

NetCrunch

v9.3

NetCrunch Guide

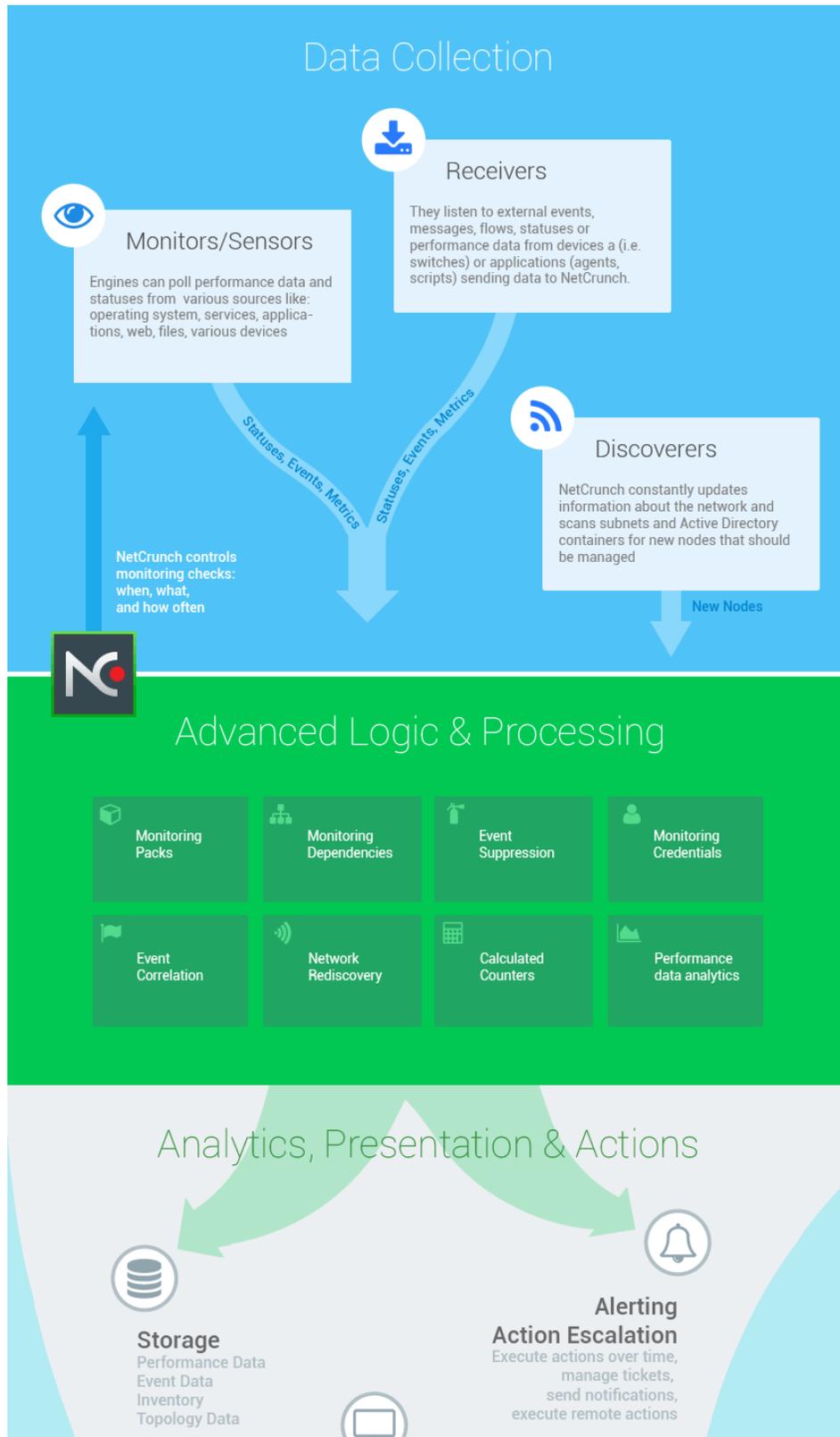
Program documentation is constantly updated with every new build. It is also available on-line.

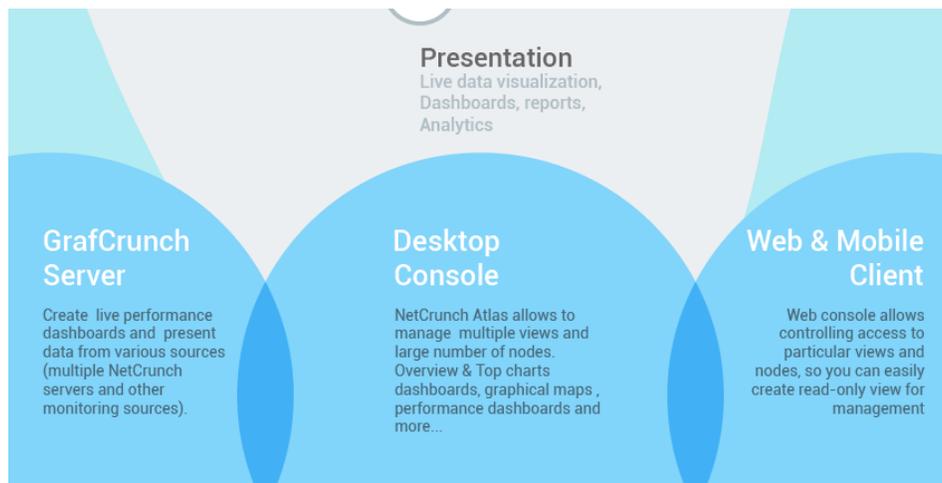
Please help us make it better. If you find any topic incomplete or missing - let us know. Click on the small icon at the top right corner and send us an anonymous comment.

System Overview

Discover all of NetCrunch capabilities, concepts and components.

In order to monitor today's complex networks, NetCrunch employs unique concepts for data organization and monitoring settings management.





Architecture and Concepts

Overview of NetCrunch Server architecture. Learn more about Monitoring Engines, NetCrunch Consoles, databases, additional tools, and key concepts of advanced network visualization.

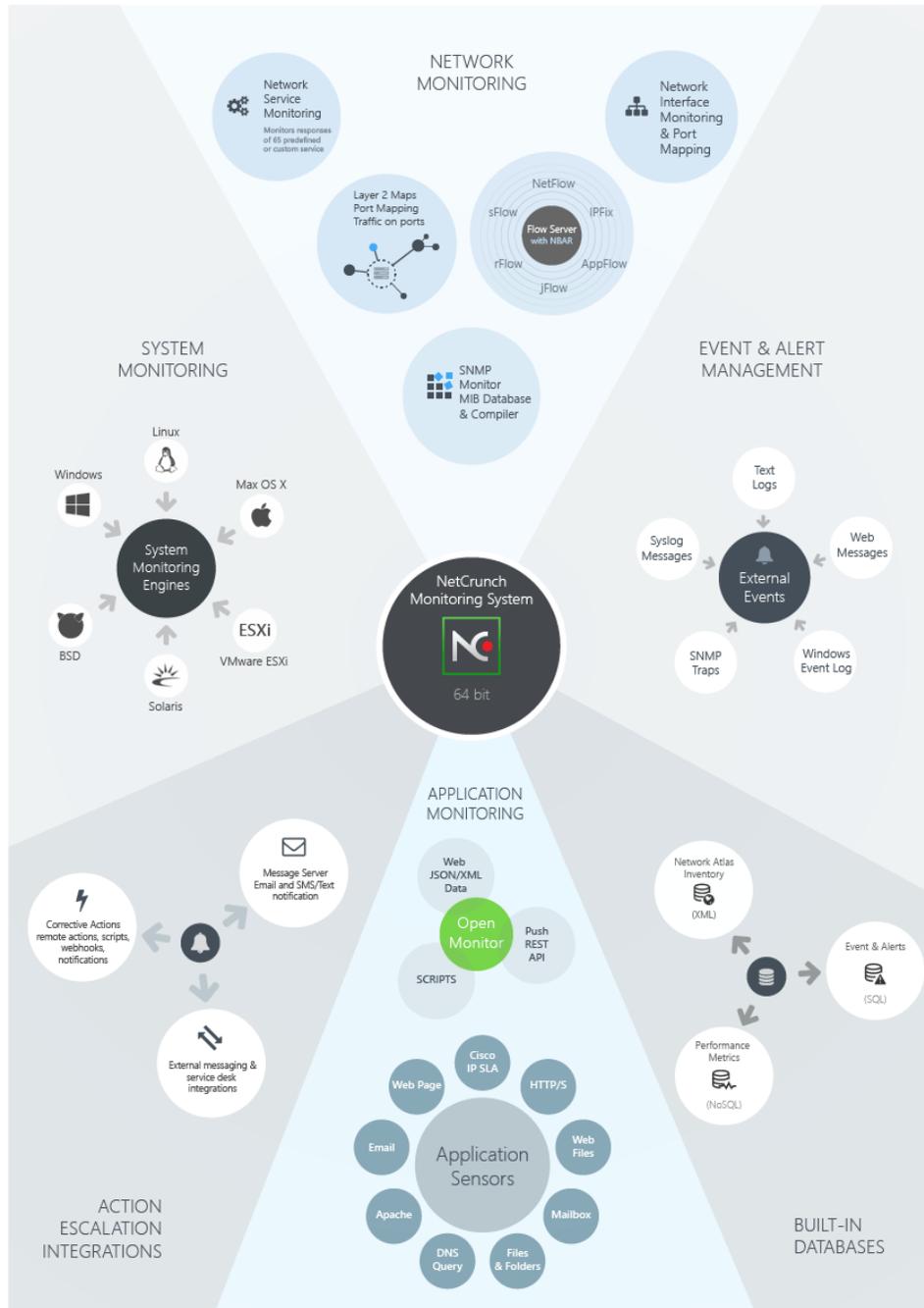
Architecture

NetCrunch is very comprehensive system consisting of many components communicating with each other. Most of them work on the NetCrunch Server. GrafCrunch Server can be installed on a separate machine.

The server should be a dedicated machine (it can be virtual) with the appropriate resources assigned. If you want to process Gigabytes of data then you need more than SATA disks or dual processor machines. Read more in [System Requirements](#).

NetCrunch Server also works well with vSphere Fault Tolerance which provides continuous availability for NetCrunch Server.

NetCrunch Server Components



Server Services

The picture above reflects a high-level organization of the NetCrunch components. The complete list of NetCrunch services is as follows:

NetCrunch Server

Central server providing monitoring logic, most of monitoring engines, trend storage and communication infrastructure for other components.

NetCrunch Advanced SQL Server

Provides storage, processing and controlled access to the Event Log data.

NetCrunch Message Server

Alerting notification service.

NetCrunch NetFlow Server

Collects and analyzes flow data (NetFlow, sFlow , IPFix and other) data.

NetCrunch Task Scheduler

Schedules various operations like backup, performance database maintenance, auto discovery, report generation and automatic distribution.

NetCrunch Data Updater

Updates various files used by NetCrunch.

NetCrunch Guard Service

Watches server execution.

NetCrunch WebApp Server

NetCrunch embedded Web Server provides services for Web Console, Mobile Console and REST API.

NetCrunch Open Monitor

NetCrunch interface to REST agents, allows to run custom monitoring scripts on NetCrunch Server.

NetCrunch Sensor Monitor

Runs growing list of NetCrunch application sensors, such as: Web Page, Email, File and Folder, FTP, HTTP, DNS, IP SLA and many others.

Monitoring Engines

Engines are responsible for collecting the monitoring data from various sources, including SNMP, various operating systems, ESX/i, Network Services and others. Some of them are part of the NetCrunch Server; many of them are run as separate processes.

Operating System monitoring engines such as: Windows, Mac OS X, Linux, Solaris, BSD or ESXi need a proper device type to be set. See: [Automatic Monitoring and Organizing](#).

Monitoring Sensor Engine

Many of the application monitoring sensors are implemented by the separate engine.

Consoles

Administration Console

NetCrunch can be accessed from the Desktop Administration Console, which you can install on any Windows system with TCP access to the NetCrunch Server. The console uses encryption and compression, so it can be used even through public internet connection.

The console caches large amount of data and only the changes are transferred over the network. As a result, all data changes instantly on the screen without any refreshing and delays. The console allows creating complex screen layouts (for multiple screens) and can save them. It also can show automated (rotating

screens) full screen views.

GrafCrunch

NetCrunch comes with a fork of the open source project Grafana - one of the top open source performance visualization projects. GrafCrunch greatly increases the possibilities of creating live performance dashboards and allows you to present data from various sources. GrafCrunch has its own installer and is tightly integrated with NetCrunch. It can be used to create dashboards from multiple NetCrunch Servers.

Web Console

It is AJAX and HTML5-based console, allowing instant access to the server data. The access can be managed through user accounts and access rights, and limited to particular screens and operations.

The console however, does not support all configuration operations such as editing graphical maps.

Mobile Access

Web-based service tailored for smartphones and tablets. It gives you the access to various views, node status and last alerts.

Basic Concepts

NetCrunch is dedicated to managing the monitoring of thousands of components. Instead of setting individual alerts and reports for each monitored node (which takes several minutes per node in other programs) many things are done automatically.

Network Atlas

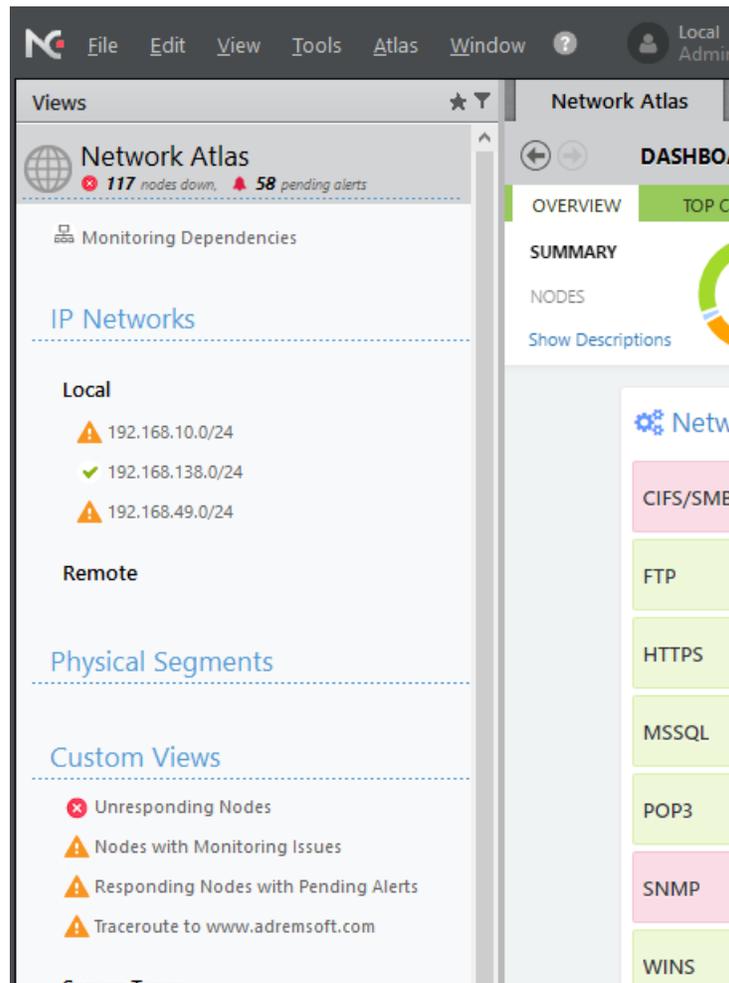
Network Atlas is a central database containing all your network data. It's organized by the hierarchy of the Atlas Node Views.

It contains all your network data and helps organizing them into various views. Many of them are created automatically.

Basic element of the atlas is a network node - single address network endpoint. Atlas Tree shows hierarchy of all views and helps you quickly recognize status of each element.

Atlas Views

Atlas Node View shows various aspects of the group of nodes in the Network Atlas and consists of multiple pages such as nodes, maps, dashboards and other.



Atlas begins with a top root view of all nodes. This view shows top-level dashboards such as: Status, Top Charts and NetFlow.

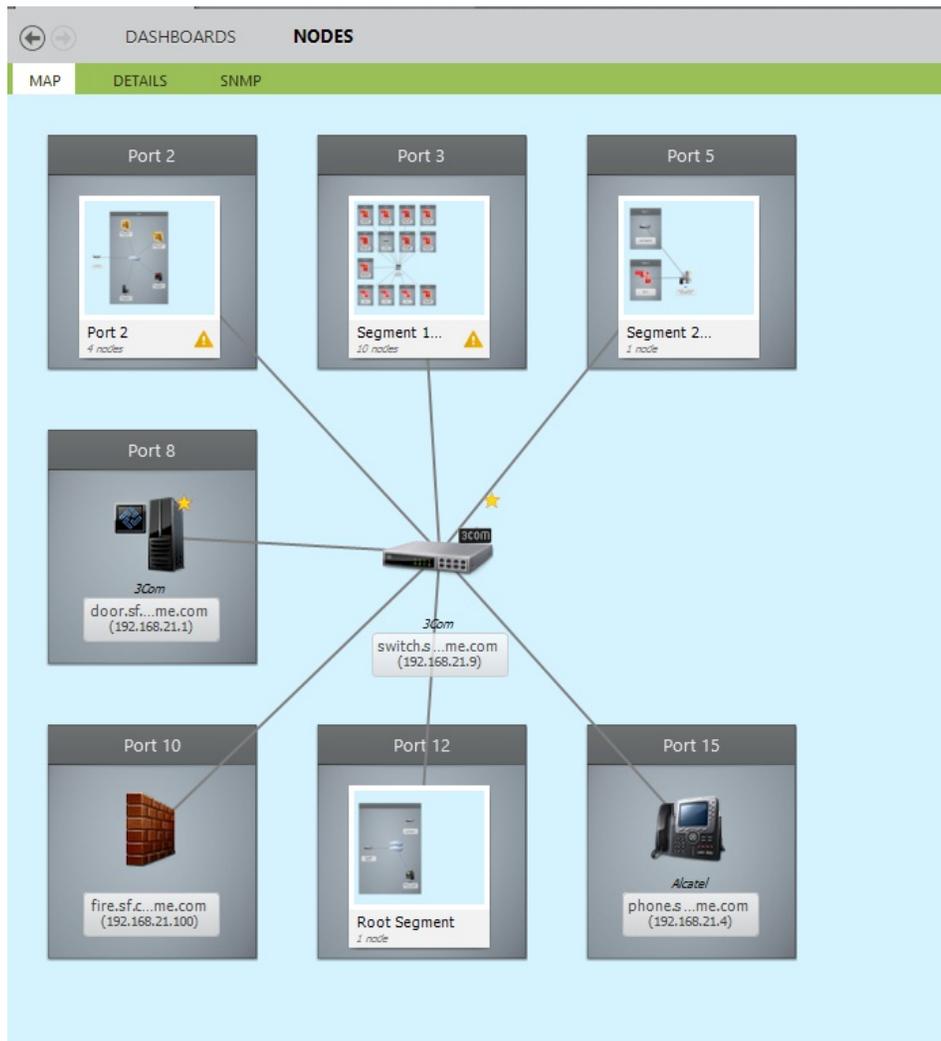
The rest of the views are divided into sections:

IP Networks

This section consists of IP network views/maps. Each network can be periodically re-scanned to reflect its current state. Usually, network maps are automatically arranged and nodes are grouped by device model and OS name.

Physical Segments

This section can contain many views showing Layer 2 connection segments hierarchy. Each view is automatically arranged and can also show the traffic summary on each switch port. To see Physical Segments maps you need to configure it first.



Custom Views

This section allows organizing your network data in any way you need. It contains both user created views and predefined automatic views.

Dynamic Views & Folders

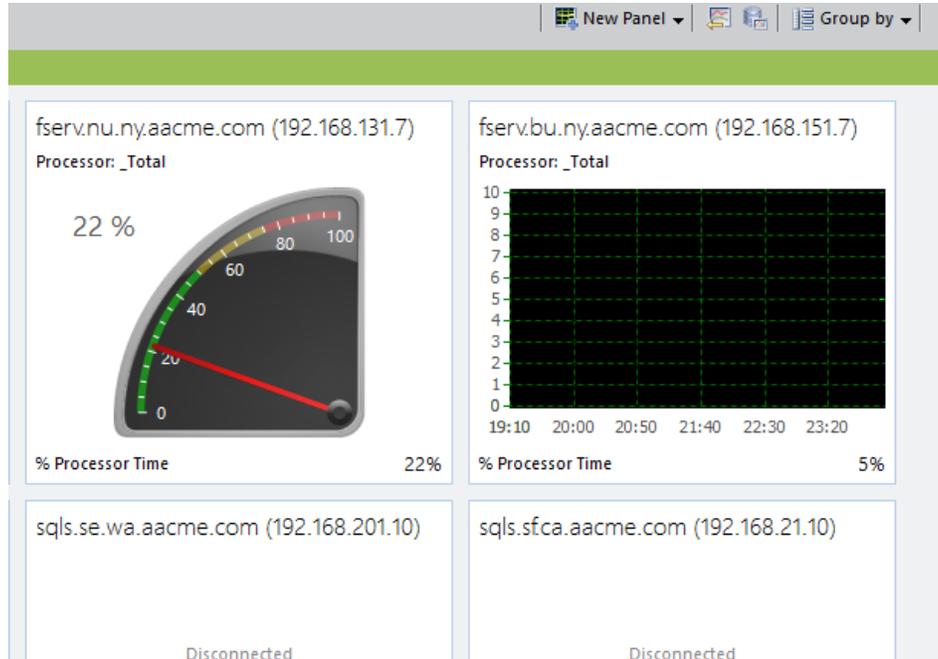
Basing on typical customer atlases we prepared many automatic (and dynamic) views for you such as:

- Nodes with Issues
- Un-responding Nodes
- Server Types (*i.e. Linux, Windows Server, etc.*)
- Device Groups (*Printers, Switches, Wireless, etc.*)
- Workstation Types (*Windows 7, Windows Vista, Windows 8, etc.*)
- Locations (*Office, Building 1, Server Room - based on SNMP or manually entered data*)
- Network Roles (*Network, Printers, Servers, Workstations*)
- Windows Domains

The views are dynamic, which means that they are automatically updated as needed.

Performance Views

Performance views consist of charts or gauges showing last read values of different parameters for multiple nodes. The charts can be grouped per nodes or per counter. In "per counter" mode, you can compare performance trends from multiple nodes on a single chart.



Monitoring Dependencies

Monitoring Dependencies reflect network connections and allow for preventing false alarms and disable monitoring of unreachable network components.

NetCrunch allows setting of dependencies automatically upon node routes, virtualization hosts and known switch *Layer 2* connections.

Monitoring Packs

Monitoring Pack is a group of performance parameters and events to be monitored and collected for the reports.

Monitoring Packs can be assigned to a node automatically (by a specific rule like *for every Windows Server* or *for every Cisco switch*), or manually. NetCrunch comes with many predefined [Monitoring Packs](#).

Automatic Monitoring by Device Types

Many predefined Monitoring Packs are automatically assigned to nodes upon node Device Type setting. The Device Type setting can be either automatically discovered or set manually.

That is why setting proper Device Type is one of the most important things in NetCrunch.

For network components retrieved from Active Directory - device types are likely to be set automatically. Many SNMP devices can also be discovered automatically.

Other devices like printers or Linux machines need proper setting of the Device Type to be monitored.

For example:

When you need to monitor a new *Mac OS X* device, there are only two easy steps to do:

- add the device to atlas
- and set its device type to *Mac OS X* system.

Then node will automatically receive *Mac OS X* Monitoring Pack.

It's that simple.

See also:

[Automatic Monitoring and Organizing](#)

You don't have to setup views and maps by hand, or node by node. NetCrunch takes care of these tasks

Events & Alerts

Alerts are important part of monitoring program. It's one of the very basic use cases for monitoring. NetCrunch allows advanced alert processing including correlation, conditional events, conditional actions and escalation.

Just to make things clear: events just happen - whenever we watch them or not. "Event" becomes an "Alert" as we assign some reaction to it.

The simplest (*default*) action is storing information about the event in the NetCrunch Event Log. We can assign a different list of actions to each event. The actions can include a notification (email, SMS texting) or some corrective actions, like executing scripts or programs (also on a remote machine). Actions are executed after alert starts and when it's closed (finished).

External Events

As a monitoring program, NetCrunch is a primary source of many events such as: status events and alerts on performance metrics (counters). The program is also able to monitor external events. It matches incoming events with rules and triggers alerting actions for them. This allows you to trigger alerts and actions on SNMP traps, syslog messages, text logs or Windows Event Log entries.

Pending Alerts

As many alerts are short lived and they can be self corrected (like connection or power loss), administrators should concentrate on existing problems instead of constantly looking into log. NetCrunch simplifies alert management by correlating all internal alerts, so if they are closed, they disappear from the Pending Alerts View.

The program allows for correlating external events (SNMP Traps, syslog, etc) by defining list of closing events for each external alert.

Conditional Alerts

Simple alerts work when alerting condition is met, such as "node is down" or when some external notification has been received.

What about something that did not happen or does not happen regularly? This can be solved with conditional alerts, which allow more complex scenarios such as: notifying when syslog message was not received or notifying only if the event happened in a specified time range.

Available conditions:

- On event
- if event happened after (x) of time
- if event happened more than (x) of time
- Only if within a time range
- Only if not within a time range
- If event has not happened in a given time range
- if event did not happen after (x) of time
- if event is pending for more than (x)

Correlation

Advanced correlation allows you also to trigger events only if multiple events (from different nodes) have happened within a given time range, or all of them are pending at the same time. Pending event correlation requires all correlated alerts to be in a pending state. This easily allows you to define an alert when two redundant interfaces are down.

Alert Actions and Escalation

As the response to an event, NetCrunch can execute a sequence of actions. Actions can be executed immediately or with a delay (if the alert is not clear) and last action can be repeated. For example, you can decide to send a notification to a particular person and then, after some time, to execute a server restart operation.

See: [Alerting Actions](#)

Conditional Actions

Each action can be limited to run only if a triggering network node belongs to a given atlas view (these can be created by rules or manually) or within a given time range. This allows you to create flexible alerting scripts: for example, sending different notifications depending on the node location. Alerting scripts can be used for multiple alerts, so you can limit actions to be executed only if an alert has a given severity.

Preventing False Alerts

NetCrunch uses various technologies to avoid false alerts or protect against alert floods which might be caused by a device malfunction. When a device sends Syslog or SNMP traps to NetCrunch, the program waits for several seconds and if the same message appears several times, it won't trigger multiple alerts. Another technique (event suppression) is used for detecting false events caused by intermediate connection failures.

NetCrunch Tools

iTools

iTools is a set of network monitoring tools that allows testing availability of devices, network services on a host, scanning ports, checking the routes of test packets or connection bandwidth.

WMI Tools

A tool that allows viewing information based on WMI data - it also allows creating your own WQL views and performing WMI commands.

Performance Trend Analyzer

A tool for accessing NetCrunch performance data. You can analyze trends charts and data distributions for a given time period. You can compare multiple parameters on a single chart.

SNMP MIB Compiler

This program allows compiling MIB files in order to extend NetCrunch MIB library.

SNMP View Editor

Manage custom SNMP Views for specific device types in order to make displaying and setting SNMP data much easier.

Device Types Editor

Manages Device Type definitions needed to automatically set Device Type for SNMP devices.

Report Viewer

Allows for viewing and managing various NetCrunch reports.

System Requirements

Basic requirements are: 64 bit Windows Server, 4 processors and 4 GB of RAM - SSD driver is also welcome. NetCrunch is designed for running efficiently on both virtual and physical server machines.

Hardware Requirements

Server

NetCrunch must be installed on a 64 bit Windows Server (*Windows 2008 R2, Windows 2012, Windows 2012R2, Windows 2016*). It comes with its own Web Server and embedded SQL database for storing monitoring event data.

NetCrunch can be installed on a virtual machine, provided you assign at least 4 processors and 4GB RAM.

More processors are better for monitoring 1000+ nodes; the recommended number in such case is at least 8.

Monitoring large number of performance metrics (100,000 network interfaces) requires additional RAM (500,000 performance metrics will require additional 4GB).

The other important component is the hard drive. We strongly recommend to use SSD drives.

The [Architecture and Concepts](#) section explains why this is so important.

Administration Console

NetCruch Console runs on 32 bit or 64 bit Windows 7, 8.1 or newer systems with at least 2GB of RAM. It requires 24-bit color depth and high resolution. It's recommended to run it on Full HD screens or with multiple monitors. The console also works great with touch screen Windows tablets (Windows 8 or later).

Web Access

Web console works best with new HTML5-enabled browsers. Its compatible with IE10+, Safari, Chrome, Opera and Firefox. However, It works best with WebKit based browsers: Safari, Chrome and Opera.

Warning

Some features (part of HTML5) are not supported or are wrongly implemented in some browsers. That's why, for example, dashboard zooming works only in WebKit browsers and Edge but not in Firefox.

No Antivirus on Server

Interference with NetCrunch

NetCrunch keeps part of the data in memory, while some is written to trend log files (NetCrunch opens thousands of them) and other data goes to a SQL database. A problem can occur when NetCrunch writes files containing a snapshot of in-memory data and antivirus software tries to access the file at the same time. This causes high disk and processor utilization and sometimes may cause data to be inaccessible by NetCrunch.

Antivirus software is for workstations.

Software viruses spread mostly through email, browsers and Word documents. Servers are protected by firewalls and are suitable for jobs other than browsing the Web. That's why Windows makes it almost impossible to use the browser on a server. Avoid working on a server - because if you do, you need Antivirus software on it.

There is no need to run NetCrunch Console on a server. You should install the NetCrunch Administration Console on a workstation and control the server remotely.

If you have to...

We know that sometimes you can't change the company policy. In this case, you have to exclude all data directories where NetCrunch writes data. If it works, that's great.

In case of strange problems we can't support such installation. Our experience shows that servers behave very strangely and uniquely depending on the antivirus vendor. From the server standpoint, antivirus software is actually kind of a virus - because it hooks deeply into the operating system.

Please also note that disabling antivirus software works a bit similar to disabling stability control in your vehicle - it's only partially off. The hooks installed in system are still in place and change the system behavior in an unpredictable way. The other problem is that antivirus software might sometimes cause 100% of the single processor utilization.

No Server Sharing

NetCrunch can heavily utilize a server machine depending on the amount of processed data. Avoid

competing for resources with other programs. In fact, NetCrunch contains many servers, such as database, monitoring, web server etc. so we are already putting a heavy load on a single machine.

In some cases NetCrunch is not able to process all data (events) due to hardware limitations. Remember that the speed of a single machine is limited by its slowest components (it is usually a hard drive).

Use Appropriate VM resources

Assign appropriate amount of processing time to the machine. Memory must also be physically available as disk swapping should not occur. You have to reserve at least 4 cores and 4 GB.

NetCrunch is a real-time Network Monitoring System.

What can I monitor?

NetCrunch can monitor nearly anything: devices, applications, systems, files. The program can be easily extended using scripts; data from various sources can be sent to NetCrunch.

There are many different scenarios for NetCrunch. In general, NetCrunch retrieves and processes three types of data:

Events

Information about some occurrences generated by NetCrunch or external sources such as: Windows Event Log, SYSLOG or SNMP trap.

Performance Counters (Metrics)

Numerical values (64 bit integer or floating point)

Status

The status of various objects in NetCrunch such as nodes, services, monitoring engines, etc.

The server allows you to set various conditions to filter incoming events or set alerts on performance counters. You can even create new calculated counters. See the [Managing Calculated Performance Counters](#) topic for details.

Network Infrastructure (SNMP)

NetCrunch can be used just for network monitoring, where we mainly pay attention to SNMP devices, such as printers, switches, routers, cameras and other. NetCrunch supports SNMP v1/v2c/v3, including encryption and authentication.

Connectivity & Response Monitoring

NetCrunch monitors the availability of over 65 predefined TCP/UDP network services, including DNS, FTP, HTTP, POP3, SMTP, and more.

The program can monitor network service performance measured by number of packets sent and packets received, response times, and the percentage of packets lost and received.

For each monitored service, the program checks connectivity, validates service response and measures response time. For each sensor the program allows monitoring various conditions (for example: text contains some pattern, a file exists, and so on) and performance metrics (such as: response time or data size).

You can create custom service definitions or duplicate existing definition and change its port. Services support TCP, UDP and SSL connections. Response patterns can be defined as text, binary data or by regular expressions.

Node Up/Down Status

NetCrunch determines node up/down status based upon network service status and other monitors (in case of servers). When a node is down, only the leading service is being monitored. So, a node is considered "down" when no services are responding, and is considered "up" when the leading service responds.

DNS Health Monitoring

DNS is the most important service in a network. Without it, nothing works at all. Therefore, monitoring the DNS service in order to check its availability is an obvious task for a monitoring system. But availability monitoring only checks if the service is responding and what's its response time.

On top of simple availability monitoring, NetCrunch allows you to verify DNS responses to given queries, which in turn allows you to discover unexpected (unauthorized) DNS changes.

Switch and Router Monitoring

NetCrunch supports various aspects of switch and router monitoring, including the status of network interfaces and bandwidth monitoring. It allows monitoring traffic on interfaces, port mapping and creating Layer 2 graphical maps.

NetCrunch allows you to monitor Cisco IP SLA operations. It tracks the status of operations and also performance parameters. This allows you to monitor VOIP jitter and other protocols and parameters.

Vendor Specific Monitoring via SNMP

SNMP is very ubiquitous, but implementation varies. NetCrunch contains MIB compiler which allows you to add vendor specific MIBs.

Since basic MIB's have only been partially defined in RFC documents, vendor MIBs are sometimes difficult to compile. If you have no experience with compiling MIBs and find it difficult, please ask AdRem support for help. We will try to help you and if the device is popular on the market, we can add it to the set of pre-compiled MIBs. Please note that, NetCrunch built-in database contains more than 3500 vendor MIBs already.

Servers and Applications

NetCrunch supports agentless monitoring of the major operating systems including: Windows, Mac OS X, Linux, BSD and VMWare ESX/i. Additionally, the Windows system supports the monitoring of applications by monitoring their performance parameters and service status.

You can also use SNMP for monitoring these systems, but please be advised that using SNMPv2 can create a security loophole in operating systems, as SNMPv2 transmits data in plain text.

Windows Monitoring

Performance Counters

NetCrunch allows you to remotely monitor all Windows performance counters, including disk counters. The list of available counters depends on the particular system and applications installed. You can set alert triggers on counters using 8 different trigger types.

[Event Triggers for Counters](#)

Windows Services

Monitoring Windows services is important for monitoring most applications installed on Windows Server. The most frequent alert set on services is *Service is not Running*. NetCrunch also offers a view of services in the node status window, allowing remote service control.

Windows Event Log

NetCrunch can remotely gather, filter and analyze data from multiple Windows machines using WMI.

The program allows you to define simple alert filters to convert event log events into NetCrunch alerts. These filters are automatically converted into complex WQL queries.

Hardware and Software Inventory

NetCrunch can collect hardware and software inventory information from Windows computers. The program shows detailed information about each machine and also displays a list of installed fixes. NetCrunch allows you to compare each audit and can show changes in hardware and software. The program includes a software summary view for multiple nodes.

Monitoring Files, Folders and Text Logs

These type of monitoring is available for Linux (and other Unix family systems) and Windows. These file sensors allow you to monitor file presence, its size or if and when it was modified. It can be also used to search file contents, or for finding new text log entries and converting them into NetCrunch alerts.

The Folder sensor allows you to watch specific folder contents, for example, when a new file is added or if any files are removed.

Linux, Mac OS X and BSD

NetCrunch can track over 100 performance counters to determine the health of Linux servers running kernel 2.4 or newer. The program has been tested with monitoring the following Linux distributions: Cent OS, RedHat, Fedora, Novell OES, Ubuntu Desktop and Server.

NetCrunch also offers fully integrated Mac OS X monitoring. All Mac OS X versions are supported, including "EI Capitan".

Most Important parameters being monitored:

- System (uptime/downtime, logged in users)
- Processor utilization
- Memory usage
- Disk usage
- Network interface statistics
- Processes (CPU & memory per process utilization)

- User (CPU & memory)
- TCP statistics

Monitoring ESXi

NetCrunch supports ESX/i version 5.5 and 6. It connects directly to the ESX servers, so it does not need vSphere to be installed. NetCrunch comes with pre-configured Automatic Monitoring Packs to monitor ESX as soon as the device type is set to ESX.

Web Monitoring

Monitoring Web Pages or Applications

NetCrunch includes an advanced Web Page monitor which is able to load and render dynamic web pages containing Javascript as if they were loaded by a browser. It also allows you to check pages requiring login (supporting standard HTML or custom login forms).

Available Web Page alerts:

- page size or load time
- page content change
- alert if text is present or missing
- if a page does not exist
- page load error
- page resource load error
- page authentication error

Available performance metrics:

- % Availability
- HTTP Status Code
- JS Errors
- Load Time
- Main Frame Body Size
- Resource Count
- Resources Error Count
- Total Size

See also:

[Operating Systems Monitoring](#)

Read about monitoring of Windows, Mac OS X, Linux, BSD, Solaris and ESXi systems.

[Custom Monitoring with NC Open Monitor](#)

Read how to extend monitoring capabilities of NetCrunch. Run programs or scripts or send data from

Monitoring HTTP Requests

This sensor is more suited to send REST requests, so it simply retrieves data over HTTP and checks the response. It also allows you to check response content. It supports GET, HEAD and POST requests,

Custom Monitoring

NetCrunch can schedule programs and script to run in the same security context as the NetCrunch Server. The program or script can write its output to a standard output and NetCrunch can capture it automatically. The monitoring program can also save data to disk. The program supports CSV, JSON and XML file formats.

NetCrunch Open Monitor

NetCrunch utilizes *NetCrunch Open Monitoring Engine*, which receives data by REST API or can get them from scripts, programs or files.

If you can present some external data as counters, then you can deliver them to NetCrunch and set various thresholds on them to create alerts.

Open monitor data are not attached to any node.

Generic Agent Sensor

Allows to define a sensor on the node to receive data from an external source (device, script, app). Data can be sent using REST API and you can set alerts on received metrics and status objects.

See also:

[Custom Monitoring with NC Open Monitor](#)

Read how to extend monitoring capabilities of NetCrunch. Run programs or scripts or send data from

Automatic Monitoring and Organizing

You don't have to setup views and maps by hand, or node by node. NetCrunch takes care of these tasks automatically.

Monitoring by Device Types

The fundamental concern in the NetCrunch configuration process is the setting of proper Device Types for nodes. Whenever possible, NetCrunch sets a Device Type automatically by getting information from SNMP or Active Directory. If it can't, the Device Type must be set manually. This is very simple and you can do this

for multiple nodes at once. See: [Managing Multiple Node Settings](#).

Monitoring Engines

Many monitoring engines are bound to types of nodes and start monitoring them after you set the Device Type. Here is the list of engines with required Device Type settings:

Windows

Device Class : *Server/Workstation*

Operating System: *Windows Server* or *Windows Workstation*

OS Version: *<any>*

Linux

Device Class: *Server/Workstation*

Operating System: *Linux*

OS Version: *Generic Linux System*

Mac OS X

Device Class: *Server/Workstation*

Operating System: *Mac OS X*

OS Version: *<any from the list>*

BSD

Device Class: *Server/Workstation*

Operating System: *BSD Family*

OS Version: *FreeBSD* or *OpenBSD* or *NetBSD*

ESX/i

Only ESX/i version 5 is supported by the engine.

Device Class: *Server/Workstation*

Operating System: *VMWare ESX/i*

OS Version: *ESX/i 5.5, 6.0*

SNMP

[Node Settings](#) ▶ [Monitoring](#)

SNMP monitoring depends on the profile you must select for each node. The profile specifies version of the protocol and protocol specific settings like community or username and password (v3).

If you want to receive SNMP v3 traps you have to create SNMP notification profiles in order to provide authentication and encryption parameters. The profiles are matched with trap by *User* field. SNMP v1 and v2 traps does not need any profiles. All incoming traps are visible in External Events window which

simplifies defining alert for traps.

Monitoring Packs

NetCrunch comes with many automatic Monitoring Packs configured to be added to nodes when they meet certain conditions.

For example:

Basic Cisco (SNMP) Monitoring Pack

```
Device Class should be "Hardware Router" or Switch  
Manufacturer name contains "Cisco"
```

Active Directory Monitoring Pack

```
Operating System equals "Windows Server"  
Network Service List contains LDAP or "Secure LDAP" services.
```

You can view the complete list in the [Monitoring Packs](#) article.

Auto Discovery

Networks are dynamic and new devices are connected over time. NetCrunch can automatically add them to the Network Atlas and start monitoring them.

Discovering Nodes

NetCrunch can run auto discovery process for each IP network and Active Directory container. All discovered nodes can be added automatically or program can put results in a *Server Tasks Notification Window*, so you can decide later which nodes should be added.

Discovering Services and Device Types

When node is added to atlas, the program automatically runs service discovery. It's only checking list of services set in [Tools ▶ Options ▶ Monitoring ▶ Auto Discovered Services](#)

The program also tries to detect device type depending on available information from SNMP or Active Directory. If the device type can't be detected, it should be set manually. Device type is an important information allowing proper monitoring settings for the device.

Program also discovers ESX/i machines.

See: [Auto Discovery](#)

Automatic Node and Views Status Calculation

Node Status

NetCrunch automatically identifies the status of every of network service working on the node, and at least one service must be monitored. If any of the services is not in *OK* (green) mode, then the node's status goes to *Warning* state (*yellow*). When no service is responding or all monitoring engines are *DOWN*, *then the node becomes DOWN* (*red*).

When node is in *DOWN* state, only service marked as *leading* is monitored. If the service responds than all other monitoring starts again.

Atlas View Status

Status for the view is calculated upon all node statuses included in the given View. If there is any node in *Warning* or *DOWN* state, then the View Status is *Warning*.

When all nodes are *DOWN*, then the View is *DOWN* and when all nodes are *OK* the view is *OK*.

Additionally, static views can contain map links to other views that are also calculated in the same way as the node statuses.

See also:

[Network Services Monitoring](#)

Automatic Atlas Views

Based on typical Atlases used by our customers, we prepared many automatic views in NetCrunch. You can delete or edit them if they don't meet your data organization needs. Grouping nodes makes creating reports and watching the status of a given group easy.

The status of each group is automatically calculated based upon the included elements (nodes or map links).

Dynamic Views

The dynamic view presents a group of nodes upon giving filtering criteria (query):

Predefined Dynamic Views:

- Nodes with Issues
- Unresponding Nodes

Dynamic Folders

In addition to views, there is another level of grouping: The Dynamic Folder. It creates Dynamic Views automatically.

For example, we can easily create the following hierarchy:

- Folder for each location
- Each view shows servers in a location grouped by OS types

List of predefined and automatic folders:

- Server Types (i.e. Linux, Windows Server, etc.)
- Device Groups (Printers, Switches, Wireless, etc.)
- Workstation Types (Windows 7, Windows Vista, Windows 8, etc.)
- Locations (Office, Building 1, Server Room - based on SNMP or manually entered data)

- Network Roles (Network, Printers, Servers, Workstations)
- Windows Domains

The views are dynamic, which means that they are automatically updated as needed.

Auto Layout Maps

Each Dynamic View can be presented as a map with node icons. This allows for grouping nodes visually and presenting their status graphically. Because maps update dynamically, they are also laid out automatically.

You can change any map layout or create new ones.

Accessing NetCrunch...

[Read how you can access NetCrunch remotely and how you can send data to NetCrunch.](#)

From the Desktop

NetCrunch Administration Console has been built for the desktop and it runs best on larger monitors or with multiple monitors. NetCrunch can easily manage even six monitors on a single machine.

The program caches a lot of data on the disk and only synchronizes changes with the server. Also, it gives the best user experience by responding to actions fast and showing data instantly. No refresh is needed, as the console is updating data in almost real time.

The console can be installed on any Windows system, newer than Windows 8, but we recommend using the latest system versions (Windows 10). NetCrunch also supports touch screens, so it works great on devices such as *Microsoft Surface Pro*.

From a Web Browser

The Web console is also designed for a desktop browser and matches the console UI very closely. It lacks some configuration and administration features, but allows you to browse through Atlas Views and Event Log. You can also access reports. The web access is great for quick checking a network of status from any place.

Additionally, it allows you to restrict access to views and features for a given user or group. For example, you can create a limited access to a view and pass it to high-level managers or low-level administrators.

The web console supports HTML5 capable browsers like IE10 or newer, Safari, Chrome, Firefox and Opera. However, it's been optimized for WebKit-based browsers and thus we recommend Chrome, Safari or Opera.

On the Go

The Mobile Access console is an HTML5 JS application accessible through a Web Browser from any modern Smartphone (IOS, Android and Windows 10). It gives you access to atlas status and the latest alerts.

Sending to NetCrunch

NetCrunch can receive external performance data, object statuses and events (it can be also a status value) through HTTP. These can be sent from external agents (scripts).

See also:

[Sending Data to NetCrunch](#)

Read how to send data to NetCrunch and create a custom monitor. You can easily turn any application or script into a NetCrunch agent.

[External Event Sources](#)

Extending and Customizing

Add more MIBs, counters, create views, icons, notification message formats, extend the node database and create alerting action scripts.

NetCrunch comes with a number of predefined resources, like pre-compiled MIBs, Monitoring Packs, icons, etc. In the real environment, however, this might not be enough, so we let you extend these resources accordingly.

Extending

SNMP MIBs

Now you can find any MIB for your device on the internet. There are sites keeping a collection of more than 65 thousand of MIB definitions.

NetCrunch has its own MIB compiler, so you can compile any MIB. Because many MIBs contain bugs, they are not very easy to compile.

If you have any problem with compiling MIBs, please let us know and we will try to compile MIBs for you.

SNMP Views

NetCrunch contains definitions of forms and tables displayed upon SNMP data from a device. There are groups of specific, detailed views for certain devices like printers, switches or anything else.

Using these forms you can change SNMP variables. You can also create your own view forms and tables using *SNMP ViewEditor*.

Additional Node Fields

Network nodes in NetCrunch are kept in an in-memory database and stored in XML files.

You can extend node data by adding additional fields. This will allow classifying and grouping network data in a way which suits your needs. This allows creating dynamic views based on these additional fields and

also allows you to manage alerting action execution using such views.

Example:

Problem:

You want to notify different group of Administrators (using groups is always more flexible than single user) depending on department and location where the server is located.

Solution:

Add custom field for department and set the location. Setup dynamic node views based on the department and location fields. In Alert action list you can now define different notifications for each defined view.

Calculated Counters (aka Virtual Counters)

Sometimes the device returns data which needs to be calculated before processing. For instance, the data needs to be divided or we want to have percentages instead of raw values. At this moment you can add a new Calculated Counter, which is calculated upon the given expression. See [Managing Calculated Performance Counters](#).

Custom Monitoring

You can extend NetCrunch monitoring using external scripts or programs. They can be run on NetCrunch servers or you can send data from any location using REST API.

Examples:

- If you want to monitor your coffee maker and you know how to get data from it.
- Web application or web site can easily send the number of visits per hour, so you can track the statistic on charts and get notification when thresholds are breached.
- Create a script or a program to get data from SQL database and save it in the JSON format for NetCrunch.

You can do it easily with [Custom Monitoring with NC Open Monitor](#) or [Sending Data to NetCrunch](#)

Customizing

Custom Views & Network Maps

In NetCrunch you can create your own views of node groups that make managing alerts and reports easier. You can create your own network maps with customized icons and backgrounds. For each node group view, there are separate tables showing data which is specific to a given *Monitoring Engine*. This can be changed on demand or you can create custom layouts. Every time you get back to that view, the data will be grouped and sorted in the way you wanted.

Customizing of Administration Console Screen Layout

The Administration Console can be very flexible: it can run on multi-monitor systems and you can create custom multi-monitor layout.

As a second option you can divide space on a large monitor and dock several windows to be visible. The layout is automatically stored so when you run console next time, NetCrunch will bring all windows to the same place.

Customizing Notification Messages

Most events contain a large number of details describing the event context. You would not be able to see all of them in the text message on your smartphone or even in the email. This is why you can create message formats suitable for notification types and specific alert types. Simply include the information you need. This way the program automatically creates HTML emails (you can also customize them) when sending an email and it uses plain text for SMS/text messages.

Integrating NetCrunch with other NMS

You can easily integrate NetCrunch with existing management systems to extend its capabilities.

It might happen that you already have had some management system and you want only to use NetCrunch as the extension of the existing system.

Integration by SNMP traps

if your system uses SNMP the best way would be use SNMP. In this way you can monitor and collect data in NetCrunch and send alerts using SNMP traps to the external system. NetCrunch can automatically define SNMP traps for each alert defined, so after reading NetCrunch generated MIB, you can successfully receive these traps in external system.

Integration by Event files

➤ Alert Escalation Scripts ▶ Add Alerting Script ▶ Add ▶ Action to run immediately (or after some time) ▶ Action ▶ Logging ▶ Write to Unique File

This is most effective method of transferring events from NetCrunch to external program. You can use alerting action which can write each alert to the separate, uniquely named file.

The action allows to use (write) some program periodically scanning the folder and importing event files to external system. In this way the disk is used as the queue for alerts. Each event can be stored in XML format.

Webhook Action

➤ Alert Escalation Scripts ▶ Add Alerting Script ▶ Add ▶ Action to run immediately (or after some time) ▶ Action ▶ Logging ▶ Trigger Webhook

Webhook is very simple action that sends an HTTP request to given URL with all event data as JSON or XML.

Integration by External Program

Unlike previous methods, this method should be used only for small number of events, as it requires running the process for each transferred alert, which can be quite slow.

Exporting NetCrunch SNMP MIB with traps

➤ [File ▶ Export NetCrunch SNMP MIB](#)

After you define all alerts you might decide to generate NetCrunch MIB, which will contain SNMP trap definitions for all NetCrunch defined alerts.

Exporting Performance Trends Data

You can also periodically export performance data from NetCrunch and import to the other systems. See [@trend-exporter](#).

Active Directory Integration

It makes agentless windows monitoring much easier

Server Integration

NetCrunch runs on Windows Server and needs to integrate with Active Directory in order to properly access other Windows machines in the domain.

Starting from Windows 2003, every newer Windows Server version takes security settings to a higher level. This makes it impossible to access Windows machines without explicitly setting access rights.

When installing on a Server in Active Directory

- The default option is to run program on the local system account (which makes easy accessing local resources and communication between server components) and setup default credentials or credentials for each machine (it's easy as you can use multi-selection to do it). The only drawback is that it might cause a security warning on some server systems, as the process running on the local system account attaches first as a machine object (machines are separate objects in AD) and then it logins with given credentials.
- You can run NetCrunch Server services on a domain account being a member of a local Administrator group of each monitored computer (including server running NetCrunch Server). From the security standpoint this is the best solution, but sometimes hard or impossible to configure. It requires modifying AD security policies (remember they need time to replicate).

Setting Active Directory account for the NetCrunch will give you:

- Proper security settings to remotely access performance data.
- Access to AD information about computers and their systems.

You will still be able to access other Windows machines by giving local credentials for them, but it means that you have to input the necessary settings for each of them individually

See also:

User Accounts Integration

NetCrunch can use Active Directory user accounts as NetCrunch users. This allows keeping single passwords and makes management easier.

All you need to do is tick the *Active Directory User* checkbox when adding a new user. The user name should be in format: <Domain>\.

NetCrunch Editions

Quickly review differences between NetCrunch editions.

Premium

Premium edition of NetCrunch is designed for smaller, less intensively monitored networks up to 300 nodes.

Premium XE

The Premium XE edition is a scalable version optimized for larger or more intensively monitored networks and is intended to be run on a dedicated machine.

Features Exclusive to XE edition:

Prioritized Monitoring

The program automatically sets up node monitoring order and time upon monitoring dependencies hierarchy. It means that nodes which are more important, are monitored more frequently. Prioritized monitoring is also designated to support event suppression technology and helps to determine the real source of the problem.

False Alert Detection

When NetCrunch receives an event related to a node connected through some intermediate link, it ensures first if the link is OK, so the event might be a result of that the connection has been broken.

IP SLA Sensor

Allows monitoring of status and parameters of IP SLA operations defined on Cisco devices.

Integrations

Allows two way integration with external Help Desk, productivity and messaging systems. This module includes integration with many popular services like: Slack, ZenDesk, FreshDesk, Jira, Twitter, Pagerduty, Connectwise and many more.

Text Log Sensor

It allows you to easily parse text logs to be alerted on error messages. You can define own

parsing expressions or use defined in program.

External Alerts

All Incoming traps, web messages and syslog messages (even from nodes not being monitored in Atlas) are visible in the External Events window which shows last 1000 traps or messages received. This allows for a quick setup of syslog or trap alerts *by example*.

Alert Correlation

Advanced correlation allows also you to trigger events only if multiple events have happened within a given time range, or are pending at the same time. For example, this allows you to define an alert when two redundant interfaces are down.

Alerting Conditions

Alert on if certain conditions are met, such as: when event did not happen, or happened only after some time. Allows monitoring heartbeat events or missing backups. Additional 7 Conditions makes program definitely more flexible.

VLAN Support

Supports VLANs in port mapping and Layer 2 maps.

Support for STP, CDP, SONMP

Program supports additional protocols in order to create accurate automatic Layer 2 maps.

Built for scalability

Version initiates up to 100 connections at once which is 2 times more than in Premium edition.

Important Changes in Version 9

This is short list of important or breaking changes in version 9

List of new features is listed on [What's New Page](#) and you can find more detailed list in [Release Notes](#)

Version 9.3

This version automatically imports old access rights from version 9.1 or older.

Version 9.2

Console Login Support

Now all users must login to the console and need Server Administration right in order to access it. Accounts have been imported from previous version of Web Access users. Administrator must login for the first time to the Administration Console on NetCrunch Server with empty password, which should and must be changed immediately.

Access Rights

Access rights has been completely redesigned and simplified. The import of access rights has been implemented in version 9.3.

Auto Close Local Console

Console run on NetCrunch server using remote access is closed automatically 3 minutes after user disconnects the console.

Version 9.0

New version comes with a lot of new features and many changes made in order to make easier to use. Over 450 change requests were processed until initial release.

Changes

Pending Alerts

All alerts can have pending state, even the minor and informational ones. New filter in Pending Alerts Windows allows showing: Critical and Warnings, All except Minor, All

Linux Memory Counter calculation

%Free memory on Linux is now taken from other system counter. The old %Free memory trends were incorrect.

SNMP Traps

Alerts

Defining alert is simplified. You can select generic alerts or alert by OID (entering OID manually or from a MIB browser). Additionally, after selecting OID you can filter traps by their variables.

SNMP v1, v2 Traps

It's not needed to enable SNMP monitoring on node sending the trap. Currently traps will be received regardless of the community set in the trap. Version XE allows for receiving traps from nodes not being monitored (you can add them to Atlas when you want to turn traps into alerts).

SNMPv3 Notifications

In v3 notifications are identified by the User and authentication profile. In order to receive v3 notifications, you have to define *Notification Profiles* for each *User* used to send the notification. Program will find the profile to decode profile automatically. Single profile can be used for multiple nodes. Also in this case, you do not have to enable SNMP monitoring on the node from which the notification comes.

Data Folder Location

Directory of the NetCrunch Data does not contain version number anymore. We want to keep data in the same place in future versions.

Monitoring of PING

PING monitoring has been delegated to a system component for more accurate timing. However, the old method is still available through options setting. In [Tools ▶ Options ▶ Monitoring ▶ Advanced](#) you can find the option named Use legacy ICMP PING. (This option has been completely discontinued starting from version 9.3)

Terminology

Virtual Counters are now *Calculated Counters* as we believe that the new name better reflects their role.

Run MIB Compiler from Remote Console

MIB Compiler now runs remotely, previously was accessible on NetCrunch Server.

Maps

Map Background can no longer be image. Use [✈ Insert ▶ Picture](#) instead.

Important Bug Fixes

- Fixed various issues in map editor
- Wrong calculation of % Failure Rate could cause superfluous *Connection Reliability Degradation* alerts
- Flow protocols decoding has been improved
- (Performance Trend Viewer) Isolated points where not visible on charts
- Fixed various issues in browsing MIB data when selecting counters and traps
- Gaps in report graphic
- Server should open ports on firewall in order to receive SNMP traps and syslog
- Custom data range was missing in event log

Configuration

Everything you should know about configuring NetCrunch, including Initial configuration, alerts and reports.

Before You Begin

Configuration of NetCrunch can be easy or hard - depending how you start. There are many configuration possibilities, and if you know which one to choose in a given situation - everything becomes simple.

Read [Architecture and Concepts](#) to quickly review the options.

Important task you need to perform first: [Windows Monitoring Setup](#).

Discovering your Network

This article will help you get through network discovery and the initial configuration process.

In order to monitor your network, NetCrunch must know device addresses, their names and how to connect to them (credentials, types, etc.). The Network Atlas is a database holding all that information, so the first step will be adding nodes to your Network Atlas.

Manually

You can add nodes from a file, then you can do the rest of the setup manually. It's even possible to create an empty Atlas and add every node manually, but it's unlikely you would need that feature.

Nodes can be added from the text file which can contain either names or addresses of nodes - one per line.

By Network Discovery Wizard

Discovery Methods

Search Active Directory Domain

The program searches Active Directory and adds all devices, including other operating systems like Mac OS X machines.

The program scans only the Domain, where the NetCrunch Server machine object is located.

All machines from Active Directory will be added even if they are currently not connected to the network.

Discover IP Networks

The program will scan a given range of network addresses using ICMP (Ping) packets. Only connected and responding nodes can be discovered in this way, so results may vary depending on that when you perform

the discovery.

Networks to Be Discovered

NetCrunch fills the list with known networks, you can add more networks as desired.

The program uses ICMP PING packets to discover nodes in a given network, and only nodes responding to ICMP PING packets are added.

Discover and scan neighborhood networks option

During the discovery process, NetCrunch finds connections to neighborhood networks. Enable this option if you want NetCrunch to follow these links automatically. You can limit the scanning depth by specifying the number of maximum hops to remote networks.

Discovery Mode

All Devices

The program will try to find and add as many devices as it can find. It will look into SNMP and use PING sweep.

Infrastructure Devices Only

The program will skip workstations.

Only devices matching SNMP filter

Build filter query which will include desired devices. The program will try to communicate with any given SNMP profiles and will find out which profile needs to be used for the device.

Configure SNMP Profiles

This is the point where you should enter all the SNMP profiles used by your SNMP devices.

If you fail to do this now, you can enter them later, but you will have to assign them one by one to each device.

If you enter the SNMP profiles now, the program will automatically determine which profile is used by which device, and devices will be marked as SNMP manageable. Additionally, device types should be discovered (by sysObjectId SNMP variable or device SNMP name) and automatic monitoring will be set.

SNMP Agents Port

SNMP Discovery assumes that you use only one SNMP port for all devices in your network. By default, SNMP agents respond on a port 161. If you use different ports for different devices you have to setup them manually.

Select Network Services To Be Discovered

NetCrunch recognizes and can monitor about 65 network services. Discovering all of them might be time consuming and might generate unnecessary network traffic. As a result, we decided to include 14 common services by default, and you can add more services as desired.

Here you can also set the initial parameters for each monitored network service.

Repeat Count

The number of requests to be sent in one monitoring pole to properly determine average response time

Additional Repeat Count

The number of additional requests to be sent in case base requests fail

Timeout

Time after which NetCrunch gives up on waiting for a response

You can decrease timeouts if you monitor network devices in a network with low latency. Default values should work great, even for internet connections.

Next Step...

After NetCrunch discovers all your devices, it starts discovering network services on each device in the background.

This is a time for further configuration which includes: automatic Monitoring Packs, setting credentials, setting NetCrunch users and administrator profiles and default alerting scripts.

Go to: [Configuration Wizard](#)

Configuration Wizard

Configure automatic Monitoring Packs, set credentials for different systems, setup NetCrunch users and Administrator profile. Customize default alerting script.

Automatic Monitoring Packs

There is a set of predefined Monitoring Packs that bind to nodes by their device type. Monitoring Packs contain alerting and reporting settings for these nodes.

You can manage them by groups or switch to detailed view where you can manage each Monitoring Pack separately.

Default Credentials

To be able to monitor anything, NetCrunch must connect to various systems using the proper credentials. Here you can set all default credentials (to be used if you do not set specific one for the node) for supported operating systems (Windows, Linux, BSD, Mac OS X, Solaris, ESX/i).

Note:

If NetCrunch runs in the Domain, it connects using NetCrunch Server account credentials by default.
See: [Windows Monitoring Setup](#)

NetCrunch Administrator Profile

The administrator profile is called "Admin" and can't be deleted. Also you need to set a non-empty (at least 6 characters) password.

You can specify emails and phone numbers to be used by notification actions.

NetCrunch Users and Groups

To be able to access NetCrunch remotely, you need to use NetCrunch user profiles. Additionally, profiles can be used for notification management, as you can easily change user profile notification settings instead of editing all alerting actions. You can manage profiles by groups and manage the notifications to be sent to various groups.

User notification profiles allow setting different notification types (email, SMS with message format) to be sent on different days and times

Examples:

- *Database Admins* group may receive only notification about database systems problems
- Administrator *John* can receive notifications during workdays by email and additionally on weekends by SMS

Default Alerting Script

All predefined alerts use the *Default Alerting Script*. You can edit it later in detail, but here you can make some preliminary settings

That's not all...

We've made significant progress in configuration settings, but in real life there might still be issues that need to be fixed.

We call these problems *Monitoring Issues*, because we want to differentiate monitoring configuration problems from others.

Go to: [Configuration Tips](#)

Managing Network Atlas Views

[Read about types of views and how they organize your data.](#)

Network Atlas is a central database containing all your network data. It's organized by the hierarchy of the Atlas Node Views.

The Network Atlas is a part of the [Automatic Monitoring and Organizing](#) concept. It is a central repository of all views, grouping network nodes by different categories like: nodes from the same network, node of single layer 2 segment, or nodes located within the same area.

The basic element of the Atlas is a network node: a single address network endpoint. Because many devices use multiple interfaces, they can be grouped together and then you can decide to monitor only the *primary interface*.

Atlas Views

Atlas Node View shows various aspects of the group of nodes in the Network Atlas and consists of multiple pages such as nodes, maps, dashboards and other.

The hierarchy of the Atlas Views helps you recognize status of each node group. A top level (root) view of the Atlas contains all nodes, all sub-views and Dashboards show information aggregated for all nodes.

Common Views Properties

- Alerting & Reporting - You can define alerts which will be inherited by all nodes in the view,
- Web Access Rights - For existing Web Access Profile specify rights to the view.
- Top Charts - (Custom Views Only) Specify a list of top charts for the view.
- Appearance - Enable Automatic Node Arrangement by specifying grouping layout.

Top Level Views

Monitoring Dependencies

This view shows the Monitoring Dependency diagram.

Monitoring Dependencies reflect network connections and allow for preventing false alarms and disable monitoring of unreachable network components.

Routing Map

Automatically created view; it shows a map of logical connections between IP networks. The view is updated on demand as routing is not changing often. You can update it by clicking

[↗ Atlas ▶ Recreate Routing Map](#)

Atlas Sections

Below the Top Level Views there are 4 main sections of the Atlas Views.

IP Networks

This section consists of IP network views/maps. Each network can be periodically re-scanned to reflect its current state. Usually, network maps are automatically arranged and nodes are grouped by the device model and OS name.

As each node is a part of some network, these views contain all nodes. Deleting a node from any of the IP Network View will cause deletion of the node from the Atlas. Deleting the View will delete all nodes contained in the View.

Local and Remote folders

Networks are grouped in two folders. By default, local networks are those accessible directly by NetCrunch Server's network interfaces. All others are remote. You can drag and drop networks between folders as desired.

(0.0.0.0/8) - Empty Network

Sometimes a node is specified by its name, and its address has not been resolved yet. In this case it belongs to the empty network view.

Specific Properties

- **Set Traffic Limit - IP Networks** allows setting a limit on monitoring traffic to the IP Network. This helps to preserve the router's bandwidth. Setting a limit may cause an automatic increase of monitoring time for nodes of the network.
- **Auto Discovery** - You can set (*enabled by default*) each network to be automatically scanned in order to find active nodes. The minimum re-discovery time is 1 hour. You can also specify exclusions list - a kind of a black list.
- **Discover New Nodes** - you can perform manual discovery of network nodes.

Physical Segments

This section contains a hierarchy of views showing Layer 2 connection segments. Each view is automatically arranged and can also show the traffic summary for each switch port.

To start monitoring Physical Segments you need to point the switches to be used to a network topology map. Switches must be defined in the Atlas and have properly set SNMP profiles.

After you click on [➤ Enable Physical Segments Monitoring](#), the Physical Segments Configuration Wizard will start. It will try to find switches, matching the above requirements. If you can't find the switch, but know it supports RFC 1493 MIB, please check if it's in the Atlas and its device type is set to *Switch*.

To create topology maps NetCrunch uses SNMP Forwarding Tables (RFC 1493) and additionally can utilize information from the following protocols: STP, Cisco CDP or SONMP.

Specific Properties

- **Appearance** - You can set the port name style, and style of the port box, and ports can be sorted by name or number. You can also customize how the node state is signaled, or even disable showing the node states on the topology map.
- **Port Mapping** - You can click on the segment map and open [➤ Port Mapping](#) which will open the segments' switch status displayed on the Port Mapping page. Port Mapping shows the status of all ports, including V-LAN information.

Custom Views

This section allows you to organize your network data. Here you can add your own Node Group Views and manage them using folders. You can create graphical maps for the view and create links between them. You can also create automatic folders to manage automatically created views.

Automatic Views & Folders

The primary goal of NetCrunch is to keep track of the current network state as it changes over time. It's better to set up views and configure by rules to keep network relations dynamic.

Predefined

Based on typical customer Atlases we prepared some Automatic Views (aka dynamic views) for you:

- Nodes with Issues
- Non-responding Nodes
- Server Types (i.e. Linux, Windows Server, etc.)
- Device Groups (Printers, Switches, Wireless, etc.)
- Workstation Types (Windows 7, Windows Vista, Windows 8, etc.)
- Locations (Office, Building 1, Server Room - based on SNMP or manually entered data)
- Network Roles (Network, Printers, Servers, Workstations)
- Windows Domains

All Automatic Views are dynamic, they are automatically updated as needed.

Automatic Folders

You can create a set of views that are automatically updated as data changes. For example: You want to have separate view for each city where devices are located.

You need to specify the node field for creating views, and you may also decide to create only separate views consisting of more than 5 elements. Otherwise, nodes will be placed on a single view.

As these views are automatically created and deleted you can't manage alerts and reports using them.

Specific Properties

- Node Filtering - Specify the filtering condition
- Folder Content - Specify the grouping field and manage groups and their icons
- Appearance - Set an icon for the folder

Dynamic View

The views are managed through specified filtering conditions (query). Additionally, you can specify how to arrange a node using node layouts.

Layouts allow two levels of node grouping. For example: by Device Class and SNMP Location. You can also specify the style of a grouping box.

Example:

```
Domain is equal to ad.adrem
```

Specific Properties

- Node Filtering - Specify filtering conditions

Performance Views

This type of view displays performance data for a given group of nodes. Charts can be grouped by node or by performance counter. In this case you will be able to see multiple counter values for each node in the view on a single chart. Chart tiles can be displayed as:

- Chart – linear chart showing the last 60 samples of collected data (by default it's 3 hours)
- Bar – a single bar showing the last read samples
- Gauge – an angular gauge showing the last read samples

All performance data are kept in the NetCrunch Trends Database for future analysis. You can select a single panel and choose [Show Counter History](#) from the context menu, in order to open [@trend-viewer](#).

Dynamic Chart View

The dynamic chart view allows selecting filtering condition to select nodes for the view and single performance counters for the view.

For Example:

Windows Server Processor Utilization View

Filtering Condition

Operating System is equal to Windows Server

Content\Performance Counter:

Processor(_Total)\% Processor Time

Note:

You can create new Views through drag and drop.

Select multiple nodes using the existing view (i.e. atlas grid or map) and drag and drop in a desired place in Custom Views or Performance Views.

Alert and Report Management

Read about scheduling reports, the difference between an event and alert, Monitoring Packs and message formats.

Monitoring Packs and Node Settings

Although alerting and reporting serve different purposes, their settings are very similar.

Note:

- In order to create an alert you need to specify the event condition to trigger the alert.
- In order to create a desired report, data needs to be collected first.

NetCrunch manages alerts and reports in the same place through Monitoring Packs and Node Settings.

Report Scheduling

By adding reports to certain *Monitoring Packs* or nodes you enable data collection for the given report. Each added report can be also scheduled by selecting one of the defined Report Scheduling Schemes and specifying the user or a group which should receive the report.

Read more about [Customizing NetCrunch Reports](#)

Events and Alerts – what's the difference?

Event is the description of the thing that happens or takes place, especially one of importance.

As we assign an event condition to be watched or received by the program, it turns into an alert, which also contains a log of operations taken as well as the response to the event.

Alert - the condition being watched for an action as the reaction to potential danger or to get attention.

In other words: the program is the alert guard watching for specified event conditions. When we decide to create a new alert, the default action is to write it to the NetCrunch Event Log. You can assign a common Action List to an alert or create custom sequences of actions for each alert.

Defining Events

Each Monitoring Engine defines its own set of events to watch. There is a number of predefined event conditions, especially to track well known object states like: Windows Services, Network Services, Nodes, etc.

There are many more events than defined in the software. For instance, when you monitor external syslog events you need to describe which ones you want to be NetCrunch events. If you decide to turn all syslog messages into a single event description, then you won't be able to set different alerts for different messages.

The most important types of events you can define are [Event Triggers for Counters](#) which you can be set on any performance counter value, and allow you to set logic for observed counter values.

Common Event Definitions

When you create a new event condition to set an alert, it can be saved for later use and you can add it later to another node or policy. This way both nodes (or Monitoring Packs) will share the same event condition. When you want to change it, you can modify it for one node or for all nodes sharing the same condition.

By default, new rules are saved as common definitions.

If you want to change this setting uncheck *Save as common definition* before you save a new event.

If you want to manage common definition or remove unused ones, go to:

[Monitoring](#) ▶ [Monitoring Packs & Policies](#) ▶ [Common Alerts](#)

Setting Alerts & Reports

Setting alerts using Monitoring Packs: [Monitoring](#) ▶ [Monitoring Packs & Policies](#) .

You can override or add alerts and Monitoring Packs to a node or multiple nodes by clicking on a node (or selecting multiple nodes) [Node Settings](#) ▶ [Monitoring](#)

See [Managing Multiple Node Settings](#)

Report Types

Basically, there are two main types of reports: aggregated for a group of nodes and single node reports. Both of them need data.

Data collection management is very similar to alert management. It needs to be specified for a certain node. this can be done through Monitoring Packs, Atlas Views (Maps) or you can set it directly in Node Settings window.

Monitoring Packs

Monitoring Pack is a group of performance parameters and events to be monitored and collected for the reports.

Automatic Monitoring Packs

Automatic monitoring packs specify a node filtering condition, which allows you to automatically apply the Monitoring Pack to nodes.

Most predefined Automatic Monitoring Packs bind through a specifying operating system type and some additional condition.

Example:

Active Directory is added to a node if

Operating System is *Windows Server* and,
Network Service List contains one of the following: LDAP, "Secure LDAP".

Each Automatic Monitoring Pack has an Exclusion List, which specifies nodes that should be excluded from the given condition.

Static Monitoring Packs

You can add a Static Monitoring Pack manually to a node using [Node Settings](#) ▶ [Monitoring](#) or you can open the properties of the Monitoring Pack and click on [Assigned to](#) page.

See list of predefined [Monitoring Packs](#)

Global Monitoring Packs

➤ [Monitoring](#) ▶ [Monitoring Packs & Policies](#)

In the NetCrunch Alerting & Reporting Settings window there is a special group named Global.

It contains a list of special predefined Monitoring Packs. Some of them apply to all nodes; some are Monitoring Packs that refer to globally collected data such as: NC Open Monitor or NetFlow traffic summary. When you modify Node Status and Connection Status packs, be aware that each alert will be automatically monitored for all nodes.

- Node Status - Sets monitoring alerts of node status for all nodes.
- Service Status - Alert on connection reliability degradation, PING RTT > 1000 ms, Any Service is DOWN, Any Service is UP.
- Open Monitor - Here you can set triggers on data in NC Open Monitor space. See: [Event Triggers for Counters](#).
- Global Flows - Here you can set triggers on summary counters from the NetFlow server. See: [Network Traffic Monitoring](#).
- NetCrunch - Set Alerts for adding or removing nodes from the Atlas. You can also set a NetCrunch Status Event: a kind of heartbeat event generated periodically containing NetCrunch status.
- NetCrunch Self Monitor - This monitoring pack for NetCrunch Server monitoring it contains alerts about various NetCrunch Server components. It contains alerts about NetCrunch maintenance, backup and more.
- NetCrunch Audit - tracks all NetCrunch users' logins and logouts, including failed logins to the program's desktop and web consoles.
- NetWork Traffic (SNMP) - Defines data collection for traffic monitoring: *Summary of Network Traffic, Network Traffic by Interface, Interfaces Utilization*. It's automatically applied to devices of class: *Hardware Router, Switch or Network Storage*. See: [Network Traffic Monitoring](#).
- Correlations - Here you can add alerts triggered when two or more alerts on different nodes happens at the same time. For example: alert when two connection links are down. You can add correlations using Pending Alert state or by defining time range window in which all events must be triggered.

Overriding Monitoring Pack settings

When you add Monitoring Packs to the node (or if they've been added as automatic Monitoring Packs), the node settings become a sum of settings from multiple packs applied to the node.

You may override the settings for a specific node. Select a node (or multiple nodes) and open

➤ [Node Settings](#) ▶ [Monitoring](#)

and click on desired Monitoring Pack, then you will be able to disable or override alert actions defined by given *Monitoring Pack*. Automatic Monitoring Pack can be disabled on particular node.

Alerting Actions

Actions are executed as a reaction to an alert. Actions are always grouped in Action List sequence.

See [Alerting Actions](#)

Action List (aka Escalation Scripts)

The action list is the sequence of actions executed as a response to the alert. It's grouped according to the execution delay time.

Escalation

Some actions may be executed immediately, and others may wait several minutes to start. The last action in list can be repeated until alert is closed (issue is resolved). You can also define list of actions executed when alert is closed. Each action can have restrictions allowing to execute it only in certain conditions like: alerts in certain time range, for node only being member of given Atlas View or alert has to have certain severity.

Managing Message Formats

Event descriptions are very different. There are several fields common to each NetCrunch event, but most of the data comes from various external sources like syslog, SNMP traps, Windows Event Log or different Monitoring Engines.

It is hard to define a single message format for each event and notification target. It's rather obvious that sometimes you might expect to receive an HTML email full of content and other times a short SMS with only the most important info identifying the problem.

Another application for Message Formats is passing parameters to various external actions like executing a program or writing event data to a file.

Internally, NetCrunch uses the XML format for event representation. Although it's text format, you can hardly call it "human readable" format.

Managing Message Formats

[Monitoring > Alert Message Formats](#)

In this window you can see the default message format assignments for all actions.

Message Format Types

There are eight predefined message formats used by different actions:

- txt - text format
- short-txt - short text format
- sms - short text for SMS messages
- syslog - text message for sending to syslog server
- export-txt - text format
- email - HTML email format
- email-txt - text email format
- ticket - text email format

Modifying Message Format Assignment

Each action type has a default message format assigned. You can change the assignment by clicking on a format name in the column Message Format.

Customizing Message for Specific Event

Switch to page Message Definitions. Here you can see message definitions grouped by Message Format. For each message format you can define a custom message format for a specific event or an event class.

Example:

We want to create a new custom SMS text message variant for the Node State Event to include the value of the Location field.

Click [sms-txt](#) .

Click on [Add](#) in the bottom left corner.

Select Node State event from the [Event Class](#) drop down menu. In the 'Event' field select and Apply.

Edit the displayed message content. Click [Add Parameter](#) to add the Location parameter. You can remove the fields that you do not want to include in these alerts..

Save the modified Message by clicking OK.

Customizing NetCrunch Reports

[Read about what custom reports you can create and how it can be done.](#)

Options

➤ [Tools](#) ▶ [Options](#) ▶ [General Options](#) ▶ [Reports](#)

Limit the size of email with reports

Enable the option to avoid sending large emails, as some reports can have multiple pages and can be too large for you mailbox.

Add footer signature

It will be placed at the bottom of each report page. It can be: *atlas name*, *netcrunch server* or *custom text*.

Add logo image

Image of size 100x40 pixels will be placed in every page footer. If you selected larger image it will be automatically resized.

Creating Custom Reports

[Monitoring](#) ▸ [Monitoring Packs and Policies](#) or [Node Settings](#)

You can add new reports to the Monitoring Pack or directly to the node. For instance, you may want to add a report for PING for the specific node.

Open [Node Settings](#)

Click `PING`

[+ Add Report](#)

This will add predefined [PING] Availability Report

NetCrunch contains several template reports, predefined reports and the ability to create custom reports. Template reports are pre-defined reports that need only parameterizing.

Template Reports

Network Services Node Report

Service Availability

The report contains

- Uptime summary,
- Chart of service response time in a given time range,
- Service Think Time – service response delay

Network Services Node Group Reports

The report contains a summary and comparison of a given number (10 by default) of top nodes.

Service Availability Comparison

The report contains a comparison of:

- Check Time - Nodes with the lowest and highest average check time
- Failure Rate - Nodes with the lowest and highest average failure rate

Service Think Time Comparison

Service Think Time is an estimate of the time the service spent on generating the response. It's calculated by subtracting an average PING RTT from the total request time.

Service Uptime Comparison

- Service Availability (by total service uptime)
- Longest service availability and longest service downtime

Performance Reports

Chart Report for Performance Counter

Single Node

This report allows for presenting multiple counter trends in one document. It shows a single line trend chart for a given time range. You can choose to have 4 charts per page or just one.

Multiple Nodes

This report is similar to single node, but trends from multiple nodes (and the same counter) can be grouped on a single chart. You can customize the number of trends put on a single chart.

Custom Inventory Report

You can add a custom report for the Inventory Monitoring Engine: it can show a subset of columns visible on the Inventory page of Node View.

Scheduling Reports

You can have each report you add to a Monitoring Pack or Node Setting be automatically created and scheduled.

Scheduling Scheme

[Monitoring > Report Sheduling Scheme](#)

Each report can be scheduled using a predefined schema with criteria for each report type (*daily*, *weekly*, *monthly*). You need to specify the recipients.

Event Triggers for Counters

NetCrunch allows set up various threshold conditions on performance data regardless of their origin. This works for all performance data channels: from SNMP to custom data received through REST API.

Performance Trigger generates an event upon the condition set on performance counter value.

Thresholds

Thresholds trigger an event when a value is crossing a given border. Depending on the change of direction, it can be a rising or falling threshold condition.

A simple threshold specifies only the threshold value and the direction.

Raising threshold example:

```
% Processor Utilization > 50%
```

Falling threshold example:

```
% Free Disk Space < 10%
```

Hysteresis

Hysteresis can be used to avoid generating too many threshold events on fast changing values. This is simply done by setting an additional reset value.

Example:

Trigger on

% Processor Utilization >= 50%

Reset if

value < 45%

Thresholds can be used on last value or average value calculated in a given time range.

Flat Value Trigger

This trigger generates an event when the counter has (or has not) the same value in the given time period.

Example:

```
% Processor Time = 25 for last 25 min
```

State Trigger

This trigger generates an event when the monitored value changes from one value to another. You need to specify at least one value.

Example:

```
Network Card Error State changes from <any> to 5
```

The event can be reset when the value changes back to the previous state or to any other value.

Value Missing/Exists Trigger

This trigger generates an event if value exists or is missing (can't be read or received) in a given time range.

Example:

It might happen that there is no value until an error occurs, so in such cases the program can react to a value exists condition.

Delta Trigger

This trigger alerts you when the current counter value keeps growing or decreasing by a given value. Or you can define an opposite condition when the counter is not growing or decreasing as expected. Delta is the difference between the last and the previous value.

Example:

```
Device Internal Timer Delta < 1
```

Deviation Threshold Trigger

This trigger allows you to specify how a counter value can differ from the calculated average over a given period of time. The deviation can be set as a percentage or by absolute value.

Example:

Trigger on

% Round Trip Time Deviation > 10%

Reset if

Deviation < 8%

Calculate an average for last

5 min

Range Trigger

This trigger simplifies configuration. You can specify a range instead of two separate thresholds for a low and high boundary. The event triggers if a counter value is in the range or outside the range. Additionally, you can specify the reset tolerance value.

Example:

Trigger on

Server Room Temperature is not in range [20 ... 22]

Reset if

Falls in the range with margin of 2

Baseline Trigger

This triggers allows to set threshold on deviation from observed baseline data. Baseline data are collected by program over a week and stored for the reference. The baseline is calculated for each hour and each day of the week. The user can specify allowed deviation from baseline value (by number of percentage).

Example:

Trigger on

Server Room Temperature deviation from baseline > 1 degree

Note:

Averages can be calculated only if more than 20% of data exists in the time range

Managing NetCrunch Users and Notifications

[Read about users and notification groups in NetCrunch.](#)

NetCrunch User Profiles

The main purpose of NetCrunch user profiles is, of course, authentication and user identification. All consoles (Administration Desktop Console and Web Console) require users to login. Additionally, the Web Console allows you to manage user access to particular Atlas views and nodes.

Administration Console

This console uses profiles to authenticate users and store various settings in user profile on the server. This has been changed in version 9.2 and allows synchronizing user data between workstations where console is installed. It's important that now console user can be recognized when the user adds notes and comments. Each user can easily change its profile now by clicking at user name on top bar.

Web Console

Web Console allows managing access to atlas views, nodes and program features through Access Profiles.

Built-in Admin Profile

NetCrunch uses special, predefined profile "Admin". It cannot be deleted and has always full access to the server. This is very similar to root in Unix family systems. Admin's password can be reset on NetCrunch Server using NCCLI.exe program.

Creating Public Profile

The program allows creating a shared profile which can't be changed by the user. The profile password must be not empty, however.

Resetting the password

We created simple schema for resetting a user's password. Password for built-in user "Admin" can be reset by NCCLI.exe program located in NetCrunch Server folder. Administrator can set a one time password for the user from Desktop Console, and user must change the password at next logon.

Active Directory Integration

NetCrunch allows you to link an NetCrunch account to Active Directory account in order to simplify password management.

After creating a new user, select the [Link with AD Account](#) button and find the desired AD account in object picker dialog.

User can login using NetCrunch user name or AD name. Regardless of which name he uses, his password is checked in AD.

User Notification Profile

Each NetCrunch user has a list of notification profiles that can be restricted to weekdays or day times. User can manage his/her profile and disable particular notification type when needed.

The screenshot shows a user profile configuration interface. At the top, there are fields for 'Username:' (containing 'Admin') and 'Display Name:' (containing 'Administrator'), along with a 'Change Password' button. Below this is a section titled 'Notification Profile' with a plus sign icon. It contains two rows of notification settings. The first row is for 'email' with the contact 'john.doe@gmail.com' and a schedule of 'Everyday, all the time'. The second row is for 'sms via gsm' with the contact '123123' and a schedule of 'Everyday, all the time'. Each row has 'ON' and 'OFF' toggle buttons.

Profile Fields:

The screenshot shows a configuration panel for profile fields. It includes the following sections:

- Type**: *Email of SMS via GSM*
- Message Format**: Message format template
- Email or Phone**: *Email Address or Phone Number* depending on profile type
- Time Restrictions**: You can set *weekdays*, a single day and *day time range* for each profile

Managing Notifications

We recommend to use *Notification Groups* and user profiles for managing notifications. This way groups allow role based management, and each user can adjust his/her notification profile.

Access Profiles

A user can have an assigned access profile to restrict access to certain program features. There are two predefined Access Profiles. You can modify or create new profiles as desired.

Predefined Access Profiles:

- Full Access
- Read-Only

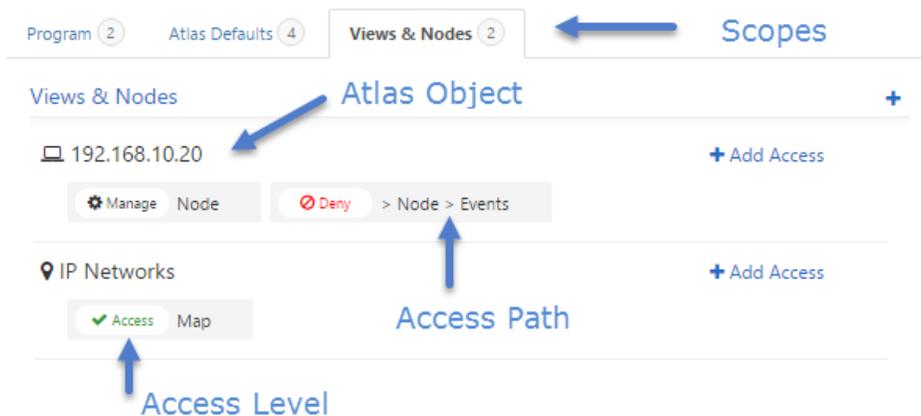
read more about [Managing NetCrunch Access Profiles](#).

Managing NetCrunch Access Profiles

NetCrunch access profile is group of access settings to particular NetCrunch views and operations. Currently it's mostly supported by web access console, remote console requires administration rights.

NetCrunch allows managing access to program objects and its features like views and actions. This allows restricting someone's access to only given Atlas views and nodes.

NetCrunch supports maximum 64 access profiles. Each user can have only one access profile assigned .



The Model

NetCrunch uses simple and powerful model. You can set access to program objects and to their properties. The rights are processed from bottom to top. This means that the lowest level access (longest path) overrides all defaults.

Access Levels

Each element can have one of three access levels:

- Deny - Deny access to a given element,
- Access - Basic access to the object for viewing, reading, executing etc.
- Manage - Allow to change the object properties or configuration

Access Paths

Access path defines the object and hierarchy of its features. The simplest path is Node, and therefore this means we can set access level to the node object. Then we can add another element to the path and Node > Events will mean access to node events. The longer the path, the more detailed access is defined. For example Node > Events > Delete.

- NetCrunch Object hierarchy - Atlas, Atlas Views and Nodes.

- Access Path hierarchy - When we want to narrow scope of functionality for an given object. For example: Node > Events means that we can assign rights to events of the node.

Access Scope

Basically we can add rights to 3 scopes.

- Program - Here you add rights to programs and to manage the server.
- Atlas Defaults - Define default object rights for all Atlas objects here.
- Views & Nodes - Override defaults by adding each object and set access to the object features.

Note:

Administration Console has only one access right defined by now. It allows to use the console and manage the NetCrunch Server. The console requires authentication and user need Server Administration access right. Restrictions to Atlas objects are available in Web Console only.

Managing Calculated Performance Counters

[See how you can create calculated counters from existing ones.](#)

➤ [Monitoring > Calculated Performance Counters](#)

We created Calculated Counters in order to create counters calculated from source counter data upon given arithmetic expression. For example, we may require a counter that represents percentage of free memory, but a particular Cisco device delivers only raw memory values.

Calculated counters extend counters from given Monitoring Engine by arithmetic calculations.

After defining them, you can access them in the same way as any other counters from the given Monitoring Engine. For existing calculated counters, the counter expression can be edited. You can also add new calculated counters.

Creating New Calculated Counter

Click Add Calculated Counter and select Target Monitoring Engine

Select from the list the monitor to which the calculated counter will be added.

Set Counter Name

Each counter is defined by Object, Name and Instance. As the instances are object instances and apply to

each counter of the object, you should specify the object for the counter.

Calculated Counters can inherit instances from source counters only if they are extending the existing source object.

For example:

Object
Cisco Memory
Counter Name
% Memory Used

Edit Counter Expression

To create a counter, add the desired counters and use an arithmetic operation to create an expression. When you add a new counter to the expression, the program automatically creates a variable name and puts it into the expression.

Example:

```
100 * (ciscoMemoryPoolUsed / (ciscoMemoryPoolUsed + ciscoMemoryPoolFree))
```

Devices with multiple network interfaces

In NetCrunch is the network end point unlike in some programs which treat nodes as devices. Instead, a node in NetCrunch is a single interface of the device. As each interface can host different services it makes monitoring of interfaces easier.

You can group nodes by selecting one interface and making it primary (monitored) and other interfaces become secondary interfaces which will be automatically switched into simplified monitoring mode. Simplified monitoring only checks up/down status of the node and does not store any performance data.

- You can group nodes by selecting multiple nodes and then choosing from menu Represent nodes as one device
- You can open properties of primary interface and add secondary interfaces
- You can open secondary interface and select primary interface

Time Restriction Scheme

The program uses time restrictions to time conditions for various elements, such as: monitoring of the node, alerting conditions or notification schemes.

The time restriction scheme has been extended in version 9 to allow for setting a different scheme for each weekday. Basically, you can select single by inclusion or you can exclude specific time range.

Day	Restriction	Time Range
Default range	all the time	▼
Monday	between	▼ 08:00 20:00
Tuesday	default	▼ (all the time)
Wednesday	default	▼ (all the time)
Thursday	default	▼ (all the time)
Friday	default	▼ (all the time)
Saturday	never	▼
Sunday	never	▼

[Reset All to Default](#) [Clear All](#)

OK Cancel

Time Restriction Scheme is used for:

Actions

You can set when action can be executed

Event Condition

For some conditions like for example: *event happens in some time range*

Notification

In user profile each notification type can have different time range settings. For example: you can receive some notification type on weekdays and other on weekends.

Atlas Monitoring

Here you can set a global monitoring scheme for whole Atlas.

Node Monitoring Scheme

Each node can have its own monitoring time scheme

Node State Alert

You can set alert if node is not in right state at the given time (for example in instances when it should be DOWN at night).

Action Restrictions

Each action can have its own restrictions.



The screenshot shows a configuration panel titled "Restrictions". It contains three fields: "Time:" with a dropdown menu showing "Everyday, all the time", "Atlas View:" with a dropdown menu, and "Severity:" with a dropdown menu.

This makes action lists more flexible as you can limit execution of the action to:

- Time Range (see [Time Restriction Scheme](#))
- Atlas View
- Alert Severity

This allows defining complex conditions and action lists.

Usage Examples

Run different actions for different time of day or weekday

Define two actions and set different time of day or different weekdays.

Run actions based on node location

You may want to notify different groups of administrators depending on the node location. NetCrunch creates Atlas views for locations. Also you can create custom views based on custom fields (it can be additional information like department or other organizational unit) and you can assign each view to each action.

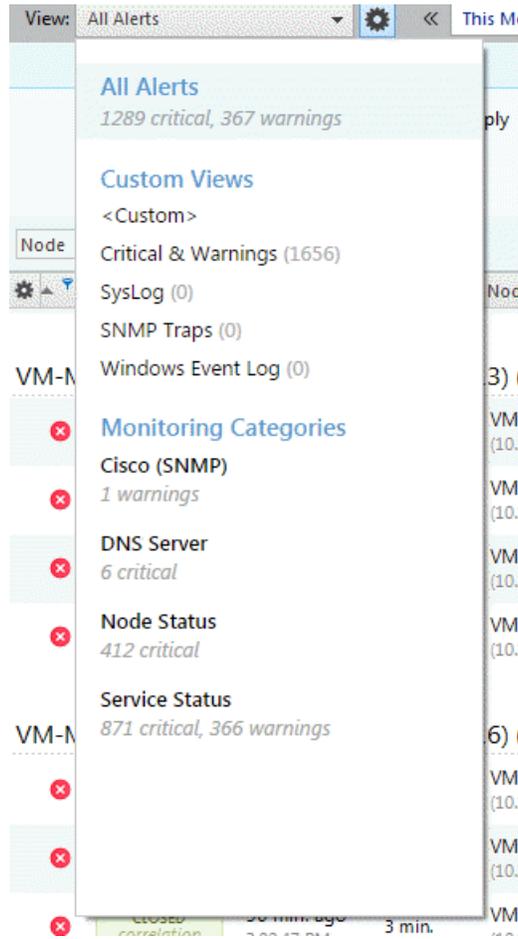
Restart server if the alert is critical after some time. Weekend nights only.

You might assign a restart action to specific alert, but you might also decide to restart servers after certain amount of time if the issue is not resolved. This needs limiting action only to nodes being members of *Servers* view. As each action has time after it executes since the alert started, you can decide to run restart after some time. Additionally you might want to repeat these steps only on weekend nights.

Managing Event Log Views

Automatic Views

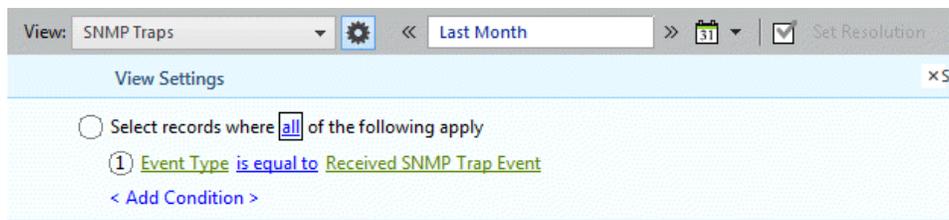
There are many views created automatically that you can select from the view dropdown. These views are created automatically for Monitoring Packs and Sensors.



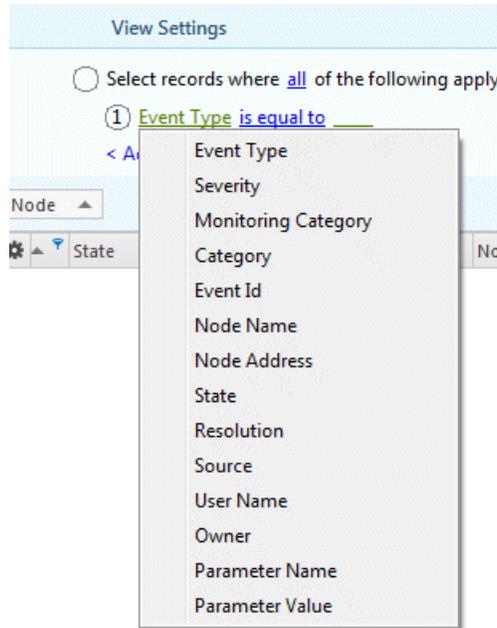
Creating Custom Views

NetCrunch can store millions of events in history event log. These events are fully searchable but searching them in the text form would be very slow.

To make it easy, we added a visual query builder, which allows you to create filtering conditions for the database. Once you define your view, you can store it and use it later.



You can create views using many event fields. You can also search event parameters.



Searching Event Parameters

The best way to search is to create a bracket condition and select a parameter name and value you are looking for.

Selecting Time Range for the view

View definition does contain time range. So you can select any time range for each view. You can do it before or after selecting given view.

Managing Multiple Node Settings

Read how to change node settings such as: monitoring time, device type for multiple nodes, and how to manage alerting and reporting settings differently.

You can always select multiple nodes, but if you want to manage nodes as a group, consider creating a separate Atlas view for the group.

It's very easy to create a new Atlas View by selecting multiple nodes and dragging them into the Custom Views section.

Changing Node Settings

NetCrunch allows you to easily manage multiple node settings. Select multiple nodes in a table or on the graphical icon view and then select `Node Settings` from the context menu.

You can use existing Atlas views to narrow your selection.

For instance, if you want to select all Windows servers, you can go to the view

`Custom Views > Server Types > Windows Server` and press `Ctrl+A` to select all nodes in the view. Then

you can select [Node Settings](#) from the context menu or press [Shift-F2](#).

Managing Alerts & Reports

We strongly encourage you to create Monitoring Packs in any instance where you want to have the same settings for multiple nodes. You will then be able to select multiple nodes and add your new Monitoring Pack to them.

On every single node you can override settings assigned by Monitoring Packs. For instance, you can change actions (add some) or even disable some alerts.

When you select multiple nodes, however, you can only add or remove Monitoring Packs.

See also:

[Alert and Report Management](#)

Read about scheduling reports, the difference between an event and alert, Monitoring Packs and

Configuring Notification Services

[Read to configure emails and text messages \(SMS\) with NetCrunch notifications.](#)

You can receive notifications about alerts and reports as emails or text messages (SMS). First, you need to setup the configuration parameters.

Go to the [Options](#) page and click [Notification Options](#).

Configuring external email server

You can use the built-in NetCrunch mechanism or an external SMTP mail server. The system also supports the TLS encryption protocol, if needed.

For the external SMTP server you must provide its name, the port number or login credentials. We recommend to use this option due to possible network security restrictions.

Reply Address

The reply email address used by the program when sending emails. It displays in the "from" field when you receive a NetCrunch email message.

Configuring GSM modem or phone

For alert notifications via text messages (SMS), you need to select the COM port used to communicate with the mobile phone, SMS settings and options related to the GSM device such as the PIN of the SIM card. You may also need to enter AT+C commands, if they require additional configuration.

You can even use a standard cable attachment, which after installation will be visible in the system as one of the computer's COM ports.

COM Port

Shows the COM port selected for communication with the mobile phone. Clicking the [Browse](#) button opens the *GSM Device Discover* dialog window, where you should choose/configure the appropriate COM port and test if NetCrunch can connect to the device. If there are any connection problems, check the modem documentation to see what speed is required to work properly.

SMS Settings

Limit to Single Message

Messages longer than 160 chars will be split into several messages. You will receive them as a single piece, but you will pay for the number of messages at your operator's rates.

Encoding

The *Auto* option is enabled by default here. Change it if you receive text messages with encoding issues.

Modem Settings

Initialization AT+C Commands

Select this check box if the modem requires additional initialization AT+C commands.

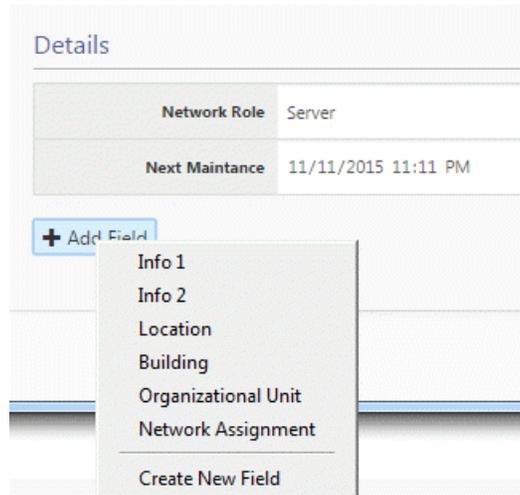
See also:

- [Options: General](#)
- [Options: Monitoring](#)
- [Options: Notification](#)
- [Options: Map](#)

Managing Node Custom Data

You can easily extend NetCrunch node database by adding custom fields to each node. This allows creating views and controlling alerting action execution.

NetCrunch allows you to add custom fields to node properties. There are several defined fields; some fields (like info1, info2) are there for compatibility with old versions. Fields can be used for organizing data.



You can add these types of fields: number, text or date and time.

Custom node fields can be used for creating dynamic views (defined by filtering condition). Atlas view membership also controls alerts and action execution. See [Action Restrictions](#)

Configuration Tips

Read what configuration problems you should fix in order to make sure everything works fine.

NetCrunch does many things automatically, but in some places it needs your input. For instance, it's necessary to provide proper credentials to access operating systems, and valid SNMP profiles.

Resolve the Monitoring Issues

Monitoring Issues are problems related to the monitoring process. They definitely require your attention. In most cases, they are related to invalid or missing credentials.

Fix Missing Device Types

NetCrunch uses Device Types to implement automatic monitoring. For example, if a device is not set to be a Windows type, then Windows monitoring will not start or even be accessible for the node.

Open a network map and look out for icons with a question mark, which indicate that they are missing a device type. NetCrunch automatically detects device vendors, which can be helpful in setting the device type.

Only Windows (other systems if they were added to Active Directory) and SNMP devices should have their device type set automatically. For other devices (those displayed as the "unknown device type") you need to set their device type manually.

Review the Settings page

Please check the Settings page to see what is enabled or disabled in your monitoring configuration.

For example, you can enable monitoring of the Physical Segments or NetFlow here.

See also:

[Devices with multiple network interfaces](#)

Monitoring

Everything about monitoring configuration. Read how you can monitor various aspects of the network hardware and software.

Monitoring Concepts

Review basic concepts like Monitoring Engines, Node State Monitoring and how to disable monitoring.

The main purpose of NetCrunch is to monitor your network, however it also offers you documenting and organizing services.

So, everything in NetCrunch is well organized - the Atlas root is at the top, the views go below, then there are nodes, and finally on the lowest level there are various objects and their performance counters.

Monitoring Engines

Monitoring Engine is a software component responsible for specific type of monitoring.

Monitoring Engines simplify the monitoring configuration, as they are responsible for certain channel of monitoring.

For example, you might define many things being monitored using SNMP Engine but all of them share the same configuration parameters for a particular node like: SNMP port and profile to be used. Very similar situation happens with other operating systems where the configuration contains credentials for the connection.

Note:

If you are familiar with other concepts such as: sensors or probes, please be advised that they are much more comparable to Monitoring Packs in NetCrunch.

NetCrunch Monitoring Engines:

- SNMP - *requires SNMP to be configured on node*
- Windows - *node must be Windows Type*
- Linux - *node must be Linux type*
- Mac OS X - *node must be Mac OS X and system newer than 10.4*
- BSD - *node must be one of BSD systems*
- Solaris - *node must be Solaris type*

- Inventory - *node must be Windows*
- ESXi - *node must be ESXi version > 4.5*
- Network Services
- NetFlow
- NC Open Monitor - *currently not available on a node level*

Monitoring Sensors:

- Apache Server - it collects and aggregates performance data from Apache Web Server
- Email Alert (IMAP4) - It allows receiving and parsing emails in order to convert them into alerts. It uses IMAP4 protocol.
- Email Alert (POP3) - The same like the one above, but it uses POP3 protocol.
- Email Mailbox (IMAP4) - It allows monitoring of a given mailbox. It can check access, authentication and alert if mailbox has not been emptied for some time.
- Email Mailbox (POP3) - The same like the one above, but it uses POP3 protocol.
- Remote File (FTPS/S) - Check remote file content, size, authentication and more. Supports FTP, SFTP and FTPS.
- Web File (HTTP/S) - Very similar to one above but retrieves files using HTTP or HTTPS.
- Windows File - Very similar to one above but retrieves files windows SMB protocol. Supports ASCII, UTF-8, UTF-16 and Base64 files.
- Text Log - Analyze remote text logs. Supports FTP/S and HTTP/S.
- Remote Folder (FTP/S) - Check folder content, authentication and other conditions. Supports FT, SFTP and FTPS.
- Windows Folder - Similar to the one above, but uses windows SMB protocol.
- Round Trip Email - Send email to mailbox and retrieve it, in order to check if both incoming and outgoing mail services are working properly. Supports POP3, IMAP4 and SMTP and secure protocol versions.
- DNS Query - Check if record in DNS has not been altered and return expected values. Supports IPv4 and IPv6
- Reverse DNS Query - Send reverse query (address to name resolution). Supports IPv4 and IPv6
- Basic HTTP/S Request - Send simple request using GET, HEAD or POST in order to check some service or API. It measures response time and allows validating response content.
- Web Page Sensor - Fully validate web page content and loading process in the same way like a regular browser does it . Alert on content, authentication error or resource load errors. Supports HTTP and HTTPS.
- Generic Agent Sensor - It's actually a placeholder for external data. After adding it to node, agent can send data (using REST API) to this node as it would be regular sensor. Define counter triggers and alert on received status objects. It's the simplest way to extend NetCrunch monitoring using Curl.

Node State Monitoring

The central part of monitoring is determining the node state. Everything being monitored on the node depends on its state.

When the node is considered to be DOWN, monitoring of the node almost stops until NetCrunch recognizes that the node is responding again.

NetCrunch determines node state upon monitoring of network services (see: [Network Services Monitoring](#))

Read more about [Monitoring of Network Nodes](#)

Managing Monitoring Engines

When you go to [Node Settings](#) ▶ [Monitoring](#) you will get the list of monitors available (bound) to the node. There you can manage engine properties for the node. Many monitoring engines bound to the node only when the node has a certain device type.

Monitoring by Device Type

Setting Device Type is not only important for the Operating System monitors, but also for automatic [] (@monitoring-packs).

Read more in [Automatic Monitoring and Organizing](#)

Monitoring Issues

Monitoring Issue is a problem related to the monitoring process, like missing credentials or improper response received from the device.

We decided to make issues related to monitoring process stand out of other problems related to your network services and devices. They are usually related to wrong or missing credentials or Windows security settings.

See also: [Configuration Tips](#)

Disabling Monitoring

As the monitoring is enabled by default, it's more interesting to know how you can disable it. You can disable it at any level.

Disabling the Atlas

[Atlas](#) ▶ [Properties](#)

You can disable monitoring for the entire Atlas. You can disable it for some time (for maintenance) or schedule being disabled for a future time range.

Disabling the Network

[IP Network](#) ▶ [Properties](#)

You can go to IP Networks and disable monitoring of a specific Network indefinitely, or just for a given time

period.

Automatically by Dependencies

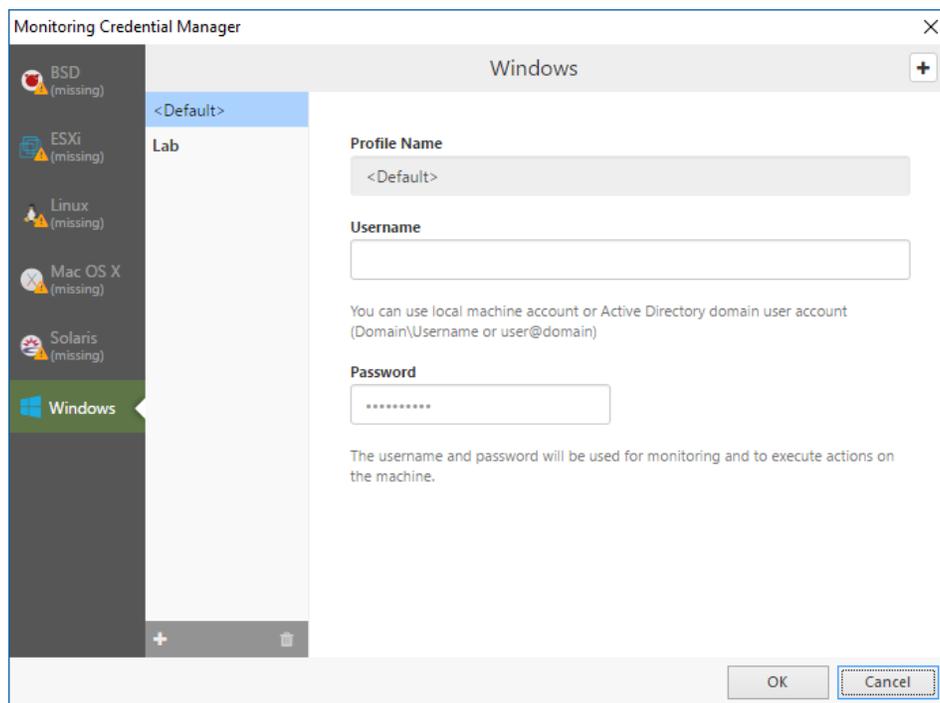
NetCrunch manages monitoring dependencies that should reflect connection dependencies. If properly set, NetCrunch automatically disables monitoring of nodes depending on a certain connection. This helps to lower monitoring traffic and also prevents false alerts.

Read more in [Preventing False Alarms](#)

Managing Monitoring Credentials

In order to access monitored systems, NetCrunch needs valid credentials for each system. You can manage these credentials using profiles or set custom credentials for each node and system separately.

To maintain compatibility with previous versions there is one default profile for each operating system defined. You can create more profiles by pressing plus button located at the top right corner of the window.



All credentials are stored securely on the NetCrunch Server and passwords can never be retrieved by the console.

Monitoring of Network Nodes

[Read about node state evaluation and node monitoring settings](#)

As the node represents a single network endpoint (not the device), which is a basic subject of monitoring,

everything else in monitoring depends on its state.

Node State

NetCrunch defines the following node states:

OK

Everything is fine - node and its services are responding.

Warning

There is some problem with a service or the Monitoring Engine on the node. Monitoring Issues also turn node into a Warning state.

DOWN

Node is not responding - in this state only one service is monitored to bring node back to the normal monitoring state.

Unknown

Node state can't be determined - for example the node has no IP address or has not been monitored yet.

Disabled

Node monitoring is disabled for some reason. There are a number of reasons why node is disabled, such as: by Atlas, by Dependencies or by the User.

Simplified Monitoring

Default setting for workstations - program determines only a node and a service state by checking network services availability (for example PING) without tracking any performance metrics.

Extended Service Monitoring

In some cases, status of the node can be determined every second using the leading service, which can be set to extended monitoring mode.

Read more about [Network Services Monitoring](#)

Monitoring Dependencies

Monitoring of each node can depend on its parent node (for example network switch or router). This causes an automatic disabling of node monitoring in case the parent link or device is DOWN.

Read more in [Preventing False Alarms](#)

Prioritization and Event Suppression (XE only)

In large networks with remote intermediate routers NetCrunch organizes monitoring by priorities. By default, the nodes being closer to NetCrunch Server and intermediate routers are monitored before others.

Event Suppression is the technique of preventing false alarms caused by intermediate network connection failure.

Monitoring Time Mask

When a node is active in certain hours or days only, then you might want to decide to limit its monitoring to a given time range and weekdays.

Network Services Monitoring

Availability and performance monitoring of 65 network services like: FTP, HTTP, SMTP, etc.

Monitoring of network services is the basic monitoring type in NetCrunch. A node state is determined basically on the availability of network services. When a node is in the DOWN state, it's only monitored by a single network service.

Availability and Response Time

Services monitoring checks basically:

- Connectivity
- Response received
- Response Time and Failure Rate

NetCrunch sends a request, appropriate for a given service protocol, and then checks if the response matches the defined response. In order to measure a response time the process should be at least repeated 3 times and then the average response time is calculated. For each request you can set an appropriate time to wait for a response.

Each monitored service provides the following performance metrics:

- Round Trip Time - total time to send and receive a single response or a connection time for extended services
- Check Time - total checking time (including sending multiple request in extended mode)
- % Failure Rate - calculated per each service checking
- % Packet Lost - calculated per each service checking
- Transfer Rate kb/s - only services transferring data, like HTTP/S or FTP.

Leading Service

Leading Service is a network service designed to be checked as the only service, when the node is DOWN.

Extended Monitoring

For some critical nodes you might want to react in seconds instead of minutes. Then you can set service monitoring to Extended Monitoring.

Extended Service Monitoring allows checking a leading service more often than once per minute whenever

a node is DOWN or alive. As an option, you can set NetCrunch to determine the node state solely upon the state of the leading service; otherwise it can check other services immediately after the leading service fails.

Customizing Existing Services

NetCrunch allows monitoring of network services by their respective default ports. When you need to monitor a service on different ports, go to [Tools ▶ Options ▶ Monitoring ▶ Network Services](#) and choose [New Service](#). Then you will be able to duplicate the service with the new name and different port. For example, simply define HTTP_8080 for checking HTTP on port 8080.

Simple TCP checking service

In some cases all you need is a simple TCP port connection checking, without sending any further data. In order to create simple TCP port checking go to [Tools ▶ Options ▶ Monitoring ▶ Network Services](#), click [New Service](#) and select a desired option.

Defining Custom Services

As a third option, you might decide to create a full request/response checking service definition. In order to create such definition, go to [Tools ▶ Options ▶ Monitoring ▶ Network Services](#) click [New Service](#) and select Create from Scratch option.

Set Protocol Type (TCP, UDP or TLS/SSL) and Port Number.

Define Request to be sent (after connection if TCP used) - you can enter either text or binary data in the hexadecimal format.

Define response patterns. You can set multiple patterns and decide how they should be checked. Patterns can be: *text*, *hexadecimal binary data* or *regular expression*. Pattern matching options are:

- any of the patterns match
- all of the patterns match
- none of the patterns match

Automatic Discovery or Network Services

As the node status depends on the services, each time a new node is added to atlas, NetCrunch automatically discovers services running on the node. By default, NetCrunch is configured to check only a subset of the total list of defined services.

You can manage the list of services being automatically discovered in

[Tools ▶ Options ▶ Monitoring ▶ Auto Discovered Services](#)

Troubleshooting

In order to get some monitoring services to work properly, an additional configuration task should be performed.

DHCP Server Checking Restrictions

- NetCrunch must be able open the 68 UDP (DHCP client) port to receive a response to the *DHCP inform* request. The monitoring will not be working if NetCrunch is running on a machine where the DHCP Server has been installed, as it uses the same port.
- The IP address of NetCrunch machine must be included in any checked DHCP Server Scope (WINDOWS DHCP Server tested). For example, if the machine where NetCrunch is running has the address 192.168.1.100, and the DHCP Server is on 192.168.88.10, then the server must have the scope with any range from 192.168.1.x addresses. A Linux-based DHCP Server must have the authoritative option enabled.

Monitoring MSSQL Express

In order to monitor MSSQL Express, the TCP/IP protocol must be enabled, and the server instance must accept a remote connection on TCP port 1433. Please refer to the MSSQL Express documentation for information about how to enable the TCP/IP and allow the remote connection to the server instance.

SSH Service

NetCrunch allows monitoring SSH service using a protocol ver. 1 and/or ver.2. By default, the SSHv2 protocol is monitored on the network nodes. Use SSHv1 service in the case when the node supports the older version of SSH.

Operating Systems Monitoring

[Read about monitoring of Windows, Mac OS X, Linux, BSD, Solaris and ESXi systems.](#)

All operating system in NetCrunch are monitored without installing any agents. It's convenient, but sometimes requires extra settings to be set on the systems.

Windows

Monitoring Windows computers without agents depends on two factors:

- Windows setup - allowing remote access to the system and getting performance information,
- Node Device Type - it must be set as *Windows* system - which you have to set manually when you are adding a new computer to NetCrunch manually.

Setup

Windows is the most common desktop OS system now, and will be for a while. The Windows is full of paradoxes: it's the most secure OS with all security turned off for years; it's based on the same kernel but hardly can connect between different versions. Finally, the latest server system from Microsoft does not

allow any remote operations without prior configuration. All problems seem to be solved when Windows computers work in the AD Domain, as every problem has roots in various security settings. So, before you start monitoring Windows, please read [Windows Monitoring Setup](#) article, which will explain how to configure Windows systems for monitoring. It describes various situations, including standalone servers and workstations. We also prepared a shell script for configuring settings for standalone systems.

What can be monitored

Windows monitoring relies on a monitoring of Windows services, Windows Event Log and perfmon counters. You can take Windows Monitoring Packs as an example of how various aspects of the system and application can be monitored on Windows.

Monitoring of Windows services and Event Log can be enabled in [Node Settings ▶ Monitoring ▶ Windows](#).

Windows Services

You can control services manually ([Node Status ▶ Windows Services](#)) or by the alert action, you can setup an alert on Windows Services' state. You can setup alerts on the following conditions:

- Selected service is Paused
- Selected service is Running
- Selected service is Stopped
- Selected service is not Running

Windows Event Log

NetCrunch can monitor Event Log entries on a given computer. It's being done by a WQL query that NetCrunch automatically builds upon your parameters. You can setup this monitoring by adding an alert to *Windows Event Log* sensor in [Node Settings ▶ Monitoring](#). Many of predefined Monitoring Packs also enable Event Log monitoring.

Note:

Specify a narrowest query possible. Windows Event Log entries are large and monitoring all Windows Event Log entries on even relatively small number of computers will overwhelm the NetCrunch Event Log database.

Performance Counters

Windows offers a number of built-in performance counters which are extended by the installed applications. NetCrunch allows defining several types of [Event Triggers for Counters](#) on Windows counters. You can setup triggers by creating a new alert and adding it to a node or to the Monitoring Pack.

[Settings ▶ Monitoring Settings](#) or [Node Settings ▶ Monitoring](#)

Windows OS Monitoring Packs

There are several Monitoring Packs which you can use for monitoring different aspects of your Windows environment

Active Directory (automatic)

Observe Replication and Service error. Watch Active Directory services status.

Operating System must be Windows Server

Monitored network services list contains: LDAP

Basic Windows Monitoring (automatic)

Provides basic workstations monitoring. Observe processor utilization, memory usage and free disk space.

Operating System must be Windows Workstation

Simplified Monitoring must be Disabled

DHCP Server

Observe DHC Server service and its errors.

Distributed File System (DFS)

Observe Windows Event Log for specific DFSR warnings and errors

DNS Server (automatic)

Observe DNS errors. Watch DNS network service and Windows service status.

Operating System must be Windows Server

Monitored network services list contains: DNS

Hyper-V Server

Observe the overall processor utilization of Hyper-V environment and watches it's Windows service status

Network Services Health

Watch for DHCP, DNS, WINS or other TCP/IP errors.

Processes (Windows)

It allows collect information for processes

Security Audit

Watch for Account events, logon and password problems

Terminal Services

Watch number of Active and Inactive sessions

Windows Server (automatic)

Monitor typical system performance indicators like %Processor Time, Memory, Disk Free space, disk latency etc.

Operating System must be Windows Server

Windows vCenter 5.1

Observe state of vCenter Windows services

Windows Applications

Exchange 2003

Monitor key Exchange Windows services, monitor Windows event log for Exchange event errors and watch the important performance metrics such as mailbox or SMTP queues

Exchange 2007-2010

Exchange 2007-2010 Client Access Server

Monitor key Windows services and performance counters of the Client Access Server, generates *IMAP4 and POP3 availability report*

Exchange 2007-2010 Mailbox Access Server

Monitor key Windows services and performance counters of the Mailbox Access Server

Exchange 2007 - 2010 Transport Access Server

Monitor key Windows services and performance counters of the Transport Access Server, generates *SMTP availability report*

Exchange 2013

Exchange 2013 Client Access Server

Monitor key Windows services and performance counters of the Client Access Server, generates reports about *IMAP4, SMTP, UMCallRouter and POP3*

Exchange 2013 Mailbox Access Server

Monitor key Windows services and performance counters of the Mailbox Access Server

Forefront TMG 2010

Monitor key Windows services and performance counters of the TMG 2010 such as server cache, number of denied packets, web proxy requests and other

IIS

Monitor key IIS performance metrics such as ASP requests, IIS Private Bytes and monitor windows event log for ASP, SMTP and WWW errors

ISA Server

Monitor key performance metrics, Windows Services and ISA Server event log errors

MS BizTalk Server 2009/2010

Monitor key Windows Services of the BizTalk Server

Ms Dynamics AX 2012 Server

Monitor number of active sessions and Windows Service of the Ms Dynamics Server

MS Dynamics CRM 2011 Server

Monitor key performance metrics, Windows Services and MS CRM errors

MS Dynamics NAV Server

Monitor key Windows Services of the MS dynamics NAV Server

MS Index Server

Monitor status of Microsoft Indexing Service

MS Project Server

Monitor key Windows Services of the MS Project Server

MS SQL Server

Monitor key performance metrics, Windows and Network Services, MS SQL event log warnings and errors. Also contains several reports such as:
Processor Bottleneck Analysis, Disk Usage and Performance, Memory Usage Analysis, MS SQL Server CPU Report, MS SQL Server I/O Report

SharePoint

Watches the status of SharePoint Windows Services, number of rejected requests, cache size and number of queued requests

Actions

Run Windows Program

Program can be copied to and executed on the desired machine.

Run Windows Script

Run script that can be copied to and executed on the desired machine by given scripting host.

Terminate Windows Process

Terminates process by its name (Windows nodes only).

Start, Stop, Pause Windows Service

Perform control action on given service. (Specify service by its name or select from the list or running services).

Linux

NetCrunch monitors Linux without agents using an SSH script, which is automatically copied to a remote machine. In order to start monitoring Linux system, you must set a node Device Type to Linux to let NetCrunch recognized it properly.

Preparing Node for Linux Monitoring

Open [Node Settings > Properties](#) and set the Device Type to Linux (*Device Class: Server/Workstation, Operating System: Linux*).

After that changing Device Type closes the Node Settings window in order to save changes and let Linux Monitoring Engine recognize the change. Click .

Open [➤ Node Settings ▶ Monitoring](#) . Now you can see parameters to monitor Linux. Enter SSH credentials unless you use default settings for Linux that you could enter in [➤ Tools > Options > Monitoring > Default Node Settings](#) .

Linux Monitoring Packs

Monitor most important Linux performance indicators such as: processor and memory utilization, free disk space, available swap, and create Linux Server Report.

Linux (automatic)

Monitors basic system parameters like: *Available Memory, Swap Available, Processor Utilization* and *Volume Free Space*

Linux (SNMP)

It requires SNMP to be enabled. It monitors: *CPU Load, Load Check, Minimum Swap Space, Swap Space*

Network Traffic (automatic)

It allows collecting data for traffic statistic on Top Charts

Processes

It allows to collect information for processes

Actions

Run SSH Script,

Run script using SSH connection can be copied to and executed on the desired machine by given scripting host.

Scripts

- Shutdown Linux Machine
- Reboot Linux Machine
- Restart Linux Machine
- Mount CD-ROM
- Dismount CD-ROM

BSD

Monitoring Packs

Monitor most important BSD performance indicators, such as processor and memory utilization, free disk

space, and create BSD Report.

BSD (automatic)

Monitors basic system parameters like: *Available Memory, Processor Utilization and Volume Free Space*

Processes

It allows collecting information for processes

Network Traffic (automatic)

It allows collecting data for traffic statistic on Top Charts

Actions

Run SSH Script,

Run script using SSH connection can be copied to and executed on the desired machine by given scripting host.

Solaris

NetCrunch monitors Solaris without agents using only an SSH script. This script is automatically uploaded to the remote machine. In order to start monitoring a Solaris system, you must set a node Device Type to Solaris to let NetCrunch recognize it properly.

Monitoring Packs

Monitor most important Solaris performance indicators such as: processor and memory utilization, free disk space, and create Solaris Report.

Network Traffic (automatic)

It allows to collect data for traffic statistic on Top Charts

Processes

It allows to collect information for processes

Solaris (automatic)

Monitors basic system parameters like: *Available Memory, Swap Available, Processor Utilization and Volume Free Space*

Solaris (SNMP)

It requires SNMP to be enabled. It monitors: *CPU Load, Load Check, Minimum Swap Space, Swap Space*

Actions

Run SSH Script,

Run script using SSH connection can be copied to and executed on the desired machine by given scripting host.

Mac OS X

Monitoring Packs

Monitor most important Mac OS X performance indicators. such as processor and memory utilization, free disk space, and create Mac OS X Report.

Mac OS X (automatic)

Monitors basic system parameters like: *Available Memory, Processor Utilization and Volume Free Space*

Processes

It allows to collect information for processes

Actions

Run SSH Script,

Run script using SSH connection can be copied to and executed on the desired machine by given scripting host.

Other Operating Systems

VMWare ESX/ESXi

NetCrunch supports directly monitoring of ESX and ESXi (VCenter is not required). Read more in [VMWare Monitoring](#).

Novell NetWare

As each NetWare system has a pre-installed SNMP agent, monitoring is possible using SNMP.

NetWare Monitoring Pack

NetWare (SNMP)

It requires SNMP to be enabled. It monitors: *CPU Load, Cache buffers, Volumes Free Space, Novell Volumes*

IBM AIX and AS/400

IBM AIX and AS/400 systems can be monitored via SNMP.

IBM Monitoring Packs

AIX (SNMP)

Monitor processor, memory and file system

AS/400 (SNMP)

Monitor processor, memory, errors, sessions, RWS controller

Application Monitoring

NetCrunch supports monitoring many applications by Monitoring Packs. It's also easy to create agents for sending data to NetCrunch.

Monitoring Windows Applications

NetCrunch contains predefined monitoring packs for common Windows applications.

Exchange 2003

Monitor key Exchange Windows services, monitor Windows event log for Exchange event errors and watch the important performance metrics such as mailbox or SMTP queues

Exchange 2007-2010

Exchange 2007-2010 Client Access Server

Monitor key Windows services and performance counters of the Client Access Server, generates *IMAP4 and POP3 availability report*

Exchange 2007-2010 Mailbox Access Server

Monitor key Windows services and performance counters of the Mailbox Access Server

Exchange 2007 - 2010 Transport Access Server

Monitor key Windows services and performance counters of the Transport Access Server, generates *SMTP availability report*

Exchange 2013

Exchange 2013 Client Access Server

Monitor key Windows services and performance counters of the Client Access Server, generates reports about *IMAP4, SMTP, UMCallRouter and POP3*

Exchange 2013 Mailbox Access Server

Monitor key Windows services and performance counters of the Mailbox Access Server

Forefront TMG 2010

Monitor key Windows services and performance counters of the TMG 2010 such as server cache, number of denied packets, web proxy requests and other

IIS

Monitor key IIS performance metrics such as ASP requests, IIS Private Bytes and monitor windows event log for ASP, SMTP and WWW errors

ISA Server

Monitor key performance metrics, Windows Services and ISA Server event log errors

MS BizTalk Server 2009/2010

Monitor key Windows Services of the BizTalk Server

Ms Dynamics AX 2012 Server

Monitor number of active sessions and Windows Service of the Ms Dynamics Server

MS Dynamics CRM 2011 Server

Monitor key performance metrics, Windows Services and MS CRM errors

MS Dynamics NAV Server

Monitor key Windows Services of the MS dynamics NAV Server

MS Index Server

Monitor status of Microsoft Indexing Service

MS Project Server

Monitor key Windows Services of the MS Project Server

MS SQL Server

Monitor key performance metrics, Windows and Network Services, MS SQL event log warnings and errors. Also contains several reports such as:
Processor Bottleneck Analysis, Disk Usage and Performance, Memory Usage Analysis, MS SQL Server CPU Report, MS SQL Server I/O Report

SharePoint

Watches the status of SharePoint Windows Services, number of rejected requests, cache size and number of queued requests

Custom Monitoring

Using Windows Services and Counters

You can look at various Monitoring Packs defined in NetCrunch. Any other application can be monitored in a very similar way - if it supports perfmon counters; otherwise you can rely on monitoring Windows services and processor and memory parameters for specific processes.

Using Scripts and Agents

This is another option for monitoring Windows applications as well as any other application, including the hosted remote application or even websites.

You can create a script, which will poll necessary information for NetCrunch, or you can modify the existing application to send REST requests.

It doesn't involve much programming as you can use cUrl, an open source project available for almost any platform.

You can find it at:

Read more about [Sending Data to NetCrunch](#)

VMWare Monitoring

See [howto monitor virtual machines, datastores, hardware status, and the performance counters of host and guest VMs](#)

NetCrunch supports ESX and ESX/i version 4, 5 and 6. It connects directly to the ESX servers, so it doesn't need vSphere to be installed. NetCrunch comes with pre-configured Automatic [Monitoring Packs](#) to monitor ESX as soon as the device type is set to ESX.

ESXi nodes View

The view gives you quick overview of monitored ESX hosts.

St...	Icon	Node	Engine Status	Last Response	Version	Up Time	CPU Usage %	Memory Usage %	Virtual Machines
✓	ESX	labesi52.lab.ad.adrem 192.168.2.52	✓ Connected	< 1 min. ago	VMware ESXi 5.0.0 build-1311175	55 days	29%	79%	Running: 23, Total: 23
✓	ESX	labesi21.lab.ad.adrem 192.168.2.21	✓ Connected	< 1 min. ago	VMware ESXi 5.0.0 build-1311175	55 days	27%	83%	Running: 2, Total: 2
✓	ESX	labesi38.lab.ad.adrem 192.168.2.38	✓ Connected	< 1 min. ago	VMware ESXi 5.0.0 build-1311175	55 days	5%	59%	Running: 3, Total: 102
✓	ESX	labesi51.lab.ad.adrem 192.168.2.51	✓ Connected	< 1 min. ago	VMware ESXi 5.0.0 build-1311175	53 days	5%	30%	Running: 9, Total: 15
✓	ESX	vm-monit-097.lab.ad.adrem 192.168.3.197	✓ Connected	1 min. ago	VMware ESXi 5.5.0 build-1623387	55 days	4%	31%	Running: 3, Total: 3
⚠	ESX	labesi61.lab.ad.adrem 192.168.2.61	✓ Connected	< 1 min. ago	VMware ESXi 5.0.0 build-1311175	55 days	3%	19%	Running: 9, Total: 9
⚠	ESX	labesi62.lab.ad.adrem 192.168.2.62	✓ Connected	1 min. ago	VMware ESXi 5.0.0 build-1311175	55 days	3%	19%	Running: 7, Total: 8

Virtual Machines

For each ESXi machine, NetCrunch lists and monitors virtual machines. You can add virtual machine to atlas to be monitored only if the machine is running and has IP address assigned.

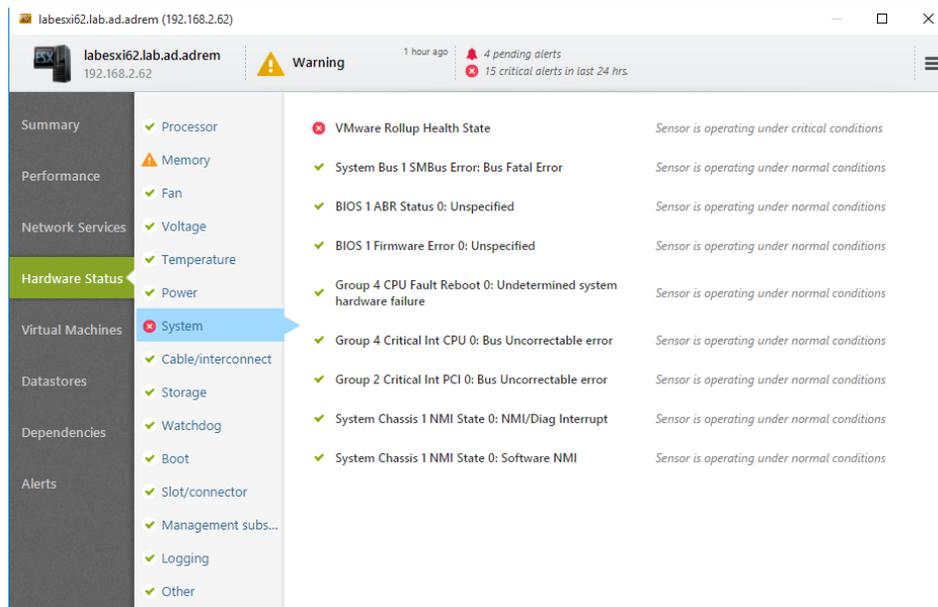
Datastores

NetCrunch allows monitoring of VMWare data stores. You also see their properties in node status window.

[➤ Status](#) [▶ Datastores](#)

ESX Hardware Status

NetCrunch tracks all hardware status provided by ESX server.



Counters can be monitored by ESX or using the VM machine instance Counters:

Performance Counters

NetCrunch provides counter for both host ESX system and guest VM.

You can set triggers on those values using [Event Triggers for Counters](#)

Datastore - Counter Object

- % Free Space
- Capacity Bytes
- Free Space Bytes
- Host Count
- Uncommitted Space Bytes
- Virtual Machine Count

Summary - Counter Object

The object provides performance counters for a given ESX server.

- %CPU Usage
- % Memory Usage
- Disk IO rate Bytes per sec.
- Network Utilization Bytes per sec.
- Running VM Count
- Total VM Count

- Up Time
- Used CPU Hz
- Used Memory Bytes

Virtual Machine -Counter Object

The object provides performance counters for a given guest system (VM).

- %CPU Usage
- % Guest Memory Usage
- % Host Memory Usage
- Guest Used Memory Bytes
- Host Used Memory Bytes
- Network Utilization Bytes per sec.
- Used CPU Hz

Setting Alerts

If you want to add alerts to the single ESX server, just go to [Node Settings ▶ Monitoring](#). You can add new or overwrite alert rules defined by ESX Monitoring Pack.

The other option is to modify ESX Monitoring Pack then go to [Monitoring > Monitoring Packs & Policies](#) and edit VMWare ESX/ESXi monitoring pack.

Adding ESX/ESXi guests

[Node Settings ▶ Monitoring ▶ ESX/ESXi](#)

You may decide to add each guest system to NetCrunch automatically. Each guest will be automatically monitored according to defined Automatic Monitoring Packs.

The option is disabled by default.

When the option is enabled, and you want to remove a VM guest from Atlas you need to set a specific exclusion in the Network View. Otherwise the machine will be added over and over again.

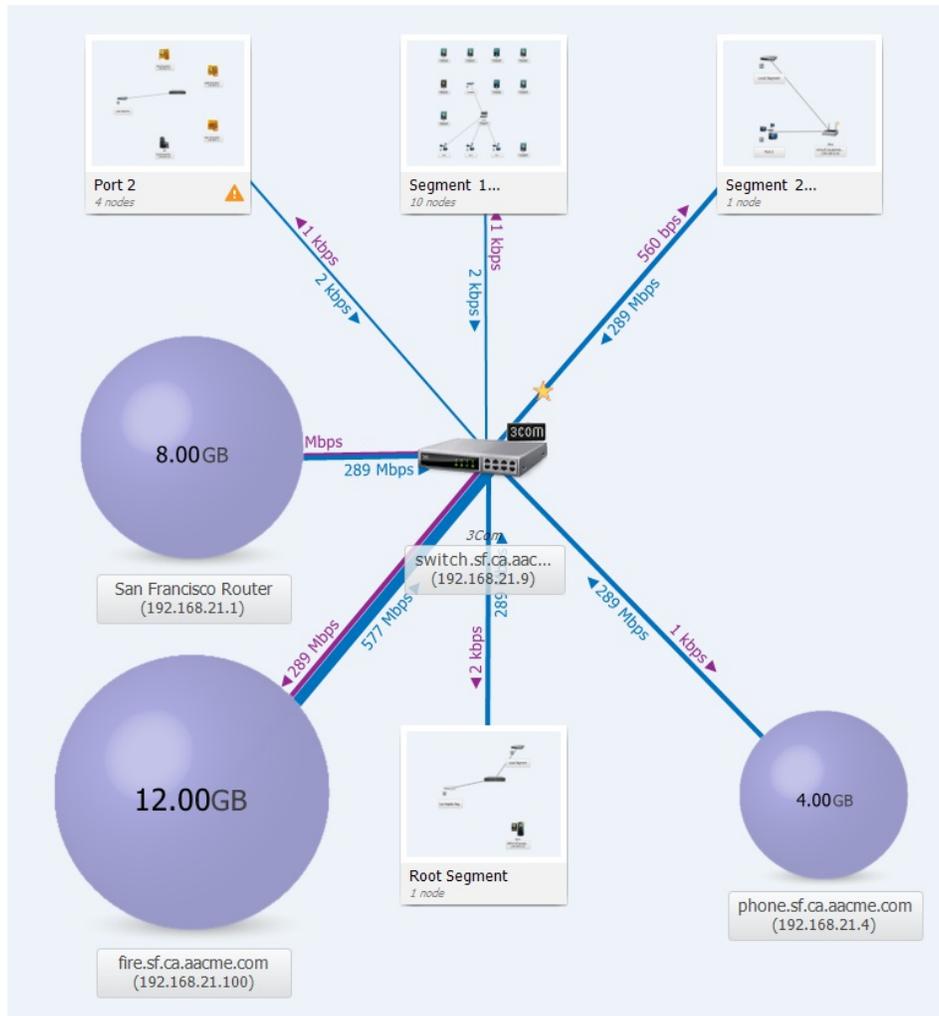
Network Traffic Monitoring

NetCrunch supports many flow protocols as application monitoring using Cisco NBAR

Netcrunch supports two technologies that allow network traffic monitoring. The first one gathers information from switches about traffic on particular ports. The second one can collect flow data from routers and switches.

Monitoring Traffic on Switch Ports

As switches collect statistics about traffic on their ports, the information is available in Physical Segments views. These views can give you an overview of traffic between two switches or between the switch and some server.



You can see aggregated traffic information in the last hour or in last 24 hours. When you select a link on the map, then you can open the Connection Status details and the Connection Traffic History.

Monitoring Traffic with NetFlow

Introduction

There are many flow export formats on the market today. Netflow is Cisco trademark (aka cflow), but there are many similar protocols like jFlow, rFlow, NetStream, AppFlow and sFlow.

For the NetFlow suite of protocols we most often see version 5 (supported by the majority of devices), some combined v5/v7 (the Catalysts), and some version 9 of the newer devices.

Supported Flow Technologies

NetCrunch supports following flow protocols: - NetFlow v1, v5, v8, v9 and IPFix, - NetStream - CFlow - AppFlow - rFlow

NetCrunch collects and analyses flows for aggregation in the 15 minutes and 1 hour ranges. This gives you both the ability to analyze data in a short period time, and to store long term performance trends.

Currently, NetCrunch supports single flow aggregation, so it can receive data from multiple flow sources. However, they are aggregated together on the single dashboard.

sFlow Disclaimer

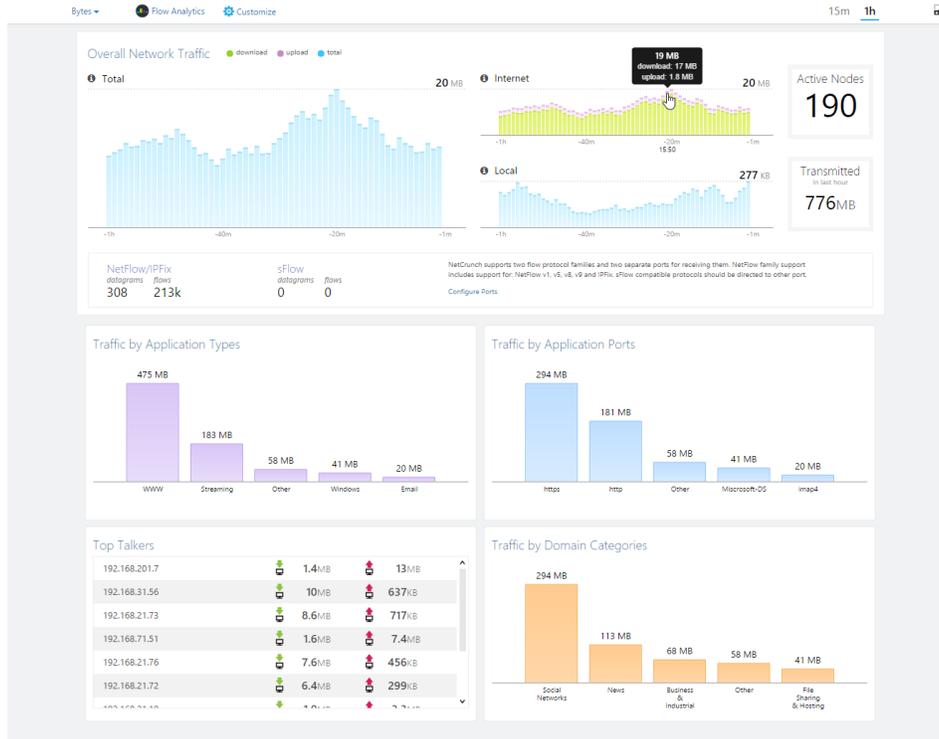
NetCrunch is capable of receiving sFlow, but the statistical nature of the protocol requires different processing and aggregating data over a much longer time period.

NetFlow Views

NetFlow Dashboard

[Dashboards](#) > [Flow](#)

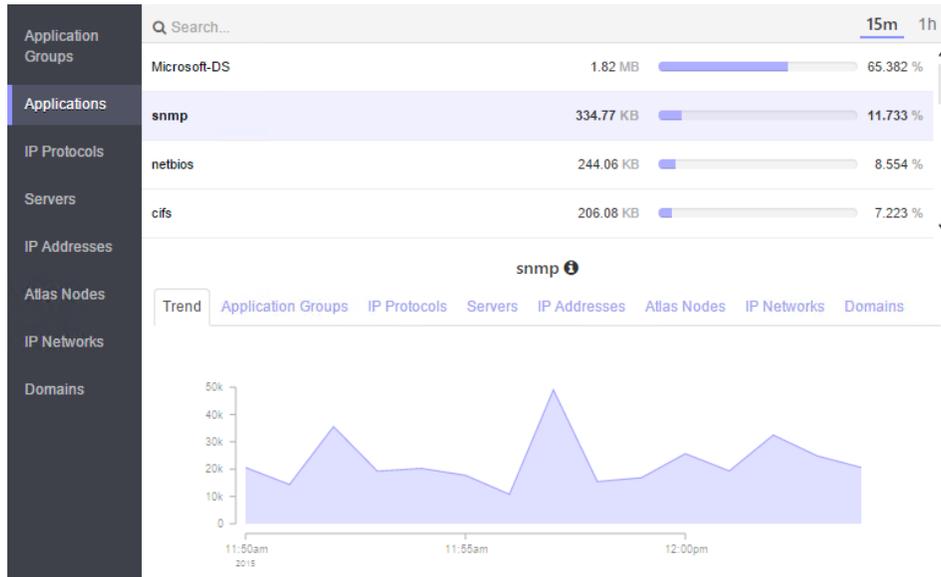
NetCrunch shows global flow traffic statistic on the top Atlas Dashboard. When you click on any chart bar, you can see the details of traffic for a selected element.



The view shows current aggregated statistic from all flow sources sending data to NetCrunch.

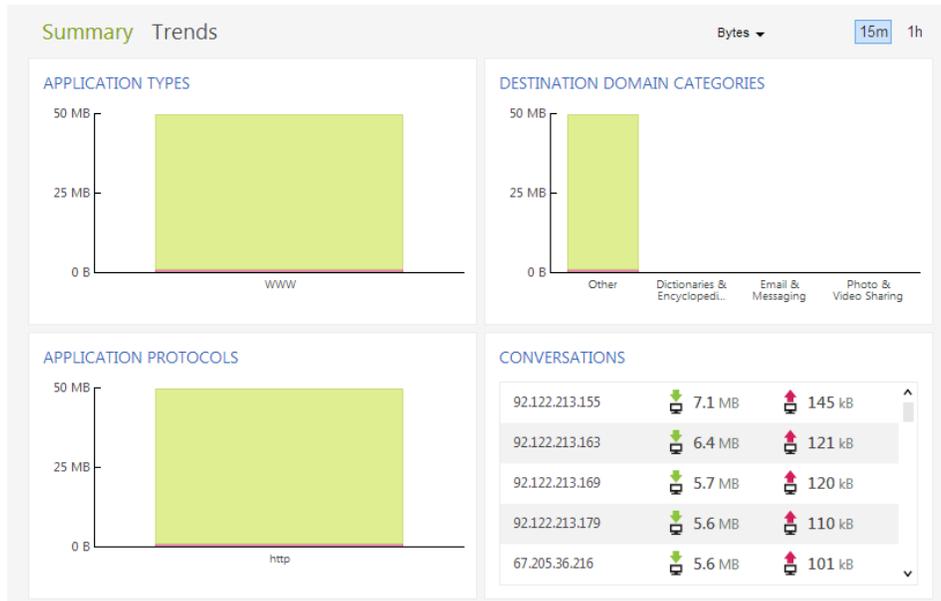
Flow Analytics

NetCrunch allows you to analyze traffic using various criteria. The program allows you to create custom application definitions and also supports Cisco NBAR technology for application monitoring.



The program also allows creating custom application definitions based on protocol and ports.

Node Traffic



Monitoring SNMP Devices

NetCrunch fully supports SNMP including support for v3 and MIB compiler.

SNMP was developed on UNIX back in 1988 - so it's quite mature technology now. Despite of various new protocols, it's probably the most widely implemented management protocol today. SNMP is defined in RFC (Request For Comments) by IETF (Internet Engineering Task Force) and is used everywhere: server, workstation, router, firewall, switch, hub, printer, IP phone, appliance...

SNMP Versions

Today, most devices support SNMPv2c. There are also agents for operating systems, but unless SNMPv3 is used, they can be exposed to a security hole. Top hardware devices usually support SNMP v3, which adds more security to the protocol (authentication and encryption). NetCrunch supports all SNMP versions: SNMP v1, SNMP v2c and SNMP v3, including decoding traps. NetCrunch SNMP implementation supports SNMPv3 following encryption algorithms: DES, 3DES, AES128, AES 192 and AES 256.

SNMP Profiles

NetCrunch introduces the idea of SNMP profiles. The profiles allow you to use the same SNMP security settings for multiple nodes. SNMP Profiles encapsulate settings necessary to communicate with the particular SNMP Agent.

Profiles define SNMP protocol version and information related to protocol version and security settings: the community string (SNMP v2); or authentication user, password, and encryption to be used (SNMPv3).

Additionally the profile allows specifying different protocols to be used for reading and writing operations. For example, you can setup reading operations to use SNMPv2 and disable writing.

In order to receive and decode SNMPv3 traps, the profile must contain a *notification* section.

Monitoring SNMP Variables

To access SNMP data on particular node, you must make it SNMP enabled by [Node Settings ▶ SNMP](#). In order to create an alert on the SNMP counter value, you need to setup Event Trigger alert.

Setting Alerts on Numerical Values

Select the node, and open [Node Settings ▶ Monitoring](#) and you'll see SNMP section and SNMP trap sensor below. If SNMP is not configured, press [SNMP](#) button at the top right corner. You can now select one of [Event Triggers for Counters](#) and select the SNMP variable as the counter.

We have following options for SNMP counters:

MIB Database

In order to select an SNMP variable you can browse SNMP MIB data. This requires MIB data to be compiled and placed into the NetCrunch MIB database first. NetCrunch includes a database of 2000 popular MIBs. There are websites maintaining on-line databases of 7000 to 12000 MIBs. You can download them and use NetCrunch MIB compiler to extend your MIB database.

Predefined Counters

The second option is to choose from predefined SNMP counters - this is some kind of common counters used. You can extend this list for later use to avoid tedious MIB tree browsing.

Counter OID

This option allows you to enter an arbitrary OID number - you might obtain it just from the device using a simple tool like MIB walker. This allows to access SNMP data even if you do not have the MIB for the device.

Setting Alerts on Text Values

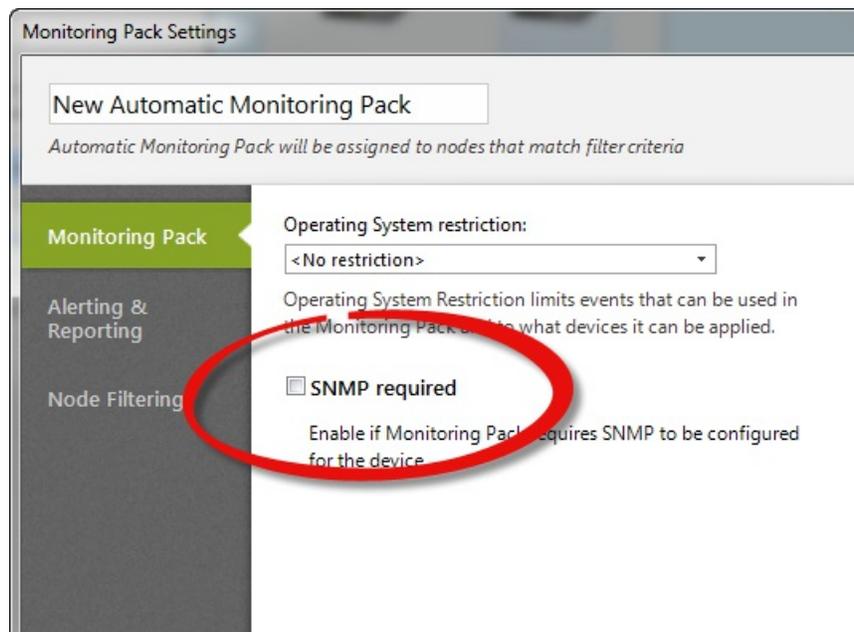
Besides numerical values, you can also monitor text values returned by SNMP agent. In order to this you have to select and enter OID or select the object from MIB database. Then you can check following conditions:

- Value equals to given value
- Value not equals to given value
- Value contains given text
- Value does not contains given text
- Value matches regular expression
- Value not matches regular expression
- Value changed
- Value unchanged

Creating of Monitoring Pack for SNMP

[Monitoring > Monitoring Packs & Policies](#)

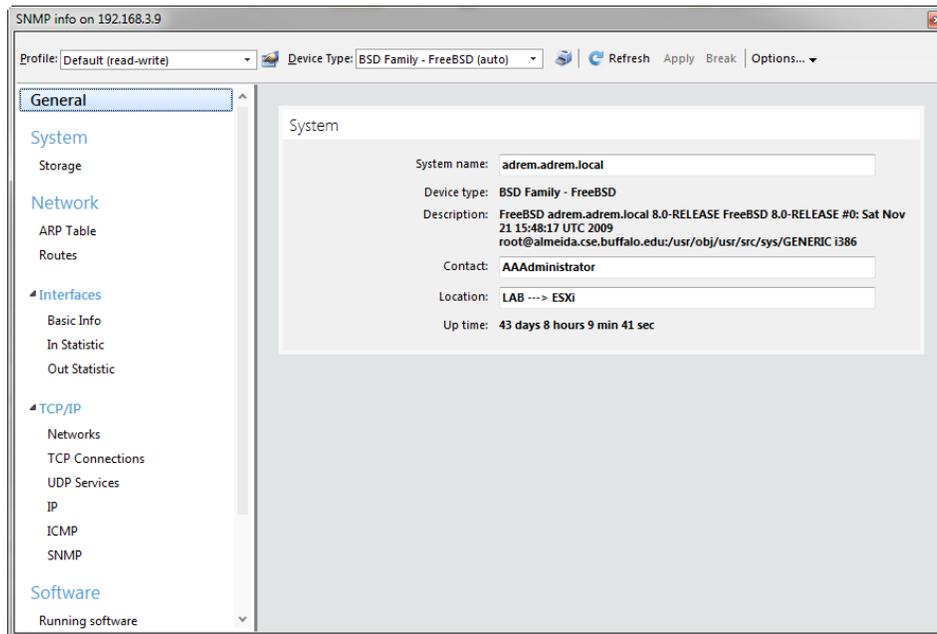
When creating a new monitoring pack remember to check SNMP required option.



SNMP Views

SNMP only describes the variables and get/set operations, and as such browsing OID MIB tree is rather difficult. SNMP tables often refer to other tables and when raw data is not readable. NetCrunch SNMP Views allow for creating tables and forms which are more human readable, allowing reading and entering SNMP data.

Additionally, views are automatically managed according to device type and supported MIB.



Receiving SNMP Traps

Check if the SNMP trap listener is enabled. [Tools > Options > Monitoring > SNMP Traps](#)

[Node Settings > Monitoring](#) or [Monitoring > Monitoring Packs & Policies](#)

NetCrunch allows receiving SNMP traps of version 1, SNMPv2c and SNMPv3 including encryption. In order to define the alert for the trap, the node must have an SNMP profile assigned. For SNMPv3 traps, the profile must have defined notification part containing username, password and encryption.

Forwarding SNMP Traps

[Tools > Options > Monitoring > SNMP Traps](#)

After receiving an SNMP trap you can forward it as is to other SNMP manager.

MIB Compiler

NetCrunch MIB compiler allows you to extend NetCrunch MIB database used for selecting SNMP traps and variables during configuration and in the process of resolving OID to names for incoming SNMP data (traps).

It's an advanced multi-pass compiler with the ability to set up module name aliases in order to help compile otherwise incompatible modules.

Us...	Module Name	Vendor	Enterprise	Last Updated	Variables	Traps
	A3COM0073IGMP-SNOOP		3Com (43)		14	0
	A3COM0304-RESILIENTLINKS		3Com (43)		24	2
	A3Com-products-MIB		3Com (43)		0	0
	A3COM-SWITCHING-SYSTEMS-BRIDGE-MIB		3Com (43)		87	4
	A3COM-SWITCHING-SYSTEMS-FD-DI-MIB		3Com (43)		50	11
	A3COM-SWITCHING-SYSTEMS-FILE-TRANSFER-MIB		3Com (43)		21	0
	A3COM-SWITCHING-SYSTEMS-FILTER-MIB		3Com (43)		46	0
Total count: 2489					192580	7968

MIB database last update: 11/9/2012 4:42:19 PM

Windows Monitoring Setup

Reading this topic will make your Windows monitoring experience much better.

NetCrunch can monitor Microsoft Windows systems without installing additional agents. However, due to tightened security rules, remote monitoring is possible only after initial configuration, which depends on your Windows environment.

MONITORING SERVER

NetCrunch Server can be installed on Windows Server 2008 R2 or Windows Server 2012 R2. If you manage most of the servers by Active Directory, it is better to install NetCrunch on a machine within an Active Directory domain. This just makes configuration much easier.

MONITORED SYSTEMS

SERVERS

Most server systems come with the firewall being enabled, which blocks remote administration. This is the first thing you need to change. It could be done either from Active Directory Group Policies, or manually one by one. We suggest using a simple script.

Download it here: www.adremsoft.com/download/SetWinForNC.zip.

WORKSTATIONS

If you manage your workstations by Active Directory, preparing them for monitoring will be the same as for the servers (by Active Directory Group Policies or using the script).

Monitoring of workstations in Workgroups is a little harder to configure, because starting from Windows Vista, all later systems use UAC (User Account Control). UAC does not allow remote connections to inherit administration rights from the local Administrators group. In this case, you can choose to use built-in local Administrator account, or create a new account and manually assign necessary rights directly to this

account.

CONFIGURATION STEPS OVERVIEW

Setting Access Rights

NetCrunch needs a user account for monitoring which has proper access rights to DCOM, WMI (root\cimV2) and (Read Access) to the registry key (HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib). The easiest way you can do it is by adding this user to the local Administrators group.

Setting Firewall Rules

Firewall rules must allow traffic of RPC, Performance Monitoring, Named Pipes and WMI.

Enabling PerfMon monitoring

Remote Registry service must be running and its startup type should be set to Automatic.

Disable UAC remote restrictions

User Account Control remote restrictions need to be disabled for non-Domain servers.

This requires changing value of a registry key:

KEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Policies\System\LocalAccountTokenFilterPolicy

CONFIGURING ACTIVE DIRECTORY DOMAIN

Note:

The procedure below requires a working knowledge of Active Directory Users and Computers and Group Policy Management Administrative Tools

If you manage most of the servers by Active Directory, the best solution is installing NetCrunch on a server in the Active Directory domain, and creating a dedicated user for monitoring. If you have not yet created such user in your Active Directory, you should abort your NetCrunch installation now and configure your Active Directory first, allowing time for the user to propagate across your machines. You can start the NetCrunch installation again, after your configuration has been propagated to all servers – it takes approximately 2 hours.

This is done in order for NetCrunch to be able to discover all servers in AD and automatically setup monitoring for them. Other servers in untrusted domains or workgroups can be configured separately (see Configuration of Separate Windows Server chapter).

SETTING ACCESS RIGHTS

STEP 1 - CREATE USER FOR MONITORING

Create Active Directory user account (for example nc-mon-user) that will be used by NetCrunch Server for monitoring. You will be asked later for this user credentials during the NetCrunch installation.

STEP 2 - SET UP RIGHTS FOR THE USER

The user account needs administrative rights to all monitored Windows computers (including the server where NetCrunch Server is installed). There are two different ways to accomplish this, depending on your Active Directory architecture and your needs:

IF IN SINGLE DOMAIN WHERE YOU WANT TO MONITOR ALL MACHINES

Add created user account to predefined Domain Admins Active Directory group.

IF MULTIPLE TRUSTED DOMAINS OR ONLY SUBSET OF COMPUTERS NEEDS MONITORING

You need to use Group Policy to modify local Administrators groups (on each monitored Windows machine).

a) Create Active Directory group named *Monitoring Users* and add previously created user account (*nc-mon-user*) to this group.

Note:

In the multi - domain forest, default Active Directory group scope (which is Global) should be sufficient for this group, because global groups can be used to assign permissions to resources in any domain in a forest.

b) Create a new Group Policy Object (GPO) and name it, for example *Local Administrators* group membership for NetCrunch.

c) Create the rule for Monitoring Users group membership.

Go to: [Computer Configuration](#) ▶ [Policies](#) ▶ [Windows Settings](#) ▶ [Security Settings](#) ▶ [Restricted Groups](#)

and add Monitoring Users to local Administrators group using section 'This group is a member of'

d) Link Local Administrators group membership for NetCrunch GPO to appropriate Organization Unit(s) (OU) in your Active Directory domain(s).

SETTING FIREWALL RULES

Create a new Group Policy Object and name it, for example, "Windows Firewall rules for monitoring by NetCrunch".

Use two different branches in GPO to configure both built-in firewall types in Windows machines which you want to monitor:

a. For XP and Server 2003 R2,

Go to:

➤ [Computer Configuration](#) ▶ [Administrative Templates](#) ▶ [Network](#) ▶ [Network Connections](#) ▶ [Windows Firewall](#) ▶ [Domain Profile](#)

and set these settings to Enabled:

- Windows Firewall: Allow inbound file and printer sharing exception
- Windows Firewall: Allow inbound remote administration exception

b. For Vista /7/8 and Server 2008/2008 R2/2012,

Go to:

➤ [Computer Configuration](#) ▶ [Policies](#) ▶ [Windows Settings](#) ▶ [Security Settings](#) ▶ [Windows Firewall with Advanced Security](#)

and add these rules to Inbound Rules, choosing them from a predefined list:

- File and Printer Sharing
- Remote Administration

Link Windows Firewall rules for monitoring by NetCrunch GPO to appropriate Organization Unit(s) (OU) in your Active Directory domain(s).

For a security reasons, it is recommended to customize remote administration rules to narrow the list of allowed IP addresses to the address of your NetCrunch Server only.

ENABLING PERFMON MONITORING

Create a new Group Policy Object and name it, for example, *Windows services for monitoring by NetCrunch*.

Setup *Remote Registry* service.

Go to:

➤ [Computer Configuration](#) ▶ [Policies](#) ▶ [Windows Settings](#) ▶ [Security Settings](#) ▶ [System Services](#) and set *Remote Registry* Windows service startup mode to Automatic.

Link *Windows services for monitoring by NetCrunch* GPO to appropriate Organization Unit(s) (OU) in your Active Directory domain(s).

Right after the policy refresh, the service should immediately start on every computer.

Note:

By default, Windows built-in firewall doesn't block outgoing traffic – if you have changed this behavior, add rules with the same names from predefined list to *Outbound Rules*.

CONFIGURING OF SEPARATE WINDOWS SERVERS

SETTING ACCESS RIGHTS

Create nc-mon-user account using shell commands and add it to local Administrators group.

```
net user /add nc-mon-user <Password>
net localgroup Administrators /add nc-mon-user
```

SETTING FIREWALL RULES

For Windows Server 2003 and Server 2003 R2:

```
netsh firewall set service type=fileandprint scope=all profile=all
netsh firewall set service type=remoteadmin scope=all profile=all
```

For Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012

you can create rule for IP address of NetCrunch server only.

```
netsh advfirewall firewall add rule name="NC-Mon" dir=in action=allow remoteip=192.168.1.100
netsh advfirewall firewall add rule name="NC-Mon" dir=out action=allow remoteip=192.168.1.100
```

ENABLING PERFMON MONITORING

Setup Remote Registry service startup and start the service.

```
WMIC SERVICE where name="RemoteRegistry" call ChangeStartMode StartMode=Automatic
WMIC SERVICE where name="RemoteRegistry" call StartService
```

DISABLING UAC REMOTE RESTRICTIONS

Modify UAC behavior for Windows Server 2008/2008 R2, and Windows Server 2012

<http://support.microsoft.com/kb/951016>

COMPLETE SCRIPT CAN BE DOWNLOADED FROM:

Summary of Technologies Used by NetCrunch

Windows technologies have been built layer by layer, one on top of another. For example, RPC is working on top of the Named Pipes, Remote Registry needs RPC, and WMI is using DCOM which is using also RPC for communication. Everything needs proper firewall and security settings. Here is the short list of technologies used by NetCrunch that need proper configuration:

RPC & Named Pipes – (Needs enabling File Sharing, firewall settings)

Remote Registry – (Needs firewall settings, and Remote Registry service running)

WMI & DCOM – (Needs Firewall settings, DCOM & WMI security settings)

It is simple in the case when the user designated for monitoring is a member of local Administrators group – as described in this document. This is the simplest way to configure servers for monitoring, but not the most

secure. In case when security is a significant concern, it is possible to set up exact rights to monitor the account.

Custom Monitoring with NC Open Monitor

Read [howto extend monitoring capabilities of NetCrunch. Run programs or scripts or send data from external systems to NetCrunch.](#)

What NetCrunch Open Monitor Can do?

NetCrunch Open Monitor allows for delivering external numerical data to NetCrunch. As a result, you can set alerts (thresholds), create reports, and display trend charts. It's ideal to monitor external devices or applications using a custom interface. The NetCrunch Open Monitor engine is not intended to extend monitoring of multiple nodes - it's a simple way to provide data by a script or a program that runs on NetCrunch server. Therefore, counters delivered through Open Monitor are not bound to any node and appear in the Global Data container.

If you want to send data or statuses related to particular node, you should use [Generic Agent Sensor](#) instead.

Data Format

Natively, NetCrunch accepts data in two most popular data formats, but formats can also be extended through JavaScript transform script. For instance, the latest version of NetCrunch also added support for CSV formatted data through such script; you can read about the transform script at the end of this article.

XML

```
<ncopenmon>
  <counters>
    <counter path="xobj/cnt.1">123</counter>
    <counter path="xobj/cnt.2">245</counter>
  </counters>
</ncopenmon>
```

JSON

```
{
  "counters": {
    "crm.emails-in-1h": 10.2,
    "crm.emails-out-1h": 231
  }
}
```

CSV

object, counter, instance, value

```
processor,% Utilization,_total,20
memory,private bytes,,23523578
```

First line is a header (ignored). Each next line is a counter.

Sending Data to NetCrunch

NetCrunch can receive data in JSON or XML format. Before you send any data you must create an API key for your requests. It's a type of simple identification that assures the server that data is coming from a legitimate sender. In order to create an API key, please go to

➤ [Tools > Options > Monitoring > NetCrunch Open Monitor](#) .

The simplest tool you can set to send a request to NetCrunch is cUrl open source project available for almost any platform. You can find it at: curl.haxx.se

Retention Time

Because NetCrunch does not know how often you are going to send data to it, you need to specify a "retention time" for data after which data will expire and will be cleared out from the memory.

Note:

This can be even used for a kind of heartbeat monitoring. If you set the *retention time* to 1 minute, then you can expect data every minute. You can also set an alert on missing data. Missing data means that NetCrunch didn't receive data within a given time frame, which is the *retention time* plus one minute.

Protocol

There are only a few commands you can use. One for sending few counters just by encoding parameters in the URL, and the second you can use for bulk updates.

```
GET /ncintf/rest/1/openmon/counter
```

You can send a counter using GET request and parameters in the URL.

For Example

```
<server-address>/ncintf/rest/1/openmon/counter?crm%2Fday-orders=121&crm%2F
```

```
GET /ncintf/rest/1/openmon/incCounter
```

This request will increment counter by give value;

```
<server-address>/ncintf/rest/1/openmon/incCounter?crm%2Fday-orders=1&crm%2
```

```
GET /ncintf/rest/1/openmon/decCounter
```

This request will increment counter by give value;

```
<server-address>/ncintf/rest/1/openmon/decCounter?crm%2Fday-orders=1&crm%2
```

```
POST /ncintf/rest/1/openmon/incCounter
```

This request will increment counter by give value. Data should be send as JSON. For example:

```
{
  "apikey": "MDAwMDAwMDAxM0FDRDhDQj==",
  "counters": {
    "crm/day-orders" : 1,
    "crm/day-emails" : 1,
  }
}
```

POST /ncintf/rest/1/openmon/decCounter

This request will decrement counter by give value;

POST /ncintf/rest/1/openmon/update

For Example

```
{
  "retain": 1,
  "apikey": "MDAwMDAwMDAxM0FDRDhDQj==",
  "counters": {
    "PBX/line status.0" : 1,
    "PBX/line status.1" : 0,
    "PBX/version" : "Mock Phone System 1.0"
  }
}
```

Remember to set MIME content header "application/json".

As you see in the above example, you can pass any type of data as a counter. However, only numeric counters can be used in threshold events and trends. Other values can be used in the *Open Monitor Data* window only.

Running Programs or Scripts

➤ [Tools > Options > Monitoring > NetCrunch Open Monitor](#)

Click

Choose File - Read data from disk.

Specify program or script to be executed. (*scripts can be JScript or VBScript with .js or .vbs extensions*)

Set Data Format, which your program will return.

Depending on your program: you can keep the "Data file name" field empty if the program writes data to standard output stream (stdout), or add a file name where data will be located.

Set a scheduling time (default every minute).

Please remember to specify full paths accessible by NetCrunch Server service. As the service is running on the account defined during installation, scripts must be accessible and they inherit NetCrunch Server rights.

Retrieving data from Files

➤ [Tools > Options > Monitoring > NetCrunch Open Monitor](#)

Click [Add Source](#) .

Choose `File` - Read data from disk.

You can keep the program name blank (data can be written by some service or externally scheduled program).

Set the data format .

Set a path to the data file.

Set a reading time (default every minute).

Please remember to specify full paths accessible by NetCrunch Server service. As the service is running on the account, defined during installation, scripts must be accessible and they inherit NetCrunch Server rights.

Retrieving from Web service

➤ [Tools > Options > Monitoring > NetCrunch Open Monitor](#)

Click [Add Source](#)

Choose `Web`- Read data from REST HTTP/s service.

Specify URL.

Set format or keep `AUTO`, then the format will be automatically determined upon the received MIME

header.

Set a reading time (default every minute)

You can check sample services at:

<http://www.adremsoft.com/dev/open-mon-test.php>

or

<http://www.adremsoft.com/dev/open-mon-test-xml.php>

Technical Constraints

Counters are global, if want to bind them to a node use [Generic Agent Data Sensor](#). A counter name can be in the form: Object/Counter/Instance or Object.Counter.Instance. An instance is optional.

Data Transform Scripts

Scripts can be written in JavaScript, which is executed in WSH environment with *E5Shim.js* to support ECMA5 extensions.

To add a new format:

Create script including function:

```
function transform(options, data) {...}
returning expected JavaScript object.
```

Place the script in the NetCrunch Server program directory:

Usually: 'c:\Program Files (x86)\AdRem\NetCrunch\Server\9.0\External\openmon\formats'

The name of the script will be used as the format name.

Adding Open Monitor Source

➤ [Tools > NC Open Monitor Data](#) or ➤ [Tools > Options > Monitoring > NetCrunch Open Monitor](#)

In this window you can define sources for NC Open Monitor Engine.

Viewing Open Monitor Data

➤ [Tools > NC Open Monitor Data](#)

In the Open Monitor Global Data you can see current data received or retrieved by the NC Open Monitor. The list contains all data, including non-numeric.

Setting Alerts on NC Open Monitor Data

To set an alert on data retrieved/received by NC Open Monitor, add thresholds to Open Monitor Monitoring Pack located at the bottom of the list in Global group.

Note:

By default, NC Open Monitor creates counters upon existing data. After you configure NC Open Monitor source and get some data, you will be able to see the counters.

To enter counter manually:

Click in Add Open Monitor Counter window.

Monitoring Files and Folders

NetCrunch allows monitoring files, folders and text logs

File Sensors

NetCrunch contains three sensors that allow monitoring of various files. The sensors can access remote files using Windows file sharing, FTP/s or HTTP/s protocols. You can monitor file changes, file content, and text logs. All sensors have the same monitoring capabilities.

Windows File Options

The program can process files using following encoding: *UTF-8, ASCII or Base64*.

Remote File Options (FTP/s)

- File encoding: *UTF-8, ASCII or Base64*.
- Transport: *FTP, FTPS, SFTP*
- Port
- Timeout
- Passive mode

Web File HTTPSs

- Type: *HTTP, HTTPS*
- Port
- Timeout

Critical File Events

- Authentication Error
- Connection Error
- File does not exist

Warning File Events

- File exists
- File is empty
- File is not empty
- File modified
- File not updated in given time (file age)
- File read error
- File updated too often (file age)

Monitoring File Content & Text Logs

Text Search event monitor can search for some text using plain text pattern or use regular expression. Incremental search will monitor only new occurrences in file.

Text Log Entry event monitor tries to match lines, and if new line is found, alert is triggered. Search pattern can be plain text or regular expression. If multiple lines matching alert are detected, they can be grouped into a single alert.

Example: Alerting on cron log

For example, we want to know when there is at least 1 job ran on exit. So let's see a sample cron log:

```
Nov  5 01:01:01 localhost CROND[22826]: (root) CMD (run-parts /etc/cron.ho
Nov  5 01:01:01 localhost run-parts(/etc/cron.hourly)[22826]: starting 0an
Nov  5 01:01:01 localhost anacron[22836]: Anacron started on 2015-11-05
Nov  5 01:01:01 localhost anacron[22836]: Normal exit (1 job run)
Nov  5 01:01:01 localhost run-parts(/etc/cron.hourly)[22838]: finished 0an
```

We can define following expression to a more specific line:

```
anacron.*Normal exit \([1-9]*
```

Performance Metrics

- % Availability
- Size

File Sensor report

When you add a sensor to a node, a new report is also added, which collects sensor performance metrics.

Folder Sensors

Folder sensor allows observing folder content with a list of files. The sensor can trigger alert when file is deleted or removed, and on various other conditions.

Remote Folder (FTP) Options

- File encoding: *UTF-8, ASCII* or *Base64*.
- Transport: *FTP, FTPS, SFTP*
- Port
- Timeout
- Passive mode

Critical Folder Events

- Authentication Error
- Connection Error
- Folder access error
- Folder does not exist

Warning Folder Events

- Folder empty
- Folder exists
- Folder not empty

Monitoring Folder Content

You can check list of files matching give file mask.

- Files exist
- Files not exist
- New file added
- File removed

Performance Metrics

- % Availability
- File Count

Monitoring Web Pages & HTTP/HTTPS Requests

NetCrunch can monitor requests, pages and files on the web.

➤ [Node Settings](#) ▶ [Monitoring](#) ▶ [Add Monitoring Sensor](#)

NetCrunch includes two sensors for web monitoring.

Web Page Sensor (HTTP/s)

The sensor renders page like a browser, it loads all resources and runs scripts. It's intended for monitoring modern pages or applications. Also, it supports standard login and custom login forms.

Options

- Username and password
- Connection Type and Port
- Load images
- Run Javascript
- Allow redirect
- Timeout

Alerts:

- Page size or load time
- Page content change
- Alert if text is present or missing
- Page does not exist
- Page load error
- Page resource load error
- Page authentication error

Performance Metrics:

- % Availability - Web page availability - 100% if no page loading error occurs, 0% otherwise.
- HTTP Status Code - status code returned by the server
- JS Errors - JavaScript execution exception count
- Load Time - Total page loading time (ms)
- Main Frame Body Size - Length of the main frame content (bytes)

- Resource Count - Page resource count.
- Resources Error Count - Page resource loading error (timeout) count
- Total Size - Total length of all page resources (bytes)

Report

Default report added collects: *% Availability, Load Time , Total Size, Resource Count, Resource Error Count*. These parameters also available in [@trend-viewer](#)

Basic HTTP/s Request

This sensor sends a single request and can alert on response code or checking response data. It can send GET, HEAD and POST requests.

Alerts:

- HTTP request timeout
- HTTP response code is not OK

Performance Metrics:

- % Availability - HTTP server availability
- Content Length - Response content length (bytes)
- Response Length - Full HTTP response length in bytes (includes header size).
- Response Time - HTTP response time (ms)

Report

By default, sensor adds report containing: *% Availability, Response Time* and *Content Length* charts. These metrics will be collected and available as a report or through [@trend-viewer](#)

HTTP/s File Sensor

Check remote file content, authentication parameters, monitor remote text logs, file size or change time, presence and more. See [Monitoring Files and Folders](#)

Cisco IP SLA operations

NetCrunch supports Cisco SLA technology

Sensor

[Node Settings](#) ▶ [Monitoring](#) ▶ [Add IP SLA Operation](#)

This sensor allows for monitoring of the status of IP SLA operations on the Cisco devices. When you are

adding the sensor, you need to select operation previously defined on a Cisco device. Operations are grouped by protocol type.

Program can alert on following conditions:

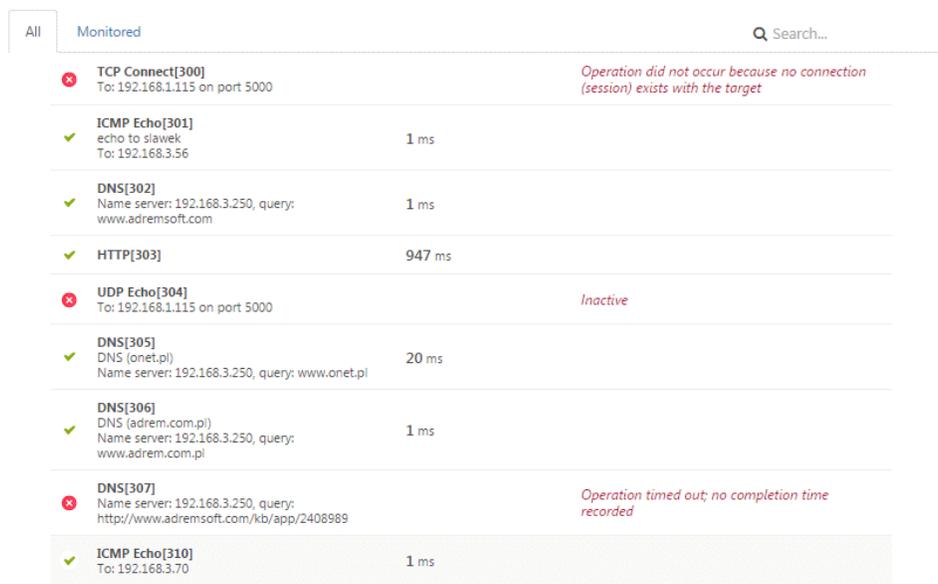
- Operation completed with error
- Operation is inactive

additionally you can set performance triggers (thresholds) on following metrics:

- % Availability - A sense code for the completion status of the latest RTT operation.
- Completion Status
- RTT - Round trip time (ms).

Status

In node status window NetCrunch shows status of all IP SLA operations defined on the Cisco device.



The screenshot shows a table of IP SLA operations. The table has columns for status (indicated by a green checkmark or red 'x'), operation name and details, and completion time or status. The operations listed are:

Status	Operation Name & Details	Completion Time / Status
✗	TCP Connect[300] To: 192.168.1.115 on port 5000	Operation did not occur because no connection (session) exists with the target
✓	ICMP Echo[301] echo to slawek To: 192.168.3.56	1 ms
✓	DNS[302] Name server: 192.168.3.250, query: www.adremsoft.com	1 ms
✓	HTTP[303]	947 ms
✗	UDP Echo[304] To: 192.168.1.115 on port 5000	Inactive
✓	DNS[305] DNS (onet.pl) Name server: 192.168.3.250, query: www.onet.pl	20 ms
✓	DNS[306] DNS (adrem.com.pl) Name server: 192.168.3.250, query: www.adrem.com.pl	1 ms
✗	DNS[307] Name server: 192.168.3.250, query: http://www.adremsoft.com/kb/app/2408989	Operation timed out; no completion time recorded
✓	ICMP Echo[310] To: 192.168.3.70	1 ms

NetCrunch Self Monitor

[Read about how NetCrunch monitors itself.](#)

NetCrunch Server is very much like any complex application with multiple processes, lots of data processing, and high demand for storage. In fact, NetCrunch has multiple logs, and monitors thousands of its parameters. These parameters are important when diagnosing potential performance problems. Be aware: the program is always limited to the hardware capabilities it's running on.

So, NetCrunch Self Monitor is the monitor which watches the NetCrunch Server.

Some programs have a tendency to simply hide problems, which is troublesome, mainly because it creates the illusion that something is working when in actuality it's not. It's hard to detect and impossible to solve when there is no visible problem. Most problems are caused by overloading - NetCrunch Server overloading or monitored device overloading. Many Cisco devices have protection against too many monitoring requests sent to the device. In many cases, the solution is simple: increasing monitoring time or timeouts.

Status of NetCrunch server is available on [NetCrunch Status](#) dashboard.

Alerts

➤ [Monitoring](#) > [Monitoring Packs & Policies](#) > [Global](#) > [NetCrunch Self Monitor](#)

Global monitoring pack contains default alerting settings for this monitor and they trigger *Default* action.

NetCrunch alerts on:

- (Critical) SNMP monitoring heavily overloaded
- (Warning) SNMP monitoring overloaded
- (Warning) Network Services monitoring overloaded
- (Critical) Database overloaded. Some events have been dropped
- (Warning) Writing to database is slow
- (Critical) Low disk space (< 512MB). Please free up some space on NetCrunch data disk.
- (Warning) Low disk space (< 1GB. Please free up some space on NetCrunch data disk.
- (Critical) The system is low on memory
- (Warning) The system is low on memory
- (Critical) Server Memory usage is too high. Please restart NetCrunch Service.
- (Critical) Virtual memory for NetCrunch Server fragmented. Performance might be negatively affected. Machine restart is required.
- (Warning) Physical Segments not refreshed on time
- (Warning) Monitoring overload - threads limit reached
- (Warning) Database size is close to maximum capacity
- (Warning) System is running for too long
- (Warning) Server handle usage too high
- (Warning) Atlas backup has been executed more than two day ago
- (Critical) Last backup failed
- (Warning) Limit of licensed nodes nearly reached
- (Informational) New version is available
- (Warning) NetCrunch maintenance subscription is about to expire
- (Critical) NetCrunch trial period is about to expire

Monitoring DNS Health

DNS service is most vital part of every infrastructure. NetCrunch allows checking for DNS query results and detects unwanted changes.

DNS sensors allow you to check if the DNS responses are valid. This helps identify problems in DNS or to check whether records has been altered.

DNS Query Sensor

[➤ Node Setting](#) ▶ [Monitoring](#) ▶ [Add Sensor](#) ▶ [DNS Query](#)

This sensor allows you to send query to a given DNS server (you should add it on DNS server node). You can enter a name to check (a domain name) and you can define alerts to check if DNS records match expected results. The sensor will alert you if DNS server is not responding. In addition, you can measure and alert on Response Time.

You can match value of following record types:

- Resolve IPv4 address (A)
- Resolve IPv6 address (AAAA)
- Resolve canonical name (CNAME)
- Mail exchange record (MX)
- Location record (LOC)
- Service locator (SRV)
- Name server record (NS)
- Text record (TXT)

Report

Default report collects performance data for the sensor:

- % Availability
- Response Time

Reverse DNS Query Sensor

[➤ Node Setting](#) ▶ [Monitoring](#) ▶ [Add Sensor](#) ▶ [DNS Query](#)

This sensor is just opposite of the previous one. Instead of looking up address from a name, we can look if the given address has a proper reverse record which corresponds to expected names.

You can query IPv4 and IPv6 addresses. As in previous DNS query sensor, you can check responses for all DNS record types - but now you can check if the response matches proper name as well.

- Resolve IPv4 address (A)
- Resolve IPv6 address (AAAA)
- Resolve canonical name (CNAME)
- Mail exchange record (MX)
- Location record (LOC)
- Service locator (SRV)
- Name server record (NS)
- Text record (TXT)

Apache Web Server Monitoring

Monitor most popular Apache web server.

[Node Settings](#) ▶ [Add Sensor](#) ▶ [Apache Server](#)

NetCrunch allows monitoring performance of Apache web servers. The Apache sensor allows you to monitor various performance metrics grouped in objects like Country, Summary and Virtual Host.

Summary Counters

- Bytes per Second
- Client Count
- Process Count
- Requests per Second
- Total Requests
- Total Transfer
- Up Time

Country and Virtual Host Counters

- Avg. CPU Timer
- Avg. Request Elapsed Time
- Avg. Request Processing Time
- Client Count
- Max CPU Count

- Max Request Elapsed Time
- Max Request Processing Time
- Transfer

Monitoring Mail Services

NetCrunch allows monitoring mail content, mailboxes and checking basic mail server functionalities by using a round trip email sensor

[Node Settings](#) ▶ [Add Sensor](#)

NetCrunch allows various aspects of email monitoring.

Supported Protocols

All email sensors support IMAP4 and POP3, including secure connections.

Monitoring Mailbox - Mailbox Sensors

These sensors allow for monitoring of mailbox authentication, activity, performance and size. You can check if the mailbox is properly processed by checking oldest email in mailbox or when the last message has been received. The sensor is operating on the mailbox owned by other user, so the sensor is not changing the content of the mailbox.

Available alerts:

- Authentication Error
- Connection Error
- Protocol Error
- Trigger for Performance Counter
- Mailbox Activity Events: (if the oldest message is older than..., no new messages in last...)

Performance Counters:

- % Availability
- Alerts Triggered
- Check Time
- Emails Processed
- Emails Rejected
- Number of Messages
- Size of Largest Message

- Size of Messages

Monitoring Emails - Email Alert Sensor

The sensor allows to trigger alerts based on email sender, subject or body. It can match emails using simple text patterns or using parsing expressions (regular expressions).

Available alerts:

- Authentication Error
- Connection Error
- Protocol Error
- Alert on email by parsing expression
- Alert on email by search text

Performance Counters:

- % Availability
- Emails Processed
- Alerts Triggered
- Emails Rejected

Note:

In this case, the sensor must own the mailbox. It will automatically delete all processed emails.

Monitoring Mail Server - Round Trip Email Sensor

This sensor is intended to check the mail server functionality by sending and receiving test emails. It must use a dedicated mailbox, and it removes test mails from mailbox automatically.

Note:

Because SMTP server might be configured on a different IP address than POP3, you should add the sensor to the POP3 server, and then you can set SMTP connection IP address separately.

Available alerts:

- If alert has not been received
- If alert has not been sent
- Authentication Error

- Connection Error
- Protocol Error
- Trigger for Performance Counter

Performance Counters:

- % Availability
- % Incoming Availability
- % SMTP Availability
- Receive Time
- Send Time
- Total Operation Time

Monitoring Text Logs

NetCrunch allows monitoring text file content and has a special sensor for text logs.

➤ [Node Settings](#) ▶ [Monitoring](#) ▶ [Add Monitoring Sensor](#)

NetCrunch allows two levels of monitoring of log files. Simple monitoring can be configured with file sensors (using remote windows, ftp or http) which look for specific text pattern in log files.

More advanced log monitoring is possible with "Text Log" sensor which can parse the file and then the program can collect and alert on parsed entries.

File Sensors

File sensors allow analyzing a text file content using two alerts:

Text Search

event monitor can search for some text using plain text pattern or use regular expression. Incremental search will monitor only new occurrences in file.

Text Log Entry

event monitor tries to match lines and if new line is found, an alert is triggered. Search pattern can be plain text or regular expression. If multiple lines matching the alert are detected, they can be grouped into a single alert.

When log entry is collected program stores whole line of the log as signe parameter. Each time file sensor looks for text log entry it analyzes lines appended since last check.

See also:

Monitoring Files and Folders

NetCrunch allows monitoring files, folders and text logs

Alerting Actions

Text Log Sensor (Premium XE)

This sensor parses a file and converts each entry into a list of properties which later can be filtered like any other types of logs (window event log, syslog). This gives you more control over how alerts are triggered, and also allows better analyzing of collected log entries in the event log.

Text Log sensor allows remote log monitoring using:

- Windows
- FTP/S
- HTTP/S

Text log sensor can remotely monitor large log files on Windows (tested on gigabyte files); unfortunately, FTP or HTTP requires whole files to be downloaded.

NetCrunch contains sample text log formats and allows defining custom formats using text parsing expressions.

Text Log Parsing Expressions

➤ [Monitoring > Text Parsing Expressions > Text Log Expressions](#)

Separated Values

Best suited for simple log formats where each line contains fields separated by single character.

For example such line can look like this:

```
11/19/15 7:20:38 am,Information,Monitor started
```

And we can define that program should convert this to fields:

- Time
- Severity
- Message

Regular Expressions

In case of logs where there is no separator between fields we can use regular expressions. The expression must contain search groups to identify each field.

Simple example.

In this example our log can contain a number at the beginning and then message until the end of the line.

10345 : Error during loading module.

Expression:

`([0-9]*) : (.*)`

We can define two fields to manage such log:

- ProcessId
- Message

Parsing expression editor allows immediate testing of your expressions;

Text Parsing Expressions

← Back

Text Log Expressions

Name: Simple Log Type: Regular Expression

Regular Expression: `([0-9]*) : (.*)`

Variables: + Add

Processid	10345
Message	Error during loading

Test Text: 10345 : Error during loading

+ Add

OK Cancel

Alerts

- Alert on log file read error
- Alert on text log entry

For example, for our simple log we can define alert:

Any line of log

Log entry matching expression

Select log entry where all of the following apply

① Message starts with Error

< Add Condition >

Counters

The program can also trigger an alert on following performance counters:

- % Availability
- Alerts Triggered
- Log Entries Processed
- Log Entries Rejected
- Size

Monitoring Disk Parameters on Unix family systems

[Read what disk performance metrics can be monitored on Linux, Mac OS X, BSD and Solaris.](#)

Linux

Counters

- *Reads completed* - The total number of reads completed successfully.
- *Reads merged* - Reads which are adjacent to each other may be merged for efficiency. Thus two 4K reads may become one 8K read before it is ultimately handed to the disk, and so it will be counted (and queued) as only one I/O. This field lets you know how often this was done.
- *Sectors read* - The total number of sectors read successfully.
- *Read Bytes* - The total number of bytes read successfully.
- *Milliseconds spent reading* - The total number of milliseconds spent by all reads (as measured from `__make_request()` to `end_that_request_last()`).
- *Writes completed* - The total number of writes completed successfully.
- *Writes merged* - Writes which are adjacent to each other may be merged for efficiency. Thus two 4K writes may become one 8K write before it is ultimately handed to the disk, and so it will be counted (and queued) as only one I/O. This field lets you know how often this was done.
- *Sectors written* - The total number of sectors written successfully.
- *Write Bytes* - The total number of bytes write successfully.
- *Milliseconds spent writing* - The total number of milliseconds spent by all writes (as measured from `__make_request()` to `end_that_request_last()`).
- *I/Os currently in progress* - The counter that should go to zero. Incremented as requests are given to appropriate struct request_queue and decremented as they finish.
- *Milliseconds spent on I/Os* - This counter increases so long as *I/Os currently in progress* is

nonzero.

- *Weighted milliseconds spent on I/Os* - This counter is incremented at each I/O start, I/O completion, I/O merge, or read of these stats by the number of I/Os in progress *I/Os currently in progress* times the number of milliseconds spent doing I/O since the last update of this field. This can provide an easy measure of both I/O completion time and the backlog that may be accumulating.
- *Transferred bytes* - The sum of *Read Bytes* and *Write Bytes*
- *Transferred sectors* - The sum of *Sectors read* and *Sectors written*
- *Average requests size (Bytes)* - The average size (in bytes) of the requests that were issued to the device.
- *Average queue length* - The average queue length of the requests that were issued to the device.
- *Average time for I/O requests* - The average time (in milliseconds) for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.
- *Average read requests time* - The average time (in milliseconds) for read requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.
- *Average write requests time* - The average time (in milliseconds) for write requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.
- *Average queue length*

Calculated counters

- *Reads completed/sec* - The total number of reads completed (requests).
- *Writes completed/sec* - The total number of writes completed (requests).
- *Reads merged/sec* - The total number of reads merged.
- *Writes merged/sec* - The total number of writes merged.
- *Sectors read/sec* - The total number of sectors read.
- *Sectors written/sec* - The total number of sectors written.
- *Read Bytes/sec* - The total number of bytes read successfully.
- *Write Bytes/sec* - The total number of bytes write successfully.
- *Transferred Bytes/sec* - The total number of read and write bytes.
- *Transferred sectors/sec* - The total number of read and write sectors.

Mac OS X

Counters

- *Completed I/Os* - The sum of completed reads and writes

- *Transferred Bytes* - The sum of read and written bytes

Calculated counters

- *Completed I/Os/sec*
- *Transferred Bytes/sec*

BSD, Free BSD, Net BSD, Open BSD

Supported BSD distributions

- FreeBSD: *Reads completed, Read Bytes, Writes completed, Write Bytes, Completed I/Os, Transferred Bytes, Seconds spent on I/Os, Transactions queue length, Reads completed/sec, Writes completed/sec, Read Bytes/sec, Write Bytes/sec, Transferred Bytes/sec, Completed I/Os/sec*
- NetBSD: *Reads completed, Read Bytes, Writes completed, Write Bytes, Completed I/Os, Transferred Bytes, Seconds spent on I/Os, Reads completed/sec, Writes completed/sec, Read Bytes/sec, Write Bytes/sec, Transferred Bytes/sec, Completed I/Os/sec*
- OpenBSD: *Completed I/Os, Transferred Bytes, Seconds spent on I/Os, Transferred Bytes/sec, Completed I/Os/sec*

Counters

- *Reads completed* - The total number of reads completed successfully.
- *Read Bytes* - The total number of bytes read successfully.
- *Writes completed* - The total number of writes completed successfully.
- *Write Bytes* - The total number of bytes written successfully.
- *Completed I/Os* - The sum of *Reads completed* and *Writes completed*
- *Transferred Bytes* - The sum of *Read Bytes* and *Write Bytes*
- *Seconds spent on I/Os* - The total number of seconds spent by all reads and writes.
- *Transactions queue length*

Calculated counters

- *Reads completed/sec* - The total number of reads completed (requests).
- *Writes completed/sec* - The total number of writes completed (requests).
- *Read Bytes/sec* - The total number of bytes read successfully.
- *Write Bytes/sec* - The total number of bytes written successfully.
- *Transferred Bytes/sec* - The total number of read and write bytes.
- *Completed I/Os/sec* - The sum of *Reads completed* and *Writes completed*

Solaris

Solaris 10 and Solaris 11 are supported.

Counters

- *Reads completed* - The total number of reads completed successfully.
- *Read Bytes* - The total number of bytes read successfully.
- *Writes completed* - The total number of writes completed successfully.
- *Write Bytes* - The total number of bytes written successfully.
- *Completed I/Os* - The sum of *Reads completed* and *Writes completed*
- *Transferred Bytes* - The sum of *Read Bytes* and *Write Bytes*
- *Elements in wait state*
- *Elements in run state*
- *I/Os currently in progress* - The sum of *Elements in wait state* and *Elements in run state*
- *Milliseconds spent waiting* - Cumulative wait (pre-service) time
- *Weighted milliseconds spent waiting* - Cumulative wait length*time product
- *Milliseconds spent running* - Cumulative run (service) time
- *Weighted milliseconds spent running* - Cumulative run length*time product
- *Milliseconds spent on I/Os* - The sum of *Milliseconds spent waiting* and *Milliseconds spent running*
- *Weighted milliseconds spent on I/Os* - The sum of *Weighted milliseconds spent waiting* and *Weighted milliseconds spent running*
- *Soft errors* - A disk sector fails the CRC check and needs to be re-read
- *Hard errors* - Re-read fails several times for CRC check
- *Transport errors* - Errors reported by I/O bus
- *Total errors* - The sum of *Soft errors*, *Hard errors* and *Transport errors*
- *Average wait queue length*
- *Average run queue length*
- *Average transactions queue length* - The average queue length of the requests that were issued to the device.
- *Average wait time for I/O requests (ms)*
- *Average run time for I/O requests (ms)*
- *Average time for I/O requests (ms)* - The average time (in milliseconds) for I/O requests issued to the device to be served.
- *Average requests size (Bytes)* - The average size (in bytes) of the requests that were issued to the

device.

Calculated counters

- *Reads completed/sec* - The total number of reads completed (requests).
- *Writes completed/sec* - The total number of writes completed (requests).
- *Completed I/Os/sec* - The total number of completed IO requests.
- *Read Bytes/sec* - The total number of bytes read successfully.
- *Write Bytes/sec* - The total number of bytes written successfully.
- *Transferred Bytes/sec* - The total number of read and write bytes.

Sending Data to NetCrunch

Read howto send data to NetCrunch and create a custom monitor. You can easily turn any application or script into a NetCrunch agent.

Generic Agent Data Sensor

You can add the Generic Agent Data Sensor from sensor list on any node. The configuration is there is very minimal. It needs a name and it automatically creates an API key for an external agent required to send data to NetCrunch. The API key consists of the sensor name (without spaces) and node id number. For example, an API key can be JMX@1034, if we name our agent "JMX".

You can add multiple sensors on a single node for each application you need to monitor.

Retention Time

Because NetCrunch does not know how often you are going to send data to it, you need to specify a "retention time" for data, after which the data will expire and will be cleared out from the memory.

REST API

We wanted to keep the API for the sensor very simple. The simplest tool you can use to send requests to NetCrunch is cUrl open source project, available for almost any platform. You can find it at: curl.haxx.se

The API consists of only 5 requests:

Update

POST /api/rest/1/sensors/<api-key>/update

The request data must be send application/json content type:

Example

```
{
  "retain": 1,
  "counters": {
```

```
    "PBX/line status.0" : 1,  
    "PBX/line status.1" : 0  
  },  
  "statuses": {  
    "AC" : "On",  
    "Power": "On"  
  }  
}
```

As you can see, you can send multiple statuses and counters in a single request.

Counter

`GET /api/rest/1/sensors/<api-key>/counter?<parameter-list>`

Example

```
<nc-server-address>/api/rest/1/sensors/<api-key>/counter?Temp=65&Wind=4&@r
```

Counter/inc, Counter/dec

As NetCrunch stores counter values in memory, the agent can increment it without storing its actual value. The request will increase or decrease counter by given value.

Example

```
<nc-server-address>/api/rest/1/sensors/<api-key>/counter/inc?Door.Opened=-
```

Status

`GET /api/rest/1/sensors/<api-key>/status?<parameter-list>`

Example

```
<nc-server-address>/api/rest/1/sensors/<api-key>/counter?Door=Opened&@reta
```

Web Messages

You can easily send event messages to NetCrunch using an HTTP request. Program accepts POST and GET requests. In the examples above we skip the first part of URL, which is your NetCrunch server Web Access URL. We strongly recommend to configure the server to use the HTTPS protocol.

See also:

[External Event Sources](#)

See how you can create an alert for syslog messages, SNMP traps, Web Messages and Windows Event Log entries.

Alerting

External Event Sources

See how you can create an alert for syslog messages, SNMP traps, Web Messages and Windows Event Log entries.

NetCrunch can act as a log server for external events. It can store them in NetCrunch Event Log and perform defined alert actions (i.e. *notifications*) as the response. You can also correlate incoming events in order to track pending alerts.

Syslog Server

➤ [Tools](#) ▶ [Options](#) ▶ [Monitoring](#) ▶ [Syslog Server](#)

You can change the port on which NetCrunch listens to syslog messages (default 514) and set the option to forward each message to another syslog server.

Message Grouping

NetCrunch waits a given time frame and groups the same messages. This helps to avoid flooding with the same messages. This may happen in case of a device or service failure.

Configuring Syslog Alerts

Configuring NetCrunch syslog server is the first step, while the next one is creating alerts.

You can add an alert to a node that is sending syslog messages and specify a filtering condition for the received message. As such, you can create different alerts for different messages.

Only messages matching defined alerts (filters) pass into NetCrunch, whereas others are discarded.

Go to ➤ [Settings](#) ▶ [Monitoring Packs and Policies](#) or

➤ [Node Settings](#) ▶ [Monitoring](#) and click on Syslog tile in Node Monitoring section.

Click ➤ [Add Alert](#) and choose `<Create new Received Syslog Message Event>`.

Web Messages

➤ [Tools](#) ▶ [Options](#) ▶ [Monitoring](#) ▶ [Web Messages](#)

You can easily send an event message to NetCrunch using HTTP request. Program accepts POST and GET requests. In examples we skip first part of URL which is your NetCrunch Server Web Access URL. We strongly recommend to configure the server to use HTTPS protocol.

You can use cURL program (available on multiple platforms) to send requests to NetCrunch. You can download it from curl.haxx.se

Service URL

`http://<nc-server>/api/rest/1/event/<node-identification>`

Node identification is node IP address or DNS name.

Sending message by GET

The simplest way is attaching message as a search string.

```
api/rest/1/event/crm.acme.com?CRM%20must%20be%20restarted
```

Because URLs cannot contain spaces, it must be properly encoded.

If you form query string as a parameter list, it will be converted to JSON object and then you will be able to create alerts based on these parameters.

Example:

```
api/rest/1/event/crm.acme.com?error=1
```

Sending message by POST

NetCrunch accepts data encode as typical form encoding `application/x-www-form-urlencoded` and allows to filter events on parameters. Another accepted encoding is JSON `application/json`.

Curl Example

```
curl -d "error=1" http://192.168.10.112:8008/api/rest/1/event/192.168.1
```

Configuring Web Messages

NetCrunch can receive messages by default. You can disable this feature and set message grouping option.

Message Grouping

NetCrunch waits a given time frame and groups the same messages. This helps to avoid flooding with the same messages. It may happen in case of a device or service failure.

Configuring Web Message Alerts

You can add an alert to a node that is sending web messages and specify filtering condition for the received message. So you can create different alerts for different messages.

Only messages matching defined alerts (filters) pass into NetCrunch, whereas others are discarded.

Go to [Settings ▶ Monitoring Packs and Policies](#) or

[Node Settings ▶ Monitoring](#) and click on Syslog tile in Node Monitoring section.

Click [Add Alert](#) and choose `<Create new Received Syslog Message Event>`.

Receiving SNMP traps

NetCrunch receives SNMPv1, SNMPv2 and SNMPv3 traps. It can also forward all received traps to another SNMP manager. Forwarding can be set in [Tools ▶ Options ▶ Monitoring ▶ SNMP Traps](#)

[Settings ▶ SNMP Trap Receiver](#)

On the *Settings* page you can check the current status of the SNMP trap receiver. If SNMP is not enabled, click the button and then you can enable it and set its options.

You can change the port on which NetCrunch listens for SNMP traps (default 162) and set the option of forwarding traps to another SNMP manager.

Trap Message Grouping

NetCrunch waits a given time frame and groups the same SNMP trap messages. This helps to avoid flooding with the same messages.

Configuring SNMP Trap Alert

You can add an alert to the node that is sending SNMP trap by creating a trap alert. You can also receive traps in the External Events window and click on desired trap to create an alert for it.

To add trap alert go to [🚩 Node Settings ▶ Monitoring](#) and add click SNMP Traps time in Node Monitoring section.

In order to add an SNMP trap to the Monitoring Pack, it must have SNMP required option checked. Otherwise SNMP events will not be visible.

Monitoring Windows Event Logs

[🚩 Tools ▶ Options ▶ Monitoring ▶ Windows Event Log](#)

Monitoring Event Log is enabled by default. In fact, it's not just a passive receiver like SYSLOG or SNMP trap. Event Log monitor connects to a remote machine and needs authentication in order to register for receiving Windows Event Log events. It is rather an extension of Windows Monitoring Engine.

When you click the **Windows Event Log** button in the *Monitoring* tab, you can change global options for Windows Event Log engine.

Event Log Entries grouping

NetCrunch waits a given time frame and groups the Windows Event Log entries.

Configuring Windows Event Log Alert

Windows Event Log monitoring has the same requirements as standard Windows monitoring. The node must be Windows type. Then you can go to [🚩 Windows Event Log](#) sensor under the Windows section and add alert.

Alternatively, you can create a new Monitoring Pack with a set of rules to monitor Windows Event log in

[🚩 Monitoring ▶ Monitoring Packs & Policies](#) .

Alert Conditions & Correlation

Alerting on an event is simple. NetCrunch is capable of alerting, for example, when an event did not occur, or in more complex conditions. Advanced correlation allows correlating events from multiple nodes.

Conditional Alerts

NetCrunch allows you to define additional conditions for each defined alert, regardless if it is a node status, an event log alert or SNMP trap. These conditions allow you to trigger an action, even if an event has not been triggered. For example, if there is no log entry confirming an operation (i.e. backup). Also, NetCrunch can receive heartbeat events and notify if one is missing. Other conditions allow you to suppress alert execution for some time (as alert won't be triggered, the actions set to run on alert close won't be executed).

Available conditions

- On event
- if event happen after x time
- if event happen more than x time
- Only if time range
- Only if time not in range
- If event not happen in given time range
- if event not happen after x time
- if event is pending for more x

NetCrunch supports alerting rules ranging from simple time range rules to complex schemes.

Advanced Correlation

NetCrunch (PremiumXE only) contains a global Monitoring Pack with correlation events allowing you to correlate events from multiple nodes. This can be helpful when you want to define an alert only if alternate resources have failed (redundant connections).

Alerts can be triggered when all events are in a pending state (all events must have pending correlation), or by defining a time frame in which they have to occur. These correlated alerts can be for any events previously defined on any node in the Atlas.

Pending Alerts & Event Log Views

This feature makes NetCrunch special among other monitoring systems. All internal alerts are automatically correlated and NetCrunch tracks their pending state. Additionally external alerts can be correlated by the user. No more digging in event logs or manually closing alerts!

Pending Alerts

This separate view shows only current alerts instead of forcing administrators to browse the event log which offers a history of all alerts. Event log views can be synchronized with the Atlas Tree Window. This means that when you click on a specific view, such as location or node group (i.e. servers), pending alerts are automatically displayed for this view.

Summary

The Summary view shows alert statistics for a given view. The statistics are grouped by monitoring category and also by custom views. This gives you a quick overview of what types of alerts happened in a given time range.

Custom Event Log Views

NetCrunch offers many predefined event log views and allows you to create custom views using an intuitive query builder. Views can be saved and used for any node group in the Atlas.

Event Details

For each event in the event log, NetCrunch offers a Details view containing all alert details and parameters. This window shows all executed actions and also the event that closed a given alert.

If the alert has been triggered on a performance counter value, the Details view displays a chart showing values at the time of the alert.

Alerting Actions and Escalation

Each alert can have its own alert escalation script. This means that NetCrunch executes actions if the alert condition persists or can execute actions when alert is closed.

Actions

As a response to an event, NetCrunch can execute a sequence of actions. Actions can also be executed when alert ends (on close). NetCrunch contains various actions including: Notifications, Logging, Control Actions and Remote Scripts.

Notifications are very flexible and can be controlled by user profiles and groups. Additionally, they can be combined with a node group (atlas view) membership, so it's possible to send notifications to different groups based on network node location or some other type of relationship.

Predefined Actions

- Basic Actions: Play Sound, Display Desktop Notification Window, Add Traceroute to Alert Message, Add Network Service Status to Message, Notify user of group, email, SMS Text Message (via email), SMS Text Message via Mobile Phone
- Computer Control Actions: Run Windows Program, Run Windows Script, Run SSH Script, Restart Computer, ShutDown Computer, Set SNMP Variable, Terminate Windows Process, Control Window Service, Wake on LAN
- NetCrunch Control Actions: Change Node Monitoring State, Modify Node Issue List, Set Event Arrived Issue, Clear Event Arrived Issue
- Local Logging Actions: Write to File, Write to Windows Event Log, Write to Unique File,
- Remote Logging Actions: Send SNMP Trap, Send Syslog Message, Trigger WebHook

- Linux Remote Scripts: Shutdown, Reboot, Restart SNMP Daemon, Mount CD-ROM, Dismount CD-ROM
- Windows: Run Disk Defragmenter, Start SNMP Service, Stop SNMP Service

See [Alerting Actions](#)

Action Escalation & Conditional Execution

Actions can be executed immediately or with a delay (if the alert is not finished), and the last action can be repeated. Additionally, you can specify actions to be executed automatically when an alert is closed.

For example, you can decide to send a notification to some person and then, after some time, execute a server restart operation.

The script above executes only notifications for critical alerts and restarts the node causing this event if this is a Windows Server node.

Receiving SNMPv3 Notifications

SNMPv3 uses a different model for notifications than previous SNMP versions. To receive SNMPv3 notifications, you need proper authentication profile, since they have to be decoded with password and given encryption settings.

Because SNMPv3 allows authentication and encryption of receiving traps, you need to define SNMPv3 Notification profile for them, otherwise the program won't be able to decode anything.

Unlike in v1 and v2 profiles, profiles for v3 are global which means you can defined them via

➤ [Monitoring > SNMP Communities and passwords](#) and you do not have to assign them to nodes.

Edit SNMP Profile

Profile:

SNMPv3 Traps & Trap Info

This profile needs to be set in order to receive and decode SNMP v3 traps and trap info messages.

User:	Authentication:	Password:	Encryption:	Encryption Password:
<input type="text" value="admin"/>	<input type="text" value="HMAC-MD5-9"/>	<input type="text" value="****"/>	<input type="text" value="AES 256"/>	<input type="text" value="*****"/>

Remote SNMP Engine Id:

OK Cancel

NetCrunch supports following authentication and encryption protocols:

Authentication Types

- HMAC-MD5-96
- HMAC-SHA-96

Encryption

- DESC
- 3DES
- AES 128
- AES 192
- AES 256

Preventing False Alarms

Learn what situations increase alerts' volume and how NetCrunch helps you prevent false alarms.

False alarms can arise in many situations. Here are some common examples of events which can generate false alarms:

- When an intermediate device fails, all events and devices depending on this device tend to cause false alarms.
- Too sensitive Counter Event Triggers - you are alerted on momentary value change
- Too sensitive Network Service monitoring - you are alerted when single or two packets are dropped or on single connection lost
- Slow responding SNMP devices - sometimes heavily loaded SNMP devices tend to respond with a long delay.

NetCrunch helps you avoid alert overload by implementing the following functionalities:

Prioritized Monitoring

(XE edition only)

The order and frequency of monitoring nodes depend on the priority; intermediate nodes have a higher priority than nodes connected through them.

Event Suppression

(XE edition only)

Event Suppression is the technique of preventing false alarms caused by intermediate network

connection failure.

When NetCrunch receives an event related to a node connected through some intermediate link, it ensures first if the link is OK, so the event might be a result of that the connection has been broken. You can define exceptions when you definitely want to receive event from descendant nodes.

Additionally NetCrunch can suppress network service events when node is DOWN.

Example Monitoring Dependency Tree with suppression options:

Monitoring dependency tree	Node events	Descendant events
10.20.1.51		✖ Suppress
192.168.10.4		✖ Suppress
war-b2-link.telia.net (213.248.89.85)	✖ Suppressed	✖ Suppress (inherited)
softlayer-ic-307612-ash-b1.c.telia.net (62.115.47.2...)	✖ Suppressed	✖ Suppress (inherited)
cisco2960g-5-lab.adrem (192.168.1.114)		✖ Suppress
cisco2950-lab.adrem (192.168.1.113)	🔔 Triggered	✖ Suppress (inherited)
labesxi52.lab.ad.adrem (192.168.2.52)		✖ Suppress
10.20.1.56		✖ Suppress
labesxi51.lab.ad.adrem (192.168.2.51)		✖ Suppress
labesxi62.lab.ad.adrem (192.168.2.62)		✖ Suppress
labesxi38.lab.ad.adrem (192.168.2.38)		✖ Suppress

Fix Counter Event Triggers

When a counter value often changes you should set a trigger on the average value instead of the actual counter value. Also, you can define hysteresis by adding a reset threshold to the trigger.

Read more about [Event Triggers for Counters](#)

Fix Network Service Monitoring Parameters

Usually, NetCrunch sends multiple requests in a row to check network service response. To make sure the service responds, increase timeout for the service or set *additional repeat count*.

Fix SNMP Monitoring Engine settings

SNMP works over UDP protocol, which is not reliable, as packets can be lost. In such case, the program is waiting a given time and when it does not get any response, it repeats the request.

Because of the nature of UDP communication, the program can't recognize whether the packet is lost or device is busy and delays the response.

Go to [Node Settings](#) ▶ [SNMP](#) and increase SNMP timeout for busy devices or SNMP retry county for unreliable connections.

Alert Integrations

NetCrunch can easily integrate alerting with external service desk, productivity and messaging systems.

Integration profiles allow you to configure integration actions that can be executed in response to alerts. NetCrunch can automatically send messages and manage tickets in external systems. Several systems allow two way integration, which means that NetCrunch can close a pending alert in response to a notification sent from an external system.

From NetCrunch to External Service

Integration Profiles

There are several steps that need to be done before NetCrunch can send data to the external system. Each of these steps require some kind of authentication by the API Key created for the application.

Integration Profiles store connection settings for these systems. So before we add an action to the alerting script, we need open [Monitoring > Integration Profiles](#) and add profile for the system with which you want to integrate NetCrunch.

We can group these external systems by type of services provided.

Messaging & Notifications:

- Campfire - send message to Campfire chat room,
- HipChat - send message to HipChat room
- IFTTT - trigger IFTTT event
- Pushover - send notification to multiple devices using pushover,
- Ryver - send notification via Ryver
- Slack - send message to Slack
- Twitter - send tweet or direct message

SMS/Text Messaging:

- Clicatell - Send text message to the phone via Clicatell service

Service Desk & Productivity Systems:

- Asana - Create, update or close task
- Connectwise - Create or close ticket
- Flowdock - Create or close ticket
- Freshdesk - Create or close ticket
- Freshservice - Create or close ticket
- JIRA - Create or comment on ticket
- JiTBIT - Create or close ticket
- LiveAgent - Open or resolve ticket
- Mojo Helpdesk - Create or close ticket

- Ops Genie - Create or close alert
- Pagerduty - Create or resolve incident
- Trello - Create card, mark card as close, move card to another list
- Zendesk - Create or close ticket

Adding an Action

Once you have a profile, you can add integration actions to the alerting script. Each action has different options depending on the integration. You can also start from adding an action and you can create a profile from the action editor.

From External Service to NetCrunch

Several systems allow sending information back to NetCrunch when tickets are closed so we can close the alert from the NetCrunch side. We need to open [Monitoring ▶ API Key Manager](#) to configure back-links to NetCrunch. The configuration is simple: you add a profile for given system, and then you can copy part of URL to send data to NetCrunch from external system.

NetCrunch Web Access must be accessible outside the firewall. We recommend using reverse proxy from some edge server as it gives you more flexibility and allows for using already issued wildcard certificates. The other option is to install SSL certificate into the NetCrunch Web Server, which always runs on latest OpenSSL version.

Supported Systems:

- ConnectWise
- JIRA
- JitBit
- LiveAgent
- OpsGenie
- Pagerduty

Maintenance

Read about [howto keep NetCrunch running smoothly over time](#), [howto upgrade it](#) and [keep your network information up to date](#).

Troubleshooting

We would like to claim that NetCrunch is trouble-free software. The program goes through various testing procedures and we also maintain a development process designed to lower defect rates.

According to various studies (the situation didn't change over the last 20 years) - only 75% to 90% of bugs can be detected before the release. Additionally, software like NetCrunch can face many edge cases – by interacting with many vendor devices and technology implementations.

Networks are full of non-standardized technologies which are only defined through RFC documents. Often vendors which posted the RFC documents do not comply with them when producing devices – this often happens with SNMP and more so with Netflow.

Different versions of Operating systems also have their issues – for instance, there are unfixed bugs in WMI in Windows 2003 Server systems.

Releases

To address various issues and bugs in the software, we release NetCrunch minor releases several times a year. These releases can be installed over previous versions and they usually do not change the data format. Read [Update, Migrate and Backup](#) .

Issue Sources

Let us explain where the problems come from, and how you can fix them easily, or help us to fix them for you.

Configuration

Most monitoring issues are caused by invalid configuration, as there is no way for NetCrunch to connect to servers with wrong OS credentials or wrong SNMP profiles (communities or password).

NetFlow

There is a range of protocols based on NetFlow implemented from various vendors. They usually conform to NetFlow v5 protocol. Each device has specific settings and starting from NetFlow v9, data contained in flows depends on configuration. First, you need to setup the device to send data to NetCrunch – there is a range of articles about Netflow configuration on the Internet.

If NetCrunch is not able to decode the flow data, you need to capture such data with Wireshark and send it to us. Sometimes the devices have bugs in their NetFlow implementations. Although we can't fix them, sometimes we can go around and accept invalid data.

SNMP MIBs

There are more 10,000 of MIBs circulating on the Internet. As there is no standard for MIB compiler (standard MIBs were defined only in RFC documents) it is somewhat hard to compile them. Often, they were written once and never compiled or compiled with some specific compiler, in a very specific environment.

Usually, the source of problems in MIBs is wrong syntax or missing module that some MIB may depend on. This can be often fixed by specifying module name alias.

If you are not familiar with MIB's and you can't solve these problems, just contact our support – we will try to find a solution.

Windows

Windows is a complex system by its history. It's built by layer on top of another layer. The easiest way to manage Windows configuration is by Active Directory. But we know that in real life there are many unlinked systems and Windows versions.

Problems with Windows (Vista/2008 or later) are always related to Windows settings. See [Windows Monitoring Setup](#). For instance, monitoring of a workgroup Windows 7 workstation (server is easier) is hard without using a built-in local Administrator account.

NetCrunch Performance Limits

They're always limitation to NetCrunch

Licensing

it may not impose limits on the number of nodes being monitored

Hardware

such as: memory, disk and network set the ultimate limit of the software

NetCrunch is a multi-threaded system which scales well with a number of processors – especially because many tasks are dispatched to different processes.

Slow SATA disks are very inefficient in reading large quantities of data. We recommend using SSD drives instead.

Diagnostic Report

[↗ \(?\) ▶ Diagnostic Report](#)

The report gathers statistics about your NetCrunch configuration. It contains information such as memory used, number of consumed internal resources and program queues. The report can be exported to file in XML format or sent directly to AdRem Software from the console. It helps us to determine much loaded NetCrunch puts on your system.

Bug Reports

Bug Reports are no crash dumps.

Mostly they are well handled exceptions, but something that was not expected by the program.

It is always beneficial when you let us receive them automatically. The reports contain information about your system, memory, processor and program execution context.

We do not know your computer address except its Windows name. All reports come directly through email to our internal secure database and are fully confidential.

You can review those files (if there are any) using NCDiag program located in NetCrunch server directory.

Logs

As many NetCrunch components run as background processes, they use text logs to store their activity and potential issues. You can review them using NetCrunch Console [↗ View ▶ Logs](#).

NetCrunch Logs:

Atlas Backup

Contains NetCrunch auto-backup process activity log

Atlas Import

Contains import log

Auto Discovery

Contains activity or issues of NetCrunch autodiscovery process

Inventory Audit Writer

Log of process writing inventory data to the database

NetCrunch Open Monitor

Log of open monitor activity

NetCrunch Sensor Monitor

Logs of all NetCrunch sensors activity

Task Scheduler

Task scheduler is responsible for running Report Generator and Auto Discovery process

Report Generator

Log of process generating automatic reports

NetFlow Server

Log of NetFlow server, which may contain

Message Sender

Logs on messages sent by NetCrunch

NetCrunch Audit

The log contains logs of all NetCrunch users' login/logout events

Quick configuration backup

Sometimes we need only to save configuration files without saving collected monitoring data (performance trends and event log data).

In order to do it, follow the procedure below:

➤ Atlas ▶ Maintenance

Select Quick Backup from the Backup Menu

License Installation

Read to install NetCrunch license properly.

To install a new program license key or check for any available updates go to

➤ Tools ▶ Options ▶ General options ▶ License Manager .

What is NetCrunch License

A single license key is as a combination of the unique .als file and the common Activate.key. This means that you can use the same Activate.key for all your AdRem NetCrunch licenses.

First Installation

Go to [MyAdRem customer portal](#). Login using credentials that you've received in the order confirmation email.

In the Licensed Products list you can see your license key. Click on the product name.

Licensed Products:

Licenses	Product Name ▲	Service Agreement	Version	Released
1	 NetCrunch 7.x Premium	September 19, 2013 (expired)	7.2.2.2770	August 30, 2013
2	 NetCrunch 7.x PremiumXE	October 9, 2013 (valid)	7.2.2.2770	August 30, 2013
1	 NetCrunch 7.x Remote Access	September 19, 2013 (expired)	7.2.2.2770	August 30, 2013

In order to download program file and licenses click on the product name.

In *Program Downloading* window go to License Downloading page.

Click *Download all licenses in .ZIP file* link.

Serial Number	Created	Expire on	Units	Activation Key	Key File
 74135304.als	June 21, 2013	September 19, 2013	125	MZDAJFKMOFRAOHJ	 Activate.key
 74135314.als	July 11, 2013	October 9, 2013	1	MZDAJFKMOFRAOHJ	 Activate.key

 Download all licenses in .ZIP file

In NetCrunch, go to [Options ▶ General Options ▶ License Manager](#) page.

Click the [Install License](#) button. The Open window displays.

Select the `.als` file and click [Open](#). The license will be installed on the machine and added to the Installed License List on the *License Manager* page.

Updating the License

[Options ▶ General Options ▶ License Manager](#).

Click the [Update license](#) button. The program automatically connects to the AdRem Software servers, checking for available licenses.

New licenses available for the installed product will be downloaded and installed.

Remote Access License

A remote access is a separate product and needs a separate license key (.als file).

It requires additional download from MyAdRem Customers Portal as it is not included in the *Download all licenses in .ZIP file*.

The installation procedure is the same as for the NetCrunch license.

Auto Discovery

The primary NetCrunch goal is to keep all your network data up to date. This includes also network nodes and virtual machines.

Discovering Network Nodes

NetCrunch can periodically discover nodes in a given IP network. Go to [IP Networks](#) select IP network view and open [Properties ▶ Monitoring](#) to schedule a periodic network discovery. The minimum network discovery time is 1 hour.

Discovering Active Directory Containers

NetCrunch creates automatic folders with all known containers. To enable auto scanning of the AD container, go to given view and open [Properties ▶ Monitoring](#). Discovered nodes can be automatically added to the Atlas or they will be kept in Task notification window so you can filter them manually.

Virtual Machines (ESX/i)

As NetCrunch monitors ESX/I virtual machine hosts, it can add newly discovered machines to Network Atlas automatically. The option is disabled by default - you can also add virtual machines from ESX/I VM machine list manually.

NetCrunch Databases

NetCrunch is optimized for speed and scalability. Each database serves a different purpose.

NetCrunch Atlas (XML, JSON)

NetCrunch Atlas holds current network objects, and their state, maps and all dependencies. Atlas is an in-memory database which is saved periodically to disk as a number of XML formatted files. The Atlas also holds all NetCrunch monitoring configurations.

NetCrunch Event Log (SQL)

All events and action execution logs are kept in SQL database, accessible through ODBC driver. The

database is a highly utilized disk – which is key to its performance. The requirements for SQL databases are pretty much the same as for other SQL databases. SATA disks can handle effectively up to a few GB. Use SSD drives or RAID for more than 5GB of data.

NetCrunch Status DB (NoSQL)

NetCrunch keeps state of various objects in hierarchical in-memory database. These objects include state of triggers, state of monitors and any other monitored object. The object can be also set from external sources by agents and sensors. Database is stored on the disk for persistence.

NetCrunch Performance Trend Database (NoSQL)

This is a specialized NoSQL database optimized for writing a large number of performance metrics. Incoming data are stored in many journaling files which are processed and compressed daily. This database contains raw collected, and some pre-calculated data, for effectively accessing data in a long period of time (weeks, months or years). This assures that NetCrunch can easily store and process 250 million of performance records per day.

Update, Migrate and Backup

[Learn howto perform NetCrunch updates, migrate it to other machine or setup backup](#)

Updates

NetCrunch is updated several times per year, and depending on an update type, the procedure might be slightly different.

Maintenance

These updates (i.e. 9.0.1) contain mostly bug fixes and small improvements which don't require changes in configuration data. They can be safely performed by simply installing the new version (installer will automatically uninstall previous one).

Check your maintenance license expiration date - If you install version released after you maintenance license expired, it will work only in demo mode (30 day trial).

Minor Versions

The minor version (i.e. 9.2 or 9.3) updates contain some new features, so they also require some changes to the existing configuration. Thus, we recommend performing configuration backup before starting the update.

You don't need to save all trend and event data – just save the configuration. See [Quick configuration backup](#) procedure.

Major Version Upgrades

Major versions usually introduce some changes to the data in order to increase program performance, stability and accommodate to features.

Upgrading from one version to another seems to be fairly easy. Starting from version 9 all upgrades will be performed in-place without copying data.

When you upgrade from older version (than 9), the program transfers all data from old version folder to the new one; you don't need to perform any backup. All previous version data are intact. You can reinstall previous version at any time.

Make sure that disk usage is less than 50%.

The installer automatically uninstalls the previous version of the program (keeping all data) and asks you to import data from it. (Make sure you have the latest version of the old program version installed).

Migrate

In case you need to migrate NetCrunch to new hardware, you should export all data to a single file, copy it to another machine and import it to freshly installed NetCrunch instance.

Export Data

To export data to file go to [Atlas > Maintenance > Backup](#) and select [Backup to separate folder...](#) .

Make sure you select [Full backup for migrating data to other machine](#) option

Import previously exported data

Go to [Atlas > Maintenance > Import Atlas from other machine...](#) . Select a file with the backup.

Atlas Backup

[Atlas > Maintenance > Backup](#) and select [Modify Atlas backup schedule...](#) at the bottom or go to [Atlas > Atlas Properties > Maintenance](#)

Atlas backup is, by default, scheduled to be run automatically every day at 1 am. The backup contains: Performance Trends data (trendDB), Event Log database and encrypted password database, The procedure keeps the last 3 backups, but you can increase the number of backups to keep.

Atlas backups store only network monitoring data. The program configuration, such as NetCrunch server options, is not stored.

You can run backup on demand by going to [Atlas > Maintenance > Backup](#) .

Quick Backup

Quick backup can be used before you start making configuration changes, so you can go back to the previous state at any time. It stores only Atlas configuration files such as: nodes, alert configuration, maps etc. It does not store even log database nor performance data.

Backup Folder

[Tools > Options > General Options > Maintenance](#) .

Each Atlas backup is stored in a single file in the directory specified in the program options. The default location points to a subfolder of the program data directory.

It's recommended to store backup files on a separate disk or disk partition. For example "E:\NetCrunch\backup".

Restoring the Atlas

[Atlas](#) ▶ [Maintenance](#) ▶ [Restore](#) .

You can easily restore previous Atlas versions by choosing a backup date in the Restore window. You can decide to restore only configuration or full data.

Atlas Export/Import

This can be done using Backup and Restore options in [Atlas](#) ▶ [Maintenance](#) window now.

Restarting

[Read restart recommendations.](#)

To improve the system performance, we recommend restarting NetCrunch Server service once a month. We also recommend to install Microsoft updates and patches so you whole systems should also be restarted at least once per month.

NetCrunch includes auto restart feature so it restarts its service at given day and time. To change auto restart parameters go to: [File](#) ▶ [General Options](#) ▶ [Maintenance](#) . The shortest time between restarts is 1 day and the longest is 100 days.

Reference

The reference lists of NetCrunch resources, operations and definitions.

Alerting Actions

NetCrunch allows using many alerting actions such as: notifications, remote execution, creating tickets in help desk systems, sending messages to other system and NetCrunch configuration actions.

Actions are executed in response to an alert, and they organized in sequences by Alert Action Lists.

Alerts are grouped to make finding the desired Action easier. Groups refer to their purpose and are gathered into tabs in the *Add Action* window.

Basic Actions

Desktop

Play Sound

The action play sound file (in .wav format) as the response to an alert.

Display Desktop Notification Window

The action shows a small notification window above Windows Task Bar

Diagnostics

These actions can extend alert information so when next action would be sending email notification it will contain also diagnostic action result.

Add Traceroute to Alert Message

Execute traceroute to determine the point where the remote connection is broken,

Add Network Services Status to Alert Message

Adds status of all node network services to the alert information.

Notifications

Notify User or Group

This is recommended notification to be used. It automatically chooses right users and profiles depending on defined user profiles and groups.

Simple Notification

This is simple, but use it only in case of single administrator or very simple notification scheme. You can directly specify the notification type (email or SMS) and the recipient.

Note:

To use the SMS notification via GSM phone, you need to connect and setup the GSM phone or modem to the NetCrunch Server.

Control Actions

Controlling Computers

NetCrunch can execute various control actions remotely. Event description can be passed to action when needed in XML format. Actions by default can be executed on a node causing alert or on any other node from the Atlas.

Windows

Run Windows Program

Program can be copied to and executed on the desired machine.

Run Windows Script

Run script that can be copied to and executed on the desired machine by given scripting host.

Terminate Windows Process

Terminates process by its name (Windows nodes only).

Start, Stop, Pause Windows Service

Perform control action on given service. (Specify service by its name or select from the list or running services).

Unix Family

Run SSH Script,

Run script using SSH connection can be copied to and executed on the desired machine by given scripting host.

Any OS

Shutdown Computer

Shutdown remote node (it can be executed on every supported OS)

Restart Computer

Restart remote node (it can be executed on every supported OS)

Other

Set SNMP Variable

Set a given SNMP variables on a remote node (you can use MIB database or enter OIDs manually)

Wake on LAN

Send "Wake on LAN" packet, which if supported by the device can turn the device power on.

NetCrunch Actions

Change Node Monitoring State

Enable/disable node monitoring. Action can disable node monitoring indefinitely or only for a specified time.

Change Node Monitoring Time

Change node monitoring time

Modify Node Issue List

Turn alerts into issues. Set or clear issues on a node.

Modify Node Custom Field

Change or clear node custom field

Set Issue

Add the issue upon an alert to the node issue list.

Clear Issue

Clear issue related to alert (created by set issue) from the node issue list.

Integrations

This section allows integration with external service desk, productivity and notification systems:

Notification

Campfire

Send message to campfire chat room,

HipChat

send message to hipchat room

IFTTT

trigger IFTTT event

Pushover

send notification to multiple devices using pushover,

Ryver

send notification via Ryver

Slack

send message to slack

Twitter

send tweet or direct message

SMS Gateway

Clicatell

Send text message to the phone via Clicatell service

Service Desk

Asana

Create, update or close task

Connectwise

Create or close ticket

Flowdock

Create or close ticket

Freshdesk

Create or close ticket

Freshservice

Create or close ticket

JIRA

Create or comment on ticket

JITBIT

Create or close ticket

LiveAgent

Open or resolve ticket

Mojo Helpdesk

Create or close ticket

Ops Genie

Create or close alert

Pagerduty

Create or resolve incident

Trello

Create card, mark card as close, move card to another list

Zendesk

Create or close ticket

Logging

Local

Actions execute on the *NetCrunch Server* machine. You can specify desired message format for the action.

Write to File (Append)

Action appends information about the alert at the end of the selected file. The file must be accessible from NetCrunch Server. File will be created if does not exist.

Write to Windows Event Log

Writes information about the alert into specified Windows Event Log.

Write to Unique File

Writes detailed information about the alert to unique per each alert file.

Remote

Send SNMP Trap

Send SNMP alert with given alert information. In order to understand NetCrunch SNMP traps by the remote SNMP manager export NetCrunch SNMP MIB and import/compile in that manager. [Integrating NetCrunch with other NMS](#)

Send Syslog Message

Send *syslog* message using defined message form to remote *syslog* server.

Trigger Webhook

Send HTTP POST request to given URL with event data as payload. Supports XML and JSON data format.

Note:

Please note that while selecting the Write to File and Write to Unique File actions on a remote Administration Console, the Filename or Directory fields require providing the path manually. The Select Directory or Open Files icons are grayed out.

Predefined Scripts

Linux

- Shutdown Linux Machine
- Reboot Linux Machine

- Restart Linux Machine
- Mount CD-ROM
- Dismount CD-ROM

Windows

- Run Disk Defragmenter
- Start SNMP Service
- Stop SNMP Service

Monitoring Packs

NetCrunch Monitoring Packs allows manage settings for multiple nodes by policy. Currently program includes more than 150 monitoring packs for monitoring devices, applications and operating systems.

Monitoring packs monitor any atlas, map or even single node.

A monitoring pack may consist of two elements: alerting and data collection (reports).

It is a set of rules that defines what event condition should be checked (alerting) and which performance data should be collected (reporting).

NetCrunch includes a set of predefined monitoring packs with associated events and reports.

Predefined Monitoring Packs

Operating Systems

Windows

There are several Monitoring Packs which you can use for monitoring different aspects of your Windows environment

Active Directory (automatic)

Observe Replication and Service error. Watch Active Directory services status.

Operating System must be Windows Server

Monitored network services list contains: LDAP

Basic Windows Monitoring (automatic)

Provides basic workstations monitoring. Observe processor utilization, memory usage and free disk space.

Operating System must be Windows Workstation

Simplified Monitoring must be Disabled

DHCP Server

Observe DHC Server service and its errors.

Distributed File System (DFS)

Observe Windows Event Log for specific DFSR warnings and errors

DNS Server (automatic)

Observe DNS errors. Watch DNS network service and Windows service status.

Operating System must be Windows Server

Monitored network services list contains: DNS

Hyper-V Server

Observe the overall processor utilization of Hyper-V environment and watches it's Windows service status

Network Services Health

Watch for DHCP, DNS, WINS or other TCP/IP errors.

Processes (Windows)

It allows collect information for processes

Security Audit

Watch for Account events, logon and password problems

Terminal Services

Watch number of Active and Inactive sessions

Windows Server (automatic)

Monitor typical system performance indicators like %Processor Time, Memory, Disk Free space, disk latency etc.

Operating System must be Windows Server

Windows vCenter 5.1

Observe state of vCenter Windows services

Linux

Monitor most important Linux performance indicators such as: processor and memory utilization, free disk space, available swap, and create Linux Server Report.

Linux (automatic)

Monitors basic system parameters like: *Available Memory, Swap Available, Processor Utilization* and *Volume Free Space*

Linux (SNMP)

It requires SNMP to be enabled. It monitors: *CPU Load, Load Check, Minimum Swap*

Space, Swap Space

Network Traffic (automatic)

It allows collecting data for traffic statistic on Top Charts

Processes

It allows to collect information for processes

Mac OS X

Monitor most important Mac OS X performance indicators. such as processor and memory utilization, free disk space, and create Mac OS X Report.

Mac OS X (automatic)

Monitors basic system parameters like: *Available Memory, Processor Utilization and Volume Free Space*

Processes

It allows to collect information for processes

BSD

Monitor most important BSD performance indicators, such as processor and memory utilization, free disk space, and create BSD Report.

BSD (automatic)

Monitors basic system parameters like: *Available Memory, Processor Utilization and Volume Free Space*

Processes

It allows collecting information for processes

Network Traffic (automatic)

It allows collecting data for traffic statistic on Top Charts

Solaris

Monitor most important Solaris performance indicators such as: processor and memory utilization, free disk space, and create Solaris Report.

Network Traffic (automatic)

It allows to collect data for traffic statistic on Top Charts

Processes

It allows to collect information for processes

Solaris (automatic)

Monitors basic system parameters like: *Available Memory, Swap Available, Processor Utilization and Volume Free Space*

Solaris (SNMP)

It requires SNMP to be enabled. It monitors: *CPU Load, Load Check, Minimum Swap Space, Swap Space*

VMware

VMware ESXi (automatic)

Automatically enables monitoring of your VMware hosts. It will inform you if any of the hardware sensors reports warning/error, number of running vm's changed or if the host is overutilized.

VMware Virtual Center Server 6.0

Monitor key VCenter Windows Services

NetWare (SNMP)

NetWare (SNMP)

It requires SNMP to be enabled. It monitors: *CPU Load, Cache buffers, Volumes Free Space, Novell Volumes*

IBM

AIX (SNMP)

Monitor processor, memory and file system

AS/400 (SNMP)

Monitor processor, memory, errors, sessions, RWS controller

Hardware

Network Devices

Alcatel OmniSwitch (SNMP)

Monitor CPU utilization, IO utilization, memory and switch temperature

Barracuda Spam and Virus Firewall (SNMP)

Creates reports about *Inbound Messages, Outbound Messages, and Statistics Report*

Cisco (SNMP) (automatic)

Monitor Memory, Processor Utilization and checks if the hardware components are working properly. Creates *Cisco Device Report*

Device Class must be Hardware Router or Switch

Manufacturer must be Cisco

HP ProCurve (SNMP) (automatic)

Monitors CPU utilization and memory as well as the hardware sensors state. Creates the *Performance Report*

Juniper EX Switches HealthMon (SNMP)

Monitor important Juniper switches performance indicators such as CPU, memory and file storage utilization

Juniper Sensors (SNMP)

Monitors control plane CPU utilization and memory

Juniper SRX (SNMP)

Monitor important Juniper SRX performance indicators such as packet forwarding memory, routing engine CPU usage, temperature, session count and others

Other

APC PowerChute (SNMP) (automatic)

Monitor battery status, capacity and temperature

SNMP must be Enabled

Device class must be UPS

Manufacturer must be APC

Cisco ASA VPN Tunnels (SNMP)

Creates reports about *VPN traffic* and *Session Statistics*

Cisco Unified Communications Manager (SNMP)

Monitor the Call Manager status and unregistered devices in the network

Dell OpenManage (SNMP)

Monitor various statuses and temperature

F5 Local Traffic Manager

Monitor important SNMP traps and variables of the Traffic Manager. Event will be generated if there is a problem with temperature, fan speed, power supply, CPU and memory usage, DOS or Brute Force attack was detected or one of many monitored traffic statistics is higher than usual

Generic CMYK Printer (SNMP)

Monitor printer Ink level

HP System Insight Manager (SNMP)

Monitor overall system health condition, hardware sensors status and interfaces status

NetApp Disk Usage (SNMP)

Monitor disk usage and create reports for *LUN*, *Free Files*, *INodes*, *Deduplication*, *Free Disk Space*

NetApp I/O Operation Reports (SNMP)

Create reports about multiple NetApp I/O

NetApp System Health (SNMP)

Monitor overall system health condition and hardware sensors status. Create *Disk* and *System Health* reports

NetApp Traps (SNMP)

Contain SNMP trap events definitions

Applications

Anti-Virus Software

Below you can find a list of Monitoring Packs which you can add to check if the following Anti-Virus application is running on the monitored node.

- Avast!
- AVG Anti-Virus 2013
- AVG Internet Security 2013
- Avira
- BitDefender Antivirus Plus 2013
- BitDefender Internet Security 2013
- BitDefender Total Security 2012
- BullGuard Anti-Virus 2013
- BullGuard Internet Security 213
- eScan Antivirus Edition v11
- eScan Internet Security Suite v11
- ESET NOD32 Smart Security 6
- F-Secure Anti-Virus 2013
- F-Secure Internet Security 2013
- G Data Anti Virus 2013
- G Data Internet Security 2013

- K7 AntiVirus Plus
- K7 AntiVirus Premium
- K7 TotalSecurity
- Kaspersky Endpoint Security 8
- Kingsoft AntiVirus
- Kingsoft Internet Security 9 Plus
- Lavasoft Ad-Aware Total Security
- Lavasoft Pro
- McAfee Total Protection 2013
- Norman Security Suite 2012
- Norton 360
- Norton AntiVirus 2013
- Norton Internet Security 2013
- Outpost Antivirus Pro 8
- Panda Global Protection 2012
- PC Tools Internet Security 9.0
- Sophos Anti-virus and Firewall
- Symantec Backup Exec Server
- Symantec Backup Exec Remote Agent
- Symantec Endpoint Protection Client
- Symantec Endpoint Protection Server
- Symantec NetBackup Client
- Symantec NetBackup Server

- Trend Micro Titanium 2013
- Vipre Antivirus 2013
- Vipre Antivirus Enterprise
- Vipre Antivirus Premium
- Webroot SecureAnywhere Antivirus 2012
- Webroot SecureAnywhere Antivirus 2013
- Webroot SecureAnywhere Essentials 2012
- Windows Defender
- ZoneAlarm PRO Antivirus + Firewall 2013

Microsoft

Exchange 2003

Monitor key Exchange Windows services, monitor Windows event log for Exchange event errors and watch the important performance metrics such as mailbox or SMTP queues

Exchange 2007-2010

Exchange 2007-2010 Client Access Server

Monitor key Windows services and performance counters of the Client Access Server, generates *IMAP4 and POP3 availability report*

Exchange 2007-2010 Mailbox Access Server

Monitor key Windows services and performance counters of the Mailbox Access Server

Exchange 2007 - 2010 Transport Access Server

Monitor key Windows services and performance counters of the Transport Access Server, generates *SMTP availability report*

Exchange 2013

Exchange 2013 Client Access Server

Monitor key Windows services and performance counters of the Client Access Server, generates reports about *IMAP4, SMTP, UMCallRouter and POP3*

Exchange 2013 Mailbox Access Server

Monitor key Windows services and performance counters of the Mailbox Access Server

Forefront TMG 2010

Monitor key Windows services and performance counters of the TMG 2010 such as server cache, number of denied packets, web proxy requests and other

IIS

Monitor key IIS performance metrics such as ASP requests, IIS Private Bytes and monitor windows event log for ASP, SMTP and WWW errors

ISA Server

Monitor key performance metrics, Windows Services and ISA Server event log errors

MS BizTalk Server 2009/2010

Monitor key Windows Services of the BizTalk Server

Ms Dynamics AX 2012 Server

Monitor number of active sessions and Windows Service of the Ms Dynamics Server

MS Dynamics CRM 2011 Server

Monitor key performance metrics, Windows Services and MS CRM errors

MS Dynamics NAV Server

Monitor key Windows Services of the MS dynamics NAV Server

MS Index Server

Monitor status of Microsoft Indexing Service

MS Project Server

Monitor key Windows Services of the MS Project Server

MS SQL Server

Monitor key performance metrics, Windows and Network Services, MS SQL event log warnings and errors. Also contains several reports such as:

Processor Bottleneck Analysis, Disk Usage and Performance, Memory Usage Analysis, MS SQL Server CPU Report, MS SQL Server I/O Report

SharePoint

Watches the status of SharePoint Windows Services, number of rejected requests, cache size and number of queued requests

Other

AdRem NetCrunch Server

Monitor windows services of the NetCrunch Server

Apache Server (automatic)

Reports: *Apache Server Report, Detailed Report of Apache servers*

Apache Monitoring Engine must be enabled

APC Windows Events

Monitor specific APC windows event log events

ARCServe

Monitor specific ARCServe windows event log events

Avaya Modular Messaging Server

Monitor key Messaging Server Windows Services

Blackberry Enterprise Service (BES10)

Monitor key BES10 Windows Services

CiscoWorks Lan Management

Monitor key Lan Management Windows Services

Citrix Xen App 6.0 Server

Monitor key Citrix Xen App Server Windows Services

Java Application Server (SNMP)

Generate report about *memory, threads and classes usage*

Lotus Domino Server (SNMP)

Monitor key performance metrics such as database cache, number of pending and undeliverable mail messages, routing failures and other

Oracle 11.2.0

Monitor key performance metrics such as buffer cache miss ratio, log buffer, Windows Services and other

Squid 3 (SNMP)

Monitor cache hit ratio and CPU utilization

Veeam Backup and Replication Server

Monitor key Veeam Backup and Replication Server Windows services

Global

Settings for all nodes in the Atlas.

Correlations

Allows define alerts correlating multiple alerts from multiple nodes

Global NetFlow

Here you can specify NetFlow thresholds for your entire network

NetCrunch

Generate an event when node was added or deleted from Atlas

NetCrunch Audit

Enables tracking users access to NetCrunch Console and failed login attempts to the Console.

NetCrunch Self Monitor

Tracks various NetCrunch Server parameters and triggers notifications to administrators if any problems occur

Network Traffic (SNMP)

Collects data for traffic statistic and Top Charts

Node Status (automatic)

Generate an event when node state changed to up or down

This monitoring pack is automatically assigned to all nodes in the Atlas

Open Monitor

Here you can define events based on the data gathered by NetCrunch Open Monitor

Service Status (automatic)

Monitor connection reliability and network services status

This monitoring pack is automatically assigned to all nodes in the Atlas

Glossary of Terms

Network Atlas

Network Atlas is a central database containing all your network data. It's organized by the hierarchy of the Atlas Node Views.

Atlas Node View

Atlas Node View shows various aspects of the group of nodes in the Network Atlas and consists of multiple pages such as nodes, maps, dashboards and other.

Monitoring Issues

Monitoring Issue is a problem related to the monitoring process, like missing credentials or improper response received from the device.

Node

Node is a single address network endpoint (interface).

Monitoring Pack

Monitoring Pack is a group of performance parameters and events to be monitored and collected for the reports.

Monitoring Engine

Monitoring Engine is a software component responsible for specific type of monitoring.

Monitoring Dependencies

Monitoring Dependencies reflect network connections and allow for preventing false alarms and disable

monitoring of unreachable network components.

Threshold

Threshold is the limit or a boundary point that must be exceeded or dropped below to trigger some response action.

Performance Triggers

Performance Trigger generates an event upon the condition set on performance counter value.

Alert

Alert - the condition being watched for an action as the reaction to potential danger or to get attention.

Event

Event is the description of the thing that happens or takes place, especially one of importance.

Service Think Time

Service Think Time is an estimate of the time the service spent on generating the response. It's calculated by subtracting an average PING RTT from the total request time.

Event Suppression

Event Suppression is the technique of preventing false alarms caused by intermediate network connection failure.

Leading Service

Leading Service is a network service designed to be checked as the only service, when the node is DOWN.

Monitoring Sensor

Monitoring Sensor is a software module focused on monitoring single object, service or device (web page, file, folder, query, etc.).

Program Reference

Administration Console

External Events Window

The window allows to see all incoming traps and syslog messages and define alerts with a single click.

(XE Only)

The window shows incoming buffers for SNMP traps, syslog and web messages. It contains all message and traps sends regardless the sending node is added to Atlas or not.

The screenshot shows the 'External Events' window in the Administration Console. The window has tabs for 'Syslog Messages', 'SNMP Traps', and 'Web Messages'. A search bar and an 'Enabled' button are visible. A message table is displayed with columns: Monitored, Received, Atlas Node, Node Identification, Received From, and Data. Two messages are shown: one received 24 minutes ago and another 25 minutes ago. A 'New Alert Rule for 192.168.10.1' dialog box is open, showing a 'Description' field with a 'Warning' level and 'operational' category. The 'Event Condition' section has a text area containing 'error'. The 'Filter' section is set to 'Match incoming data where all of the following apply' and includes a condition 'error is equal to 1'.

Monitored	Received	Atlas Node	Node Identification	Received From	Data
	24 minutes ago Saturday October 8, 2016 1:16:14pm	(192.168.10.1)	192.168.10.1	192.168.10.112	Some message
Set Alert	25 minutes ago Saturday October 8, 2016 1:14:58pm	(192.168.10.1)	192.168.10.1	192.168.10.112	["error":1]

This allows creating alerts in one click. Just send data to NetCrunch and decide which turn in to alerts. Program will automatically add node to the Atlas if necessary and configure alert based on received event.

Each buffer contains last 1000 messages.

Device Types

➤ [Tools > Device Type Manager](#)

NetCrunch Device List Editor displays the table lists all currently defined network devices recognized by device group.

The program offers an easy way of updating your device list directly from the AdRem Software Website:

- New version of this list will be periodically updated by AdRem based on information provided by NetCrunch clients.
- You can also send definitions that you have created to AdRem so that other NetCrunch users can update their device lists.

To update the device list click the *Update* icon and follow the directions specified in the Device Update wizard.

Each device contains the following information:

Icon

Specifies the icon to be used by the device in NetCrunch. The icon with this name and corresponding image must directly relate to one of the defined in the list in the Properties window of Node settings ([➤ Node settings > Properties > Default icon](#)).

During network scanning, when NetCrunch recognizes a device based on the sysObjectID value, it will use the particular icon specified here to display it in the *Network View* window.

Name

Specifies the name that will be associated with the device in NetCrunch.

SysObjectID

Specifies the MIB object identifier of the device (based on the unique sysObjectID value). If an incorrect value is filled in, NetCrunch is not able to discover and distinguish the device during the scanning process and add the corresponding icon on a map in the *Network View* window.

Match String

Specifies short information related to the device based on sysDescr value. Some devices may be recognized by NetCrunch based solely on their sysDescr value instead of the SysObjectID.

Options: General

[➤ Tools > Options > General Options](#)

The *General Options* page allows modifying such program settings as error reporting, changing password, setting parameters used during network discovery process, managing remote access licenses or update notifications.

Windows and fields descriptions

Server

Allows the configuration of basic NetCrunch Server connection settings like password or port number. The unique password protects the server from unauthorized or accidental connections.

Password

The password used when connecting via *Administration Console*.

Port

The port number on which the *Administration Console* is connecting to NetCrunch Server.

Web Access

The HTTP port to use when connecting to NetCrunch using Web access.

Use Open SSL for encryption

Enables encryption to secure NetCrunch remote access from desktop browsers and mobile devices.

Key file

The name and path of the Key File to be used for SSL connection. You may click the *Browse* icon to search for it.

Certification file

The name and path of the Certification File to be used for SSL connection. You may click the *Browse* icon to search for it.

Root certificate file

The name and path of the Root Certification File to be used for the SSL connection. You may click the *Browse* icon to search for it.

Startup Scripts

Allows defining a script which will be executed automatically during the NetCrunch startup procedure before NetCrunch is started.

For example, you may want to perform the hard drive mapping before running NetCrunch. When a startup script is defined, it will be executed automatically beginning from the next startup time of NetCrunch but prior NetCrunch is running.

Script File Name

In this field you can select a script file to be executed automatically. NetCrunch allows executing files with following extensions: *.bat, *.cmd, *.wsh, *.js, *.vbs.

Parameters

Allows defining additional parameters to the selected script.

Wait Until Script Finishes

When this check box is selected allows defining a delay time before NetCrunch will resume the startup procedure.

If the script execution process exceeds the delay time, then NetCrunch will interrupt it and resume the startup procedure. The script is executed as an independent process. If this check box is unselected, the script is executed concurrently, regardless of the NetCrunch startup procedure.

Test Script

Tests the selected script.

View Log

Opens the script executing log file (StartupScript.txt). The log file is created during the script execution of the NetCrunch startup procedure.

Clear Log

Clears the log file.

Error Reporting

Allows sending error reports directly to AdRem Software. The error report contains only technical information about the program state and configuration of the computer running NetCrunch.

You may also indicate your email address so that AdRem Software will be later able to contact you to fix the problem.

Confirmations

Contains the list of messages included in a several Confirm dialogs of NetCrunch.

In the State field you can specify what action NetCrunch should perform for each listed message.

Depending on the user selection NetCrunch will perform appropriate action automatically.

By default, all messages are defined with *ASK* option.

You can choose from the following options:

Ask

The described action will not be executed and the program will always ask the user to decide what action should be performed (default option).

Yes

NetCrunch will execute the described action automatically without asking the user.

No

NetCrunch will not execute the described action and the user will not be asked for a decision.

Network Discovery

Changes certain parameters which NetCrunch uses during the network discovery process.

Maximum Node Scan Time

Indicates the maximum time that NetCrunch will have to obtain information about a particular node during network discovery.

Warn Before Scanning Foreign/Internet Networks

Setting this checkbox warns before scanning foreign/internet networks.

Warn before scanning large subnets

Setting this checkbox warns before scanning large subnetworks.

Try Count

The maximum number of attempts that NetCrunch should use to send packets using ICMP before considering the node non-existent in a specific network.

Timeout

The maximum time (in milliseconds) that NetCrunch should wait for a reply from the node to which the packet was sent using ICMP, before considering it lost.

Timeout

The maximum time that NetCrunch should use to obtain information about the node using SNMP.

Maintenance

Allows configuring several program settings related to atlas backup and program maintenance.

Max Event age

The number of days after which generated events older than this value should be deleted from the Event Log database.

Max performance trend data age

The number of days for which trends are to be kept.

Max Remote Access Audit Log age

The number of days after which Remote Access session logs should be removed.

Backup data location

The path where the backup files are to be stored. You can use the *Browse* icon to browse for a specific path.

Event Database

Specifies the number of events per page displayed in the *Event Log* window and time interval of verifying the integrity of the events database.

Number of events per Event Log page:

Specifies the number of events per page displayed in the *Event Log* window.

Verify database at every startup:

Specifies the time interval of verifying the integrity of the events database.

Trend Export

Enables automatic trend export to SQL databases.

Update Notification

Allows enabling/disabling automatic check for updates as well as setting proxy server options.

License Manager

Manages the program and remote access licenses.

Reports

Allows setting properties of emails with reports such as the maximum size of the email and the footer signature.

See also:

[Configuring Notification Services](#)

Read to configure emails and text messages (SMS) with NetCrunch notifications.

[Options: Monitoring](#)

Options: Monitoring

[Tools](#) ▶ [Options](#) ▶ [Monitoring](#)

General

NetCrunch Node address

Allows selecting the NetCrunch node address from the list of local interface addresses. Since the monitoring of all nodes in the atlas depends on NetCrunch node, therefore other nodes will be monitored only when selected interface in the NetCrunch Node Address field is enabled and working properly.

Do not trigger "Node is DOWN" alert for newly discovered nodes

Full monitoring of newly discovered nodes starts only after the node has responded to at least one monitoring request.

Automatically change NetCrunch node

In the case that the selected interface becomes unavailable, the program changes the NetCrunch node automatically to another available interface. Using the DHCP service can cause this change in the local machine IP address. In this case, it is recommended to select the Automatically Change NetCrunch Node check box.

Use WINS to resolve node properties

Forces the program to use the WINS to resolve properties of the node.

Nodes Defaults

Specifies the default settings for node identification method and monitoring time.

Identification method

A drop-down list that enables selecting how NetCrunch should identify nodes by default: Auto, IP address or DNS Name.

Monitoring time

Default monitoring time that will be automatically set for any newly discovered or inserted nodes.

SNMP profile

SNMP profile saves the Read/Write communities, SNMP authentication credentials and encryption password, depending which SNMP version is implemented for the node. The SNMP Profile drop-down list allows selecting the correct SNMP profile for the node. You can also click the *Manage SNMP Profiles* icon, next to the Profile field to create a new SNMP profile.

Port

Specifies the port used for SNMP monitoring on the selected node.

Timeout

Specifies the maximum time in milliseconds NetCrunch should wait for a reply from SNMP before timing out.

Retry count

This parameter specifies how many requests should be sent if SNMP does not respond correctly (timeout, SNMP error, etc.).

Data collection time

The exact time used for collecting inventory data.

Auto Discovered Services

Displays a list of services automatically discovered for nodes. It also enables adding network services to the list or removing them.

Network Services

Displays a list of network services that can be monitored at an extended level. It also enables adding/deleting network services to/from the list and editing their parameters.

Configuration of the extended monitoring level is performed in a set of different windows which contain specific parameters related to the type of the configured service.

The following network services can be configured and monitored at the extended level:

DNS

Domain Name System - service is used for name resolution of domain names on the

network.

FTP

File Transfer Protocol - service is used to transfer files over the network.

HTTP

Hyper Text Transport Protocol service is used for communication and transfer of information on intranets and the WWW servers.

HTTPS

HTTP service running on Secure Sockets Layer (SSL).

POP3

Post Office Protocol 3 service is used to hold incoming emails over the network.

SMTP

Simple Mail Transfer Protocol service is used to transfer and forward email messages over the network.

Physical Segments

Enables the Physical Segments Monitoring. Once, the monitoring is enabled you can choose several analysis methods that can be used to create the most accurate physical segment topology.

Analysis Options

Processing of forwarding tables

The base method that is always used.

Spanning Tree (STP) Tables

The STP is used in switched networks to prevent loops.

Cisco Discovery Protocol (CDP)

The CDP is used to share information about other directly connected Cisco equipment. It runs on most Cisco devices.

SONMP

SynOptics Network Management Protocol (SONMP). It supplies topology information of network devices that support SONMP. The SONMP is implemented in SynOptics, Bay and Nortel Networks devices.

Hide inactive nodes

By default, NetCrunch is trying to remember the last position of all nodes. If this check box is selected, the down nodes will not be shown on Physical Segments Maps.

Data refresh time

Specifies the time interval when NetCrunch will collect information about the physical topology of the monitored atlas. You can specify a different time for refreshing all physical segments in the monitored atlas from the time of monitoring services.

Since the topology of physical maps is not changing frequently it is suggested to define a

longer period. All maps in the Physical Segments section of the Atlas Views window will be updated if any change is discovered.

Map Layout Options

When you enable the displaying of physical segment topology maps of your network via a wizard, NetCrunch creates the physical representation of the network. You can also specify some layout elements of the created maps:

Port Name Style

Specifies the style of the port name on maps.

Port Box Style

Specifies the style of the port shape on maps.

Sort Ports By

Allows sorting ports by number or name.

SNMP Traps

Listen to SNMP traps

Allows the program to listen to SNMP traps on a specific port, and possibly redirecting such SNMP traps to a different node in the network

Traps Group Time

NetCrunch uses the mechanism of grouping identical SNMP trap messages received during monitoring. The field specifies the number of seconds when received messages will be combined into. The program default is 15 seconds.

Syslog

Listen to syslog messages

Allows the program to listen for incoming Syslog messages so that they can be processed as NetCrunch events or redirected to other nodes. Clearing this check box will disable this feature.

Redirect syslog messages

Allows the program to redirect incoming Syslog messages to a remote host for processing.

Group the same messages

NetCrunch uses the a mechanism of grouping identical Syslog messages received during monitoring.

In this field you can specify the time when received messages will be combined. The program default is 15 seconds.

Windows Event Log

Reconnection Time

Specifies the reconnect time interval when NetCrunch connects to the Windows machines selected in alerting and gather information specified in created events.

Group the same log entries

Combines the identical entries into one.

Grouping time-frame

Specifies the number of seconds into which the received entries will be grouped.

NetFlow

Enable NetFlow traffic monitoring

Enables NetFlow i sFlow monitoring on selected ports.

NetCrunch Open Monitor

Displays a list of external data types monitored by the program. You can add/delete data source types and edit their properties.

Advanced

DNS Resolver

Use Direct DNS Name Resolver

Displays the current information about DNS name for nodes in the DNS Name column in the Map window.

Flush Name Cache

When this button is used, the cache memory of the DNS names is cleared and NetCrunch will query the DNS server to resolve DNS names.

It is important when the DNS server configuration in the monitored network is changed.

HTTP/S Proxy

Proxy settings can be used to send HTTP/S requests by Web Page Sensor and HTTP/S sensor.

See also:

[Configuring Notification Services](#)

Read to configure emails and text messages (SMS) with NetCrunch notifications.

[Options: General](#)

Options: Notification

➤ [Tools](#) ▶ [Options](#) ▶ [Notification](#)

Notification Window

Enables changing alert dialog settings such as a display window, flash, close screen saver, play sound, number of last alerts kept.

Email

In the program, you have an option to use the built-in SMTP server or an external SMTP mail server for sending email notifications in the alerting process.

Reply Address

The reply email address to be used by the program when sending emails.

Use External SMTP Mail Server

Enables usage of an external email server for sending NetCrunch email notifications (instead of the built-in SMTP server).

Mail Server

The name of the external email server that you plan to use for the NetCrunch alerting process.

Port

The port of the external SMTP mail server.

Server Requires Authentication

Select this check box if the external SMTP mail server requires authentication.

Username

The user name to login to the external SMTP mail server.

Password

The password to login to the external SMTP mail server.

GSM Device

Enables notifications via a mobile phone connected to the computer running NetCrunch via a COM port. You may use a standard cable attachment, Bluetooth, IRDA or any other type of connection - as long as it uses one of the computer COM ports.

You must define the settings in NetCrunch for the GSM mobile device.

Select the COM port to be used, define port parameters and other options related to the GSM device such

as additional initialization AT+C commands.

COM Port

Shows the COM port selected for communication with the mobile phone. Clicking the Browse button opens the *GSM Device Discover* dialog, where you may choose/configure the appropriate COM port.

SMS Settings

Split messages longer than 160 chars

Splits the message over 160 characters into two or more notification messages.

Modem Settings

Initialization AT+C Commands

Select this check box if you plan to send initialization AT+C commands. Enter the commands in the field below.

See also:

[Configuring Notification Services](#)

Read to configure emails and text messages (SMS) with NetCrunch notifications.

[Options: General](#)

Options: Map

➤ [Tools](#) ▶ [Options](#) ▶ [Map](#)

The map options allow specify such program settings as:

- Snapping node icons to a grid, using transparent edit dialog boxes and selecting the program default drawing scheme,
- Selecting the default icons for specific node types,
- Changing the information related to node captions on a map,
- Defining new styles of map objects (background shape and text),
- Changing the default background area of a map,
- Changing the options relating to connection lines,
- Changing the signaling method of nodes (used to display the status of network nodes being

monitored),

- Changing the image cache memory,
- Selecting map links to be shown as preview images in the desired sections of the monitored atlas.

General

Snap to grid

Allows objects in a map to snap to a grid (its increment size is specified in the fields below) when they are moved while map editing.

Icons

Enables changing the properties of a selected node icon. NetCrunch automatically resizes icons to the predefined size and colors icons depending on the status. It also enables adding new icon types.

Captions

The program automatically displays the DNS name of a node below its icon on a map as caption. In this page you can change what is displayed below the node icon to one of the following:

SNMP System Name

This is the *System Name* field read from the SNMP agent running on the node.

IP Address

The program will display the node IP address.

DNS Name

The program will display the DNS host name of the node.

You can additionally select to always add the IP address of the node to the caption.

When zooming out from a map, its scale decreases to less than the default 100%. The program automatically hides node icons and draws them as colored rectangles when the map scale is less than 50%.

You can change this setting to a value other than 50% or disable hiding node icons and replacing them with rectangles, altogether. In such a case, when zooming out, the icons will continue to be decreased in size.

Visibility

Enables selecting either the Normal or Autohighlight mode for displaying node icon captions.

Opacity

If the Autohighlight mode was selected, use the *Opacity* sliding bar to select the proper opacity level for displaying captions of node icons that currently do not need attention: none, low, normal, high.

Use fade effect while hiding

Selecting this check box tells the program to utilize fade effect while hiding the caption under a node icon.

Styles

Enables adding new shape types and editing their properties. You can change only the properties of styles and shapes that you added.

Background

Enables changing the background settings for all maps in the atlas and default map margins:

Type

A drop-down list that allows selecting background type: solid color, texture or gradient. You can choose between default and custom colors.

Connection Lines

Enables changing settings for connection lines displayed on the maps in the program.

Several options related to connecting lines can be modified:

- Dash style.
- Thickness of the line.
- Color of the line.
- Diagonal connection type (oblique, rectangular, data bus).

Signaling

Specifies the signaling method for the current state of a node:

- colorize icon
- drawing a background color behind the icon,
- drawing a frame color around the icon.

Flash the icon

When it is enabled (the program default), an icon flashes for a specified amount of time if its status changes to a worse state (for example, from OK to WARNING or from WARNING to DOWN).

Flash the line

Permits connection lines on physical maps (representing real physical cables) to flash when they are disconnected from devices such as bridges.

Time

The exact amount of time that icons and physical lines flash may be changed to values between 300 ms and 12 seconds.

Maximum Levels of Map State Calculation

Specifies the number of hops (via map links) through which the map link will display the occurrence of any issue.

If 0 is selected, it will only indicate the status of the map it is linked to. In this case, the map icons in the Atlas Views window will change color only for the maps containing nodes with issues and the links to these maps will be added to the Maps with Issues section located in the Favorite Maps window.

By selecting 1 or more, the map links and map icons (in the Atlas Views and Favorite Maps) will indicate the status of nodes and will pass through a number of map links (hops) specified in this field.

Image Cache

Maximum cache size

Specifies the maximum size of this special image cache.

Minimum cache size

Specifies the minimum size of the image that can be cached.

Links

Defines the way in which map links are displayed in any section of the Atlas Views window.

You can select whether map links should be displayed as a preview image in a chosen section of the Atlas Views window. Otherwise, map links will be displayed as standard icons.

Appearance

Enables selection of signaling methods for a node state.

NetCrunch can signal a node state by changing the color of the node icon.

When a node or one of its network services is down the program can display overlay text.

The program can also warn the user by displaying additional small, overlaying icons on the node icon which inform about issues, alerts and state of node configuration.

See also:

[Configuring Notification Services](#)

Read to configure emails and text messages (SMS) with NetCrunch notifications.

[Options: General](#)

Additional Programs

iTools - Network Diagnostic Tools

➤ [Tools > IP & SNMP Tools](#)

iTools is the application containing set of usable network diagnostic tools such as: Ping, Traceroute, Lookup, Connection, Scanner and SNMP.

Connection Tools

This utility allows testing reliability and connection bandwidth between a particular host node and a computer running iTools.

Connection tool consists of two separate utilities: one related to bandwidth and the other for reliability of connection to a remote node.

Bandwidth

The tool checks the maximum transfer rate available between the local and other remote node. The transfer rate can be measured in bits or bytes per second (bps or Bps).

Timeout

Specifies in milliseconds how long the tool should wait for a reply from a specified node before considering it lost.

Delay

Specifies in milliseconds the amount of time the program should wait before sending the next packet.

Units

Specifies the units for the bandwidth (data transfer rate) results graph - can be either bytes per second or bits per second.

Data Size

Indicates the size of packets which are sent to the remote node in order to test network bandwidth to that host.

Reliability

The tool helps determine the quality of the link between your and other device. It sends a series of growing UDP packet samples, starting from the initial packet size.

Note:

For remote nodes the maximum size of a used packet should be 1024 Bytes. Some remote hosts may not accept large packets. For large packets increase the time between sending each of them. In local networks, however, a larger packet size improves measurement statistics.

Timeout

Specifies in milliseconds how long the tool should wait for a reply from a specified node before considering it lost.

Delay

Specifies in milliseconds the amount of time the tool should wait before sending the next packet.

Initial Packet Size

indicates the initial size of the packet for the series.

Packet Samples

specifies the number of packet samples to be sent with each size increasing step.

Packet Size Step

indicates the increment that is to be used for the series of packets. The program will begin sending packets starting with the initial size, then increase them in a similar manner until all packets in the series have been sent. The process will be repeated starting with the initial size.

DNS Lookup

Allows viewing information about a host stored on a DNS server by resolving the host names into IP addresses or vice-versa.

Domain Name Server

Indicates the IP address of the DNS server that the tool will obtain information from about the selected node.

Port

Specifies the port number to communicate with the DNS server (53 is used by default).

Timeout

Specifies in milliseconds how long the tool should wait for a reply from a specified node before considering it lost.

Ping

The tool tests devices in a network by sending and receiving test packets to/from another location. It discovers if a particular node is available in a network and whether it is able to communicate with other nodes via TCP/IP.

Timeout

Specifies in milliseconds how long the tool should wait for a reply from a specified node before considering it lost.

Delay Between Packets

Specifies in milliseconds the amount of time that the tool should wait before sending the next packet.

Keep Last Packets

Specifies the maximum number of packets to keep - only the specified number of packets will be shown in the diagram and table panel.

Scanner Tools

Services

The tool discovers network services available on a host. It checks a list of 65 popular services including POP3, SMTP, FTP, HTTP etc.

Ports scanner options

The tool can check ports in a given range or from the predefined list of well known ports, and the list of defined known trojans. Options located in the *Navigation Pane* allow you to change the following features:

Ports List

This drop-down list lets you choose the range of ports to be scanned by the program. Specifically, you can select a start and end port for the range (i.e. ports in the range), scan well known ports or typical ports used by trojans.

Start

Specifies the port number that the scanning should begin with - applies only if ports in the range option was selected.

End

Specifies the port number that the scanning should end on- applies only if ports in the range option was selected.

Network scanner options

The tool allows scanning given network range using one of two methods: ICMP ping or SNMP ping.

Timeout

Specifies in milliseconds how long the tool should wait for a reply from a specified node before considering it lost.

Select Services

Clicking this button lets you choose the list of network services that should be scanned on a

particular node. If you select a check box located next to the name of a particular service, it will be included in the scanning process. If you clear such check box, the network service will be omitted during the scanning process.

Options located in the *Navigation Pane* allow you to change the following features:

Start

Specifies the number of the last part of the network IP address (0.0.0.x) that the program should begin scanning from.

End

Specifies the number of the last part of the network IP address (0.0.0.x) that the program should end scanning on.

SNMP Community

Specifies the SNMP community to be used to obtain information about nodes found on the network (relates to SNMPv1, SNMPv2c and SNMPv3).

Only SNMP Nodes

Selecting this check box tells the program to find only nodes running SNMP agents. If you leave this check box cleared, all nodes will be found, irrespectively if they are running SNMP agents or not.

SNMP

This utility allows obtaining SNMP information from a particular host. A *MIB Browser* tool may be used for this purpose.

SNMP tool consists of two separate applications that allow you to view/set data via SNMP on a specified node.

All information is gathered using the SNMP protocol. SNMP agents must be running on the nodes that you want to view/set information for.

There are two methods of obtaining information using the SNMP tool:

SNMP Info

Obtain selected information in several categories.

MIB Browser

Provides access to the information from the MIB (Management Information Base).

Note:

You can run the *Scanner* tool on the network to find out which nodes are SNMP-manageable. Nodes from this list can be used with the SNMP Info or MIB Browser tools.

Info

SNMP Profile

This drop-down list allows you to select an appropriate SNMP profile for read-only or read-write access to the node.

Device Type

This drop-down list lets you select the type of device for which to use the Info tool - the number of available groups and forms may be narrowed-down, depending on its type.

Packet Timeout

Specifies in milliseconds the maximum amount of time that the tool should wait for a packet before considering it lost.

Retry Count

Specifies the maximum number of attempts to send a packet to the selected SNMP node.

Remote Port

Specifies the SNMP port number of the node that the *Info* tool should communicate with for the purpose of reading/setting data.

Note:

For the *Device Type* drop-down list you can select the Auto Select item to tell the program to automatically select the device type of the node you want to display SNMP information about. This is the default setting.

You can create, edit or delete SNMP profiles to be used with the *Info* tool.

MIB Browser options

SNMP Filters

This drop-down list allows you to select an appropriate filter option.

SNMP Profile

This drop-down list allows you to select an appropriate SNMP profile, for example, read-only or read-write access to the node.

Packet Timeout

Specifies in milliseconds how long the tool should wait for a reply from a specified node before considering it lost.

Retry Count

Specifies the maximum number of attempts to send a packet to the selected node before considering it lost.

Refresh Time

Specifies in seconds how often to refresh the displayed SNMP data using MIB.

Remote Port

Specifies the network port number of the node that the Info tool should communicate with for the purpose of reading/setting SNMP data.

SNMP MIB Compiler

➤ [Tools > SNMP MIB Database & Compiler](#)

NetCrunch delivers a pre-compiled MIB library containing about 3800 MIBs including Cisco, Nortel, 3Com, Alcatel and others.

You can add new MIBs to the NetCrunch MIB database using the *SNMP MIB Compiler* program. The compiler is part of the server and should be run from the Administration Console installed on the server.

The program allows you to do the following:

- edit MIB module content,
- compile MIB modules,
- browse MIB module contents by tree, defined variables or defined traps,
- create aliases for a specific MIB module,
- remove a MIB module,
- find MIB object,
- sorting and filtering the loaded MIB module list.

In order to find newly compiled MIB, use search window (CTRL+F) and type in the desired variable.

Device Types Editor

➤ [Toolbar menu > Actions > Manage Device Types](#)

NetCrunch Device List Editor displays the list of all currently defined network device types recognized by device group.

You can update the device types list directly from the AdRem Software Website:

- New version of this list is periodically updated by AdRem based on information provided by NetCrunch clients.

- You can also send definitions that you have created to AdRem so that other NetCrunch users can update their device lists.

To update the device list click the *Update* icon and follow the directions specified in the Device Update wizard.

Each device definition contains the following information:

Icon

Specifies the icon to be used by the device type in NetCrunch. The icon with this name and corresponding image must directly relate to one of the defined in the list in the Properties window of Node settings ([➤ Node settings](#) ▶ [Type](#) ▶ [Default icon](#)).

During network scanning, when NetCrunch recognizes a device based on the sysObjectID value, it will use the particular icon specified here to display it in the *Network View* window.

Name

Specifies the name associated with the device in NetCrunch.

SysObjectID

Specifies the MIB object identifier of the device (based on the unique sysObjectID value). If an incorrect value is filled in, NetCrunch is not able to discover and distinguish the device during the scanning process.

Match String

Specifies short information related to the device based on sysDescr value. Some devices may be recognized by NetCrunch based solely on their sysDescr value instead of the SysObjectID.

See also:

[Configuring Notification Services](#)

Read to configure emails and text messages (SMS) with NetCrunch notifications.