




 **Centreon**

Poller States	Hosts	Up	Down	Unreachable	Pending	Services	Ok	Warning	Critical	Unknown	Pending
  	208	207	1	0	0	3652	3342	11/24	73/207	69/79	0

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




Monitoring Views Reporting

Services | Hosts | Event Logs

Monitoring Services Details

2013/05/11 23:19

&gt;&gt; By Status

-  Unhandled Problems
-  Service Problems
-  All Services



&gt;&gt; By Host

-  Details
-  Summary

&gt;&gt; By Host Group

-  Details
-  Summary




&gt;&gt; By Service Group

-  Details
  - Problems
  - Acknowledged
  - Not Acknowledged
-  Summary

&gt;&gt; Meta Services

-  Meta Services

&gt;&gt; Nagios

-  Scheduling Queue
-  Downtime
-  Comments

# Nagios®

## Notifications



Up To: Contents

See Also: Escalations, Timeperiods, On-Call Rotations

### Introduction

I've had a lot of questions as to exactly how notifications work. This will attempt to explain exactly when and how host and service notifications are sent out, as well as who receives them.



Notification escalations are explained here.

### When Do Notifications Occur?

The decision to send out notifications is made in the service check and host check logic. Host and service notifications occur in the following instances...

- When a hard state change occurs. More information on state types and hard state changes can be found here.
- When a host or service remains in a hard non-OK state and the time specified by the `<notification_interval>` option in the host or service definition has passed since the last notification was sent out (for that specified host or service).

### Who Gets Notified?

Each host and service definition has a `<contact_groups>` option that specifies what contact groups receive notifications for that particular host or service. Contact groups can contain one or more individual contacts.

When Nagios sends out a host or service notification, it will notify each contact that is a member of any contact groups specified in the `<contactgroups>` option of the service definition. Nagios realizes that a contact may be a member of more than one contact group, so it removes duplicate contact notifications before it does anything.

### What Filters Must Be Passed In Order For Notifications To Be Sent?

Just because there is a need to send out a host or service notification doesn't mean that any contacts are going to get notified. There are several filters that potential notifications must pass before they are deemed worthy enough to be sent out. Even then, specific contacts may not be notified if their notification filters do not allow for the notification to be sent to them. Let's go into the filters that have to be passed in more detail...

#### Program-Wide Filter:

The first filter that notifications must pass is a test of whether or not notifications are enabled on a program-wide basis. This is initially determined by the `enable_notifications` directive in the main config file, but may be changed during runtime from the web interface. If notifications are disabled on a program-wide basis, no host or service notifications can be sent out - period. If they are enabled on a program-wide basis, there are still other tests that must be passed...

#### Service and Host Filters:

The first filter for host or service notifications is a check to see if the host or service is in a period of scheduled downtime. If it is in a scheduled downtime, **no one gets notified**. If it isn't in a period of downtime, it gets passed on to the next filter. As a side note, notifications for services are suppressed if the host they're associated with is in a period of scheduled downtime.

The second filter for host or service notification is a check to see if the host or service is flapping (if you enabled flap detection). If the service or host is currently flapping, **no one gets notified**. Otherwise it gets passed to the next filter.

The third host or service filter that must be passed is the host- or service-specific notification options. Each service definition contains options that determine whether or not notifications can be sent out for warning states, critical states, and recoveries. Similarly, each host definition contains options that determine whether or not notifications can be sent out when the host goes down, becomes unreachable, or recovers. If the host or service notification does not pass these options, **no one gets notified**. If it does pass these options, the notification gets passed to the next filter... Note: Notifications about host or service recoveries are only sent out if a notification was sent out for the original problem. It doesn't make sense to get a recovery notification for something you never knew was a problem.

The fourth host or service filter that must be passed is the time period test. Each host and service definition has a

`<notification_period>` option that specifies which time period contains valid notification times for the host or service. If the time that the notification is being made does not fall within a valid time range in the specified time period, **no one gets contacted**. If it falls within a valid time range, the notification gets passed to the next filter... Note: If the time period filter is not passed, Nagios will reschedule the next notification for the host or service (if its in a non-OK state) for the next valid time present in the time period. This helps ensure that contacts are notified of problems as soon as possible when the next valid time in time period arrives.

The last set of host or service filters is conditional upon two things: (1) a notification was already sent out about a problem with the host or service at some point in the past and (2) the host or service has remained in the same non-OK state that it was when the last notification went out. If these two criteria are met, then Nagios will check and make sure the time that has passed since the last notification went out either meets or exceeds the value specified by the `<notification_interval>` option in the host or service definition. If not enough time has passed since the last notification, **no one gets contacted**. If either enough time has passed since the last notification or the two criteria for this filter were not met, the notification will be sent out! Whether or not it actually is sent to individual contacts is up to another set of filters...

### Contact Filters:

At this point the notification has passed the program mode filter and all host or service filters and Nagios starts to notify all the people it should. Does this mean that each contact is going to receive the notification? No! Each contact has their own set of filters that the notification must pass before they receive it. Note: Contact filters are specific to each contact and do not affect whether or not other contacts receive notifications.

The first filter that must be passed for each contact are the notification options. Each contact definition contains options that determine whether or not service notifications can be sent out for warning states, critical states, and recoveries. Each contact definition also contains options that determine whether or not host notifications can be sent out when the host goes down, becomes unreachable, or recovers. If the host or service notification does not pass these options, **the contact will not be notified**. If it does pass these options, the notification gets passed to the next filter... Note: Notifications about host or service recoveries are only sent out if a notification was sent out for the original problem. It doesn't make sense to get a recovery notification for something you never knew was a problem...

The last filter that must be passed for each contact is the time period test. Each contact definition has a `<notification_period>` option that specifies which time period contains valid notification times for the contact. If the time that the notification is being made does not fall within a valid time range in the specified time period, **the contact will not be notified**. If it falls within a valid time range, the contact gets notified!

### Notification Methods

You can have Nagios notify you of problems and recoveries pretty much anyway you want: pager, cellphone, email, instant message, audio alert, electric shocker, etc. How notifications are sent depend on the notification commands that are defined in your object definition files.



Note: If you install Nagios according to the quickstart guide, it should be configured to send email notifications. You can see the email notification commands that are used by viewing the contents of the following file: `/usr/local/nagios/etc/objects/commands.cfg`.

Specific notification methods (paging, etc.) are not directly incorporated into the Nagios code as it just doesn't make much sense. The "core" of Nagios is not designed to be an all-in-one application. If service checks were embedded in Nagios' core it would be very difficult for users to add new check methods, modify existing checks, etc. Notifications work in a similar manner. There are a thousand different ways to do notifications and there are already a lot of packages out there that handle the dirty work, so why re-invent the wheel and limit yourself to a bike tire? Its much easier to let an external entity (i.e. a simple script or a full-blown messaging system) do the messy stuff. Some messaging packages that can handle notifications for pagers and cellphones are listed below in the resource section.

### Notification Type Macro

When crafting your notification commands, you need to take into account what type of notification is occurring. The `$NOTIFICATIONTYPE$` macro contains a string that identifies exactly that. The table below lists the possible values for the macro and their respective descriptions:

Value	Description
PROBLEM	A service or host has just entered (or is still in) a problem state. If this is a service notification, it means the service is either in a WARNING, UNKNOWN or CRITICAL state. If this is a host notification, it means the host is in a DOWN or UNREACHABLE state.
RECOVERY	A service or host recovery has occurred. If this is a service notification, it means the service has just returned to an OK state. If it is a host notification, it means the host has just returned to an UP state.
ACKNOWLEDGEMENT	This notification is an acknowledgement notification for a host or service problem. Acknowledgement notifications are initiated via the web interface by contacts for the particular host or service.
FLAPPINGSTART	The host or service has just started flapping.
FLAPPINGSTOP	The host or service has just stopped flapping.
FLAPPINGDISABLED	The host or service has just stopped flapping because flap detection was disabled..
DOWNTIMESTART	The host or service has just entered a period of scheduled downtime. Future notifications will be suppressed.
DOWNTIMESTOP	The host or service has just exited from a period of scheduled downtime. Notifications about problems can now resume.
DOWNTIMECANCELLED	The period of scheduled downtime for the host or service was just cancelled. Notifications about problems can now resume.

### Helpful Resources

There are many ways you could configure Nagios to send notifications out. Its up to you to decide which method(s) you want to use. Once you do that you'll have to install any necessary software and configure notification commands in your config files before you can use them. Here are just a few possible notification methods:

- Email
- Pager
- Phone (SMS)
- WinPopup message
- Yahoo, ICQ, or MSN instant message
- Audio alerts
- etc...

Basically anything you can do from a command line can be tailored for use as a notification command.

If you're looking for an alternative to using email for sending messages to your pager or cellphone, check out these packages. They could be used in conjunction with Nagios to send out a notification via a modem when a problem arises. That way you don't have to rely on email to send notifications out (remember, email may \*not\* work if there are network problems). I haven't actually tried these packages myself, but others have reported success using them...

- Gnokii (SMS software for contacting Nokia phones via GSM network)
- QuickPage (alphanumeric pager software)
- Sendpage (paging software)
- SMS Client (command line utility for sending messages to pagers and mobile phones)

If you want to try out a non-traditional method of notification, you might want to mess around with audio alerts. If you want to have audio alerts played on the monitoring server (with synthesized speech), check out Festival. If you'd rather leave the monitoring box alone and have audio alerts played on another box, check out the Network Audio System (NAS) and rplay projects.