Purpose

This document describes how to install Oracle plugins in Nagios XI and how to configure your environment to utilize the Oracle plugins to monitor Oracle databases with Nagios XI.

Target Audience

This document is intended for use by Nagios XI Administrators who wish to monitor Oracle databases to ensure their system is functioning properly.

Acquire The Required RPM Files And Install Script

Establish a terminal session to your Nagios XI server as the root user. Download the install script to the `/tmp` folder of your Nagios XI server using the following commands:

```
cd /tmp
wget https://assets.nagios.com/downloads/general/scripts/oracleinstall.sh
```

First thing we will need to determine is what architecture is running on your Nagios XI Linux server. If you already know it, you can skip this step, but be sure to double-check as you will need to know it later in the document. At your Linux command prompt type the following command:

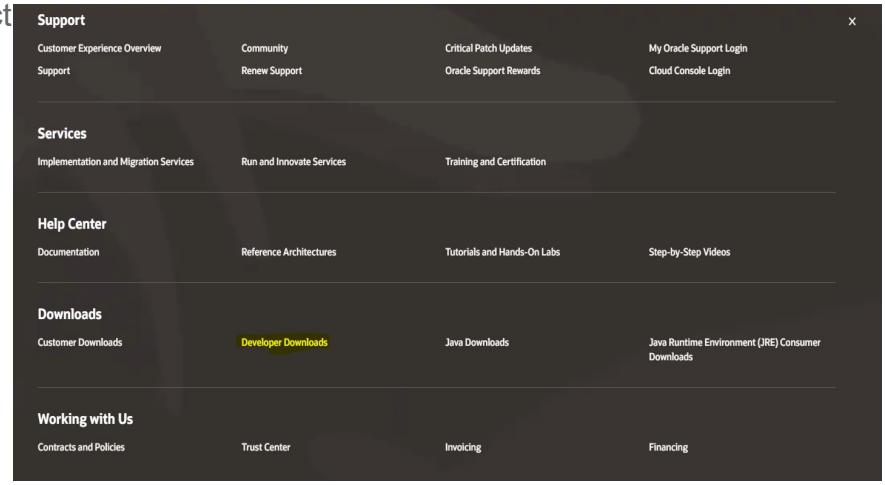
```
uname -i
```

This will return the architecture of your system. It will either be:

`i?86` OR `x86_64`

The `?` in `i?86` represent any numbers returned. However for our purposes it boils down to one thing, your system is 32 bit. If its equal to the bottom then your architecture is 64 bit. With this information we can now download the proper RPMs.

Due to legal reasons, Nagios cannot provide direct download links to these RPMs so you will have to go to Oracle's website and download them. Navigate to http://www.oracle.com/ and create an account.



Once you've created your account and logged in, click Resources > Developer Downloads link on the main menu. Then, click on the "Database" section on the Downloads page.

Next, find the **Instant Client** link and click it.

Now based on the results from the previous `uname -i` command, select the proper install for your particular Nagios XI installation, either 32 bit or 64 bit.

Once you've selected your architecture, the next (and final screen) will come up. Download all **three** of the `.rpm` links indicated to the right for the Oracle Client version you decide to use. Also place these `.rpm` files in the same directory as you downloaded the install script, in this document we used the `/tmp` directory.

Run The Install Script

NOTE: CentOS/RHEL 8/9 systems require the following command to be run before the Oracle Client install begins:

```
dnf install libnsl -y
```

Once you've moved the script and all the .rpm files to the same directory, its time to execute the script:

```
chmod +x oracleinstall.sh
./oracleinstall.sh
```

This script will take care of most of the dirty work. It will install some dependencies, then grab the `check_oracle_health` tarball from the internet, configure and install it, then install the Perl dependencies through CPAN. At the very end, will dump out some variables that you will need to apply to your particular case. In the example presented in this document, they look like this (the version number will vary depending on the oracle version you downloaded):

```
ORACLE_HOME=/usr/lib/oracle/18.5/client64
LD_LIBRARY_PATH=/usr/lib/oracle/18.5/client64/lib
```

Copy the variables presented on screen somewhere as you will need them later in the document.

NOTE: The Oracle Wizards have been tested with Oracle clients up to version 18.5. We cannot guarantee that they will work with any newer version of the Oracle client.

Configure Oracle Environment (New)

This section can only be used if you are running **all** of the following software versions:

- Nagios XI 5.6.1 or later
- Oracle Query Wizard 1.3.5 or later

- Oracle Serverspace Wizard 1.5.5 or later
- Oracle Tablespace Wizard 1.5.6 or later

If you are not using all of these, please refer to the next section (Configure Oracle Plugin Commands).
As of Nagios XI 5.6.1, the Oracle check_commands will no longer need to be modified directly. Instead, the Oracle-related environment variables will be read from **/usr/local/nagiosxi/etc/configwizards/oracle/oracle**. If the wizards installed correctly, the file's contents should look similar to this:

```
LD_LIBRARY_PATH=/usr/lib/oracle/18.5/client64/lib
ORACLE_HOME=/usr/lib/oracle/18.5/client64
```

NOTE: The '18.5' version number presented above needs to match the version number within the variables saved after successful installation. Also, the 'client' directory may need to be changed depending on your system type. For example, 32 bit systems would need to be changed to 'client'. See your saved variables from above for exact verbiage.

```
LD_LIBRARY_PATH=/usr/lib/oracle/18.5/client/lib
ORACLE_HOME=/usr/lib/oracle/18.5/client
```

Change the paths in **/usr/local/nagiosxi/etc/configwizards/oracle/oracle** to the paths shown by `oracleinstall.sh` in the previous section if needed. You can use a text editor such as “vi”.

Configure Oracle Plugin Commands (Legacy)

This section is only needed if you are using any of the following software versions:

- Nagios XI 5.6.0 or earlier
- Oracle Query Wizard 1.3.4 or earlier
- Oracle Serverspace Wizard 1.5.4 or earlier
- Oracle Tablespace Wizard 1.5.5 or earlier

We need to make a few modifications to commands used by the Oracle wizards. The Oracle check plugins requires access to outside libraries, and Nagios does not spawn processes with access to outside environment variables, and these environment variables are particular to your computer. The install script should have created two environment variables for you (`ORACLE_HOME` and `LD_LIBRARY_PATH`).

Navigate to **Configure** > **Core Config Manager** > **Commands** and click **>_ Commands** from the left-side menu. Now you'll need to find the oracle services, use the **search** function to search for **`oracle`** and you should see the following three commands (`check_xi_oraclequery`, `check_xi_oracleserverspace`, `check_xi_oracletablespace`) as per the screenshot below.

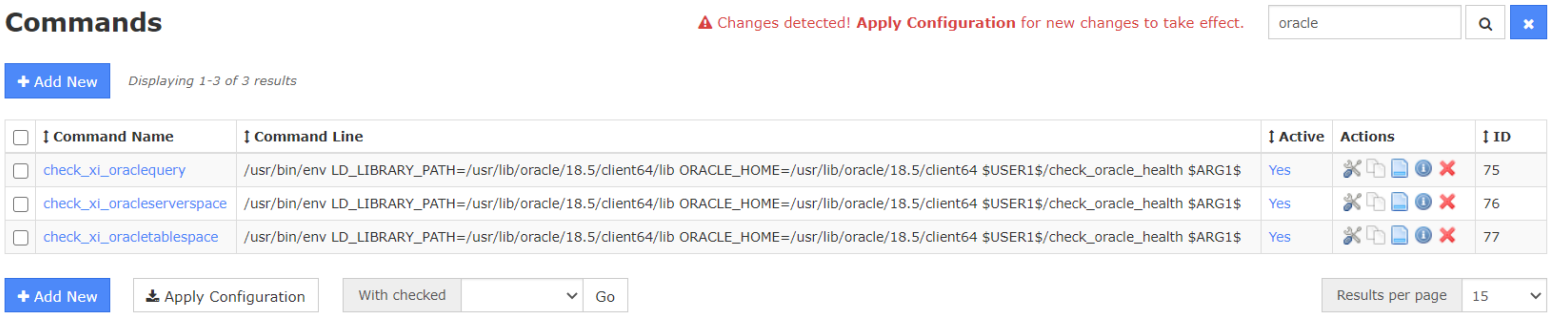
Command Name	Command Line	Active	Actions	ID
check_xi_oraclequery	/usr/bin/env LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client/lib ORACLE_HOME=/usr/lib/oracle/11.2/client $USER1$/check_oracle_health $ARG1$	Yes		73
check_xi_oracleserverspace	/usr/bin/env LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client/lib ORACLE_HOME=/usr/lib/oracle/11.2/client $USER1$/check_oracle_health $ARG1$	Yes		74
check_xi_oracletablespace	/usr/bin/env LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client/lib ORACLE_HOME=/usr/lib/oracle/11.2/client $USER1$/check_oracle_health $ARG1$	Yes		75

One at a time, edit them by clicking the name of the command. When the editing screen will look like this:

In the screenshot above you'll see the text box where the command is actually held, **Command Line**. Remember the variables the script dumped out? You'll need to input them here. Your variable might be the same as whats is already there, but its good to double check. Replace the `LD_LIBRARY_PATH` and `ORACLE_HOME` variables in the Command Line field with the variables given by the script above.

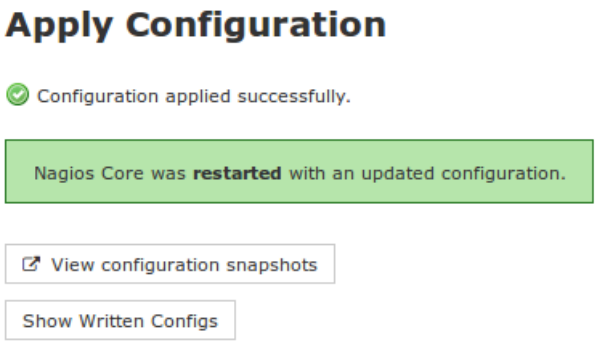
Click **Save** once the changes have been made.

Once you've replaced them you will need to do the same for the two other Oracle check plugins. In the following screenshot you can see the commands reflect the changes that were made.



After you've updated all of the commands, click the **Apply Configuration** button at the bottom of the Commands page.

Once the configuration has been applied you are now ready to use the different Oracle Wizards within Nagios XI.



Common Problems

Nagios reports CRITICAL with the message:

```
install_driver(Oracle) failed: Can't load
'/usr/lib/perl5/site_perl/5.8.8/i386-linux-thread-
multi/auto/DBD/Oracle/Oracle.so' for module DBD::Oracle: libocci.so.11.1:
cannot open shared object file: No such file or directory at
/usr/lib/perl5/5.8.8/i386-linux-thread-multi/DynaLoader.pm line 230.
```

This problem is due to environment variable not being set properly. This Oracle plugin relies on these environment variables being set. Make you have probably set the variables as indicated above. Also verify that the proper path is being given. Usual paths are:

```
x86:      /usr/lib/oracle/<YOUR ORACLE VERSION>/client
x86_64:   /usr/lib64/oracle/<YOUR ORACLE VERSION>/client64
```

See the section about editing the environment variables of the services. Make sure the environment variables are set to proper value. These can be verified by actually going to them on the command line.

Nagios gives this error:

```
CRITICAL - cannot connect to 192.168.5.55:1521. install_driver(Oracle) failed:
Can't locate DBD/Oracle.pm in @INC (@INC contains:
/usr/local/nagios/libexec /usr/lib/perl5/site_perl/5.8.8/i386-linux-thread-
multi /usr/lib/perl5/site_perl/5.8.8 /usr/lib/perl
```

This is due to the Perl module not being installed properly. To remedy this problem, execute the following from the command line:

```
export ORACLE_HOME=<path to your Oracle which was given by the script>
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
cpan -i DBD::Oracle
```

If you were not able to install the DBD:Oracle perl module by following the above instructions, you could try:

```
cd /tmp
wget http://www.cpan.org/modules/by-module/DBD/DBD-Oracle-1.74.tar.gz
tar zxvf DBD-Oracle-1.74.tar.gz
cd DBD-Oracle-1.74
perl Makefile.PL -l
make && make test
make install
```

Nagios gives 'OID generation failed' error:

This problem is due to the hostname of the server being changed and not existing in the hosts file. You will need to edit the /etc/hosts file to add the hostname of the server.

```
vi /etc/hosts
```

Finishing Up

This completes the documentation on how to install and configure the Oracle client & plugins in Nagios XI.

If you have additional questions or other support related questions, please visit us at our Nagios Support Forums:

https://support.nagios.com/forum

The Nagios Support Knowledgebase is also a great support resource:

https://support.nagios.com/kb