Understanding Notification Escalations in Nagios XI 5

Purpose

This document describes how to How To Understand Notification Escalations in Nagios XI 5.

Note: If you are using **Nagios XI 2024**, please refer to the updated document.

Notification Basics

Nagios XI has Host Escalations and Service Escalations that allow for enhanced notifications. A brief explanation of notifications is required first.

A host or service object is configured with several notification options and are also tied to the check intervals of the host or service object. When a Nagios XI Object enters a HARD state (that is not UP or OK) the following occurs:

- Nagios XI checks to see if the object has notifications enabled and appropriate contacts
- Nagios XI checks to see if there is a notification delay defined
- Nagios XI sends the notification to the contacts defined on the object (including contacts inherited through templates)
- This notification is the first notification (notification #1)
- Nagios XI uses the objects **notification interval** to determine when the next notification will be sent (commonly 60 minutes)
- Nagios XI schedules the next check of the object using the **check interval** directive (commonly every 5 minutes)

When the next check of the object occurs, the following happens:

- Nagios XI checks the objects **notification interval** to determine if enough time has passed since the last notification was sent
- In the above example using a check interval of 5 minutes, the object needs to be checked 12 more times (12 x 5 = 60) before the next notification is sent
- When the next notification is sent, this is determined to be notification #2

This continues until the object problem is acknowledged OR the object returns to an UP or OK state.

What Are Escalations And Why Are They Required

Escalations allow you to send notifications based on different criteria. In the **Notification Basics** section, the notifications will always be sent to the contacts defined on the object. Escalations allow you to send the notifications to different contacts after x number of notifications have been sent.

Hence you can escalate a problem from the **Help Desk Team** to the **On Call Team** and then to the **Department Manager**.

Escalations allow you to send notifications to the required people at the right time. The **Department Manager** does not want to be notified of a problem initially, they only want to know about it if the **Help Desk Team** and the **On Call Team** have not addressed the issue.

Escalations Take Priority

Escalations will overwrite ANY notification options on the host or service object when they take affect (this includes the contacts/contactgroups). This is very important to remember when determining who should be notified. When an escalation no longer takes effect then the notification options on the host or service object will be used.

When Are Notifications Escalated

Notifications are escalated if and only *if* one or more escalation definitions match the current notification that is being sent out. If a host or service notification does not have any valid escalation definitions that apply to it, the contact group(s) specified in either the host group or service definition will be used for the notification.

In the **Notification Basics** section, it was explained that each notification is numbered, notification #1. notification #2 and so on. Every escalation definition defines the first and last notification, this is what determines when an escalation definition takes effect. This means you create multiple escalations for the different notification numbers.

Here's an example:

- Help Desk Team
    - First Notification: 1
    - Last Notification: 3
- Help Desk Team, On Call Team
    - First Notification: 4
    - Last Notification: 6
- Help Desk Team, On Call Team, Department Manager
    - First Notification: 7
    - Last Notification: 0 (using 0 means to keep using this escalation entry until the problem is resolved)

NAGIOS®

Contacts vs Contact Groups

It is best practice to add your contacts to contact groups and use these groups in your escalations. It is easier to add/remove users from groups, as every object that uses these groups will be automatically updated. Otherwise, you will need to update every object / escalation whenever you need to make contact changes, this can take a long time and can lead to mistakes, resulting in notifications not going to the correct contacts.

Host Escalation Example

Let's use the example under **When Are Notifications Escalated** and create three escalations. Navigate to **Configure > Core Config Manager** and under **Alerting** select **Host Escalations**.
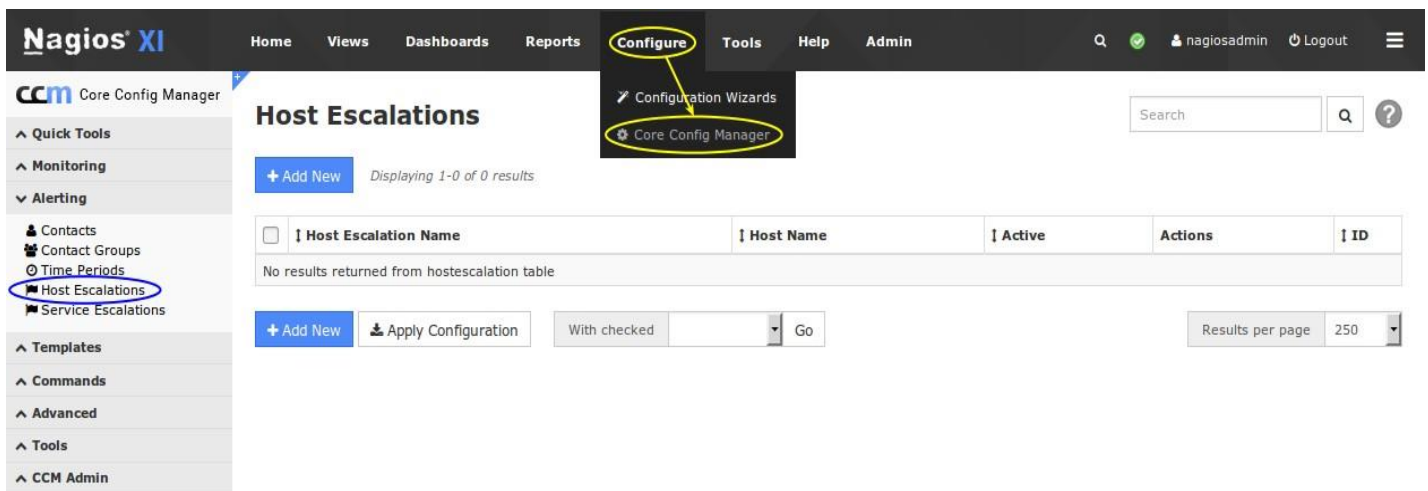
Click the **Add New** button to create a new escalation.

In the Config Name field provide a name.

The **First Notification** is **1** and **Last Notification** is **3**. These fields determine when this escalation will be used.

The **Notification Interval** is **60**. This determines how frequently these notifications will be sent.
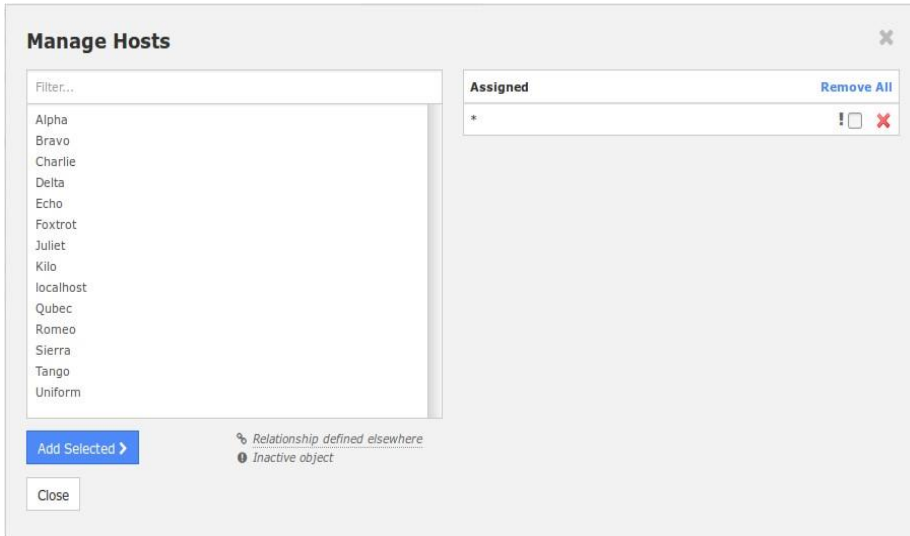
The **Escalation Options** has **Down**, **Up** and

Unreachable selected, notifications will be sent when a host is in any of these states.

Make sure a **Host Escalation Period** is selected.

Nagios®

Click the **Manage Hosts** button to select the hosts that this escalation applies to.

In this example we want all hosts to use this escalation, so we have used the asterisk * to do this. You can also use a Host Group to make it easier to target a specific set of hosts.

Click **Close** when you finish selecting hosts.

Click the **Manage Contact Groups** button to select the contact groups that this escalation applies to.
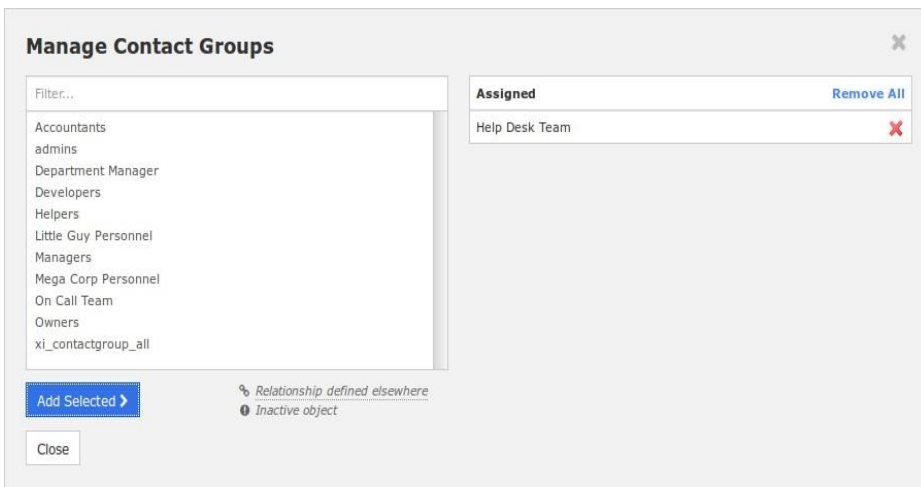
In this example the **Help Desk**

Team group has been selected.

Click **Close** when you have finished selecting contact groups.

Nagios®

Click the **Save** button to create the new host escalation.

Repeat the above steps to create these additional escalations:

- Config Name: **All Hosts - 4 to 6**
  - First Notification: **4**
  - Last Notification: **6**
  - Notification Interval: **45**
  - Escalation Options: **Down**, **Up** and **Unreachable**
  - Hosts: *****
  - Contact Groups: **Help Desk Team**, **On Call Team**
- Config Name: **All Hosts - 7 to 0**
  - First Notification: **7**
  - Last Notification: **0**
  - Notification Interval: **30**
  - Escalation Options: **Down**, **Up** and **Unreachable**
  - Hosts: *****
  - Contact Groups: **Help Desk Team**, **On Call Team**, **Department Manager**

Apply Configuration to make these escalations become live.

In the example above, three escalations were created to implement host notifications for all hosts on the Nagios XI server. The options used were deliberately simplified to demonstrate how escalations can be used.

When defining escalations, it is important to keep in mind that any contact groups that were members of "lower" escalations (i.e. those with lower notification number ranges) should also be included in "higher" escalation definitions. This should be done to ensure that anyone who is notified of a problem *continues* to get notified as the problem is escalated (you can see this in the example above).

Recovery notifications

Recovery notifications behave slightly differently than problem notifications when it comes to escalations. Using the example above, after three problem notifications the host returns to an UP state. When the recovery notification is sent out, who gets notified?

Nagios®

The recovery is the fourth notification that gets sent out. However, the escalation logic is smart enough to determine that only those people who were notified about the problem on the third notification should be notified about the recovery. In this case, only the **Help Desk Team** would be notified of the recovery, and the **On Call Team** would not receive any notifications.

Notification Intervals

In Nagios XI you can change the frequency at which escalated notifications are sent out by using the **Notification Interval** option.

When the host notification is escalated on the 4th, 5th and 6th notifications, an interval of **45** minutes will be used between notifications (instead of the 60-minute interval defined on the previous escalation). On the 7th and subsequent notifications, the notification interval will be 30 minutes. This way, as time passes by since the problem originally occurred, more people will be notified more frequently.

Since it is possible to have overlapping escalation definitions for a particular host or hostgroup, and since a host can be a member of multiple hostgroups, Nagios must decide on what to do as far as the notification interval is concerned when escalation definitions overlap. In any case where there are multiple valid escalation definitions for a particular notification, Nagios will choose the smallest notification interval.

One last note about notification intervals, an interval of 0 means that Nagios should only send a notification out for the first valid notification during that escalation definition. All subsequent notifications will be suppressed.

Service Escalation Example

Service escalations are almost identical to host escalations. With that in mind, we're going to create a more complicated series of escalations.

There are two hosts (**Alpha** and **Bravo**) that have the same service called **CPU Usage**. The details of each service are as follows:

- Alpha CPU Usage

    o Contact Groups: **Accountants**

    o Interval: **30**

- Bravo CPU Usage

    o Contacts: **Bob**

    o Interval: 60

Nagios®

We are going to create the same escalations that will be used by both (host/service).

- Help Desk Team
    - First Notification: **2**
    - Last Notification: **4**
- On Call Team
    - First Notification: **3**
    - Last Notification: **6**
- Department Manager
    - First Notification: **7**
    - Last Notification: **0** (*using 0 means to keep using this escalation entry until the problem is resolved*)

Navigate to **Configure > Core Config Manager** and under Alerting select Service Escalations.

Click the **Add New** button to create a new escalation.

In the Config Name field provide a name.

The **First Notification** is **2** and **Last Notification** is **4**.

The **Notification Interval** is **60**.

The **Escalation Options** has **Warning**, **Critical**, **Ok** and **Unknown** selected.

Make sure a **Service Escalation Period** is selected.

Click the **Manage Hosts** button.

Select the hosts **Alpha** and **Bravo**.

Click **Close** when you finish selecting hosts.

Click the **Manage Services** button.

Select the service **CPU Usage**.

This list of services is of all the services that exist on the Nagios XI server. It's important that all the hosts that are assigned to this escalation have the service(s) you select here.

Click **Close** when you've finished selecting services.
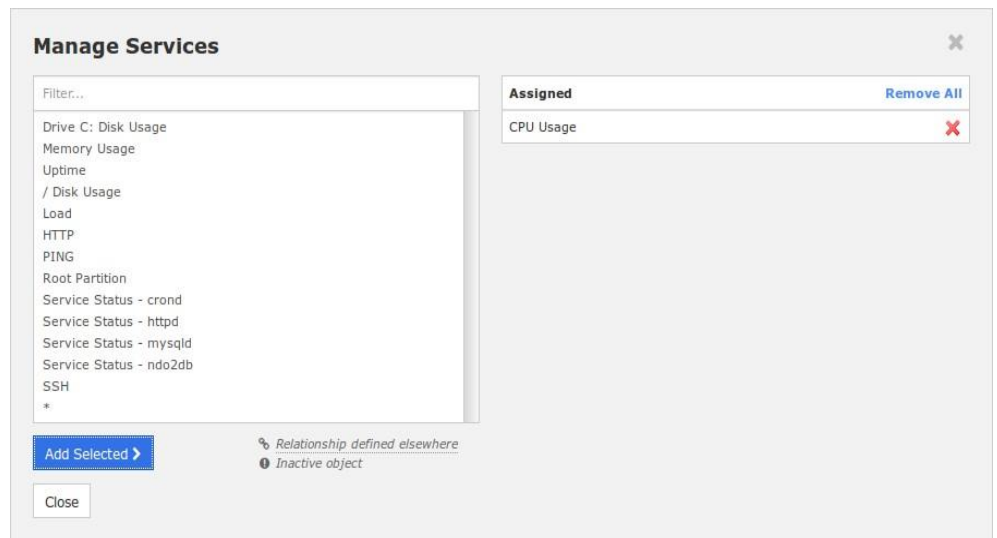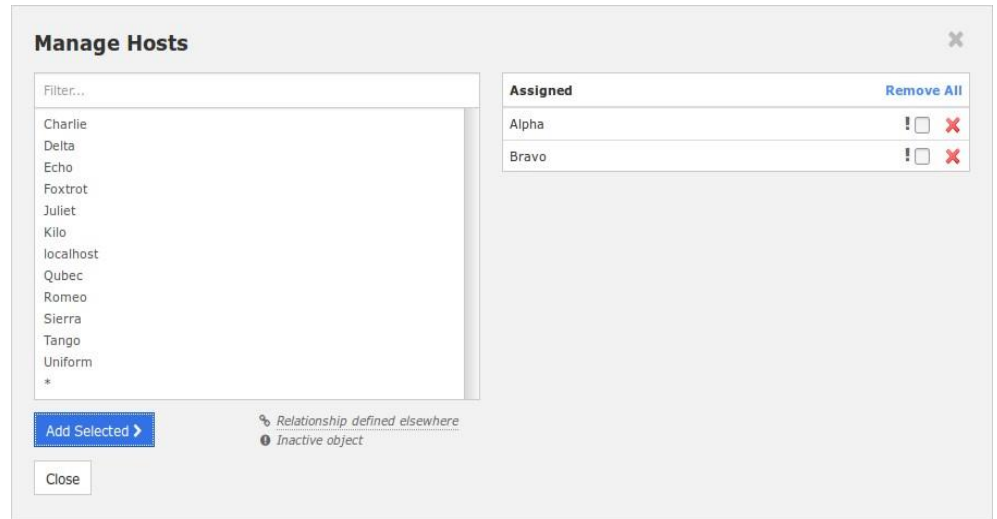
Click the **Manage Contact Groups** button and select the contact groups that this escalation applies to.

Click **Close** when you have finished selecting contact groups.

Click the **Save** button to create the new service escalation.

Repeat the above steps to create these additional escalations and then **Apply Configuration**:
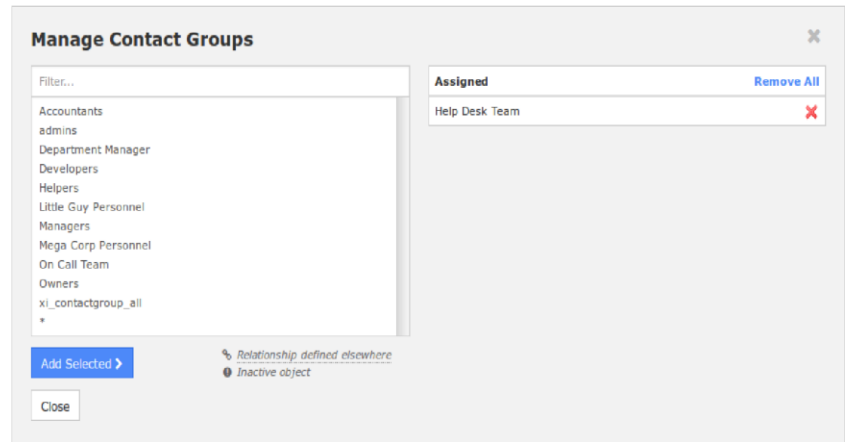
- Config Name: **CPU Usage - 3 to 6**
    - First Notification: **3**
    - Last Notification: **6**
    - Notification Interval: **45**
    - Escalation Options: Warning, **Critical** and **Unknown**
    - Hosts: Alpha and **Bravo**
    - Contact Groups: **On Call Team**
- Config Name: **CPU Usage - 7 to 0**
    - First Notification: **7**
    - Last Notification: **0**
    - Notification Interval: **30**
    - Escalation Options: Critical and **Unknown**
    - Hosts: Alpha and **Bravo**
    - Contact Groups: **Department Manager**

The following table demonstrates the notifications that will be sent and how these escalations apply (or not).

Time	Event	Alpha CPU Usage	Bravo CPU Usage
03:00:00 AM	Service enters HARD WARNING state	Notification sent to Accountants Next notification @ 03:30:00 This is Notification #1	Notification sent to Bob Next notification @ 04:00:00 This is Notification #1
03:30:00 AM	Service still HARD WARNING state	Notification sent to Help Desk Team Next notification @ 04:30:00 This is Notification #2	
04:00:00 AM	Service still HARD WARNING state		Notification sent to Help Desk Team Next notification @ 05:00:00 This is Notification #2
04:30:00 AM	Service still HARD WARNING state	Notification sent to Help Desk Team Notification sent to On Call Team Next notification @ 05:15:00 This is Notification #3	
05:00:00 AM	Service still HARD WARNING state		Notification sent to Help Desk Team Notification sent to On Call Team Next notification @ 05:45:00 This is Notification #3
05:15:00 AM	Service still HARD WARNING state	Notification sent to Help Desk Team Notification sent to On Call Team Next notification @ 06:00:00 This is Notification #4	
05:45:00 AM	Service still HARD WARNING state		Notification sent to Help Desk Team Notification sent to On Call Team Next notification @ 06:30:00 This is Notification #4
06:00:00 AM	Service still HARD WARNING state	Notification sent to On Call Team Next notification @ 06:45:00 This is Notification #5	
06:30:00 AM	Service still HARD WARNING state		Notification sent to On Call Team Next notification @ 07:15:00 This is Notification #5
06:45:00 AM	Service still HARD WARNING state	Notification sent to On Call Team Next notification @ 07:30:00 This is Notification #6	
07:15:00 AM	Service still HARD WARNING state		Notification sent to On Call Team Next notification @ 08:00:00 This is Notification #6
07:30:00 AM	Service still HARD WARNING state	Notification sent to Accountants Next notification @ 08:00:00 This is Notification #7	
08:00:00 AM	Service still HARD WARNING state	Notification sent to Accountants Next notification @ 08:30:00 This is Notification #8	Notification sent to Bob Next notification @ 09:00:00 This is Notification #7

NAGIOS®

08:14:23 AM	Service recovers to OK state	Notification sent to Accountants This is Notification #9	Notification sent to Bob This is Notification #8

The example above was to demonstrate how the notification options on the service object are still used when the criteria on the escalations are not met. Most noticeably, because this service was in a WARNING state the **CPU Usage - 7 to 0** escalation was not used because it was only configured for CRITCAL or UNKNOWN states.

You will have also noticed that the escalation **CPU Usage - 2 to 4** only applied from notification #2. This meant that the service object's notification options were used for notification #1. The Alpha CPU Usage service had a notification interval of 30, so notification #2 was sent sooner than Bravo CPU Usage. Additionally, when the escalation **CPU Usage - 2 to 4 applied**, the notification interval increased back to 60 minutes.

The other difference in this example is that the First / Last escalations overlapped AND only the specific contact groups were used in each escalation. This meant that the **Help Desk Team** contact group did not receive notifications after notification #4, however the **On Call Team** received notifications from #3 onwards.

The purpose of the example above was to demonstrate how you can create complicated and flexible escalations.

Finishing Up

This completes the documentation on How To Understand Notification Escalations in Nagios XI. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum   Visit Nagios Knowledge Base   Visit Nagios Library

Nagios®