How To Configure Audio Alerts With Nagios XI 2024 And 2026

Purpose

This document describes how to configure audio alerts with Nagios XI 2024 and 2026.

Overview

An audio alert from Nagios is just another type of notification. Instead of receiving an email informing you of a problem, an audio alert would be heard without needing to look at the computer. So, what exactly is an audio alert?

- This solution generates the audio using a program called **Flite**, it is a text to speech synthesis engine.
- In real time the Pi will generate the audio and output it to the via HDMI or 3.5mm headphone jack.
- This means you can customize the alert to say whatever you want, for example:
    - The host server01 is currently down
    - The service disk usage C: for the host server01 is currently in a warning state, the last output was:
      `C:\ - total: 99.51 Gb - used: 25.89 Gb (26%) - free 73.62 Gb (74%)`

A Raspberry Pi is used in this solution because:

- It is a cheap device that is commonly used to display Nagios health on a TV or screen in NOC.
- It has HDMI and or 3.5mm headphone jack audio output.
- The audio alerts are not physically tied to the Nagios XI Server.
    - Commands are sent to the Pi using an SSH connection.
    - This allows you to easily scale up this solution to multiple Pi's in different rooms/locations.
- It provides an easy to implement solution

This solution is not limited to a Raspberry Pi, as you follow the documentation you will see that it can be implemented on any Linux Distribution.

Requirements

To be able to implement this solution the following requirements must be met:

- Raspberry Pi (3 Model B used in this example)
- A TV/Screen with built in speakers to play audio or external speakers that can connect to the 3.5mm headphone jack.
- Raspbian Stretch Desktop (Lite also works if you wanted an audio only solution without a display).
- A static IP address on the Pi as the Nagios Server will be sending commands to it to play the audio.
- An active internet connection as several components will need to be installed.

Once you have met all those requirements you will be able to proceed with the steps in this document.

Enable SSH / Create Nagios User

The first step is to enable the SSH service and create a nagios user. SSH is how Nagios will send the commands to play the audio. It will also provide the ability to monitor the Pi itself using the SSH Proxy configuration wizard in Nagios XI.

On your Pi Desktop open a terminal session. Execute this command to become the root user:

```
sudo su
```

Enable the SSH service so it starts on boot and start the service by executing these commands:

```
systemctl enable ssh.service
systemctl start ssh.service
```

Now create the nagios user along with adding the user account to the audio group so it has permission to output the audio. You will need to provide a password for the nagios user. Execute these commands:

```
useradd -m nagios
usermod -a -G audio nagios
passwd nagios
```

Once it is completed you need to configure the SSH keys between the Nagios XI server and the Pi (this allows passwordless authentication). Establish a terminal session to your Nagios XI server as the root user. Execute this command to become the nagios user on the Nagios XI server:

```
su nagios
```

Execute this command to create an SSH key:

```
ssh-keygen
```

Press **ENTER** (accepting defaults) when prompted for a filename and passphrase. Public and private SSH keys will be generated and saved in the following directory:

```
/home/nagios/.ssh
```

In the following commands replace **xxx.xxx.xxx.xxx** with the **IP address of your Pi**. This command will copy the SSH key to the Pi:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub nagios@xxx.xxx.xxx.xxx
```

You will be prompted to add the host to the list of new hosts, you need to type **yes** to proceed and then you will need to type the password for the nagios user (the nagios user you created on the Pi).

Now to verify that you can login to the Pi without supplying a password. Continuing with the terminal session on the Nagios XI server execute the following command:

```
ssh nagios@xxx.xxx.xxx.xxx
```

If the SSH keys are configured properly you should be able to login to the Pi machine without supplying credentials. Simply type exit to close the SSH session.

Also test the `check_by_ssh` plugin, run the following command:

```
/usr/local/nagios/libexec/check_by_ssh -H xxx.xxx.xxx.xxx -C uptime
```

If things are setup properly, you should get output from the “uptime” command on the remote server that looks like the following:

```
12:01:42 up 35 min, 1 user, load average: 0.00, 0.01, 0.05
```

If your output is similar, then you are ready to proceed to the next step.

Install Flite

Next is to install the **Flite application** and test that the audio works.

Return to your Pi Desktop terminal session and execute the following command:

```
apt-get install -y flite
```

Once this has been installed you can test it straight away.

Make sure your speakers are turned on and execute the following command:

```
flite -t "testing 1 2 3"
```

If you heard that come through your speakers, then everything is working correctly. If it did not, double check your display or external speakers have the volume turned up and execute the command again. If you still do not hear the audio, then please continue to the Troubleshooting section.

There are several voices available in **Flite**, personally I like the "slt" voice, for example:

```
flite -t "testing 1 2 3" -voice slt
```

In the remainder of this documentation, I am going to use this voice.

Test Flite From Nagios XI

Now you can test from the Nagios XI server to see if everything is configured correctly. In your terminal session on your Nagios XI server execute the following commands (replace **xxx.xxx.xxx.xxx** with the **IP address of your Pi**):

```
cd /usr/local/nagios/libexec/
./check_by_ssh -H xxx.xxx.xxx.xxx -C '/usr/bin/flite -t "testing 1 2 3" voice slt'
```

You should hear the audio and receive output like:

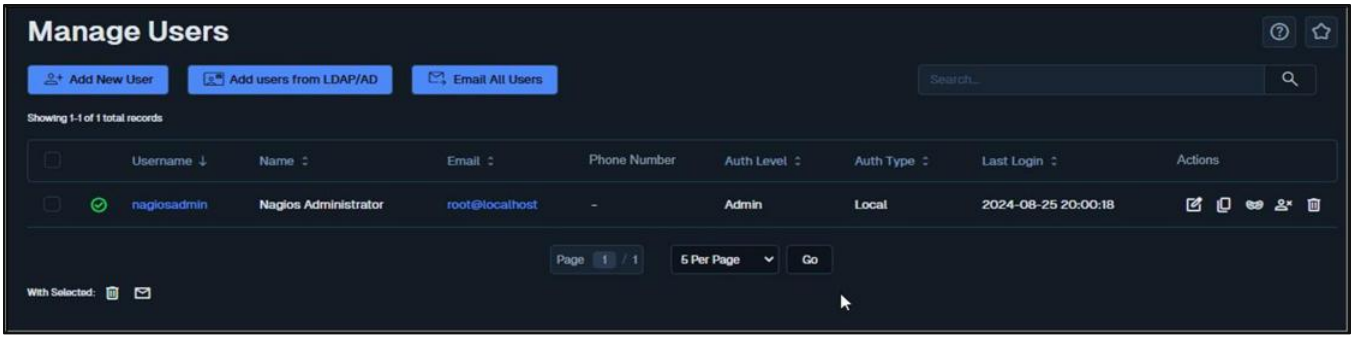
```
OK - check_by_ssh: Remote command '/usr/bin/flite -t "testing 1 2 3" -voice
slt' returned status 0
```

If you heard the audio, then you are ready to proceed to the next step. If you do not hear the audio, then please continue to the Troubleshooting section.

Configure Nagios XI

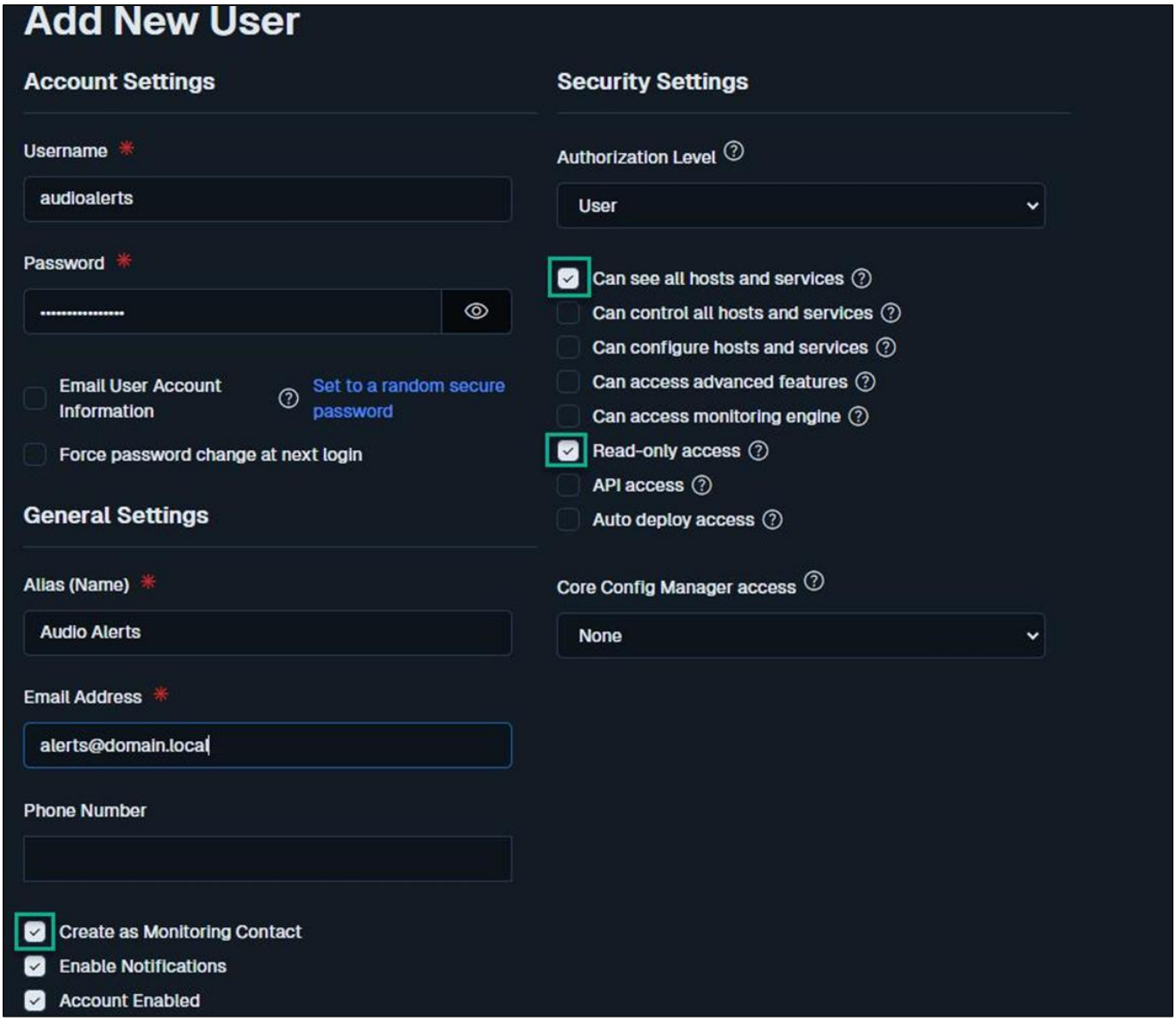
The way Nagios XI will decide if it needs to generate an audio alert is the same as if it needs to send an email alert. A Nagios XI user account will be created and used to send notifications too, this is how you will be able to selectively define what hosts and services will generate audio alerts. The next step will create the user and define the host and service notification commands that will be triggered when a notification is sent.

1. Login to your Nagios XI server and navigate to **Admin > Users > Manage Users**.



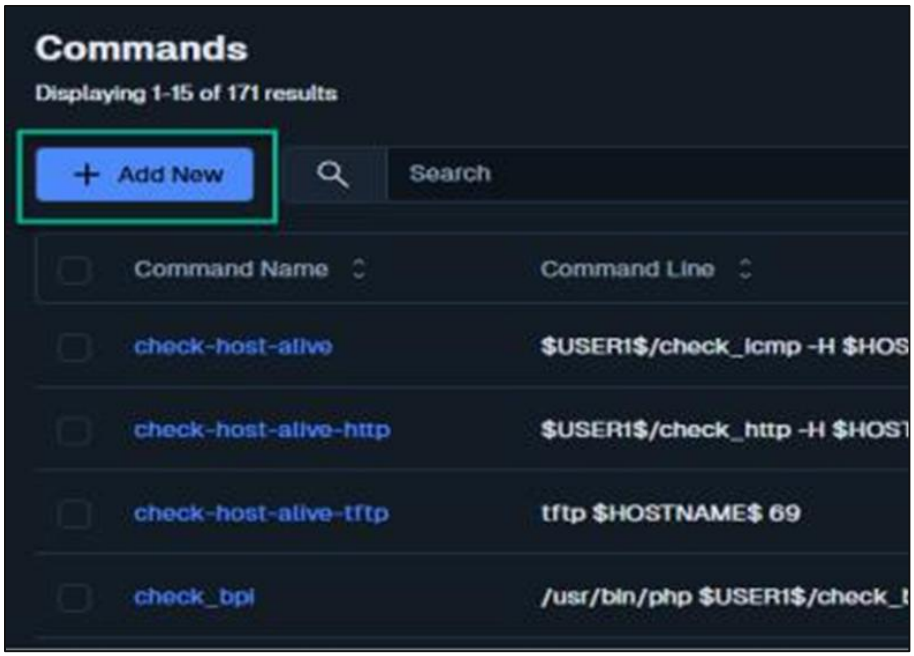
2. Click the **Add New User** button and populate the fields as required.

3. The check boxes highlighted in the screenshot are the most important ones that need to be enabled.

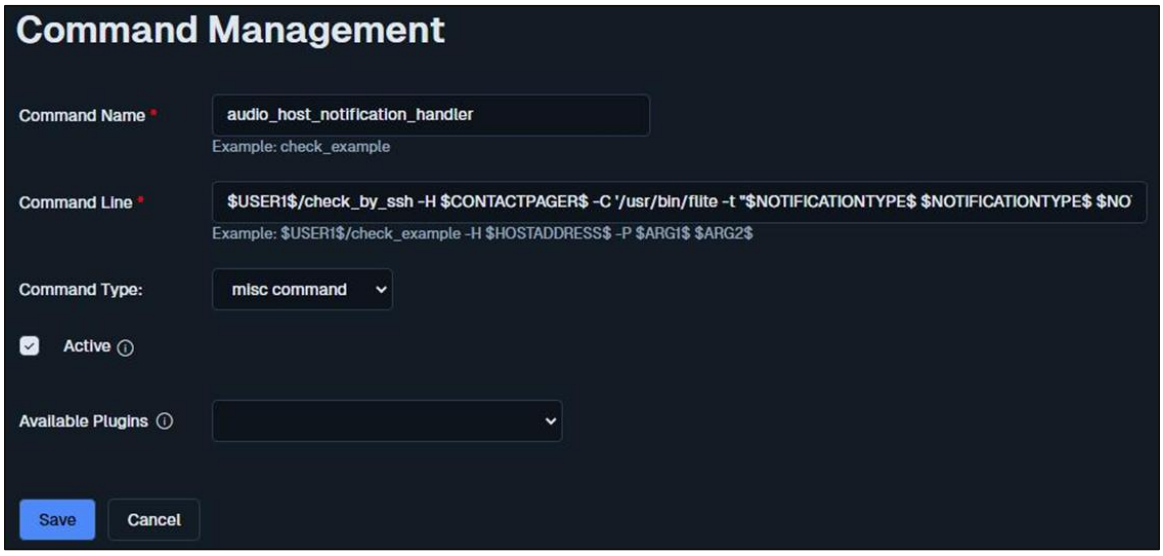


4. Click the **Add User** button to create the new user.

5. Now you need to create commands for the notifications, navigate to **Configure > Core Config Manager**.



6. In the nav pane expand **Commands** and click **>_ Commands**.
7. Click the **+Add New** button.



8. Populate the **Command Name** field with `audio_host_notification_handler`. The **Command Line** field is shown on the following page. Make sure the **Command Type** is defined as "misc command".

9. Click the **Save** button once you populated the fields.
10. You will need to repeat the same steps to create the `audio_service_notification_handler` command. The full details of both commands are shown below. The commands are one extensive line, they are wrapped over several lines due to their size.

Command Name:

```
audio_host_notification_handler
```

Command Line:

```
$USER1$/check_by_ssh -H $CONTACTPAGER$ -C '/usr/bin/flite -t
"$NOTIFICATIONTYPE$ $NOTIFICATIONTYPE$ $NOTIFICATIONTYPE$ the host $HOSTNAME$ has a $HOSTSTATE$ state" \-
voice slt'
```

Command Name:

```
audio_service_notification_handler
```

Command Line:

```
$USER1$/check_by_ssh -H $CONTACTPAGER$ -C '/usr/bin/flite -t
"$NOTIFICATIONTYPE$
$NOTIFICATIONTYPE$ $NOTIFICATIONTYPE$ the service $SERVICEDESC$ on host
$HOSTNAME$ has a $SERVICESTATE$ state" -voice slt'
```

These commands are how Nagios XI dynamically generates the audio alert. It can pull values like `$HOSTNAME$` and `$SERVICEDESC$` to provide alerts specific to the object the alert is for.

You can see that `$NOTIFICATIONTYPE$` has been used three times at the beginning of each message, there are two reasons for this:

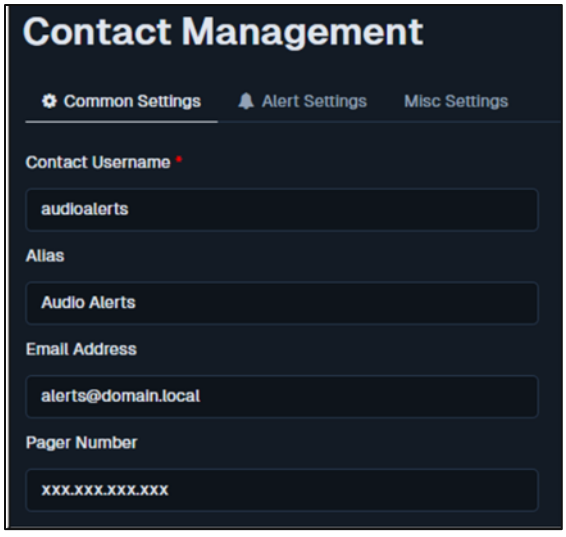
- To give you a moment to realize you need to listen to the audio alert that is going to be announced.
- Depending on the HDMI / audio equipment being used the beginning of the message can be cut off, this can do with the audio equipment "waking up". Having `$NOTIFICATIONTYPE$` three times ensures you don't miss the valuable information.

You will also notice the command uses `-H $CONTACTPAGER$` for the Pi address. In the next step you will configure the audioalerts contact object with the IP address of the of the Pi. By doing it this way you can easily use multiple Pi's to send audio alerts to, it allows the solution to scale easily.

Here is a screenshot of both commands that have been created:

11. Now that the commands have been created, they need to be assigned to the contact. In the nav pane expand **Alerting** and click **Contacts**.

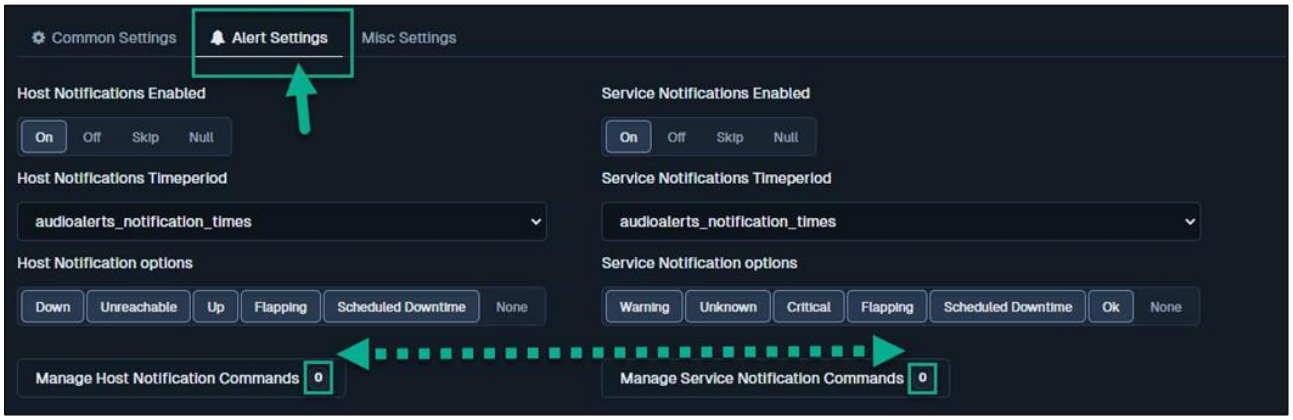
12. On the **Common Settings** tab you will need to provide the IP address of the Pi in the **Pager Number** field.



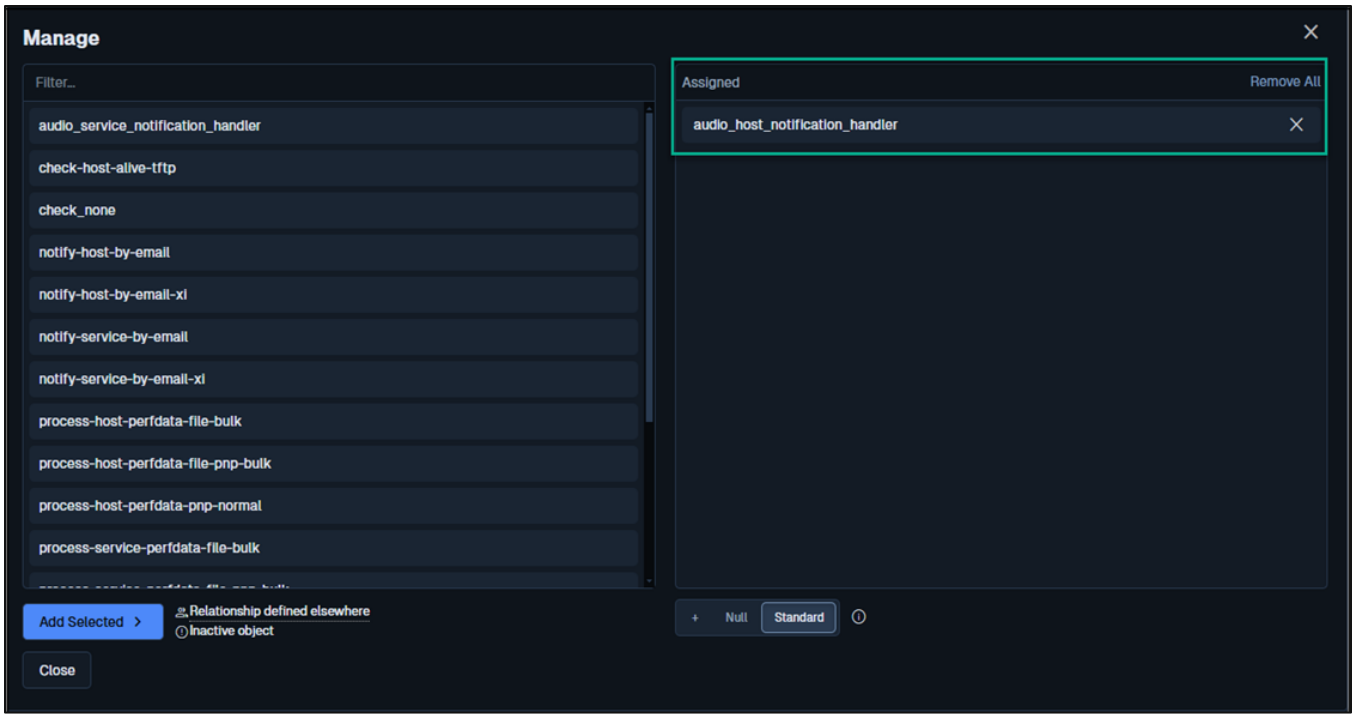
13. Click the **Alert Settings** tab. There are two buttons here:

    Manage Host Notification Commands

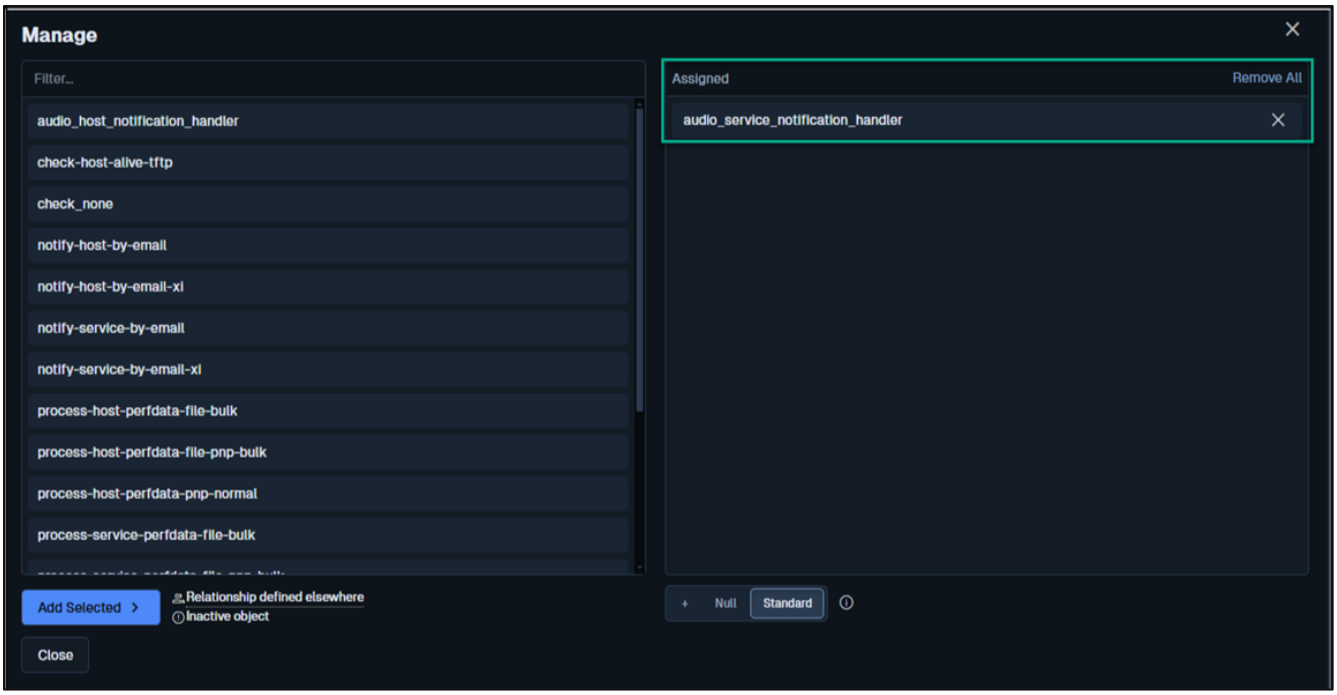
    Manage Service Notification Commands



14. Click the **Manage Host Notification Commands** button, this will bring up the **Manage** window.



15. In the **Manage** window you need to select the `audio_host_notification_handler` command in the left pane and then click the Add Selected button, so it appears in the right hand pane. This is for HOST notifications, hence why you added the `audio_host_xxx` command only.



16. Click the **Close** button.
17. Click the **Manage Service Notification Commands** button, this will bring up the **Manage** window.
18. In the **Manage** window you need to select the `audio_service_notification_handler` command in the left pane and then click the **Add Selected** button, so it appears in the right hand pane.

    This is for SERVICE notifications, hence why you added the `audio_service_xxx` command only.

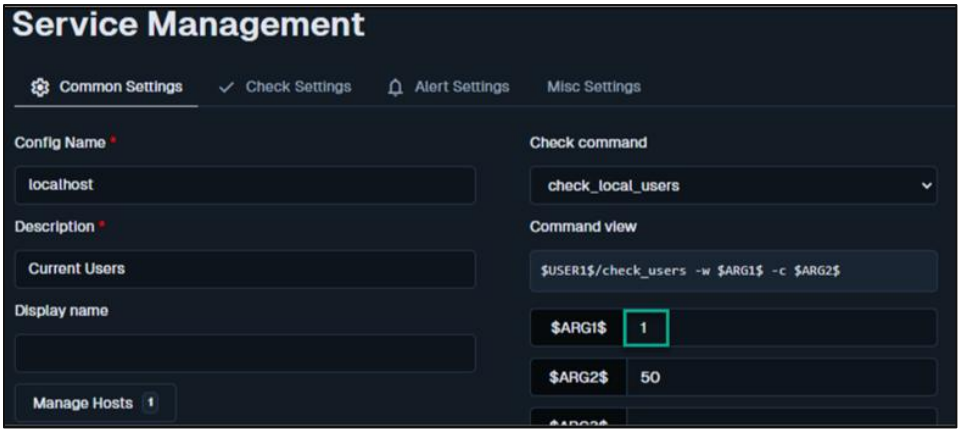
19. Click the **Close** button.
20. Click the **Save** button, this completes the changes required for the **Contact** object.

Once you've made your selections click the **Apply Configuration** button to push these changes into the running monitoring configuration.

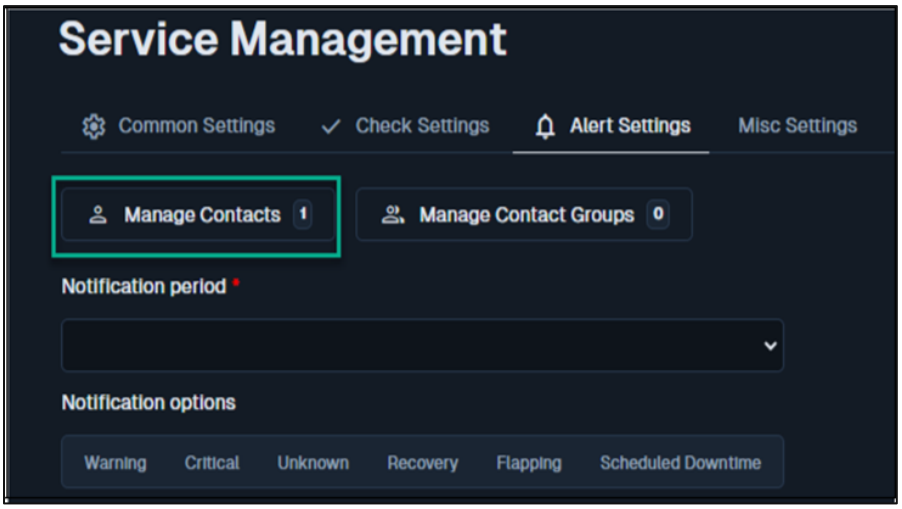
Assign Contact To Objects And Test

At this point all that is remaining is to assign the audioalerts contact to the Nagios Host and Service objects that you want to receive audio alerts for. This example is going to use the **Current Users** service for the localhost object. The thresholds will be adjusted so the service can be put into a warning state, and you will be able to hear the audio alert.

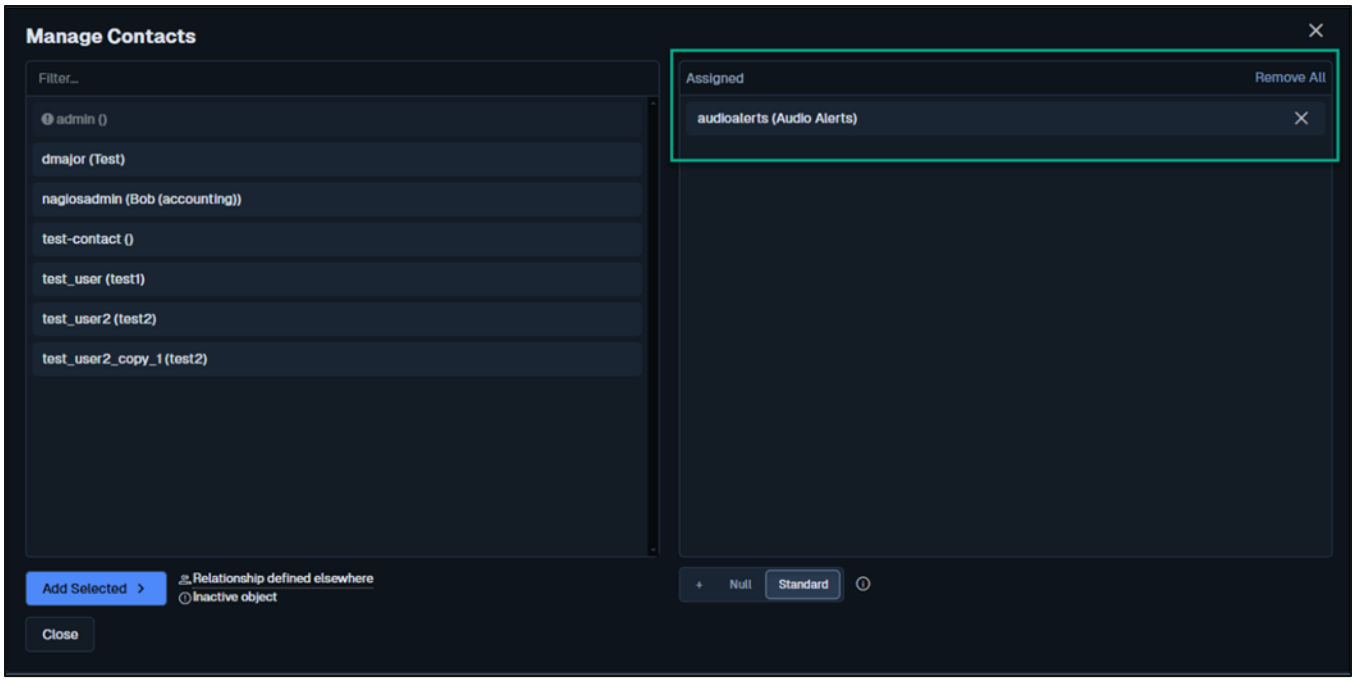
1. In the nav pane expand **Monitoring** and click **Services**.
2. Click the **Current Users** service object to edit it.
3. Change the `$ARG1$` field to 1, this will cause it to go into a warning state when more than 1 user is logged in.



4. Click the **Alert Settings** tab and then click the **Manage Contacts** button.

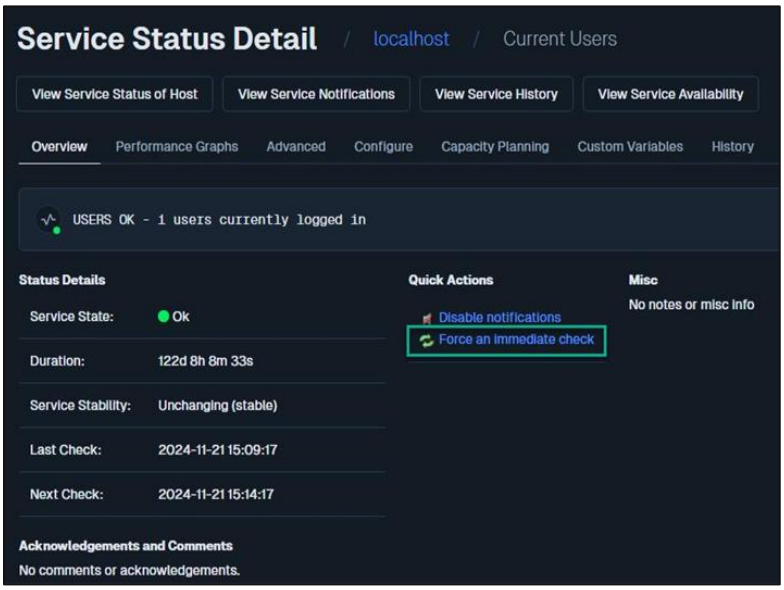


5. In the **Manage Contacts** window you need to select the **audioalerts contact** in the left pane and then click the **Add Selected** button, so it appears in the right hand pane.

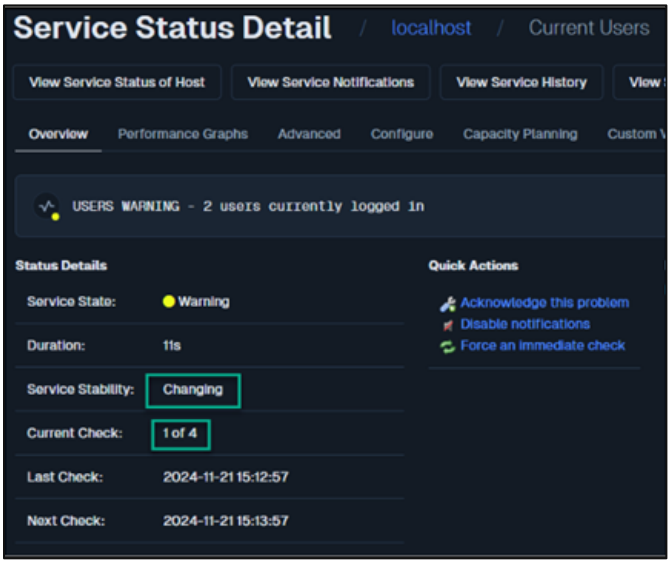


6. Click the **Close** button and then the **Save** button.
7. Once you've made your selections click the **Apply Configuration** button to push these changes into the running monitoring configuration.
8. Now you are ready to generate an audio alert. Navigate to **Home > Service Detail** and click the **Current Users** service.
9. The service will be in an OK state. Earlier on the thresholds were changed so that this service will enter a WARNING state when more than 1 user is logged onto the Nagios XI server. To force it into a WARNING state all you need to do is establish 2 SSH sessions to your Nagios XI server (you should already have one open).

10. Once you have the 2 SSH sessions open click the **Force an immediate check** link under **Quick Actions**.



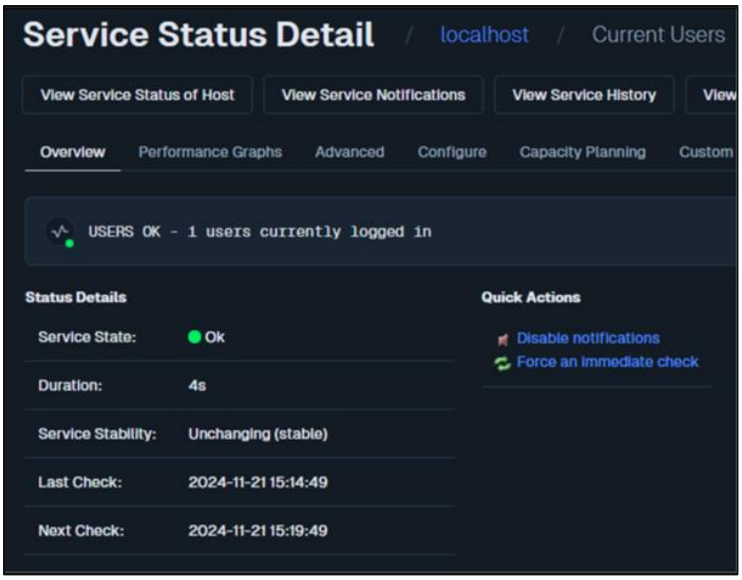
The service should go into a WARNING state; however, a notification will not be sent yet as it is in SOFT state. It must be in a WARNING state 4 times before it is in a HARD state.



11. **Force an immediate check** 3 more times to force the notification to be sent.

Once it goes into a hard state you will hear the audio notification.

12. If you close one of the SSH sessions and then force another check you will hear the recovery audio notification.



Now the service will return to an OK state.

13. You can also verify that the notification was generated by clicking the **Service Notifications Button** above the **Overview** tab

Date / Time	Host	Service	Reason	Escalated	State	Contact	Dispatcher	Information
2024-11-21 15:17:12	localhost	Current Users	Service Problem	No	Warning	audioalerts	Nagios XI	USERS WARNING - 2 users currently logged in
2024-11-21 15:17:12	localhost	Current Users	Service Problem	No	Warning	nagiosadmin	Nagios XI	USERS WARNING - 2 users currently logged in

    You can see in the screenshot above how a notification was sent to the audioalerts contact for the WARNING and OK states.

This completes all the steps required to implement audio alerts in Nagios XI using a Raspberry Pi.

Considerations

This documentation has provided you with a solution for generating audio alerts with Nagios XI however it has only scratched the surface of what is possible.

Multiple Pi's

- Simply create more Nagios XI users and define the IP address in the **Pager** field of the contact
- Assign the different contact accounts to the relevant host and service objects, different teams in different rooms can receive audio alerts that only they need to hear

Selective Alerts

- You can change the time period assigned to a contact so that audio alerts are only generated when you want them to
- The types of alerts for a contact can be defined on the contact object, only receive DOWN, WARNING and CRITICAL events if that is what you require

Descriptive Alerts

- Any of the Nagios Macros can be used to define the audio alert messages, they can be found in this Macros page.

Extensive Notification Handler

- Currently the `audio_host_notification_handler` and `audio_service_notification_handler` are oriented towards host/service problem/recovery notifications
- Other types of alerts will also be sent to these contacts, like:
    - flapping start/stop

    - custom notifications
    - acknowledgments
    - however, they might not have all the required text content
- You can define exactly what text string is sent to the Flite program however you will need to write a wrapper script that performs the logic behind all of that
- If you look at some of the other commands like the `xi_host_notification_handler` you can see that it sends all the valid macros to the handle_nagioscore_notification.php script, it is this script that performs all the logic and then actually sends the notification

Speech Synthesis Markup Language (SSML)

- Flite supports reading text that is in SSML format, this allows you to customize the audio output

Monitor The Pi

You should start monitoring the Pi from Nagios XI, it has been configured ready to run the SSH Proxy configuration wizard against it. Documentation on this is located in this Monitoring Hosts Using SSH documentation.

You will need to have some plugins installed for monitoring. This KB article shows you how to install Nagios Plugins.

The check library has a list of separate ways to monitor a device, this will help you with the SSH Proxy configuration wizard. This is located in the Nagios Knowledgebase.

Troubleshooting

If you are not receiving audio, then there are several troubleshooting steps you can follow this webpage for audio issue.

Audio output to HDMI or 3.5mm headphone jack is automatic however you can override this. If you execute this command, you can get a list of the devices on the Pi.

If you execute this command, you can get a list of the devices on the Pi:

```
/usr/bin/amixer controls
```

The output will be something like:

```
numid=3,iface=MIXER,name='PCM Playback Route'
numid=2,iface=MIXER,name='PCM Playback Switch'
numid=1,iface=MIXER,name='PCM Playback Volume'
```

```
numid=5,iface=PCM,name='IEC958 Playback Con Mask'
numid=4,iface=PCM,name='IEC958 Playback Default'
```

The numid=3,iface=MIXER,name='PCM Playback Route' device is what is used to output the audio. You can change the preferred output with the following command:

```
amixer cset numid=3 2
```

Where 0=auto, 1=headphones, 2=hdmi. In that command above you can see that the output was defined as number 2 which is the HDMI. After making the change you can do a test to see if it is working:

```
flite -t "testing 1 2 3" -voice slt
```

It has been observed that if the display is turned off when the Pi boots up the HDMI audio may not work, a reboot with the display turned on is the resolution to the problem.

Finishing Up

This completes the documentation on configure audio alerts with Nagios XI 2024. 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