
op5 Monitor administrator manual

op5 Monitor administrator manual
Version 6.1, Rev 1
Author: Professional Services

© 2012 op5 AB

op5, and the op5 logo are trademarks, service marks, registered service marks or registered trademarks of op5 AB.

All other trademarks, service marks, registered trademarks, and registered service marks mentioned herein may be the property of their respective owner(s). The information contained herein is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

op5 web site
www.op5.com

op5 support
www.op5.com/support

op5, and the op5 logo are trademarks, service marks, registered service marks or registered trademarks of op5 AB



Contents

Introduction

About introduction	1
Using this manual	2
About op5 Monitor	3

Backend

Introduction	6
Backend parts	7
Folders and files	8
Tweaks	9
About	9
Ramdisk	9
Enable ramdisk	9

Agents

Introduction	12
op5 NSClient++	13
Installing and configuring op5 NSClient++	13
Plugins used with op5 NSClient++	13
check_nt	13
check_nrpe	14
Configuration files	14
Changing the configuration	15
To change the configuration	15
NRPE	17
Installing NRPE	17
Configuring NRPE	17
Adding commands to NRPE	18
Plugins used with NRPE	19
Novell	20
Installing Novell MRTGEXT	20
More information	20
Windows SyslogAgent	21
Installation	21
Upgrading	21
Configuration	22
Configuring the elementary functions	23

Monitoring objects configuration

Introduction	26
Workflow	26

The basics	28
Start working	28
Submitting changes	28
Save the changes	29
Undo changes	30
Historical Configuration Changes	31
Filter config changes	31
Main objects	33
Required directives	33
Hosts	33
Required directives	34
Services	35
Required directives	35
Contacts	37
Required directives	37
Local users	39
Required directives	39
Access rights in detail	39
Recommended settings	40
Time periods	40
A time period in detail	41
Commands	43
Directives	43
Plugins	43
Groups	45
Host groups	45
Services on Host groups	45
Nested host groups	46
Service groups	46
Contact groups	47
Permission to host and services	47
Using templates	49
How they work	49
Managing objects	50
Before you start	50
Add new	50
Configuration files	50
Help	50
Templates	51
Contacts	51
Adding a contact template	51
Adding a contact	51
Modify a contact	52
Delete a contact	53
Hosts	53
Adding a host with new host wizard	53
Adding hosts with network scan	55
Modifying a host	56
Deleting a host	56
Renaming a host	57

Network autoscan	57
Adding a new autoscan configuration	57
Adding a host to blacklist	58
The result	59
Services	59
Adding a service	59
Modifying a service	61
Test this service	61
Deleting a service	62
Scanning host for network services	62
Scanning a host for snmp interfaces	63
Scanning host for windows services	64
Scan for services using agent	64
Scan for service using WMI	65
Custom Variables	67
Creating a new custom variable	68
Example	68
Dynamic Button	69
Configuration	69
Escalations	70
Adding a host escalation	70
Modifying a host escalation	72
Deleting a host escalation	73
Access rights and contacts	73
Connecting access rights to contacts	73
Make things easy	75
Profiles	75
Creating a Profile	75
Using a Profile	75
Cloning objects	75
Cloning from an existing Host	75
Cloning services	76
Copy objects	76
Propagate settings	77
Bulk delete	78
Time periods	81
Add a time period	81
Macros	82
..... Pre-defined macros	82
Custom macros	82
Writeprotected configuration files	84
Features not supported by Configure	85
Plugins	
Introduction	88
Paths and macros	89
Before you start	90
The plugin interface	91
Status output	91

Performance data	92
Return code	93
Adding your first plugin to op5 Monitor	95
Creating the plugin.....	95
Configuring op5 Monitor to use the plugin	95
Creating a more complex plugin	96
More information	97
Widgets	
Introduction.....	100
The widget basics.....	101
The widget rules	101
File structure	101
Widget Class.....	101
View file.....	102
Writing a simple widget.....	103
Creating the directory structure	103
Writing the widget file.....	103
Writing the view file.....	104
Multiple instances	104
Adding the widget to the widget table	104
Removing a widget	105
Viewing the widget.....	105
Take your widget a step further	106
Packaging your widget.....	107
Creating the Manifest.xml	107
Creating the widget package	107
Access widgets externally	108
Server side setup.....	108
External website setup.....	108
Widget Porting guide.....	110
Java script.....	110
View	110
Controller	111
Extra Settings	114
Multiple Instances	115
GUI skins	
Introduction.....	118
The files and folders.....	119
Make your own skin.....	121
Before you start	121
Install SASS.....	121
Creating your own skin	121
Compiling skins.....	122
Modify report skin	122
Changing what theme op5 Monitor use	122

Making changes in the user interface	123
Changing the logo	123
Creating custom logo per login	123
Change the default font.....	124
User menus	
Customize user menus.....	126
Localization	
Introduction	128
Downloading and starting the tools	129
Adding a new language	130
Changing basic language file settings.....	131
Applying the new language to the server	132
Graphs	
Introduction	134
Graph web front end	135
Collections.....	136
About Collections	136
Creating a new collection.....	136
GUI selection	136
Regex selection	137
Viewing Collections	137
Combined Graphs	138
What is a combined graph?	138
Creating combined graphs.....	138
Viewing combined graphs	139
Graph templates.....	140
Changing Graph template	140
Graph prediction	141
Activate graph prediction.....	141
Business Service	
Introduction	144
Business services.....	145
About business services	145
Creating a new group.....	145
Creating a sub-element.....	146
Rules types	146
Worst state.....	147
Best state.....	147
Simple At least.....	147
At least.....	148
At most.....	148
Scores.....	149
Custom rules.....	149
Publish as a service	149

Reporting	150
Notifications	
Introduction.....	152
How does notifications work?.....	153
Notification filters	153
Notification commands	153
Notification macros	154
Notification e-mail sender	155
Notification skins.....	156
The content of a notification skin	156
Creating custom notification skins	157
Dial-up notification	159
Adding a dial up notification command	159
Configuring the contacts	160
SNMP trap notifications	161
Adding SNMP notification commands	161
Configuring the contacts	162
LDAP Integration	
Introduction.....	164
Default.....	165
LDAP and Active Directory	166
Before we start.....	166
Prepare your domain	166
Connection parameters	166
Example configuration for Active Directory	169
Test your connection.....	170
Apache.....	170
Authorization.....	172
Group rights	172
Expand/Contract authorization categories	172
Select/Deselect all rights.....	173
Lookup user	173
Filter groups	174
Add, delete, rename groups.....	175
Configuration files used by authorization	175
Authorization points	175
Backup and Restore	
Configuration backup tool	182
Backup/Restore actions	182
Backing up the configuration	182
Restoring a configuration backup	182
op5-backup.....	184
About	184
Configuration	184
Create a backup	184

Creating a full backup	184
Creating a custom backup	184
Creating a change arch backup	185
Restoring a backup	185
Verify a backup	185
Deleting a backup	186
Upgrade	
Introduction	188
Upgrading with yum	189
Upgrading with tar.gz files	190
Scalable Monitoring	
Distributed Monitoring	192
Introduction	192
Before we start	192
The configuration	193
Setting up the new distributed monitoring solution	193
Adding a new poller	194
Adding a new host group to a poller	194
Removing a poller	195
Master takeover	195
File synchronization	195
Folder synchronization	196
Access right synchronization	196
One way connections	196
Recovery	197
More information	197
Load balanced monitoring	198
Introduction	198
Before we start	198
The configuration	199
Setting up the load balanced solution	199
Adding a new peer	199
Removing a peer	200
File synchronization	200
Folder synchronization	201
More information	201
Merlin	202
About	202
Merlin components	202
merlin-mod	202
merlind	202
merlin database	202
Illustration	202
The mon command	204
About	204
The commands	204
Start	204
Stop	204

Restart	204
Ascii	204
Check.....	205
db	206
ecmd	207
Log	207
Node	208
oconf	210
SSHKey	211
Sysconf	211
Test.....	211
VRRP.....	213
About	213
Setup	213
Activate VRRP	214
.....	215
op5 Monitor API and CLI	
Introduction.....	218
REST API	219
About	219
Example.....	219
REST and LDAP	220
GUI API	221
Configure API.....	222
op5 Monitor Configuration CLI.....	223
To execute the op5 Monitor CLI	223
Wiki	
Introduction.....	226
Managing wiki pages.....	227
Create a wiki page	227
Deleting a wiki page.....	227
Third party configuration import	
Introduction.....	2
Pre-requirements	2
Limitations.....	2
Import configuration.....	3
Preparing nagios configuration	3
Import nagios configuration	3

Introduction

About introduction

This chapter covers the following topics:

Subject	Page	Subsections
<i>Using this manual</i>	2	
<i>About op5 Monitor</i>	3	

Using this manual

This manual includes information about the more advanced parts op5 Monitor and its components.

The manual is written with the goal to give the reader help about how to manage the different parts of op5 Monitor.

This manual is targeted for a technical audience. The manual covers how to use and configure op5 Monitor through its web interface. For configuration using direct console access or SSH, see the op5 System manual.

About op5 Monitor

op5 Monitor is a highly flexible monitoring system for monitoring of IT infrastructure. op5 Monitor is based on the widely known open source monitoring system Nagios.

op5 Monitor is used and configured in a web interface using any standard browser. The most common browsers Internet Explorer, Firefox and Opera have been tested.

The interface is protected by using both authentication (username and password) and by SSL which enables a secure manner for accessing the web interface using encryption.

Backend

Introduction

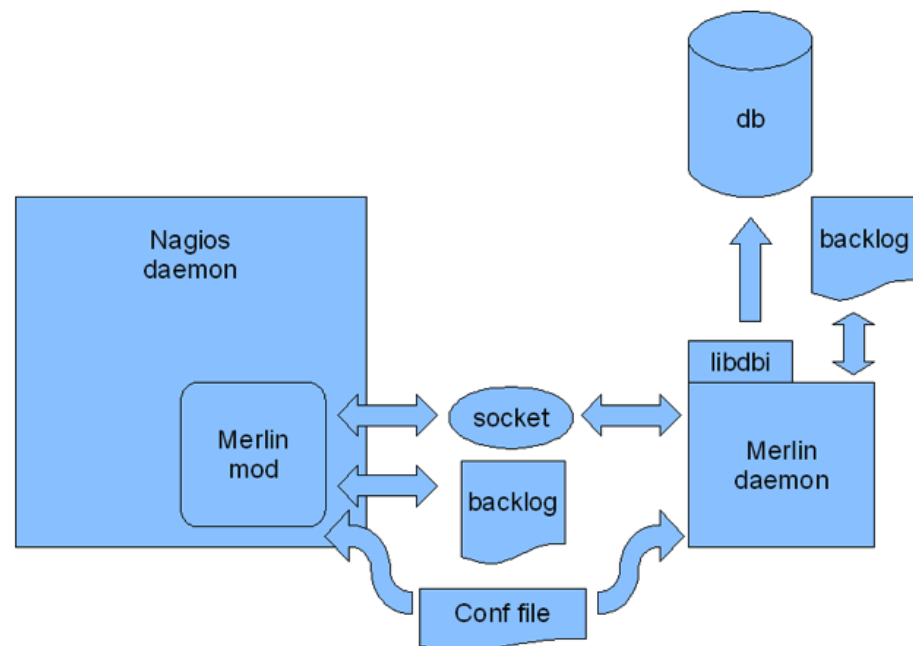
The op5 Monitor backend is called Merlin (Module for Effortless Redundancy and Load balancing In Nagios). It was initially started to create an easy way to set up distributed Nagios installations, allowing Nagios processes to exchange information directly as an alternative to the standard nagios way using NSCA.

When starting making our own GUI for op5 Monitor, called Ninja, we realized that we could continue the work on Merlin and adopt the project to function as backend for the new GUI by adding support for storing the status information in a database, fault tolerance and some other things. This means that Merlin now are responsible for providing status data, acting as a backend, for the op5 Monitor GUI.

Backend parts

- **merlin-mod**
Responsible for jacking into the NEBCALLBAC_* calls and send them to a socket. If the socket is not available the events are written to a backlog and sent when the socket is available again.
- **merlind**
The Merlin daemon listens to the socket that merlin-mod writes to and sends all events received either to a database of your choice (using libdbi) or to another merlin daemon. If the daemon is unsuccessful in this it writes to a backlog and sends the data later.
- **merlin database**
This is a database that includes Nagios object status and status changes. It also contains comments, scheduled downtime etc.

Figure 1 Describes the workflow of the op5 Monitor backend



Folders and files

The main parts of op5 Monitor is located in `/opt/monitor` in the file system.

Table 1 describes the folders in `/opt/monitor`

Folder	Description
<code>bin/</code>	The monitor daemon binary.
<code>etc/</code>	The op5 Monitor configuration files.
<code>op5/</code>	All o5p specific add-ons.
<code>sbin/</code>	The old CGI files.
<code>share/</code>	Main parts of the old GUI based on the CGIs.
<code>var/</code>	Most logs and the cmd pipe all commands are sent to.

Tweaks

About

Tweaking your system to improve performance can be a good way to use you hardware more efficient.

Ramdisk

A ramdisk can be enabled for storing spools for performance data and checkresults.

By storing these spools on a ramdisk we can lower the disk I/O significant.

Enable ramdisk

To enable ramdisk, see [Ramdisk](#) in Administrators Manual.

Agents

Introduction

Most of the monitoring in op5 Monitor is used with the help of agents. The plugins are contacting the agents and let them do the job.

There are mainly four agents available for download at the op5 support site.

Agent	Environment
op5 NSClient++	Microsoft Windows
NRPE	Unix/Linux
MRTGEXT	Novell
Windows SyslogAgent	Microsoft Windows

op5 NSClient++

This is the agent used for monitoring Windows type operating systems.

This agent has the ability to function as a drop in replacement for NSClient providing the same features as NSClient combined with the ability to execute scripts on the monitored Windows server.

op5 NSClient++ runs as a service under

- Windows 2000
- Windows XP
- Windows 2003
- Windows 7
- Windows 2008.
- Windows 2008 R2
- Windows 2012

Installing and configuring op5 NSClient++

The installation and configuration of op5 NSClient++ is covered in detail in the op5 video tutorial **How to monitor windows servers**:

<http://www.op5.com/support/documentation/video-tutorials>

Plugins used with op5 NSClient++

There are mainly two plugins that is used to communicate with op5 NSClient++:

- check_nt
- check_nrpe

check_nt

This plugin is used for all basic tests like

- cpu
- memory
- disks

But it can also be used to check

- Windows services
- performances counters

Note: The preferred way is to use the check_nrpe_win commands.

check_nrpe

check_nrpe can also be used in the communication with op5 NSClient++. This one is normally used when you are performing checks on the Windows server with custom scripts.




Configuration files

NSClient++ operation is configured in a couple of plain text files called:

- NSC.ini
- op5.ini
- custom.ini

They are located in the install directory.

Table 1 Description of the configurations files

File	Description
NSC.ini	<p>This is the standard configuration file. This contains the default settings for NSClient++</p>  <p>This file might be overwritten during an update of NSClient++</p>
op5.ini	<p>This is a op5 specific configuration file. Here are the changes made by op5 entered.</p>  <p>This file might be overwritten during an update of NSClient++</p>
custom.ini	<p>This is where you shall place your own files.</p>  <p>It will never be overwritten during any update of NSClient++.</p>

The default configuration provided is fully functional but there are some options that likely need to be changed.

Changing the configuration

To change the configuration

To change the configuration open the `custom.ini` file using your favorite text-editor (e.g. WordPad). This file is empty and but take a look at `NSC.ini` to view all settings.

Read the `NSC.ini` file carefully to get a complete understanding of all configuration options. Lines starting with `;` (semicolon) are comments.

Before the changes will take effect NSClient++ service must be restarted.

Options most likely in need for configuration are described bellow, section by section.

```
[Settings]
```

```
allowed_hosts=
```

This option lists all servers that are allowed to talk to the agent. Enter the IP-address of the op5 Monitor/Statistics server. If this option is left blank anybody will be able to communicate with the agent.

```
[log]
```

```
debug=0
```

Set debug to 1 to enable debugging. This is normally not needed but can be very useful when debugging.

```
[NSClient]
```

```
port=1248
```

This is the port used for NSClient style requests, i.e. using the `check_nt` plugin. If any other application is already using the default port it might be necessary to change this option.

Note: If a non default port is used you also need to make changes on the op5 Monitor server.

```
[NRPE]
```

```
port=5666
```

This is the port used for nrpe style requests. In order for a minimum of configuration on the op5 Monitor server it's recommended that this option is left with the default value. If this is changed new nrpe check commands using the configured port need to be created on the op5 Monitor server.

```
allow_arguments=0
```

Set this to 1 to enable the possibility to include arguments in nrpe requests. This could be considered a security risk so only enable this if needed. Also, make sure to set the `allowed_hosts` option described above if arguments are allowed.

```
[NRPE Handlers]
```

The nrpe handlers provide a way to execute any custom plugin/check command on the monitored Windows server. In this section you configure all the commands that should be available.

Example 1 *Adding a custom script/plugin to NSClient++*

```
command[my_custom]=c:\mycustomdir\my_prog.exe
```

Or the simplified syntax:

```
my_custom=c:\mycustomdir\my_prog.exe
```

NRPE

NRPE is a Unix client for executing plugins on remote hosts.

It is distributed as

- rpm-packages
- deb-packages
- portable source-code.

NRPE is used in combination with a set of local plugins. By default in op5 Monitor the plugins are placed in:

`/opt/plugins`

There are only a few plugins shipped with the op5 NRPE packages but you may use the ones located on the op5 Monitor server.

Installing NRPE

To install NRPE

- 1 Download the package for your environment from the download section at the support site at www.op5.com
- 2 Put the package to the host you like to install it on.
- 3 Install the package the same way as you do normally with packages on that host.


Configuring NRPE

Before we can start use the NRPE agent for monitoring with op5 Monitor we need to configure the agent.

The NRPE agent is located in:

- `/etc/nrpe.conf`

Table 2 NRPE main configuration file settings

Setting	Description
server_port	The port NRPE should listen on. Default: 5666
allowed_hosts	Add the IP of you OP5 Monitor server on this line multiple addresses can be separated with , ie: allowed_hosts=1.2.3.4,1.2.3.5  Make sure you do not add any space between the comma (.). Default: empty
nrpe_user	The user the NRPE daemon is executed as. Default: nobody
nrpe_group	The group the NRPE daemon is executed as. Default: nobody
debug	Set to 1 if you need to debug the NRPE. Default: 0
command_timeout	The default time out, in seconds, a check shall have. Default: 60
dont_blame_nrpe	Set to 1 to be able to send arguments to NRPE. Default: 0

Adding commands to NRPE

NRPE comes with a few predefined commands. Those commands are located in:

/etc/nrpe.d/op5_commands.cfg

You may add your own commands and you should do that in your own file in:

/etc/nrpe.d/




You must set .cfg as extension to your configuration file or else it will not be loaded into NRPE when the daemon is restarted.

NRPE command definition

The NRPE command definitions is divided into two parts.

Table 3 NRPE command parts

Part	Description
<code>command [name]</code>	<p>The string between the square brackets will be the name of this command. The name is used when you executes the command with <code>check_plugin</code>.</p>  <p>Do not use space in the command name.</p>
<code>/opt/plugins/...</code>	This is the command line used to execute the plugin you are going to use in your command.

To add a command to NRPE

Here we will add a command that is looking for a process named `smstd` using the plugin `check_procs`, which is installed by default.

- 1 Login to the host you have installed NRPE on as root user over ssh.
- 2 Create a new configuration file and open it up with your favorite editor.
- 3 Add a command line looking like this:

```
command[proc_smstd]=/opt/plugins/check_procs -w 1: -c 2:2 -C  
smstd
```
- 4 Save the file and restart NRPE:

```
service nrpe restart
```

Plugins used with NRPE

The only plugin used with NRPE is

- `check_nrpe`

To use the plugin with the command defined in [Adding commands to NRPE](#) on page 18 you shall use the following command line in your service definition:

```
/opt/plugins/check_nrpe -H $HOSTADDRESS$ -C proc_smstd
```

Novell

MRTGEXT was originally written as an NLM for Novell Netware to obtain values used with the widely known MRTG (predecessor of cacti), but it can also be used to poll values from op5 Monitor.

Installing Novell MRTGEXT

To install this extension, simply copy the MRTGEXT.NLM to each NetWare server's SYS:SYSTEM directory that you wish to gather statistics from. Then edit the server's AUTOEXEC.NCF to "LOAD MRTGEXT" so it will load each time the server is restarted.

The MRTGEXT.NLM has three command line switches available:

- `-port=<port>`
will change the port that MRTGEXT listens on for statistic requests. By default, MRTGEXT will use port 9999. For example, to have MRTGEXT use port 1023, add `-port=1023` to the load line. If you change the port number on the command line, be sure to modify the perl script as well.
- `-debug`
will enable some debugging output to the System Console screen. This is only really useful when you are first configuring the extension.
- `-mla=<license>`
For those with an MLA license (mostly for NetWare 5), the MRTGEXT.NLM currently can not obtain a valid value for the server license count. Using this option will tell the MRTGEXT.NLM the license count max to report. This is important if you use the NWEXTCFG.PL to create configuration files or if you use the servstat.pl script. For example, if you have a NetWare 5 MLA license and you really only have a 100 user server, then you would add `-mla=100` to your load command line.

More information

For more information please read here:

<http://download.op5.com/agents/novell/1.46b/readme.txt>

Windows SyslogAgent

op5 SyslogAgent runs as a service under

- Windows 2000
- Windows XP
- Windows 2003
- Windows 2008.
- Windows 2008 R2
- Windows 2012

It formats all types of Windows Eventlog entries into syslog format and sends them to a syslog host (The op5 Monitor server or the op5 LogServer). The agent can also forward plaintext log-files.

Introduction

The entries in the Event log are sent to the op5 Logserver or op5 Monitor server. Text based application logs are also supported.

The op5 SyslogAgent is a repackaged version of the Datagram SyslogAgent, which initially is a bug fixed version of Sabre Net's old NT_Syslog. The op5 SyslogAgent is licenced as GPL software.

Installation

The op5 SyslogAgent installation package consists of an msi installer. To install simply double click the installation msi file and follow the on-screen instructions.

By default the op5 SyslogAgent will be installed in an op5 subdirectory to the program files folder. Usually:

```
C:\Program Files\op5\SyslogAgent\
```

You will also have the possibility to choose if you want to create start-menu, desktop and quick launch shortcuts to the SyslogAgent-configuration tool.

After the installation is completed you will be asked if you want to start SyslogAgentConfig. If you don't do this the agent won't be configured and cannot be started. If you choose to start the

SyslogAgent configuration continue to the section.

Upgrading

If a prior version of the SyslogAgent is installed it should, to avoid problems, be stopped and

uninstalled as a service and then uninstalled. Stopping and uninstalling the service can be done

from the SyslogAgent Configuration tool. Follow these steps to stop and uninstall the

SyslogAgent service:

1. Start the SyslogAgent Configuration tool
2. Press the “Stop”-button (see Fig 3. in the section Configuration)
3. Press the “Uninstall”-button

After the service have been stopped and uninstalled you should uninstall the previous version of

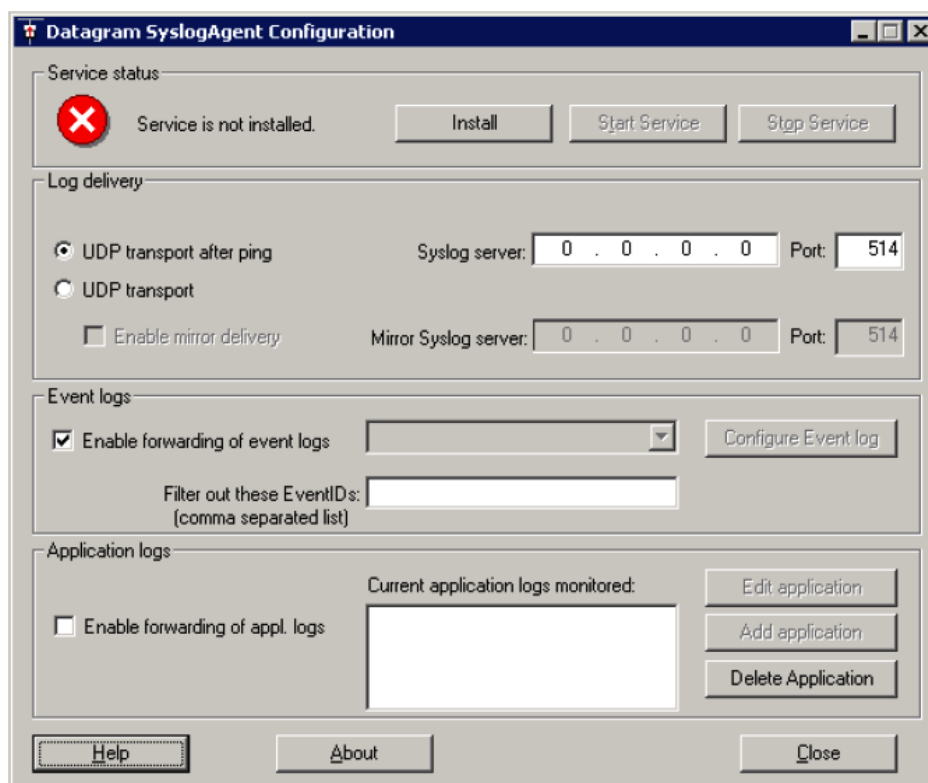
the SyslogAgent from “Add/Remove software” on the windows control panel.

Now you can proceed with the installation of the new version as usual. Note that your previous

settings will be used directly when the installation is complete.

Configuration

When the configuration tool is started the following window should be displayed:



Configuring the elementary functions

To configure the elementary functions and start the SyslogAgent started follow the following steps:

- 1** Press **Install**, this will install the SyslogAgent as a service.
- 2** Enter the IP address in the field **Syslog Server:**. This IP should be the one to your op5 Logserver or op5 Monitor server.
- 3** Make sure the check box “**Enable forwarding of event logs**” is checked.
- 4** Press **Start Service**.

Your SyslogAgent is now configured and should be sending logs to your op5 Logserver or op5

Monitor server.

Monitoring objects configuration

Introduction

There are two ways of changing the configuration of the op5 Monitor:

- Editing the configuration files in `/opt/monitor/etc`.
- Using the web UI **op5 Monitor configuration tool**.

In this chapter we will take a look at how the **op5 Monitor Configuration tool**, from now on called only **Configure**, is used.

Workflow

All configuration in op5 Monitor is saved in configuration files (text files) in `/opt/monitor/etc/`. The Configure works with a database and this makes it possible to do any changes in the configuration without saving it before you are satisfied.

The table below describes the workflow.

Step	Description		
1	Configure opens and the configuration files are compared to the data in the database.		
	If	then	else
	The configuration files are newer than the last change of the database	import the configuration files into the Configure database.	Do nothing besides open up the Configure.
2	Edit the configuration		
3	Save the changes to the Configure database by clicking Submit on the object you just added/changed.		
4	When you are done with editing the configuration save the Configure database to the configuration files by clicking Save .		

Step	Description		
5	A preflight check is made on the configuration before it is exported to the configuration files.		
	if	then	else
	the preflight check finds any error	an error message is displayed and nothing will be exported	the configuration in the Configure database is exported and op5 Monitor is reloaded

The basics

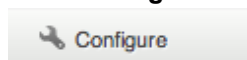
In The basics section we will take a look at the basic step you need to know about when working with **Configure**.

Start working

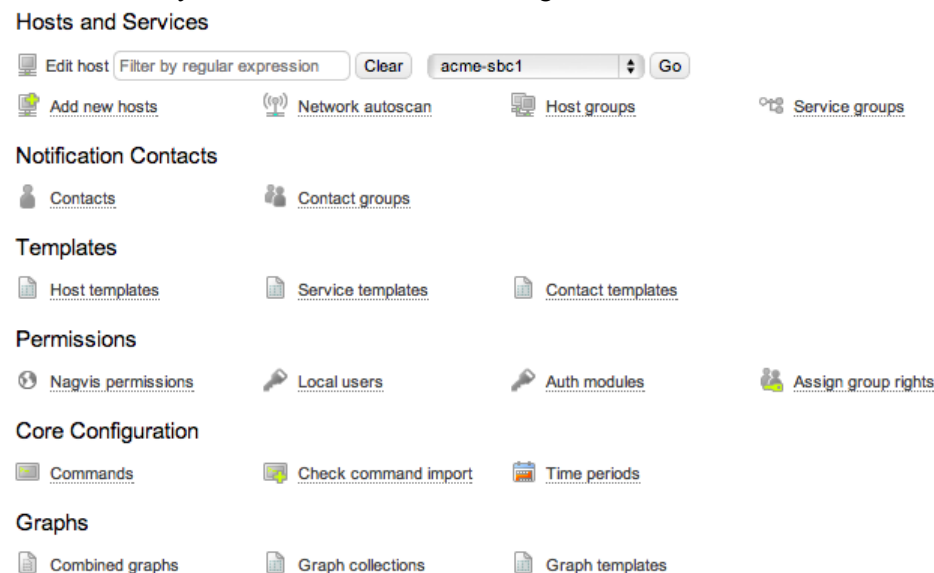
There are many ways to jump in to Configure and start working with the configuration of op5 Monitor.

To start working in Configure

Click **Configure** in the configuration menu



This will take you to the main menu of Configure.



or

Click the **Configure** icon on any object in the monitoring part of op5 Monitor



This will take you directly to the configuration part for the object you clicked on.

Submitting changes

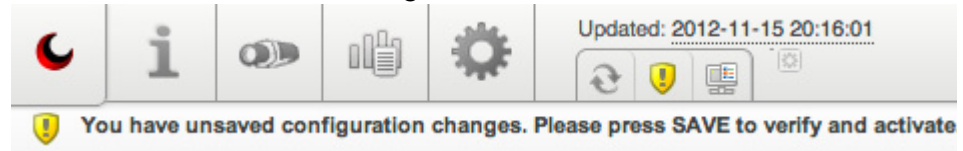
When you have made any changes to an object you have to submit it to the Configure database.

To submit the new configuration to the database

Click **Submit** at the bottom of the page



As soon as the data has been saved you will get the following warning telling you there is unsaved data in the Configure database.



Continue work until your work is done for this time.

Save the changes

When you have finished working and consider your new configuration is ready to be used by op5 monitor you need to save the changes in the Configure database to the configuration files.


This will also make op5 Monitor start using the new configuration.

To save the changes and reload op5 Monitor

Click **save** icon at the top of the page.



Before the configuration is saved to disk, you have the opportunity to review the changes.

 **You are about to save your configuration.**

This will overwrite your current configuration, so make sure you've done everything right.

Changes

There are 1 changes to 1 host objects. [More info.](#)

Do you want to save your new configuration?

To view what changes that will be written to disk click on **More info**.

If you are another users is doing changes on the same objects that you have access to you will save the other users changes as well. The other users changes will be shown under 'More info' as well.

In the screenshot below you will see an example where we created a new host group and jsmith at the same time added a new host.



Other users' changes will be included in the save as well

Be sure to review those changes below before saving.

Changes

There are 3 changes to 1 host objects, 1 service objects and 1 hostgroup objects by you and 1 other user. | info.

- ▶ Created new host object **My_Server** by **jsmith** at 2011-09-15 14:19:23
- ▶ Created new service object **My_Server;PING** by **jsmith** at 2011-09-15 14:19:23
- ▶ Created new hostgroup object **DEMO_group** at 2011-09-15 14:20:04

Are you sure you want to save your new configuration?

Yes, save

No, I am not done yet

When done click **Yes, save** to write all the changes to disk.

Now the preflight check is performed and the data is saved to the configuration files.



Preflight configuration check turned out ok.

Monitor has successfully loaded the new configuration.



Note: if two users with the same permissions are editing the same host all configuration regarding the host or service will be saved.

Undo changes

Sometimes it might be handy to reset the configuration to the state it was in where you started to work in Configure. The only thing you have to do then is to undo your changes.



The undo function will only work as long as you do not have saved the data to the configuration files.

To undo the configuration changes.

Click **undo** icon at the top of the Configure page.



This will revert the your changes since the last successful preflight check.

Changes reverted.
You might want to do a complete reimport?

To undo all users changes click on **complete reimport**. This will re-read the configuration files and all changes will be reverted. If any changes were made directly into the configurations files these changes will now be loaded in to the web configuration

Import forced
Configuration has been imported, overwriting database changes.

Historical Configuration Changes

Historical configuration changes can be used to track changes in the configuration. In the log you will find all changes in the configuration on objects that you have access to.

To access the historical configuration changes log, go to **Configure** and click on the **Historical configuration changes** icon in the upper right corner.



Limited users will only see changes that are made to the hosts and services they are contacts for.

Full access users will see all changes.

Historical configuration changes

User	Object type	Object	Action
monitor	host	router1	Changed attribute icon_image to 'HPij8500p.png' from router40
monitor	service	DNS on host linux-server1	Added member(s) c, o, u, w to attribute flap_detection_option . Removed member(s) from attribute flap_detection_options . Added member(s) c, f, r, s, u, w to attribute notification_option . Removed member(s) from attribute notification_options . Changed attribute notes_url to '/dokuwiki/doku.php/hosts/\$HOSTNAME/\$SERVICEDESC\$'.
monitor	host	win-server1	Changed attribute address to '122.0.0.1' from 127.0.0.1.
monitor	hostgroup	Citrix_server	Created new object Citrix_server
monitor	host	switch1	Changed attribute alias to 'Swith 1 Gothenburg' from Switch 1.

Filter config changes

To filter the configuration changes select **Toggle filter bar**
Historical configuration changes

Toggle filter bar

Using the filter bar it is possible to filter on

- User
- Object Name
- Time

Filter user name: <input type="text"/> <input type="button" value="Filter"/>	Filter object name: <input type="text"/> <input type="button" value="Filter"/>	Filter start time: <input type="text"/> Filter end time: <input type="text"/> <input type="button" value="Filter"/>
---	---	---

Main objects

The configuration is based on objects. There are several types of objects, each one defining different things in the monitoring process.

Each object consists of a object name and a couple of variables that needs to be configured.

For example on a host object you configure

- host name
- address
- notifications
- active checks
- etc.

In Configure you can

- add new objects
- modify existing objects
- remove existing objects.

A lot of objects can be cross referenced in the configuration and Configure helps you with this to.

In most of the listings you will find a small text field called **Filter by regular expression**. Use this to filter out the content you are interested in when viewing the different lists.

Required directives

All objects have a list of directives that are required when adding a new object. The other directives can be left out. They will then get the op5 Monitor defaults value.

This does not mean you have to set every directive for every object. One solution is called templates. They make it a lot easier to manage a large set of objects. Read more about templates in [Using templates](#) on page 49.

Hosts

Hosts are one of the central objects in the monitoring logic. Important attributes of hosts are as follows:

- Hosts are usually physical or virtual devices on your network (servers, workstations, routers, switches, printers, etc) but it could be practically anything you can reach and monitor from the op5 Monitor server.
- Hosts have an address of some kind, IP address or host name.

- Hosts does not need a service directly associated to them, the services can be inherited from a hostgroup. A host can also exist without services.
- Hosts can have parent/child relationships with other hosts, often representing real-world network connections, which is used in the network reachability logic.

Required directives

The following directives are required for a host object.

- `host_name`
- `address`
- `max_check_attempts`
- `check_period`
- `contacts`
- `contact_groups`
- `notification_interval`
- `notification_period`

The table below describes the required directives for the host object

Directive	Type	Description
<code>host_name</code>	string	This is the id of the object. It may not contain any space in the value.
<code>alias</code>	string	A more describing name for the object.
<code>address</code>	string	The address the host is reached by, preferably an IP address to make sure the host is reachable even if the DNS is down.
<code>max_check_attempts</code>	integer	Is used to define the number of times op5 Monitor will retry checking the host if it returns any kind of problem state. Setting this value to 1 will cause op5 Monitor alert directly without any retry.
<code>check_period</code>	time_period	During this period the host is checked. It can be any time period defined in op5 Monitor.

Directive	Type	Description
contacts	contact	Single contacts used to send notifications to and gives access to this host for users who do not have access to all hosts.
contact_groups	contact_group	Contact groups used to send notifications to and gives access to this host for users who do not have access to all hosts.
notification_interval	integer	Number of minutes between renotifications. Set this to 0 if you only want to have one notification sent out.
notification_period	time_period	During this period the notifications are sent out if any alerts are created. It can be any time period defined in op5 Monitor.

Services

A service can be practically any thing that you can measure and monitor on a host. It is almost only your imagination and programming skills that sets the limit for what you can monitor with a service.

A service

- must be connected to a host
- can check things by tcp, agents, snmp etc.
- use a check command ([Commands](#) on page 43) to communicate with the plugin ([Plugins](#) on page 43) that gets all the data.

Required directives

The following directives are required for a service object.

- host_name
- service_description
- check_command
- max_check_attempts
- check_interval
- retry_interval
- check_period

- notification_interval
- notification_period
- contacts
- contact_groups

The table below describes the required directives for the host object

Directive	Type	Description
host_name	host_name object	The host the service is connected to.
service_description	string	This is the id of the object. It must be unique on a host but may be reused on other hosts.
check_command	command object	This is the short name of the command that is executed during service checks.
max_check_attempts	integer	Is used to define the number of times op5 Monitor will retry checking the host if it returns any kind of problem state. Setting this value to 1 will cause op5 Monitor alert directly without any retry.
check_interval	integer	The number of minutes between normal service checks.
retry_interval	integer	The number of minutes between retry checks when a service has gone into a problem state before the state becomes hard.
check_period	time_period	During this period the service is checked. It can be any time period defined in op5 Monitor.
contacts	contact	Single contacts used to send notifications to and gives access to this host for users who do not have access to all hosts.
contact_groups	contact_group	Contact groups used to send notifications to and gives access to this host for users who do not have access to all hosts.

Directive	Type	Description
notification_interval	integer	Number of minutes between renotifications. Set this to 0 if you only want to have one notification sent out.
notification_period	time_period	During this period the notifications are sent out if any alerts are created. It can be any time period defined in op5 Monitor.

Contacts

A contact is used for two purposes:

- to send notifications to
- permissions to view a objects in the monitoring part of op5 Monitor.

A contact is not the same as the login account given access rights to the system.

Required directives

The following directives are required for a service object.

- contact_name
- host_notifications_enabled
- service_notifications_enabled
- host_notification_period
- service_notification_period
- host_notification_options
- service_notification_options
- host_notification_commands
- service_notification_commands

The table below describes the required directives for the host object

Directive	Type	Description
contact_name	string	The id of the contact object.

Directive	Type	Description
host_notifications_enabled	yes/no	Used to determine whether or not the contact will receive notifications about host problems and recoveries.
service_notifications_enabled	yes/no	Used to determine whether or not the contact will receive notifications about service problems and recoveries.
host_notification_period	time_period object	The time period when the contact will receive any host notifications.
service_notification_period	time_period object	The time period when the contact will receive any service notifications.
host_notification_options	Down, Unreachable, Recovery, Flapping start and stop, Scheduled downtime start and stop	Used to set what type of host notifications the contact shall receive.
service_notification_options	Critical, Warning, Unknown, Recovery, Flapping start and stop, Scheduled downtime start and stop	Used to set what type of service notifications the contact shall receive.
host_notification_commands	command object	The command used to send the host notifications
service_notification_commands	command object	The command used to send the service notifications.

Directive	Type	Description
notification_period	time_period	During this period the notifications are sent out if any alerts are created. It can be any time period defined in op5 Monitor.

Local users

Local users are user accounts that makes it possible to login to the op5 Monitor GUI using the default driver. For more information about drivers see [LDAP Integration](#) chapter

Local users does not have any thing to do with notifications or the permissions of viewing objects in op5 Monitor.

Local users can be connected to a contact by giving the username the same name as the id (contact_name) of a contact.

A local user can also be created by checking the box “Configure access rights for this contact” when creating a contact.

Required directives

The following directives are required for a access rights object.

- username
- password

The table below describes the required directives for the host object.

Directive	Type	Description
username	string	The username is the id of the access rights and also used as login username.
password	string	The password is used for the login.

Access rights in detail

The table below gives you a description of the settings of an access right object.

Directive	Description
<code>authorized_for_system_information</code>	Gives the user access to the system / process information.
<code>authorized_for_configuration_information</code>	Gives the user access to view and change configuration.
<code>authorized_for_system_commands</code>	Gives the user access to issuing commands in the web gui. With commands you can control certain functions in op5 Monitor, for example: enable/disable notifications, scheduled downtime, acknowledge problems etc.
<code>authorized_for_all_services</code>	Gives the user access to view all services, se Customizing views below for more information.
<code>authorized_for_all_hosts</code>	Gives the user access to view all hosts, se Customizing views below for more information.
<code>authorized_for_all_service_commands</code>	Gives the user access to issue commands for all services, se Customizing views below for more information.
<code>authorized_for_all_host_commands</code>	Gives the user access to issue commands for all hosts, se Customizing views below for more information.

Recommended settings

Recommended settings for

- an administrator would be to check all boxes
- help desk staff it could be
 - `authorized_for_system_information`
 - `authorized_for_system_commands` so they can acknowledge problems but not change the configuration.

Time periods

Time periods is time defining objects that span over a week. You can define included time for each day of the week in the time period definition.

You can also:

- use already defined time periods as excludes

- add exceptions based on dates and ranges of days

The time period objects are used at many places in the configuration. Most noticeably are in the contact objects where the time periods defines when notifications should be sent out.

You can also use time periods to define when a service or a host should be monitored or when you are creating availability reports.

A time period in detail

The following tables describes the directives of a time period and how to use them.

The table below describes the first part of directives of a time period.

Directive/option	Description
timeperiod_name	short name of the time period
alias	descriptive name of the time period
Monday to Sunday	which time to include for each day. you can define multiple times by separating them with comma. Example 00:00-01:00,03:00-06:00
Exception type	Specify what type of exception you want to use; Date or Day

Depending on what kind of exception type you have chosen you will get different settings choices. The two lists below describes them all.

The table below describes the exception part of a time period.

Directive/option	Description
exclude	Other predefined time period definitions that should be excluded from this time period.
Exception type	Specify what type of exception you want to use; Date or Day

The table below describes exception by **Date**:

Directive/option	Description
Interval	Choose Single ate or Date range
Date	Choose the date that is supposed to be used in this Exception.
From date	If you chosen date range you will here set the start date To date.

Directive/option	Description
To date	If you chosen date range you will here set the end date.
Frequency	How often the exception is repeated. Valid values are positive integers greater than one. E.g: <ul style="list-style-type: none"> • Date range "2012-01-01 - 2012-12-31 / 5" means every fifth day of 2012. • Day range "1 monday march - 3 sunday may / 3" means every third day between the first monday and the third sunday every month. • Date range "2012-06-01 / 14" means every 14th day from first of june 2012. Note that this exception has no end.
Hours	Which time to include for this exception. You can define multiple times by separating them with comma. Example: 00:00-01:00,03:00-06:00

The table below describes exception by **Day**:

Directive/option	Description
Interval	Choose Single day or a Day range
Weekday	Choose the weekday that is supposed to be used in this Exception.
From weekday	If you chosen Day range you will here set the start day.
To weekday	If you chosen Day range you will here set the end day.
Frequency	How often the exception is repeated. Valid values are positive integers greater than one. E.g: <ul style="list-style-type: none"> • Date range "2012-01-01 - 2012-12-31 / 5" means every fifth day of 2012. • Day range "1 monday march - 3 sunday may / 3" means every third day between the first monday and the third sunday every month. • Date range "2012-06-01 / 14" means every 14th day from first of june 2012. Note that this exception has no end.

Directive/option	Description
Hours	Which time to include for this exception. You can define multiple times by separating them with comma. Example: 00:00-01:00,03:00-06:00

Commands

A command is exactly what it sounds like. It can use macros and arguments. Mostly they are used with services but they can actually be used as

- service or host check command
- notification command
- event handler
- obsession.

Directives

A command has got only two directives

- `command_name`
- `command_line`

Directive	Description
<code>command_name</code>	This is the id of the command and also the name shown in Configure.
<code>command_line</code>	is the actual command line used by the services, notifications, event handlers and obsession.

Plugins

Plugins are compiled executable or scripts that can be run from a command line to check the status of a host or service.

There are many plugins included in the op5 Monitor software. A list of the plugins can be found in the **list-of-plugins** at the support section at www.op5.com.

If you are looking for a plugin not found in op5 Monitor by default there are a bunch of other places to look

- contact op5 for a specific development

- www.op5.org
- exchange.nagios.org

You can use any plugin written for Nagios but you might need to modify them a bit before they work in the op5 Monitor environment.

Groups

The groups in op5 Monitor is used to group objects of the same type. There are three types of groups in op5 Monitor

- host groups
- service groups
- contact groups

They are all good to use to get things a bit more organized and they have also special functions op5 Monitor.

The following subsections will give you a brief description about how they can be used.



You may not have any empty groups in op5 Monitor so when you create a new group, no matter what type it is, you should have at least one object to add to the group.

Host groups

Host groups can be used group hosts together in any way you like.

- A host can be connected to any number of hosts.
- A host group can be connected to an other host group.

There are a few host groups included in the initial setup of op5 Monitor but you can create your own matching your own needs.

There are a infinite ways of using host groups and here are a couple of examples.

Grouping hosts by

- geographic placements
- what company they belongs to
- who owns the hosts
- who should be able to see the hosts in the group
- function or operating system.

The list can be long.

Services on Host groups

A host group can contain service checks. These service checks will be inherited on all hosts connected to the host group.

A service on a host group work in the same way as a service for a host.

To add a service to a host group go to 'Configure' and 'Host Groups'. Choose the host group you want to add services to then select 'Services for hostgroup'

Hostgroup

Filter by regular expression windows_servers

► Need help?

► Services for hostgroup windows_servers

For example a windows servers host group could contain the checks that are common for all windows servers. By doing this you will only need to change command arguments on the service in the host group instead of changing the arguments on all windows host.

If you add new checks to the service group all hosts in the host group will get the new service once you save your configuration.

If a host group service and a host service should get the same name, the host group service will be used, the host service will still be visible in the configuration and if the host is lifted out from the host group the host service will become active.

Nested host groups

Host groups can be connected to each other.

When nesting host groups together the services on host groups also will be inherited to the nested host group. This only works one way.

For example:

Host group A has service X and host 1 is a member of host group A

Host group B has service Y and host 2 is a member of host group B

If host group B is added as a member of host group A then host 1 will get service Y but host 2 will not get service X.

A good way to use this feature is to have i.e. a Windows host group and then a MSSQL host group. When adding the Windows host group as a member to the MSSQL host group the hosts added to MSSQL will get both the service checks that are standard for all Windows host and the default MSSQL service checks.

Service groups

The service groups are used to group services together in the same way as for host groups. On the other hand there is almost no use at all for example group service groups by geographic placements.

One good way to use service groups is to create groups containing services needed for a service you deliver to your customers.

Example 1 *An email service group*

Let us take a simplified email service and show how the service groups can be used.

To be able to deliver an email service to our customers the following services need to be working:

- *DNS*
- *SMTP*
- *IMAP / POP3*
- *WAN Connection*
- *File Storage*

*We take all those services and place them in a service group called **Customer email**.*

*If we get a problem with any of the services in the **Customer email** group we can easily see that the whole email service has got a problem.*

The service group in the example above is perfect to use in Service Level Agreement reports (SLA in the op5 Monitor user manual) to make sure we deliver the service as we promised.

Contact groups

Contact groups are mainly used to setup where to send service and host notifications. It can also be used to setup permissions about who should be able to see what object in the op5 Monitor GUI.

The members of a contact group associated with a certain host and/or service are the one that will get all notifications for that object.

A Contact group can be populated with a contact or another contact group.

Permission to host and services

If a user does not have the access rights to see all hosts that user needs to have a contact connected to the contact group associated with the host or service the user should be able to see.

Show partial hostgroups

If an unprivileged user is not a contact for all hosts in a hostgroup, he will not be able to see the host group in the "Hostgroup summary/overview/grid" views.

To enable viewing of partial host group edit follow these steps logged in as root:

- 1** Create and edit the file `/opt/monitor/op5/ninja/application/config/custom/groups.php` with your favorite editor.
- 2** Put the following into the file:

```
<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```
$config['see_partial_hostgroups'] = true;
```

3 Save the file.

Using templates

Even though Configure makes it easy for you to add and change the configuration of op5 Monitor it is still a lot of things to edit and tweak. To make the software even more easy to use templates have been built in.

There are three types of templates to use:

- host templates
- service templates
- contact templates

op5 Monitor comes with a couple of predefined templates for each object type described above. They are just there to be examples and you should really create your own.

How they work

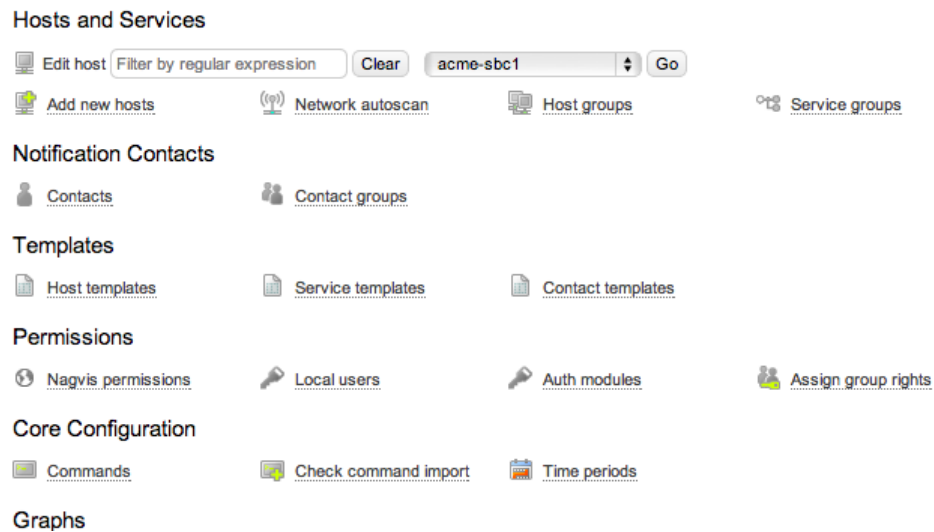
- Any directive set in a template will be used in the objects using the template. But if you set a directive explicit on an object that value will override the templates.
- Any directive not set in neither a template or directly on the object will have the op5 Monitor default value.
- If you change any value on a directive in a template it will only be valid on the objects where the same directive is not set explicit.

Managing objects

Now let us be a bit more hands on. In this section we will take a look at how to add/edit/delete objects using the Configure.

There are sometimes many ways to do things in op5 Monitor but we will only show a few examples.

In the subsections to Managing objects we will assume that you start from the main page of Configure.



Before you start

Add new

Every time you comes to a page where you can handle an object you will have the **Add new...** dialog ready for you to add a new object.

Configuration files

Every object is placed in a configuration file. You may change what file the object is placed in at the bottom of every configuration page. This is normally not necessary and only used in special cases.

Help

In the guides we will only describe the directive that are differ from the default value. Click the **help icon**

Templates

Because handling templates is the same for all kind of templates, only the directives differ, we will only add a template in [Contacts](#) on page 51.

Contacts

Adding a contact template

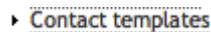
Before we start to add any new contacts we will create a contact template to use with the contact in the next section. In this guide we only describes the directive we will not use the default value in.

To add a contact template

- 1 Click **Templates** on the main page.



- 2 Click **Contact templates**.



- 3 Give the contact template a name



- 4 Change **can_submit_commands** to yes. ¹



- 5 Click **Submit**.

- 6 Click **Save**.

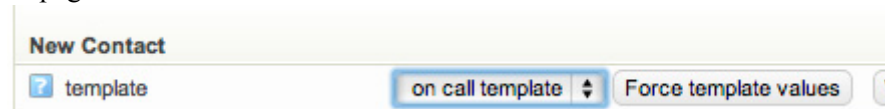
Adding a contact

To add a contact

- 1 Click **Contacts** on the main page.



- 2 Use the template on call template we created in [Adding a contact template](#) on page 51.



- 3 Type in a contact_name



¹.This gives this the user connected to this contact the possibility to execute commands like acknowledge problems etc.

4 Type in an alias

5 Type in the email address

6 Click **Submit**.

7 If you want to create access check the “Configure access rights for this contact” box, otherwise save changes

☐ enable_access ☐ Configure access rights for this contact

8 When Configuring access right for this contact select the access rights the contact should have, after that save the changes.

☐ Access Levels

- ☐ authorized_for_system_information
- ☐ authorized_for_configuration_information
- ☐ authorized_for_system_commands
- ☐ authorized_for_all_services
- ☐ authorized_for_all_hosts
- ☐ authorized_for_all_service_commands
- ☐ authorized_for_all_host_commands
- ☐ Select all

☐ enable_access ☒ Configure access rights for this contact

Modify a contact

To modify a contact

1 Click **Contacts** on the main page.



2 Choose the contact you like to modify in the drop down list.

Filter by regular expression:

3 Click **Go**.

4 In the view you will get only directives differ from the template will be shown. To change the other directives click **Advanced**.

5 Make your modifications and click **Submit**.

6 Click **Save**.

Delete a contact

- 1 Click **Contacts** on the main page.
- 2 Choose the contact you like to modify in the drop down list.

Contact

- 3 Click **Go**.
- 4 Click on **Delete**.

- 5 Click **Save**.

Hosts

There are many ways to add a host. A host can be added by

- the **new host** wizard
- a **network scan**
- cloning of a host
- using a profile

In this guide we only describes the directive we will not use the default value in.

Adding a host with new host wizard

To add a new host using the new host wizard - Part 1

- 1 Click **New host** on the main page.

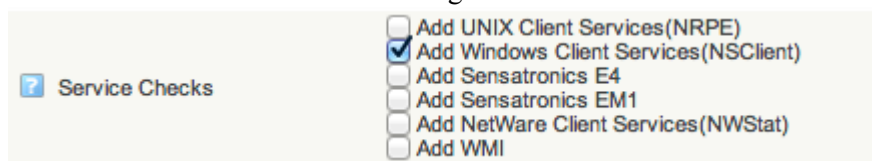


- 2 Type in a host_name.

- 3 Type in an alias.

- 4 Type in the address to the host, IP address is mostly the best choice.

- 5 We assume this is a Microsoft windows server and that NSClient++ has been installed. Check for the following service checks.



When using WMI a administrators account must be selected. It is also possible to create a user with less privileges, see how-to's on www.op5.com

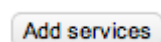
- 6 Click host logo to set the icon that will be displayed for this host in lists and maps.



A list looking like this will be displayed. Click the icon you like to use.



- 7 Click **Add services**.



To add a new host using the new host wizard - Part 2

- 1 Leave the initial settings ¹ as it is and scroll down to the services.

1.All new services will inherit the Initial Service Settings. If you choose not to enter a value for one or more required variable, those variables must be set in the selected template.

- The scan has found out that NSClient++ is installed plus two other services that can be added to this host. Check **Select All** to add all services found or
NSClient++ found in NSClient++ 0.3.7.493 2009-10-12

example_host @ 172.27.86.97 (An example host)

☐ Select All

NET

☒ PING

☐ Nameserver

NSCLIENT

☐ Disk usage C:

☐ CPU Load

☐ Uptime

☐ Mem usage

select the one you like to add for this host.

- Click **Continue to step 3**.
- Now either click the host or service links or click **Save**.

Added 1 host.
Added 6 services.

example_host Services for example_host

Adding hosts with network scan

Network ranges can be specified in a very free form. Each of the four parts of the IP-address may contain any combination of comma-separated numbers, 'from-to' ranges and single numbers, as such: 10.1,2,0,4-10.1-50.

You can specify multiple ranges, separated by spaces, if you like.

To add hosts with network scan

- Click **New host** on the main page.
- Click **Network scan**.
- Fill in the desired network range. We will scan for hosts in the range from 172.27.86.8 - 172.27.86.97.

IP Range: 172.27.86.8-97

Top Domainname:

Poll from: This server

Scan Ranges

- 4 Click **Scan Ranges**.
- 5 In this case we found ¹ three hosts.
Scan completed in 16 seconds.
Found 3 responding hosts.
- 6 Repeat *To add a new host using the new host wizard - Part 1* on page 53 for each host, except for the last step. If here is one or more host you do not like to add choose **No** in **Add this host?**
 When you are finished click **Scan hosts for services**.
- 7 Repeat *To add a new host using the new host wizard - Part 2* on page 54 for each host, except for the last step.
 When you are finished click **Continue to step 3**
- 8 Click **Save**.

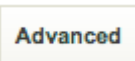
Modifying a host

To modify a host

- 1 On the start page choose the host you like to modify in the drop down list.



- 2 Click **Go**.
- 3 In the view you will get only directives differ from the template will be shown. To change the other directives click **Advanced**.



- 4 Make your modifications and click **Submit**.
- 5 Click **Save**.

Deleting a host

To modify a host

- 1 On the start page choose the host you like to delete in the drop down list.



- 2 Click **Go**.
- 3 Click **Delete**.



- 4 Click **Delete all affected objects**.

1. Only hosts that aren't previously configured will be listed

5 Click **Save**.

Renaming a host

When renaming a host in the web GUI it will only rename the host and will not rename the host name.

To rename the host name in log-files as well a script has to be run manually. The script will rename the host in log-files. If this is not done the host will lose all it's alert history.

To run the script logon to the op5 monitor via SSH as root user and execute the following command:

```
# mon stop; /opt/monitor/op5/merlin/rename --rename-all; mon start
```

If there is a lot of history this script can take a while to execute and during this time the op5 monitor service will not be running.

Note that this does not yet work on schedule downtime objects. If a host is renamed that has a scheduled downtime the scheduled downtime will be lost.

Network autoscan

It might get handy to let op5 Monitor scan and notify you if there are any new hosts on a particular network range.

The network autoscan function will

- scan certain range for new hosts
- notify you when new are found
- be executed every night by cron on the op5 Monitor server.



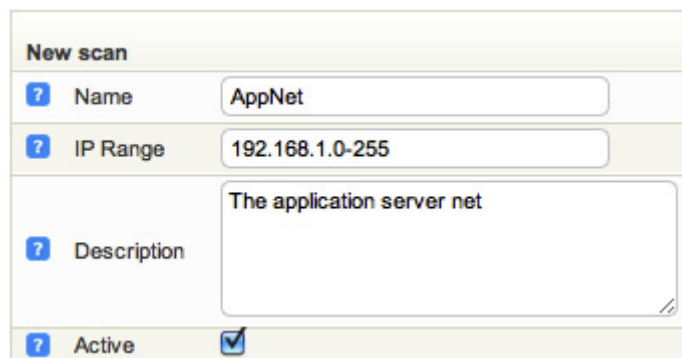
No host will be automatically added. The network autoscan function will only find the hosts for you.

Adding a new autoscan configuration

You may add as many autoscan configuration as you wish. When adding a your network range you may use the same syntax as when you manually scans a network from the Add new host wizard.

To add a new autoscan configuration

- 1 Click **Configure** in the main menu.

2 Click **Network Autoscan**.**3** Fill in the **New scan** form

The 'New scan' form contains the following fields:

- Name:** AppNet
- IP Range:** 192.168.1.0-255
- Description:** The application server net
- Active:** ☒

- **Name:** The identifier of this autoscan configuration
- **IP Range:** In this case a complete C net.
- **Description**
- **Activate:** Make this autoscan configuration active and in use.

4 Click **Save**.

Adding a host to blacklist

In certain ranges you are scanning with the network autoscan there might be hosts you do not want to include in the result. Then you should add that host or hosts to the blacklist.

To add a host to the blacklist**1** Click **Configure** in the main menu.**2** Click **Network Autoscan**.**3** Add a host (IP address) in the **Host** field

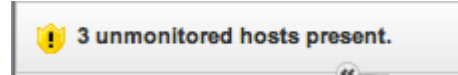
The 'Host blacklist' form contains the following fields:

- List:** A list box showing the hosts added to the blacklist.
- Host:** 192.168.1.2
- Actions:** Add, Remove

4 Click **Add**.

The result

After the networks scan has been executed a small result will be shown in the upper left corner of the op5 Monitor GUI



To add the hosts that has been found you only need to click on the text to the right of the icon. You will then come to the Add new host wizard the same as when you have done a manual network scan.

Services

Services can be added in a few different ways in Configure. You may add a service by using

- **add service for this host**
- **scan host for network services**
- **scan host for snmp interfaces**
- **scan host for windows services with agent**
- **scan host for windows services using WMI**

We will take a look at the **add service for this host**.

In this guide we only describes the directive we will not use the default value in. The default service template will used.

Adding a service

To add a service using add service for this host

- 1 On the start page choose the host you like to add a new service to in the drop down list.



- 2 Click **Go**.
- 3 Click **Services for host...** under related items menu to the right.

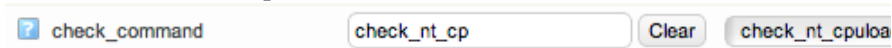


The add new service pages is shown.

- 4 Type in a service_description.



- 5 We will use the `check_nt_cpuload` command for this service.
Type in as many chars you need in the filter by regular expression field until the command shows up.



- 6 Click **Syntax help** to see what arguments are needed for this command.

Command line of selected check command:

```
/opt/plugins/check_nt -H $HOSTADDRESS$ -p 1248 -v CPULOAD -l$ARG1$
```

Plugin syntax:

This plugin collects data from the NSClient service running on a Windows NT/2000/XP/2003 server.

```
Usage:check_nt -H host -v variable [-p port] [-w warning] [-c critical] [-l params] [-d SHOWALL] [-t timeout] [-T timeout_status]
```

Options:

```
-h, --help
    Print detailed help screen
-V, --version
    Print version information
```

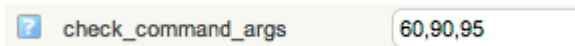
Options:

```
-H, --hostname=HOST
```

You can see that we have a macro called **\$ARG1\$**. This is the first, and in this case the only, argument we need to give to this command.

- 7 Click **Syntax help** again to hide the help text.

- 8 Type in the argument ¹.



- 9 Click **Submit**.

- 10 Click the **Save** icon.



If the arguments include an exclamation mark (!) this has to be escaped with a back slash (\).

Example: `username!crypticpassword\!!warning!critical`

This will output "crypticpassword!"

¹. If more than one the shall be separated by a ! like this: `argone!argtwo`.

Modifying a service

To modify a service

- 1 On the start page choose the host you like to modify a service on in the drop down list.



- 2 Click **Go**.
- 3 Click **Services for host ...**



- 4 Choose the service you like to modify in the drop down list.



- 5 Click **Go**.
- 6 In the view you will get only directives differ from the template will be shown. To change the other directives click **Advanced**.



- 7 Make your modifications and click **Submit**.
- 8 Click **Save**.

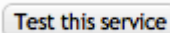
Test this service

Test this service makes it possible for you to test the service you added or modified before you save the new configuration and reload monitor. This is a nice way to make sure the service works as it is supposed to.

In the guide below we will work with the service created in [Adding a service](#) on page 59.

To test a service

- 1 Pick up the service you like to test as it is done in [Modifying a service](#) on page 61.
- 2 Click **Test this service**, at the bottom of the page.



- 3 The output looks like the one below. If you get any errors it will be shown

```
Command: /usr/bin/asmonitor /opt/plugins/check_nt -H 172.27.86.97 -p
1248 -v CPULOAD -160,90,95
```

Seen command (one argument per line):

```
/opt/plugins/check_nt
-H
172.27.86.97
-p
1248
-v
CPULOAD
-160,90,95
```

Plugin output:

```
CPU Load 1% (60 min average) | '60 min avg Load'=1%;90;95;0;100
```

Plugin return code: 0

here in the output

- 4 Click **Test this service** again to hide the output.

Deleting a service


To delete a service

- 1 On the start page choose the host you like to delete a service from in the drop down list.

 Edit Host

- 2 Click **Go**.

- 3 Click **Services for host ...**

 Services for host win-server1

- 4 Choose the service you like to modify in the drop down list.

- 5 Click **Delete**.

- 6 Click **Save**.

Scanning host for network services

When you added your host ([Hosts](#) on page 53) you had the opportunity to add services found during the scan for network services. This scan function can also be reached afterwards.

To scan a host for network services

- 1 Open up the host, in **Configure**, you like to add new services on.
- 2 Click **Scan host for network services**.
- 3 Select the new services found and click **Continue to step 3**.

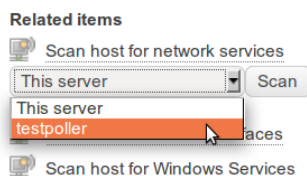
NSClient++ found in NSClient++ 0.3.6.481 2009-03-04

win-gbg-2 @ 192.168.1.8 (Win Gbg 2)	
<input type="checkbox"/> Select all	
NET	
<input type="checkbox"/> PING	
NSCLIENT	
<input checked="" type="checkbox"/>	CPU Load
<input checked="" type="checkbox"/>	Disk usage D:
<input checked="" type="checkbox"/>	Disk usage E:

[Continue to step 3](#)

- 4 Click either the host or service link to go back to the place where you started.
- 5 Click **Save**.

Additional information: In a distributed environment a selectbox will appear when hovering over the menu item “Scan host for network services” where you can select from which op5 Monitor system that should preform the scan.



Scanning a host for snmp interfaces

In many times when you are about to monitor a switch or a router you need to setup a lot of services. It is hard work and takes a lot of time to add them one by one.

Instead of adding all interface services one by one you should use the scan for snmp interfaces function.

To add snmp interfaces

- 1 Open up the host, in **Configure**, you like to add new services on.
- 2 Click **Scan host for SNMP interfaces**.
- 3 Set the SNMP community.
- 4 Chose SNMP version.

5 Click **Scan host**.

sth-sw01@ 172.27.76.202

Interface	Status	Traffic	Errors
Select all: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
Interface 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface 5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface 6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface 7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface 8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface 25	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interface 26	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Add selected services

- 6 Select the services you like to add.
- 7 Click either the host or the service link to get back.
- 8 Click **Add selected services**.
- 9 Click **Save**.

Scanning host for windows services

There are two ways to scan a windows host for services:

- Using the windows agent NSClient++
- Using WMI, Windows Management Instrumentation

The following sections will describe how to accomplish this using the different techniques.

Scan for services using agent

Adding a service that checks a windows services is many times harder than you think. You need to

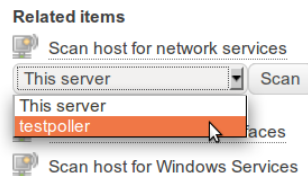
- have access to the windows server
- know the exact name of the windows service

With op5 Monitor you do not need to do anything more than make sure the latest agent (NSClient++) is installed and follow the next few steps.

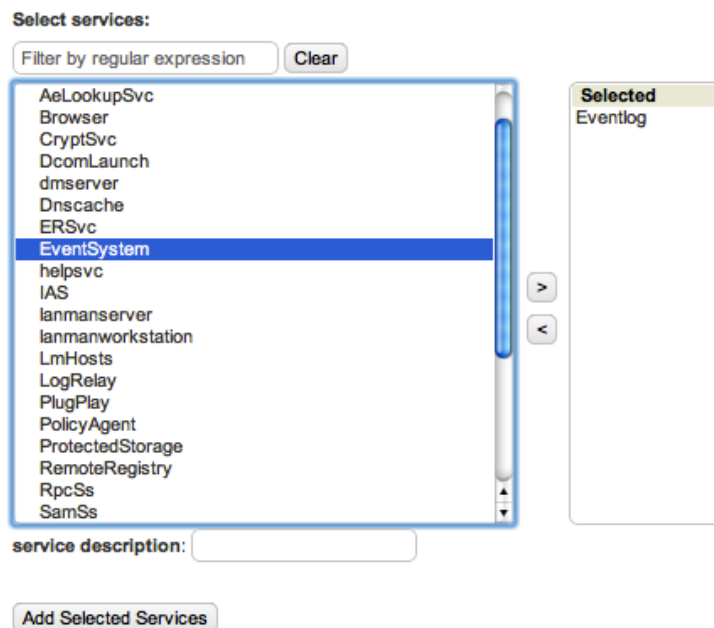
To add windows services

- 1 Open up the host, in **Configure**, you like to add new services on.
- 2 Click **Scan host for Windows Services**.

- 3 Choose which server to preform the scan:



- 4 Select the Windows Services you like to add as a new service in op5 Monitor.



- 5 Give the new service a **Service description**.
- 6 Click **Add Selected Services**.
- 7 Click either the service link or the **Scan for more service** button.
- 8 Click **Save**.

Scan for service using WMI

Scan for services using Windows Management Instrumentation has a number of dependencies to be able to work:

- WMI enabled on the windows server
- User account on the windows server with sufficient privileges

There are two ways to scan for WMI on a windows host:

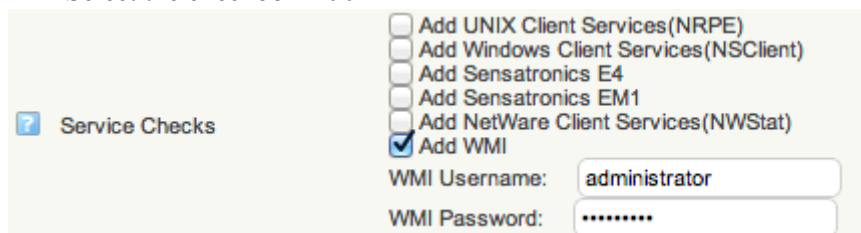
- When adding a new host
- Scanning a existing host

Scanning for WMI when adding a new host

To scan a host for WMI-counters and services upon adding the host to your op5 Monitor configuration as partly described in: [Adding a host with new host wizard](#) on page 53.

To scan for WMI counters when adding a new host:

- 1 Select **Configure** in the main menu
- 2 Click on **New Hosts**
- 3 Enter the information about the host
- 4 Select the checkbox **Add WMI**

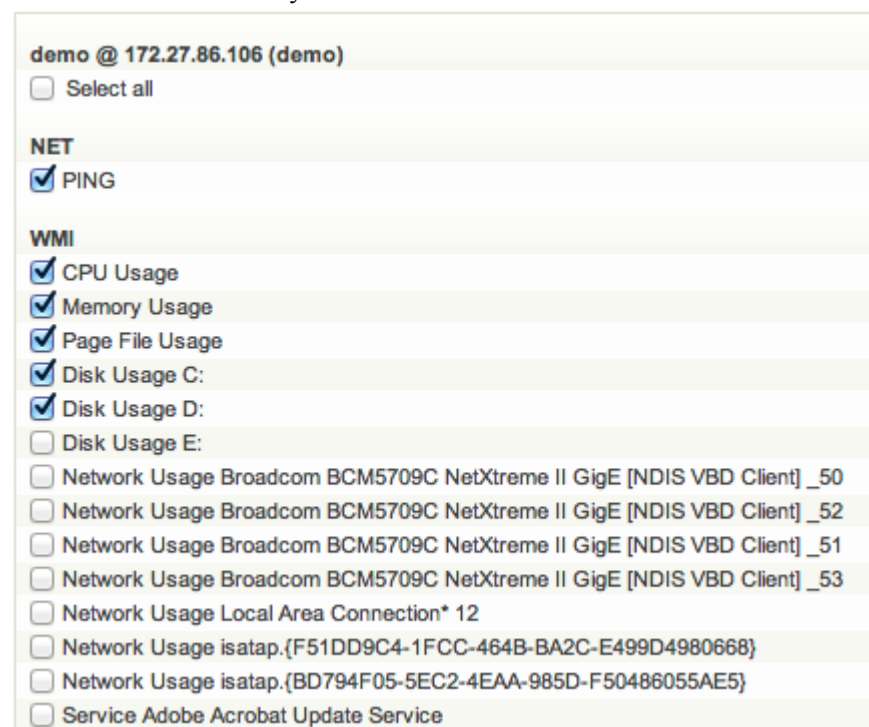


☒ Add WMI

WMI Username: administrator

WMI Password:

- 5 Enter username and password
- 6 Press **Add Services**
- 7 Select the services you wish to add from the list:



demo @ 172.27.86.106 (demo)

☐ Select all

NET

☒ PING

WMI

☒ CPU Usage

☒ Memory Usage

☒ Page File Usage

☒ Disk Usage C:

☒ Disk Usage D:

☐ Disk Usage E:

☐ Network Usage Broadcom BCM5709C NetXtreme II GigE [NDIS VBD Client] _50

☐ Network Usage Broadcom BCM5709C NetXtreme II GigE [NDIS VBD Client] _52

☐ Network Usage Broadcom BCM5709C NetXtreme II GigE [NDIS VBD Client] _51

☐ Network Usage Broadcom BCM5709C NetXtreme II GigE [NDIS VBD Client] _53

☐ Network Usage Local Area Connection* 12

☐ Network Usage isatap.{F51DD9C4-1FCC-464B-BA2C-E499D4980668}

☐ Network Usage isatap.{BD794F05-5EC2-4EAA-985D-F50486055AE5}

☐ Service Adobe Acrobat Update Service


- 8 Press **Finish** at the end of the page.


The host is added and you can save your configuration.

Done adding new host

Added 1 host.
Added 8 services.

demo Services for demo

- 9 Press **Save** in the top right corner 
- 10 Review your changes then by clicking on **More info** press **Save objects I have changed**

 You are about to save your configuration.

This will overwrite your current configuration, so make sure you've done everything right.


Changes

There are 1 changes to 1 host objects. [More info.](#)

Do you want to save your new configuration?

















[Save objects I have changed](#)

After this the configuration will be saved and a final preflight configuration check has been performed.

 Preflight configuration check turned out ok.

Monitor has successfully loaded the new configuration.

Your configuration is saved and the host and its services are ready to be monitored.

Status	Host	Status	Service	Actions	Last Check	Duration	Attempt	Status Information
	winserver_hyperv		CPU Usage	 	2012-11-15 22:41:00	4m 47s	1/3	OK (Sample Period
			Disk Usage C:	 	2012-11-15 22:41:45	4m 2s	1/3	OK - C: Total=121.
			Memory Usage	 	2012-11-15 22:42:29	3m 18s	1/3	OK - Physical Mem
			PING	 	2012-11-15 22:43:13	2m 34s	1/3	OK - 172.27.86.104
			Page File Usage	 	2012-11-15 22:43:57	1m 50s	1/3	OK - Total: 3.250G Free: 3.250GB (10

Custom Variables

Custom variables can be used to store custom information for hosts, services and contacts in the configuration. These variables can be used as a macro in command arguments and notifications for example.

All custom variables will automatically get an underscore “_” as a prefix to prevent name collisions with the standard variables.

The custom variable will also automatically be converted to upper case.

In order to prevent name collision among custom variables from different object types, Nagios prepends “_HOST”, “_SERVICE”, or “_CONTACT” to the beginning of custom host, service, or contact variables, respectively, in macro and environment variable names.

These variables can be used as macros in same way as the standards macros in op5 Monitor.

When using a custom variable as a macro a “\$”-sign is always used before and after the variable name.

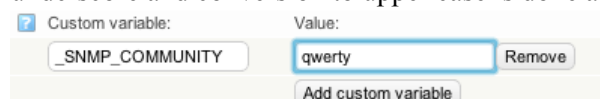
Entered Name	Variable name	Macro name
snmp_community	_SNMP_COMMUNITY	\$_SNMP_COMMUNITY\$
location	_LOCATION	\$_LOCATION\$

Creating a new custom variable

Go to the configuration for a host, service or contact and click on **add custom variable**.



Enter a variable name and the value of the variable. Note that the prefix underscore and conversion to upper case is done automatically.



Click on **submit** and save the configuration.

Example

Instead of using the SNMP community name hardcoded in the check command or in the command arguments in the service check we will create a custom variable that we will use as a macro in the command arguments.

In this example we will move the SNMP community name on a traffic check on a switch port from being in the command arguments to a custom variable.

First we create a custom variable on a switch traffic check, see [Creating a new custom variable](#) on page 68.

Name the variable: snmp_community (the prefix and upper case conversion will be done automatically).

Enter the name of your SNMP community as a value. Let's say for this example that the community name is "qwerty"

Change the command argument of the command argument from

"qwerty!2!100mbit!70!90" to

"\$_SERVICESNMP_COMMUNITY\$!2!100mbit!70!90"

The screenshot shows a configuration form for a service named "firewall1-sth;IF 2: eth0 Traffic". The form has several fields and buttons:

- template:** A dropdown menu set to "default-service" with a "Force template values" button and a "View ten" button.
- service_description:** A text field containing "IF 2: eth0 Traffic".
- check_command:** A text field containing "Filter by regular expression" with a "Clear" button and a "check_traffic" button.
- check_command_args:** A text field containing "\$_SERVICESNMP_COMMINTY\$!2!100mbit!70!90".
- file_id:** A dropdown menu set to "etc/services.cfg".
- Custom variable:** A section with a "Value:" label and an "Add custom variable" button.

Click on **submit** and save the configuration.

Dynamic Button

The dynamic button is a customizable button which any script can be added to.

If defined, a link in the service information page will appear under "Service Commands" on the service ext info page.

Configuration

To configure the dynamic button two custom variables has to be created on the service which the button should be added to.

The first one is the command line and the second one is the permissions.

The prefix `_OP5` symbolizes that this is a dynamic button variable. If an H is added to the prefix (`_OP5H`) the custom variables will not be visible in the Service State Information table.

`_OP5H__ACTION__NAME`

`_OP5H__ACCESS__NAME`



Note that there are two underscores!

Action

The action has the variable name `_OP5H__ACTION__NAME`

The value of the action is the path to the script that should be executed.

The name of the button is set by replacing “NAME” in the variable name. When using spaces in the name, this should be replaced by one underscore.

Example:

To name the dynamic button “Restart Service” and it will execute the script /opt/plugins/custom/restart_service.sh. The variable name should be:

Variable name: _OP5H__ACTION__RESTART_SERVICE

Value: /opt/plugins/custom/restart_service.sh

Access

The _OP5H__ACCESS__NAME sets who will be able to use the dynamic button. This is set on contact-groups only.

If a user is not in a group that is specified in the access variable the button will not be visible for the user.

The access variable name must have the same name as the action name.

Example

If you want to give access to the “Restart Service” action to the support-group and windowsadmins groups the setup should look like this:

Variable name: _OP5H__ACCESS__RESTART_SERVICE

Value: support-group,windows-admins

Escalations

Escalations let you configure escalation of notifications for this host. The idea is that if you have a really important host you can send the first notification to the default contact group in order for them to solve the problem. If the problem is not solved in lets say 30 minutes you can send the notification to a broader range of contacts.

Host and service escalations works exactly in the same way so we will only take a look at host escalations from now on.

Adding a host escalation

In this guide we will add a small escalation chain that does the following

- First notification is sent to the support-group
- After 10 minutes the second (the last one) is sent to the sysadmins group.

To add a host escalation

- 1 On the start page choose the host you like to add an escalation to in the drop down list.



- 2 Click **Go**.

3 Click **Escalations**.

Escalations

4 Add the escalation number one.

a Choose the contact group that shall have the notification.



b Set the start number in the escalation chain.



c Set the end number in the escalation chain ¹.




d Set the notification interval which is the number of minutes to wait to the next notification.



e Choose the time period when this escalation will be in use.



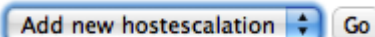
f Choose what states this escalation will be valid for.



In this case we do not use the escalation for unreachable or recovery which means that unreachable and recovery notifications will be sent to the contact group set on the host.

5 Click **Submit**.

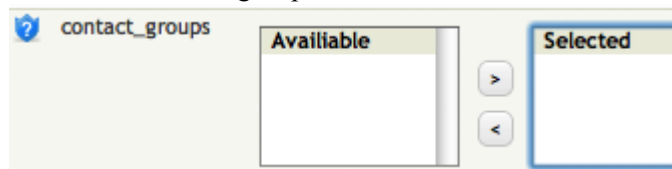
6 Choose Add new host escalation



7 Click **Go**.

8 Add the escalation number two.

a Choose the contact group that shall have the notification.



¹.If the start number is 1 and the end number is two it means that the first and the second notification will be handled by this escalation.

- b** Set the start number in the escalation chain.

 first_notification


- c** Set the end number in the escalation chain ¹.

 last_notification


- d** Set the notification interval which is the number of minutes to wait to the next notification. ²

 notification_interval

- e** Choose the time period when this escalation will be in use.

 escalation_period

- f** Choose what states this escalation will be valid for.

 escalation_options ☒ Down
☐ Unreachable
☐ Recovery

In this case we do not use the escalation for unreachable or recovery which means that unreachable and recovery notifications will be sent to the contact group set on the host.

- 9** Click **Submit**.

- 10** Click **Save**.

Modifying a host escalation

To modify a host escalation

- 1** On the start page choose the host you like to modify an escalation on in the drop down list.

 Edit Host

- 2** Click **Go**.

- 3** Click **Escalations**.

Escalations

- 4** Choose the escalation you like to modify.

- 5** Click **Go**.

- 6** Make the modifications you like to do and click **Submit**.

- 7** Click **Save**.

1. We have set the first notification and the last notification to 2 because this escalation will only be used once.

2. The escalation interval is set to 0 because there will be no more escalations when this one is done.

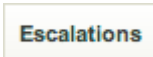
Deleting a host escalation

To delete a host escalation

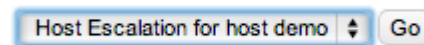
- 1 On the start page choose the host you like to delete an escalation from in the drop down list.



- 2 Click **Go**.
- 3 Click **Escalations**.



- 4 Choose the escalation you like to modify.



- 5 Click **Go**.
- 6 Click **Delete**.



- 7 Click **Save**.

Access rights and contacts

To be able to login to op5 Monitor you need to have a user, described in [Local users](#) on page 39. But you need to have a contact, described in [Contacts](#) on page 37, to be able to receive notifications and in some cases even be able to see any hosts or services.

By connecting access rights to a contact you will be able to login and get notifications with the user created in access rights.

So basically what you need to do is to configure a new contact. Add the contact to an existing contact group or create a new contact group specific for the new contact. If you created a new contact group make sure to add the contact group for the hosts and services that you want to make available in the customized view.

Add new access rights and connect it to the contact you created earlier.

Connecting access rights to contacts

To connect access rights to a contact

- 1 Configure a new contact.
- 2 Add the contact to an existing contactgroup or create a new contactgroup specific for the new contact.
If you created a new contactgroup make sure to add the contact group for the hosts and services that you want to make available in the customized view.

- 3** Configure a user in access rights with the exact same name as the contact you created.
- 4** Set the options for the new access right.
When selecting options do not use the last four options, authorized for all.
By doing this the new user will only see the hosts and services that uses the contactgroup that he is a member of.

Make things easy

Profiles

Creating a Profile

To create a profile

- 1 On the start page choose the host you like to create a profile of in the drop down list.



- 2 Click **Go**.
- 3 Click the **Clone** button
- 4 Select the services you wish to include
- 5 Select **Save as Profile**
- 6 Enter name and description for the profile you are creating
- 7 Click **Clone**

You are then presented with the option of creating clones based on this new profile. If you do not wish to do this now, you can simply use the left hand web menu to return to Configure or another part of op5 Monitor.

Using a Profile

To use a profile

- 1 From the start page click **Profiles**
- 2 Click **use** next to the profile
- 3 Select what parts of the profile you want to include
- 4 Fill in the number of copies and click **Continue...**
- 5 Fill out host details for the clones and click **Create**

Cloning objects

Cloning from an existing Host

To clone a host

Follow the instructions from in [Creating a Profile](#) on page 75, except do not click **Save as Profile**.

Cloning services

If you want to create the same service check on multiple host first create the service check on the host, then clone the service check to one or more hosts.

It is also possible to clone multiple services to one or more hosts or hostgroups.

To clone a service to an other host

- 1 Choose the Configure web menu.
- 2 Choose your host you want to copy from, then click **Go**
- 3 Click **Services for host...** in the 'RELATED ITEMS' menu.
- 4 Select the service (or one of the services) you want to clone then click on **Go** and then on **Clone**.
- 5 Select the service(s) you want to clone.
- 6 You can chose to clone the service(s) to a list of hosts, a hostgroup or all hosts in a hostgroup.
- 7 Click **Clone**.

Copy objects

There are a number of objects that can be copied in the configuration tool and make a exact copy of the object, besides the name that must be unique.

These are the objects that is possible to make a copy of:

- Hosts
- Services
- Hostgroups
- Servicegroups
- Check commands
- Contacts
- Contactgroups
- Templates
- Timeperiods
- Host Dependencies
- Service Dependencies
- Host Escalations
- Service Escalations

The copy will inherit all the values set on a object except the name.

To illustrate this let us make a copy of a check command and modify it slightly:

- 1 Click on **Configure** in the main menu:



- 2 Select **Commands** in the configuration menu

- 3 Search for a command to copy:

Search results for 'check_esx3_host_vmfs':

- Related items:
 - Check Command Import
 - Bulk delete objects
- host_vmfs:
 - check_esx3_host_vmfs
 - Syntax help
 - line: \$USER1\$/check_esx3 -H \$HOSTADDRESS\$ -u \$ARG1\$ -p \$ARG2\$ -l vmfs -s \$ARG3\$ -w \$ARG4\$ -c \$ARG5\$
 - etc/checkcommands.cfg
 - mand

- 4 Click **Copy**

- 5 Make the changes you want. A new name is required. and i.e create a listing of the attached VMFS-storages:

New command form:

- command_name: check_esx3_host_vmfs_list
- Syntax help
- command_line: \$USER1\$/check_esx3 -H \$HOSTADDRESS\$ -u \$ARG1\$ -p \$ARG2\$ -l vmfs
- file_id: etc/checkcommands.cfg
- Test this command
- Submit

- 6 Click **Submit**

This approach should apply to the most objects that are possible to copy.

Propagate settings

To change the same directive on many objects of the same type can be a really time consuming work. This is where the propagate function in op5 Monitor is very handy.

With the propagate function you can copy the value of a directive from one object to one or many other objects of the same type.

In the guide below we will use the propagate function to copy the parents from one host to a couple of other hosts.

To propagate a value of a directive

- 1 On the start page choose the host you like to propagate a directive value from in the drop down list.

Edit Host form:

- Filter by regular expression
- Clear
- win-server1
- Go

- 2 Click **Go**.

3 Click **Propagate**.

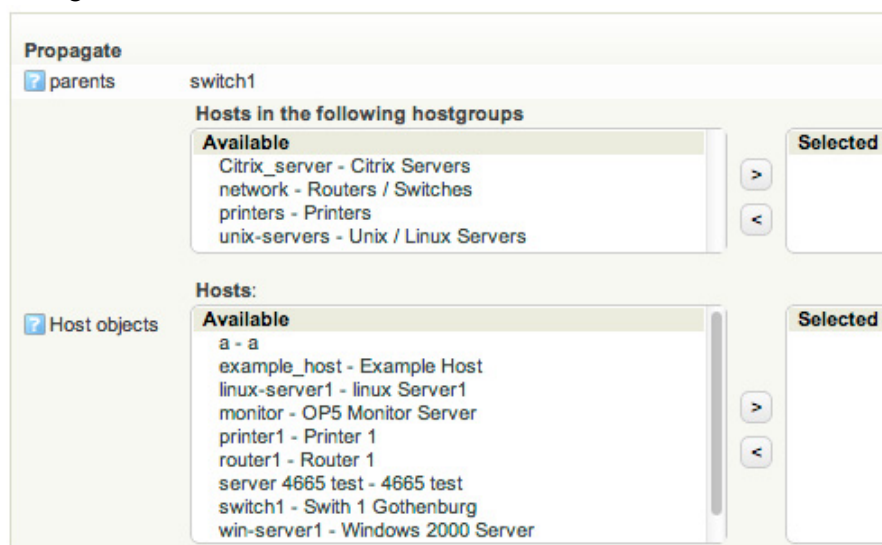
Propagate

Now you will see a check box in front of every directive like this.

☒ parents

4 Check the check box for parents and click **Propagate selected settings**.

5 Select the host objects (host or whole host groups) you like to propagate the settings to.



6 Select how you want to propagate your settings.

☒ Propagation format for multi-value attributes **Replace**

You can choose the following options

Option	Description
Replace	Replace the destination values.
Append	Append the source values to the destination values.
Subtract	Subtract the source values from the destination values.

7 Click **Go**.

8 Click **Save**.

Bulk delete

Bulk delete is powerful tool to remove several host or services at once.

Bulk delete support the following objects:

- Hosts
- Services

- Hostgroups
- Servicegroups
- Contacts
- Contactgroups
- Commands
- Time Periods

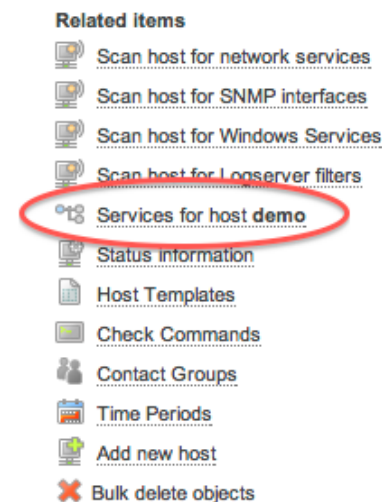
As an example, we will delete two services "Ping" on two different hosts, but the process is similar on all objects listed above.

To delete multiple services this is preformed trough Configure

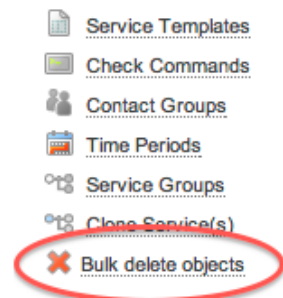
- 1 Select a host which services you want to delete and click **GO**



- 2 Click **Services for host** in the right menu.



- 3 Click on **Bulk delete objects**



- 4 Select the services that you want to delete and click **Delete**

Delete the following services

Services in the following servicegroups

Available		Selected
monitor_sg - monitor_sg	>	
Web_services - Web Services	<	

Services:

ping

Available		Selected
a;PING		demo;PING
example_host;PING		win-server1;PING
linux-server1;PING		
printer1;PING		
router1;PING		
server 4665 test;PING		
switch1;PING		
winserver_hyperv;PING		

- 5 Click **Save submitted changes**

Time periods

Add a time period

Time periods is time defining objects that span over a week. You can define included time for each day of the week in the time period definition.

You can also:

- use already defined time periods as excludes
- add exceptions based on dates and ranges of days

The time period objects are used at many places in the configuration. Most noticeably are in the contact objects where the time periods defines when notifications should be sent out.

You can also use time periods to define when a service or a host should be monitored or when you are creating availability reports.

Macros

Macros can be used to a lot of things. It can for example be used for paths, passwords and retrieving information from op5 monitor.

You can read more about notification commands in *Notification macros* on page 154 in the *Notifications* chapter.

Pre-defined macros

By default op5 monitor has a number pre-defined macros. All from path to plugin folder to retrieving information about the last state of service check.

Below is a list of some macros a complete list of macros can be found at nagios home page: http://nagios.sourceforge.net/docs/3_0/macrolist.html

MACRO	DESCRIPTION
\$USER1\$	Path to /opt/plugins.
\$ARGn\$	The nth argument passed to the command
\$HOSTNAME\$	Short name for the host.
\$HOSTADDRESS\$	Address of the host.
\$HOSTSTATES\$	A string indicating the current state of the host ("UP", "DOWN", or "UNREACHABLE").
\$SERVICEDISPLAYNAME\$	An alternate display name for the service.
\$SERVICESTATES\$	A string indicating the current state of the service ("OK", "WARNING", "UNKNOWN", or "CRITICAL").

Custom macros

It is possible to create your own macros. This can be used to store passwords or user names for example.

All custom macros should be put in the file /opt/monitor/etc/resource.cfg

A custom macro should use the \$USERn\$ macro.

To define a password for a check, first add the macro in resource.cfg

```
# Password for vmware user
$USER10$=secretpassw0rd
```

After that add the macro to check command, in this example we use the check_esx3_host_cpu_usage check command.

check_esx3_host_cpu_usage	
command_name	check_esx3_host_cpu_usage
	Syntax help
command_line	\$USER1\$/check_esx3 -H \$HOSTADDRESS\$ -u \$ARG1\$ -p \$USER10\$ -l cpu -s usage -v
file_id	etc/checkcommands.cfg
Test this command	

```
command_name=check_esx3_host_cpu_usage
```

```
command_line=$USER1$/check_esx3 -H $HOSTADDRESS$ -u $ARG1$ -p  
$USER10$ -l cpu -s usage -w $ARG2$ -c $ARG3$
```

This check will use the following macros:

\$HOSTADDRESS\$ - Will get the address of the host from the configuration

\$ARG1\$ - Use the first argument from the check command.

\$USER10\$ - Use the argument specified in resources.cfg with the same name.

\$ARG2\$ - Use the second argument from the check command.

\$ARG3\$ - Use the third argument from the check command.

Writeprotected configuration files

By making a file write-protected it cannot be changed by op5 Configuration.

This is equivalent to the nagios function called **notouch**.

Create the file `/opt/monitor/op5/nacoma/custom_config.php`

Add the following content:

```
<?php
$notouch_file_prefix = 'static_';
?>
```

If a configuration file is renamed with the `static_` prefix op5 configuration will not be able to change this file.

Features not supported by Configure

Even though some features are not supported by the op5 Monitor configuration tool you can still use them.

The `hostgroup_name` is one of them.

What you have to do is to add a separate configuration file not read by the `import` function in `Configure`. Then you add your other configuration tricks into that file.

To add a configuration file not read by Configure

- 1 Open up a ssh connection to the op5 Monitor server and login as root.
- 2 Create the following file with an editor of your choice:
`/opt/monitor/op5/nacoma/custom_config.php`
- 3 Add the following code to the file you just created:

```
<?php
$notouch_file_prefix = "_";
?>
```
- 4 Create a configuration file with “_” as a prefix to the file name like this:
`touch /opt/monitor/etc/_custom_objects.cfg`
- 5 Add the file to the `/opt/monitor/etc/nagios.cfg` with by adding the following line below the other `cfg_file` variables in `nagios.cfg`:
`cfg_file=/opt/monitor/etc/_custom_objects.cfg`
- 6 Restart op5 Monitor.
`service monitor restart`

Now you may add your objects to the new configuration file and they will not be loaded into `Configure`. But you can still see the objects using `View config` as it is described in the op5 Monitor user manual.

Plugins

Introduction

op5 Monitor is shipped with many plugins that cover most monitoring needs. But what to do if one of your corporate applications can not be monitored straight out of the box?

Often you can find a plugin at www.nagiosexchange.org, and since op5 Monitor and Nagios uses the same plugin format you can often simply download a plugin, put it in `/opt/plugins/custom/` and start using it.

However, if you can not find a suitable plugin anywhere you might have to write your own plugin. Since the plugin interface is very straight-forward, anyone with a fair amount of UNIX scripting experience can do this.

Paths and macros

All standard plugins shipped with op5 Monitor are installed in:

`/opt/plugins`

The macro you use to reach the plugins folder is:

`$USER1$`

The plugins you add to the system by your own must be placed in:

`/opt/plugins/custom`

And they will then be reached with the following macro/path:

`$USER1$/custom`

The reason for placing your own plugins in `/opt/plugins/custom` is because then they will not be touched by any upgrade from op5.

Before you start

Before you can start developing your own plugins you need to make sure you have SSH access or terminal access to your op5 server. The possibility to transfer files to your op5 Monitor server. Any kind of editor, vim and jed are installed by default on your op5 Monitor server.



Microsoft Windows users may use PuTTY for terminal access via SSH and WinSCP for file transfers via SFTP (SSH).

Macintosh or UNIX/Linux users may use the commands `ssh` or `scp` from a local terminal window.

The plugin interface

A plugin is a small executable that takes optional command line parameters as input and

- 1 Performs a test
- 2 Reports a diagnostic message on stdout (will be shown in the web GUI)
- 3 Returns an exit code.

Example 1 Execute `check_tcp` to test the port 80/tcp on 193.201.96.136

```
monitor!root:~# /opt/plugins/check_tcp -H 193.201.96.136 -p 80
TCP OK - 0.043 second response time on port
80|time=0.042824s;0.000000;0.000000;0.000000;10.000000
monitor!root:~# echo $?
0
monitor!root:~# /opt/plugins/check_tcp -H 193.201.96.136 -p 143
Connection refused
monitor!root:~# echo $?
2
monitor!root:~#
```

In the [Example 1](#) on page 91 we first execute `check_tcp` to test that port 80/tcp on 193.201.96.136 responds, which it does, hence the exit code of 0.

Then we check port 143/tcp on the same host and that port is not open, hence the result is Critical - exit code 2.

The result output is actually built upon two parts divided by a `|` sign (pipe). The text on the

- left hand side of the `|` is the status information
- right hand side of the `|` is the performance data.



The performance data is not mandatory but you need it if you want your plugin to be able to produce graphs for you in op5 Monitor.

Status output

The Status output is the text describing the result in human readable form. The plugin must print the status output to stdout when your plugin is executed.

You will see it in the Status state information on the Service or Host information page.

Status Information	HTTP OK: HTTP/1.1 302 Found - 502 bytes in 0.007 seconds response time
--------------------	---

This text can be anything, you like to use to describe the status situation for your plugin, including HTML.

Performance data

The performance data is data displaying the result in numbers. The plugin must print the status output to stdout when your plugin is executed. It is also to produce performance graphs in op5 Monitor.

So if you want graphs from your plugin you need to have performance data in your output.

The performance data is setup like this:

```
'label'=value[UOM];[warn];[crit];[min];[max]
```

Table 1 Performance parts with descriptions.

Part	Description
label	The label can contain any characters. If space is included quotes are needed.
value	The plugin was able to check the service, but it appeared to be above some "warning" threshold or did not appear to be working properly
UOM	Can be any of: <ul style="list-style-type: none"> no unit assuming an integer as a value s - seconds (also us, ms) % - percentage. B- Bytes (also KB, MB, GB and TB) c - A continuous counter like bytes transmitted on an interface.
warn, crit, min, max	<ul style="list-style-type: none"> Can all be null and trailing unfilled semicolons can be dropped. min and max is not needed if UOM is %. value, warn, crit, min and max must be of the same UOM.

Example 2 Performance data output

```
time=0.218901s;;;0.000000 size=42236B;;;0
```

The [Example 2](#) on page 93 shows a performance data output from a plugin with two values separated with one space in the output.

Return code

The return code is the one that op5 Monitor uses to determine what state the services is in. It may be one of the following:

0, 1, 2, 3

Any return code above 0 is to be known as **problem states**.

Table 2 The return codes in detail.

Nr	Name	Description
0	Ok	The check did ok and everything seems to be working fine.
1	Warning	The plugin was able to check the service, but it appeared to be above some "warning" threshold or did not appear to be working properly
2	Critical	The plugin detected that either the service was not running or it was above some "critical" threshold
3	Unknown	Something unknown happened during the check. Things like invalid command line arguments or low-level failures internal to the plugin shall not be reported as Unknown state.

Adding your first plugin to op5 Monitor

In this section we will create a very simple plugin. We will write it as a bash script in a ssh connection to the op5 Monitor server.

This plugin will not actually be very useful but we will use it to describe the steps needed when you start to add other, more useful, plugins.

Creating the plugin

To create a simple example plugin as a bash script

- 1 `cd /opt/plugins/custom`
`touch helloworld`
`chmod 755 helloworld`
- 2 Open up the script with your favorite text editor and type in the following example plugin:

```
#!/bin/sh
echo 'WARNING: Hello world!'
exit 1
```

- 3 Save and exit your editor
- 4 Execute it from the terminal:
- 5 `./helloworld`
`WARNING: Hello world!`
`echo $?`
`1`

The script prints the status output (WARNING: Hello world!).

`echo $?` prints the return code of the last executed command.

Configuring op5 Monitor to use the plugin

To configure op5 Monitor to use the plugin

- 1 Go to **Configure** and choose **Commands**.
- 2 Add a new command with:
`command_name: check_local_helloworld`
`command_line: $USER1$/custom/helloworld`
- 3 Click **Apply** and then **Save**.

Now you may use your check command with a service as described in [Services](#) on page 59

Creating a more complex plugin

In this section we will create a more complex and useful plugin compared to the one we created in [Adding your first plugin to op5 Monitor](#) on page 95. We will stick to bash, because of the simplicity.

We will create a plugin that checks that the storage path specified in `/etc/op5backup.conf` exists, to make sure that `op5backup.sh` is configured properly for local operation.

To create a more complex plugin

- 1 Create the script and editing it:

```
cd /opt/plugins/custom
touch check_op5backup
chmod 755 check_op5backup
```
- 2 Open up the script with your favorite text editor and type in the following code:

```
#!/bin/bash
# Create a function to print the storage path
storagepath() {
  grep ^storagepath /etc/op5backup.conf |
  tail -1 |
  sed 's/^[^"]*"//g' | sed 's/"$//g'
}

# Put the storage path in an environmental variable
STORAGEPATH=`storagepath`

# Test if the storagepath exists and is a directory
if [[ ! -d "$STORAGEPATH" ]]; then
  # Print a warning message for the web gui
  echo op5backup.sh is not properly configured for local
  operation
  # Exit with status Warning (exit code 1)
  exit 1
fi

# If the script reaches this point then the test passed
# Print an OK message
echo $STORAGEPATH exists
# Exit with status OK
exit 0
```

- 3 Add a `check_command` like this using the op5 Monitor web gui:

```
command_name: check_op5backup
command_line: $USER1/custom/check_op5backup
```
- 4 Enter the service configuration for your monitor server, and add a service with `check_op5backup` as the `check_command`.
- 5 Save configuration.

More information

This chapter has only scratched on the surface of how to write your own plugins.

To read more about plugin development take a look at the **Nagios plugin development guidelines**:

<http://nagiosplug.sourceforge.net/developer-guidelines.html>

Widgets

Introduction

In the op5 Monitor user manual we describes how the widgets works in the user interface. There you can read about how to

- hide and show them
- move around the widgets
- change the widgets refresh rate
- restore to default settings.

In this chapter we will take a look at how you could create your own widgets.

The op5 Monitor user interface is using Kohana as framework some of the backend parts for the widgets are handled by Kohana. But this chapter will only describe the widget development in it self.

In this chapter we will focus on creating a small hello world widget.

The widget basics

The widget rules

All widgets need to follow a few rules. They need to have a correct

- File structure
- Widget class
- View file

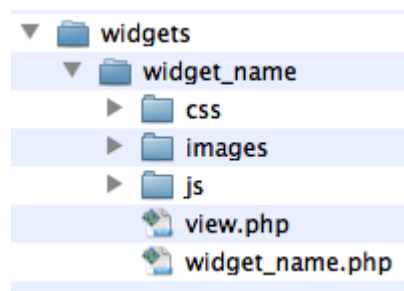
File structure

The widget has to be placed in a folder with the exact name as the widget.

All default widgets shipped with op5 monitor is placed in:

```
/opt/monitor/op5/ninja/application/widgets
```

And the file structure below the widgets folder looks like this



But as we are developing our own widget we need to place it in:

```
/opt/monitor/op5/ninja/application/custom_widgets
```



Everything is case sensitive here.

Widget Class

Each widget has one main widget class. It needs to be in a PHP-file with the same name as the widget you are creating, and must be called an uppcased version of the same name, with a `_Widget` suffix.

It should have a function called `index` that prints the widget.

It may optionally also have an `options` function that specifies custom widget arguments as shown in the example [Writing the widget file](#) on page 103

If you call your widget `hello_world`, then `hello_world.php` can contain a simple echo as shown below:

```
{
    public function index()
    {
        echo "Hello World!";
    }
}
```

View file

While you could print your widget content from the index function as above, it's better to move your content to a separate view file to distinguish functions and content.

If you create the file `view.php` with the content: Hello World!

then you include it from your index function in the widget file by changing it to:

```
public function index()
{
    require($this->view_path('view'));
}
```

More about these sections in: [Writing a simple widget](#) on page 103

Writing a simple widget

In this example we will create a small `hello_world` widget. We will assume you have:

- ssh access to the server
- required knowledge about PHP
- knowledge about how to use an editor in a Linux environment.

We also assume that you, before you start, log in to the op5 Monitor server.

Creating the directory structure

To create the directory structure

- 1 Go to the application folder:

```
cd /opt/monitor/op5/ninja/application
```
- 2 Create the following folders:

```
mkdir -p custom_widgets/hello_world  
mkdir custom_widgets/hello_world/css  
mkdir custom_widgets/hello_world/images  
mkdir custom_widgets/hello_world/js
```

Writing the widget file

To write the widget file

- 1 Go to the widget folder:

```
# cd /opt/monitor/op5/ninja/application/custom_widgets/  
hello_world
```
- 2 Create a file, with your favorite editor, called:
`hello_world.php`
- 3 Type in the following content in the file:

```
<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```
class Hello_world_Widget extends widget_Base {  
    public function options()  
    {  
        // Load default options, like refresh frequency  
        $options = parent::options();  
        // Add option for specifying a custom greeting  
        // (widget_name, option_name, label,  
        // option_type, special_options, default)  
        $options[] = new option('hello_world', 'greeting',  
'Greeting',  
            'input', array(), 'World');  
        return $options;  
    }  
}
```

```
}  
public function index()  
{  
    // get the widget arguments, based on the options above  
    $arguments = $this->get_arguments();  
    require($this->view_path('view'));  
}  
}
```

4 Save and exit from your editor.

Writing the view file

To write the view file

- 1 Go to the widget folder:

```
# cd /opt/monitor/op5/ninja/application/custom_widgets/  
hello_world
```
- 2 Create a file, with your favorite editor, called:
`view.php`
- 3 Type in the following content in the file:

```
<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```
?>  
Hello <?php echo $arguments['greeting']; ?>!
```
- 4 Save and exit from your editor.

Multiple instances

It is possible to spawn multiple instances of a custom widget, like the ones we are shipping in op5 Monitor.

You could use this to create a gazillion of “Hello World”-widgets, or you could create a widget that has multiple datasources like the built-in widget “Unacknowledged service problems” that is described in op5 Monitor User Manual.

To add this functionality, insert the following in the top of your widget class:

```
protected $duplicatable = true;
```

Adding the widget to the widget table

Before we can see the widget on the tactical overview we need to add it to the widgets table in mysql.

To add the widget to the database

Run the following command from your shell:

```
# php /opt/monitor/op5/ninja/index.php cli/save_widget \  
--name=hello_world --friendly_name='Hello World' --page='tac/  
index'
```

Removing a widget

If you for some reason should want to delete a added widget completely you can remove it by deleting it from the mysql database:

```
# mysql -uroot merlin  
> DELETE FROM ninja_widgets WHERE NAME='hello_world'  
> quit
```

Viewing the widget

If everything is done correctly we will now be able to view are first simple op5 Monitor widget.

To view the widget

Open up the Tactical overview in the op5 Monitor user interface and it will look like this:



Take your widget a step further

The example we have been working on here in this chapter is very basic and the output is not much to use. Once you have understood the basics you will probably like to create a more useful widget.

One way to get more information about how you can create a more advanced widget is to take a look at one of the widgets shipped with op5 Monitor.

The Network health widget is a good example. That one can be found here:

```
# /opt/monitor/op5/ninja/application/widgets/netw_health
```



If you are changing any of the default widgets remember to create a copy of the widget, with a new name, and place it in:

```
# /opt/monitor/op5/ninja/application/custom_widgets/
```

Packaging your widget

To make it easy for other users to install and start using your widget you should make a package of it. Then one can install the package in the Tactical Overview in the op5 Monitor GUI.

The package is actually a normal zip file that contains

- the widget in it self
- manifest.xml

The manifest.xml file contains basic data needed by op5 Monitor so that it knows how to install the widget.

Creating the Manifest.xml

To create a Manifest.xml file

- 1 Create an xml file that looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Manifest file for widget to be used in Ninja4Nagios -->
<widget_content>
<author>John Doe</author>
<version>1.0</version>
<Friendly_name>My cool widget</Friendly_name>
<description>A cool widget for op5 Monitor Tactical
Overreview.</description>
<page>tac/index</page>
</widget_content>
```
- 2 Place it in the folder where your widget is located and make sure it is called: manifest.xml

Creating the widget package

Now your widget should be ready and located in the filesystem of your op5 Monitor server.

In this instruction we assume that your widget is called:

my_own_widget

To create a widget package

- 1 Go to the folder above the my_own_widget folder where your widget is located.
- 2 Create the package with the zip command like this

```
# zip -r my_own_widget.zip my_own_widget/
```

Access widgets externally

To access a widget from an external site, like an intranet or network status page.

Server side setup

To configure this you need to configure a contact and a group, edit php-settings for external widgets and insert an iframe on your external web site.

Contact configuration

To set up a widget contact you first need to create a access right. When that is configured create a contact with the same name as the access right and specify which contact group(s) it should be a member of.

Same as a normal contact the contact groups defines which hosts are visible.

Login as the access right created for the widget.

PHP settings

After the contact is created log in to the op5 Monitor server using SSH.

```
# cp /opt/monitor/op5/ninja/application/config/external_widget.php  
/opt/monitor/op5/ninja/application/config/custom  
# cd /opt/monitor/op5/ninja/application/config/custom
```

Edit the file `external_widget.php` with your favorite editor.

This file has two variables, “`widget_name`” specifies which widget that should be shown by default if no widget is set in the iframe. The next one is “`username`” and this sets the user that should be allowed to fetch the widget.

When we set a user name here that user will no longer be able to login to op5 Monitor and will only be a “widget user”.

Example:

```
$config['widget_name'] = 'netw_health';  
$config['username'] = 'jsmith';  
$config['groups'] = array();
```

This example the contact `jsmith` will be used to view widgets and by default it will show the network health widget.

External website setup

On the external website you will need to add an iframe in which the widget is displayed.

The format of the iframe look like this:

```
<iframe src="http://<SERVER_NAME>/ninja/index.php/  
external_widget/show_widget/<OPTIONAL WIDGET_NAME>" height="500px"  
frameborder=0 width="600px" scrolling='no'></iframe>
```

In this iframe you will need to change the `<SERVER_NAME>` to you op5 monitor host name and `<OPTIONAL_WIDGET_NAME>` can either be removed and the default widget will be used or you can specify a widget name to view another widget.

The widgets names can we found in the folder `/opt/monitor/op5/ninja/application/widgets`. The folder names is the same as the widget name.

Widget Porting guide

In op5 Monitor 5.6 the whole widget system is redeveloped to add more functionality such as multiple instances of a widget. If you have created widgets in op5 Monitor prior to 5.6 they will not work out of the box.

Therefore we have created a porting guide to make your home brewed widgets to work in op5 Monitor 5.6 and later.

This information can also be found in your op5 Monitor installation:

```
# /opt/monitor/op5/ninja/applications/widgets/PORTING_GUIDE
```

Java script

In old-style widgets, you usually needed to create a java script file with the following content:

```
$(document).ready(function() {  
var my_widget = new widget('my_widget', 'widget-content');  
});
```

If that was all your java script file contained, you can now safely remove it. If it did more things, you may no longer initialize the widget yourself, but must instead wait for the widget system to load your widget:

```
widget.register_widget_load('my_widget', function() {  
var my_widget = this;  
});
```

If your java script also kept track of your custom configuration options, that, too, can likely be removed - see [Extra Settings](#) on page 114

If you do custom things to your java script, you may need to adjust it, should you want to make it possible to create multiple instances of the widget. This is described in [Multiple Instances](#) on page 115

View

Old widgets that supported ajax refreshes all had to copy-paste the following:

```
<?php defined('SYSPATH') OR die('No direct access allowed.');?>  
<?php if (!$ajax_call) { ?>
```



```

<div class="widget editable movable collapsable removable
closeconfirm" id="widget-<?php echo $widget_id ?>">
<div class="widget-header"><span class="<?php echo $widget_id
?>_editable" id="<?php echo $widget_id ?>_title"><?php echo $title
?></span></div>
<div class="widget-editbox">
<?php echo form::open('ajax/save_widget_setting', array('id' =>
$widget_id.'_form', 'onsubmit' => 'return false;')); ?>
<fieldset>
<label for="<?php echo $widget_id ?>_refresh"><?php echo $this-
>translate->_('Refresh (sec)') ?></label>
<input size="3" type="text" name="<?php echo $widget_id ?>_refresh"
id="<?php echo $widget_id ?>_refresh" value="<?php echo
$refresh_rate ?>" />
<div id="<?php echo $widget_id ?>_slider"></div>
<!-- EXTRA CONTROLS HERE -->
</fieldset>
<?php echo form::close() ?>
</div>
<div class="widget-content">
<?php } ?>
<!-- WIDGET CONTENT HERE -->
<?php if (!$ajax_call) { ?>
</div>
</div>
<?php } ?>

```

With the new widget system, you should remove everything from this file that isn't content. That is, the only thing you should keep in the view, is what you had where it says `<!-- WIDGET CONTENT HERE -->` - everything else should go. Any

extra controls will need to be migrated to the controller - see [Extra Settings](#) on page 114.

Controller

This is the old template for the controller, i.e. the file that had the name of your widget. Not all widgets had all of this, but most of them had most of it:

```

<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```

class My_widget_Widget extends widget_Core {
public function __construct()
{

```

```
parent::__construct();

# needed to figure out path to widget
$this->set_widget_name(__CLASS__, basename(__FILE__));
}

public function index($arguments=false, $master=false)
{
    # required to enable us to assign the correct
    # variables to the calling controller
    $this->master_obj = $master;

    # fetch widget view path
    $view_path = $this->view_path('view');

    if (is_object($arguments[0])) {
        $current_status = $arguments[0];
        array_shift($arguments);
    } else {
        $current_status = new Current_status_Model();
    }

    if (!$current_status->data_present()) {
        $current_status->analyze_status_data();
    }

    $widget_id = $this->widgetname;
    if (isset($arguments['refresh_interval'])) {
        $refresh_rate = $arguments['refresh_interval'];
    }

    $title = $this->translate->_('My Widget');
    if (isset($arguments['widget_title'])) {
        $title = $arguments['widget_title'];
    }

    # let view template know if wrapping div should be hidden or not
    $ajax_call = request::is_ajax() ? true : false;

    /**
     * Actually do stuff
     */
}
```

```
# fetch widget content
require_once($view_path);

if(request::is_ajax()) {
# output widget content
echo json::encode( $this->output());
} else {
$this->js = array('/js/my_widget');
$this->css = array('/css/my_widget');
# call parent helper to assign all
# variables to master controller
return $this->fetch();
}
}
}
```

This is the new-style equivalent:

```
<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```
class My_widget_Widget extends widget_Base {
public function __construct($model)
{
parent::__construct($model);
/**
* Do any global initiation here
*/
}

public function index()
{
# fetch widget view path
$view_path = $this->view_path('view');

$current_status = $this->get_current_status();
$arguments = $this->get_arguments();

/**
* Actually do stuff
*/

$this->js = array('/js/my_widget');
```

```
$this->css = array('/css/my_widget');
require($view_path);
}
}
```

Note: The widget must inherit from widget_Base instead of widget_Core.

Note: The constructor now takes an argument, index takes none.

Note: You must not use require_once to include the view if you intend to allow multiple widget instances

Note: Still no file endings on java script and css resources.

Extra Settings

This used to be a free-form div in the view, however, it was mostly just cut and pasted from widget to widget, so we have implemented the redundant stuff

once, so you won't have to. You now add a method to the controller, options, and have it return an array of your extra options. This is the last example widget again, but with two extra settings:

```
<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```
class My_widget_Widget extends widget_Base {
    public function options() {
        $options = parent::options();
        $options[] = new option('my_widget', // your widget name (or
            something else unique)
                'option1', // a unique option name
                $this->translate->_('My first option'), //
your label
                'input', // option type - input, checkbox,
dropdown, etc
                array('size'=>5), // extra attributes for
the field
                'default1'); // default value
        $options[] = new option('my_widget',
                'option2',
                'My second option',
                'input',
                array('size'=>5),
                'default2');

        return $options;
    }
}
```

```

    }

    public function index()
    {
        # fetch widget view path
        $view_path = $this->view_path('view');

        $current_status = $this->get_current_status();
        $arguments = $this->get_arguments();

        /**
         * Actually do stuff
         */

        require($view_path);
    }
}

```

Note: This will automatically create java script to save any changes and refresh the page on changes. If you want to do this manually, you must call `should_render_js(false)` on the option object.

Note: If you want to, you can return a pure HTML string of the widget settings you want to keep track of. That way, you will get to do everything yourself.

Multiple Instances

For simple widgets, to enable multiple instances you will only have to add one single line of code to the constructor: `"protected $duplicatable = true;"`. This is a simple hello world widget that can be duplicated:

```

<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```

class My_widget_Widget extends widget_Base {
    protected $duplicatable = true;
    public function index()
    {
        print "Hello world!";
    }
}

```

If your widget is more complicated, you will probably have to change more things.

First, it's likely that your widget includes fields with the id attribute. Doing so is no longer valid - you must either create a globally unique name using both the widget's name and instance id, or you should use a class attribute instead.

Then, in your javascript, you must take care to use a combined selector to retrieve the HTML node you want to for the correct widget instance. In the past, a common pattern in java script files was the following:

```
$(document).ready(function() {  
  var my_widget = new widget('my_widget', 'widget-content');  
  $('#my_widget_setting').change(function() {  
    my_widget.save_custom_val($(this).val(), 'my_widget_setting');  
    do_something();  
  });  
});
```

Again, if do_something is only a widget reload, you can remove this code

completely. If you do more things, you need to take more care.

This is how the above should be written with new-style, multi-instance widgets:

```
widget.register_widget_load('my_widget', function() {  
  var my_widget = this;  
  $('#'+my_widget.widget_id+'  
    .my_widget_setting').change(function() {  
    my_widget.save_custom_val($(this).val(), 'my_widget_setting');  
    do_something();  
  });  
});
```

That is, you can safely search for the class within the widget instance id.

GUI skins

Introduction

All views in the op5 Monitor user interface are built up with help of skin templates. In the default op5 Monitor installation there is three of them:

- Default
- Dark, a darker theme that is suitable for big screens
- Pink-n-fluffy, it's pink n' fluffy!

If you like to change any of the parts of the op5 Monitor user interface the best way to do that is to create your own theme. That makes sure you do not lose any changes in an upgrade later on. Of course you have to update your theme by your self to be able to enjoy many of the new features that comes with op5 Monitor updates.

In this chapter we will take a closer look of how the theme is built up and some minor changes that can easily be made.

The files and folders

An op5 Monitor theme includes a lot of folders and files. Most of them are never a subject to be changed but you still need them.

All skins shall be placed in a folder of its own directly under:

```
/opt/monitor/op5/ninja/application/views/css/
```


All views have their own folder named after the controller they belong to. The view folders can contain everything from one single PHP file to a complex structure of folders, code files (PHP, java script, css), images etc.



Almost all controllers have their corresponding view in the user interface.

Beside the view folders we have the following folders and files:

- admin/
- css/
- css_header.php
- error.php
- icons/
- js/
- js_header.php
- kohana_unit_test.php
- login.php
- menu.php
- ninja_start.php
- template.php
- unauthorized.php

A more detailed description of the files listed above is shown in the table below.

File/Folder	Description
admin/	For future functions in the user interface. Not in use at the moment.
css/	CSS files that is used by the controllers. The controller it self decides what file to use.
css_header.php	Locates and enables the files in <code>css/</code> for the controllers.  Do not touch this file!
error.php	A general error messages template.
icons/	All icons used in the user interface.
js/	Java scripts that is used by the controllers. The controller it self decides what file to use.

File/Folder	Description
js_header.php	Locates and enables the files in js/ for the controllers.  Do not touch this file!
kohana_unit_test.php	 Do not touch this file!
login.php	The user interface login page.
menu.php	Deprecated!
ninja_start.php	Deprecated!
template.php	The template file in it self. This is the one that creates the main parts of the user interface.
unauthorized.php	A general unauthorized messages template.

Make your own skin

Before you start

Before you can start making changes to the please make sure you have

- ssh and sftp access to the op5 Monitor server
- created your own theme. In all instructions in the rest of the chapter we assume you already have logged in via ssh on the op5 Monitor server. We also assume that you have the basic knowledge needed in PHP and knows how to work in a Linux environment.

The theme we create here will be called:

```
my_skin
```

Install SASS

SASS or Syntactically Awesome Stylesheets is a CSS metalanguage. It is a scripting language that is interpreted into CSS.

To install SASS go to the folder `/opt/monitor/op5/ninja/` and run the command:

```
make install-sass
```

This will install the SASS and compass components that we will use later. This only has to be done once.

Creating your own skin

To create your own skin

- 1 Go to the skin folder:

```
cd /opt/monitor/op5/ninja/application/views/css
```
- 2 Copy the Pink n' Fluffy skin to a new directory with the name of your new skin:

```
cp -a pink_n_fluffy/ my_skin
```
- 3 Go to the folder `/opt/monitor/op5/ninja/application/views/css/my_skin/sass`
- 4 Edit the file `common.sass`
Change the variables to change the look of you skin.



All variables can be found in the file `/opt/monitor/op5/ninja/application/views/css/base-sass/config.sass`

- 5 Save the file and compile the skin. See [Compiling skins](#) on page 122

Compiling skins

For the skin to work the SASS files has to be compiled to CSS.

To do this go to the skin css folder:

```
/opt/monitor/op5/ninja/application/views/css/my_skin
```

Run the command

```
compass compile
```

Modify report skin

It is possible to change the header and footer of the report.

Start by creating new header and footer html files, the originals can be found at: /
opt/monitor/op5/ninja/application/media

Use these files as templates and save your files in the same directory.

After this the reports.php needs to be modified to include you header and footer.
Copy the reports.php to the custom folder:

```
cp /opt/monitor/op5/ninja/application/config/reports.php /opt/  
monitor/op5/ninja/application/config/custom
```

Edit the file /opt/monitor/op5/ninja/application/config/custom/
reports.php

At the end of the file this line has to be changed, point the report_header.html and
report_footer.html to your modified files.

```
$config['pdf_command'] = '/usr/bin/wkhtmltopdf -q --print-media-  
type --disable-internal-links --disable-external-links --margin-  
top 10 --margin-bottom 10 --header-spacing 5 --footer-spacing 5 --  
header-html application/media/report_header.html --footer-html  
application/media/report_footer.html - -';
```

Changing what theme op5 Monitor use

To change what theme op5 Monitor shall use

- 1 Go to the application folder:

```
cd /opt/monitor/op5/ninja/application/config
```
- 2 Open up config.php in your favorite text editor.
- 3 Look up and change the following line and change the theme name from
default/ to my_theme/ in this case:

```
$config['current_skin'] = 'default/';
```
- 4 Save and exit.

Making changes in the user interface

As you probably already have realized you can do almost any kind of changes in the op5 Monitor user interface. Covering them all would require a complete manual of its own. So in this chapter we will only take a look at a few of them.

- Changing the logo.
- Adding hostname to the Quick bar.
- Change the default font.

The topics listed above should give you knowledge to do other modifications by your own.

Changing the logo

One thing you might want to do is to change the default logo up in the left corner of the user interface.



Before you starting

To change the logo

- 1 Make sure you have followed the instructions in:
[Make your own skin](#) on page 121.
- 2 Create your own logo file. It shall meet the following criteria:
Width: 19px
Height: 19px
Type: PNG
- 3 Place the logo in the folder
`/opt/monitor/op5/ninja/application/views/css/my_skin/images/`
We call this `my_logo.png`
- 4 Change the logo in the file:
`/opt/monitor/op5/ninja/application/views/css/my_skin/sass/common.sass`
- 5 Change the following line:

```
background: url('../icons/icon.png') no-repeat  
to  
background: url('../icons/my_logo.png') no-repeat
```
- 6 Save the file and compile the skin. See [Compiling skins](#) on page 122

Creating custom logo per login

It is possible to have custom logos based on the name of the username.

Ie. when hosting a monitoring solution for other companies these companies can use their logo.

The logo is selected using regex on the login name.

If user 'companyA_Joe' is logged in the companyA logo is displayed, but if companyB_Joe is logged in company B logo is displayed.

To enable this feature edit the following file:

`/opt/monitor/op5/ninja/application/config/customlogo.php`

- 1 Change `$config['enable'] = false;` to `$config['enable'] = true;`
- 2 Put the logo file (size 19px x 19px) in the folder:
`custom_logos/`
This folder is relative to your icon folder.

Change the default font

Many visual parts in op5 Monitor are setup in css files. Therefore it's a good idea to take a look at them and see how they are used.

In this example we will change the default font and make it a bit bigger.

To change the default font

- 1 Make sure you have followed the instructions in:
[Make your own skin](#) on page 121.
- 2 Go to the folder of your theme:
`cd /opt/monitor/op5/ninja/application/views/themes/my_theme`
- 3 Open up the `css/default/common.css` file in your favorite text editor.
- 4 At the top of the file `common.css` you will find the following lines:

```
* {
text-decoration: none;
font-size: 1.0em;
outline: none;
padding: 0;
margin: 0;
}
```

Change `font-size: 1.0em;` to:
`font-size: 1.02em;`
- 5 Save and exit from the editor.
- 6 Refresh the op5 Monitor user interface in your browser and you can see that the default font is a bit bigger now.

User menus

Customize user menus

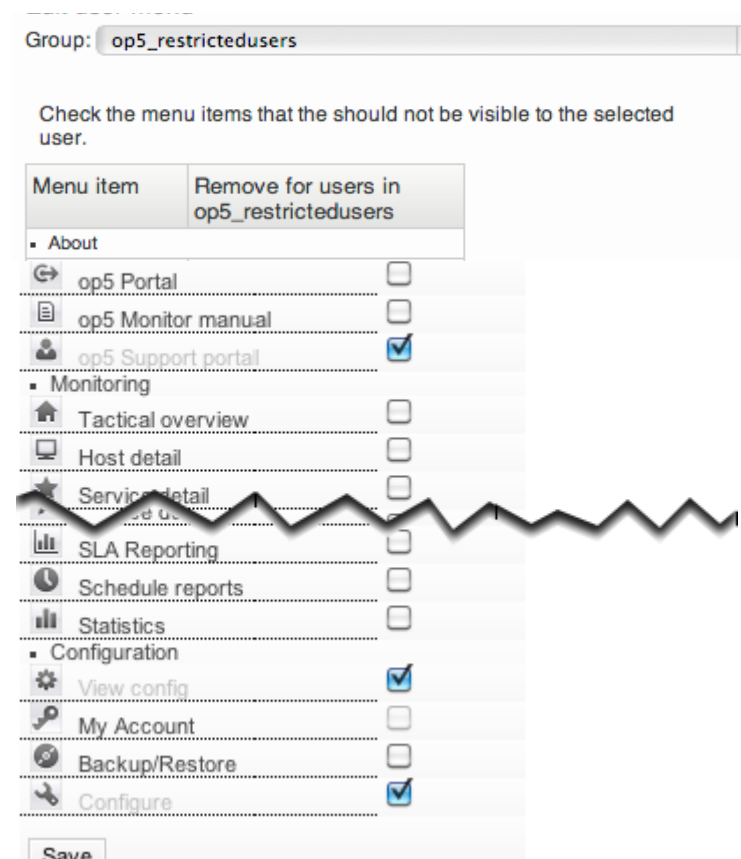
It is possible for an administrator to customize users menu.

The customizing is done per usergroup and not on individual users

Only user with full access can edit usergroup menus.

To change a specific usergroup menu, go to 'My Account' in the menu and click on 'Edit user menu'. Select the group you want to change the menu for.

You can now hide the options in the menu that you don't want to be visible for that members in that group. In the example below we have removed 'op5 Support portal', 'View Config' and 'Configure' options.



Group:

Check the menu items that the should not be visible to the selected user.

Menu item	Remove for users in op5_restrictedusers
■ About	
➔ op5 Portal	<input type="checkbox"/>
📄 op5 Monitor manual	<input type="checkbox"/>
👤 op5 Support portal	<input checked="" type="checkbox"/>
■ Monitoring	
🏠 Tactical overview	<input type="checkbox"/>
🖨 Host detail	<input type="checkbox"/>
🌟 Service detail	<input type="checkbox"/>
📊 SLA Reporting	<input type="checkbox"/>
🕒 Schedule reports	<input type="checkbox"/>
📈 Statistics	<input type="checkbox"/>
■ Configuration	
⚙ View config	<input checked="" type="checkbox"/>
🔑 My Account	<input type="checkbox"/>
💾 Backup/Restore	<input type="checkbox"/>
🔧 Configure	<input checked="" type="checkbox"/>

When you are done, click on save.

Localization

Introduction

In op5 Monitor we have the possibility to show all texts in your own language. This so called localization is done with help from gettext, which is a part of the Linux translation project.

There are many different ways to work with the gettext files. You can use

- the cli commands from the gettext installation
- a program like poedit.

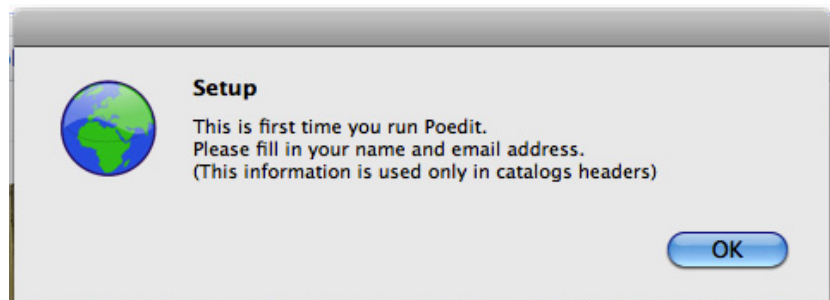
Here in this chapter we will use poedit to add a new language to op5 Monitor.

Downloading and starting the tools

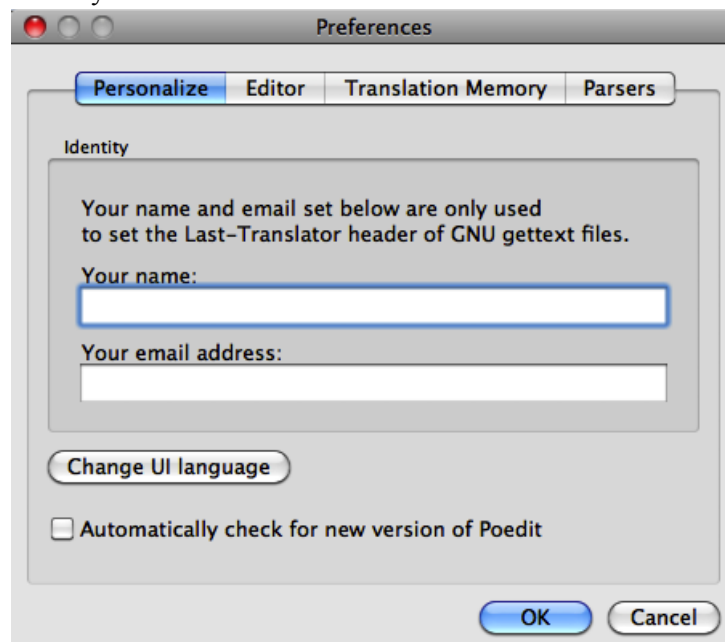
The first thing we need to do is to download the tools. In this case just only the Poedit

To download and starting the tools:

- 1 Go to:
<http://www.poedit.net/download.php>
- 2 Download and install the version needed for you OS.
- 3 Start poEdit and follow the instructions.
 - a Click **OK**



- b Fill in your name and email address and click **OK**.



Adding a new language

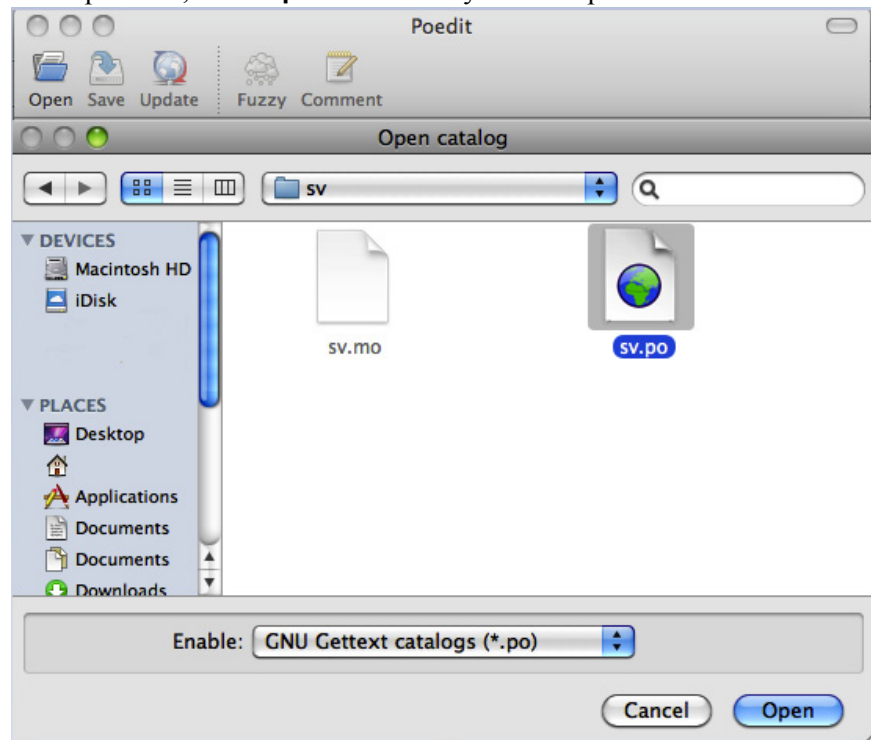
In this example we will create add a new language (swedish). We will also use the Poedit tool and work with the files locally on our workstation and then copy the new language files to the op5 Monitor server.

To add a new language

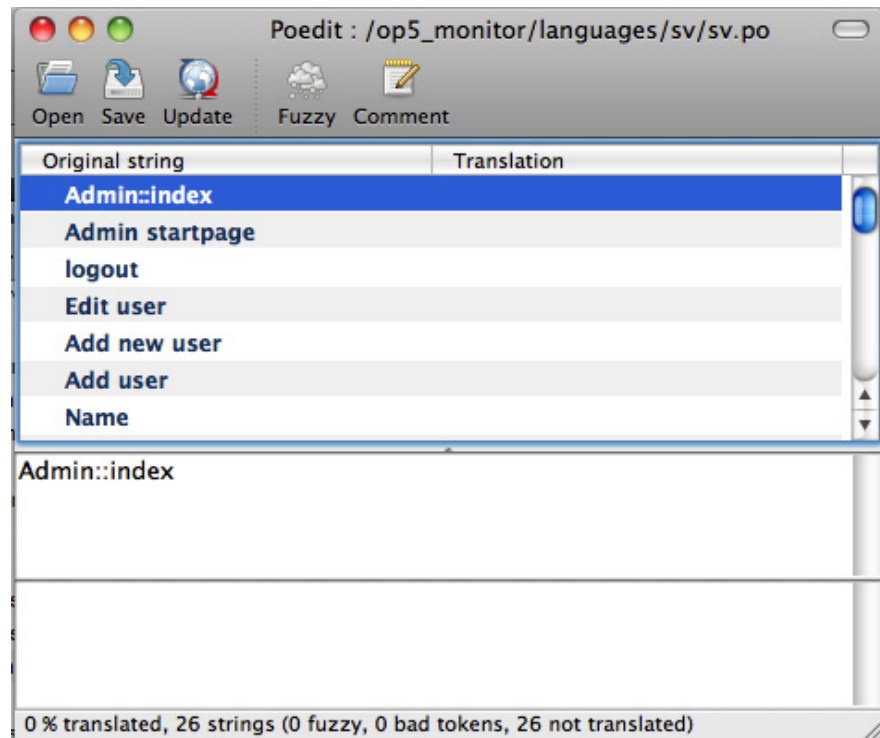
- 1 Copy the language folder to your workstation.
- 2 Create a new folder inside the language folder with the short name of the language you like to create.

```
mkdir sv/
```
- 3 Copy the files from the en/ folder (english) to the new one and rename them to the same name as the folder like this:

```
cp en/en.mo sv/sv.mo  
cp en/en.po sv/sv.po
```
- 4 Start up Poedit, click **Open** and locate your new .po file.



- 5 Mark the line you like to add your translation to and type in the field at the bottom of the window.



- 6 Save the file when done.

Changing basic language file settings

Now the last thing we shall do, before we upload the files to the server, is to change some of the basic settings in the language file.

The things we are going to change is:

- Team
- Language
- Country

To change the basic language file settings

- 1 Open up the language file (sv.po, in this example) in Poedit.
- 2 In the main menu click **Catalog** -> **Settings...**
- 3 Change the **Team** to what ever you want and the **Language** and **Country** to reflect the language you are creating. In this example:
Language: Swedish
Country: SWEDEN
- 4 Click **OK**.

Applying the new language to the server

Now the last thing we need to do is to send up the new language to the server.

To apply the new language to the server

- 1 Copy the new language folder (in this example `sv/`) and its content to the following folder on the op5 Monitor server:
`/opt/monitor/op5/ninja/application/languages/`
- 2 Open up your browser and change the settings so the new language will be the first one to use.
In FireFox this is done in:
Preferences -> Content -> Languages
- 3 Go to the op5 Monitor user interface login page. If you have translated all lines you will now see the login page in your new language.

Graphs

Introduction

op5 Monitor is using PNP to create the graphs available for most standard services in the user interface.

PNP is an add-on to nagios which analyzes performance data provided by plugins and stores them automatically into RRD-databases (Round Robin Databases).

PNP only processes performance data built according to the Developer Guidelines for nagios plugins. With this limitation we want to honour the work of Nagios Plugin Developers who stick to the guidelines.

This is a short description of how to use PNP and it's functions pages and templates.

For more info please refer to the online manual for pnp

<http://www.pnp4nagios.org/pnp/start>

Kudos to **Joerg Linge** for letting us use his text.

Graph web front end

The behavior of the PNP Web-Front end can be controlled through the config file

`/opt/monitor/etc/pnp/config.php`.

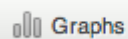
This file will be overwritten during updates of PNP as the paths and options are detected during `./configure`.

Own adjustments should be made in:

`/opt/monitor/etc/pnp/config_local.php`

If this file does not exist the file `config.php` can be taken as a guideline.

To access the PNP web front end through the GUI click on **Graphs** in the menu.



Collections

About Collections

Collections provides the opportunity to collect graphs of different hosts and services on to one page. That way - as an example - you can display the traffic rates of all tape libraries.

Creating a new collection

The setup of Graph Collections is done through the configuration page.

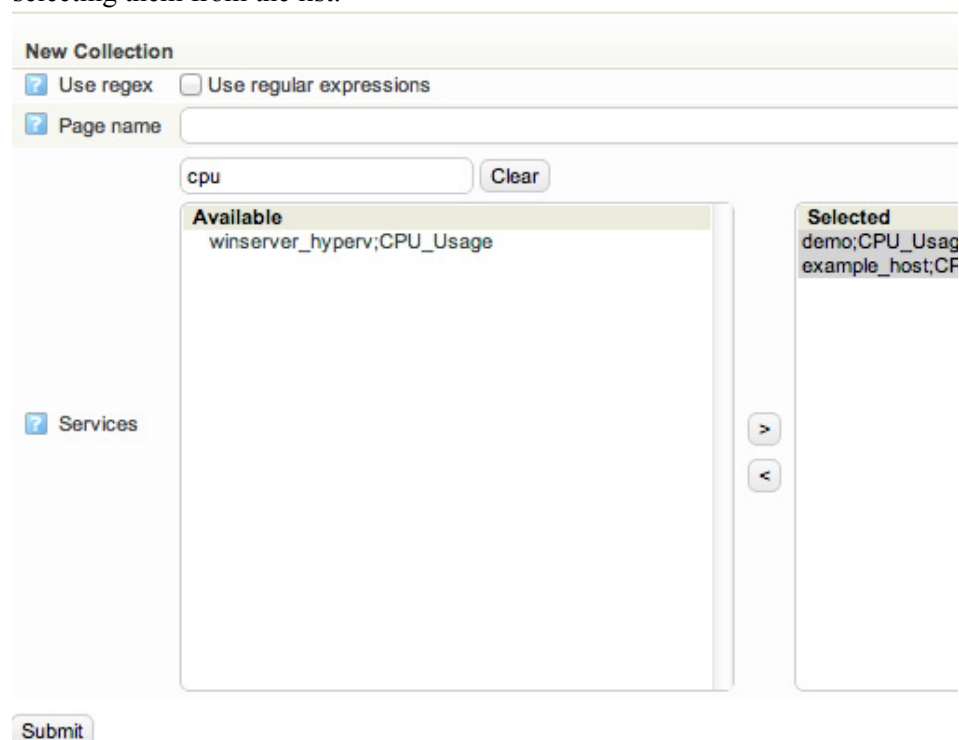
Go to Configure and click on the **Graph Collections** icon.



There are two ways to select which services to show in the graph, either use the GUI to select the services from the list or use regular expressions.

GUI selection

Enter a collection name and select which services to put in the collection by selecting them from the list.



The screenshot shows the 'New Collection' configuration page. At the top, there's a title bar 'New Collection'. Below it, there are two checkboxes: 'Use regex' (checked) and 'Use regular expressions' (unchecked). Below these is a 'Page name' input field. Underneath the input field is a text box containing 'cpu' and a 'Clear' button. To the left of the main list is a 'Services' section with a help icon. The main area is divided into two panes: 'Available' and 'Selected'. The 'Available' pane contains the text 'winservr_hyperv;CPU_Usage'. The 'Selected' pane contains the text 'demo;CPU_Usag' and 'example_host;CF'. Between the two panes are two arrow buttons, one pointing right and one pointing left. At the bottom left is a 'Submit' button.

Regex selection

Check the checkbox for **Use regex**

New Collection	
<input type="checkbox"/> Use regex	<input checked="" type="checkbox"/> Use regular expressions

The host and services is now selected by a regular expression.

In the example below we select all graphs from the host which names starts with “switch” and services that contains “Interface” and “Traffic”. Note that regular expressions are case sensitive.

Switch_traffic	
<input type="checkbox"/> Use regex	<input checked="" type="checkbox"/> Use regular expressions
<input type="checkbox"/> Page name	Switch_traffic
<input type="checkbox"/> Host name	^switch
<input type="checkbox"/> Service description	Interface.*Traffic

Viewing Collections

The collections are found under graphs in the main menu on the left and click on the **Collection icon**



Note that this icon is only visible when at least one collection is created.

All the collections are listed in the panel to the right.

Collections
<input checked="" type="checkbox"/> CPU Usage
<input checked="" type="checkbox"/> Switch_traffic

Combined Graphs

What is a combined graph?

A combined graph overlays several graphs in the same graph which will give a better overview of how two different checks are performing.

It takes one or more service from one or more host and lays them on top of each other in the same graph.

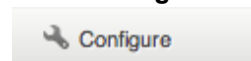


The service checks must have the same name on all the hosts for combined graphs to work.

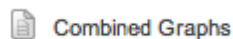
Creating combined graphs

The combined graph is created through the configuration.

Go to **Configure** in the menu



Click on **Combined Graphs**



Enter a name of the combined graph and click on **Add**

Name of Combined Graph:

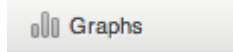
Select the service to graph and from which hosts this service should be fetched from. Also add a name and comment.

<input checked="" type="checkbox"/> Title	<input type="text" value="CPU Usage"/>
<input checked="" type="checkbox"/> Comment	<input type="text" value="Windows CPU Usage"/>
<input checked="" type="checkbox"/> Type	<input type="text" value="Area"/> <input type="button" value="Clear"/>
<input checked="" type="checkbox"/> Services	<div><div>Available</div><div>demo;CPU Usage example_host;CPU Load</div><div>Selected 172.27.86.208;CP winserver_hyperv</div></div>

Click on **Save**

Viewing combined graphs

The combined graphs can be found under **Graphs** in the main menu



and click on the **Combined Graphs** icon.



All the combined graphs can be found under **Combined graphs** in the right side menu.



Graph templates

Graph templates controls how your graphs look.

By changing the template for on check command you can customize how the graph for that check will look.

Changing Graph template

By default a the template default is used for drawing graphs. This can be changed for a specific check command.

To change the template for a check command go to **configuration** and select **graph templates**. Select a template that you would like to modify, then press **go**.

Graph template

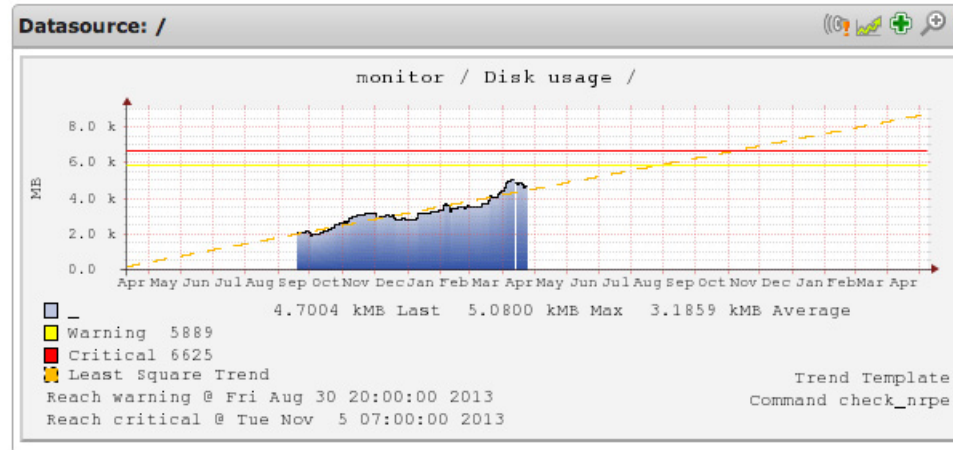
Filter by regular expression

Select the check command that will use this temple and select the template. There are several templates in the system by default.

check_nrpe_win_drivesize Template	
<input type="checkbox"/> Check command	<input type="text" value="check_nrpe_win_drivesize"/>
<input type="checkbox"/> Graph template	<input type="text" value="IsIprediction-label"/>
<input type="button" value="Submit"/>	

Graph prediction

A graph prediction plots a line across the graph to predict when it will cross the thresholds.



Activate graph prediction

The graph prediction is not used by default. To enable graph prediction you have to change the template that is used for the specific check command.

Edit the template for the check command, follow instructions in [Changing Graph template](#) on page 140.

Select the template **IsIprediction-label** and click on **submit**.

check_nrpe Template	
Check command	check_nrpe
Graph template	IsIprediction-label
<input type="button" value="Submit"/>	

Note: This will change the layout of your graphs.

Business Service

Introduction

The business process view is designed to combine your IT monitoring and your business service management (BSM) to give an overview of the applications and/or services that your organisation is providing either to customers or internally.

Business services

About business services

A business object is a group that can be populated with hosts and services from hosts and host groups.

It is also possible to add sub-groups that can have their own rule-set.

Creating a new group

To create a new group go to “Business Process” in the menu.

Business Services

Click on the “New Object” button at top-right area

New Business Service

Select a rule for your business process group

Select

Business Service Groups



At most

State depends on problems thresholds



Worst state

Returns the worst state of all its subelements



Best state

Returns the best state of all its subelements

Fill in the name of the group

Name:

Object name displayed in Business Services

Demo Service

Enter the parameters of the rule, if any.

WARNING threshold:

Count of problems to become WARNING

5

CRITICAL threshold:

Count of problems to become CRITICAL

3

Unit:

Unit for the above value (percentage or actual number)

num

Click on the “Create” button

create

Click on the “Save” icon.



Creating a sub-element

A sub-element is either a service of a host or another group with it’s own rule-set.

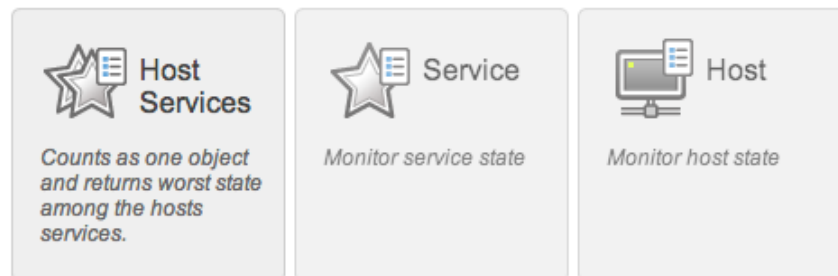
To add the sub-element click “add sub-element” icon in actions icons column



Add a monitored object

Select what type of object you want to add.

Monitor Objects



Select which object by clicking in the empty text field and select you object from the drop-down menu.

Service: <i>Host name and service description</i>	win-server1
State type: <i>State type</i>	hard

create

- CPU usage
- DHCP Server
- Disk usage C:
- Disk usage E:
- FTP
- IIS Admin Service
- Memory usage
- PING
- POP3 Server

Click on the “Save” icon.



Add a group as sub-element

Select the type of rule-set the group shall have. Then follow the steps in [Creating a new group](#) on page 145.

Rules types

There are currently 6 different rule types to choose from, each group has their unique rule set.

Group	Description
Worst state	Returns the worst state of all its sub-elements
Best state	Returns the best state of all its sub-elements
Simple at least	Returns OK if at least one sub-element are OK
At least	Returns OK if at least X sub-elements are ok and WARNING if Y sub-elements is OK.
Scores	The state depends on the number of points scored by its sub-elements.
Custom	Custom rules sets can be created.

Worst state

Group state will be the worst state of all its sub-elements

Examples

- Worst State of {OK, WARNING, CRITICAL} => CRITICAL
- Worst State of {OK, WARNING, CRITICAL, UNKNOWN} => UNKNOWN

Best state

Group state will be the best state of all its sub-elements

Examples

- Best State of {WARNING, CRITICAL} => WARNING
- Best State of {OK, WARNING, CRITICAL, UNKNOWN} => OK

Simple At least

Means to express the idea that you need some amount of services up and running for the delivered service to be functional. The number of sub-elements that has to be OK is specified in percentage or actual amount. If the number of sub-elements that are OK are equal or more than the at-least number or percentage then the group will be OK, or else the group will get the worse state of its sub-elements.

Examples

- Simple At least(2, num) of {OK, OK, CRITICAL, CRITICAL} => OK
- Simple At least(3, num) of {OK, OK, WARNING, CRITICAL} => CRITICAL

- Simple At least(3, num) of {OK, OK, WARNING, WARNING} => WARNING
- Simple At least(50, %) of {OK, OK, WARNING, CRITICAL} => OK
- Simple At least(50, %) of {OK, OK, WARNING, CRITICAL, CRITICAL} => CRITICAL

At least

Means to express the idea that you need some amount of services up and running to be functional and lesser amount to be semi-functional (e.g. with degraded performance). Two thresholds are specified, percentage or actual among is possible:

If the number of OK sub-elements is greater or equal than the OK threshold then group is OK

If the number of OK sub-elements is less than the OK threshold but greater or equal than the WARNING threshold then group is WARNING

If number of OK sub-elements is less than the WARNING threshold then group is CRITICAL

Examples

- At least(2,1,num) of {OK, OK, WARNING, CRITICAL} => OK
- At least(3,2,num) of {OK, OK, WARNING, CRITICAL} => WARNING
- At least(3,2,num) of {OK, WARNING, WARNING, CRITICAL} => CRITICAL
- At least(3,2,num) of {OK, WARNING, WARNING, WARNING} => CRITICAL

At most

Means to express the idea that you can tolerate some amount of problems. Two thresholds are specified either in percentage or actual among.

If number of problematic sub-elements is greater or equal to the CRITICAL threshold then group is CRITICAL

If number of problematic sub-elements is less than CRITICAL threshold but greater or equal to the WARNING threshold then group is WARNING

If number of problematic sub-elements is less than the WARNING threshold then the group is OK

Examples

- At most(2,1,num) of {OK, OK, WARNING, CRITICAL} => OK
- At most(3,2,num) of {OK, OK, WARNING, CRITICAL} => WARNING

- At most(3,2,num) of {OK, WARNING, WARNING, CRITICAL} => CRITICAL
- At most(3,2,num) of {OK, WARNING, WARNING, WARNING} => CRITICAL

Scores

Means to express the idea that having several WARNING sub-elements is the same as having few OKs and few CRITICALs. Groups sums the problems points of all its children using:

OK state gives 0 problems points

WARNING = 1

CRITICAL = 2

UNKNOWN = 3

Then checks it against two specified thresholds.

If sum is less than the WARNING points then group is OK

if sum is between the WARNING and CRITICAL points then group is WARNING

if sum is greater or equal than the CRITICAL points then group is CRITICAL

Examples

- Scores(4,3,num) of {OK, OK, WARNING, CRITICAL} => WARNING
- Scores(4,3,num) of {OK, WARNING, WARNING, WARNING} => WARNING
- Scores(4,3,num) of {WARNING, WARNING, WARNING, WARNING} => CRITICAL
- Scores(4,3,num) of {OK, OK, CRITICAL, CRITICAL} => CRITICAL

Custom rules

It is possible to create your own custom rules. This is done in a script language called LUA.

See chapter custom rules (not yet written).

Publish as a service

It is possible to publish an object as a service of a host. By doing this the object will get the same possibilities as a service in Monitor such as notifications, reports, graphs and so on.

To publish an object edit the object that you want to publish and select “as a service” and the host on which to publish the object.

<input checked="" type="checkbox"/> as a Service at: <i>Associate with host for reporting and notification</i>	<input type="text" value="demo"/>
--	-----------------------------------

The object will now be found under the host as a service.

Reporting

When creating a SLA or Availability report out of a BSM top level element that has been published to a host it is possible to include the BSM events in that report.

This will show the underlying checks that triggered a WARNING or CRITICAL alert in the BSM.













To enable this choose to include BSM event when creating a report.

If the selected BSM service is not a top level element, no BSM events will be displayed.

To enable this choose to **include BSM event** when creating a report.

 ☒ **Include BSM events**

When the report is generated the event will be included in the report.

2013-04-17 14:12:02		Webshop (saved as service on host bsm_host)	Problem count: 3. Got 0 OK(s), 2 WARNING(s), 1 CRITICAL(s), 0 UNKNOWN(s) ↳  Webservers: Problem count: 3. Got 0 OK(s), 1 WARNING(s), 2 CRITICAL(s), 0 UNKNOWN(s) ↳  Web1 ↳  Web2 ↳  Web3 ↳  Databases: Problem count: 2. Got 1 OK(s), 1 WARNING(s), 0 CRITICAL(s), 0 UNKNOWN(s) ↳  DB3 ↳  DB1 ↳  Loadbalancing: Problem count: 2. Got 1 OK(s), 1 WARNING(s), 1 CRITICAL(s), 0 UNKNOWN(s) ↳  LB3 ↳  LB2
2013-04-17 14:17:02		Webshop (saved as service on host bsm_host)	All criteria fulfilled for an OK state

Notifications

Introduction

In this chapter we will take a deeper look at the notification function in op5 Monitor. We will look at how the

- notification works
- notification skins works (mail/sms/htmlpost)
- dial up notification works
- snmp trap notification works.

How does notifications work?

In the op5 Monitor user manual we describe some of the basics with notifications. Let us take a closer look at how it really works.

Notification filters

When a notification is about to be sent it has to go through a number of filters before op5 Monitor can determine whether a notification really is suppose to be sent or not.

Table 1 Notification filters

Filter	Description
Program-wide	This tells op5 Monitor if notifications are turned on or not in a program-wide basis.
Service and host filters	<ul style="list-style-type: none">• Is the host or service in scheduled downtime or not?• Is the host or service in a flapping state?• Does the host or service notification options says that this type of notification is supposed to be sent?• Are we in the right time period for notifications at the moment?• Have we already sent a notification about this alert? Has the host or service remained in the same non-OK state that it was when the last notification went out?
Contact filters	<ul style="list-style-type: none">• Does the contacts notifications options says that this type of notification is supposed to be sent?• Are we in the right time period for notifications at the moment, according to the notification time period set on the contact?

Notification commands

How the notifications are sent is defined in either one of the two files below:

- checkcomands.cfg
- misccommands.cfg

The commands are divided into

- host notification commands

- service notification commands

The notification commands are then using scripts in the same way as the normal check commands does.

All default scripts shipped with op5 Monitor is located in:

`/opt/monitor/op5/notify`

Notification macros

Many of the arguments sent to the notification commands are macros. The macros are a sort of variables containing a, in most cases, program-wide value. You can read more about macros in the Nagios manual:

http://nagios.sourceforge.net/docs/3_0/macros.html

One of the most important macro used with notifications is:

`$NOTIFICATIONTYPE$`

This macro tells you what type of notification that is supposed to be sent. The `$NOTIFICATIONTYPE$` macro can have one of the following values.

Table 2 Notification types

Notification type	Description
PROBLEM	A service or host has just entered (or is still in) a problem state.
RECOVERY	A service or host has recovered from a problem state.
ACKNOWLEDGEMENT	A service or host in a problem state has been acknowledged by a user.
FLAPPINGSTART	The host or service has entered a flapping state.
FLAPPINGSTOP	The host or service has left a flapping state.
FLAPPINGDISABLED	The host or service flapping detection has stopped and has there fore left the flapping state.
DOWNTIMESTART	The host or service has entered a scheduled downtime.
DOWNTIMESTOP	The host or service has left a scheduled downtime.
DOWNTIMECANCELLED	The scheduled downtime for a host or service has been cancelled.

The list of macros described in the Nagios manual is very useful when you are working with new notification commands and scripts. That list can be found here:

http://nagios.sourceforge.net/docs/3_0/macrolist.html

Notification e-mail sender

Notifications are by default sent from the e-mail address "op5monitor" without any domain. The MTA adds the local domain name, which by default is "@localhost.localdomain".

To change the e-mail address that notification are sent from use the --from-mail argument for the notification command.

To change the sender e-mail address from op5monitor@localhost.localdomain to op5notification@mycompany.com simply go to the check command for the host-notify and add "--from-email op5notification@mycompany.com" without the "-" signs.

```
command_name=host-notify
command_line=$USER3$/notify/poller_notify_send.pl --from-email
op5notification@mycompany.com -c "$CONTACTNAME$" -h "$HOSTNAME$" -
f "$NOTIFICATIONTYPE$" -m "$CONTACTEMAIL$" -p "$CONTACTPAGER$"
"HOSTALIAS=$HOSTALIAS$" "HOSTADDRESS=$HOSTADDRESS$"
"HOSTSTATE=$HOSTSTATE$" "HOSTSTATEID=$HOSTSTATEID$"
"HOSTSTATETYPE=$HOSTSTATETYPE$" "HOSTATTEMPT=$HOSTATTEMPT$"
"HOSTLATENCY=$HOSTLATENCY$"
"HOSTEXECUTIONTIME=$HOSTEXECUTIONTIME$"
"HOSTDURATION=$HOSTDURATION$" "HOSTDURATIONSEC=$HOSTDURATIONSEC$"
"HOSTDOWNTIME=$HOSTDOWNTIME$"
"HOSTPERCENTCHANGE=$HOSTPERCENTCHANGE$"
"HOSTGROUPNAME=$HOSTGROUPNAME$" "HOSTGROUPALIAS=$HOSTGROUPALIAS$"
"LASTHOSTCHECK=$LASTHOSTCHECK$"
"LASTHOSTSTATECHANGE=$LASTHOSTSTATECHANGE$"
"LASTHOSTUP=$LASTHOSTUP$" "LASTHOSTDOWN=$LASTHOSTDOWN$"
"LASTHOSTUNREACHABLE=$LASTHOSTUNREACHABLE$"
"HOSTOUTPUT=$HOSTOUTPUT$" "HOSTPERFDATA=$HOSTPERFDATA$"
"HOSTACKAUTHOR=$HOSTACKAUTHOR$" "HOSTACKCOMMENT=$HOSTACKCOMMENT$"
"NOTIFICATIONNUMBER=$NOTIFICATIONNUMBER$"
"CONTACTALIAS=$CONTACTALIAS$" "DATETIME=$DATETIME$"
"SHORTDATETIME=$SHORTDATETIME$" "DATE=$DATE$" "TIME=$TIME$"
"TIMET=$TIMET$" "HOSTACTIONURL=$HOSTACTIONURL$"
"HOSTNOTESURL=$HOSTNOTESURL$" "ADMINPAGER=$ADMINPAGER$"
"ADMINEMAIL=$ADMINEMAIL$"
"NOTIFICATIONCOMMENT=$NOTIFICATIONCOMMENT$"
```

This has to be done for the command "service-notify" as well.

Notification skins

The three basic notifications (email, sms and htmlpost notifications) are all using something called notification skins. The notification skins are templates describing how the notification is supposed to look like when it is sent to its receiver.

If we will take a look at the notify folder we will find the following skins folders:

- skins.htmlpost/
- skins.mail/
- skins.sms/

Each folder contains a number of notification skins divided into host and service notification filters.

- host.ACKNOWLEDGEMENT
- host.FLAPPINGSTART
- host.FLAPPINGSTOP
- host.PROBLEM
- host.RECOVERY

- service.ACKNOWLEDGEMENT
- service.FLAPPINGSTART
- service.FLAPPINGSTOP
- service.PROBLEM
- service.RECOVERY

As you can see there is one skin for the most common notification types.

The content of a notification skin

Let us take a look at what a skin looks like.

Example 1 The sms service.PROBLEM skin

```
#SERVICEDESC# on #HOSTNAME# is #SERVICESTATE#. #SERVICEOUTPUT#
```

This is a very simple skin. The reason for that is that you can not send too much data with a normal sms.

Example 2 The mail service.PROBLEM skin

```
From: op5Monitor
To: #CONTACTEMAIL#
Subject: [op5] #NOTIFICATIONTYPE#: '#SERVICEDESC#' on '#HOSTNAME#'
is #SERVICESTATE#
#extra_host_vars#
op5 Monitor
```

```

Service #NOTIFICATIONTYPE# detected #LASTSERVICESTATECHANGE#.
'#SERVICEDESC#' on host '#HOSTNAME#' has passed the #SERVICESTATE#
threshold.

#STATUS_URL#

Additional info;

#SERVICEOUTPUT#

Host:      #HOSTNAME#
Address:   #HOSTADDRESS#
Alias:     #HOSTALIAS#
Status:    #HOSTSTATE#
Comment:   #NOTIFICATIONCOMMENT#

Service:   #SERVICEDESC#
Status :   #SERVICESTATE#
Latency:   Check was #SERVICELATENCY# seconds behind schedule
Misc      : Check took #SERVICEEXECUTIONTIME# seconds to complete

Additional links (requires configuration);

Host actions: #HOSTACTIONURL#
Host notes:   #HOSTNOTESURL#Service actions: #SERVICEACTIONURL#
Service notes: #SERVICENOTESURL#

```

The mail notifications can contain a lot more data and there we add a lot more to the mail skin file.

In both [Example 1](#) on page 156 and [Example 2](#) on page 156 you find text like:

- #SERVICEDESC#
- #HOSTNAME#

That text is called **keywords**.

The keywords will be replaced with the value of a command line argument looking like this:

```
FOO=BAR
```

So a command line argument like the one above will generate a keyword with the name FOO having the value BAR.

Note: If a notification macro, or other value sent to a corresponding keyword, is missing in the notification command it will not stop the notification from being sent. It is only the replacement that will be missing.

Creating custom notification skins

Sometimes the default notification skins needs to be changed. This shall not be done in the default folders.

To create custom notification skins

- 1 Go to the notify folder:
`cd /opt/monitor/op5/notify`
- 2 Create the custom-skins folder:
`mkdir custom-skins`

- 3** Copy the skins.* folders to the custom-skins folder:
`cp -R skins.* custom-skins/`
- 4** Make the changes you like to do and the new skins will be used at directly after you have saved the changes.

Dial-up notification

Many of the modern mobile phones are only giving you one tiny signal when a sms arrives. If you are on duty during the night you might not wake up or if you are in a very noisy environment it might take some time for you to notice the arrived sms. There for we have included a dial up notification in op5 Monitor.

This is a very simple, but effective, notification that works like this:

Table 3 *Dial up notification workflow*

Step	Action
1	op5 Monitor is scheduling a notification.
2	The notification goes through all the filters.
3	The <code>notify_dial.pl</code> script is called with the following command line: <code>/opt/monitor/op5/notify/notif_dial.pl <mobilephonenumber></code>
4	<code>notify_dial.pl</code> is shutting down <code>smstd</code>
5	<code>notify_dial.pl</code> tries to call the <code><mobilephonenumber></code> If the line is busy or no one answer the call in 45 seconds <code>notify_dial.pl</code> will hang up and try again two more times before it quits.
6	The user answer the call and <code>notify_dial.pl</code> hangs up.
6	<code>notify_dial.pl</code> is starting up <code>smstd</code> again and the execution is over.

Adding a dial up notification command

This is done in two steps:

- add the command
- configure the contacts

To add a dial up notification command

- 1 Login to the op5 Monitor user interface and go to **Configure**.
- 2 Click **Commands**.
- 3 Add a new command with the following settings:
command_name `notify_by_dial`
command_line `$USER3$/notify/notify_dial.pl "$CONTACTPAGER$"`
- 4 Click **Apply**.
- 5 Click **Save**.

Configuring the contacts

To configure the contacts

- 1 Login to the op5 Monitor user interface and go to **Configure**.
- 2 Either open up an existing contact and create a new one.
- 3 On the contact set **Pager** to a phone number on the form like this (*without* the leading '+'-sign):
46705123123
- 4 Set **host_notification_commands** and **service_notification_commands** to:
notify_by_dial
- 5 Click **Apply**.
- 6 Click **Save**.

Note: Make sure the contact is a member of the contact_group is associated with the correct objects.

SNMP trap notifications

op5 Monitor is shipped with the possibility to send notifications as SNMP traps. To start use the SNMP notifications you need to

- add a few new commands
- configure the contacts

Adding SNMP notification commands

Here we need to add two commands one for host notifications and one for service notifications.

To add a SNMP notification command

- 1 Login to the op5 Monitor user interface and go to **Configure**.
- 2 Click **Commands**.
- 3 Add the following new commands with the following settings:

```
command_name host_notify_by_snmp
command_line $USER3$/notify/notify_by_snmp.pl -H
snmp.trap.host -C SNMPCOMMUNITY -t nHostNotify
"NOTIFICATIONTYPE=$NOTIFICATIONTYPE$"
"NOTIFICATIONNUMBER=$NOTIFICATIONNUMBER$"
"HOSTACKAUTHOR=$HOSTACKAUTHOR$"
"HOSTACKCOMMENT=$HOSTACKCOMMENT$" "HOSTNAME=$HOSTNAME$"
"HOSTSTATEID=$HOSTSTATEID$" "HOSTSTATETYPE=$HOSTSTATETYPE$"
"HOSTATTEMPT=$HOSTATTEMPT$"
"HOSTDURATIONSEC=$HOSTDURATIONSEC$"
"HOSTGROUPNAME=$HOSTGROUPNAME$"
"LASTHOSTCHECK=$LASTHOSTCHECK$"
"LASTHOSTSTATECHANGE=$LASTHOSTSTATECHANGE$"
"HOSTOUTPUT=$HOSTOUTPUT$"

command_name service_notify_by_snmp
command_line $USER3$/notify/notify_by_snmp.pl -H
snmp.trap.host -C SNMPCOMMUNITY -t nSvcNotify
"NOTIFICATIONTYPE=$NOTIFICATIONTYPE$"
"NOTIFICATIONNUMBER=$NOTIFICATIONNUMBER$"
"SERVICEACKAUTHOR=$SERVICEACKAUTHOR$"
"SERVICEACKCOMMENT=$SERVICEACKCOMMENT$" "HOSTNAME=$HOSTNAME$"
"HOSTSTATEID=$HOSTSTATEID$"
"SERVICEDESCRIPTION=$SERVICEDESCRIPTION$"
"SERVICESTATEID=$SERVICESTATEID$"
"SERVICEATTEMPT=$SERVICEATTEMPT$"
```

```
"SERVICEDURATIONSEC=$SERVICEDURATIONSEC$ "
"SERVICEGROUPNAME=$SERVICEGROUPNAME$ "
"LASTSERVICECHECK=$LASTSERVICECHECK$ "
"LASTSERVICESTATECHANGE=$LASTSERVICESTATECHANGE$ "
"SERVICEOUTPUT=$SERVICEOUTPUT$ "
```

Change the following to their correct value, in both commands:

```
snmp.trap.host
SNMPCOMMUNITY
```

- 4 Click **Apply**.
- 5 Click **Save**.

Configuring the contacts

To configure the contacts

- 1 Login to the op5 Monitor user interface and go to **Configure**.
- 2 Either open up an existing contact or create a new one.
- 3 Set **host_notification_commands** to:
`host_notify_by_snmp`
- 4 Set **service_notification_commands** to:
`service_notify_by_snmp`
- 5 Click **Apply**.
- 6 Click **Save**.

Note: Make sure the contact is a member of the contact_group is associated with the correct objects.

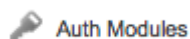
LDAP Integration

Introduction

The authentication system is handled by authentication drivers. Each driver handles authentication of the user, and resolution of the group memberships for the given user. The groups is then mapped to permissions by the authorization layer, which is described later.

An auth driver can either use a local storage of users (Driver Default), rely on apache authentication (Driver apache), or use an external system for managing users (Driver LDAP).

The authentication system is configured through the configuration, using the **Auth Modules** option under configuration.



The configuration for the authentication system is stored in the “auth” configuration file, located in `/etc/op5/auth.yml`

Default

For local users, the default driver can be used. This enables a local store of users at the op5 Monitor server. It is recommended that you always keep this driver configured with an admin account as a fallback if the system is primarily using LDAP.

When the Default driver is enabled, a configuration interface, named **Local Users** appears in op5 configuration.

In the local users page, each user has a real name, a password can be set, and group membership can be controlled. Groups needs to be created in advance. See [Group rights](#)

This driver stores the users in the `auth_users` configuration file, located in `/etc/op5/auth_users.yml`.

LDAP and Active Directory

For central user management, an LDAP server can be used, like Microsoft Active Directory or OpenLDAP. When used, op5 Monitor verifies the user with the LDAP server lookup the group membership of the users in the directory.

Before we start

This documentation assumes that you have:

- Administrator access to the domain
- Basic knowledge about LDAP structure

Prepare your domain

In op5 Monitor, permissions is handled by groups. Make sure you have one group available for each role in the system.

If the domain doesn't allow to bind anonymously to resolve group memberships or find users, a service account must be added. This account needs to have read access to resolve group membership and search for users in the system.

Connection parameters

Server

Address to the LDAP server, or servers. Can be a space separated list of addresses. Addresses are added for redundancy. Servers will be used in that order.

Port

TCP port to connect to. Leave blank for default. (389 for no encryption/start-tls, 636 for ssl)

Encryption

Which type of encryption to use for connection between op5 Monitor to the LDAP server. (none, start_tls or ssl). Make sure to have a valid ssl certificate for the LDAP server, and php recognizes it.

Bind DN

Distinguished name (or user principal name for active directory, which is username@domain) of the service account, created under "Prepare your domain" above, or empty to bind anonymously.

Bind secret

Password for the service user.

For security reasons, this can also be a path to a filename containing the password. To use this feature, enter `file:/path/to/secret/file`

It is also possible to keep the password in a separate config file, when multiple LDAP-connections is used. In this case, enter “config:configname”, which will use config file `/etc/op5/configname.yml`. The config file should then contain one line per driver: “driver name: secret”

Base DN

The distinguished name for the root of the directory to access. This is usually the DN for the domain, for example: `DC=example,DC=com`

User base DN

The base DN to search for users. This is an absolute DN, and not relative to Base DN. In almost all cases, use the same value as Base DN here.

User filter

A LDAP filter used to filter out user objects. Usually this is a filter for `objectClass`. For Active Directory “(objectClass=user)” should work.

Group Base DN

The base DN to search for groups. This is an absolute DN, and not relative to Base DN. In almost all cases, use the same value as Base DN here.

Group filter

A LDAP filter used to filter out group objects. Usually this is a filter for `objectClass`. For Active Directory “(objectClass=group)” should work.

Groupkey

The name of the attribute identifying the group. For Active Directory, “cn” should work.

Group Recursive

If groups can be nested, so that a group can be member of another group. This is possible in Active Directory, and should there be active.

With this unchecked, only members of that group directly will be treated as members of the group. If this is the case for systems which supports nested groups. This checkbox can be unchecked for performance reasons.

UPN Suffix

When binding with UPN (user principal name), this is the suffix to use after `@`, which is the domain name. For example, if the UPN of a user is “username@example.com”, the UPN suffix is “example.com”.

Userkey

The key to select the username of a user in the system. Older versions of Active Directory uses `sAMAccountName`. But in later versions, use `userPrincipalName`

Userkey is UPN

Check this if the userkey is a UPN. In that case, the domain part of the userkey will be ignored. Check this if you are using Active Directory, and userPrincipalName as userkey.

Userkey realname

The name of the attribute in the user object describing the real name of the user. For active Directory, and most other LDAP systems, “cn” should work. This is used to nicely display the username of the logged in user.

Userkey email

The name of the attribute in the user object containing the email address. For active directory, and many other systems, “mail” should work.

Memberkey

The name of the attribute in a group, which contains the reference to it’s members.

When using LDAP with posix extensions, this should be “memberUid”. When group is of class “groupOfUniqueNames”, this should be “uniqueMember”. For Active Directory, “member” should work.

Memberkey is DN

Check this box if Memberkey defines the entire DN of the member user or group, not only it’s name. For Active Directory, this is true. In a posix system, this is false.

Bind with UPN

If binding to the LDAP server should be done with the user principal name instead of the DN of the user.

For Active Directory, this is true. For all other systems, this is false.

When binding with UPN, the system constructs a UPN from the username and UPN suffix, and tries to bind with the constructed UPN and given password. If bind succeeds, it resolves the groups.

When binding with DN, the system tries to bind with “Bind DN” and “Bind Secret” to look in the directory for the user. If the user is found, it tries to rebind with the user DN and password given, and if that succeeds, the group membership is resolved.

Protocol version

The LDAP protocol version to use. Almost everyone will keep this at 3.

Example configuration for Active Directory

Auth Modules Configuration

common	Default x	op5 x	AD x	Add New Driver
Driver	LDAP			?
Server	ldap.example.com			?
Port				?
Encryption	none			?
Bind Dn	service_op5@example.com			?
Bind Secret	file:/etc/op5/ldap_secret			?
Base Dn	DC=example,DC=com			?
User Base Dn	DC=example,DC=com			?
User Filter	(objectClass=user)			?
Group Base Dn	DC=example,DC=com			?
Group Filter	(objectClass=group)			?
Groupkey	cn			?
Group Recursive	<input checked="" type="checkbox"/>			?
Upn Suffix	example.com			?
Userkey	userPrincipalName			?
Userkey Is Upn	<input checked="" type="checkbox"/>			?
Userkey Realname	cn			?
Userkey Email	mail			?
Memberkey	member			?
Memberkey Is Dn	<input checked="" type="checkbox"/>			?
Bind With Upn	<input checked="" type="checkbox"/>			?
Protocol Version	3			?

Server:ldap.example.com

Port:

Encryption:none

Bind DN:service_op5@example.com

Bind Secret:file:/etc/op5/ldap_secret

Base DN:DC=example,DC=com

User Base DN:DC=example,DC=com

User filter:(objectClass=user)

Group Base DN:DC=example,DC=com

Group filter:(objectClass=group)

Groupkey:cn
Group Recursive:yes
UPN Suffix:example.com
Userkey:userPrincipalName
Userkey is UPN:yes
Userkey realname:cn
Userkey email:mail
Memberkey:member
Memberkey is DN:yes
Bind with UPN:yes
Protocol version:3

Test your connection

To test if the system can bind using “Bind DN” and “Bind Secret”, go to **Assign Group Rights** page in op5 configuration. A column has appeared for the driver, and the corresponding group parameters is correctly set.

If a group is successfully resolved, the corresponding cell is turned green. If it is determined that the group doesn’t exist in the LDAP domain, the cell is red. In either way, a successful connection has been established.

If the connection failed, all the cells are gray.

Apache

A system can also rely on apache to authenticate the user. In this case, it is up to the user to protect the /monitor path with access in the apache web server, either by an .htaccess file or in the apache configuration.

The apache driver makes it possible to use apache modules for single sign-on authentication solutions, or other systems, like mysql or kerberos.

The driver gets the authenticated username from apache, and adds the group apache_auth_user to all users logged in.

For the apache login to work the **Auto Login** must be enabled.

Go to **Configure -> Auth Modules** and enable the **Auto Login** checkbox.

Auth Modules Configuration

common	Default	LDAP	Add New Module
Session Key	<input type="text" value="auth_user"/>	?	
Default Auth	<input type="text" value="LDAP"/>	?	
Enable Auto Login	<input checked="" type="checkbox"/>	?	
Apc Enabled	APC is not installed on this machine.		?
Apc Ttl	APC is not installed on this machine.		?



Note: The apache auth interface doesn't handle groups. Therefore it is impossible to get group membership out of the apache authentication, so each user is only member of two groups: user_<username> and apache_auth_user. When using central user management, the LDAP interface is therefore recommended where possible.

Select/Deselect all rights

To select or de-select all the rights in a group of rights, for example “Host” check the checkbox below the group of rights that you would like to add or remove.

The screenshot shows a table of rights groups. At the top, there are tabs for 'Default', 'LDAP', 'Nagios Auth', and 'Api'. Below the tabs are two buttons: 'Expand All' and 'Contract All', followed by two '+' icons. The table has a header row with a text input field containing 'Demo'. Below the header, there is a red square, a checked checkbox, and an unchecked checkbox.

The minus sign (-) in the checkbox means that the group of rights is partially selected. Some rights in the group are checked.

This screenshot is identical to the one above, but the checkbox in the 'Demo' row now contains a minus sign (-) instead of a checkmark, indicating that the group is partially selected.

Lookup user

You can find out which groups a user is a member of by entering a username in the Lookup user text box and clicking the Lookup button.

The screenshot shows a form with two labels: 'Lookup user' and 'Filter groups', each followed by a question mark icon. Below 'Lookup user' is a text input field containing the text 'monitor'. To the right of this field is a button labeled 'Lookup'. To the right of the 'Lookup' button is another empty text input field under the 'Filter groups' label.

monitor is also member of:

- Domain Users
- monitor
- sandboxusers

The groups that the user is a member will be highlighted and the authentication driver that they belong to will be indicated with an X.

	Default	op5ldap	Nagios Auth	Api	Host	Host Template	Service
Expand All							
Contract All							
admins	X						
default	X						
guest							
hsmederod							
limited_edit							
limited_view							

You will also get a list of additional groups the user is a member of underneath the Lookup user text box. An empty search string will hide the list and remove the highlights.

Filter groups

By adding a filter text in the Filter groups text box and clicking Filter groups you can set a filter on the visible groups.

Filter groups ?

 Filter Groups

Wildcard characters are neither supported nor needed. For example the filter strings “ad”, “a” and “min” will all match a group called “admins”.

Lookup user ?

Filter groups ?

 Lookup Filter Groups

	Default	op5ldap	Nagios Auth	Api	Host	Host Template	Service	Service Template	Hostgroup	Service
Expand All										
Contract All										
admins										
Submit										

An empty filter string will reset the filter.

Add, delete, rename groups

Renaming groups is done by typing a new name in the group name text box.

In the GUI you can create one new group each submit by filling the blank text box with the group name you want to create.

To add a LDAP or AD group type in the name of the group as it is named in your LDAP or AD.

Deletion of groups is done by removing the group name from the text box and leaving it blank when submitting your changes.

Configuration files used by authorization

The file `/etc/op5/auth_groups.yml` consists of all defined groups and their respective authorization points.

The GUI does not have to be used to edit authorization but we recommend that you use it to avoid syntax problems.

Authorization points

System Information

Gives the user access to the system/process information.

Configuration Information

Gives the user access to view and change configuration

System Commands

Gives the user access to issuing commands in the web gui.

Api Config

Gives the user access to the HTTP-API configuration interface.

Api Status

Gives the user access to the HTTP-API status interface.

Host Add Delete

Gives the user right to add and delete hosts.

Host View All

Gives the user right to view all hosts.

Host View Contact

Gives the user right to view hosts that he/she is contact for.

Host Edit All

Gives the user right to edit all existing hosts.

Host Edit Contact

Gives the user right to edit hosts that he/she is contact for.

Test This Host

Gives the user right to test the host that is being configured.

Service Add Delete

Gives the user right to add and delete services.

Service View All

Gives the user right to view all services.

Service View Contact

Gives the user right to view services that he/she is contact for.

Service Edit All

Gives the user right to edit all existing services.

Service Edit Contact

Gives the user right to edit services that he/she is contact for.

Test This Service

Gives the user right to test the service that is being configured.

Hostgroup Add Delete

Gives the user right to add and delete hostgroups.

Hostgroup View All

Gives the user right to view all hostgroups.

Hostgroup View Contact

Gives the user right to view hostgroups that he/she is contact for.

Hostgroup Edit All

Gives the user right to edit all existing hostgroups.

Hostgroup Edit Contact

Gives the user right to edit hostgroups that he/she is contact for.

Servicegroup Add Delete

Gives the user right to add and delete servicegroups.

Servicegroup View All

Gives the user right to view all servicegroups.

Servicegroup View Contact

Gives the user right to view servicegroups that he/she is contact for.

Servicegroup Edit All

Gives the user right to edit all servicegroups.

Servicegroup Edit Contact

Gives the user right to edit servicegroups that he/she is contact for.

Hostdependency Add Delete

Gives the user right to add and delete hostdependencies.

Hostdependency View All

Gives the user right to view hostdependencies.

Hostdependency Edit All

Gives the user right to edit hostdependencies.

Servicedependency Add Delete

Gives the user right to add and delete servicedependencies.

Servicedependency View All

Gives the user right to view servicedependencies.

Servicedependency Edit All

Gives the user right to edit servicedependencies.

Hostescalation Add Delete

Gives the user right to add and delete hostescalations.

Hostescalation View All

Gives the user right to view hostescalations.

Hostescalation Edit All

Gives the user tight to edit hostescalations.

Serviceescalation Add Delete

Gives the user right to add and delete serviceescalations.

Serviceescalation View All

Gives the user right to view serviceescalations.

Serviceescalation Edit All

Gives the user right to edit serviceescalations.

Contact Add Delete

Gives the user right to add and delete contacts.

Contact View All

Gives the user right to view contacts.

Contact Edit All

Gives the user right to edit contacts.

Contactgroup Add Delete

Gives the user right to add and delete contactgroups.

Contactgroup View All

Gives the user right to view contactgroups.

Contactgroup Edit All

Gives the user right to edit contactgroups.

Timeperiod Add Delete

Gives the user right to add and delete timeperiods.

Timeperiod View All

Gives the user right to view timeperiods.

Timeperiod Edit All

Gives the user right to edit timeperiods.

Command Add Delete

Gives the user right to add and delete commands.

Command View All

Gives the user right to view commands.

Command Edit All

Gives the user right to edit commands.

Test This Command

Gives the user right to execute commands.

Template

Gives the user right to view and change templates.

Wiki

Gives the user right to view, create and change docuwiki pages for objects he/she is authorized to see.

Wiki Admin

Gives the user right to access the docuwiki admin panel.

File

Gives the user right to change file in which an object is stored.

Access Rights

Gives the user right to edit access rights.

PNP

Gives the user right to access graphs.

Saved Filters Global

Gives the user right to create and delete global filters for listviews.

Export

Gives the user right to export it's own configuration.

Host Template View All

Gives the user right to view host templates.

Host Template Edit All

Gives the user right to edit host templates.

Host Template Add Delete

Gives the user right to add and delete host templates.

Service Template View All

Gives the user right to view service templates.

Service Template Edit All

Gives the user right to edit service templates.

Service Template Add Delete

Gives the user right to add and delete service templates.

Contact Template View All

Gives the user right to view contact templates.

Contact Template Edit All

Gives the user right to edit contact templates.

Contact Template Add Delete

Gives the user right to add and delete contact templates.

Configuration All

Gives the user right to export and import all configuration.

Nagvis Add Delete

Global permission to add and delete all nagvis maps.

Nagvis View

Global permission to view all nagvis maps.

Nagvis Edit

Global permission to edit all nagvis maps.

Nagvis Admin

Get full permission for nagvis, including global configuration

Backup and Restore

Configuration backup tool

The op5 Monitor GUI has got a built-in backup feature. This is not supposed to be a replacement to op5-backup.



The configuration backup is only backing up the op5 Monitor configuration, nothing else.

Backup/Restore actions

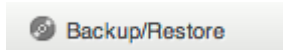
In the list of backups the second column is called **ACTIONS**. This is the functions you will find there, from the left to the right:

- View what files are included in the backup.
- Restor the backup
- Delete the backup.

Backing up the configuration

To backup your op5 Monitor configuration




- 1 Click Backup/Restore in the main menu.



- 2 Click **Save your current op5 Monitor configuration**.



- 3 Now your backup is created and can be restored at any time you like.

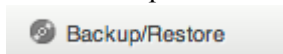
Backups	Actions
backup-2012-11-16_14.33	  

- 4 Click the backup archive name to download and save the backup archive somewhere else.

Restoring a configuration backup

To restor a op5 Monitor configuration backup

- 1 Click Backup/Restore in the main menu.



- 2 Click restor icon on the configuration backup you like to restore.



Now the backup has been restored.

op5-backup

About

The op5-backup script is a script that backs up the op5 installation. It does not backup the operating system.

Configuration

The configuration for op5-backup is located in:

`/etc/op5-backup/main.conf`

op5-backup support local or ftp backup. Local backup can be done to a mounted share.

Create a backup

Creating a full backup

A full backup will back up the following (if installed):

- op5-system
- op5-monitor
- op5-plugins
- Docuwiki
- Logserver
- Trapper

To run a full backup of your op5 server enter

```
op5-backup
```

The backup file will be stored in the location specified in the configuration file.

Creating a custom backup

It is possible to exclude or include different modules in a backup.

To get a list of the different modules type:

```
ls /etc/op5-backup/modules/legacy
```

To create a backup that excludes a specific module type:

```
op5-backup -- -<module1> -<module2>
```

To create a backup that includes only the specified modules type:

```
op5-backup -- +<module1> +<module2>
```

Creating a change arch backup

A change arch backup is used when i.e backing up a 32-bits system and restore it on a 64-bits system.

To create a change arch backup type:

```
op5-backup -m charch
```

It is also possible to combine this with the include/exclude modules option.

I.e we what to create a backup of a 32-bit system with the system configuration to restore that on a 64-bits system.

```
op5-backup -m charch -- -op5-system
```



A change arch backup will convert all graphs, in a large installation with a lot of history this can take up to a couple of hours.

Restoring a backup

To restore a full backup type:

```
op5-restore -b <path to backup file>
```



Only do a full restore when using a local terminal. Do not restore via SSH. The session will be lost if the network service is restarted.

Verify a backup

It is very good practice to verify the backups from time to time. Especially after a manual backup.

This is done using SSH or the console of the op5 server.

```
tar vft <backup-file>
```

Depending on what modules was used for the backup the list will vary. This is an example of a migration backup:

```
-rw-r--r-- root/root 1476847 2013-05-08 08:23 dokuwiki.tar.gz
-rw-r--r-- root/root 514982 2013-05-08 08:23 migrate.tar.gz
-rw-r--r-- root/root 296954 2013-05-08 08:23 nagios-
plugins.tar.gz
-rw-r--r-- root/root 1052 2013-05-08 08:23 op5-geomap.tar.gz
```

```
-rw-r--r-- root/root      26274 2013-05-08 08:23 op5-logserver-
3.tar.gz
-rw-r--r-- root/root 27206917 2013-05-08 08:24 op5-monitor.tar.gz
-rw-r--r-- root/root      142 2013-05-08 08:24 op5-notify.tar.gz
-rw-r--r-- root/root      409 2013-05-08 08:24 op5-synergy.tar.gz
-rw-r--r-- root/root 203002 2013-05-08 08:24 op5-system.tar.gz
-rw-r--r-- root/root      1917 2013-05-08 08:24 ssh.tar.gz
-rw-r--r-- root/root        4 2013-05-08 08:24 version
-rw-r--r-- root/root       16 2013-05-08 08:24 timestamp
-rw-r--r-- root/root        7 2013-05-08 08:24 architecture
-rw-r--r-- root/root        8 2013-05-08 08:24 mode
-rw-r--r-- root/root        7 2013-05-08 08:24 archive
drwxr-xr-x root/root        0 2013-05-08 08:24 modules/
-rw-r--r-- root/root      147 2013-05-08 08:23 modules/op5-geomap
-rw-r--r-- root/root    3284 2013-05-08 08:23 modules/op5-monitor
-rw-r--r-- root/root      136 2013-05-08 08:24 modules/op5-notify
-rw-r--r-- root/root      518 2013-05-08 08:24 modules/op5-system
-rw-r--r-- root/root      865 2013-05-08 08:23 modules/op5-
logserver-3
-rw-r--r-- root/root    5813 2013-05-08 08:23 modules/migrate
-rw-r--r-- root/root      116 2013-05-08 08:24 modules/ssh
-rw-r--r-- root/root     165 2013-05-08 08:24 modules/op5-synergy
-rw-r--r-- root/root     646 2013-05-08 08:23 modules/dokuwiki
-rw-r--r-- root/root     177 2013-05-08 08:23 modules/nagios-
plugins
```

Deleting a backup

Deleting a backup is really easy. It is just a matter of deleting the backup file. If the backup files are stored on the op5-server enter

```
rm <backup-file>
```

Or if the file is stored on a network share, you can browse the network share from any computer to delete the file.

Upgrade

Introduction

op5 Monitor is upgraded in the same way as the other op5 products. If you have an op5 Appliance system you can read about the upgrade procedure in the op5 Appliance system manual.

This chapter will only cover how to upgrade an op5 Monitor software version.

We will learn how to upgrade with the

- Linux command `yum`
- `tar.gz` files you may download from our support site.

If you are upgrading from one main version to an other (eg. from version 4 to 5) you need to use the `tar.gz` files found at our support site.

When upgrading over more than one main version (eg. from version 3 to 5) you shall follow the Upgrade guide found at our support site:

<http://www.op5.com/support/downloads/upgrade-guide>

Upgrading with yum

To upgrade with yum

- 1** Login to the op5 Monitor server via ssh as the root user.
- 2** Check what packages that is pending for upgrade by execute:
yum check-update
- 3** If you want to apply the upgraded packages execute:
yum update

Upgrading with tar.gz files

Before you start with the upgrade you need to make sure you have the login to the download sections at www.op5.com. Otherwise you will not be able to download the tar.gz files.

To upgrade with tar.gz files

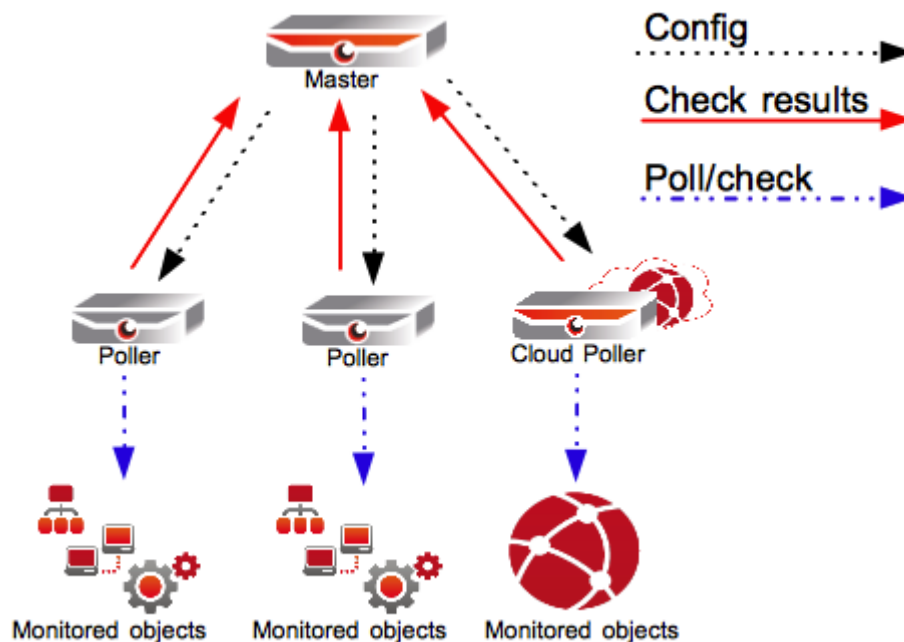
- 1** Download the tar.gz file from www.op5.com/get-op5-monitor/download/download-archive/.
Find the tar.gz file you need. You might need to open up the Archived files at the bottom of the page.
- 2** Upload the tar.gz file to the op5 Monitor server.
- 3** Login to the op5 Monitor server via ssh as the root user.
- 4** Untar the tar.gz file in the root/ folder.
- 5** Go to the folder that was extracted from the tar.gz file.
- 6** Now start the upgrade by executing the following script:
`./install.sh`

Scalable Monitoring

Distributed Monitoring

Introduction

The op5 Monitor backend can easily be configured to be used as a distributed monitoring solution. The distributed model looks like this.



In the distributed monitoring solution

- all configuration is done at the Master
- all new configuration is distributed to the pollers
- each poller is responsible for its own host group (Site).
- the Master has all the status information

Before we start

There are a few things you need to take care of before you can start setting up a distributed monitoring solution. You need to make sure

- you have at least two op5 Monitor servers of **the same architecture** up and running.
- op5 Monitor ≥ 5.2 is installed and running on both machines.
- opened up the following TCP ports for communication between the servers
 - 15551, op5 Monitor backend communication port
 - 22, ssh (for the configuration sync).

- both included servers are to be found in DNS.
- Make sure the host group, the one the poller will be responsible for, is added to the master configuration and that at least one host is added to that host group.

The configuration

Setting up the new distributed monitoring solution

This distributed configuration will have one master and one poller:

- master01
- poller01

The poller will be monitoring the host group gbg.

During the setup we will use the command:

```
mon
```

The mon command is used to make life a bit easier when it comes to setting up a load balanced solution. To get more detailed information about the command mon just execute like this:

```
mon --help
```

To setup a distributed monitoring solution with one poller

- 1 Log in to the master over ssh, as root.
- 2 Add the new poller to the configuration with the following command:

```
mon node add poller01 type=poller hostgroup=gbg
```
- 3 Create and add ssh keys to and from the second peer by
as root user:

```
mon sshkey push --all
mon sshkey fetch --all
```
- 4 Add master01 as master at poller01:

```
mon node ctrl --type=poller -- mon node add master01
type=master
```
- 5 Set up the configuration sync:

```
conf=/opt/monitor/etc/nagios.cfg
mon node ctrl -- sed -i /^cfg_file=/d $conf
```
- 6 To make sure you have an empty configuration on poller01:

```
mon node ctrl -- mon oconf hash
```

This will give you an hash looking like this (“d55d” -hash):

```
d55d3fa04bdd060bbe821b57c320fe807a096727
```

- 7 Now push the configuration to the poller:

```
mon oconf push
```
- 8 Restart and push the logs from master01 to poller01:

```
mon node ctrl --self -- mon restart; sleep 3; mon log push
```

Adding a new poller

In this instruction we will add a new poller to our distributed solution. Here we have the following hosts:

- master01
- poller01
- poller02 (This is the new one.)

To add a new poller

- 1 Log in to the master over ssh, as root.
- 2 Add the new poller to the configuration with the following command:
`mon node add poller02 type=poller hostgroup=gbg`
- 3 Create and add ssh keys for the root user:
`mon sshkey push poller02`
`mon sshkey fetch poller02`
- 4 Add master01 as master at poller02:
`mon node ctrl poller02 -- mon node add master01 type=master`
- 5 Set up the configuration sync:
`conf=/opt/monitor/etc/nagios.cfg`
`mon node ctrl poller02 -- sed -i /^cfg_file=/d $conf`
- 6 To make sure you have an empty configuration on poller01:
`mon node ctrl poller02 -- mon oconf hash`

This will give you an hash looking like this ("d 55" -hash):
`d55d3fa04bdd060bbe821b57c320fe807a096727`
- 7 Now push the configuration to the poller:
`mon oconf push`
- 8 Restart and push the logs from master01 from poller01:
`mon node ctrl --self -- mon restart; sleep 3; mon oconf push`

Adding a new host group to a poller

You might want to add an other host group for to a poller. You need to edit the merlin.conf file to do that. This is not doable with any command as it is today.

To add new host group to a poller

- 1 Open up and edit `/opt/monitor/op5/merlin/merlin.conf`.
- 2 Add a new host group in the hostgroup line like this:
`hostgroup = gbg,sth,citrix_servers`

Remember to not put any space between the hostgroup name and comma.

- 3 Restart monitor on the poller
`mon restart`
- 4 Send over the new configuration to the poller
`mon oconf push`

Removing a poller

In this instruction we will remove a poller called:

```
poller01
```

The poller will be removed from the master configuration and all distributed configuration on the poller will also be removed.

To remove a poller

- 1 Log in to the master over ssh, as root.
- 2 Deactivate and remove all distributed setup on the poller host.

```
mon node ctrl poller01 -- mon node remove master01
```
- 3 Restart monitor on the poller.

```
mon node ctrl poller02 -- mon restart
```
- 4 Remove the poller from the master configuration.

```
mon node remove poller01
```
- 5 Restart monitor on the master.

```
mon restart
```

Master takeover

If a poller goes down the default configuration is for the master to take over all the checks from the poller. For this to work all hosts monitored from the poller must also be monitorable from the master.

If the master server not should take over the checks from the poller this can be set in the merlin configuration file.

To stop the master from taking over, edit the file `/opt/monitor/op5/merlin/merlin.conf`

Add the following to the poller that you want the master not to take over.

```
takeover = no
```

Note that this is done per poller.

File synchronization

To synchronize files from the master server to the poller add a sync paragraph in the file `/opt/monitor/op5/merlin/merlin.conf`

In the example below we will synchronize the `htpasswd.users` file to the poller "poller01"

```
poller poller01 {  
    address = <ip>  
    port = <port>  
    contact_group = <contactgroup>
```

```
    sync {  
        /opt/monitor/etc/htpasswd.users = /opt/monitor/etc/  
        htpasswd.users  
    }  
}
```

Note that this is done per poller

Folder synchronization

To synchronize folders to pollers add a sync paragraph in the file `/opt/monitor/op5/merlin/merlin.conf`

In the example below we will synchronize the `/opt/plugins/custom` folder to the poller “poller01”

```
poller poller01 {  
    address = <ip>  
    port = <port>  
    contact_group = <contactgroup>  
  
    sync {  
        /opt/plugins/custom/  
    }  
}
```

Note that this is done per poller

Access right synchronization

To synchronize access rights the folder containing the access rights files must be added manually to the configuration. This will synchronize both local users and group right settings.

To do this, add the following sync command to `/opt/monitor/op5/merlin/merlin.conf`

```
sync {  
    /etc/op5/  
}
```

One way connections

If one peer is behind some kind of firewall or is on a NAT address it might not be possible for the master server to connect to the peer.

To tell the master not to connect to the poller and let the poller open the session we need to add a option to the file `/opt/monitor/op5/merlin/merlin.conf`.

Under the section for the poller that the master should not try to connect to add the following:

```
connect = no
```

Example

In the example below we have a master “master01” that can not connect to “poller01” but “poller01” is allowed to connect to “master01”.

```
poller poller01 {  
    address = <ip>  
    port = <port>  
    contact_group = <contactgroup>  
    connect = no  
}
```

It is also possible to set this option on the peer instead then the master will always initiate the session.

Recovery

After a poller has been unavailable for a master (i.e. of network outage) the report data will be synced from the poller to the master.

The report data on the poller will overwrite the data on the master system

More information

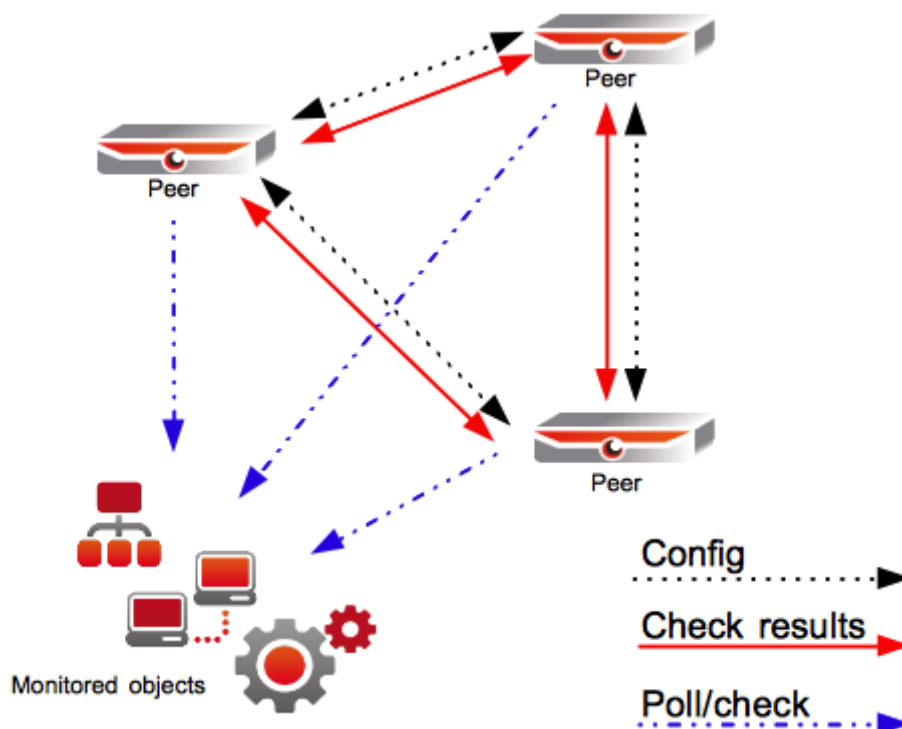
For more information and a more complex example please take a look at the howto in the git repository of the opensource project of Merlin:

<http://git.op5.org/git/?p=nagios/merlin.git;a=blob;f=HOWTO;hb=master>

Load balanced monitoring

Introduction

The op5 Monitor backend can easily be used as a load balanced monitoring solution. The load balanced model looks like this.



The load balanced solution

- have two or more peers sharing the same task (the hosts to monitor)
- allows configuration at any of the peers
- make sure that all new config is distributed to the peers
- uses the peers to divide the load automatically
- keep tracks of when one peer go down, the other(s) take over the job.

Before we start

There are a few things you need to take care of before you can start setting up an load balanced monitoring. You need to make sure

- you have at least two op5 Monitor servers of **the same architecture** up and running.
- op5 Monitor ≥ 5.2 is installed and running on both machines.

- opened up the following TCP ports for communication between the servers
 - 15551, op5 Monitor backend communication port
 - 22, ssh (for the configuration sync).
 - both included servers are to be found in DNS or the host file (/etc/hosts).

The configuration

Setting up the load balanced solution

This load balanced configuration will have two so called peers:

- peer01
- peer02

During the setup we will use the command:

```
mon
```

The mon command is used to make life a bit easier when it comes to setting up a load balanced solution. To get more detailed information about the command mon just execute like this:

```
mon --help
```

To setup a load balanced monitoring solution

- 1 Log in to one of the systems over ssh, as root.
- 2 Add the second peer to the configuration with the following command:
`mon node add peer02 type=peer`
- 3 Create and add ssh keys to and from the second peer by as root user:
`mon sshkey push --all`
`mon sshkey fetch --all`
- 4 Add peer01 as a peer at peer02
`mon node ctrl peer02 -- mon node add peer01 type=peer`
- 5 Make the first initial configuration sync
`mon oconf push`
- 6 Restart and push the logs from peer01 to peer02:
`mon node ctrl --self -- mon restart; sleep 3; mon oconf push`

Adding a new peer

In this instruction we will have the following hosts:

- peer01
- peer02
- peer03 (This is the new one.)

To add a new peer

- 1 Login to the peer01 as root user over ssh.
- 2 Add the new peer to the configuration on peer01

```
mon node add peer03 type=peer
```
- 3 Get all ssh keys in place

```
mon sshkey push --all  
mon sshkey fetch --all
```
- 4 Add the peers to one and each other

```
mon node ctrl peer02 -- mon node add peer03 type=peer  
mon node ctrl peer03 -- mon node add peer02 type=peer  
mon node ctrl peer03 -- mon node add peer01 type=peer
```
- 5 Manually push the op5 Monitor objects configuration to the new peer.

```
mon oconf push
```
- 6 Restart monitor on peer01 and send the configuration to all peers again.

```
mon node ctrl --self -- mon restart ; sleep 3 ; mon oconf push
```

Removing a peer

In this instruction we will remove a peer called:

peer02

The peer will be removed from all other peers configurations.

To remove a peer

- 1 Log in to peer01 as root over ssh.
- 2 Remove all peer configuration from peer02

```
mon node ctrl peer02 -- mon node remove peer01  
mon node ctrl peer02 -- mon node remove peer03
```
- 3 Restart monitor on peer02

```
mon node ctrl peer02 -- mon restart
```
- 4 Remove peer02 from the rest of the peers, in this case peer03

```
mon node ctrl --type=peer -- mon node remove peer02
```
- 5 Restart the rest of the peers, in this case only peer03

```
mon node ctrl --type=peer -- mon restart
```
- 6 Remove peer02 from the host you are working from.

```
mon node remove peer02
```
- 7 Restart monitor on the host you are working from.

```
mon node ctrl -- mon restart
```

File synchronization

To synchronize files between servers add a sync paragraph in the file /opt/monitor/op5/merlin/merlin.conf

In the example below we will synchronize the `htpasswd.users` file to the peer “peer02”

```
peer peer02 {  
    address = <ip>  
    port = <port>  
  
    sync {  
        /opt/monitor/etc/htpasswd.users /opt/monitor/etc/  
htpasswd.users  
    }  
}
```

Note that this is done per peer.

Folder synchronization

To synchronize folders to peers add a sync paragraph in the file `/opt/monitor/op5/merlin/merlin.conf`

In the example below we will synchronize the `/opt/plugins/custom` folder to the poller “peer02”

```
peer peer02 {  
    address = <ip>  
    port = <port>  
  
    sync {  
        /opt/plugins/custom/  
    }  
}
```

Note that this is done per peer

More information

For more information and a more complex example please take a look at the howto in the git repository of the opensource project of Merlin:

<http://git.op5.org/git/?p=nagios/merlin.git;a=blob;f=HOWTO;hb=master>

Merlin

About

Merlin is the backend engine for a load balanced and/or distributed setup of op5 Monitor.

Merlin, or Module for Effortless Redundancy and Load balancing In Nagios, allows the op5 Monitor processes to exchange information directly as an alternative to the standard nagios way using NSCA.

Merlin functions as backend for Ninja by adding support for storing the status information in a database, fault tolerance and load balancing. This means that Merlin now are responsible for providing status data and acts as a backend, for the Ninja GUI.

Merlin components

merlin-mod

merlin-mod is responsible for jacking into the NEBCALLBAC_* calls and send them to a socket.

If the socket is not available the events are written to a backlog and sent when the socket is available again.

merlind

The Merlin daemon listens to the socket that merlin-mod writes to and sends all events received either to a database of your choice (using libdbi) or to another merlin daemon.

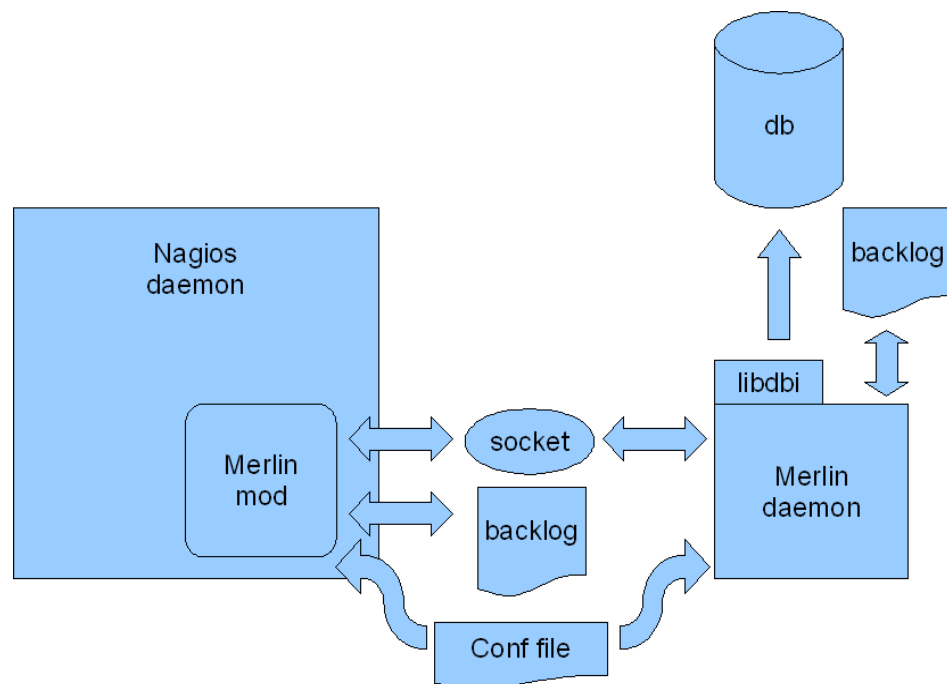
If the daemon is unsuccessful in this it writes to a backlog and sends the data later.

merlin database

This is a database that includes Nagios object status and status changes. It also contains comments, scheduled downtime etc.

Illustration

This picture illustrates the components described above



The mon command

About

The mon command is a very power command that comes with merlin. It is this command that is used to setup a distributed or a load balanced environment. This command can also be used to control the other op5 monitor servers.



The mon command is very powerful.

Handle with care!

It has the power to both create and destroy your whole op5 installation.

The commands

To use the mon command just type

```
# mon
```

The command should be used with one category and one sub-category. Only start, stop and restart categories can be used without any sub-category.

Start

```
# mon start
```

This will start the op5 monitor process on the node that you run the command from.

Stop

```
# mon stop
```

This will stop the op5 monitor process on the node you run the command from.

Restart

```
#mon restart
```

This will restart the op5 monitor process on the node you run the command from.

Ascii

Ninja

```
# mon ascii ninja
```

This will display the ninja logo in ascii art.

Merlin

```
# mon ascii merlin
```

This will display the merlin logo in ascii art.

Check

Spool

```
# mon check spool [--maxage=<seconds>] [--warning=X] [--critical=X]  
<path> [--delete]
```

Checks a certain spool directory for files (and files only) that are older than 'maxage'. It's intended to prevent buildup of checkresult files and unprocessed performance-data files in the various spool directories used by op5 Monitor.

--delete	Causes too old files to be removed.
--maxage	Is given in seconds and defaults to 300 (5 minutes).
<path>	May be 'perfdata' or 'checks', in which case directory names will be taken from op5 defaults
--warning and --critical	Have no effect if '--delete' is given and will otherwise specify threshold values.



Only one directory at a time may be checked.

Cores

```
# mon check cores --warning=X --critical=X [--dir=]
```

Checks for memory dumps resulting from segmentation violation from core parts of op5 Monitor. Detected core-files are moved to /tmp/mon-cores in order to keep working directories clean.

--warning	Default is 0
--critical	Default is 1 (any corefile results in a critical alert)
--dir	Lets you specify more paths to search for corefiles. This option can be given multiple times.
--delete	Deletes corefiles not coming from 'merlind' or 'monitor'.

Distribution

```
#mon check distribution [--no-perfdata]
```

Checks to make sure distribution works ok.



Note that it's not expected to work properly the first couple of minutes after a new machine has been brought online or taken offline

Exectime

```
# mon check exectime [host|service] --warning=<min,max,avg> --critical=<min,max,avg>
```

Checks execution time of active checks.

[host service]	Select host or service execution time.
--warning	Set the warning threshold for min,max and average execution time, in seconds
--critical	Set the critical threshold for min,max and average execution time, in seconds

Latency

```
# mon check latency [host|service] --warning=<min,max,avg> --critical=<min,max,avg>
```

Checks latency time of active checks.

[host service]	Select host or service latency time.
--warning	Set the warning threshold for min,max and average execution time, in seconds
--critical	Set the critical threshold for min,max and average execution time, in seconds

Orphans

```
#mon check orphans
```

Checks for checks that haven't been run in too long a time.

db

cahash

Calculates a hash of all entries in the contact_access table. This is really only useful for debugging purposes. The check does not block execution of other scripts or checks.

Fixindexes

Fixes indexes on merlin tables containing historical data.



Don't run this tool unless you're asked to by op5 support staff or told to do so by a message during an rpm or yum upgrade.

ecmd

Search

```
# mon ecmd search <regex>
```

Prints 'templates' for all available commands matching <regex>.

The search is case insensitive.

Submit

```
# mon ecmd submit [options] command <parameters>
```

Submits a command to the monitoring engine using the supplied values.

Available options:

```
--pipe-path=</path/to/nagios.cmd>
```

Example:

An example command to add a new service comment for the service PING on the host foo would look something like this:

```
# mon ecmd submit add_svc_comment service='foo;PING' persistent=1
author='John Doe' comment='the comment'
```

Note how services are written. You can also use positional arguments, in which case the arguments have to be in the correct order for the command's syntactic template. The above example would then look thus:

```
# mon ecmd submit add_svc_comment 'foo;PING' 1 'John Doe' 'the
comment'
```

Log

Fetch

```
# mon log fetch [--incremental=<timestamp>]
```

Fetches logfiles from remote nodes and stashes them in a local path, making them available for the 'sortmerge' command.

Import

```
# mon log import [--fetch]
```

This commands run the external log import helper.

If `--fetch` is specified, logs are first fetched from remote systems and sorted using the merge sort algorithm provided by the `sortmerge` command.

Purge

```
#mon log purge
```

Remove log files that are no longer in use.



Currently only deletes stale RRD files.

Push

```
#mon log push
```

(documentation missing)

Show

```
#mon log show
```

Runs the showlog helper program. Arguments passed to this command will get sent to the showlog helper.

For further help about the show category use:

```
#mon log show --help
```

Sortmerge

```
#mon log sortmerge [--since=<timestamp>]
```

Runs a mergesort algorithm on logfiles from multiple systems to create a single unified logfile suitable for importing into the reports database.

Node

Add

```
# mon node add <name> type=[peer|poller|master] [var1=value]
[varN=value]
```

Adds a node with the designated type and variables.

Ctrl

```
#mon node ctrl <name1> <name2> [--self] [--all|--
type=<peer|poller|master>] -- <command>
```

Execute `<command>` on the remote node(s) named. `--all` means run it on all configured nodes, as does making the first argument `'--'`.

`--type=<types>` means to run the command on all configured nodes of the given type(s).

The first not understood argument marks the start of the command, but always using double dashes is recommended. Use single-quotes to execute commands with shell variables, output redirection or scriptlets, like so:

```
# mon node ctrl -- '(for x in 1 2 3; do echo $x; done) > /tmp/foo'
# mon node ctrl -- cat /tmp/foo
```

List

```
#mon node list [--type=poller,peer,master]
```

Lists all nodes of the (optionally) specified type

Remove

```
#mon node remove <name1> [name2] [nameN]
```

Removes one or more nodes from the merlin configuration.

Show

```
#mon node show [--type=poller,peer,master]
```

Display all variables for all nodes, or for one node in a fashion suitable for being used as eval \$(mon node show nodename) from shell scripts and scriptlets.

Status

```
#mon node status
```

Show status of all nodes configured in the running Merlin daemon.

Red text points to problem areas, such as high latency or the node being inactive, not handling any checks, or not sending regular enough program_status updates.

Tree

```
# mon node tree
```

This command draws a tree of the masters and pollers. **Example:**

```
+-----+ +-----+
| ipc |---| athena |
+-----+ +-----+
|
|
| HOSTGROUP: op5-gbg,localnet +-----+
= -----| dione |
      \                               +-----+
      \ +-----+
      --| styx |
      +-----+
```

A setup of 2 peered masters and 2 peered pollers. The pollers are monitoring two hostgroups (op5-gbg and localnet).

oconf

Changed

```
#mon oconf changed
```

Print last modification time of all object configuration files

Fetch

```
#mon oconf fetch
```

Fetch the configuration from a Master, this is executed on a poller system. Useful when the poller can talk to the master but not vice versa.

Files

```
#mon oconf files
```

Print the configuration files in alphabetical order

Hash

```
#mon oconf hash
```

Print sha1 hash of running configuration

HGlist

```
#mon oconf hglist
```

Print a sorted list of all configured hostgroups

Nodesplit

```
#mon oconf nodesplit
```

Same as 'split', but use merlin's config to split config into configuration files suitable for poller consumption

Pull

```
#mon oconf pull
```

(documentation missing)

Push

```
#mon oconf push
```

Splits configuration based on merlin's peer and poller configuration and send object configuration to all peers and pollers, restarting those that receive a configuration update. ssh keys need to be set up for this to be usable without admin supervision.

This command uses 'nodesplit' as its backend.

Spit

```
#mon oconf split <outfile:hostgroup1,hostgroup2,hostgroupN>
```

Write config for hostgroup1,hostgroup2 and hostgroupN into outfile.

SSHKey

Fetch

```
#mon sshkey fetch
```

Fetches all the SSH keys from peers and pollers.

Push

```
#mon sshkey push
```

Pushes the local SSH keys to all peers and pollers.

Sysconf

Ramdisk

```
#mon sysconf ramdisk
```

To enable ramdisk:

```
#mon sysconf ramdisk
```

A ramdisk can be enabled for storing spools for performance data and checkresults.

By storing these spools on a ramdisk we can lower the disk I/O significant

Test



All commands in this category can potentially overwrite configuration, enable or disable monitoring and generate notifications.

Do **NOT** use these commands in a production environment.

Dist

```
#mon test dist [options]
```

Tests various aspects of event forwarding with any number of hosts, services, peers and pollers, generating config and creating databases for the various instances and checking event distribution among them.

For complete list of option, run

```
#mon test --help
```

Pasv

```
# mon test pasv [options]
```

Submits passive checkresults to the nagios.cmd pipe and verifies that the data gets written to database correctly and in a timely manner.

For complete list of option, run

```
#mon test --help
```



This command will disable active checks on your system and have other side-effects as well.

VRRP

About

VRRP can be used in this setup to have one DNS-name and one IP address that is primary linked to one of the master servers and if the primary master for some reason is unavailable VRRP will automatically detect this and send you to the secondary master.

Setup

To enable VRRP on you master servers follow the steps below.

In this example we have two masters that we want to use VRRP with.

The VRRP IP will be 192.168.1.3 and we will bind that IP to the network interface eth0.

The IP and interface will have to change to match your network configuration.



If you already use VRRP in your network, make sure that you use the correct `virtual_router_id`.

Edit /etc/keepalived/keepalived.conf

On the “primary” master

```
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 200
    advert_int 1
    virtual_ipaddress {
        192.168.1.3 dev eth0
    }
}
```

On the “secondary” master

```
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 51
    priority 100
```

```
advert_int 1
virtual_ipaddress {
    192.168.1.3 dev eth0
}
}
```

Activate VRRP

To activate vrrp run the following command:

```
# chkconfig keepalived on
```


op5 Monitor API and CLI

Introduction

op5 Monitor comes with a few API's that can be used to The following API's can be used.

- Ninja API, GUI API
- Nacoma API, Configure API
- op5 Monitor Configuration CLI

REST API

About

The REST-API lets you configure op5 Monitor by issuing regular HTTP requests.

Basically, you 'visit' an URI, which triggers op5 Monitor to do something, and you get a response telling you what happened.

For more information about the REST API go to <https://your-op5-monitor/api/help/>

Example

In this example we will create a new host called **my_server** with one ping service. The IP for **my_server** is **192.168.0.20**

In this example the op5 server is called **op5-server**, the username is **joe** and joe's password is **joespassword**.

By visiting the `<monitor-installation>/api/help/config/host`, you get information on how to create a host. This is what needs to be done in PHP:

```
<?php
$data = json_encode(array(
    'address' => '192.168.0.20',
    'alias' => 'My Server',
    'host_name' => 'my_server'
));
$a_handle = curl_init('<monitor-installation>/api/config/host');
curl_setopt($a_handle, CURLOPT_USERPWD, 'joe:joespassword');
curl_setopt($a_handle, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($a_handle, CURLOPT_POSTFIELDS, $data);
curl_setopt($a_handle, CURLOPT_HTTPHEADER, array('Content-Type:
application/json'));
curl_setopt($a_handle, CURLOPT_SSL_VERIFYPEER, false);
$host = curl_exec($a_handle);

$data = json_encode(array(
    'check_command' => 'check_ping',
    'service_description' => 'ping',
    'host_name' => 'my_server'
));
$a_handle = curl_init('op5-server/api/config/service');
curl_setopt($a_handle, CURLOPT_USERPWD, 'joe:joespassword');
```

```
curl_setopt($a_handle, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($a_handle, CURLOPT_POSTFIELDS, $data);
curl_setopt($a_handle, CURLOPT_HTTPHEADER, array('Content-Type:
application/json'));
curl_setopt($a_handle, CURLOPT_SSL_VERIFYPEER, false);
$service = curl_exec($a_handle);
?>
```

Before the changes are applied, you need to confirm them and then save them so that they become part of your configuration. This can be done in two ways, either by Saving changes in the op5 Monitor GUI, or by adding an additional call via the REST API:

```
<?php
$a_handle = curl_init('op5-server/api/config/change');
curl_setopt($a_handle, CURLOPT_USERPWD, 'joe:joespassword');
curl_setopt($a_handle, CURLOPT_CUSTOMREQUEST, 'POST');
curl_setopt($a_handle, CURLOPT_SSL_VERIFYPEER, false);
$save = curl_exec($a_handle);
?>
```

Now, visiting `op5-server/api/config/host/my_server` in a browser should show you the live configuration.

REST and LDAP

When you have more than one auth module, for example "Local" and "LDAP", you need to specify which to authenticate against. This is done with the dollar character ('\$').

Thus, this regular call:

```
curl -u user:password https://monitor/api/status/host [^]
```

becomes

```
curl -u 'user$LDAP:password' https://monitor/api/status/host [^]
```

or

```
curl -u 'user$Local:password' https://monitor/api/status/host [^]
```

Notice how the dollar sign ('\$') needs quoting depending on the environment (in bash, it will always need to be quoted).

The first way of calling the API:

```
curl -u user:password https://monitor/api/status/host [^]
```

will still work, provided that you want to authenticate against the default driver.

The default driver can always be specified from within the GUI: **Configure -> Auth Modules**.

GUI API

The GUI API is used to get the information that is used by op5 Monitor GUI. It can give you information about all objects used by the op5 monitor.

Widgets are one place where the GUI API will come handy.

There are only a few brief documents about that API today. It is included in the product.

Let us say that your monitor server is called op5-monitor you can reach the documentation on the following location:

<https://op5-monitor/monitor/Documentation/html/index.html>

It is generated by doxygen and contains information like

- namespaces
- structures (classes and methods)
- files

Configure API

The configure API is used to manipulate the object configuration used by op5 Monitor. It works against the configure database the same way as the op5 Monitor Configuration tool does.

You may use it to build integrations between op5 Monitor and other third party software.

Let us say that your monitor server is called op5-monitor you can reach the documentation on the following location:

<https://op5-monitor/monitor/op5/nacoma/Documentation/html/index.html>

It is generated by doxygen and contains information like classes and methods used in the op5 Monitor configuration tool.

op5 Monitor Configuration CLI

This is a tool used to edit the op5 Monitor object configuration.

You may use this one to add and remove

- hosts
- services
- contacts
- timeperiods.

You may also

- list objects
- save configuration
- undo configuration (force import of the config files to the configure database).

Executing the op5 Monitor CLI

To execute the op5 Monitor CLI

- 1 Logon to the op5 Monitor server as root
- 2 Execute the following command:
`php /opt/monitor/op5/nacoma/api/monitor.php`

The above example will give you a description about how to use the op5 Monitor Configuration CLI

Wiki

Introduction

In op5 Monitor there is a wiki included which can be used for documenting hosts and services.

Wiki pages can be created for both hosts and services.

The wiki has a built in version revisioning, this can be used to track changes in wiki pages and restore an older version. It is also possible to view changes between versions.

The wiki can be used for documenting hardware information, serial numbers and other information regarding a host or service. It can also be used to document workflows and how to act when there is a problem with a host or service.

The official docuwiki manual can be found here: <http://www.dokuwiki.org/manual>

Managing wiki pages

Create a wiki page

To create a wiki page for a host or service

- 1 Go to **Configuration**
- 2 Go to the host or service you want to create a page for.
- 3 Click on **Advanced**
- 4 Scroll down to 'notes_url' and click **Use wiki**. This will add a notes url to a wiki page.



- 5 Click **Submit** and save your configuration.
- 6 Go to the host in op5 Monitor and click on **Extra notes**

FAN status

On host	switch1-gbg.int.op5.se (switch1-gbg.int.op5.se)
Address	192.168.1.18
Member of	No servicegroups
Notifies to	it-group



- 7 Click on **Create page**.
- 8 Edit the information and click on **Save**

Deleting a wiki page

If you edit a page and remove all its content then DokuWiki will delete the page, and the associated page name.

For more information about the docuwiki

<http://www.dokuwiki.org/manual>

Third party configuration import

Introduction

Op5 Monitor has the capability to import the configuration from an nagios installation.

To do follow this manual basic knowledge in linux and nagios is necessary.

Pre-requirements

A running nagios 3.x installation and op5 Monitor.

Limitations

There are some of limitations of the import script.

- The import-script does not work with a nagios 1 or 2 installation.
- Host and service history can not be imported, but can be copied manually.
- Graph history can not be imported.

Import configuration

To import a nagios 3 configuration we need to prepare the nagios configuration files first, after that we can use the import script to import the files into op5 Monitor.

Preparing nagios configuration

Log in to the nagios server via ssh or locally.

Create a new file called templates.cfg in which you manually add both your host-templates and your service-templates. These are usually located in hosts.cfg and services.cfg.

Create a nagios pre-cache file by stopping nagios and start it with the -p option. this is done from you nagios binary directory, usually “/usr/local/nagios/bin”.

```
# service nagios stop
# ./nagios -pv <path to your nagios.cfg>
```

This will create a file called objects.precache in your “var” directory under your nagios installation.

Import nagios configuration

Make sure op5 monitor is stopped

```
# mon stop
```

Copy the files to the correct directory on your op5 Monitor server.

File	To folder
objects.precache	/opt/monitor
templates.cfg	/opt/monitor
nagios.log	/opt/monitor/var/
log archive	/opt/monitor/var/archives

Run the import script

```
# php /opt/monitor/op5/nacoma/import-reduce.php --cfg-file=/opt/monitor/templates.cfg --object-cache=/opt/monitor/objects.precache
```

Do a config-test on the imported configuration

```
# service monitor configtest
```

If you have any errors these needs to be resolved before we can continue with starting the op5 monitor service.

When there are no issues left start the monitor service

```
# mon start
```