NetXMS User Guide

Release 4.4.3

Raden Solutions, SIA

Mar 08, 2024

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 About this document

The User Manual describes the main aspects of NetXMS monitoring system. This manual enables all users to get an overview of the various functionalities of NetXMS. The main aspects outlined in here describe the possibilities and functionaries of the NetXMS interface and elucidate working processes.

1.2 Target audience

This manual is intended for NetXMS operators, and provides all information necessary to successfully operate NetXMS.

1.3 Conventions

The following typographical conventions are used in this manual.

Sample	Description
Button	Any GUI element: Button, Menu item
Another Guide	Reference to external manual or man page
`Control-M`	Keyboard shortcut
DCI	Term which could be found in glossary
File → *Exit*	Menu selection path, you must click on *File*, then *Exit*

CHAPTER

TWO

BASIC CONCEPTS

2.1 Objects

All monitored network infrastructure is represented as a set of *objects* in NetXMS monitoring system. Each object represents one physical or logical entity (e.g. host or network interface), or group of them (e.g. subnet, container). Objects are organized into hierarchical structure. Each object has it's own access rights. Access rights are applied hierarchically on all children of object. For example if *Read* access right is granted to a user on a *Container*, then user has *Read* right on all objects that this *Container* contains. Every object has set of attributes; some of them exist for all objects (like *id* and *name* or *status*), while other depend on object class – for example, only *Node* objects have attribute *SNMP community string*. There are default attributes and custom attributes defined either by user or external application via NetXMS API.

NetXMS has six top level objects – `Entire Network`, `Service Root` (named "Infrastructure Services" after system installation), `Template Root`, `Network Map Root`, `Dashboard Root` and `Business Service Root`. These objects serve as an abstract root for an appropriate object tree. All top level objects have only one editable attribute – name.

Object Class	Description	Valid Child Objects
Entire Network	Abstract object representing root of IP topology tree. All zone and subnet objects located under it. System can have only one object of this class.	• Zone (if zoning enabled) • Subnet (if zoning disabled)
Zone	Object representing group of (usually interconnected) IP networks without overlapping addresses. Contains appropriate subnet objects.	• Subnet
Subnet	Object representing IP subnet. Typically objects of this class are created automatically by the system to reflect system's knowledge of IP topology. The system places Node objects inside an appropriate Subnet object based on an interface configuration. Subnet objects have only one editable attribute - *Name*.	• Node
Node	Object representing physical host or network device (such as a router or network switch). These objects can be created either manually by administrator or automatically during network discovery process. They have a lot of attributes controlling all aspects of interaction between NetXMS server and managed node. For example, the attributes specify what data must be collected, how node status must be checked, which protocol versions to use, etc. Node objects contain one or more interface objects. The system creates interface objects automatically during configuration polls.	• Interface • Access point • Network Service • VPN Connector

continues on next page

Table 1 – continued from previous page

Object Class	Description	Valid Child Objects
Interface	Interface objects represent network interfaces of managed computers and devices. These objects created automatically by the system during configuration polls or can be created manually by user.	
Access point	Object representing wireless network access point. A node can have several access points, e.g. 2.4Ghz and 5Ghz, or in case of thin wireless access points managed by a central controller. These objects are created automatically by the system.	
Network Service	Object representing network service running on a node (like http or ssh), which is accessible online (via TCP IP). Network Service objects are always created manually. Currently, the system works with the following protocols - HTTP, POP3, SMTP, Telnet, SSH and Custom protocol type.	
VPN Connector	Object representing VPN tunnel endpoint. Such objects can be created to add VPN tunnels to network topology known to NetXMS server. VPN Connector objects are created manually. In case if there is a VPN connection linking two different networks open between two firewalls that are added to the system as objects, a user can create a VPN Connector object on each of the firewall objects and link one to another. The network topology will now show that those two networks are connected and the system will take this condition into account during problem analysis and event correlation.	
Service Root	Abstract object representing root of your infrastructure service tree. System can have only one object of this class. After system installation it is named "Infrastructure Services".	• Cluster • Chassis • Condition • Container • Node • Sensor • Subnet • Rack
Container	Grouping object which can contain any type of objects that Service Root can contain. With help of container objects you can build object's tree which represents logical hierarchy of IT services in your organization.	• Cluster • Chassis • Condition • Container • Node • Sensor • Subnet • Rack
Cluster	Object representing cluster consisting of two or more nodes.	• Node
Rack	Object representing a rack. It has the same purpose as container, but allows to configure visual representation of equipment installed in a rack.	• Node • Chassis

continues on next page

Table 1 – continued from previous page

Object Class	Description	Valid Child Objects
Chassis	Object representing a chassis, e.g. a blade server enclosure. Chassis can be configured as a part of a rack.	• Node
Condition	Object representing complicated condition – like "cpu on node1 is overloaded and node2 is down for more than 10 minutes". Conditions may represent more complicated status checks because each condition can have a script attached. Interval for evaluation of condition status is configured in Server Configuration Variables as ConditionPollingInterval with default value 60 seconds.	
Template Root	Abstract object representing root of your template tree.	• Template • Template Group
Template Group	Grouping object which can contain templates or other template groups.	• Template • Template Group
Template	Data collection template.	• Mobile Device • Node
Network Map Root	Abstract object representing root of your network map tree.	• Network Map • Network Map Group
Network Map Group	Grouping object which can contain network maps or other network map groups.	• Network Map • Network Map Group
Network Map	Network map.	
Dashboard Root	Abstract object representing root of your dashboard tree.	• Dashboard
Dashboard	Dashboard. Can contain other dashboards.	• Dashboard
Business Service Root	Abstract object representing root of your business service tree. System can have only one object of this class.	• Business Service • Business Service Prototype
Business Service	Object representing single business service. Can contain other business services or business service prototypes.	• Business Service • Business Service Prototype
Business Service Prototype	Prototype from which business service objects are automatically populated.	

2.1.1 Object status

Each object has a status. Status of an object calculated based on:

- Polling results
- Status of child objects (e.g. interfaces of node, nodes under container)
- Active alarms, associated with the object (after an alarm is resolved or terminated, it no longer affects object status)
- Value of status *DCIs* (DCI that has `Use this DCI for node status calculation` property enabled)

For some object classes, like Report or Template, status is irrelevant. Status for such objects is always *Normal*. Object's status can be one of the following:

Nr.	Status	Description
0	Normal	Object is in normal state.
1	Warning	Warning(s) exist for the object.
2	Minor	Minor problem(s) exist for the object.
3	Major	Major problem(s) exist for the object.
4	Critical	Critical problem(s) exist for the object.
5	Unknown	Object's status is unknown to the management server.
6	Unmanaged	Object is set to "unmanaged" state.
7	Disabled	Object is administratively disabled (only applicable to interface objects).
8	Testing	Object is in testing state (only applicable to interface objects).

2.1.2 Unmanaged status

Objects can be unmanaged. In this status object is not polled, DCIs are not collected, no data is updated about object. This status can be used to store data about an object that is temporary or permanently unavailable or not managed.

2.1.3 Maintenance mode

This is special status, that's why it is not included in above status list. This status prevents event processing for specific node. While this node in maintenance mode is still polled and DCI data is still collected, but no event is generated.

2.2 Data Collection Items

Every node can have many parameters, such as CPU utilization, amount of free memory or disk space usage. The management server can collect these parameters, check them for threshold violations and store them in the database. In NetXMS, parameters configured for collection are called Data Collection Items or DCI for short. One DCI represents one node's parameter, and unlimited number of DCIs can be configured for any node.

2.2.1 Thresholds

Each threshold is a combination of a condition and event pair. If a condition becomes true, associated "activation" event is generated, and when it becomes false again, "deactivation" event generated. Thresholds let you take a proactive approach to network management. Thresholds can be defined for any data collection items that is monitored, more than one threshold for a single DCI can be defined.

2.3 Events and Alarms

Many services within NetXMS gather information and generate events that are forwarded to NetXMS Event Queue. Events can also be emitted from agents on managed nodes, or from management applications residing on the management station or on specific network nodes. All events are processed by NetXMS Event Processor one-by-one, according to the processing rules defined in Event Processing Policy. As a result of event processing, some actions can be taken, and event can be shown up as alarm, sent as e-mail and notifications (SMS, instant messages). NetXMS provides one centralized location - the Alarm Browser, where the alarms are visible to your team. You can control which events should be considered important enough to show up as alarms. You and your team can easily monitor the posted alarms and take appropriate actions to preserve the health of your network.

Examples of alarms include:

- A router exceeded its threshold of traffic volume that you configured in Data Collection.
- The shell script that you wrote gathered the specific information you needed and posted it to the NetXMS as an event.
- One of your mission-critical servers switched to UPS battery power.
- An SNMP agent on a managed critical server forwarded a trap to NetXMS because it was overheating and about to fail.

2.4 Zones

As NetXMS server keeps track of an IP topology, it is important to maintain the configuration in which IP addresses do not overlap and that two IP addresses from same subnet are really within one subnet. Sometimes, however, it is needed to monitor multiple sites with overlapping IP address ranges. To correctly handle such situation, zoning must be used. Zone in NetXMS is a group of IP subnets which form non-overlapping IP address space. There is always zone 0 which contains subnets directly reachable by management server. For all other zones server assumes that subnets within that zones are not reachable directly, and proxy must be used.

CHAPTER

THREE

USER INTERFACE

Note: One of the goals of NetXMS Management Console is to provide identical user experience across all supported platforms, including Web Interface. Screenshots in this particular guide are based on Mac OS X version.

3.1 Login

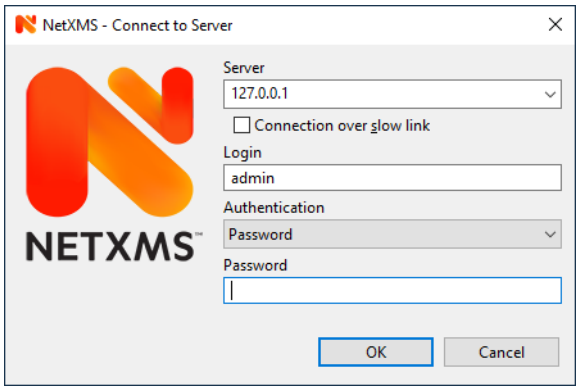


Fig. 1: Login Dialog

When Management Console is started, user is presented with login dialog. User should enter server host name or IP address, login and password.

3.2 Workbench

When user is authenticated, a single Workbench window is displayed. A Workbench window offers one or more perspectives. A perspective contains views, such as the *Object Browser*. Multiple Workbench windows can be opened simultaneously. Initially, in the first Workbench window that is opened, the *Management* perspective is displayed, with *Object Browser* and *Object Details* views visible. A shortcut bar appears in the top right corner of the window. This allows you to open new perspectives and switch between ones already open. The name of the active perspective is shown in the title of the window and its item in the shortcut bar is highlighted.

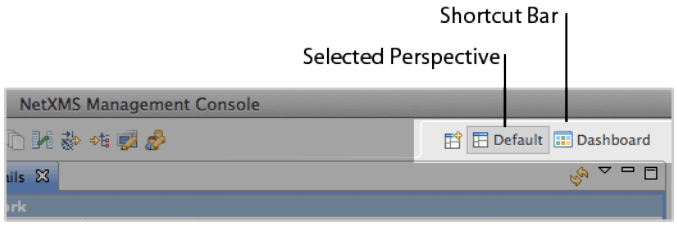


Fig. 2: Shortcut bar, *Management* perspective is selected

3.3 Views

The primary use of Views is to provide convenient navigation through the information displayed in Workbench. A view might appear by itself or stacked with other views in a tabbed notebook. To activate a view that is part of a tabbed notebook simply click its tab. Views have two menus. The first menu, which is accessed by right-clicking on the view's tab, allows the view to be manipulated in much the same manner as the menu associated with the Workbench window. The second menu, called the "view pull-down menu", is accessed by clicking the down arrow . The view pull-down menu typically contains operations that apply to the entire contents of the view, but not to a specific item shown in the view.

A view can be displayed by selecting it from the appropriate *View*, *Monitor*, or *Configuration* menu, or via *Window* → *Show View* menu. A perspective determines which views may be required and displays these on the *Show View* sub-menu. Additional views are available by choosing command link *Other* at the bottom of the *Show View* sub-menu. This is just one of the many features that provide for the creation of a custom work environment.

Through the normal course of using the Workbench you will open, move, resize, and close views. If you'd like to restore the perspective back to its original state, you can select the *Window* → *Reset Perspective* menu operation.

3.3.1 Rearranging views

You can change the position of any view in the Workbench by following the steps below:

1. Click in the title bar of the view and drag the view across the Workbench window. Do not release the mouse button yet.
2. While still dragging the view around on top of the Workbench window, note that various drop cursors appear. These *Drop cursors* indicate where the view will dock in relation to the view underneath the cursor when the mouse button is released. Notice also that a rectangular highlight is drawn that provides additional feedback on where the view will dock.
3. Dock the view in any position in the Workbench window, and view the results of this action.

3.3.2 Drop cursors

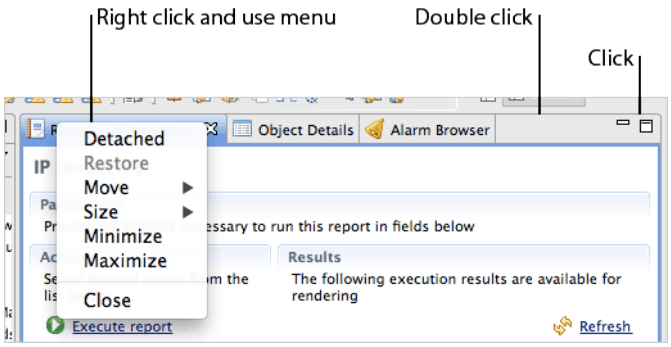
Drop cursors indicate where it is possible to dock a part in the Workbench window. Several different drop cursors may be displayed when rearranging a part.

	Dock above: If the mouse button is released when this cursor is displayed, the part will appear above the part underneath the cursor.
	Dock below: If the mouse button is released when this cursor is displayed, the part will appear below the part underneath the cursor.
	Dock to the left: If the mouse button is released when this cursor is displayed, the part will appear to the left of the part underneath the cursor.
	Dock to the right: If the mouse button is released when this cursor is displayed, the part will appear to the right of the part underneath the cursor.
	Stack: If the mouse button is released when this cursor is displayed, the part will appear as a tab in the same pane as the part underneath the cursor.
	Restricted: If the mouse button is released when this cursor is displayed, the part will not dock there.

3.3.3 Maximizing and minimizing views

The console presentation provides a rich environment consisting of one or more View Stacks (each containing one or more views). These various parts compete for valuable screen real-estate and correctly managing the amount of screen given to each can greatly enhance your productivity within the console. The two most common mechanisms for managing this issue are "minimize" (i.e. make me use as little space as possible) and "maximize" (i.e. give me as much space as you can). The console presentation provides a variety of ways to access these operations:

- Using the minimize and maximize buttons provided on a stack's border
- Using the minimize and maximize buttons provided on a stack's border
- Selecting the *Minimize* or *Maximize* item on the context (right-click) menu for a stack
- Double-clicking on a stack
- Using `Control + M`: this is a key binding for a command that will toggle the currently active part between its "maximized" and its "restored" (i.e. normal) states.



Maximize

It is desirable at times to focus your attention on one particular view to the exclusion of the others. Console implements the maximize behavior by minimizing all stacks except the one being maximized. This allows the maximized stack to completely occupy the main presentation while still allowing to access any open views in your perspective by using the icons in their *Trim Stack* (the area around the edges of the window is called the "trim").

Minimize

Another way to optimize the use of the screen area is to directly minimize stacks that are of no current interest. Minimizing a stack will cause it to be moved into the trim area at the edges of the workbench window, creating a *Trim Stack*. *View Stack* will get minimized into a trim representation that contains the icons for each view in the stack:

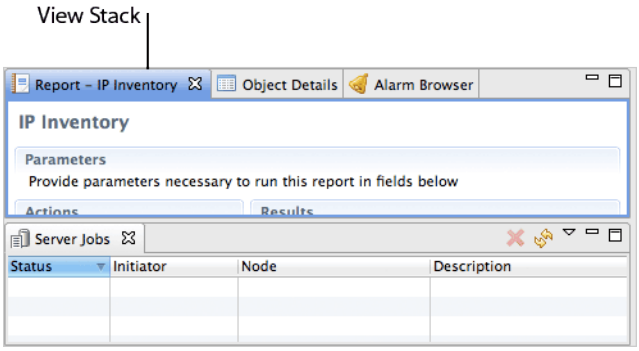


Fig. 3: Stacked Views

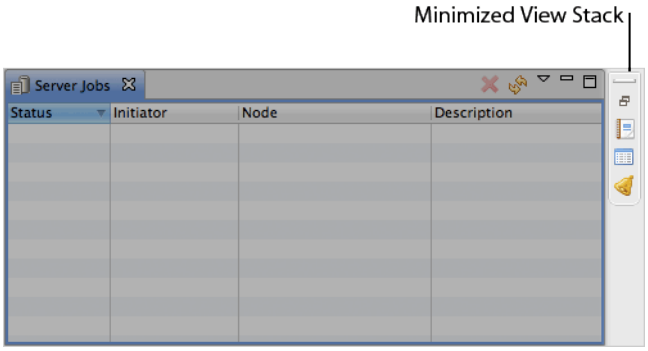


Fig. 4: Views minimized into Trim Stack

3.4 Perspectives

A perspective defines the initial set and layout of views in the Workbench window. One or more perspectives can exist in a single Workbench window. Perspectives can be opened in one of two ways:

1. In the same (existing) Workbench window.
2. In a new Workbench window.

Perspectives define visible action sets, which can be changed to customize a perspective. A perspective that is built in this manner can be saved, creating a custom perspective that can be opened again later.

3.4.1 New perspectives

There are several ways to open a new perspective within this Workbench window:

- Using the *Open Perspective* button  on the shortcut bar.
- Choosing a perspective from the *Window* → *Open Perspective* menu.

To open one by using the shortcut bar button:

1. Click on the *Open Perspective* button  .
2. A menu appears showing the same choices as shown on the *Window* → *Open Perspective* menu. Select perspective from the list or choose *Other* (in that case additional *Select Perspective* dialog will be opened).

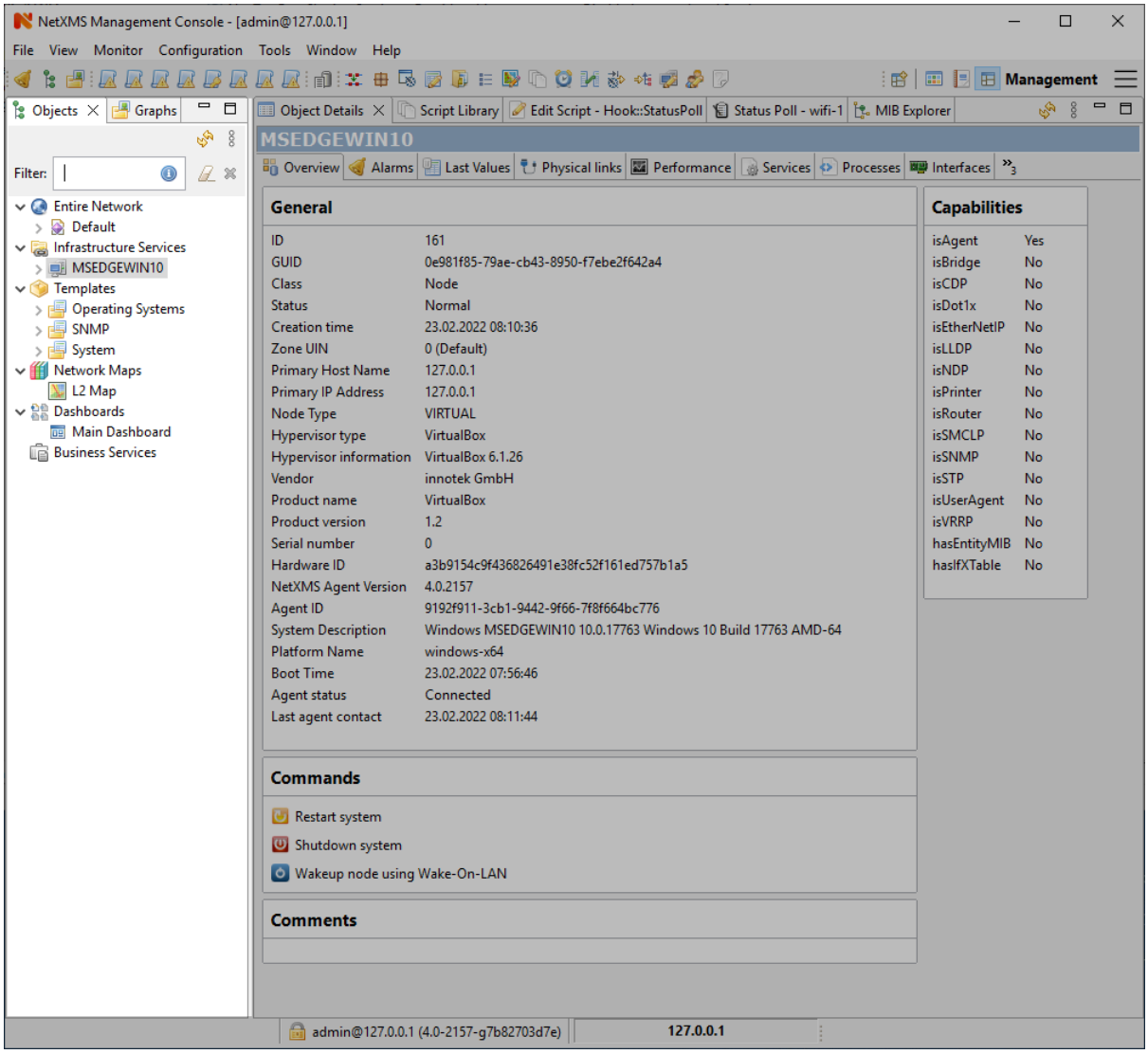
Icons of recently used perspectives will be placed on shortcut bar for quick selection.

3.4.2 Saving perspectives

The Workbench allows any current view layout to be saved for future use. To save current layout as new perspective:

1. Choose *Window* → *Save Perspective As* from main menu.
2. The *Save Perspective As* dialog allows for an existing perspective to be redefined or for a new perspective to be created. Select existing perspective to redefine or type name of new perspective, and click *OK*.
3. Answer *Yes* to the subsequent confirmation dialog. The new perspective layout will be used if the perspective is reset or if a new one is opened.

3.5 Object Browser



Object browser represents all objects in the system as a tree with multiple root objects. Tree is built based on object hierarchy and user permissions. Only objects available to currently logged in user will be shown. User has two options to interact with objects:

- Click Left mouse button to select object and display its details (see *Object Details*)
- Click Right mouse button to open context menu with actions available for this particular object type

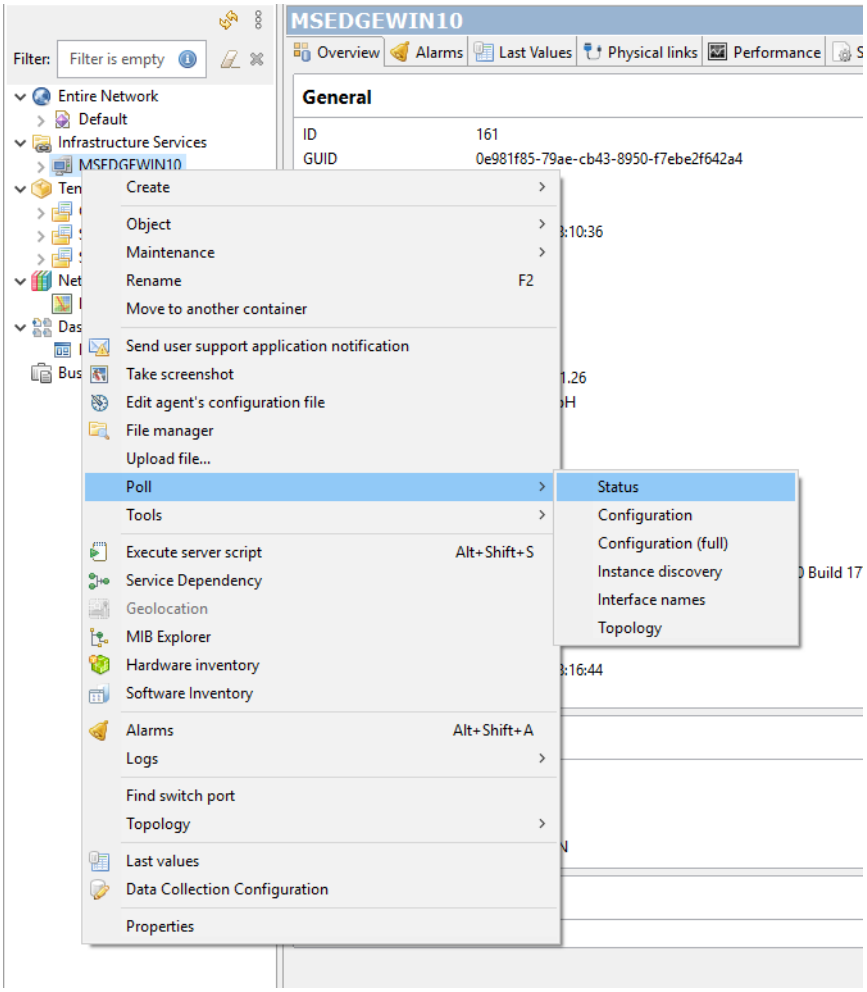


Fig. 5: Popup menu for object type *Node*

3.5.1 Object status

System track status of each object, which can range from *Minor* to *Critical*. Status is displayed as overlay on icon of each object.

3.5.2 Filtering

Above object tree there is filter field that allows to filter objects in the object tree. Filter supports a number of prefix characters that define how search is performed:

Prefix	Status
>	Search by IP address part
^	Search by exact IP address
#	Search by object ID
/	Search by comment
@	Search by zone ID

Without prefix search is performed by object name.

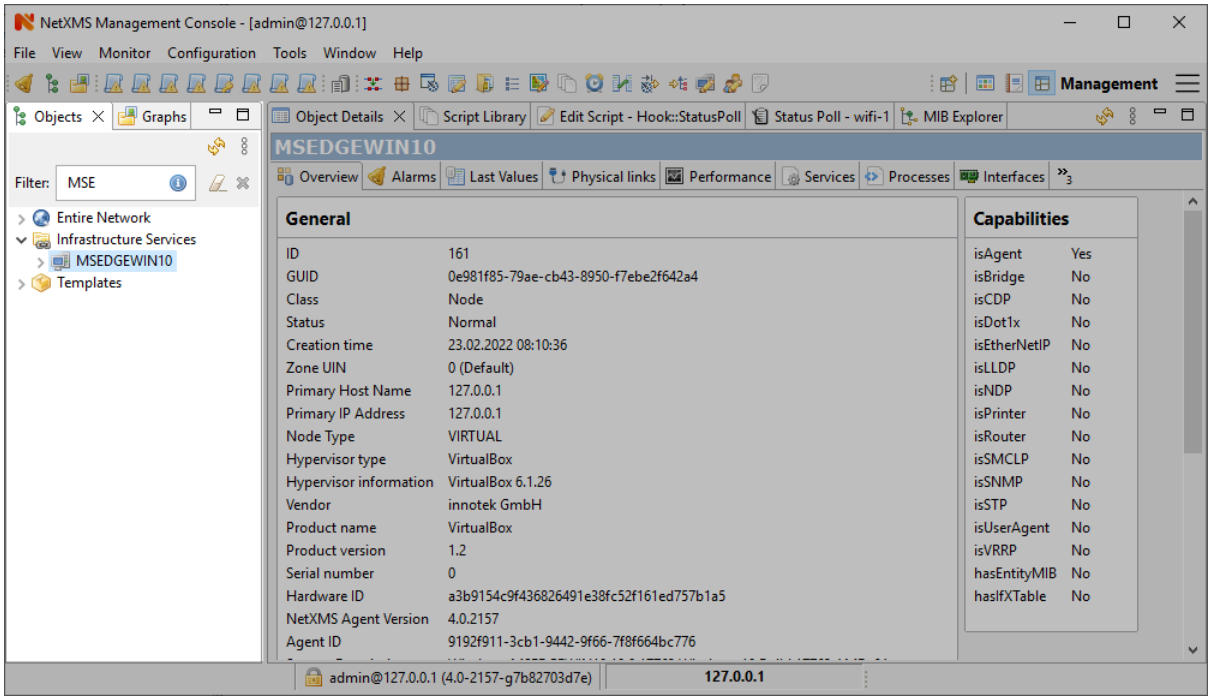


Fig. 6: As-you-type filter in action

3.6 Object Details

This view provides one or more tabs with detailed information about object currently selected in *Object Browser*. List of available tabs depends on type of the selected object.

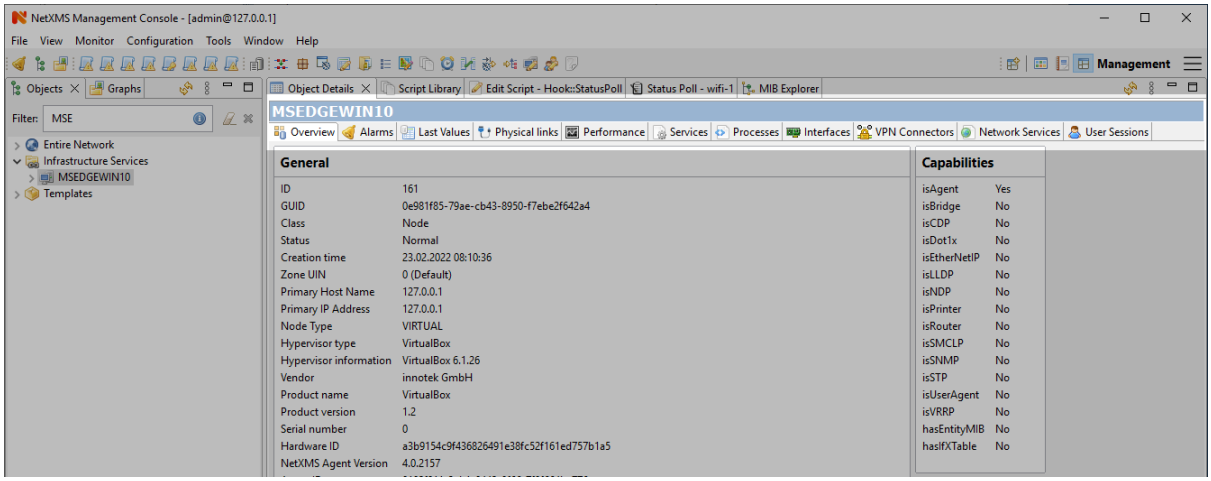


Fig. 7: Additional row of tabs (node with NetXMS agent selected in *Object Browser*)

3.6.1 Overview

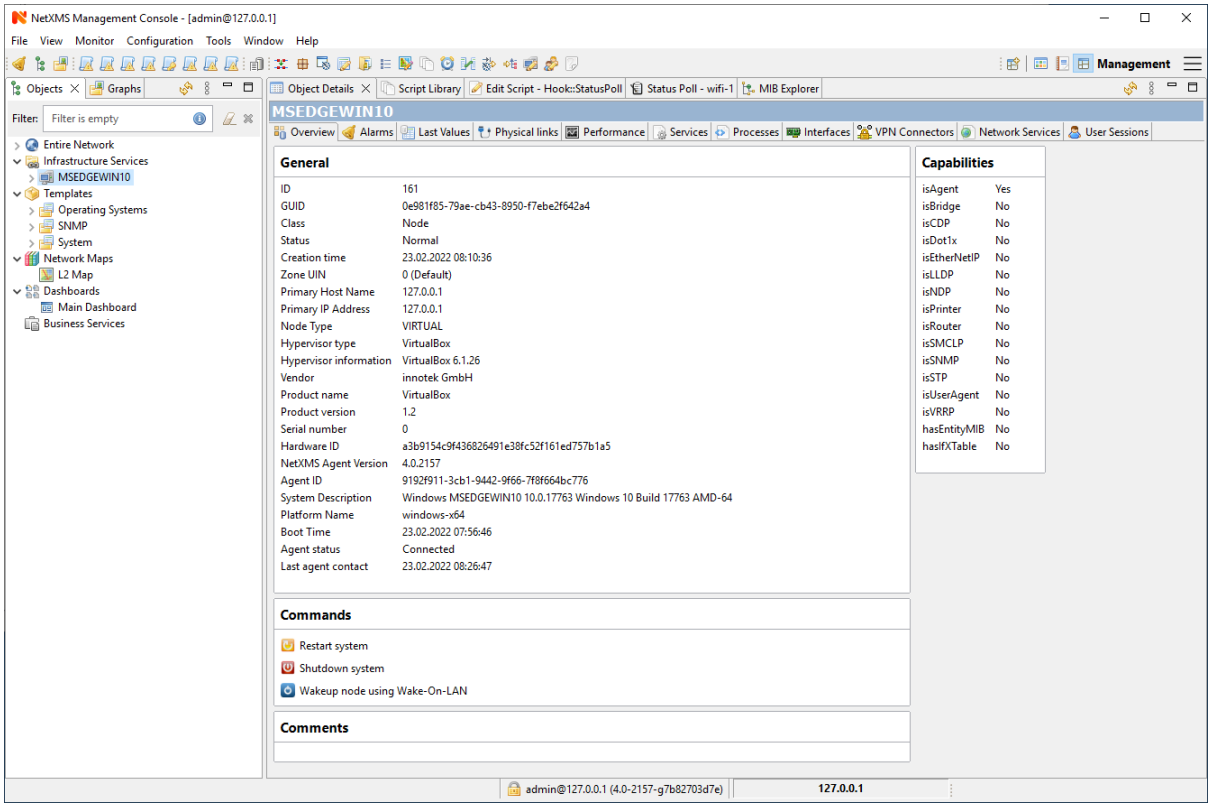


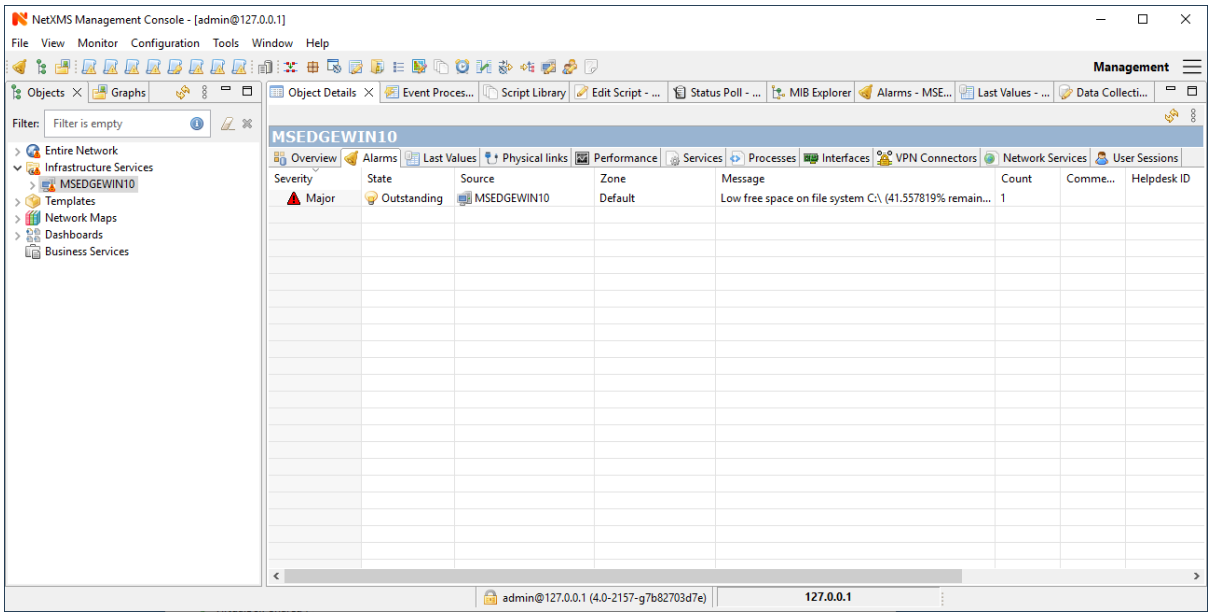
Fig. 8: Overview tab

This view provides basic information about selected object: Name, Class, Status and comments. For *Node* objects, it also show IP address, Host name, SNMP details as well as Capabilities.

Node capabilities

Capability	Description
isAgent	True if NetXMS Server can communicate with NetXMS agent installed on the node
isRouter	True if selected object can route network traffic
isSNMP	True if selected object is SNMP-capable

3.6.2 Alarms



Alarm view provides user with list of alarms for currently selected element of the tree, including all child objects. To view all alarms in the system, either use system-wide *Alarm Browser* (click *View* → *Alarm Browser* to open) or select *Entire Network* object. Right-click on the alarm will open pop-up menu with available actions

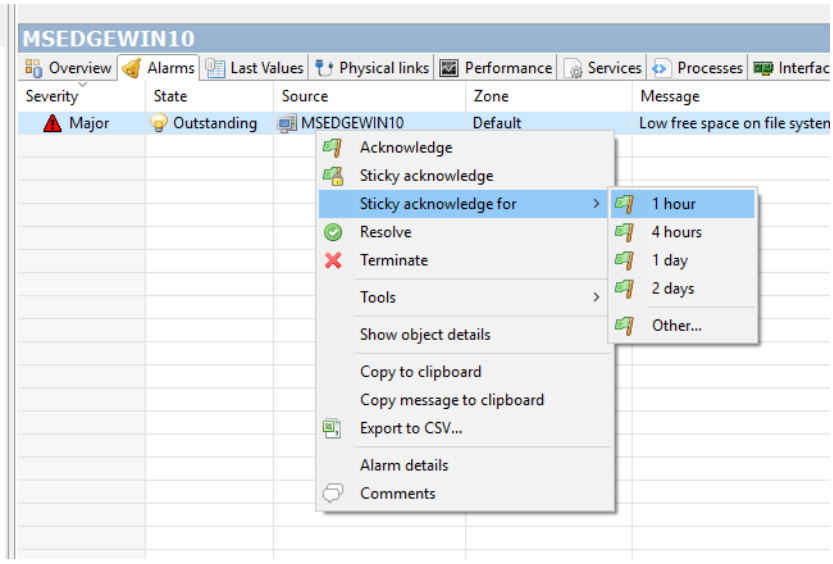
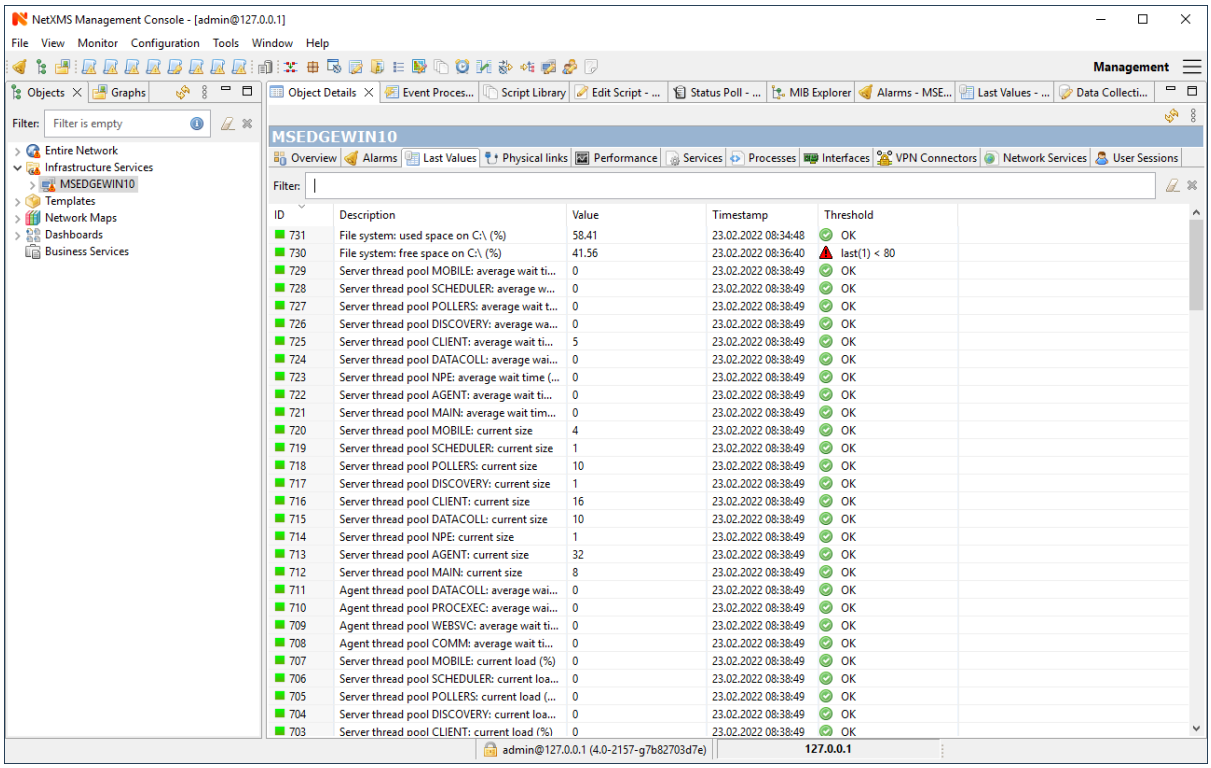


Fig. 9: Alarm context menu

Each alarm can be in one of three different states:

State	Description
Outstanding	Newly created alarm, no actions was taken by user
Acknowledged	User acknowledged raised issue, work in progress
Resolved	Issue resolved, but alarm is kept in the list. This state mostly used when alarm is automatically resolved by the system, to keep users informed about incident
Terminated	Issue resolved and alarm removed from list.

3.6.3 Last Values



This view provides access to all collected data, both latest and historical. When view is shown, it displays latest values, as well as timestamp when each value was collected. Threshold column indicates threshold violations for given *DCI*. User has two options to interact with data:

- Double click on the *DCI* will open line graph view for last hour
- Right-click on the *DCI* will open pop-up menu giving access to all available actions
  - *History* - show historical data
  - *Line Chart*, *Pie Chart*, *Bar Chart* - show historical data in graphical form
  - *Clear collected data* - remove all history for selected *DCI*

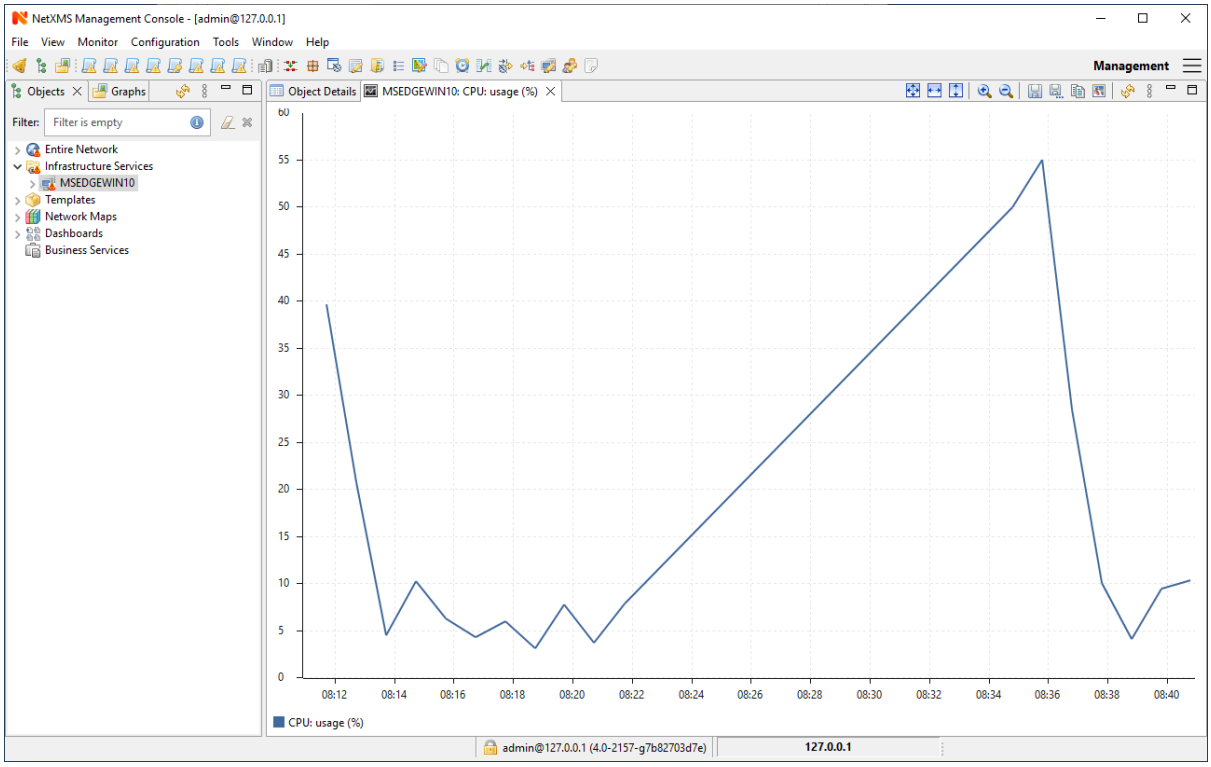


Fig. 10: Line graph built from collected data

3.6.4 Performance Tab

Performance tab is a special view that allows to quickly assess health of the selected node using one or more graphs predefined by administrator. Each graph can contain data from multiple sources.

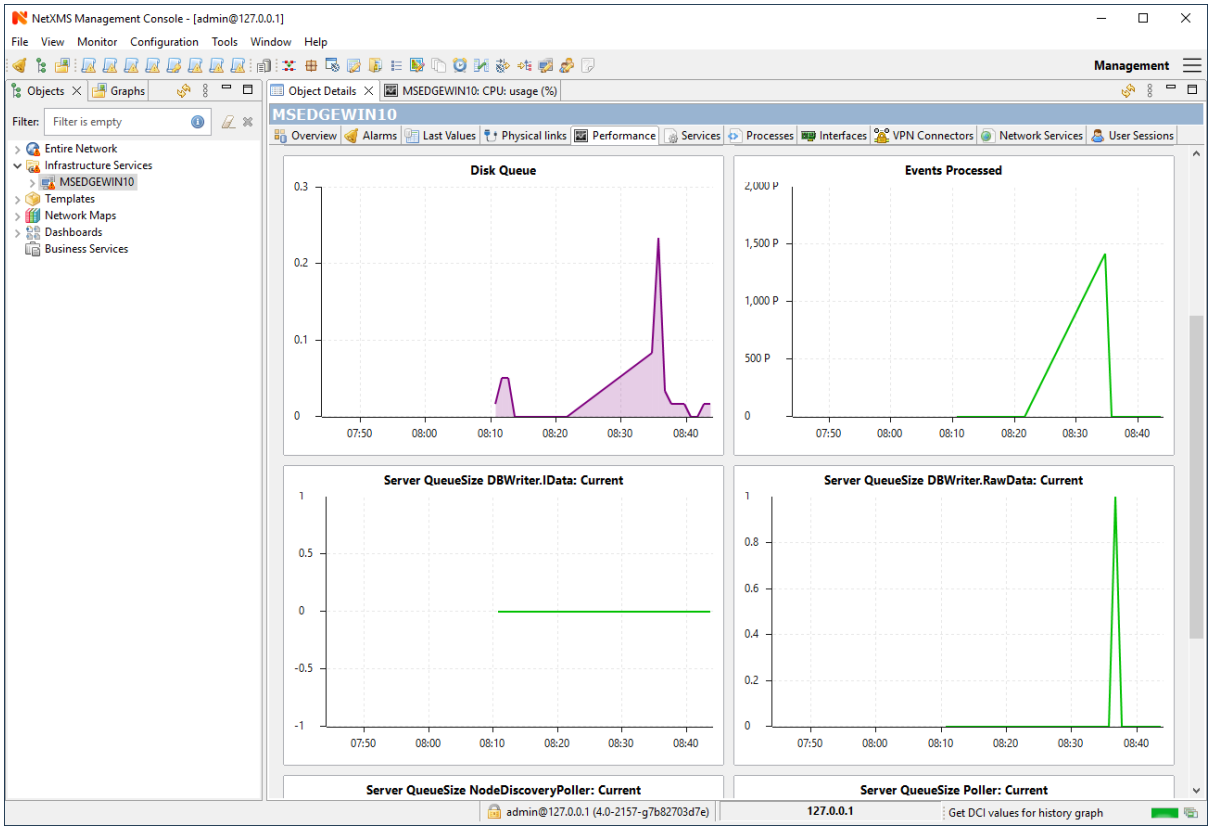


Fig. 11: Router's CPU usage displayed

3.7 Network Maps

This view allows user to see network overview in a map form. Map can be build and routed either manually or automatically for selected part of the network. Maps can be automatically generated based on:

- IP topology, both Level 2 and Level 3
- Geographical location of the objects
- Object relations

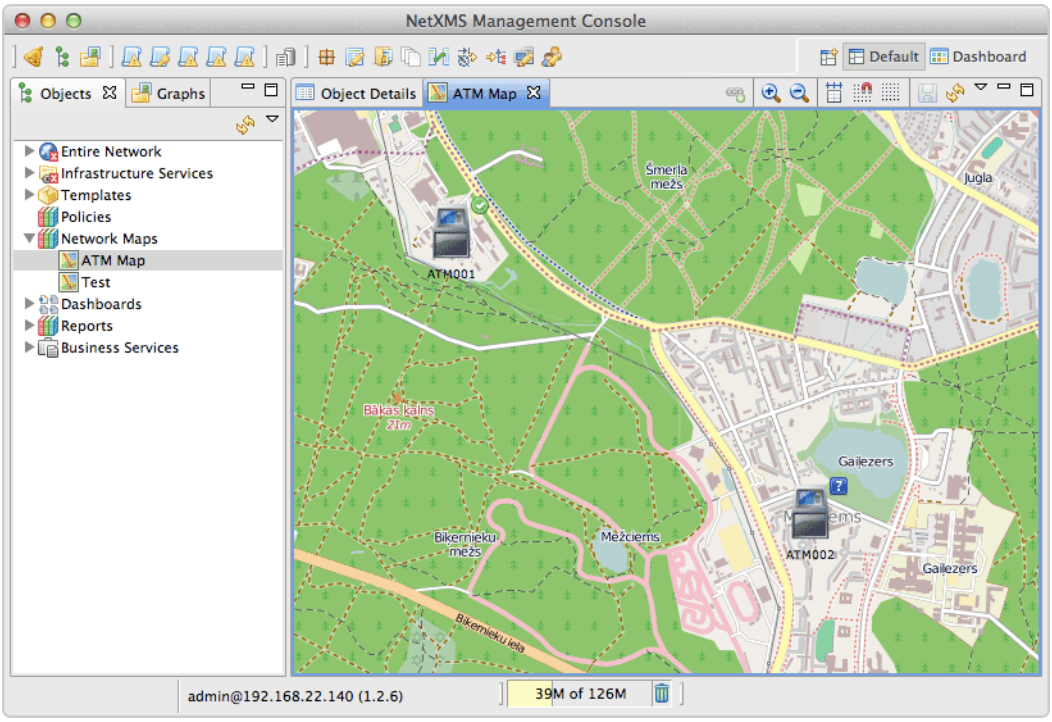
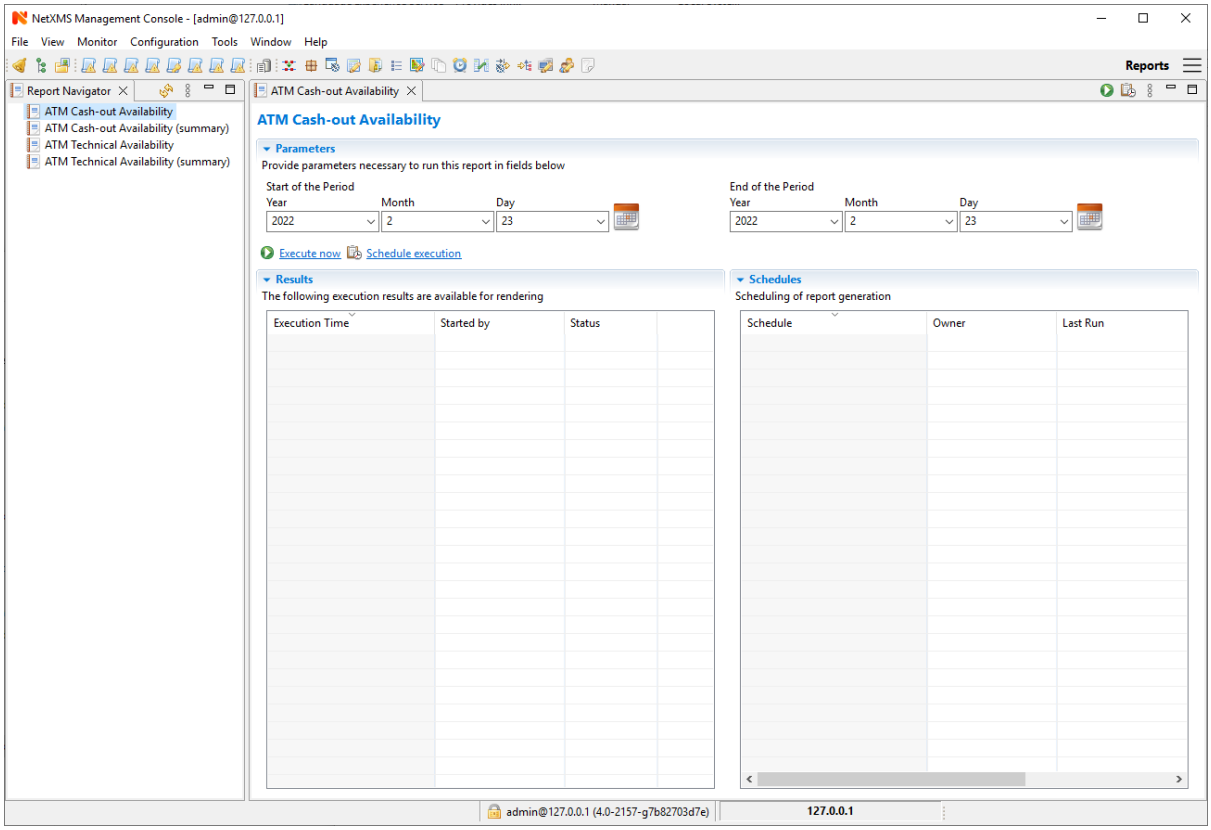


Fig. 12: Geo map showing part of the ATM network

To open existing map, either double click on the name in *Object Browser* or right-click and select *Open map* in pop-up menu.

3.8 Reports

NetXMS is integrated with *Jasper* reporting engine from Jaspersoft. This view allows user to generate report and download result as PDF file. Report generation can take long time, so it's done in background, without user interaction. When report is generated, resulting PDF can be downloaded any time, as well as any result from previous runs.



To generate report:

- Right-click on report name in *Object Browser* and select *Open report* in pop-up menu, report view will open (as show in figure above)
- In report view, fill parameters and click *Generate Report*

You can monitor progress in *Server Jobs* view. To open it, select *Window* → *Show view* → *Other* → *Server Jobs*.

When report is generated, new finished job will appear in *Results* table of the view. Select it and click on *Render to PDF* to download.

When generated report data is not longer needed, it can be deleted from the sever by selecting job in *Results* view, and then clicking *Delete*.

3.9 Dashboards

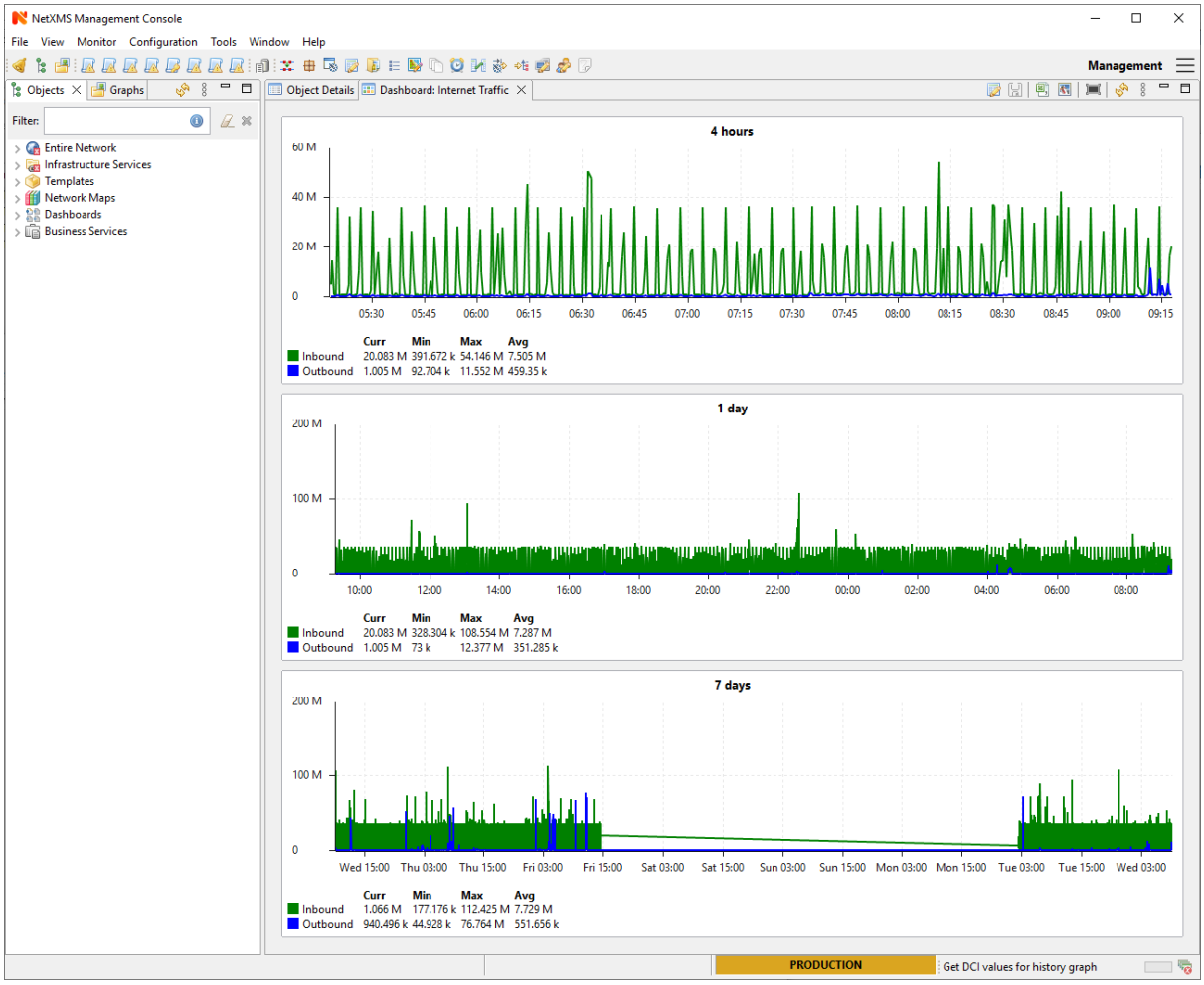


Fig. 13: Dashboard showing traffic information from core router.

Dashboards are defined by administrator and allow to combine any available visualization components with data from multiple sources in order to create high-level views to see network (or parts of it) health at a glance. There are two ways to access dashboards:

- Open dashboard from *Object Browser*
- Switch to *Dashboard* perspective and select dashboard with left-click

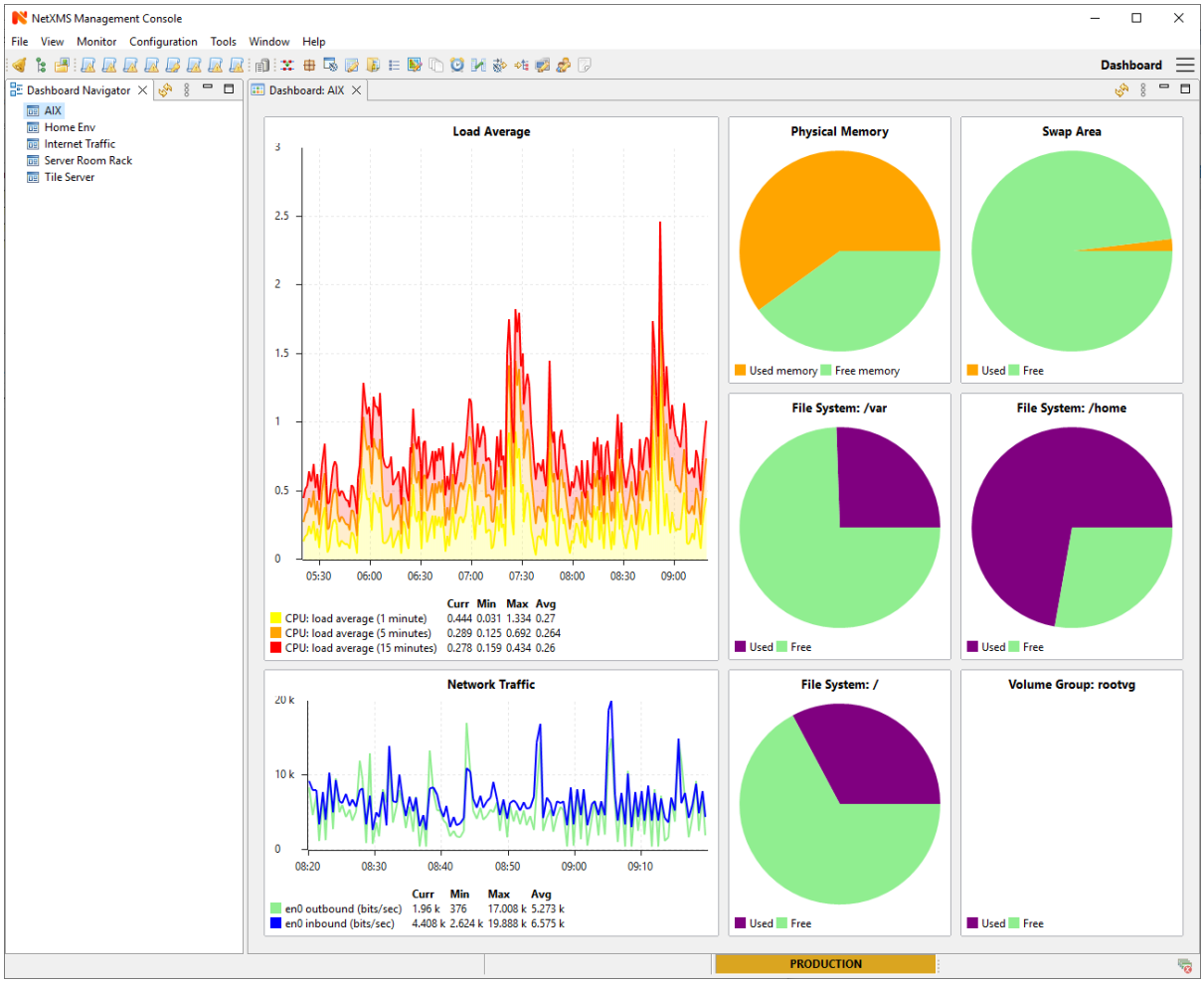


Fig. 14: Dashboards perspective

3.10 Business Services

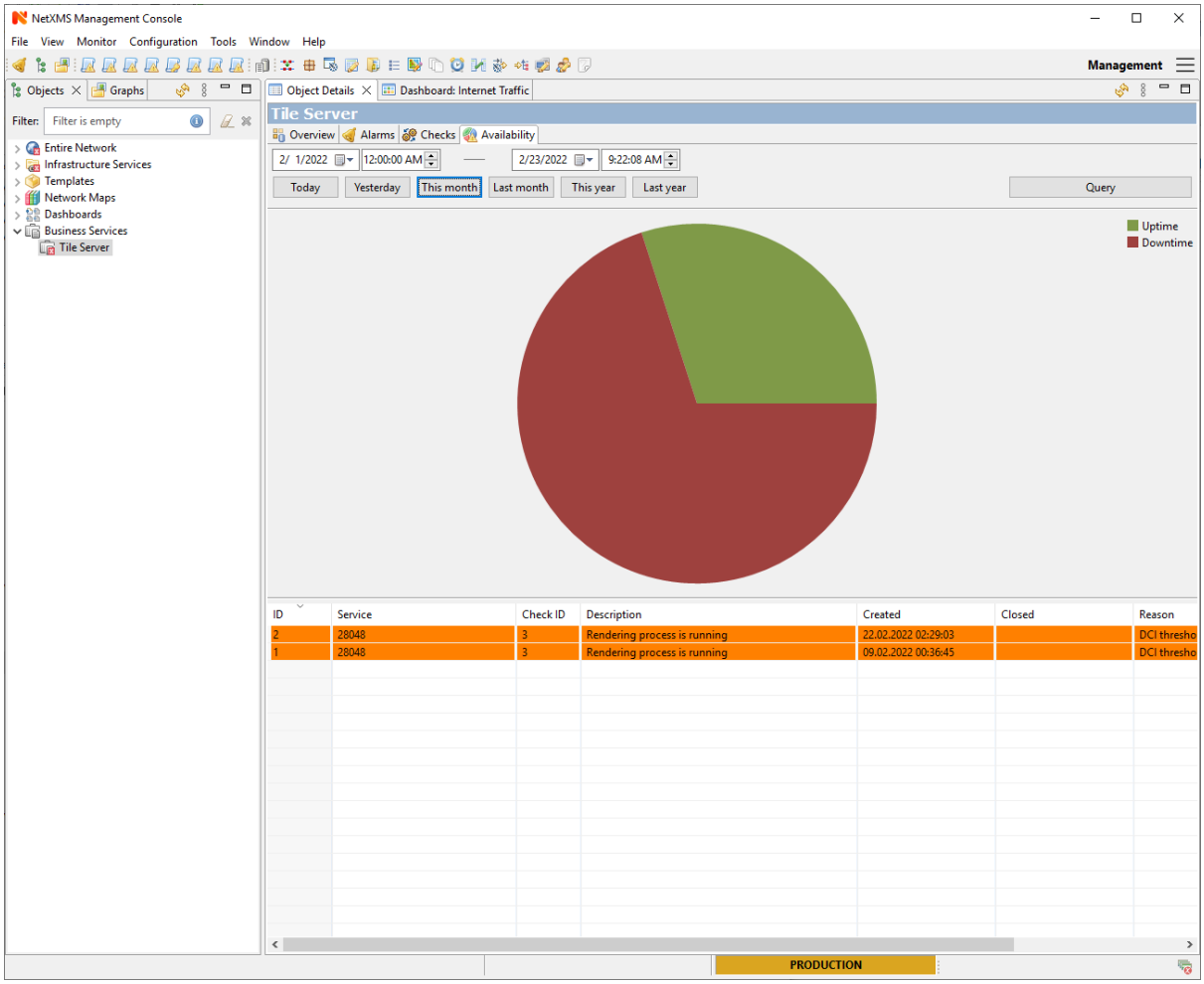


Fig. 15: Availability chart and uptime percentage for a system.

Business Services is a hierarchy of logical services as defined by administrator. Each service can represent combined state of multiple elements. For each service in the hierarchy, NetXMS calculates availability percentage and keeps track of all downtime cases. To check availability of any particular level, select it in *Object Browser*.

CHAPTER

FOUR

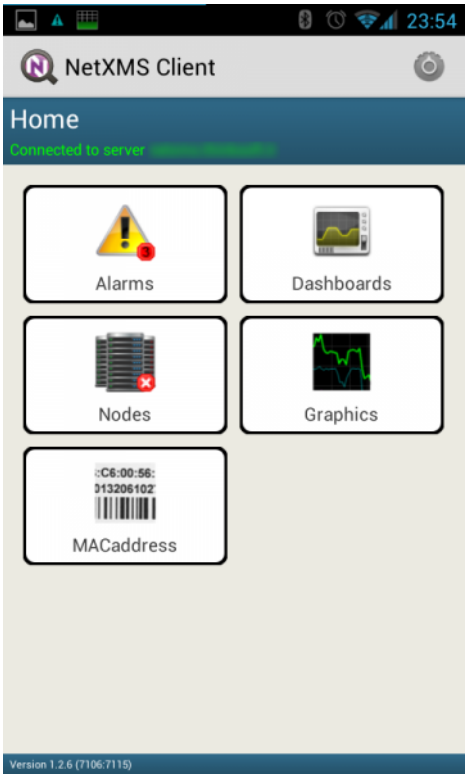
MOBILE CONSOLE

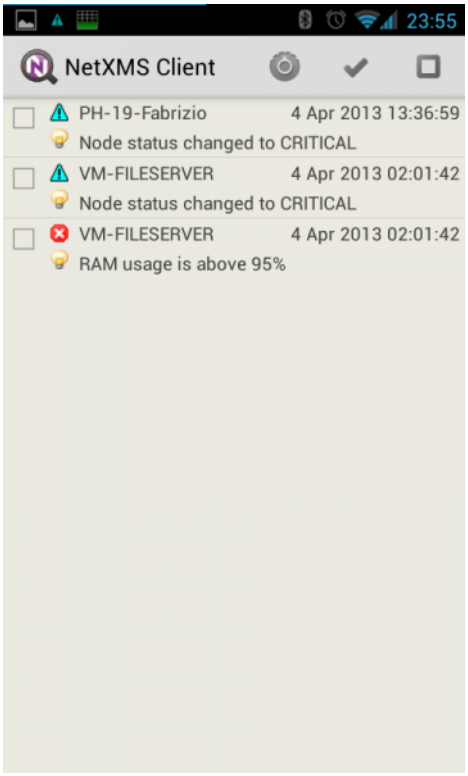
NetXMS mobile console is a monitoring tool for Android devices running version 2.2. and later.

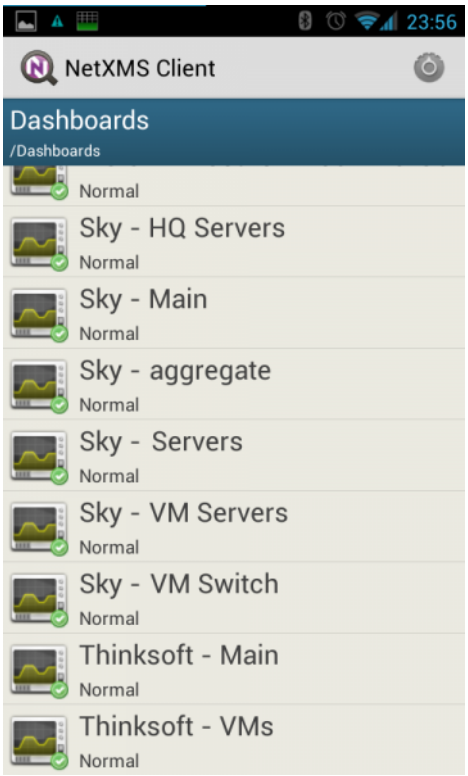
Currently, only a small subset of the functions present in the Desktop/Web edition are implemented, mainly read/only operations. The next paragraphs briefly describes each section.

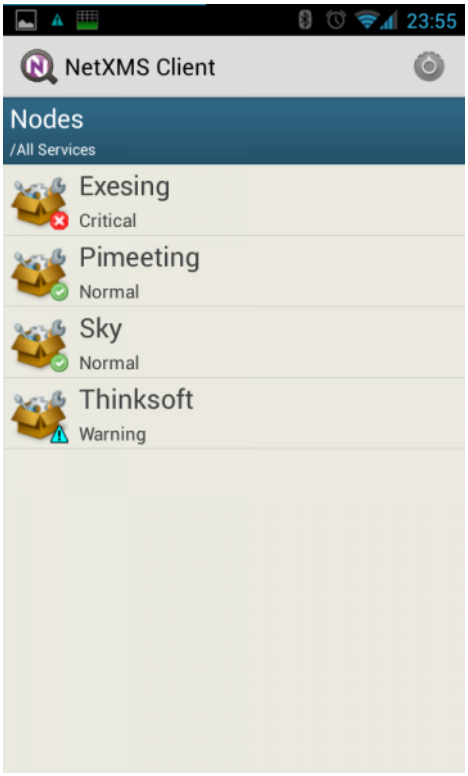
4.1 Main window

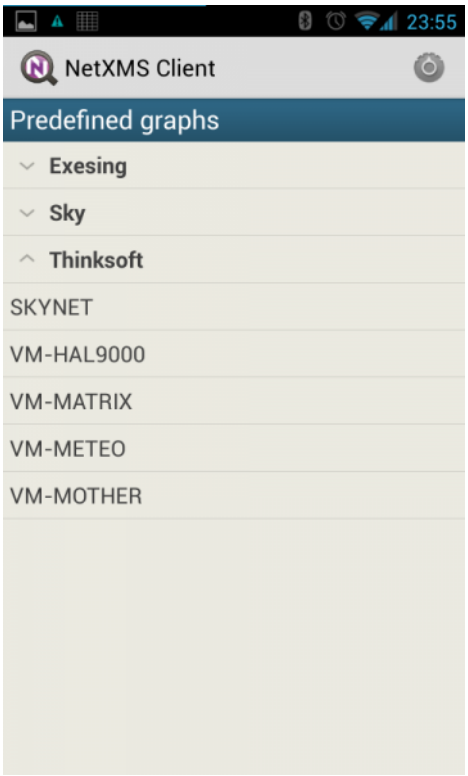
Here you can see how appears the main window and the underneath levels.

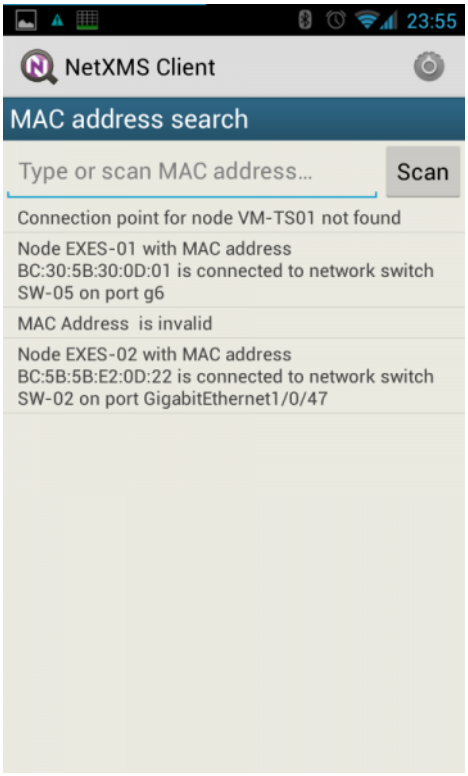



23:55
NetXMS Client
PH-19-Fabrizio
4 Apr 2013 13:36:59
Node status changed to CRITICAL
VM-FILESERVER
4 Apr 2013 02:01:42
Node status changed to CRITICAL
VM-FILESERVER
4 Apr 2013 02:01:42
RAM usage is above 95%


23:56
NetXMS Client
Dashboards
/Dashboards
Normal
Sky - HQ Servers
Normal
Sky - Main
Normal
Sky - aggregate
Normal
Sky - Servers
Normal
Sky - VM Servers
Normal
Sky - VM Switch
Normal
Thinksoft - Main
Normal
Thinksoft - VMs
Normal


23:55
NetXMS Client
Nodes
/All Services
Exesing
Critical
Pimeeting
Normal
Sky
Normal
Thinksoft
Warning


23:55
NetXMS Client
Predefined graphs
Exesing
Sky
Thinksoft
SKYNET
VM-HAL9000
VM-MATRIX
VM-METEO
VM-MOTHER



From the main window it is possible to get access to the following menu items:

- *Settings*: select this item to configure the console.
- *Reconnect*: select this item to force a reconnection to the server to gather new collected data.
- *Disconnect & Exit*: select this item to stop the console and exit from the app.

Underneath levels have menu that are context dependent, a detailed description can be found in each section.

4.2 Alarms

Alarms section is used to list and manage all pending alarms, eventually filtered on a particular node/container. Through this view it is possible to manage alarms:

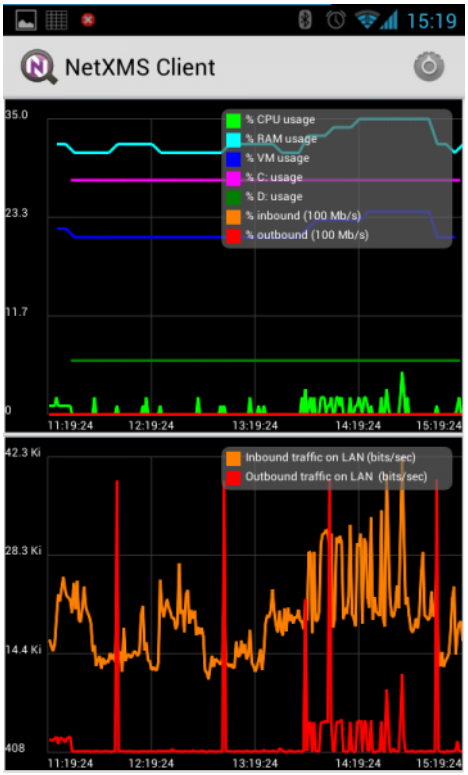
- ***Actions*:**
  - *Acknowledge*: acknowledge the alarm.
  - *Sticky acknowledge*: sticky acknowledge the alarm.
  - *Resolve*: resolve the alarm.
  - *Terminate*: terminate the alarm.
  - *View last values*: jump to the node info section to view the last values for the node that generated the alarm.
- ***Sort*:**
  - *Sort by severity ascending*: sort list using event severity as criteria, ascending.
  - *Sort by severity descending*: sort list using event severity as criteria, descending.
  - *Sort by date ascending*: sort list using date of event as criteria, ascending.
  - *Sort by date descending*: sort list using date of event as criteria, descending.

  - *Sort by node name ascending*: sort list using node name that generated the event as criteria, ascending.
  - *Sort by node name descending*: sort list using node name that generated the event as criteria, descending.

- *Select all*: select all the alarms from the list
- *Unselect all*: clear any selection of alarms from the list

4.3 Dashboard

Dashboards are defined by administrator and allow to combine any available visualization components with data from multiple sources in order to create high-level views to see network (or parts of it) health at a glance. Not all elements are currently available for the mobile console, dashboards are properly refreshed according to their schedule. Due to dashboard size, keep in mind that Smartphones cannot be the best device to show them, a tablet is much more suitable device. Here an example:

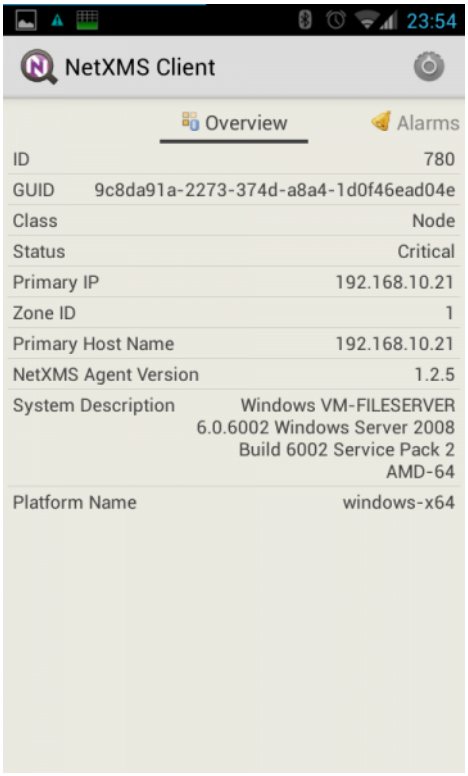


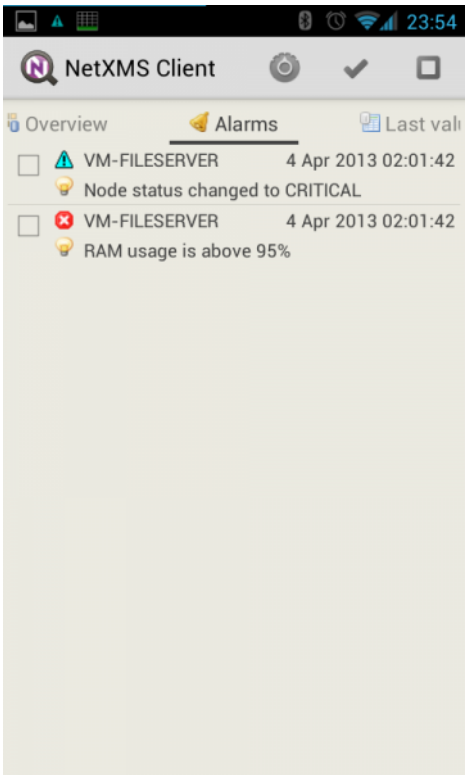
4.4 Nodes

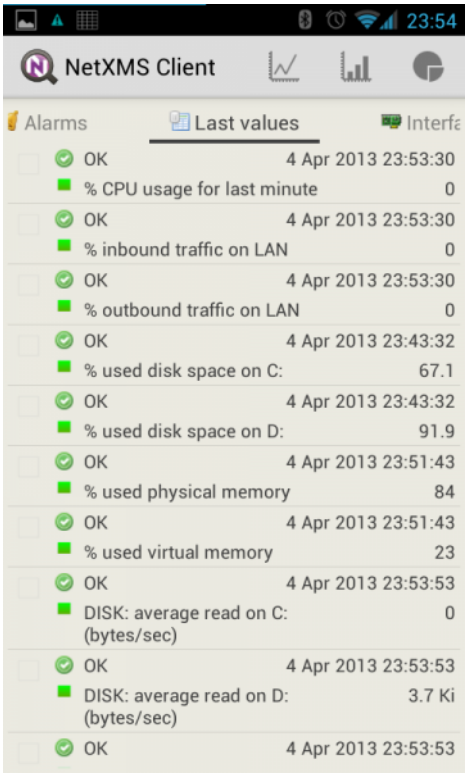
This section is used to list and manage all nodes (all network infrastructure monitored by NetXMS are represented as a set of objects. Each object represents one physical or logical entity, or group of them). Objects can be organized into hierarchical structure, the Nodes section is used to explore them. In the right bottom corner of the icon there is a symbol that indicates the status of the node/container following the same simbology used on the desktop console. Clicking on a container will show the items inside, continuing to click up to an object will show a set of swipeable pages:

- *Overview*: here are presented the main info associated to this node, such as the name, the primary IP, the status, etc.
- ***Alarms*: here are presented the list of pending alarms (if any) for this node, with the possibility to manage them with the following commands:**

  - ***Actions*:**
    - *Acknowledge*: acknowledge the alarm.
    - *Sticky acknowledge*: sticky acknowledge the alarm.
    - *Resolve:* resolve the alarm.
    - *Terminate*: terminate the alarm.
    - *View last values*: jump to the node info section to view the last values for the node that generated the alarm.
  - *Select all*: select all the alarms from the list
  - *Unselect all*: clear any selection of alarms from the list
- ***Last values*: here are presented the DCI collected for this node, as well as the possibility to draw the following graphics (for one or more values):**
  - *Last half hour*: draw one or more line graphs for the last half hour collected values
  - *Last hour*: draw one or more line graphs for the last hour collected values
  - *Last two hours*: draw one or more line graphs for the last two hours collected values
  - *Last four hours*: draw one or more line graphs for the last four hours collected values
  - *Last day*: draw one or more line graphs for the last day collected values
  - *Last week*: draw one or more line graphs for the last week collected values
  - *Bar chart*: draw a bar chart with the last collected value
  - *Pie chart*: draw a pie chart with the last collected value
- ***Interfaces*: here are presented all the interfaces associated to this node. For each interface it is possible to instruct the following commands:**
  - *Manage*: interface will be put in manage state
  - *Unanage*: interface will be put in unmanage state
  - ***Change expected state*: change the expected interface state, possible values:**
    - *UP*: interface expected state will be put in UP state
    - *DOWN*: interface expected state will be put in DOWN state
    - *IGNORE*: interface expected state will be put in IGNORE state
- *Find switch port*: will start the search for a connection point (if available)


23:54
NetXMS Client
Overview
Alarms
ID 780
GUID 9c8da91a-2273-374d-a8a4-1d0f46ead04e
Class Node
Status Critical
Primary IP 192.168.10.21
Zone ID 1
Primary Host Name 192.168.10.21
NetXMS Agent Version 1.2.5
System Description Windows VM-FILESERVER 6.0.6002 Windows Server 2008 Build 6002 Service Pack 2 AMD-64
Platform Name windows-x64

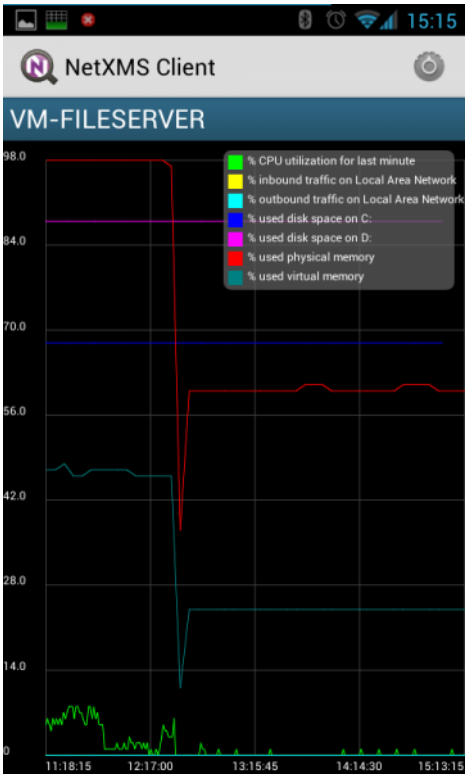

23:54
NetXMS Client
Overview
Alarms
Last val
VM-FILESERVER 4 Apr 2013 02:01:42
Node status changed to CRITICAL
VM-FILESERVER 4 Apr 2013 02:01:42
RAM usage is above 95%


23:54
NetXMS Client
Alarms
Last values
Interfa
OK 4 Apr 2013 23:53:30
% CPU usage for last minute 0
OK 4 Apr 2013 23:53:30
% inbound traffic on LAN 0
OK 4 Apr 2013 23:53:30
% outbound traffic on LAN 0
OK 4 Apr 2013 23:43:32
% used disk space on C: 67.1
OK 4 Apr 2013 23:43:32
% used disk space on D: 91.9
OK 4 Apr 2013 23:51:43
% used physical memory 84
OK 4 Apr 2013 23:51:43
% used virtual memory 23
OK 4 Apr 2013 23:53:53
DISK: average read on C: (bytes/sec) 0
OK 4 Apr 2013 23:53:53
DISK: average read on D: (bytes/sec) 3.7 Ki
OK 4 Apr 2013 23:53:53


23:54
NetXMS Client
ıst values
Interfaces
Local Area Connection
ID 1977
Name Local Area Connection
ifType 6
ifIndex 10
Slot 0
Port 0
Description Local Area Connection
MAC Address 0A:15:5D:0A:02:10
IP Address 192.168.10.21
Admin State UP
Oper State UP
Exp. State UP
Status Normal

4.5 Graphics

Predefined graphics are defined by administrator and can be used to view collected data in a graphical form (as a line chart). Currently, the mobile console doesn't autorefresh the content of the graphic selected. Here an example of a predefined graphs:



4.6 MACaddress

This section is used to list previously searched MAC addresses or to start a new search by scanning a barcode value (this feature needs the installation of Barcode Scanner from Zxing Team – freely available on the Google Play), by input it manually or by getting it directly from a node via the "Find Switch port" command.

4.7 Settings

This section is used to configure the behaviour of the console.

4.8 Global settings

- *Autostart on boot*: check to automatically start the agent on boot (to be effective, app must not be moved to SD card).

4.9 Connection

4.9.1 Parameters

Allows selecting the parameters used to connect to the server:

- *Server*: address of the server (IP or name).
- *Port*: port of the server (default 4701).
- *User name*: username to connect to the server.
- *Password*: password to connect to the server.
- *Encrypt connection*: when selected challenges an encryption strategy with the server (depending on supported/configured providers).

4.9.2 Scheduler

Enables the possibility to define periodic connections to the server. If the scheduler is not enabled the app will try to connect to the server every time it detects a new connection (data or WiFi) and remains always connected as far as the connection remains active:

- *Enable scheduler*: check this to enable the scheduler.
- *Frequency (min)*: amount of time, in minutes, that has to elapse between each tentative of connection to the server to send the gathered info.
- *Duration (min)*: amount of time, in minutes, that has to elapse before disconnect from the server.
- ***Daily scheduler*: provides the ability to define a "one range" daily on which the agent is operational. Out of the specified range the app will not try to connect to the server to gather the new events:**
  - *Daily activation on*: start time for daily activation.
  - *Daily activation off*: stop time for daily activation.

4.10 Notifications

4.10.1 Connection status

This section is to manage the notifications related to the connection status.

- ***Notification behaviour*: defines which kind of action should trigger notifications to the user. Possible options:**
  - Never: ignore connection status
  - When connected: notify when connection is successful
  - When disconnected: notify when connection is unsuccessful
  - Always: notify either connection successful and connection unsuccessful
- *Toast notification*: provides connection notification via "toast" , behaviour is defined by "Notification behaviour".
- *Icon notification*: provides connection notification via icon in the status bar, behaviour is defined by "Notification behaviour".

4.10.2 Alarms

- *Alarms notification*: select to enable alarms notification in the status bar.
- ***Alarms sound by severity*: for each of the following categories:**
  - *Normal*
  - *Warning*
  - *Minor*
  - *Major*
  - *Critical*

4.11 Interface

4.11.1 Multipliers

Allows to select the preferred multipliers to be used to show values. Allowed options: * *None*: do not apply multiplier, values are extended. * *Decimal*: applies a decimal multiplier (power of 10, e.g. 1000 -> 1K, 1000000 -> 1M, ...) * *Binary*: applies a binary multiplier (power of 2, e.g. 1024 -> 1Ki, 1048576 -> 1Mi, ...)

4.11.2 Graph text size

Allows to set the text size to be used for axis labels (if the default value is too small for high density devices).

4.11.3 Show legend in graphs

Allows to select to show or not the legend in the top right angle of the graphs. Since legend can be intrusive, especially when there are several lines plotted, user can select to disable the legend.

CHAPTER

FIVE

OBJECT MANAGEMENT

5.1 Object browser

Object browser organize all existing *objects* in hierarchical structure. NetXMS has eight top level objects – Entire Network, Service Root, Template Root, Policy Root, Network Map Root, Dashboard Root, Report Root, and Business Service Root. These objects served as an abstract root for appropriate object tree. All top level objects has only one editable attribute – name.

Overall description about objects can be found in concepts part: *Objects*.

5.1.1 Properties

Object browser has next options:

- Show filter `CTRL+F2`, that shows search line that has special syntaxes for search. Syntaxes description can be found there: *Filters*.
- Show status indicator `CTRL+F3`
- Hide unmanaged objects
- Hide check templates. This option will not show *Business Services* templates.

5.1.2 Filters

Buy default search is done by node name. In this type of search can be used '*' and '?' symbols for pattern search.

But there are few prefix that can be used for other search options:

- '/' - will search in comments
- '>' - will search by IP address

5.2 Objects

Detailed information about objects, it's usage, parents and childes can be found in concept chapter, *Objects*. In this section will be described only actions and properties that can be applied on different object classes.

Next chapters will describe

5.2.1 Subnet

Menu items:

Full subnet can be managed or unamanged. Management status will be applied to all subnet node. If subnet is deleted and is the only parent of a node, then node also will be deleted with the subnet. *Upload file* menu item will upload file from server to all nodes that have agent and have access to upload directory.

Under *Tools* menu are available predefined object tools that will be executed on each subnet node. More about object tool configuration can be found there: *Object Tools*.

Alarms menu item will open view with all subnet nodes' alarms. And *802.1x port state* will open table with port authentication states, that can be exported to CSV.

5.2.2 Node

Menu items:

When node is unmanaged/managed - all it's childes like interfaces and service monitoring are also unmanaged/managed. In unmanaged state *metrics* are not collected and no pols are scheduled.

If zones are enabled, then zone can be changed using *Change zone...* item. *File manager* will open agent file manager view. *Upload file* can be used to upload file from server to node. This action can be applied simultaneously to all nodes.

Take screenshot for now halfway implemented functionality. For now screenshot can be taken only from Windows machines.

Poll options:

Poll Name	Description
Status	
Configuration	
Configuration (full)	
Instance discovery	
Instance names	
Topology	

Under *Tools* menu are available predefined object tools that will be executed on selected node. More about object tool configuration can be found there: *Object Tools*.

If geolocation of the node is set, then with help of *Geolocation* item can be opened map with shown on it object location. *Software Inventory* will show full software list for nodes with Windows systems or Linux systems(that used rpn or deb packages) and have NetXMS agent installed. *Service Dependency* will build tree from this node with all container where this node is included. *Alarms* will open alarm view with alarms only for this specific node.

Find switch port will open view with log of searchs of switch port that with which this node is connected. Wile search we will check one by one interfaces and will show first successful result.

802.1x port state will open table with port authentication states, that can be exported to CSV.

Topology menu item contains all options of predefined network maps for this node and some other options:

Routing table IP route from... will build network map with route from selected node to node that is selected from Object selector window. *IP route to...* will build network map with route to selected node from node that is selected from Object selector window. *IP Neighbors* will show all IP neighbors of this node.

Switch forwarding database(MAC address table) VLANs Layer 2 Topology

Radio interface Wirless stations

Last values will open *Last Values view*.

5.2.3 Mobile Device

Menu items:

Each phone object can be managed/unmanaged and deleted. In umnanaged state *metrics* of this device are not collected and no pols are scheduled. When mobile object is deleted all it's data is also deleted. No history data will be left.

Geolocation History will open view were will be shown history of displacement of this device. From the menu can be selected the period to show on history map. *Geolocation* will show last known location of this device. *Alarms* menu item will open view with all subnet nodes' alarms.

Last values will open *Last Values view*.

5.2.4 Rack

5.2.5 Cluster

5.2.6 Intrface

5.2.7 Network Service

5.2.8 VPN Connector

5.2.9 Condition

Conditions may represent more complicated status checks because each condition can have a script attached. Interval for evaluation of condition status is configured in Server Configuration Variables as ConditionPollingInterval with default value 60 seconds. Input values for the condition script can be set in object properties. Such values are accessible via $1, $2, ... variables inside the script. If the script returns 0, an activation event with the defined severity is created. If the script returns any other value, then a deactivation event is created.

Menu items:

Condition can be mangedunmanaged. If condition is unmanaged, evaluation of condition is not run. Condition can be deleted.

5.2.10 Container

Containers can be created in Infrastructure Services tree. Existing nodes and subnets can be added to containers by using Bind operation, and removed by using Unbind operation. New nodes, conditions, clusters, containers, mobile devices and racks can also be created. They can be created using required menu item of container under which this object should appear. Containers and nodes inside them can be moved by *Move to another container* menu item or using drag&drop.

Menu items:

There are special menu item for each object that can be created in container. Objects like rack, container, mobile device, cluster are manually created objects. Node can be manually created or found by network discovery. In case if it is required to add already existing object to container use *Bind...* menu item. To remove node from container, but do not delete it use *Unbind...* menu item.

Using *Manage/Unmanage* all nodes will be managed/unmanaged under container. Container can be deleted. If deleted container was the only one parent of the object, then this object will be also deleted. *Upload file...* will upload file from server to all nodes under container, same as each tool under *Tools* menu item will be executed on each node.

Geolocation will show location of container on geographic map.

Alarms will open alarm view with all active alarms for all children of this container. *802.1x port state* will open table with port authentication states of all devices that are under this container. This information can be exported to CSV.

5.3 Object Details

Object details view provides main information about object. Each object has *Overview* tab that gisplays general information about object (like: ID, GUID, Class, and status of the object) and *Comments*.

5.3.1 Subnet

5.4 Object Tools

There can be created tools that will be executed on objects. Tools are shown under "Tools" item of node menu. There are some pre defined object tools, but they can be disabled or configured new by NetXMS administrator.

5.5 Last DCI values View

CHAPTER

SIX

NETWORK TOPOLOGY

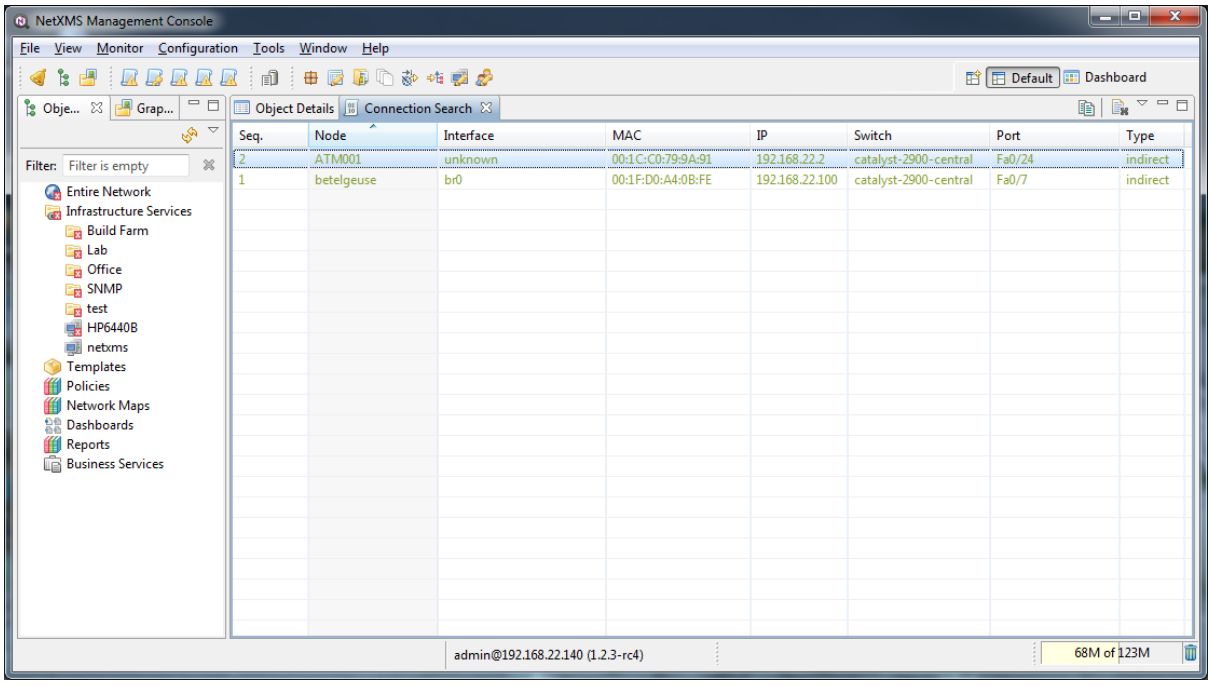
6.1 Introduction

NetXMS server automatically creates and maintains network model on different layers. All necessary information taken from ARP cache, routing tables, and switch forwarding database of managed nodes. Topology data provided by CDP, LLDP, and NDP (SONMP) protocols also used in building network model. Having network model instantly available allows NetXMS users to perform various network topology tasks much faster and easier.

6.2 How topology information built

6.3 Find where node is connected

It is possible to find switch port where any given node is connected (sometimes called "connection point" in management console). To find out node's connection point, right-click on node object, and select *Find switch port* in pop-up menu. Message box with search results will pop up, and if port is found, search results view will be opened (or updated if already open). Search results view looks like this:



Columns have the following meaning:

Seq.	Search result sequence number
Node	Name of end node object
Interface	Name of node's interface object
MAC	Interface's MAC address
IP	Interface's IP address
Switch	Name of switch node object
Port	Name of interface object representing switch port
Type	Connection type - direct or indirect. Direct connection type means that NetXMS server did not detect any other devices on sdame switch port, and most likely end node connected directly to the switch. Indirect means that some other devices was detected on same switch port. Virtual machines and virtual machine host will always be detected as indirect.

6.4 Find MAC address

It is possible to find location of any known MAC address in the network. To do this, select *Tools* → *Find MAC address*. Results of a search will be displayed in the same results view. It is not necessary that node with given MAC address be managed by NetXMS server, but if it is, appropriate details will be displayed.

6.5 Find IP address

It is possible to find location of any known IP address in the network. To do this, select *Tools* → *Find IP address*. Results of a search will be displayed in the same results view. It is not necessary that node with given IP address be managed by NetXMS server, but if it is, appropriate details will be displayed.

CHAPTER

SEVEN

GLOSSARY

Alarm Browser
: View, which shows all active alarms in the system and provides tools to interact with them

DCI
: Data Collection Item, configuration element, which contains parameter to collect (for example "CPU Usage"), collection schedule and thresholds

Entire Network
: Automatically generated hierarchy that contains all nodes known to NetXMS

Metric
: One entity of collected data

Node
: Object that represents physical server

Object
: Representation of logical or physical entity.

Trim Stack
: *View Stack* in minimized state, represented as a set of buttons, one for each View in the stack

View Stack
: Multiple views combined into single one, with tab navigation on top of it

INDEX