


 **Centreon**

| Poller States | Hosts | Up | Down | Unreachable | Pending | Services | Ok | Warning | Critical | Unknown | Pending |
|---|-------|-----|------|-------------|---------|----------|------|---------|----------|---------|---------|
|    | 208 | 207 | 1 | 0 | 0 | 3652 | 3343 | 11/24 | 72/206 | 69/79 | 0 |

[Documentation](#) - You are [service.desk](#) [Logout](#)




Monitoring Views Reporting

Services | Hosts | Event Logs

Monitoring Services Details

2013/05/11 23:20



>> By Status

-  Unhandled Problems
-  Service Problems
-  All Services



>> By Host

-  Details
-  Summary

>> By Host Group

-  Details
-  Summary




>> By Service Group

-  Details
 - Problems
 - Acknowledged
 - Not Acknowledged
-  Summary

>> Meta Services

-  Meta Services

>> Nagios

-  Scheduling Queue
-  Downtime
-  Comments

Nagios®

Nagios Plugin API



Up To: Contents

See Also: Plugin Overview, Developing Plugins For Use With Embedded Perl, Performance Data

Other Resources

If you're looking at writing your own plugins for Nagios, please make sure to visit these other resources:

- The official Nagios plugin project website
- The official Nagios plugin development guidelines

Plugin Overview

Scripts and executables must do two things (at a minimum) in order to function as Nagios plugins:

- Exit with one of several possible return values
- Return at least one line of text output to STDOUT

The inner workings of your plugin are unimportant to Nagios. Your plugin could check the status of a TCP port, run a database query, check disk free space, or do whatever else it needs to check something. The details will depend on what needs to be checked - that's up to you.

Return Code

Nagios determines the status of a host or service by evaluating the return code from plugins. The following tables shows a list of valid return codes, along with their corresponding service or host states.

| Plugin Return Code | Service State | Host State |
|--------------------|---------------|-------------------------|
| 0 | OK | UP |
| 1 | WARNING | UP or DOWN/UNREACHABLE* |
| 2 | CRITICAL | DOWN/UNREACHABLE |
| 3 | UNKNOWN | DOWN/UNREACHABLE |

Note: If the use_aggressive_host_checking option is enabled, return codes of 1 will result in a host state of DOWN or UNREACHABLE. Otherwise return codes of 1 will result in a host state of UP. The process by which Nagios determines whether or not a host is DOWN or UNREACHABLE is discussed here.

Plugin Output Spec

At a minimum, plugins should return at least one of text output. Beginning with Nagios 3, plugins can optionally return multiple lines of output. Plugins may also return optional performance data that can be processed by external applications. The basic format for plugin output is shown below.

```
TEXT OUTPUT | OPTIONAL PERFDATA
LONG TEXT LINE 1
LONG TEXT LINE 2
...
LONG TEXT LINE N | PERFDATA LINE 2
PERFDATA LINE 3
...
PERFDATA LINE N
```

The performance data (shown in orange) is optional. If a plugin returns performance data in its output, it must separate the performance data from the other text output using a pipe (|) symbol. Additional lines of long text output (shown in blue) are also optional.

Plugin Output Examples

Let's see some examples of possible plugin output...

Case 1: One line of output (text only)

Assume we have a plugin that returns one line of output that looks like this:

```
DISK OK - free space: / 3326 MB (56%);
```

If this plugin was used to perform a service check, the entire line of output will be stored in the \$SERVICEOUTPUT\$

macro.

Case 2: One line of output (text and perfdata)

A plugin can return optional performance data for use by external applications. To do this, the performance data must be separated from the text output with a pipe (|) symbol like such:

DISK OK - free space: / 3326 MB (56%); | /=2643MB;5948;5958;0;5968

If this plugin was used to perform a service check, the **red** portion of output (left of the pipe separator) will be stored in the \$SERVICEOUTPUT\$ macro and the **orange** portion of output (right of the pipe separator) will be stored in the \$SERVICEPERFDATA\$ macro.

Case 3: Multiple lines of output (text and perfdata)

A plugin optionally return multiple lines of both text output and perfdata, like such:

DISK OK - free space: / 3326 MB (56%); | /=2643MB;5948;5958;0;5968
/ 15272 MB (77%);
/boot 68 MB (69%);
/home 69357 MB (27%);
/var/log 819 MB (84%); | /boot=68MB;88;93;0;98
/home=69357MB;253404;253409;0;253414
/var/log=818MB;970;975;0;980

If this plugin was used to perform a service check, the **red** portion of first line of output (left of the pipe separator) will be stored in the \$SERVICEOUTPUT\$ macro. The **orange** portions of the first and subsequent lines are concatenated (with spaces) are stored in the \$SERVICEPERFDATA\$ macro. The **blue** portions of the 2nd - 5th lines of output will be concatenated (with escaped newlines) and stored in \$LONGSERVICEOUTPUT\$ the macro.

The final contents of each macro are listed below:

| Macro | Value |
|-----------------------|--|
| \$SERVICEOUTPUT\$ | DISK OK - free space: / 3326 MB (56%); |
| \$SERVICEPERFDATA\$ | /=2643MB;5948;5958;0;5968 /boot=68MB;88;93;0;98 /home=69357MB;253404;253409;0;253414 /var/log=818MB;970;975;0;980 |
| \$LONGSERVICEOUTPUT\$ | / 15272 MB (77%);\n/boot 68 MB (69%);\n/var/log 819 MB (84%); |

With regards to multiple lines of output, you have the following options for returning performance data:

- You can choose to return no performance data whatsoever
- You can return performance data on the first line only
- You can return performance data only in subsequent lines (after the first)
- You can return performance data in both the first line and subsequent lines (as shown above)

Plugin Output Length Restrictions

Nagios will only read the first 4 KB of data that a plugin returns. This is done in order to prevent runaway plugins from dumping megas or gigs of data back to Nagios. This 4 KB output limit is fairly easy to change if you need. Simply edit the value of the MAX_PLUGIN_OUTPUT_LENGTH definition in the *include/nagios.h.in* file of the source code distribution and recompile Nagios. There's nothing else you need to change!

Examples

If you're looking for some example plugins to study, I would recommend that you download the official Nagios plugins and look through the code for various C, Perl, and shell script plugins. Information on obtaining the official Nagios plugins can be found here.

Perl Plugins

Nagios features an optional embedded Perl interpreter which can speed up the execution of Perl plugins. More information on developing Perl plugins for use with the embedded Perl interpreter can be found here.