NEUTRO Quick Start Guide

Version **2017.2**

Contents

Important Information

Information in this document is subject to change without notice.
Companies, names, and data used in examples herein are fictitious unless otherwise noted.

Who Should Read This Guide

This guide is intended for NICE Neutro administrators wanting a brief introduction to quickly start using NICE Neutro.

Contact Us

Use one of the following to contact NICE technical support.

Email
    <helpdesk@nice-software.com>

World Wide Web
    http://www.nice-software.com

Phone
    +39 0141 901516

Mail
    NICE Support
    c/o NICE s.r.l.
    Via Milliavacca, 9
    14100 Asti
    Italy

When contacting NICE, please include your company's full name.

1

Neutro Terminology

Neutro Master is the daemon serving client requests, collecting information from Neutro cluster hosts and dispatching and managing *Tasks* to the cluster.

Neutro Agent is the daemon (running as Windows Service) running on a Neutro cluster host. It's responsible to serve *Neutro Master* requests and to gather host resource information and send them back to the *Neutro Master*.

Neutro Shell is a simple shell program that allows to check and interact with a Neutro cluster.

Resource is any computational resource, hardware (e.g. CPU, Memory, GPU model, etc.) or software (e.g. the availability of an application, a license token, etc.).

Task is a system computational unit that can be dispatched and executed by a *Neutro Agent*.

Task Definition is the static description of a *Task*. It is composed by the following parts:

* A resource requirements expression.

* A global environment settings, common to all the operations.

* A main operation with options, executed when the *Task* is submitted.

* A set of operations with arguments that can be invoked by the client to interact with the *Task*.

Task Instance represents a live *Task* instance, created when a *Task Definition* is submitted to a host. In other words, it is a running *Task Definition*. A *Task Instance* can be in a well defined state, like `DISPATCHED`, `RUN`, `DONE`, `EXIT`, `SUBMISSION_FAILED`, etc. during its own life cycle. A *Task Instance* also has associated properties, like for example a `submission time`, a `termination time` or an `exit code`.

Task Operation is an action that is possible to execute on a *Task Instance*, during its life cycle. *Task Operations*, including the main Operation, can produce an output that is returned to the client and/or can cause changes to the status of the *Task Instance*.
Standard operations have default implementation in the Neutro system and they could be overridden. Custom operations, instead should be defined into the *Task Definition*.

2

Introducing Neutro

Neutro is a distributed system to dispatch, execute and manage tasks on a Windows cluster.

Neutro manages an elastic infrastructure where hosts can dynamically join and leave the cluster anytime, anywhere. The Neutro infrastructure does not require cluster configuration: new hosts can announce themselves and dynamically join the cluster just knowing the Neutro Master(s) address(es) and/or hostnames.

Neutro is not a job queues management system. *Tasks* will not be queued waiting for the availability of computational resources. Instead, Neutro *Tasks* will be immediately dispatched when the computational resources are available or dropped.

Neutro executes only commands defined in *Task Definitions* stored on the Neutro Master side. It does not execute generic commands provided by client users.

Task execution depends on the availability of a host satisfying the *Resource* requirements expression. This may include constraints on the total memory available, number of CPU cores, GPU model, application availability, etc. Host tagging is also available.

User impersonation is not provided for *Tasks* and *Task operations* execution.

Neutro is primarily aimed to support and satisfy EnginFrame's interactive requirements.
As of today Neutro is not a stand-alone product but a component of the EnginFrame solutions.

3

System Architecture
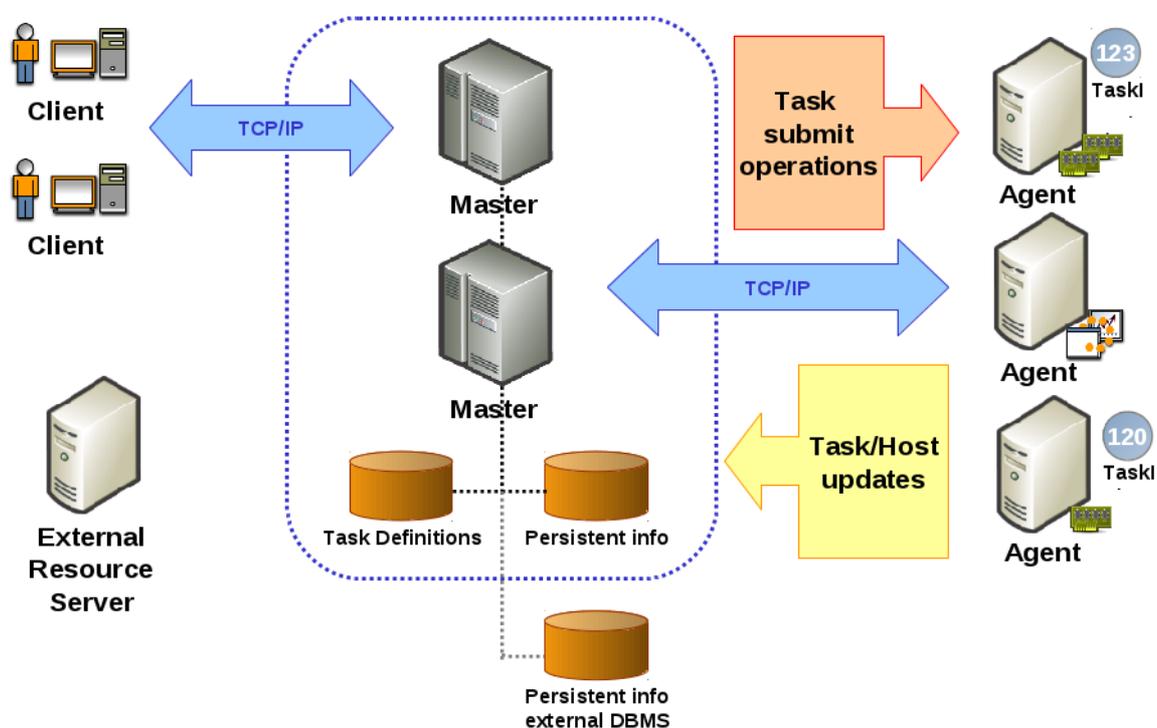
Figure 3.1. Neutro architecture

4

Getting Started

With this guide, you will be able to quickly setup and run a single Neutro Master - multi Neutro Agent cluster.

Downloading Neutro

Neutro packages can be downloaded from:

`http://www.nice-software.com/download/enginframe`

You need a valid account to access the download area. If you do not have one yet, please contact `<helpdesk@nice-software.com>` or your Neutro reseller.

Requirements

Neutro Master Requirements

Table 4.1. Neutro Master Software Requirements

Neutro Master Requirements	
Operating system	RedHat Enterprise Linux 5.x, 6.x 64-bit Amazon Linux release 2016.10 and later
Java Runtime Environment	Oracle JRE 64-bit 7 or 8

The following prerequisites, assuming your system matches one of the supported configurations, have to be met:

1. Select an installation location (*NEUTRO_ROOT* hereafter) with 500 MB free disk space.

> ⭐ **Available space required**
>
> By default, the Neutro Master is configured to use its own DB, so disk space is required accordingly with the amount of stored data.

2. Configure the maximum number of open file descriptors to a value around 10000 or greater. Please run **ulimit -n** to check such limit.

Increase open file limit

The default open file limit per process is 1024, that is usually too low for the Neutro Master process.
In order to raise the maximum number of open file descriptors a process can handle, you can edit the file

```
/etc/security/limits.conf
```

and add or change the line with "nofile" limit in order to set a value around 10000, e.g.

```
* - nofile 10000
```

This limit should be configured at least for the user running the Neutro Master.
Logout and login to affect the change.

---

3. Configure the stack size limit to a value around 10240 or greater. Please run **ulimit -s** to check such limit.

Increase stack size limit

The default stack size limit per process is usually too low for the Neutro Master process.
In order to raise the stack size limit, you can edit the file

```
/etc/security/limits.conf
```

and add or change the line with "stack" limit in order to set a value around 10240, e.g.

```
* - stack 10240
```

This limit should be configured at least for the user running the Neutro Master.
Logout and login to affect the change.

---

4. The following TCP ports should be open in order to let the Neutro Master communicate:

- 5600 (to receive Client connections)

- 5557 (to receive Neutro Agent connections)

- 5700, 9160, 7000, 7199 (internal use)

> **Internal use ports**
>
> The internal use port are used to communicate with others Neutro Master and so they should be opened only in case of a multi Neutro Master setup.

5. A static IP must be assigned to the host running the Neutro Master.

Neutro Agent Requirements

Table 4.2. Neutro Agent Software Requirements

Neutro Agent Requirements	
Operating system	Microsoft Windows XP Microsoft Windows 7 Microsoft Windows 10 Windows Server 2008 R2 Windows Server 2012 Windows Server 2016
Microsoft .Net Framework	Microsoft .Net Framework 4 Full or greater To download the Microsoft .Net Framework 4 Full go to Microsoft download center site at https://www.microsoft.com/download

Neutro Agent uses the following TCP port:

• 20120 (to receive Neutro Master messages)

During the installation, the installer will ask you to modify the firewall rules to open that port.

Installation

Installing the Neutro Master

The Neutro Master package is distributed as a Java installer, so a working Java Runtime Environment (JRE) is needed on the system. This can be installed via software repository, or set up via `JAVA_HOME` environment variable.

For a graphical installation run:

```
${JAVA_HOME}/bin/java -jar nice-neutro-master-installer-201x.x-xxxxx.jar
```

For a text installation run:

```
${JAVA_HOME}/bin/java -jar nice-neutro-master-installer-201x.x-xxxxx.jar
                        --text
```

The installer will ask for the following:

- The installation directory. Make sure you have the permission to write in the selected folder.

- A Java home path, in case you want the Neutro Master to use a Java different from the one used to launch the installer.

- The IP address of the current host.

- The Neutro Master administrator user. This will also be the owner of the tree folder where the Neutro Master is going to be installed.

> ⭐ **Admin user**
>
> Neutro Master doesn't require user root privileges to work, so it's preferable to select a standard user to run the Neutro Master daemon.

- The Neutro cluster name. It is a label to identify the cluster.

- To install the Neutro Master start script as system service in the `init.d` directory.

Neutro Master configuration

No further configurations are required after the installation step.

In case you want to change the used JRE to run the Neutro Master, you can manually point to a directory of a supported Java Runtime Environment editing the configuration file

```
${NEUTRO_ROOT}/conf/master.env.sh
```

and set the value of the variable `NE_JAVA_HOME` to the directory of a desired JRE. Note that `NEUTRO_ROOT` is the full path to the Neutro installation directory.

In case you want to change the Neutro Master IP address from the one selected during installation, you can edit the configuration file

```
${NEUTRO_ROOT}/conf/master.conf
```

and change the value of the property `NE_MASTER_ADDRESSES`, setting the IP address of the Neutro Master host, e.g.

```
NE_MASTER_ADDRESSES=192.168.1.20
```

It must be possible for the Neutro Agents and Clients to contact the Neutro Master through the network. Please check your firewall configuration as described in the requirements section in order to allow connections among the hosts of the Neutro cluster.

In case you want to change the Neutro Master administrator user, or add an administrator after the installation step, you can edit the configuration file

```
${NEUTRO_ROOT}/conf/master.conf
```

and change the value of the property `NE_ADMINS`, setting one or more username, e.g.

```
NE_ADMINS=user1,user2
```

In case you want to change the Neutro cluster name, you can edit the configuration file

```
${NEUTRO_ROOT}/conf/master.conf
```

and change the value of the property `NE_CLUSTER_NAME`, setting the desidered label e.g.

```
NE_CLUSTER_NAME=clustername
```

In case you want to tag Neutro Agent host(s), you can edit the configuration file

```
${NEUTRO_ROOT}/conf/hosttags.conf
```

and add on the left side of the equal sign a regex matching one or more hostnames and on the right side of the equal sign a comma separated list of tags, ex.:

```
hostname[1-3] = hostsDepB
hostname[4-6] = hostsDepA
```

In case you want to have a static list of Neutro Agent hosts, which means that hosts, either alive or not, will always reported in the host list, you should edit the configuration file

```
${NEUTRO_ROOT}/conf/hosts.conf
```

and add one Neutro Agent host name per row.

Start the Neutro Master

To run the Neutro Master daemon you can launch:

```
${NEUTRO_ROOT}/bin/neutro-master start
```

where `NEUTRO_ROOT` is the full path to the Neutro Master installation directory.

You can verify in every moment if the Neutro Master is running with the command

```
${NEUTRO_ROOT}/bin/neutro-master status
```

init.d

In the case you didn't choose to setup the Neutro Master script as system service at installation time you can also do it manually.

In order to add the `neutro-master` script to the `init.d` directory, you *must* create a symbolic link. Please *avoid* to copy or move the script.

```
ln -s ${NEUTRO_ROOT}/bin/neutro-master
              /etc/init.d/neutro-master
```

Next you should add the `neutro-master` script to the system services:

```
chkconfig --add neutro-master
chkconfig neutro-master on
```

Stop the Neutro Master

To stop the Neutro Master daemon you can launch:

```
${NEUTRO_ROOT}/bin/neutro-master stop
```

Installing the Neutro Agent

The Neutro Agent package is distributed as a MSI executable that should be launched on each host of the Neutro cluster.

During installation steps, the installer asks for the IP address of the Neutro Master. The Neutro Agent must be able to contact the Neutro Master on the configured IP address. The installer also asks the permission to add a rule to the Windows firewall to let the Neutro Master contact the Neutro Agent.

In the case of a Neutro Agent update, the installer allows you to choose if to renew the installation or to keep the old configuration settings. Selecting the *"Renew ..."* checkbox will drive you through the installation steps and a new configuration file will be written. Otherwise the old configuration file is used and no further question will be asked.

Neutro Agent is installed as a Windows System Service with name *NICE NEUTRO Agent*.

Unattended Installation

To deploy the Neutro Agent in a unattended mode, you can use the Windows tool **msiexec**, with the following switches:

```
msiexec /q /i nice-neutro-agent-20xx.x-xxxx-Release-xxx.msi
            RECONFIG=<1|0>
            MASTER_ADDRESSES=<masterAddress>
            DISABLE_FIREWALL=<1|0>
```

By default, thus if omitted, the value of the properties are the following

```
RECONFIG=1
MASTER_ADDRESSES=
DISABLE_FIREWALL=1
```

Please note that with parameter `RECONFIG=0` the old configuration will be kept, and the `MASTER_ADDRESSES` parameter will be ignored.

Neutro Agent configuration

The Neutro Agent does not require any post-installation configuration.

In case you want to change the Neutro Master IP address from the one selected during installation, you can edit the configuration file

```
${NEUTRO_ROOT}/NeutroAgent.exe.config
```

and change the value of the property `NE_MASTER_ADDRESSES`, setting a comma separated list of IP addresses or hostnames of the Neutro Master hosts, e.g.

```
<Setting Name="NE_MASTERS_ADDRESSES" serializeAs="String">
  <Value>192.168.1.34</Value>
</Setting>
```

Note that `NEUTRO_ROOT` is the full path to the Neutro Agent installation directory.

Start the Neutro Agent

Once the Neutro Agent is installed you should find it as running among the Windows Services.

You can access the Windows System Services panel by launching **services.msc** from the Windows Start - Run box.

From the list you can check and manage the Neutro service called *NICE NEUTRO Agent*.

Check Neutro cluster

In order to do a quick check of your Neutro cluster installation you should run the Neutro Shell client application.

The Neutro Shell is already provided by the Neutro Master installer, and it is ready to start, as it takes the configuration parameters already set up for the Neutro Master.

Start the Neutro Shell

To run the Neutro Shell you can launch:

```
${NEUTRO_ROOT}/bin/neutro-shell.sh
```

To check the Neutro cluster you can digit the command

```
hosts
```

and verify that the Neutro Agent host is listed. The output should be something similar to

```
=> hosts
Host list (1) ==>
Host (name(agent01) status(AVAILABLE) taskSlots(1) busySlots(0)
                    ip(192.168.1.10))
```

You can also check host details to verify if the host can satisfy your submission requirement

```
=> host agent01
Host (name(agent01) status(AVAILABLE) taskSlots(1) busySlots(0)
                                ip(192.168.1.10) resources list:
 - tags: (myHost)
 - os: (Win_7)
 - ncpus: (1)
 - mem_total: (16384)
 - arch: (Intel_Core2_Quad)
 - is_vm: (False)
 - agent_ver: (13.0.40009.0)
 - cpu_bit: (64)
 - logged_users: (1)
 - vnc_mirror_driver.mem: (-1)
 - nvidia.mem: (1024)
 - swp_avail: (4095)
 - swp_total: (4095)
 - cpu_load: (0)
 - apps: (RealVNC, RealVncVE, RealVncMirrorDriver, NiceDCVEndstation)
 - nvidia_geforce_9800gt.mem: (512)
 - ncores: (4)
 - mem_avail: (3038)
 - gpu: (nvidia, nvidia_geforce_9800GT, VNC_mirror_driver)
)
```

Then submit a simple *Task*, with the command

```
tasksubmit simpleTask
```

and verify that *Task* has been dispatched. The output should be something similar to

```
=> tasksubmit simpleTask
Task instance: Task (id(1) status(DISPATCHED) user(luca) type(simpleTask)
                        host(agent01) startTime(2012-11-28 15:35:44))
```

To see the list of the submitted tasks launch the command

```
tasks
```

and verify that the submitted *Task* is running. The output should be something similar to

```
=> tasks
Tasks (1) ==>
Task (id(1) status(RUN) user(luca) type(simpleTask) host(agent01)
                    startTime(2012-11-28 15:35:44))
```

You can check again the cluster status to see the status of the host where the submitted task is running. The host should result busy since by default the Neutro Agent is configured to accept at most one task running on the host.

```
=> hosts
Host list (1) ==>
Host (name(agent01) status(BUSY) taskSlots(1) busySlots(1)
                    ip(192.168.1.10))
```

Then you can decide to kill the task you just submitted with the command

```
taskop [taskid] kill
```

where [taskid] is the ID number of the submitted *Task*.

And finally check the status of the killed *Task*

```
=> tasks
Tasks (1) ==>
Task (id(1) status(EXIT) user(luca) type(simpleTask) host(agent01)
        startTime(2012-11-28 15:35:44) endTime(2012-11-28 15:39:10))
```

7

Troubleshooting

Useful information for troubleshooting can be found inside Neutro logs. They are located under different location.

Neutro Master log

On the Neutro Master logs can be found under

```
${NEUTRO_ROOT}/log
```

Neutro Agent log

On the Neutro Agent log location depends on the OS type.
On Windows 7 and later, logs can be found under

```
C:\Windows\System32\config\systemprofile\Neutro\logs
```

On Windows XP 32 bit logs can be found under

```
C:\Documents and Settings\LocalService\Neutro\logs
```

On Windows XP 64 bit logs can be found under

```
C:\Documents and Settings\Default User\Neutro\logs
```

Please consider attaching logs of interest when sending a support request to helpdesk@nice-software.com.