

# Nagios XI - Mod-Gearman Queues and Workers

Article Number: 484 | Rating: 4/5 from 5 votes | Last Updated: Mon, Mar 11, 2019 at 6:58 PM

## Overview

The purpose of this article is to explain how queues work in Mod-Gearman (MG). Having a clear understanding of this will allow you to distribute the execution of checks to specific workers.

## Editing Files

In many steps of this article 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

## The Basics

When you install the MG server broker module on your XI server, by default this also installs a worker module. At this point the checks being handed over to MG to do the executing instead of the Nagios Core engine.

A MG worker is what executes the check from Nagios, like ping, CPU load, memory usage etc. A worker can be located on the Nagios XI server or it can be on an external server. Being on an external server means the load is taken away from the Nagios XI server.

Queues in MG are how checks are handed out to the workers.

With a default configuration, when nagios starts it hands off the host and service checks to MG.

MG creates two queues called `host` and `service`. You can think of these queues as the default "catch all" queues.

By default, any worker that connects will execute checks from these queues.

It is important to remember that an external worker needs all of the plugins installed on it so it can execute the checks that are handed to it.

## Configuration File Location

MG has two separate configuration files:

### **module.conf**

This is the configuration file for the Nagios server.

### **worker.conf**

This is the configuration file for the worker. The worker exists on the Nagios server as well as your remote worker(s).

Both of these files exist in the same location, however this location is different depending on your operating system:

#### **RHEL | CentOS | Oracle Linux**

```
/etc/mod_gearman/
```

#### **Debian | Ubuntu**

```
/etc/mod-gearman/
```

## Remote Worker Considerations

---

When you add a remote worker, there are some things that need to be taken into account. The most important question you need to ask is:

"What checks should NOT be executed by the remote worker?"

Why do you need to ask such a question?

Lets look at the standard checks that the Nagios XI server has built in for the `localhost` object:

- Current Load
- Current Users
- HTTP
- PING
- Root Partition
- SSH
- Swap Usage
- Total Processes

Let's look at the service "Root Partition", the command it executes is:

```
/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /
```

If a **remote worker** was to execute this check, the results that came back would be for the remote workers root partition, not the root partition of the Nagios XI server.

So it's pretty obvious that you don't want these checks being executed on the remote workers, how do you configure MG to do this? The key to this is with host groups and service groups.

One other important point to be made about remote workers is that the plugins need to be installed on the workers. MG is passing the command that needs to be executed, the worker executes the command so it needs to be able to execute the plugin.

## Host groups and Service groups

---

The first step to stop MG from sending local checks to remote workers is to create a host group or service group in Nagios that contains the objects you don't want executed on remote workers.

What's the difference between a host group and a service group in MG?

- host group
  - If you use a host group in a MG configuration, MG will automatically include the services for the hosts in the host group
  - This allows for simple configurations
- service group
  - Using a service group in a MG configuration allows for more granular control of what services are handled by MG

For the purpose of simplicity, we will focus on a host group.

- Log into Nagios XI and open Core Configuration Manager (CCM)
- Under **Monitoring** click **Host Groups**
- Click the **+ Add New** button
  - Name: **mg\_objects\_local**
  - Description: **Mod-Gearman Objects - Local**
  - Click the **Manage Hosts** button
    - Add **localhost** to the *right hand* side
    - Click **Close**

- Click **Save**
- Click the **Apply Configuration** button

Now we need to configure MG to use this host group to exclude checks. On your Nagios XI server edit the file `module.conf`

Find the section like this:

```
# sets a list of hostgroups which will not be executed
# by gearman. They are just passed through.
# Default is none
localhostgroups=
```

Update the line as follows:

```
localhostgroups=mg_objects_local
```

Save the `module.conf` file.

Now you need to restart some services:

#### RHEL 6 | CentOS 6 | Oracle Linux 6

```
service nagios stop
service gearmand restart
service nagios start
```

#### RHEL 7 | CentOS 7 | Oracle Linux 7

```
systemctl stop nagios.service
systemctl restart gearmand.service
systemctl start nagios.service
```

#### Debian | Ubuntu

```
systemctl stop nagios.service
systemctl restart gearman-job-server.service
systemctl start nagios.service
```

From now on, MG will not execute any host or service checks for the hosts in the `mg_objects_local` group, instead Nagios Core will execute them as it normally would.

What other checks should be prevented from executing on a remote worker?

Any devices that are being monitored by the "Switch / Router Wizard" will also need to be added to the `mg_objects_local` group, as they target some files specific to MRTG which is run locally on the Nagios XI server. Once you add these hosts to the `mg_objects_local` group, when you Apply Configuration Nagios will restart and MG will then know about the updated host group.

## Using A Worker For A Specific Site

---

In this scenario, you have a remote site where you want all the checks to be executed at the remote site by the worker at the remote site.

In this example, we will focus on a host group.

- Log into Nagios XI and open Core Configuration Manager (CCM)
- Under **Monitoring** click **Host Groups**
- Click the **+ Add New** button
  - Name: `mg_objects_site_a`

- Description: **Mod-Gearman Objects - Site A**
- Click the **Manage Hosts** button
  - Add <all the hosts at the remote site> to the *right hand* side
  - Click **Close**
- Click **Save**
- Click the **Apply Configuration** button

Before we configure the MG server to use this host group as a queue, we'll configure the remote worker.

Most importantly, we **ONLY** want this worker to execute checks for the hosts in the host group. With that in mind, we'll configure the worker so it doesn't touch the default `host` and `services` queues.

On your remote worker edit the file `worker.conf`

Find the sections like this:

```
# defines if the worker should execute
# service checks.
services=yes

# defines if the worker should execute
# host checks.
hosts=yes
```

Change them to **no** as follows:

```
# defines if the worker should execute
# service checks.
services=no

# defines if the worker should execute
# host checks.
hosts=no
```

Find the sections like this:

```
# sets a list of hostgroups which this worker will work
# on. Either specify a comma seperated list or use
# multiple lines.
#hostgroups=name1
#hostgroups=name2,name3
```

Add this line as follows:

```
hostgroups=mg_objects_site_a
```

Save the `worker.conf` file.

You need to restart the worker service:

**RHEL 6 | CentOS 6 | Oracle Linux 6**

```
service mod-gearman-worker restart
```

**RHEL 7 | CentOS 7 | Oracle Linux 7 | Debian | Ubuntu**

```
systemctl restart mod-gearman-worker.service
```

Now we need to configure the MG server to use this host group as a queue. On your Nagios XT server edit the file

Now we need to configure the XI server to use this hostgroup as a queue. On your Nagios XI server edit the file `/etc/mod_gearman2/module.conf`

Find the section like this:

```
# sets a list of hostgroups which will go into separate
# queues. Either specify a comma separated list or use
# multiple lines.
#hostgroups=name1
#hostgroups=name2,name3
```

Add this line as follows:

```
hostgroups=mg_objects_site_a
```

Save the `module.conf` file.

Now you need to restart some services:

#### RHEL 6 | CentOS 6 | Oracle Linux 6

```
service nagios stop
service gearmand restart
service nagios start
```

#### RHEL 7 | CentOS 7 | Oracle Linux 7

```
systemctl stop nagios.service
systemctl restart gearmand.service
systemctl start nagios.service
```

#### Debian | Ubuntu

```
systemctl stop nagios.service
systemctl restart gearman-job-server.service
systemctl start nagios.service
```

From now on, MG will allocate any host or service checks for the hosts in the `mg_objects_site_a` group into a queue. Any workers that are configured to target it will execute them.

You can see this new queue by using the `gearman_top` command on the XI server:

```
gearman_top
```

It will look like the screenshot below:

```
2016-03-07 14:21:56 - localhost:4730 - v0.33
```

| Queue Name                     | Worker Available | Jobs Waiting | Jobs Running |
|--------------------------------|------------------|--------------|--------------|
| check_results                  | 1                | 0            | 0            |
| eventhandler                   | 10               | 0            | 0            |
| host                           | 5                | 0            | 0            |
| hostgroup_mg_objects_site_a    | 5                | 0            | 0            |
| service                        | 5                | 0            | 0            |
| worker_localhost.localdomain   | 1                | 0            | 0            |
| worker_xi-c6x-x64.box293.local | 1                | 0            | 0            |

## Using A Worker For A Specific Monitoring Plugin

---

Some monitoring plugins have the following requirements and behaviours:

- Specific modules installed
  - The more "common" workers you deploy require you to install these modules on all the workers
- Create temporary files that are accessed the next time the plugin is executed
  - These files are used for comparing the last value to the current value
  - If the check is being shifted from worker to worker, these files are not going to have valid data and cause monitoring inconsistencies

A good example of such a plugin is `check_wmi_plus.pl` which can be used to perform agent-less monitoring on Windows machines.

In this scenario we will create a service group that contains all the services that use the `check_wmi_plus.pl` plugin. We'll configure a worker to use this service group.

- Log into Nagios XI and open Core Configuration Manager (CCM)
- Under **Monitoring** click **Service Groups**
- Click the **+ Add New** button
  - Name: **mg\_objects\_wmi\_services**
  - Description: **Mod-Gearman Objects - WMI Services**
  - Click the **Manage Services** button
    - Add **<all the services that use the check\_wmi\_plus.pl plugin>** to the *right hand* side
    - Click **Close**
  - Click **Save**
- Click the **Apply Configuration** button

Before we configure the MG server to use this host group as a queue, we'll configure the remote worker.

Most importantly, we **ONLY** want this worker to execute checks for the services in the service group. With that in mind, we'll configure the worker so it doesn't touch the default `host` and `services` queues (*however there is no reason why this worker couldn't also do other service checks*).

On your remote worker edit the file `worker.conf`

Find the sections like this:

```
# defines if the worker should execute
# service checks.
services=yes

# defines if the worker should execute
# host checks.
hosts=yes
```

Change them to **no** as follows:

```
# defines if the worker should execute
# service checks.
services=no

# defines if the worker should execute
# host checks.
hosts=no
```

Find the sections like this:

```
# sets a list of servicegroups which this worker will
# work on.
#servicegroups=name1,name2,name3
```

Add this line as follows:

```
servicegroups=mg_objects_wmi_services
```

Save the `worker.conf` file.

You need to restart the worker service:

#### RHEL 6 | CentOS 6 | Oracle Linux 6

```
service mod-gearman-worker restart
```

#### RHEL 7 | CentOS 7 | Oracle Linux 7 | Debian | Ubuntu

```
systemctl restart mod-gearman-worker.service
```

Now we need to configure the MG server to use this service group as a queue. On your Nagios XI server edit the file `module.conf`

Find the section like this:

```
# sets a list of servicegroups which will go into separate
# queues.
#servicegroups=name1,name2,name3
```

Add this line as follows:

```
servicegroups=mg_objects_wmi_services
```

Save the `module.conf` file.

Now you need to restart some services:

#### RHEL 6 | CentOS 6 | Oracle Linux 6

```
service nagios stop
service gearmand restart
service nagios start
```

#### RHEL 7 | CentOS 7 | Oracle Linux 7

```
systemctl stop nagios.service
systemctl restart gearmand.service
systemctl start nagios.service
```

#### Debian | Ubuntu

```
systemctl stop nagios.service
systemctl restart gearman-job-server.service
systemctl start nagios.service
```

From now on, MG will allocate any service checks in the `mg_objects_wmi_services` group into a queue. Any workers that are configured to target it will execute them.

You can see this new queue by using the `gearman_top` command on the XI server:

```
gearman_top
```

It will look like the screenshot below:

2016-03-07 14:45:42 - localhost:4730 - v0.33

| Queue Name                           | Worker Available | Jobs Waiting | Jobs Running |
|--------------------------------------|------------------|--------------|--------------|
| check_results                        | 1                | 0            | 0            |
| eventhandler                         | 10               | 0            | 0            |
| host                                 | 5                | 0            | 0            |
| service                              | 5                | 0            | 0            |
| servicegroup_mg_objects_wmi_services | 5                | 0            | 0            |
| worker_localhost.localdomain         | 1                | 0            | 0            |
| worker_xi-c6x-x64.box293.local       | 1                | 0            | 0            |

## Ensure Worker Doesn't Touch Host And Service Queues

This was briefly mentioned in the "Using A Worker For A Specific Site" scenario, however it is worth re-iterating.

The goal is that you ONLY want the worker to execute checks in the specific queues that it has been configured for, using the `hostgroups` and/or `servicegroups` directives. You DO NOT want it executing checks in the default `host` and `service` queues.

Why would you want this type of configuration?

Let's say you have a remote worker that is executing checks for the devices located at that physically remote location. The local worker on the Nagios XI server is executing the `host` and `service` queues. If the remote worker was executing checks for the `host` and `service` queues, the following would happen:

- Remote worker gets plugin command to execute and executes it
- The host address is actually for a device back at the Nagios XI server location
- Network traffic is generated to go back across to the Nagios XI server location to perform the plugin check against the host
- Network traffic is generated to return the plugin result to the remote worker
- Network traffic is generated to from the remote worker to return the plugin result back to the MG server on the Nagios XI server

You can see that there is a lot of unnecessary traffic being generated across network links. Hence it is important when have remote workers at physically different locations, they need to be correctly configured so they don't touch the `host` and `service` queues.

On your worker edit the file `/etc/mod_gearman2/worker.conf`

Find the sections like this:

```
# defines if the worker should execute
# service checks.
services=yes

# defines if the worker should execute
# host checks.
hosts=yes
```

Change them to **no** as follows:

```
# defines if the worker should execute
# service checks.
services=no

# defines if the worker should execute
# host checks.
hosts=no
```

Save the `worker.conf` file.

You need to restart the worker service:



RHEL 6 | CentOS 6 | Oracle Linux 6

```
service mod-gearman-worker restart
```

RHEL 7 | CentOS 7 | Oracle Linux 7 | Debian | Ubuntu

```
systemctl restart mod-gearman-worker.service
```

## Final Thoughts

---

For any support related questions please visit the [Nagios Support Forums](#) at:

<http://support.nagios.com/forum/>

Posted by: **tlea** - Fri, Mar 4, 2016 at 1:34 AM. This article has been viewed 3064 times.

Online URL: <https://support.nagios.com/kb/article/nagios-xi-mod-gearman-queues-and-workers-484.html>