How To Use The Nagios XI 2024 Actions Component

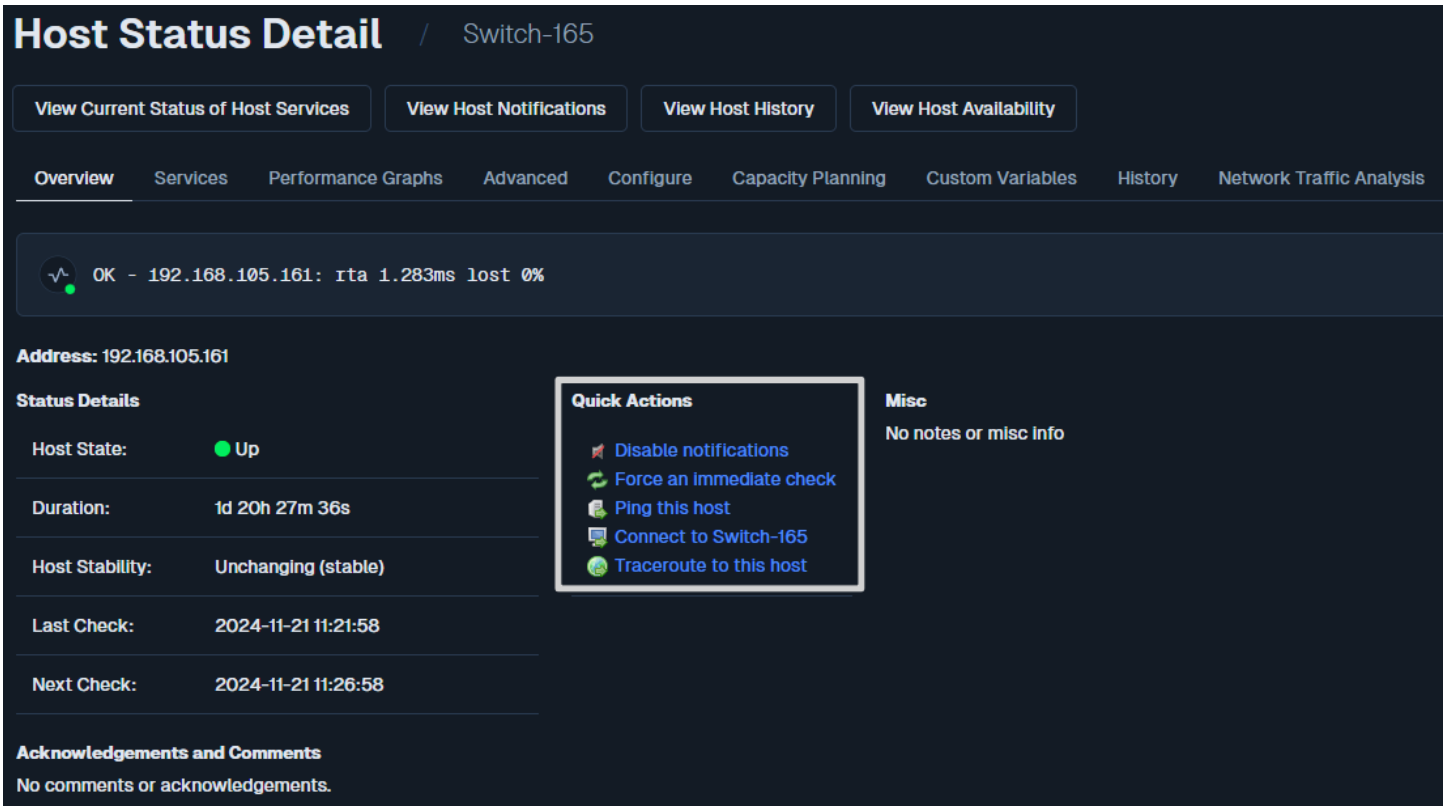
Purpose

This document describes how to use Actions Component in Nagios XI.

Editing Files

In parts of this documentation, you will be required to edit files. This documentation will use the vi text editor. When using the vi editor:

- To make changes press **i** on the keyboard first to enter insert mode.
- Press **Esc** to exit insert mode.
- When you have finished, save the changes in vi by typing `:wq` and press **Enter.**



How To Use The Nagios XI 2024 Actions Component

Component Description and Uses

The **Actions** component allows Nagios XI administrators to create customized links to do specific tasks, they appear as a link under the **Quick Actions** options on **Service** and **Host Status Detail** pages.

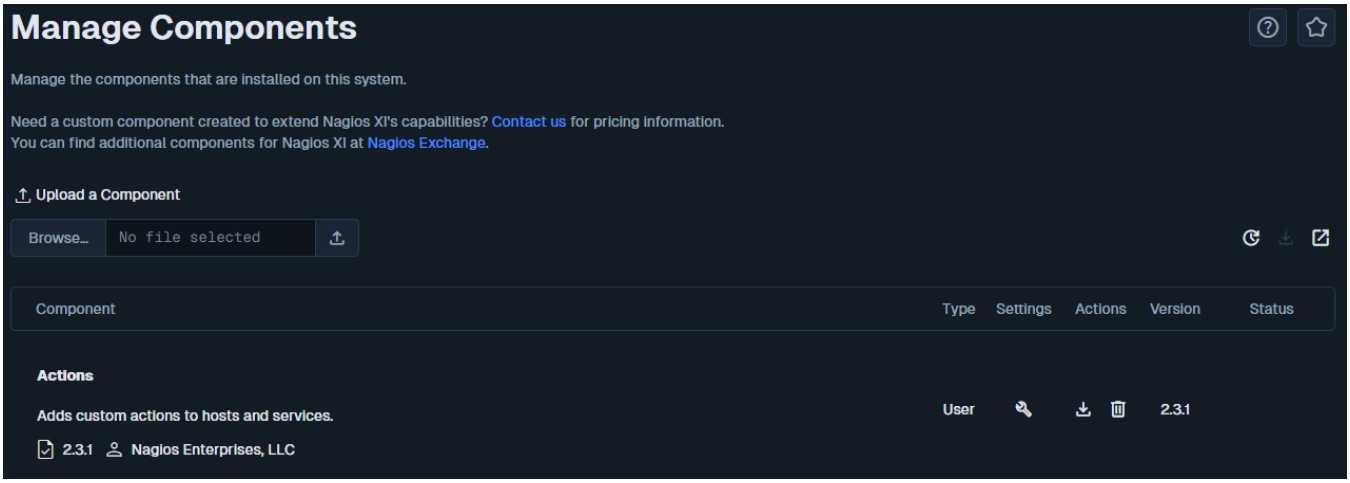
They can be as simple as opening a specific URL or as complex as passing specially formatted macros to a shell script to complete a task in bash.

The action can be configured to apply to a select number of hosts or services, specifying objects by their name, group, and/or through regular expressions.

The component also includes the ability of evaluating a block of PHP code to further limit the objects potentially affected by the action. This allows very complex sets of logic to apply to the link. This is one of the most powerful components in Nagios XI and should be deployed with care.

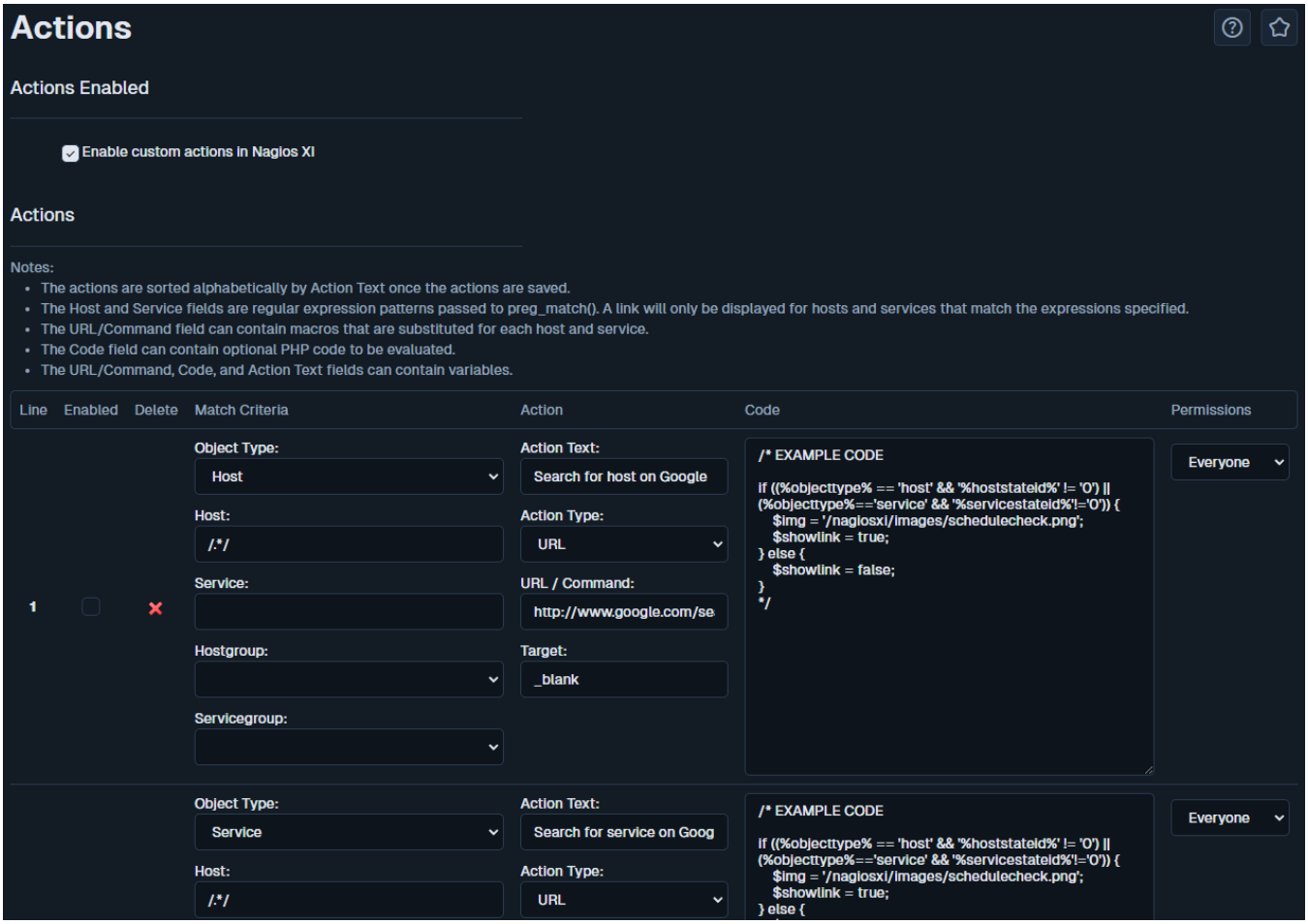
Configuring The Action Component

1. The Action component is accessible from **Admin** > **System Extensions** > **Manage Components.**
2. Configure the Action component settings by clicking the **wrench and screwdriver** icon under **Settings**.



3. This will open the Action component setting page which allows you to create **quick action** links. Make sure the **Enable Component** checkbox at the top is enabled.

4. Each quick action link you create has an **Enabled** checkbox. **The Match Criteria** is used to determine which host/services will have this quick action link on their detail page. The **Action** section identifies what action is run when the quick action link is selected. The **Code** section allows for more complex logic to limit when the quick action link is enabled. The **Permissions** allow you to limit the type of users that will be able to see and use the action.

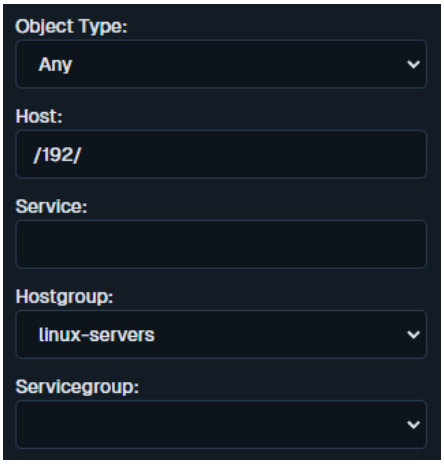


Each section of the **Action** component is explained in further detail in the following sections.

Match Criteria

The **Match Criteria** section allows you to specify which objects will receive the quick action links on their detail pages. You can limit the effected objects by object type and group type. Additionally, the **Host** and **Service** fields are for regular expression patterns to match in the host or service names of objects (you must wrap the regex with forward slashes). These options are additive, so using more than one of the match criteria will only find objects that match all the criteria.

For example, the following criteria will apply the action to all hosts and services that belong to a host with **192** in the hostname field and belong to the `linux-servers` hostgroup:

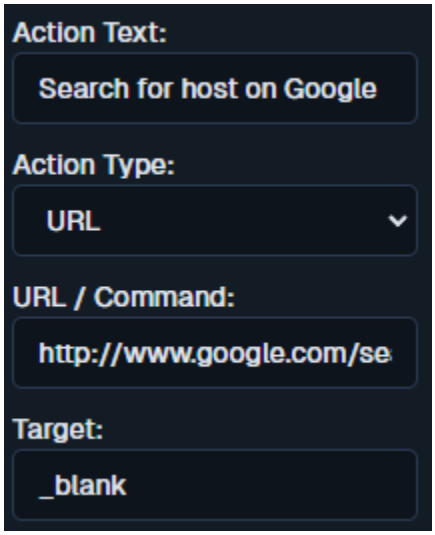


Action

The **Action** section identifies what action is run when the quick action link is selected.

There are 2 types of actions: a URL, or a shell Command. Macros (variables) can be used with either. These actions will be run when the quick action link with the respective **Action Text** label is clicked on the host or service details page for any object that matches the **Match Criteria**.

URL Action Type



This is the most basic portion of the component. When you select an **Action Type** of URL, it enables URL links to be displayed in the quick actions list for the specified hosts/services. You must enter the full URL including `http://` if required for your link.

You can additionally use macros in the URL field. You can see `%host%` is being used in the URL. This will search for the host you are currently viewing.

Here's an example. Let's assume that you have an internal web portal with information about the servers on your network, separated by individual URLs for each server, by the server's hostname in the format of:

```
http://yourwebportal.tld/<hostname>.html
```

`%host%` is the macro which represents the name of the host you will be viewing (**Host Status Details** page). The URL to enter in the URL field would resemble:

```
http://yourwebportal.tld/%host%.html
```

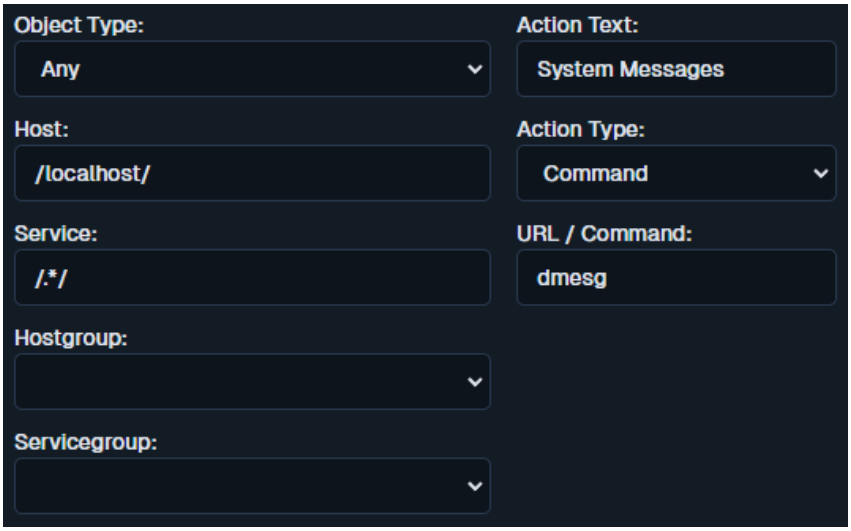
If your server in question had the hostname host1, the URL for the server's quick action would become:

```
http://yourwebportal.tld/host1.html
```

Command Action Type

This is where the real power of the **Actions** component can be found. Using the **Action Type of Command**, you can run any shell command or script from the action link, using macros and variables. If you wish to use pipes "|", percentage symbols "%" (other than macros), or other meta characters in your command, you will have to call a script as the actions component is limited on its escaping abilities.

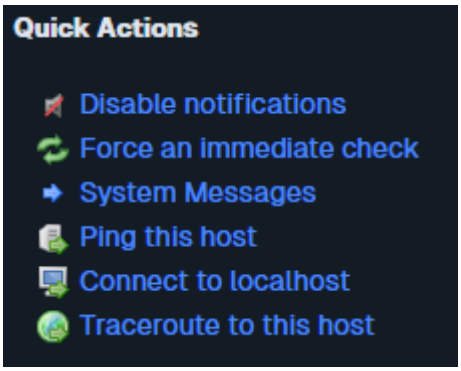
Let's start with a simple command: `dmesg`. As this command is only specific to the Nagios XI server, we will set the criteria to only match the hostname localhost. Create an **Action** with the following values:



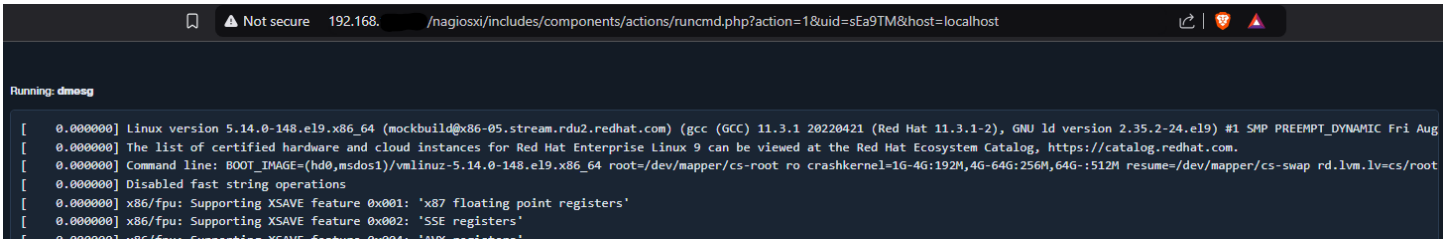
Make sure the **check box for the command** is checked to enable the command and then click the **Apply Settings** button.

Navigate to **Home > Host Status** and select the host localhost.

A new quick action should be available: **System Messages**.



Clicking this link will display the output of `dmesg`.



Here is an example of the output produced by clicking the **System Messages Quick Action**.

The screenshot has been edited to show the **command executed** (top of the screen) and the two buttons **Run Again** and **Close**.

Advanced Command Actions

Most advanced actions will require a script and action/object specific macros.

Adding a Comment to a Host Using the Command Pipe

In the following example, we will create an action that submits a comment for any given host by the user auditor. This action will include the passing of the `%host%` macro to a script that will be written to the Nagios command pipe.

Here is more information about the Nagios command pipe.

Establish a terminal session to your Nagios XI server.

You will need to create a script named `security_audit_completed.sh` in `/usr/local/nagios/libexec/`. The following command does that by opening a new file in vi:

```
vi /usr/local/nagios/libexec/security_audit_completed.sh
```

Paste the following code into the file:

```
#!/bin/bash
HOST=$1
/usr/bin/printf "[%lu] ADD_HOST_COMMENT;$HOST;1;auditor;This host has passed security audit\n"
`date +%s` > /usr/local/nagios/var/rw/nagios.cmd
```

Note: The last two lines above are one long line, the security audit `\n"` line continues immediately after has passed with a space separating the two. Save the changes, you have finished editing this file.

Make the script executable with this command:

```
chmod +x /usr/local/nagios/libexec/security_audit_completed.sh
```

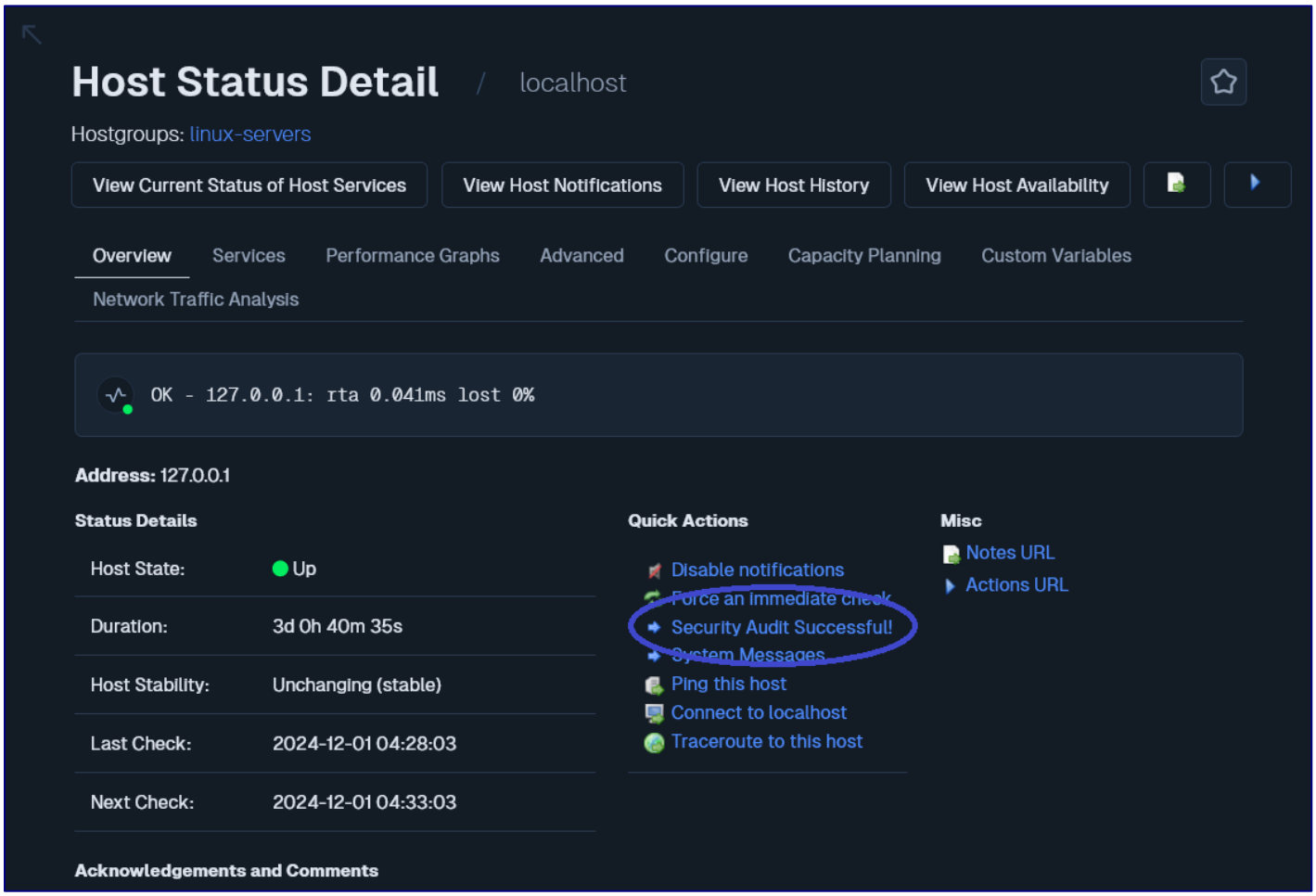
The script you just created is passed one macro, the hostname using `%host%`. The bash script receives this as `$1`.

Now the script has been created you need to define the action in Nagios XI. Return to the Actions Component to create an action. This action will apply to every host, so you use the regex `/.*/` in the **Host** field. The full details are:

After populating the action with the settings above click **Apply Settings**.

Now navigate to **Home > Host Detail** and select any of your hosts.

Click the **Security Audit Successful** link and a pop-up will appear informing you that the command that was run.



You will see a new comment will be added to the **Host Status Detail** page after a moment or two declaring that the security audit was indeed successful.

Sending an Email (Potentially to a Ticketing System)

You will need to create a script named `notify_host_action.sh` in `/usr/local/nagios/libexec/`. The following command does that by opening a new file in `vi`:

```
vi /usr/local/nagios/libexec/notify_host_action.sh
```

Paste the following code into the file:

```
#!/bin/bash
DATE=$(date)
HOST=$1
HOSTSTATE=$2
HOSTADDRESS=$3
# Set your ticketing system's email below
EMAIL=email@domain.tld
/usr/bin/printf "%b" "***** Nagios Monitor XI Alert *****\n\nHost:
$HOST\nState: $HOSTSTATE\nAddress: $HOSTADDRESS\n\nDate/Time: $DATE\n" |
/bin/mail -s "** Host Alert: $HOST is $HOSTSTATE **" $EMAIL
```

Note: The last three lines above are one long line. You will want to change the email address in the script to the email of your company. Save the changes, you have finished editing this file.

Make the script executable with this command:

```
chmod +x /usr/local/nagios/libexec/notify_host_action.sh
```

Return to the **Actions Component** to create an action. This action to apply to every host, so you use the regex /.*/ in the Host field. The above script requires Nagios XI to send it values, these are the macros in the command below and need to be in a specific order, so they match up with the bash script. The full details are:

```
Object Type:    Host
Host:      /.*/
Action Type:    Command
Command:      /usr/local/nagios/libexec/notify_host_action.sh "%host%" %hoststatetype% %hostaddress%
Action Text: Send Email
```

After populating the action with the settings above click **Apply Settings**. Now navigate to **Home > Host Detail** for any of your hosts. Click the **Send Email** link and you should now receive an email sent from the Nagios XI system.

Code

In the **Action** component the **Code** field is by far the most advanced portion of the **Actions** component. Its original purpose was to further limit the objects that will display the action, but you may discover other use cases. The code must be valid PHP code.

The default code is commented out, the characters that begin and end the commenting are `/*` and `*/`.

For example:

```
    /* This sentence could be typed in the Code field and because they are
  between the comment markers the text is ignored */
```

The code field cannot be left empty, so if you want nothing in the field simply populate it with this.

```
    /**/
```

If you remove the comment markers from the default code, it will cause the quick action link to only be displayed if the object in question was not in an OK or UP state AND it will also define the icon image for the link. Below is the default code in an easy-to-read format:

```
    If ((%objecttype%=='host' && '%hoststateid%'!='0') ||
  (%objecttype%=='service' && '%servicestateid%'!='0')) {
         $img='/nagiosxi/images/schedulecheck.png';
         $showlink=true;
    } else {
         $showlink=false;
    }
```

You can use any of the available macros in the Available Macros section of this document to further extend the logic to meet your needs.

Permissions

You can also define the type of users that you want the action to be available for.

- **Everyone**: All users get the action link
- **Admin & Users (No Read Only)**: All users EXCEPT read only users get the action link
- **Admin Only**: Admins get the action link
- **Custom: Select the specific Nagios XI users that get the action link.**

Available Macros

The **Action Component** uses its own specifically defined macros (not the standard Nagios macros). Each list is specific to the object you are running the action from, even though there is some overlap of available macros. If you configure an action for a group of host objects, only the **HOST MACROS** should be used, the same applies to service objects and the **SERVICE MACROS** below.

As previously mentioned in this document, all macros in the Actions component must be wrapped with percentage symbols (%), i.e. `%macro%`. Additionally, any macros that have a chance of containing a space should be wrapped in double quotes, i.e. `"%macro%"`, this usually includes the name of an object or return strings and text.

Below is a comprehensive list of all available macros for the component:

HOST MACROS:

host	hostperfdata	hostdowntime
hostname	hostchecktype	hostlatency
hostaddress	hostactivechecks	hostexectime
hostid	hostpassivechecks	hostlastcheck
hostdisplayname	hostnotifications	hostnextcheck
hostalias	hostacknowledged	hosthasbeenchecked
hoststateid	hosteventhandler	hostshouldbescheduled
hoststatetype	hostflapdetection	hostcurrentattempt
hoststatustext	hostisflapping	hostmaxattempts
hoststatustextlong	hostpercentstatechange	

SERVICE MACROS:

service	servicelatency	hostchecktype
servicename	serviceexectime	hostactivechecks
serviced	servicelastcheck	hostpassivechecks
servicedisplayname	servicenextcheck	hostnotifications
servicestateid	servicehasbeenchecked	hostacknowledged
servicestatetype	serviceshouldbescheduled	hosteventhandler
servicestatustext	servicecurrentattempt	hostflapdetection
servicestatustextlong	servicemaxattempts	hostisflapping
serviceperfdata	host	hostpercentstatechange
hostchecktype	hostname	hostdowntime
serviceactivechecks	hostaddress	hostlatency
servicepassivechecks	hostid	hostexectime
servicenotifications	hostdisplayname	hostlastcheck
serviceacknowledged	hostalias	hostnextcheck
serviceeventhandler	hoststateid	hosthasbeenchecked
serviceflapdetection	hoststatetype	hostshouldbescheduled
serviceisflapping	servicestatustext	hostcurrentattempt
servicepercentstatechange	servicestatustextlong	hostmaxattempts
servicedowntime	hostperfdata	

Finishing Up

This completes the documentation on how to use the Nagios XI actions component. 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