a given item belongs to. `State` field includes an item state, and `Community` field can include a name of Community object assigned to the item. `Is passive` show if state of the item may be actively changed and affect state of its owner. `Last check` field shows when for the last time a given item state was chacked. `Last status change` field shows when for the last time change of a given item was observed. `Owner` field includes owner's data (a parent object) of an item.

8.5.14. A configuration of a user and an item group



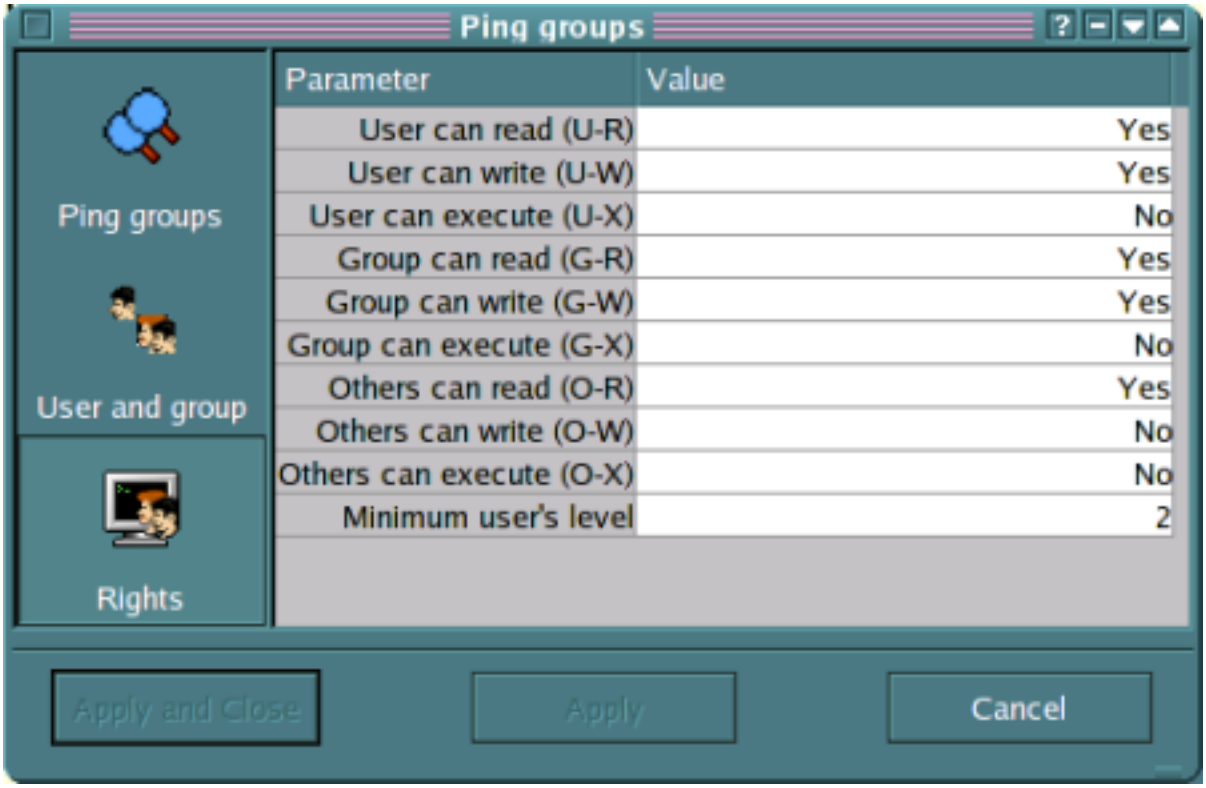
Edited windows of all items of **Network Information Database** include `User and group` tab. `Creator (UID)` field shows a creator of a given item, and `Modifier (UID)` field - the last person who modified an item. `system` value shows, that a creator or a modifier is dnmmsd server. `Creation time` field shows a creation time of an item, and `Modification time` field - its modification time. `User` field specifies an owner of the item, and `Group` field - a group to which the item belongs to.

8.5.15. A configuration of additional groups of item



`Additional groups` tab exists in edited windows of all items. It allows to add an item to additional groups of users and describes access rights for each group separately.

8.5.16. A configuration of access rights to the Database objects



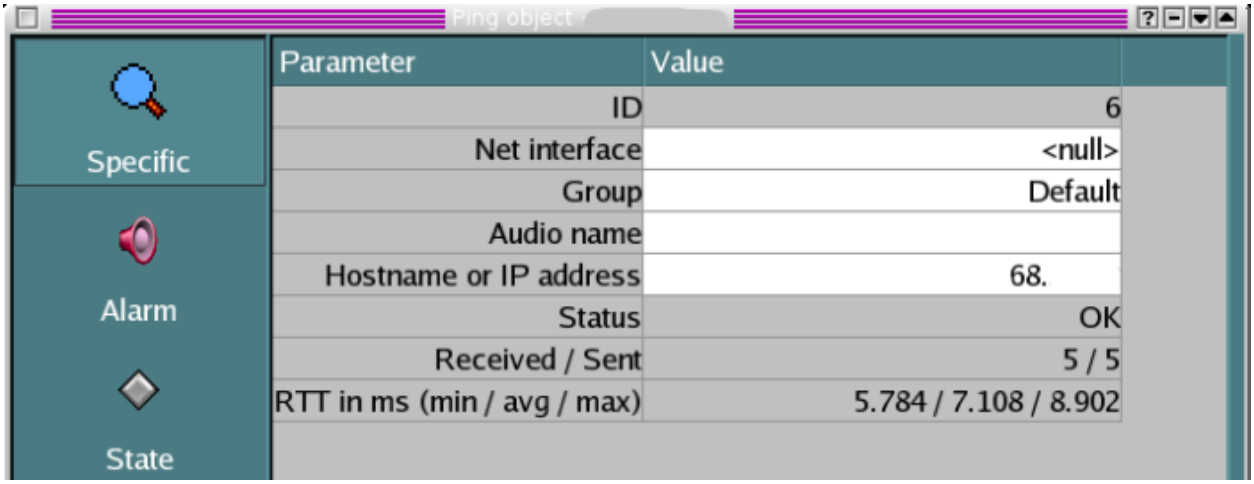
`Rights` tab exists in edited windows of all items. It allows to specify access rights to an item for an owner, a group or other users. Each of groups can have a right to WRITE, READ and EXECUTE of a given item. `Minimum user's level` field specifies a minimal level of a user that can modify an item. The word ‘minimum’ can be mistaken because a level 0 of a user is the highest level of rights, and next levels (it means n+1) define lower levels of rights.

8.6. Description of ping object panels

Each panel of ping objects agrees with groups to which the objects belong to. A name of the group is the window name. You can select many objects using the mouse and edit their properties. Double clicking on an object opens the edit window. If you click on the panel outside of any objects with the mouse, you will unselect all selected objects. You can move objects between panels (groups) in the easy way selecting objects and dragging them to a given panel (drag and drop).

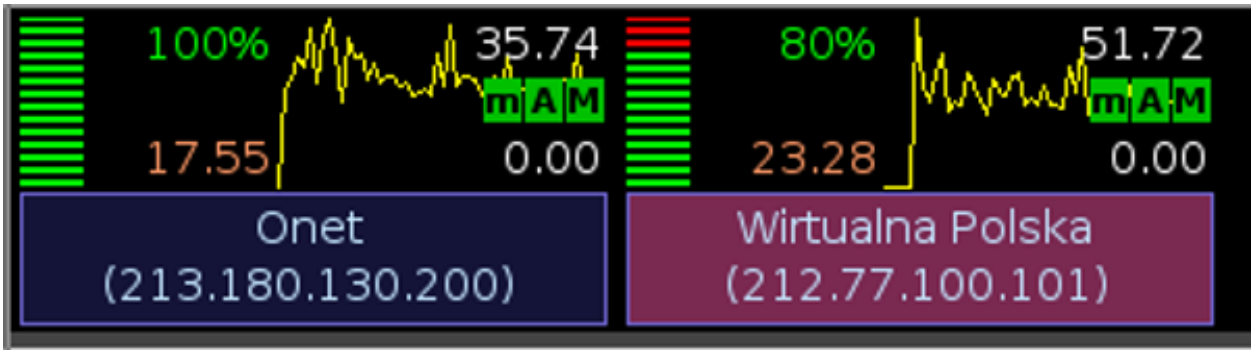
Ping objests represent IP addresses, to which ICMP ECHO packets are sent. The objects can be a result of discovering monitored devices by the system (one IP address will be choosen from among IP addresses of discovered device), or it can be created by the user directly.

If you click doble on ping object, the dialog window is shown. `Specyfic` panel presents basic properities of the object.



- `Net interface` - it shows in which interface of a given device, the object is connected. The field the most often shows end results of device discovering by the system.
- `Group` - it shows, to which group the object belongs to. Object belonging to the same group are presented insdie of the same panel. The groups may defines conservation of the objects which belong to them.
- `Audio name` - it allows to give a string, that is read by a speech synthesizer when an alarm for the object is run.
- `Hostname or IP address` - it shows IP address, to which ICMP packets are addressed.
- `Status` - it is only to read and it includes information about object status (a percent of responses/sent packets).
- `Received/Send` - number of received/sent packets.
- `RTT in ms (min/avg/max)` - round trip-time expressed in miliseconds (minimum/average/maximum).

Each ping object is represented by some graphic information about it.



There is a rectangular area in the bottom in which on the top an object name is placed and below it, in

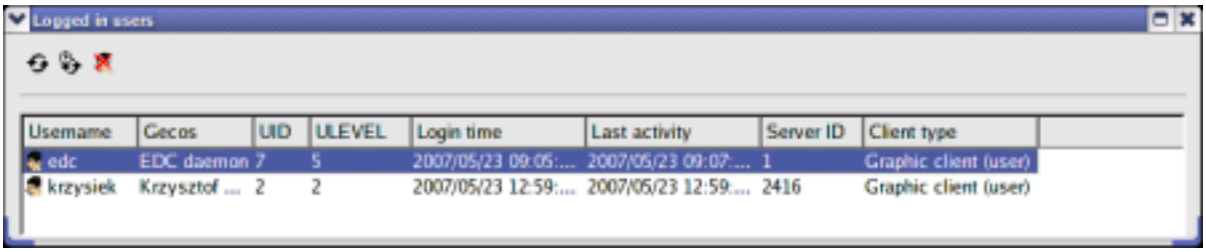
bracket, its IP address. A color of the rectangular area will change on purpure, when the objcet status is BAD. When the status is DOWN, the color will change on red and start flashing. When the object has Not managed state, the color will be gray.

The post from the left side shows a percent number of received responses. Lacking percent of responses that remains to 100%, is marked as a flashing part of the post. From the right side of the post is shown a percent of responses numerically in green color.

The rest part of the post is filled by a graph with three numbers describing it. On the bottom from the left side of the graph, the last middle responding time of a ping object is placed, and it is counted in milliseconds. The rest two values written in a white font describes a maximum (at the top) and a minimum (at the bottom) of the graph. Near the post crossing out symbol of a loudspeaker can appear. It shows that an alarm is turned off for a given object. A coloring of the symbol points at a kind of an alarm that was turned off.

There are three letters on the graph from the right side: m, A and M, that mean (m)Minimum, (A)varage and (M)aximum. They show if limits of response times are exceeded (red color) or not (green color). The limits are common for the group of ping object.

8.7. A window presenting a list of logged in users on dnmmsd server



The application allows you to preview users logged in dnmmsd server. The window has a few buttons allowing to some operations.

Table 8.25. Description of Logged in users window buttons

Button	Description
	It lets you refresh a list of logged in users.
	It turns on/off automatic refreshing of the list every 10 seconds.
	It lets you logout a selected user.

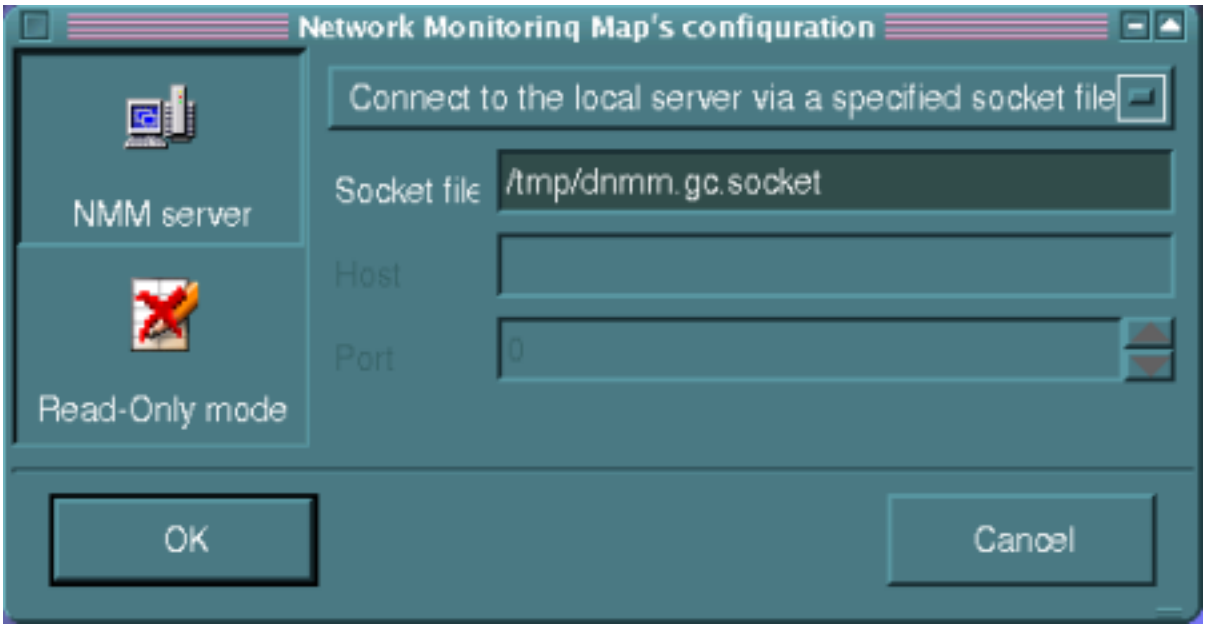
The particular columns of a user list describe:

- `Username` - a user's name;
- `Gecos` - a count description;
- `UID` - a user's identifier;
- `ULEVEL` - a right level of a user;
- `Login time` - login time of a given user (it can't be equal a connecting time with the server if an authentication took place more then one time);
- `Last activity` - time of the last user activity (for a given application instance) on dnmmsd server. It's understood as sending of any message to the server;
- `Server ID` - an identifier of a given application instance on dnmmsd server (the same user can be logged in using more then one the application instance).

8.8. A configuration of xdnmm application

The button  opens the configuration window which consists of two configuration panels.

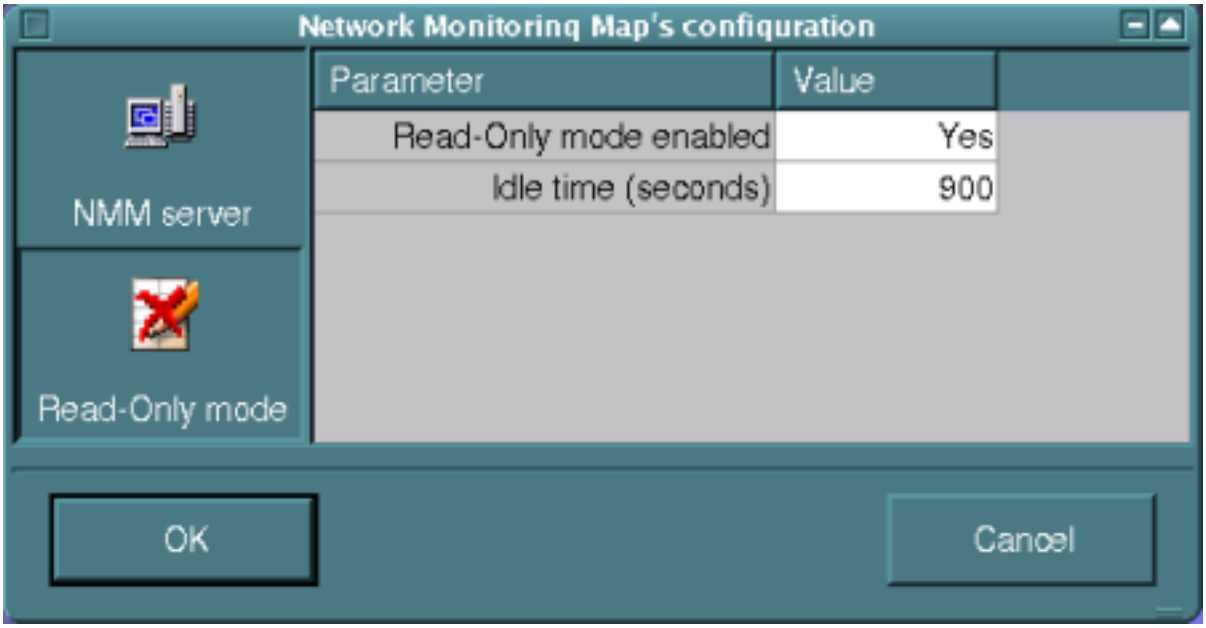
8.8.1. A configuration of dnmmsd server parameters



At the top of the dialog is placed a list with two options that lets you choose a way of connecting with dnmmsd server. The first one allows you to connect with the server by a socket file which a name you can write below. The second one lets you connect with the server by a computer network. Then you should pass a computer name or its IP address and a number of TCP port on which the server works on a given

computer.

8.8.2. A Read-Only mode configuration



`Read-Only mode enabled` field shows if a Read-Only mode is turned on or not. If it's active, it will run in a Read-Only mode when a time specified in `Idle time (seconds)` field goes on. If you want to make changes in **Network Information Database**, you should login again on dnmmsd server.

8.9. Related articles

Service of Network Management Map (dnmmsd)

Chapter 9. Network Nodes Viewer (xdnnv)

9.1. General

xdnnv application is **Network Nodes Viewer (xdnnv)** and it is a part of **Network Manager**. The application using a configuration file previously created by xdnnve generates the application which controls work of monitored network devices provided SNMP protocol. Applications generated in that way can be extremely different from one another both their appearances and their functionalities. However, there are some common functionalities for all such applications that let user have some control during their work. Control panels of network devices consist of controls that work is described in the section concerning xdnnve application.

9.2. Synopsis

xdnnv can be run with the following options: [*-V,--snmp-version version*] [*-d,--dump-packets*] [*--debug-snmplib*] [*-D,--no-communities*] [*-T,--no-toolbar*] [*-S,--no-statusbar*] [*-l,--log-facility log_facility*] [*-L,--log-level log_level*] [*-H,--host hostname*] [*-R,--read-community community*] [*-W,--write-community community*] [*-s,--stick sticking_string*] [*--oid-of-control control_name=oid*] [*--caption-of-control control_name=caption*] [*-v,--version*] [*-h,--help*]

9.3. Options

Table 9.1. xdnnv options

Option name	Description
-V,--snmp-version version	Use a specified SNMP: SNMPv1 (1) or SNMPv2C (2c) (default: 2c).
-d,--dump-packets	Dump information about each single SNMP packet in the ASN.1 notation on stdout.
--debug-snmplib	Turn on debugging of a native SNMP library, i.e.: currently library used by SNMP wrap library of David system.
-D,--no-communities	Don't display communities.
-T,--no-toolbar	Don't show the toolbar.
-S,--no-statusbar	Don't show the status bar.
-l,--log-facility log_facility	Choose log facility: daemon | user | local0 | ... | local7 (default: local6).
-L,--log-level log_level	Choose log level (on stderr and syslog) i.e. messages of selected level and more important levels will be logged: emerg | alert | crit | err | warning | notice | info | debug0 | ... | debug2 (default: warning).
-H,--host hostname	Specify a hostname you want to communicate (domain name or IP

Option name	Description
	address).
-R,--read-community community	Use a specified community for reading.
-W,--write-community community	Use a specified community for writing.
-s,--stick sticking_string	Stick this string to ID strings of selected controls.
--oid-of-control control_name=oid	Set a specified OID to a control that has a specified name.
--caption-of-control control_name=caption	Set a specified caption to a control that has a specified name.
-v,--version	Display version number on stderr and exit.
-h,--help	Display this help and exit.

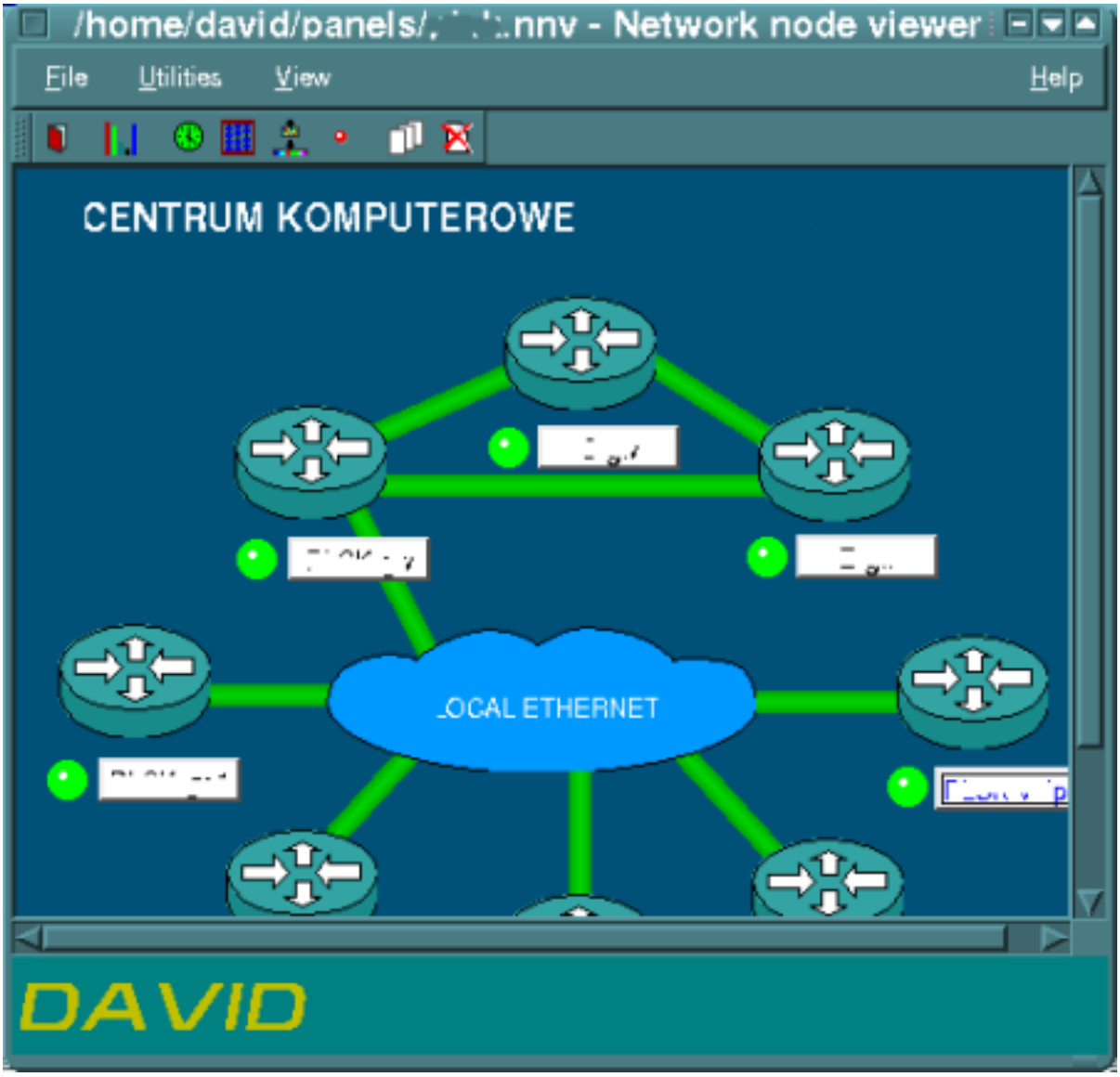
9.4. Description

One of the first step during building of a panel is creating sessions according to parameters saved in the configuration file. Each session owns some properties. Some of them can be unspecified or can get default values during the panel configuration by xdnnve. It concerns the following cases:

- if SNMP version is given as `Unspecified`, SNMP version of the session is set according to an argument of -V option of **xdnnv**;
- if no hostname is given, an argument of -H option of **xdnnv** becomes the hostname;
- if no SNMP community is given, an argument of -R option of **xdnnv** becomes the community for `Read` type sessions and an argument of -W option of **xdnnv** becomes the community for `Write` type sessions;
- if no sticking string is given, an argument of -s option of **xdnnv** becomes the sticking string.

After specification of the panel properties the following procedures come: creating controls, assigning to particular sessions such controls that want to be used in `Read` or `Write` actions and building expected connections between controls (i.e. Scroll bar, Slider). Next, lists of SNMP packets are built for particular timers of the panel. That SNMP packets are going to be sent to particular devices during the panel work.

After that initialization the panel starts its work according to timers settings saved in the configuration file and interacts with its user.



The application allows to control in same way work of a generated panel. Mostly, it can be a control of application timers and SNMP packets traveling between devices and the application and you can also trace `timeouts` for sending requests and other statistics.

Buttons on the toolbar help you to keep control of the panel work. The first button agrees with `File` menu:

Table 9.2. xdnnv - File and Utilities menu buttons

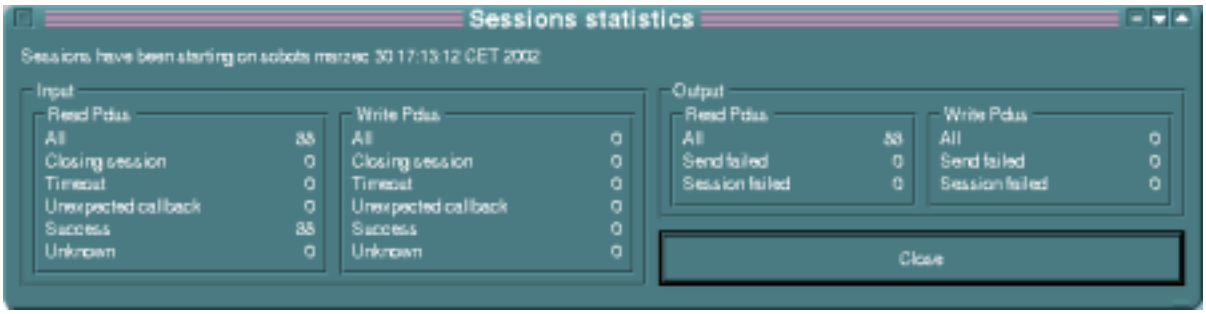
Button	Description
	It lets you exit the application.
	It opens `Sessions statistics` dialog that shows you current statistics of ingoing/outgoing packets.

Button	Description
	It opens `A basic timer list` tab of `Configure properities` dialog.
	It opens `Counters` tab in `Configure properities` dialog.
	It opens `Sessions and PDU-s` tab in `Configure properities` dialog.
	It opens `Timeouts` tab in `Configure properities` dialog.
	It lets you clear the contents of all controls on the panel.
	It allows you to clear timeout counters for all controls.

`View` menu includes two options - `Show tool bar` and `Show status bar` - they allow you to display or hide the toolbar and the status bar.

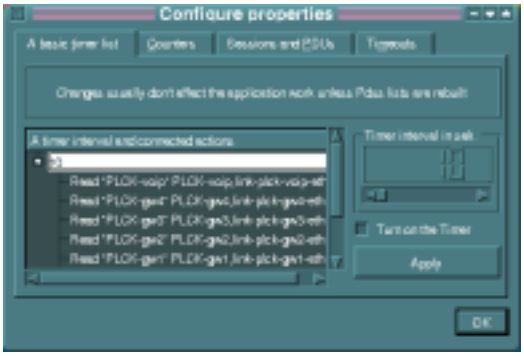
Through `Help` menu you may find out about the application version and its creation time.

9.4.1. Sessions statistics



The application counts ingoing and outgoing packets, occurred `errors` and `timeouts`. You can watch current statistics by `Sessions statistics` dialog. The panel creation time is displayed at the top of that dialog.

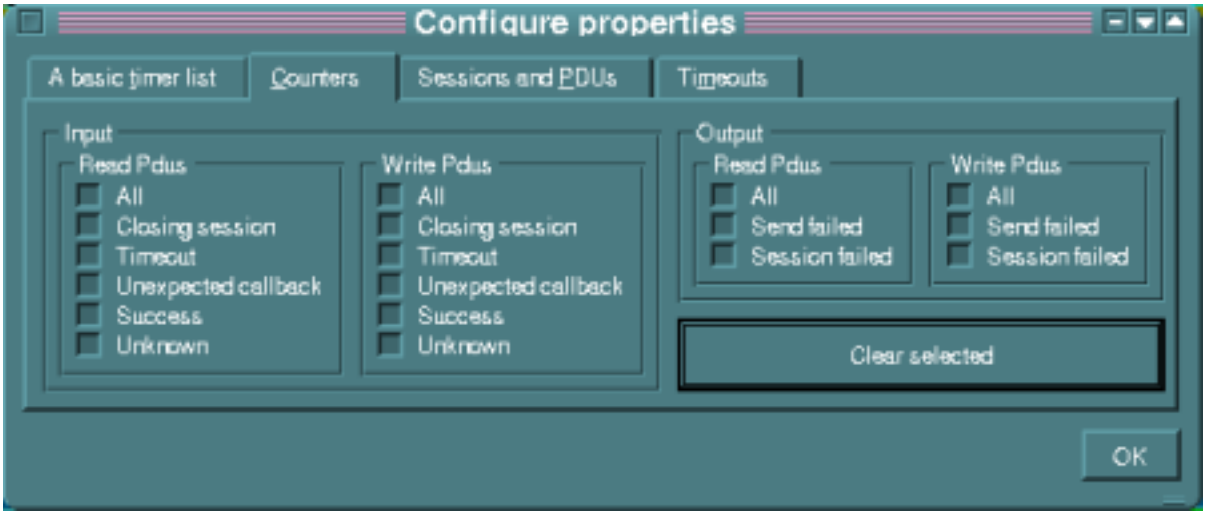
9.4.2. A basic timer list tab



Configuring by xdnnve the timers for the panel and connected with them `Read` and `Write` actions, you specify a timer interval on which particular actions will be run. If the interval is lower then 2 seconds, it

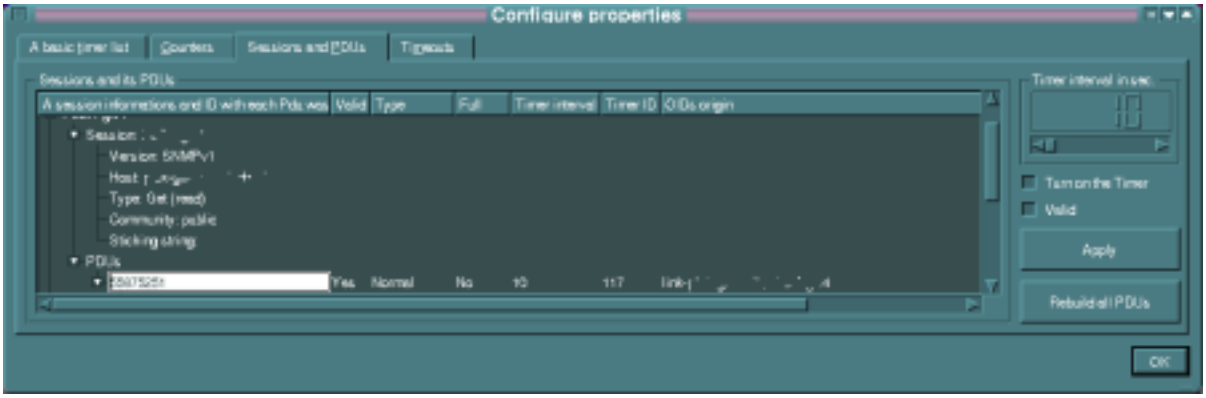
means that a given action will be run only one time on the beginning the panel work. `Turn on the Timer` option is set if the interval of a given timer is lower then 2 seconds. The timers intervals can be changed during **xdnnv** work but the effect will take place only after rebuilding of all SNMP packet lists. You may do this selecting the item of the timer list and setting its interval. Pressing `Apply` button lets you apply changes.

9.4.3. Counters tab



This tab allows to clear specified counters by selecting a specified kind of counters and pressing `Clear selected` button.

9.4.4. Sessions and PDU-s tab

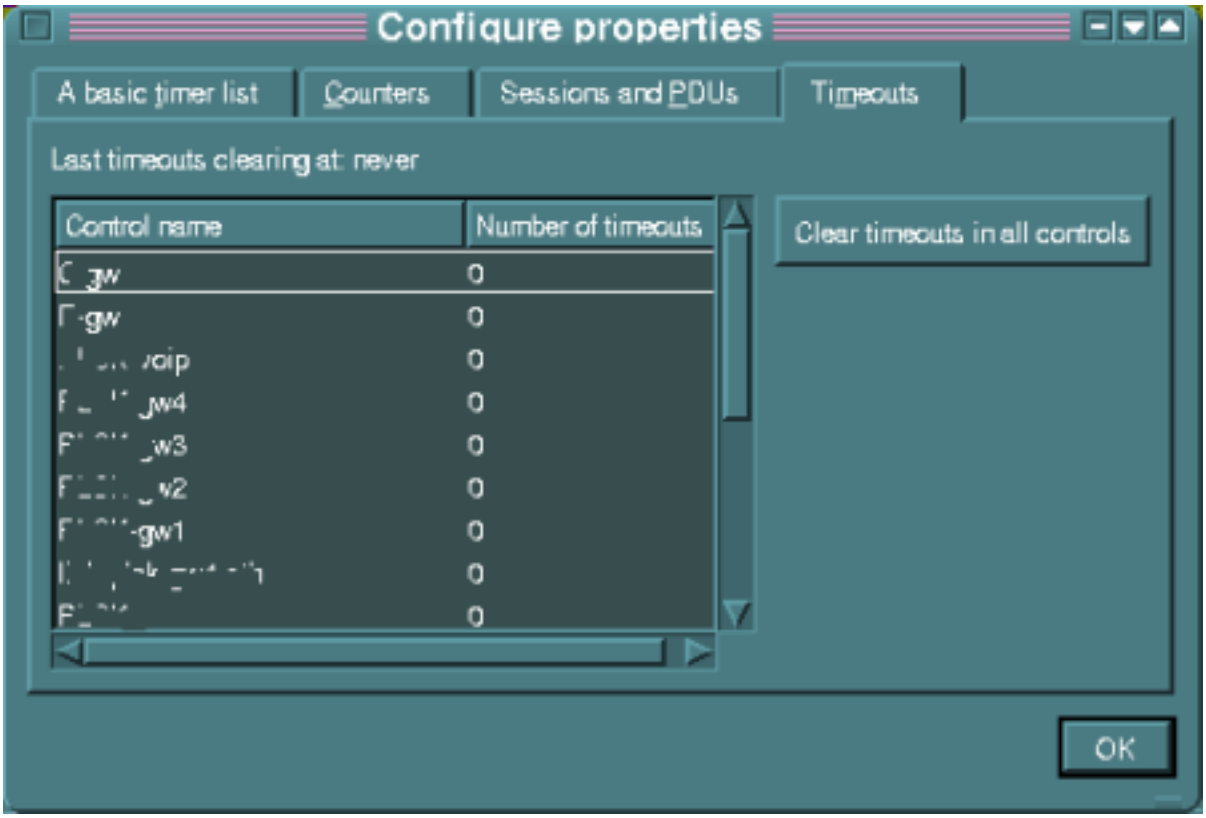


This tab lets you see current state of all built SNMP packets and find out to which session a given packet belongs and for request of which control it was built. Each packet has its identifier that appears in the first column of a presented list (`A sessions information and ID with each Pdu was sent last time and its content`). `Valid` column shows whether a given packet is valid or not (sent or not). `Type` column shows a type of packet (two possibilities: `Normal` and `Control`; the second type allows to control and build tables). `Full` column tells whether a given packet includes a maximum number of MIB identifiers (this value is predefined and it can't be changed). `Timer interval` column shows a number of seconds which elapses between sending this packet each time while `Timer ID`

column shows this timer identifier (an internal value of the application). The last column `OIDs origin` presents controls separated by commas for which requests a given SNMP packet was built.

Using `Timer interval in sec.` group, `Turn on the Timer` option and `Valid` option you can control a given SNMP packet according to meaning of a specified column of the list. Pressing `Apply` button confirms these changes. Pressing `Rebuild all Pdus` causes rebuilding of all SNMP packets for the panel.

9.4.5. Timeouts tab



`Timeouts` tab shows timeouts for particular controls. Pressing `Clear timeouts in all controls` clears timeout counters for all controls. At the top of the tab the last clearing time is displayed.

9.5. Related articles

Network Node Views Editor (xdnnve)

Chapter 10. Network Node Views Editor (xdnnve)

10.1. General

xdnnve is **Network Node Views Editor** and it is a part of **Network Manager**. This is a graphic tool application which is a kind of an editor for some resources such as control panels for network devices. The application allows you to build different configuration files. xdnnv on the base of the configuration files generates applications to manage control panels.

10.2. Synopsis

xdnnve can be run with the following options: [*-L,--log-level log_level*] [*-v,--version*] [*-h,--help*]

10.3. Options

Table 10.1. xdnnve options

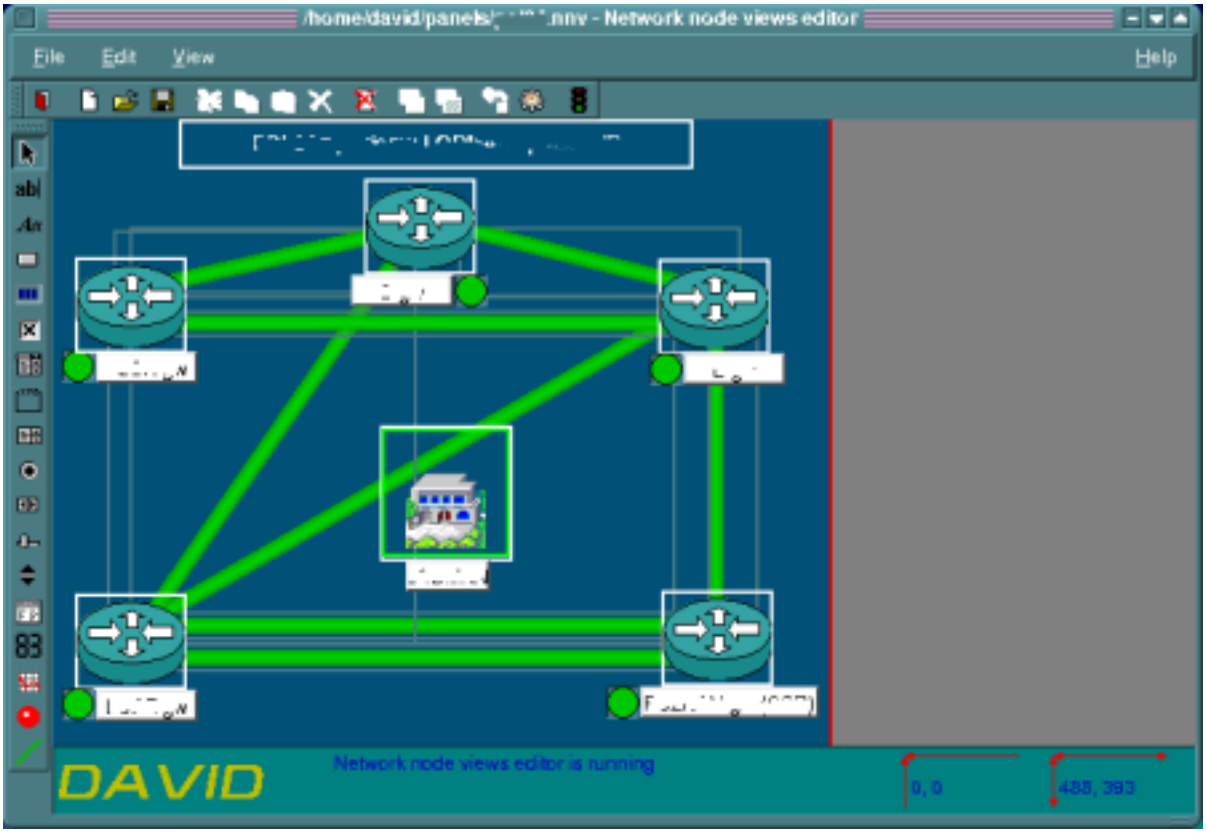
Option name	Description
-L,--log-level log_level	Choose log level (on stderr and syslog) i.e. messages of selected level and more important levels will be logged: emerg | alert | crit | err | warning | notice | info | debug0 | ... | debug2 (default: warning).
-v,--version	Display version number on stderr and exit.
-h,--help	Display this help and exit.

10.4. Description

10.4.1. Starting up and terminating the aplication

xdnnve reads its configuration parameters from `.xdnnverc` file during its startup. These parameters concern appearance of the program and other working parameters (i.e. the last edited file). The application expects to find its configuration file in a directory which name is kept in environmental variable *$DAVIDPRIVDIR*. When the file doesn't exist, the application begins its work with an empty document and with its default settings. When **xdnnve** finishes its work, it writes its current settings in the `.xdnnverc` file. The application, during its work, can write some errors on stderr instead of displaying message boxes.

10.4.2. Main window work



A configuration files edition, on which control panels are generated, is made invisible for a user. A user builds a panel putting controls on it by using the mouse and specifying their properties and connections. All operations are made using a graphic interface and a user doesn't need to know a structure of edited files.

The main panel takes a central part of the application main view. The toolbar is usually above the main panel while the status bar is placed below it. On the left side of the main panel tool buttons are placed. They help you to create your control panels.

Some information is displayed on the status bar: coordinates of a left, top corner and a size of an object over which there is the mouse pointer. When no object contains the mouse cursor, coordinates and a size of the main panel are displayed on the status bar. There is only one exception when a rectangle area is marked using the mouse. Then, the displayed data are applied to it.

10.4.2.1. Main window buttons

Buttons on the toolbar allow you to edit a created panel. The first four buttons agree with `File` menu.

Table 10.2. xdnnve - File menu buttons

Button	Description
	It lets you close the program. A user will be asked earlier for saving some changes of an edited file (if he has made any changes).
	It allows you to create a new configuration file.
	It helps you to open the configuration file.
	It allows you to save the edited file as the same file (in case of a new file a user will be asked for specifying its name).
`Save as`	It doesn't appear on the toolbar, and it allows you to save information to a file which name is specified by a user.

The edited operations applies to specified objects. When there aren't the objects, the edited operations applies to the main panel. An object is marked when the mouse cursor is above it (exactly: the object has a `focus`). An object permanently can be marked when any part of it belongs to a rectangle area of the main panel drawn by pressing the left mouse button. The specified object can be selected or unselected pressing the left mouse button while the `Ctrl` key stays pressed. Specified objects can be unselected by clicking the mouse on the main panel or marking a new area of it.

Rest of buttons agree with `Edit` menu and they are shown below:

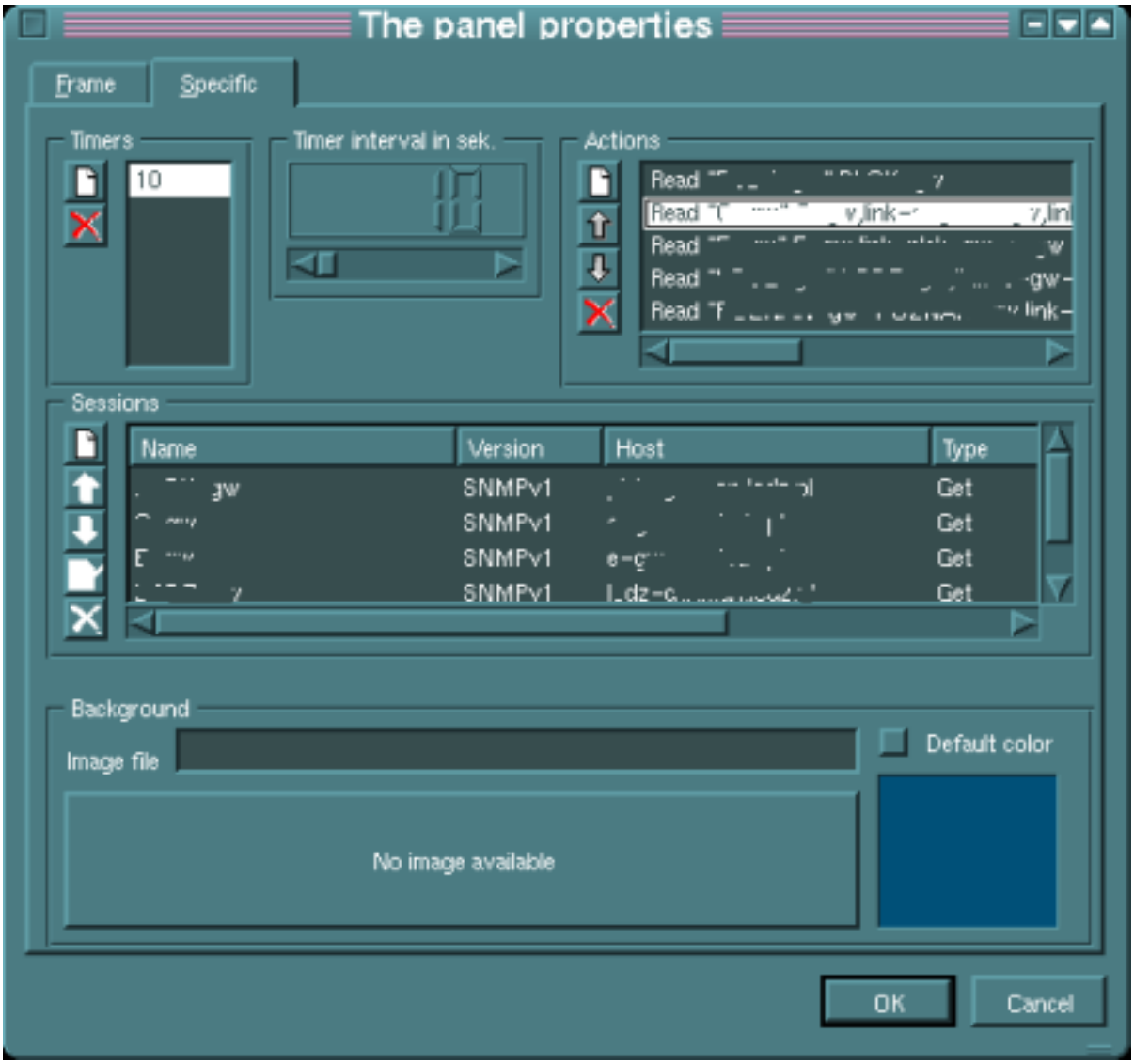
Table 10.3. xdnnve - Edit menu buttons

Button	Description
	It allows you to cut a specified item. A dialog will appears earlier and you will be able to confirm the operation. The dialog can also appear in the other cases.
	It lets you copy a specified item.
	It lets you paste a copied item.
	It helps you to delete a specified item.
	It allows you to delete all items of a created panel.
	It helps you to place an edited item on the first plan before other controls.
	It lets you place an edited item under other controls.
	It allows you to transform selected items to a specified type of a control.
	It helps you to configure properties of a specified control or panel.
	It allows you to test a created panel.

You can use `Edit` menu options by pressing the right mouse button on the selected control or the panel. Then a list with accessible options of the menu is displayed.

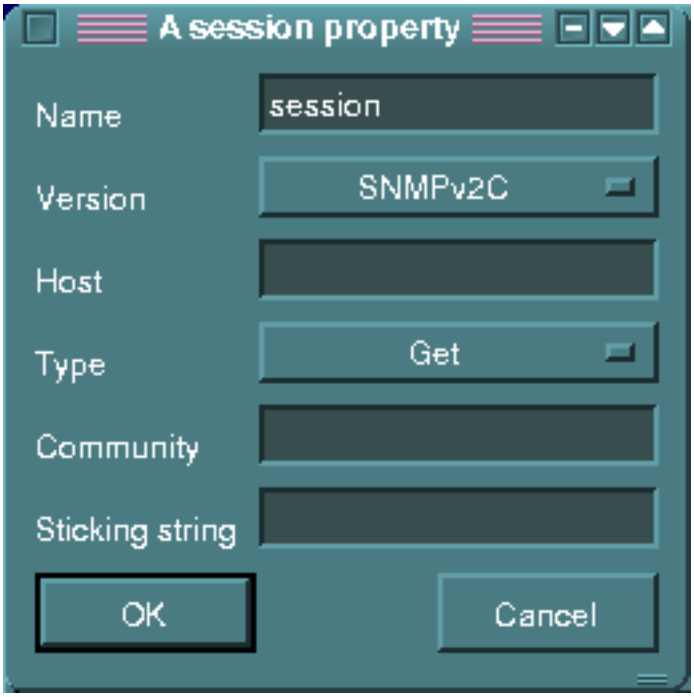
Through `Help` menu you can find out about the application version and its creation time.

10.5. Main panel configuration



A frame style of the main panel you can specify choosing Frame tab which was described later in next part of the document.

`Specific` tab allows you to define many parameters of the main panel that will influence a work of created control panel. You can set colors of the panel or choose its default color. You can set a name of a graphic file which will be the main panel background. If you want that the main panel will be able to poll every some period of time devices to update presented data, you should define at least one session. In this aim you should press the button in `Sessions` group to define a new sessions.



`A sessions property` dialog appears that you should pass a session name (it is only essential to identity within the panel), SNMP version, a name or an address of a device to which SNMP packets are sent, a type of requests (`Get` - to read an operation and `Set` - to write an operation) and a community. `Sticking string` field allows you to pass a text that can be used as the text to paste to OID-s' controls during actions are run. Fields that must be specified are: a session name (`Name`), a version (`Version` - choosing `Unspecified` field you don't have to choose a concrete version) and a version type (`Type`). During a generation of a real working panel unspecified fields are completed by input parameters of xdnnv. This fact applies when you create one panel for many devices of the same type where only the following fields will be changed: a version (`Version`), a name of device (`Host`), a community (`Community`).

In `Sessions` group of `Specific` tab the following buttons are placed:

Table 10.4. Sessions group buttons

Button	Description
	It let you move up a given item.
	It let you move down a given item.
	It lets you delete a selected item.
	It lets you modify parameters of selected session (then `A sessions property` windows is opened).

In `Timers` group you can add a new timer or delete an existed one. For each timer you can set an

interval counted in seconds. An interval lower then two seconds shows that the actions from `Actions` group, connected with this timer, are taken only one time in the moment of the panel creation. When you delete a given timer, you delete also the actions connected with it. If you choose a next timer while the choosing one earlier has got no actions, the timer will be destroyed automatically.

In `Actions` group you can set tools to manipulate an action list of a given timer. They allows you to move up and down items of a list, delete a given item of the list and add a new one. The button  lets you add a new item (an action) to the list and then a menu with two options: `Read` and `Write` is appeared. The first one includes a sessions list to a reading operation of values for specified objects to update them on the panel (an operation `Get` in SNMP) and the second one includes a sessions list to a writting operation i.e. sending values of specified object to monitored devices (an operation `Set` in SNMP). In the most of cases the second list is empty.

When you specify a given session you look `Choose objects` dialog that allows you to choose specified controls among all accessible in a given moment on the panel. After choosing controls a command of actions will be added to a list of a given timer.

10.6. Configuration of controls common properties

You can build a panel using controls which there are on the left side of the main view of the application. You should press only a selected control and click with the left mouse button above the area of the panel. When you draw a rectangle area instead of clicking the mouse, a control fills this rectangle area itself. You can use the controls as follows:

Table 10.5. Description of the controls

Control	Description
	Pointer

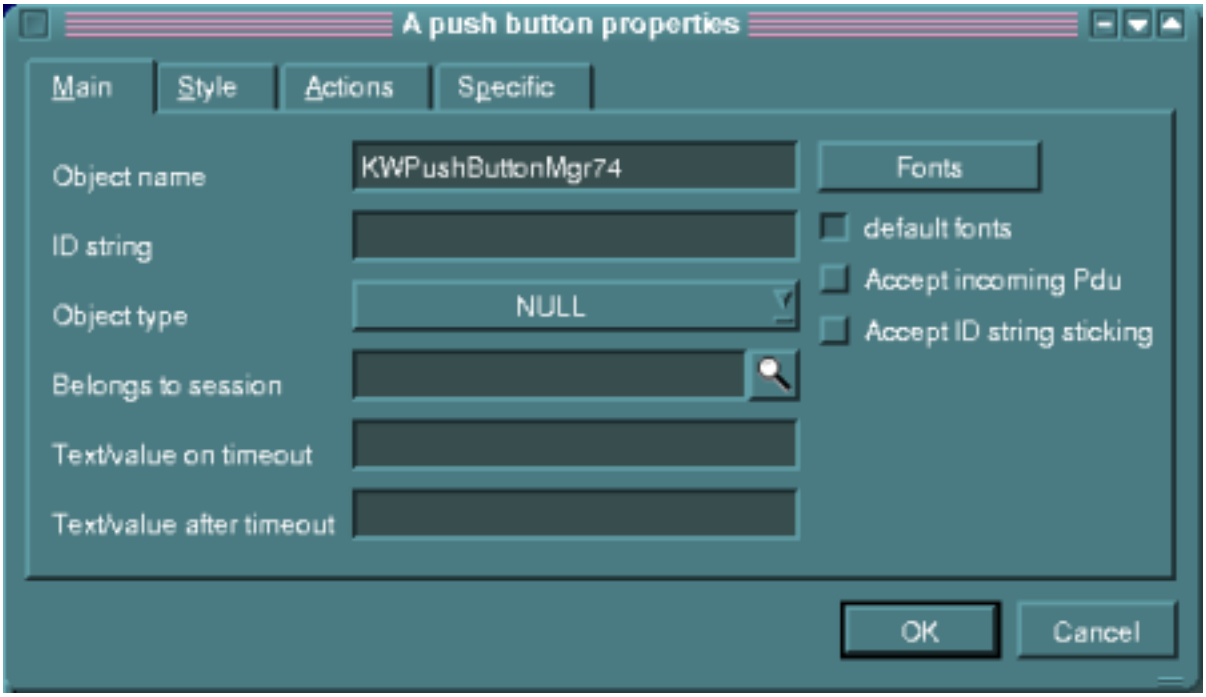
Control	Description
	Line edit
	Label
	Push button
	Progress bar
	Check box
	Combo box
	Group box
	List box
	Radio button group
	Scroll bar
	Slider
	Spin box
	List view
	LCD number
	Graph
	Control light
	Line

The first button isn't a button of a control but it allows you to unselect a pressing button of a given object.

If you press the button  , and choose any control or double click above the area of a control with the left mouse button, you can configure a choosing object. For all controls a window with 3, 4 or 5 tabs is appears. `Main`, `Style` and `Specific` tabs appear for each controls. Actions tab almost always shows while `Frame` tab shows only in same cases. A list of displaying tabs for particular controls is shown below in the table in the next part of the document.

`Main`, `Style`, `Actions` and `Frame` tabs look similarly for all controls (an exception is `Main` tab and remotely `Actions` tab) while `Specific` tab is different for each of controls.

10.6.1. Main tab

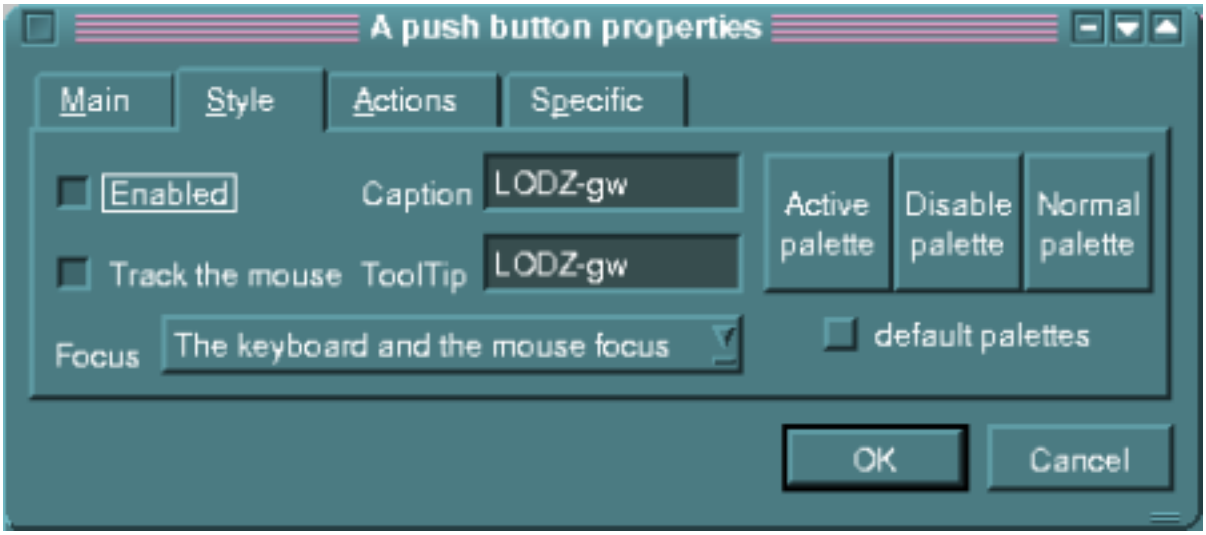


In `Main` tab is placed `Object name` field, that lets you specify an object name. An object with the name will be available for other controls and the panel. You can set also a kind of a font (`Fonts` button) different from default one. If an object is connected with SNMP i.e. if it is used as an argument of `Read` action or `Write` action, you can pass its `ID string`. If an object is used to set a value of SNMP (`Write` action), you must set a type of this variable by choosing an option of `Object type` field. You should set `Accept incoming Pdu` to update an appearance of the object basis of working results of `Read` actions. Sometimes you can want to paste an additional OID to `ID string`. Then you must set `Accept ID string sticking` in object properties. You should pass also a session name in `Belongs to session` to use an object in any action `Read` or `Write` type.

In `Text/value on timeout` field you can pass a value that a control will receive when it doesn't receive a response from a polling device in expected time. This value will be interpreted as a text when a control (i.e. `Label`) expects these values otherwise the value will be interpreted as a numerical value.

In `Text/value after timeout` field you can pass a value that a control will be received after timeout when it doesn't receive a response from a polling device.

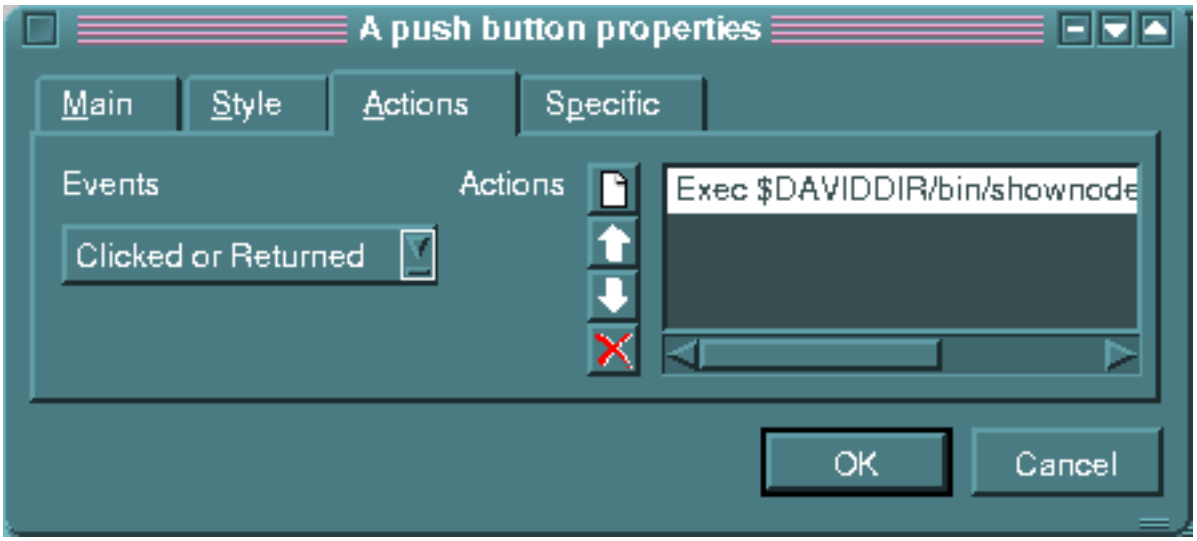
10.6.2. Style tab



`Style` tab includes a field of choice that gets you four possibilities of receiving of `Focus` for a given controls: a lack of focus, focus by pressing `Tab` key, focus by pressing the mouse button and two last options together. In `Caption` dialog you can pass an inscription which is displayed on a given control (it concerns a part of objects). In `ToolTip` dialog you can pass a text which is displayed in an response dialog above the object when you drag the mouse cursor on it (empty inscription means a lack of a response). Turning on `Enabled` option allows you to set a given control in an accessible state by the mouse or the keyboard.

Three buttons are placed on the tab and they helps you to set free color for any controls. Each of the buttons let you specify color which is characteristic for a state in which any controls can be (active, passive and normal).

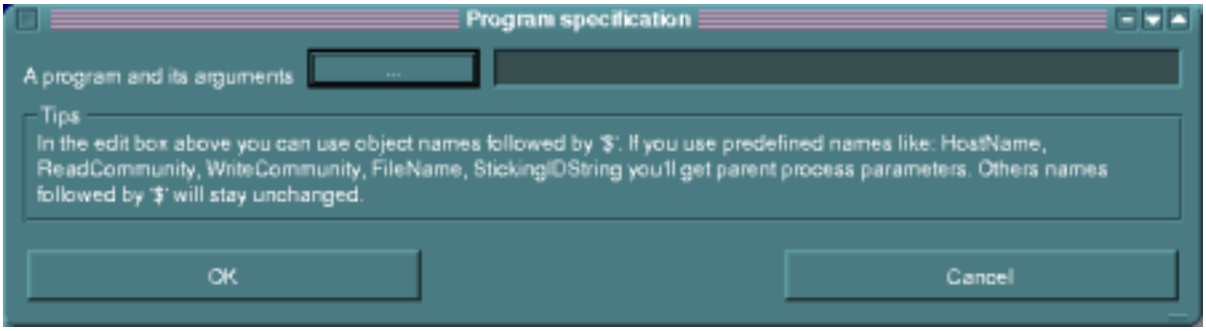
10.6.3. Actions tab



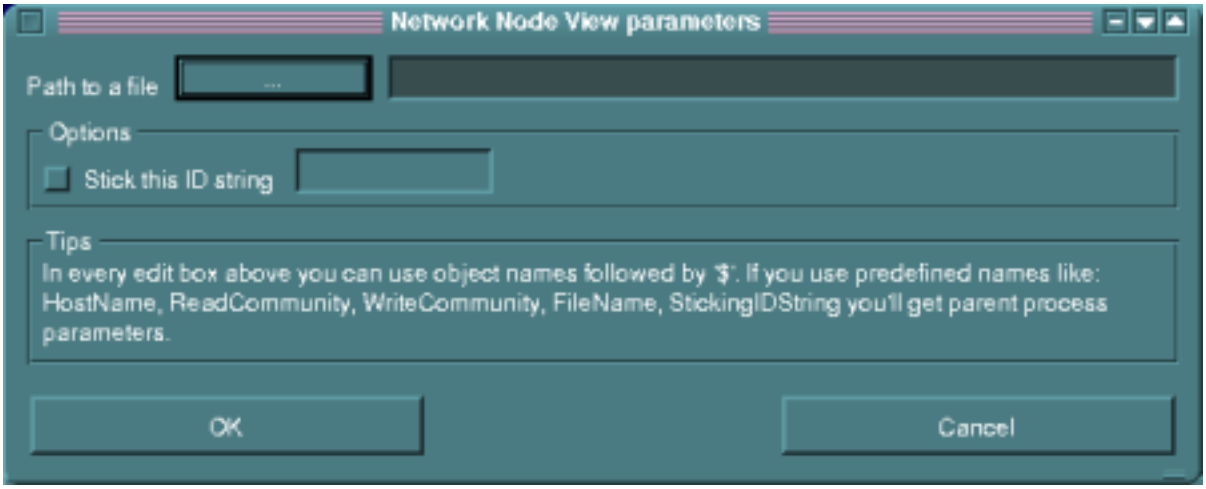
In `Actions` tab for each event you can specify a list of actions that will be done. Events for controls mostly can be in two kinds: pressing the left mouse button while the `Enter` key stays pressed and double

clicking the left mouse button. The second event in the case of the mouse always must be preceded the first one. It can cause a situation that actions joined to a double clicking cannot occur or they quickly occur after actions for a single clicking. For this reason you shouldn't define actions for this both events in the case of the same object.

A user has five different kinds of actions at his disposal. Only some type of objects have all kinds of actions. `Close` action allows you to finish a work of the panel. If the panel is like a dialog, it only closes itself. If it is like a main panel, a work of the whole application is finished. `Read` action lets you read data of specified objects to update their current view on the panel. `Write` action passes this data to a managed device.



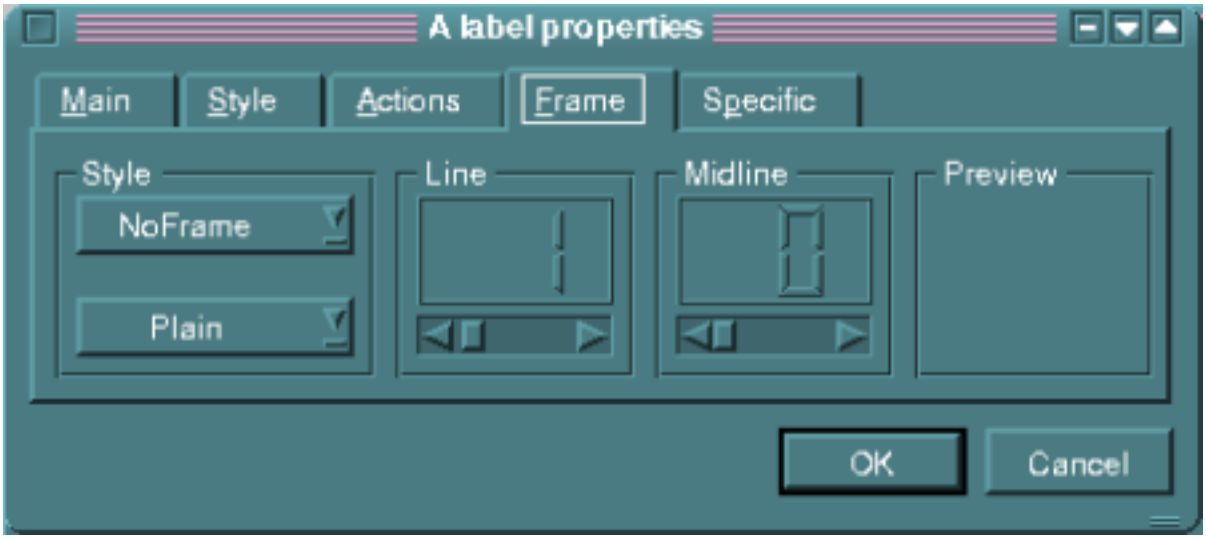
`Program specification` Program specification window defines `Exec` action and it allows you to specify the program and its parameters that will be run during a given event is done. When you specify a program and its parameters, you can pass names preceded `$` sign. A name can be: an environmental variable that will be defined during the panel is working, a control name which a value we want to use, a keyword as HostName, ReadCommunity, WriteCommunity, FileName, StickingIDString including appopriate input parameters of xdnnv.



`Network Node View parameters` window defines `Dialog` action and it allows you to pass names of a configuration file for xdnnv describing the main panel. You can choose `Stick this ID string` option and pass OID that will be pasted to controls of a given panel. As in case of `Exec` action, you can use `$` sign and pass after it names of controls or keywords (you can't use environmental variables). `Dialog` action runs a service of a specified panel in a modal dialog in contrast to `Exec` action which runs some program.

The buttons of `Actions` tab allows you to manipulate a list of actions for a given event. The button  lets you define a new action among described above. You can specify `Read` and `Write` actions in the same way like the panel configuration.

10.6.4. Frame tab



`Frame` tab lets you configure a frame style of an object and its line width that a frame will be built. In `Style` group you have some possibilities at your disposal: `No Frame`, `Box`, `Panel`, `WinPanel`, `HLine`, `VLine`, `Styled Panel`, `PopupPanel` and frame styles as: `Plain`, `Raised` and `Sunken`. In `Line` group you can define an outside line width of a frame and in `Midline` group - a middle line width. An effect of current settings is shown on the preview.

10.6.5. Tabs for particular controls

Table 10.6. Tabs for particular controls

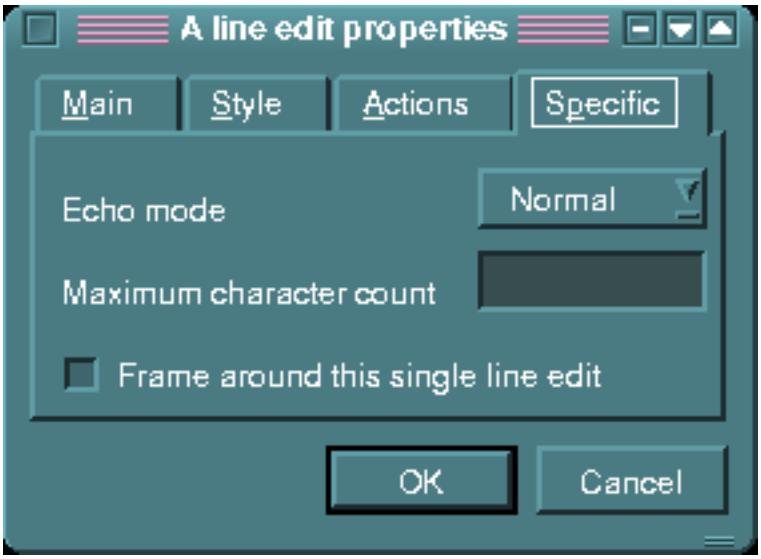
Control	Main tab	Style tab	Actions tab	Frame tab	Specyfic tab
ab| - Line edit	✓	✓	✓		✓
Aa - Label	✓	✓	✓	✓	✓
- Push button	✓	✓	✓		✓
- Progress bar	✓	✓	✓	✓	✓
- Check box	✓	✓	✓		✓
- Combo box	✓	✓	✓		✓
- Group box	✓	✓		✓	✓
- List box	✓	✓	✓	✓	✓

Control	Main tab	Style tab	Actions tab	Frame tab	Specyfic tab
- Radio button group	✓	✓	✓	✓	✓
- Scroll bar	✓	✓	✓		✓
- Slider	✓	✓	✓		✓
- Spin box	✓	✓	✓	✓	✓
- List view	✓	✓		✓	✓
- LCD number	✓	✓	✓	✓	✓
- Graph	✓	✓	✓	✓	✓
- Control light	✓	✓	✓	✓	✓
- Line	✓	✓	✓		✓

10.7. Specific tabs for particular type of controls

10.7.1. The Specific tab for Line edit control

- **Line edit** control allows you to edit a text of a single line.



`Echo` field of `Specific` tab lets you set if during a text edition the characters be able to write (`Normal`) or not (`No Echo`) and whether asterisks be able to display instead of a text (`Password`). The tab helps you to set a maximum value of characters (`Maximum character count`) that can be

written by a user. You can also choose if `Line edit` is surrounded with a frame (`Frame around this single line edit`).

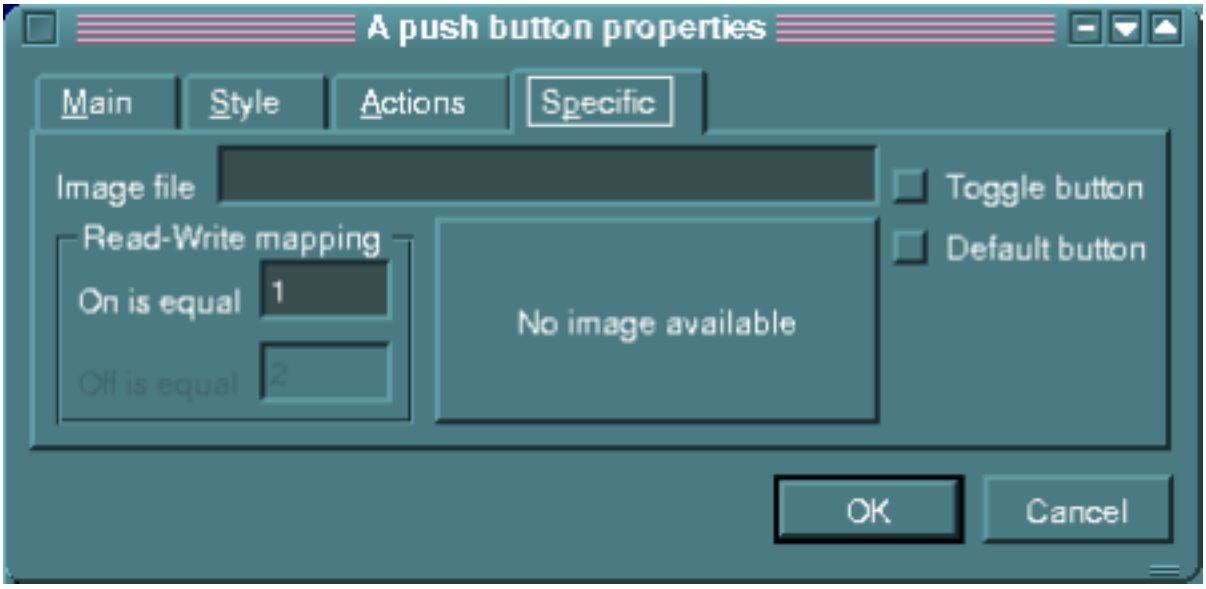
10.7.2. The Specific tab for Label control

- `Label` control displays one or more rows of a text or a picture.

`Specific` tab allows you to pass a path to a graphic file or use a preview button to choose the file using a dialog. If a graphic file can be displayed, it will replace a text of a control. If a text is displayed, you will set a kind of paragraph and whether a text be able to split into lines including whole words. You can do this using buttons from `Alignment` group. `Transparent control` option lets the control be transparent. In a case when a text is displayed, a background of the control won't be shown while in a case when a picture is presented, the control only will exist in points that the picture isn't transparent.

10.7.3. The Specific tab for Push button control

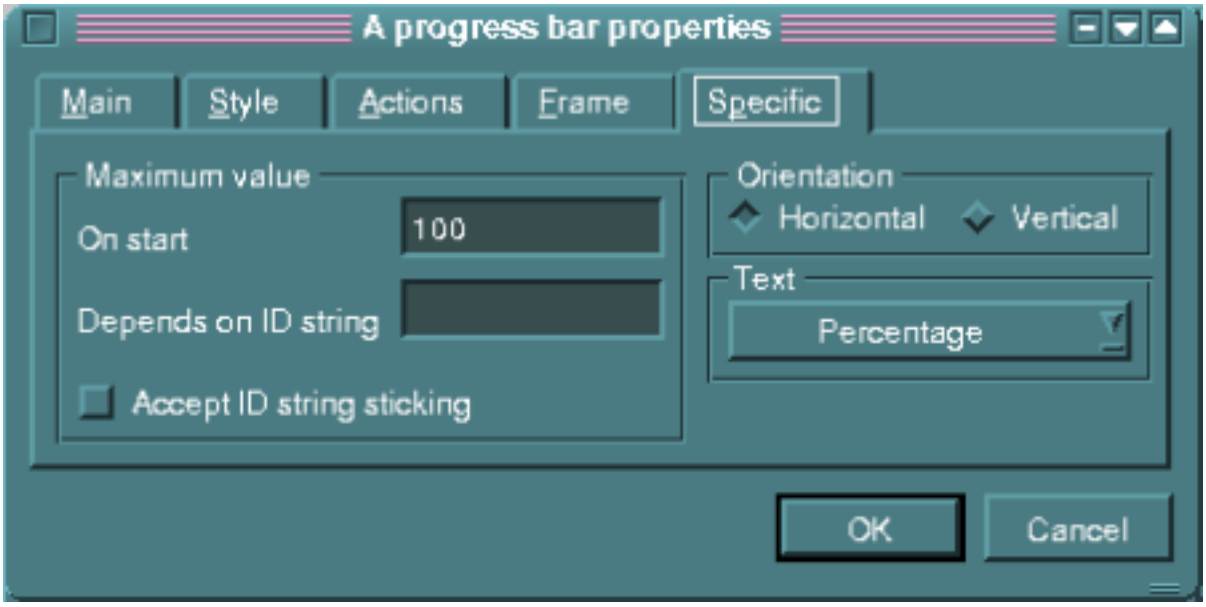
- `Push button` control confirms a user's choice.



The tab lets you pass a path to a graphic file like in a case of Label. It allows you to choose whether the control can be a default button for a dialog or it can be `Toggle button`. In this case you must pass equivalent of some integer number for each of states.

10.7.4. The Specific tab for Progress bar control

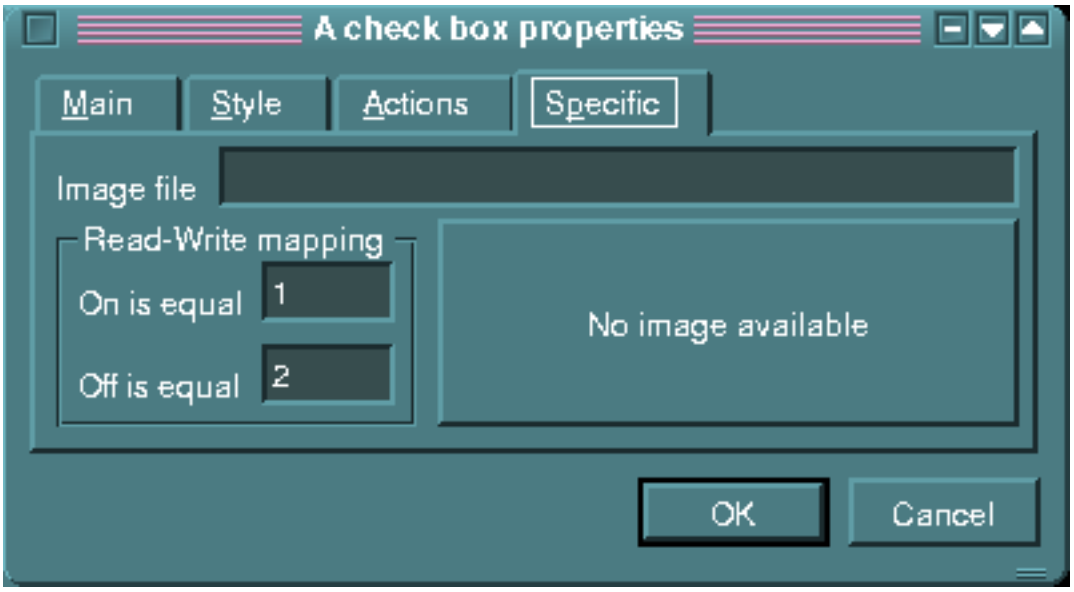
- `Progress bar` control mostly shows current value (greater or equal zero) of a specified parameter.



`Orientation` group of `Specific` tab gives you two ways of `Progress bar` presenting: `Vertical` or `Horizontal`. The control must have a maximum positive value (a minimum one is always equal zero). The value can be passed as fixed or it can depend on an object value of monitoring device when its identifier (OID) is passed. When you want to build this identifier through sticking some string, you can choose additionally `Accept ID string sticking` field. In `Text` group you can pass one of three possibilities of a current value displaying: without a text, an absolute value or a percent value.

10.7.5. The Specific tab for Check box control

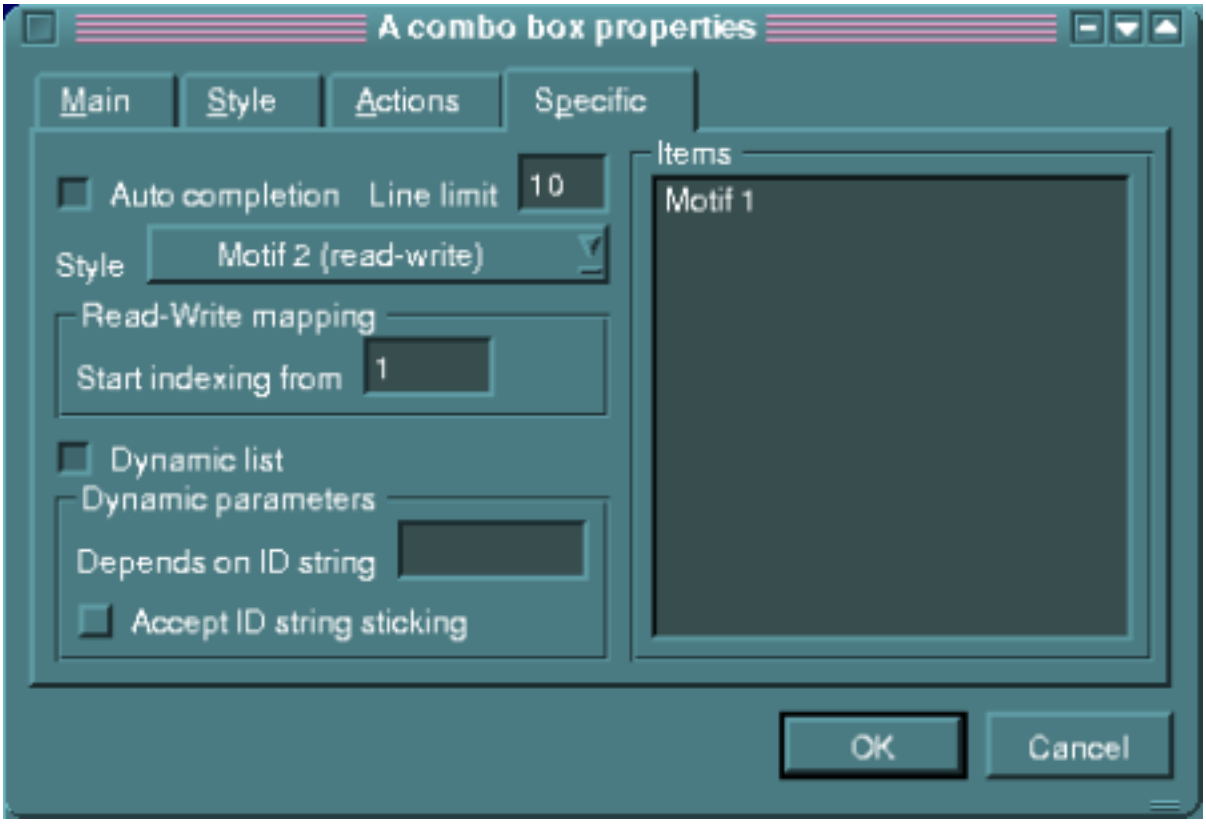
 - `Check box` control is a toggle button and allows you to select or unselect a specified option.



`Check box` is a control similar to Push button. So, `Specific` tab lets you pass a graphic file which is displayed instead of a text. Because the control has two states, you must pass some integer numbers for each of states.

10.7.6. The Specific tab for Combo box control

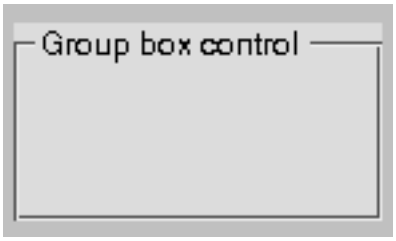
- `Combo box` control allows you to choose a specified item of available items list.



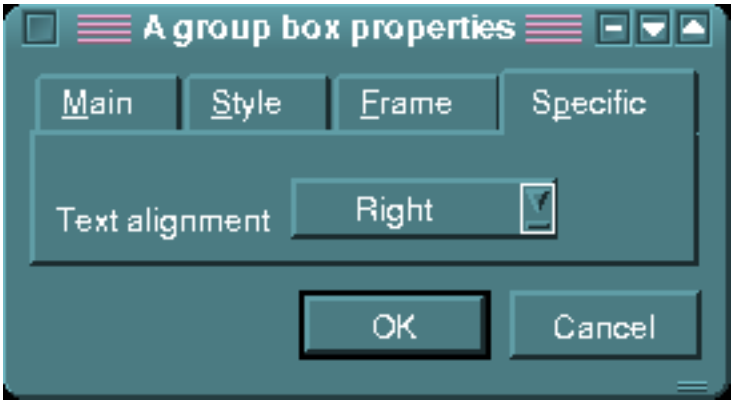
`Style` field of `Specific` tab allows you to choose one of three accessible styles. `Motif 2 (read-write)` style lets you edit `Combo box` control. In this case you can choose automatic adding of a completion from an available items list (`Auto completion` field). You must also pass how many lines a field of a control list will be high (`Line limit` field) and you must assign a number to the first line in `Read-Write mapping` group. This assignment allows to imitate values on a text from an available items list (this object sends by `Write` action and receives by `Read` action an integer value unless the list is created dynamically).

The edited field allows you to pass specified items of the list. If the list of items is created dynamically, you must choose `Dynamic list` field. Then `Combo box` ID string allows you to build items of the list. The object identified by the ID string means as the first item of MIB chart, that will be transformed into an item list of the control through the next SNMP requests of Get-Next type.

10.7.7. The Specific tab for Group box control



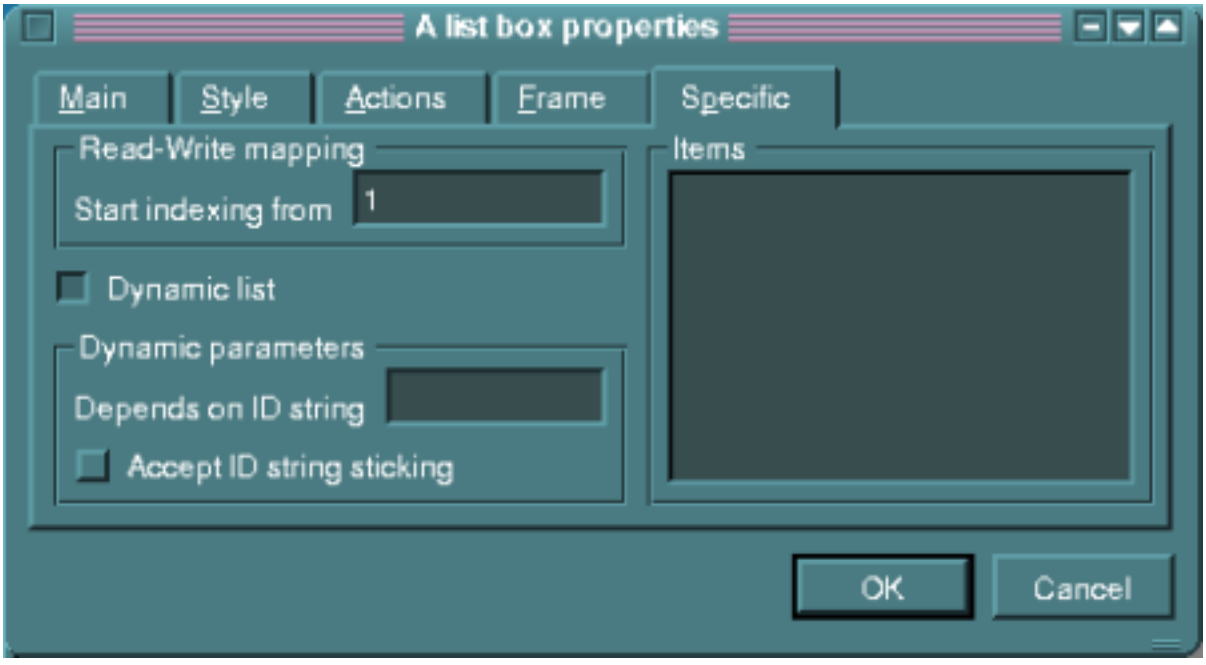
- `Group box` control groups objects.



`Specific` tab allows you to choose a kind of a title alignment as regards an object width. A text can be aligned to left or right side or central.

10.7.8. The Specific tab for List box control

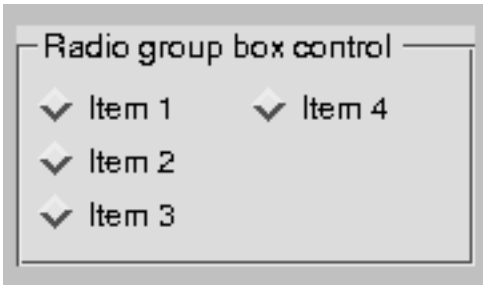
- `List box` control lets you set an item of a list.



In `Read-Write mapping` group of `Specific` tab you should pass an index value of the first item to which a control value will be mapped during its recording or reading (similarly as in Combo box control). A static item list you can pass in `Items` dialog. If you want to create a dynamic list, you should choose

`Dynamic list`. A creation procedure and working of the list was shown describing Combo box control.

10.7.9. The Specific tab for Radio button group control

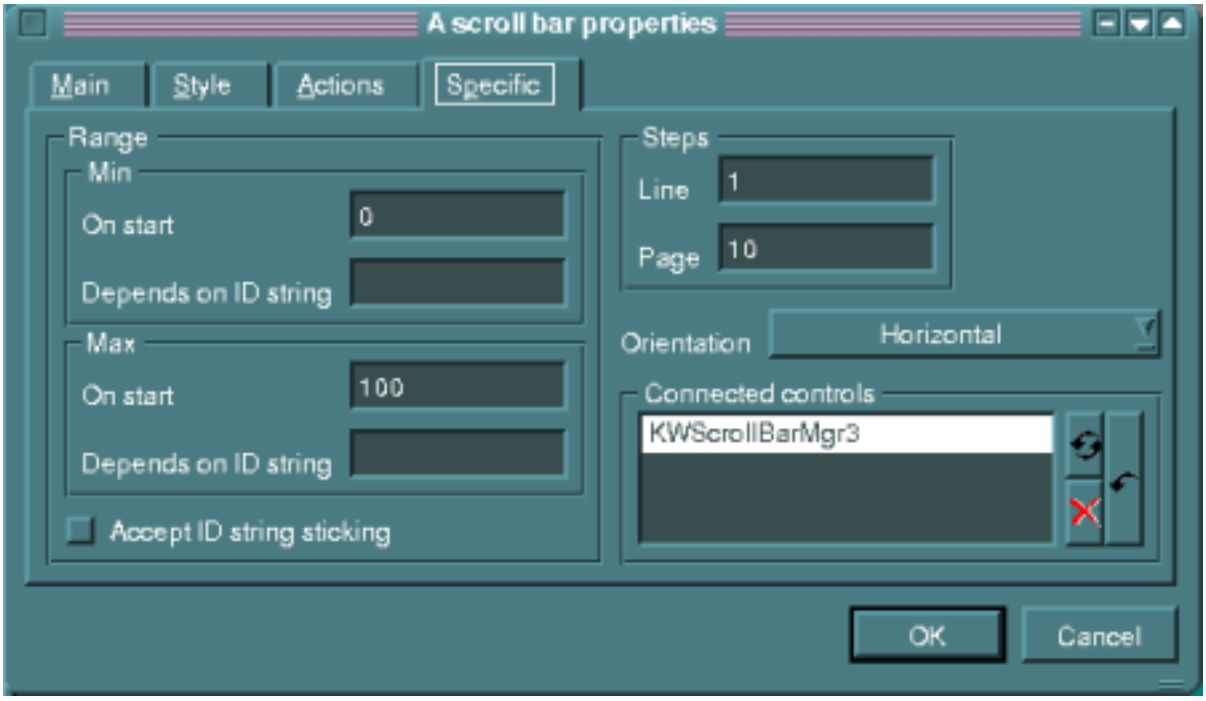


- `Radio button group` control lets you choose one of options available in a group.

`Text alignment` field of `Specific` tab lets you set a specified kind of a title alignment. You can also create a list with accessible options passing names of objects in `Radio buttons` group and you can set an order of their indexing and ascribe an initial index number to the first object (`Start indexing from` field). A defined `Radio buttons` you can configure by duble clicking it the left mouse button. Then you can pass a path to a graphic file that is displayed instead of a text. Each object like that you can move and change its size but only within `Radio button group` control.

10.7.10. The Specific tab for Scroll bar control

- `Scroll bar` control lets you manipulate other control value and present a current value between defined ranges.

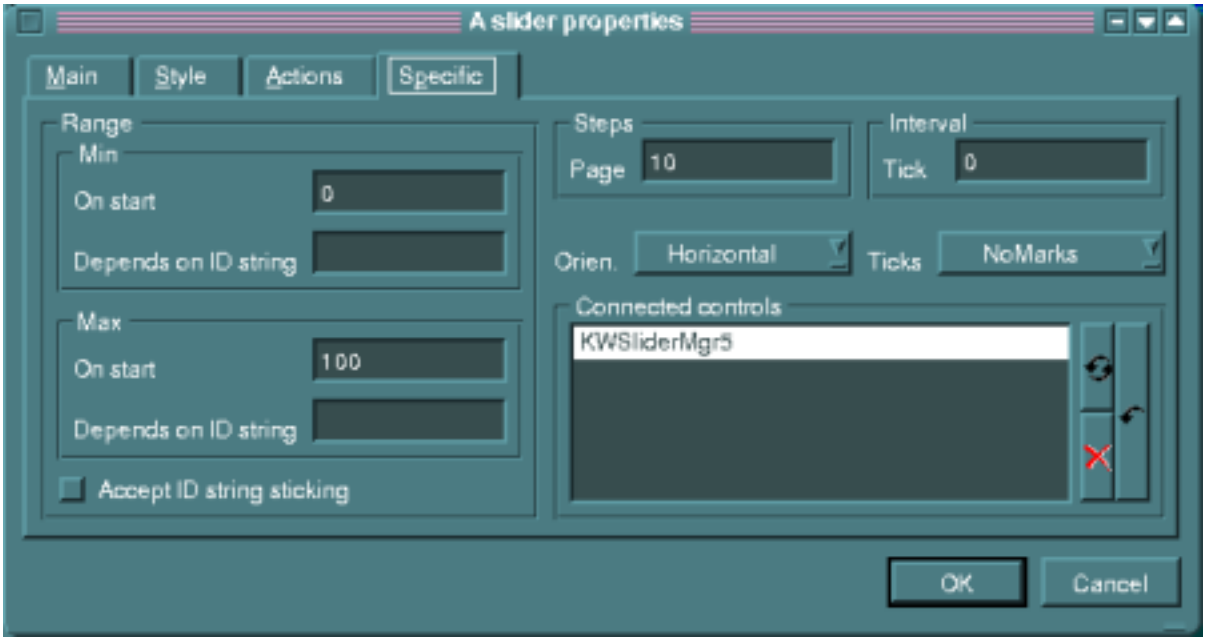


In `Range` group of `Specific` tab you can pass limited values (a minimum and maximum value). In `Steps` group you should set quantity of steps and in `Orientation` field - a kind of the control orientation. There are a list of objects connecting with the control and buttons to manipulate the list items in `Connected controls` group. These objects will react to moves of the control slide. Objects receiving new values of `Scroll bar` update their view depending on their specific properties.

As in case of Progress bar you can set that limited values will be able to change dynamically (in case of Progress bar you have one limited value while the second value is always zero). A description of `Sroll bar` and its list of items created dynamically is described in Progress bar description.

10.7.11. The Specific tab for Slider control

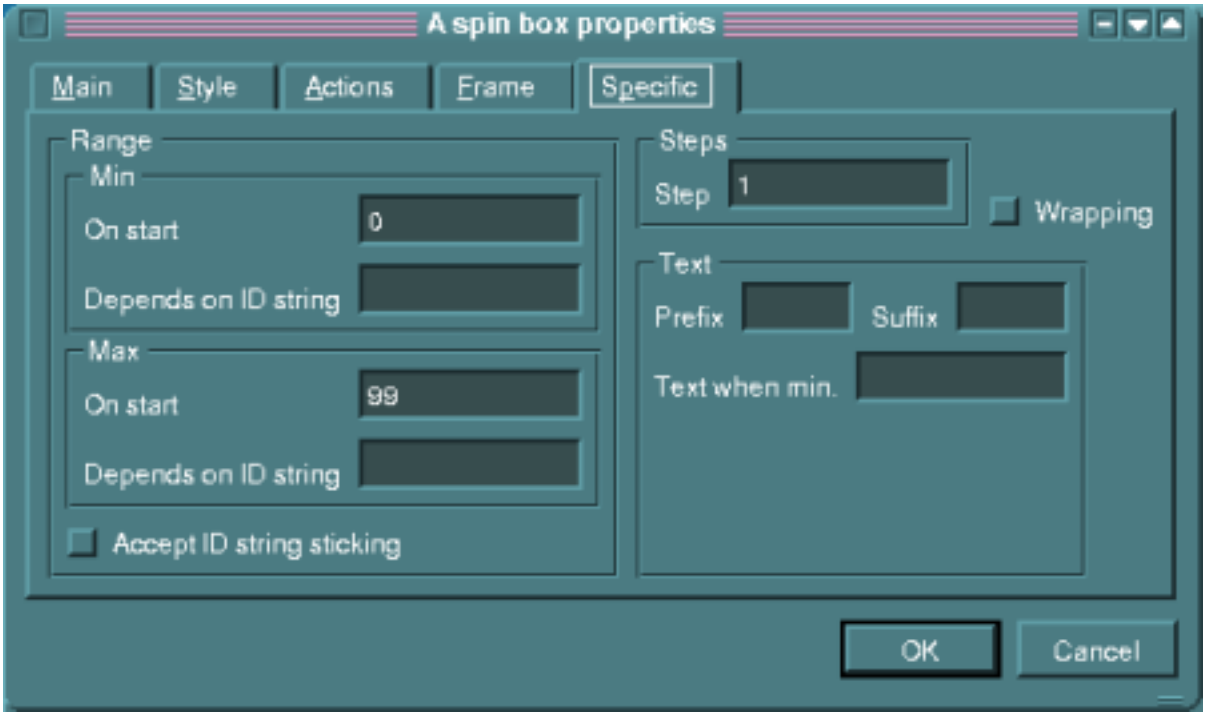
 - `Slider` control lets you manipulate values of other control and present a current value between defined ranges.



`Specific` tab includes configuration items similarly to Scroll bar. In `Ticks` field you can choose a way of drawing of a scale marks around the control while in `Interval` group you can set a distance between marks of a scale.

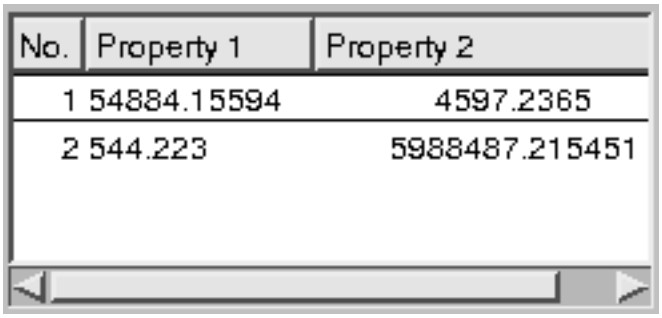
10.7.12. The Specific tab for Spin box control

- `Spin box` control allows you to manipulate a value taking specified ranges into consideration.

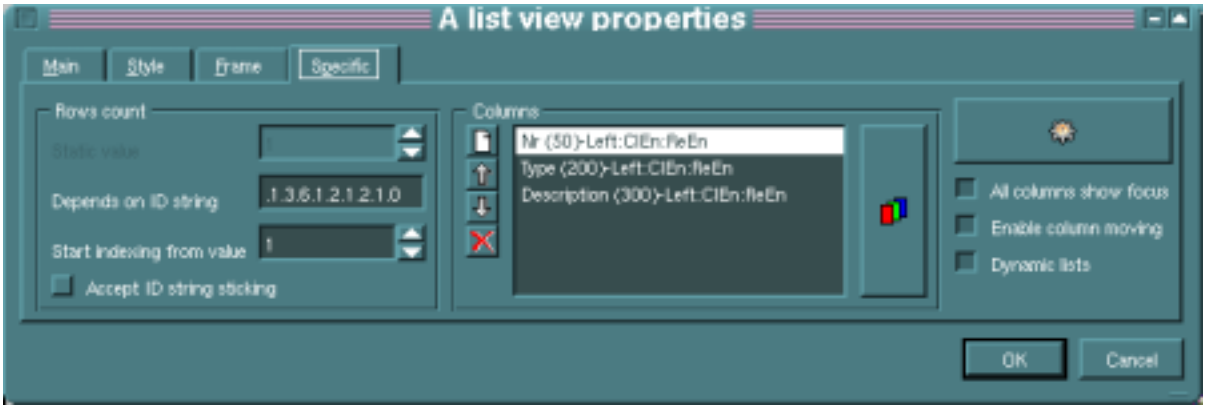


In `Range` group of `Specific` tab you should pass a range of values accepted by the control (as in Scroll bar and Slider). In `Steps` group you should pass a step quantity of the control. Choosing `Wrapping` option causes the values are wrapped i.e. after the greatest value follows the lowest one and vice versa. In `Text` group you can pass `Prefix` and `Suffix` appearing around displaying value of `Spin box`. A minimum value can be displayed as a text if you pass it in `Text when min.` field.

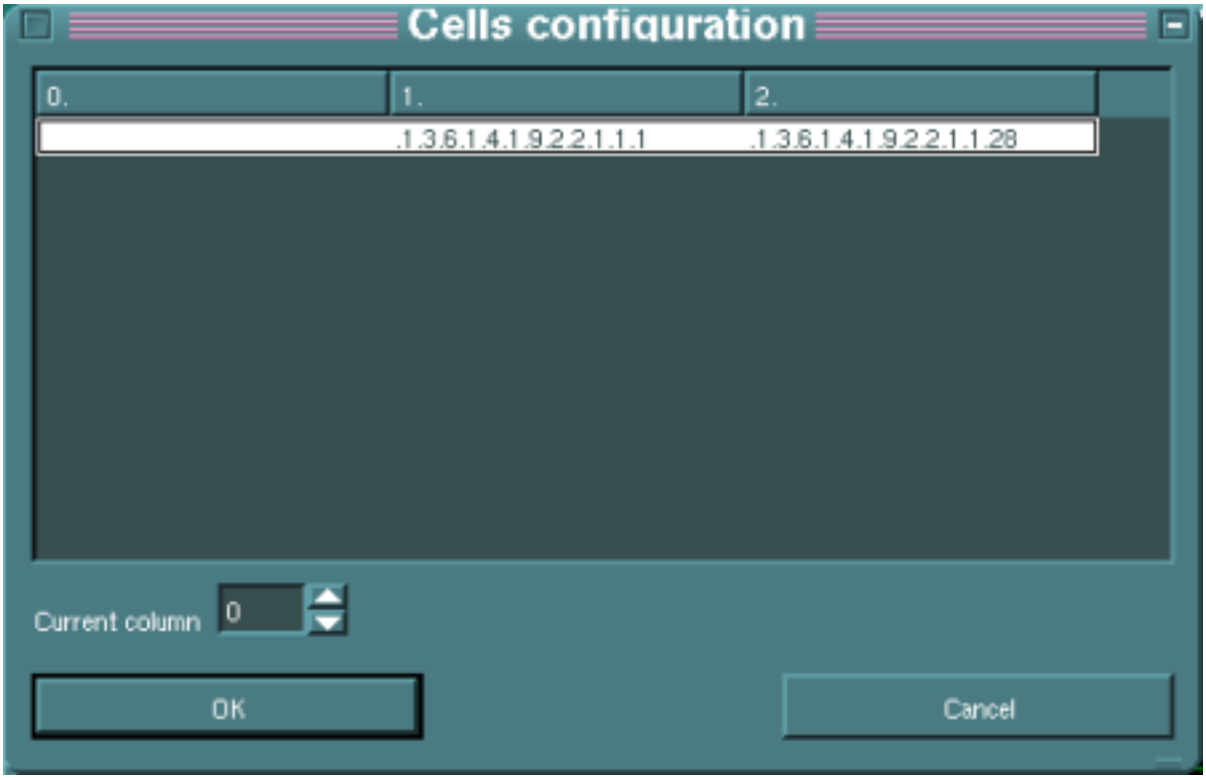
10.7.13. The Specific tab for List view control



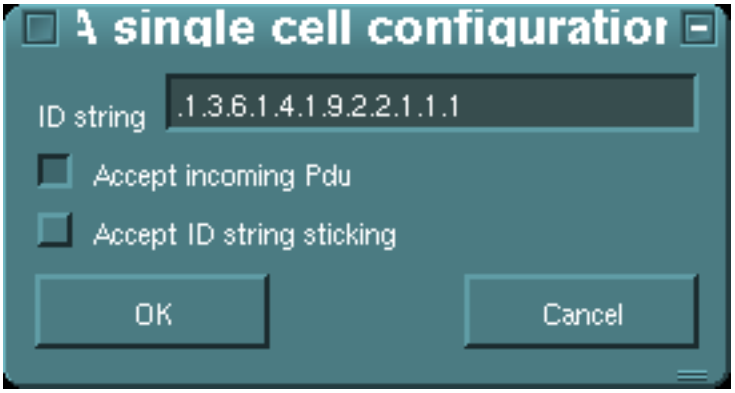
- `List view` control presents a list of items split into columns.



If you choose `Dynamic lists` field of `Specific` tab, you will be able to create a list of items dynamically for an object, otherwise you must pass in `Rows count` group, in `Static value` field a number of rows.

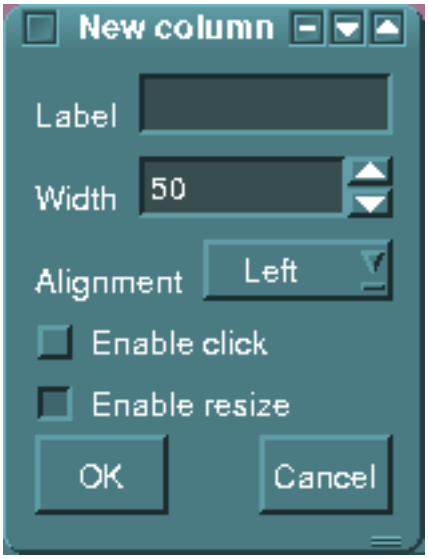


When you press the button  , `Cells configuration` dialog appears. It allows you to configure particular cells of a list showing different possibilities. They are depended on the list which is static or dynamic. For the static list, all cells (number of columns multiplied by number of rows) are available while for the dynamic one - only the first row. When you choose a column of `Current column` field and double click the left mouse button on the specified row of the list, `A single cell configuration` dialog appears for a specified cell.



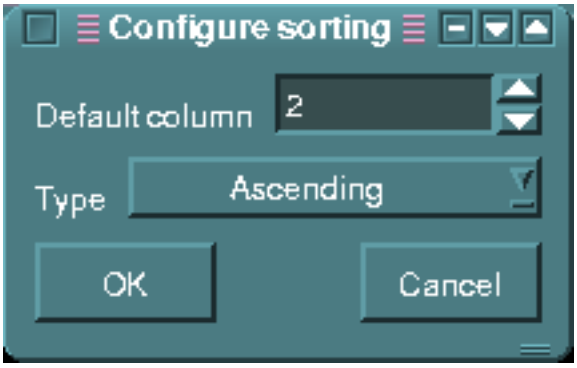
For each cell you can pass an object identifier of MIB that the cell presented a value of the object. In this case you should set `Accept incoming Pdu` field. If the identifier is built by sticking additional OID, you must choose `Accept ID string sticking` option.

In `Columns` group of `Specific` tab is presented a list of columns.



When you define a new column and press the button , `New column` dialog appears in which you should pass its title, its initial width and its kind of a text alignment and you must decide, that a column will react to clicking the mouse or you will be able to change its width.

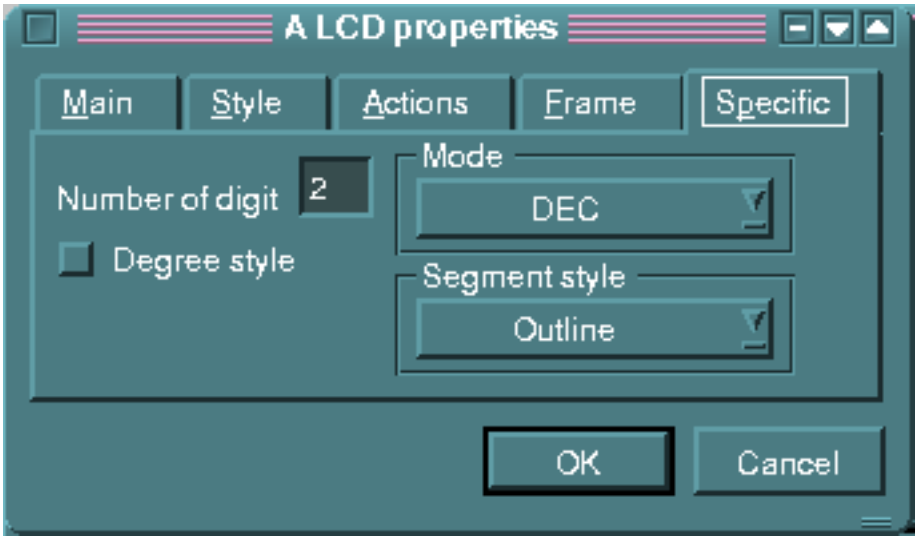
`All columns show focus` field of `Specific` tab allows you to choose that all columns will have a focus or only the first one. `Enable column moving` field shows you if the columns can be replaced.



The button  runs `Configure sorting` window that allows you to configure a sorting for a whole object: a column number and a kind of an order (increasing or decreasing).

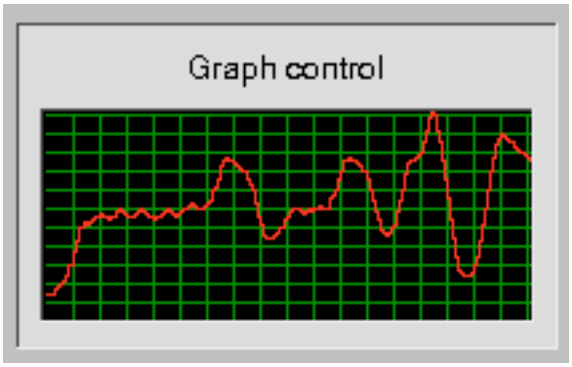
10.7.14. The Specific tab for LCD number control

 - `LCD number` presents a numerical value.

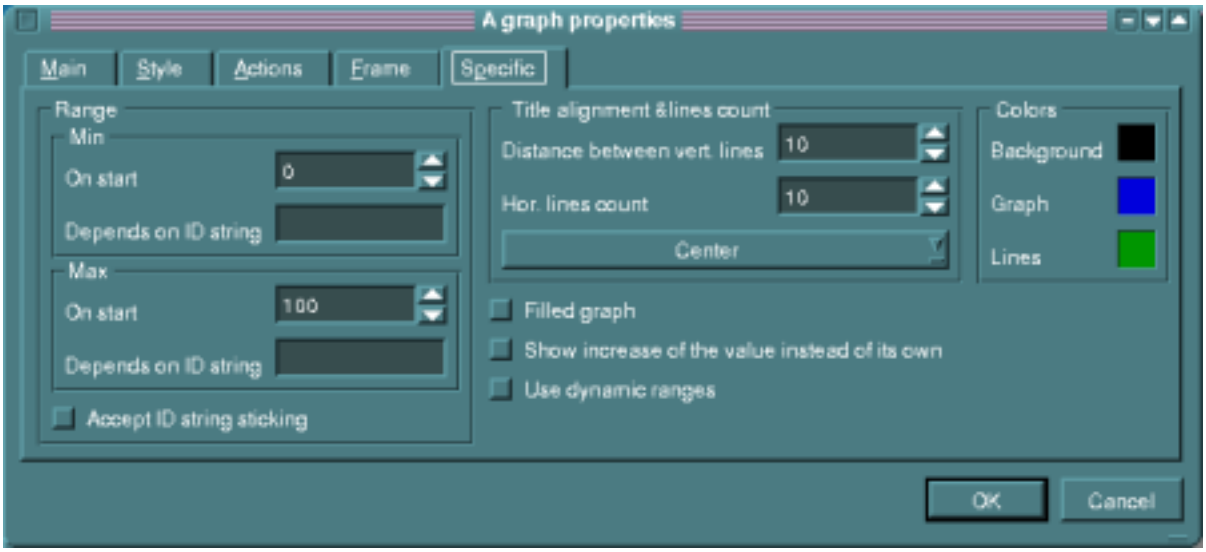


`Number of digit` field of `Specific` tab allows you to set a number of digits that will be displayed. `Degree style` option shows you if a degree character is displayed or not. In `Mode` group you can choose one of four ways of number displaying: binary, octal, decimal, hexadecimal. If `Degree style` option is set, decimal number always will be displayed. `Segment style` group lets you choose one of three ways of digit drawing.

10.7.15. The Specific tab for Graph control



- `Graph` control presents next values of data in a form of a time function graph.



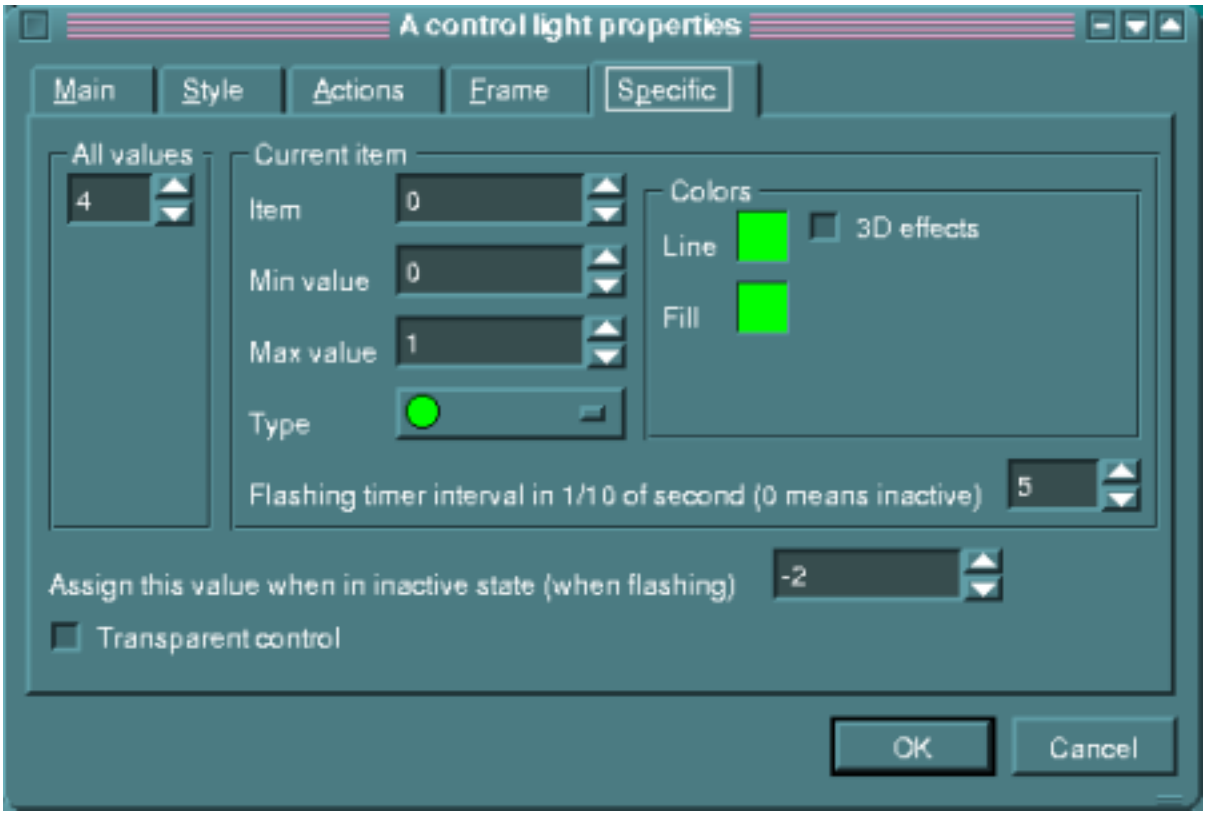
In `Range` group of `Specific` tab you should pass limited values (a minimum and maximum value). In `Title alignment and lines count` you can choose a kind of a text alignment, a number of horizontal lines and distance between them. `Filled graph` option lets you choose that a graph will be filled with color or drawn as line. `Show increase of the value instead of its own` option helps you to show an increase of the value instead of the value. In `Colors` group you can choose colors that will be used to draw a graph, a grid and a background.

As in a case of Progress bar you can set that limited values were depended on values of MIB objects. The `Graph` work is shown in a description of Progress bar control.

You can also use other way to manipulate the control ranges choosing `Use dynamic ranges` option. It shows that limited values can be changed when are exceeded during the control work.

10.7.16. The Specific tab for Control light control

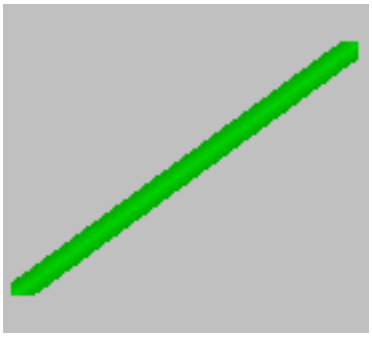
- `Control light` control shows its current state in an intuitive way.



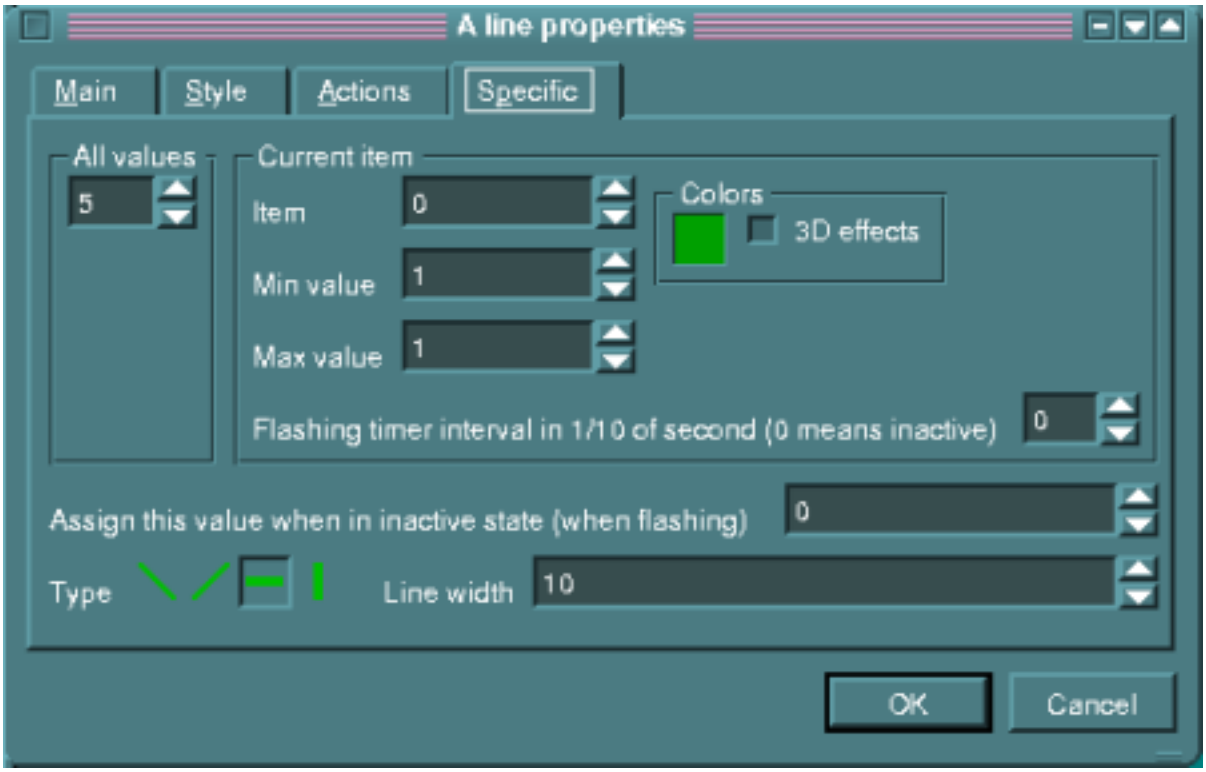
In `All values` group of `Specific` tab you can pass a number of all possible values. For each value in `Current item` group, fields: `Min value` and `Max value` assign a closed range of accepted values by the item. `Type` field allows you to choose a kind of a shape or a graphic file that will represent this value. Depending on specified value of `Type` field you can pass a path to a graphic file or a line color and a color filling a specified shape and whether the shape can be drawn flatly or spatial. `Flashing timer interval in 1/10 of second 0 means inactive)` field lets you turn on timer with a given interval in 1/10 of seconds. A zero value causes the timer isn't run. During the work of a timer `Control light` alternately accepts once a given value (a value, that is currently configured) and next time a value given in `Assign this value when in inactive state (when flashing)` field.

Chooseing `Transparent control` option causes that the control will be without a background (it will be transparent).

10.7.17. The Specific tab for Line control



- `Line` control shows its current state in an intuitive way.



A configuration of `Line` control is similar to Control light configuration. A main difference is a fact that a value of `Type` field is common for all the control values. In `Line width` filed you should pass a control width in pixels. `Line` control as distinct from Control light always is transparent.

10.8. Related articles

Network Nodes Viewer (xdnnv)