

# Nagios XI - Using The Negate Plugin In Nagios XI 5

## Purpose

This document describes how to use the Negate Plugin in Nagios XI. The Negate plugin allows for any standard plugin output to be reversed and is very useful with hosts or services that are expected to be in a Critical or Warning state but you wish to show them as OK. This function can be used for the opposite effect (i.e. showing a CRITICAL state when the actual state is OK).

If you are using the Negate Plugin in Nagios XI 2024, see [How To Use the Negate Plugin in Nagios XI 2024](#)

## What Is the Negate Plugin?

Negate is used to execute other plugins, the state returned by the other plugin can be changed by the negate plugin. For example when a check is normally considered to be in a Critical or Warning state, but the system administrator would instead prefer to see an OK when in such a state. Understand that this will not turn any check consistently to an OK state, but will reverse a critical to OK or an OK to critical, depending on the actual check being run.

## Negate Plugin Example

For our example we will use a service check for Port 4 Status on a network switch. Below is the service check for Port 4 Status which is currently not being used and is in a Critical state.

Host	Service	Status	Duration	Attempt	Last Check	Status Information
switch01	Port 4 Status	Critical	5m 43s	5/5	2017-01-27 13:54:36	CRITICAL: Interface tengigabitethernet1/0/4 (index 4) is down.
	Port 5 Status	Ok	4m 53s	1/5	2017-01-27 13:51:28	OK: Interface tengigabitethernet1/0/5 (index 5) is up.

Before showing you how to use the negate plugin with this service, lets understand how the **Port 4 Status** service works at the command line. To do that we need to view the service definition in Core Config Manager (CCM).

# Nagios XI - Using The Negate Plugin In Nagios XI 5

Navigate to Configure > Core Config Manager > Monitoring > Services and locate the Port 4 Status check.

Click the **modify** icon to view the service configuration.

Service Name	Service Description	Active	Status	Actions	ID
<input type="checkbox"/>	switch01	Yes	Applied		879
<input type="checkbox"/>	switch01	Yes	Applied		878

Notice the **Check command** drop down has `xi_service_check_ifoperstatus` selected.

The **Command View** field shows what arguments are used for this command.

The `$ARGx$` fields are the values being used for the command. When Nagios XI executes this command it replaces the `$variables$` with actual values, which results in something like:

```
$USER1$/check_ifoperstatus -H $HOSTADDRESS$ -C $ARG1$ -k $ARG2$ $ARG3$
```

\$ARG1\$ box293

\$ARG2\$ 4

\$ARG3\$ -v 2 -p 161

```
/usr/local/nagios/libexec/check_ifoperstatus -H 10.25.4.3 -C box293 -k 4 -v 2 -p 161
```

When this is executed at the command line, the output is:

```
[root@xitest ~]# /usr/local/nagios/libexec/check_ifoperstatus -H 10.25.4.3 -C box293 -k 4 -v 2 -p 161
CRITICAL: Interface tengigabitethernet1/0/4 (index 4) is down.
[root@xitest ~]# echo $?
2
```

You'll notice the first line of output is the `CRITICAL: Interface tengigabitethernet1/0/4 (index 4) is down.`

This is only for us humans to understand what the result of the plugin was.

The second line `echo $?` is telling us what the exit code of the plugin was, which is the value 2. The exit code is what tells Nagios XI that the service is in a critical state.

# Nagios XI - Using The Negate Plugin In Nagios XI 5

Now lets execute that same command again, but this time use the negate command to turn that 2 state into a 0 state.

The command below is one long command, it is just wrapped over two lines:

```
/usr/local/nagios/libexec/negate /usr/local/nagios/libexec/check_ifoperstatus -H 10.25.4.3 -C box293 -k 4 -v \
2 -p 161
```

```
[root@xitest ~]# /usr/local/nagios/libexec/negate /usr/local/nagios/libexec/check_ifoperstatus -H 10.25.4.3 -C box293
-k 4 -v 2 -p 161
CRITICAL: Interface tengigabitethernet1/0/4 (index 4) is down.
[root@xitest ~]# echo $?
0
```

The exit state returned by the plugin is what tells Nagios XI that it's in an OK state, because it's a 0.

You'll notice that the text is still saying **CRITICAL**, this doesn't affect Nagios but it can be confusing for us humans. There is an additional argument **-s** that will substitute the output text as well:

```
/usr/local/nagios/libexec/negate -s /usr/local/nagios/libexec/check_ifoperstatus -H 10.25.4.3 -C box293 -k 4 \
-v 2 -p 161
```

```
[root@xitest ~]# /usr/local/nagios/libexec/negate -s /usr/local/nagios/libexec/check_ifoperstatus -H 10.25.4.3 -C box
293 -k 4 -v 2 -p 161
OK: Interface tengigabitethernet1/0/4 (index 4) is down.
[root@xitest ~]# echo $?
0
```

Now you can see the text output says **OK**.

## Update Service To Use Negate

Now that you have tested negate from the command line and know how it works you can now implement it in your service definition.

Looking at the original service definition, the command used is **xi\_service\_check\_ifoperstatus**, and the command definition for this is:

```
$USER1$/check_ifoperstatus -H $HOSTADDRESS$ -C $ARG1$ -k $ARG2$ $ARG3$
```

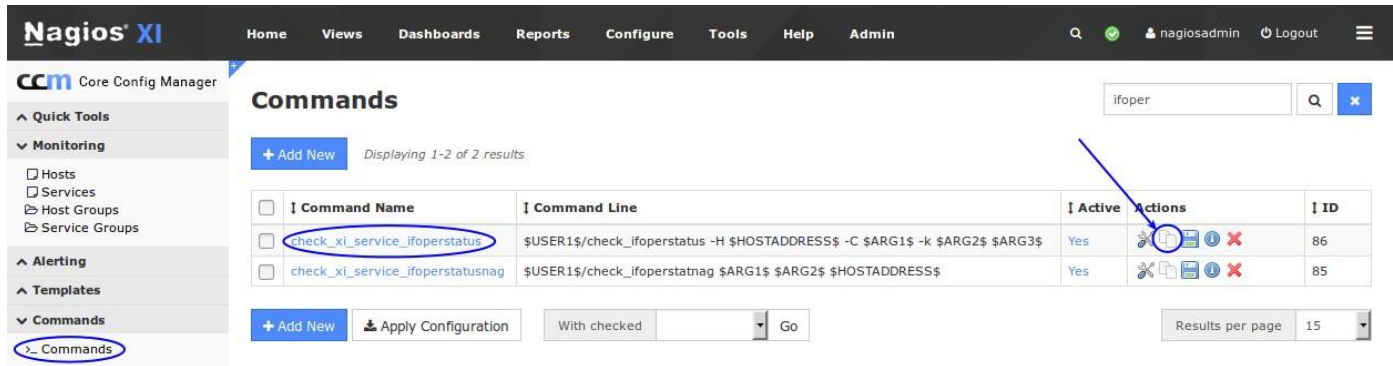
All that is required is to put the negate command in front of this like so:

```
$USER1$/negate -s $USER1$/check_ifoperstatus -H $HOSTADDRESS$ -C $ARG1$ -k $ARG2$ $ARG3$
```







# Nagios XI - Using The Negate Plugin In Nagios XI 5

However you shouldn't change the original `xi_service_check_ifoperstatus` command definition as it'll affect all services, instead you can copy the existing command to create a new command. Navigate to **Configure > Core Config Manager > Commands > >\_Commands**.

Locate the `xi_service_check_ifoperstatus` command and click the **copy** icon.

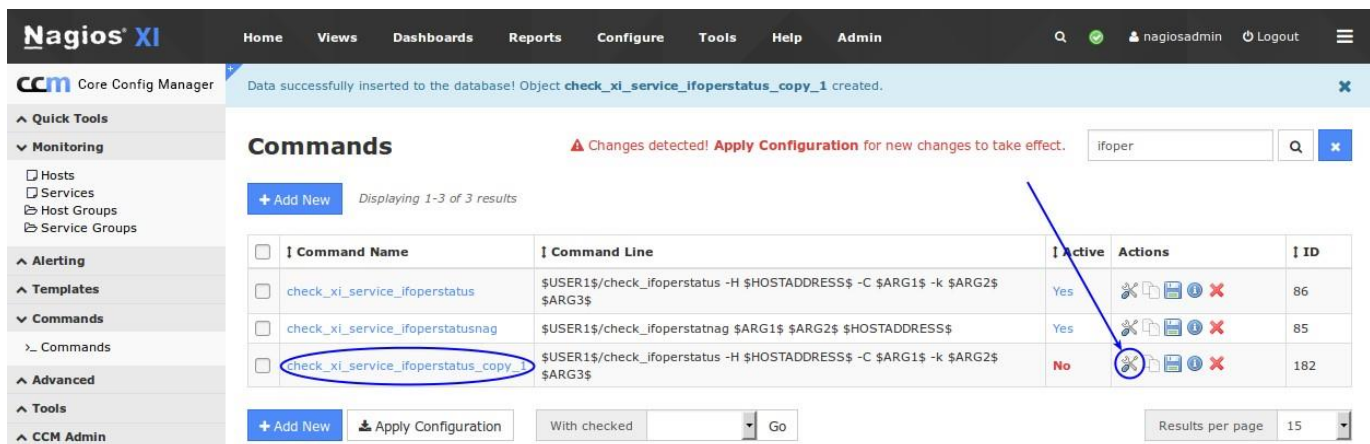


The screenshot shows the Nagios XI Core Config Manager interface. The 'Commands' section is active, displaying a table of commands. The command `check_xi_service_ifoperstatus` is highlighted with a blue circle. A blue arrow points to the 'copy' icon in the 'Actions' column of this row. The table has columns for 'Command Name', 'Command Line', 'Active', 'Actions', and 'ID'.










<input type="checkbox"/>	Command Name	Command Line	Active	Actions	ID
<input type="checkbox"/>	check_xi_service_ifoperstatus	\$USER1\$/check_ifoperstatus -H \$HOSTADDRESS\$ -C \$ARG1\$ -k \$ARG2\$ \$ARG3\$	Yes	  	86
<input type="checkbox"/>	check_xi_service_ifoperstatusnag	\$USER1\$/check_ifoperstatnag \$ARG1\$ \$ARG2\$ \$HOSTADDRESS\$	Yes	  	85

When the screen refreshes you'll have a duplicate command appended with `_copy_1`.

Click the **modify** icon to edit this command.



The screenshot shows the Nagios XI Core Config Manager interface after a new command has been created. A notification at the top states: 'Data successfully inserted to the database! Object check\_xi\_service\_ifoperstatus\_copy\_1 created.' The 'Commands' section now displays three commands. The command `check_xi_service_ifoperstatus_copy_1` is highlighted with a blue circle. A blue arrow points to the 'modify' icon in the 'Actions' column of this row. The table has columns for 'Command Name', 'Command Line', 'Active', 'Actions', and 'ID'.

<input type="checkbox"/>	Command Name	Command Line	Active	Actions	ID
<input type="checkbox"/>	check_xi_service_ifoperstatus	\$USER1\$/check_ifoperstatus -H \$HOSTADDRESS\$ -C \$ARG1\$ -k \$ARG2\$ \$ARG3\$	Yes	  	86
<input type="checkbox"/>	check_xi_service_ifoperstatusnag	\$USER1\$/check_ifoperstatnag \$ARG1\$ \$ARG2\$ \$HOSTADDRESS\$	Yes	  	85
<input type="checkbox"/>	check_xi_service_ifoperstatus_copy_1	\$USER1\$/check_ifoperstatus -H \$HOSTADDRESS\$ -C \$ARG1\$ -k \$ARG2\$ \$ARG3\$	No	  	182

# Nagios XI - Using The Negate Plugin In Nagios XI 5

You will need to give the command a new name. In this example it seems logical to append the name with **\_negate** so the command name is **xi\_service\_check\_ifoperstatus\_negate**.

Then you need to add the negate command (`$USER1$/negate -s`) to the beginning of the command line.

Lastly you need to click the **Active** checkbox.

Click the **Save** button.

Host	Service	Status	Duration	Attempt	Last Check	Status Information
switch01	Port 4 Status	Ok	9s	1/5	2017-01-27 15:19:34	OK: Interface tengigabitethernet1/0/4 (index 4) is down.
	Port 5 Status	Ok	1h 28m 15s	1/5	2017-01-27 15:15:52	OK: Interface tengigabitethernet1/0/5 (index 5) is up.

This screenshot shows the required changes.

The last step is to update the the Port 4 Status service with the new check command. Navigate to **Configure > Core Config Manager > Monitoring > Services** and edit the **Port 4 Status** check.

Use the **Check command** drop down list to select the new command **xi\_service\_check\_ifoperstatus\_negate**.

Once selected you'll see the **Command view** update, it shows the negate command being used.

Click **Save** button and then **Apply Configuration**.

After the configuration is applied and the **Port 4 Status** service is checked, the service will be in an OK state:

You can see that the **Port 4 Status** service check is in an OK state and the status information shows that the port is down.

### Command Management

**⚠ This object is currently set as **Inactive** and will not be written to the configuration files.**

**Command Name \***  
check\_xi\_service\_ifoperstatus\_negate  
Example: check\_example

**Command Line \***  
\$USER1\$/negate -s \$USER1\$/check\_ifoperstatus -H \$HOSTADDRESS\$ -C \$ARG1\$ -k \$ARG2\$ \$ARG3\$  
Example: \$USER1\$/check\_example -H \$HOSTADDRESS\$ -P \$ARG1\$ \$ARG2\$

**Command Type:**  
check command

**Active** ⓘ

**Available Plugins**  
 ⓘ

**Save** **Cancel**

**Check command**  
check\_xi\_service\_ifoperstatus\_negate

**Command view**  
\$USER1\$/negate -s \$USER1\$/check\_ifoperstatus -H \$HOSTADDRESS\$ -C \$ARG1\$ -k \$ARG2\$ \$ARG3\$

\$ARG1\$ box293

\$ARG2\$ 4

\$ARG3\$ -v 2 -p 161

# Nagios XI - Using The Negate Plugin In Nagios XI 5

## Finishing Up

This completes the documentation on using the Negate Plugin 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](#)