

Nagios XI - Defining Global Environment Variables

Article Number: 489 | Rating: 2/5 from 3 votes | Last Updated: Mon, Mar 11, 2019 at 7:20 PM

Problem Description

When setting up a check in Nagios, you may get a error similar to the following:

```
[Can't locate Monitoring/Plugin.pm in @INC (@INC contains: /usr/local/lib64/perl5 /usr/local/share/perl5 /usr/lib64/perl5/vendor_perl /usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/perl5 .) at /usr/local/nagios/libexec/check_rabbitmq_aliveness 1 BEGIN failed--compilation aborted at /usr/local/nagios/libexec/check_rabbitmq_aliveness line 15.]
```

However when you execute the check in an SSH session as the nagios user, it executes correctly.

Explanation

When you test the plugin in an SSH session, you are using a session that has environment variables loaded from your bash profile. This includes an extended PATH which allows bash to find the require modules.

In some environments, when the plugin is executed by the monitoring engine, these environment variables are not loaded and hence the plugin does not know where to find them and fails.

Define Global Environment Variables

In this solution, you will define variables required for you plugins globally.

Specifically:

- Add the path `/usr/local/important_application` to the PATH environment
- Add the variable `ORACLE_HOME=/usr/lib/oracle/11.2/client64`

This is performed by editing a specific file that nagios checks when it starts:

```
/etc/sysconfig/nagios
```

By default, this file does not exist, but you will create it just by opening vi to the file.

Open an SSH session to your Nagios XI host

Type:

```
vi /etc/sysconfig/nagios
```

Press `i` to start editing the file

Add the following lines to the file:

```
export PATH=$PATH:/usr/local/important_application
export ORACLE_HOME=/usr/lib/oracle/11.2/client64
```

Press `esc` to stop the editing mode

Type:

```
:wq
```

Now restart Nagios:

RHEL 6 | CentOS 6 | Oracle Linux 6 | Ubuntu 14

```
service nagios restart
```

RHEL 7 | CentOS 7 | Oracle Linux 7 | Debian | Ubuntu 16/18

```
systemctl restart nagios.service
```

Mod-Gearman

If you have Mod-Gearman, the following steps will need to be performed on all of your workers (including the XI server if it is a worker).

Note: The method for workers are different depending on the operating system they are running on.

RHEL 6 | CentOS 6 | Oracle Linux 6

The following is being applied:

- Add the path `/usr/local/important_application` to the `PATH` environment
- Add the variable `ORACLE_HOME=/usr/lib/oracle/11.2/client64`

This is performed by editing a specific file that `mod-gearman2-worker` checks when the service starts:

```
/home/nagios/.bashrc
```

Open an SSH session to your Mod-Gearman worker.

Type:

```
vi /home/nagios/.bashrc
```

Press `i` to start editing the file

Add the following lines to the end of the file:

```
export PATH=$PATH:/usr/local/important_application
export ORACLE_HOME=/usr/lib/oracle/11.2/client64
```

Press `esc` to stop the editing mode

Type:

```
:wq
```

Now restart the Mod-Gearman worker:

```
service mod-gearman2-worker restart
```

RHEL 7 | CentOS 7 | Oracle Linux 7 | Debian | Ubuntu 16/18

These operating systems use `systemd` which has a different method, referencing existing environment variables like `$PATH` will not work. If you wish to define the path, you will need to define the entire path variable (which is shown below).

The following is being applied:

- Re-define the `PATH` environment to include `/usr/local/important_application`
- Add the variable `ORACLE_HOME=/usr/lib/oracle/11.2/client64`

This is performed by editing a specific file that `mod-gearman2-worker` checks when the service starts:

```
/etc/sysconfig/mod-gearman2-worker
```

Open an SSH session to your Mod-Gearman worker.

Type:

```
vi /etc/sysconfig/mod-gearman2-worker
```

Press `i` to start editing the file

Add the following lines to the end of the file:

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/local/important_application
ORACLE_HOME=/usr/lib/oracle/11.2/client64
```

Press `esc` to stop the editing mode

Type:

```
:wq
```

Now reload the daemons and restart the Mod-Gearman worker:

```
systemctl daemon-reload
systemctl restart mod-gearman2-worker.service
```

RAM Disk

If you later configure Nagios XI with a RAM Disk using our installation script, the settings you just added will be lost. You will need to follow the steps above (*Define Global Environment Variables*) to re-add the environment variables and it will work again.

[Nagios XI - Utilizing a RAM Disk in Nagios XI](#)

How Can I See The Environment Variables ?

One way to see what environment variables are present when the plugin executes is to create a simple plugin that will output them. From this you create a command and service and then you can see the result in the XI GUI.

Bash script:

```
#!/bin/bash
env | grep -v '|'
exit 0
```

Final Thoughts

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

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

Posted by: **tlea** - Thu, Mar 17, 2016 at 1:51 AM. This article has been viewed 3265 times.

Online URL: <https://support.nagios.com/kb/article/nagios-xi-defining-global-environment-variables-489.html>