

## The Industry Standard in IT Infrastructure Monitoring

### 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

Open a terminal and log into your Nagios XI server as the root user. Download the install script to the /tmp folder of your Nagios XI server:

```
cd /tmp
wget http://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

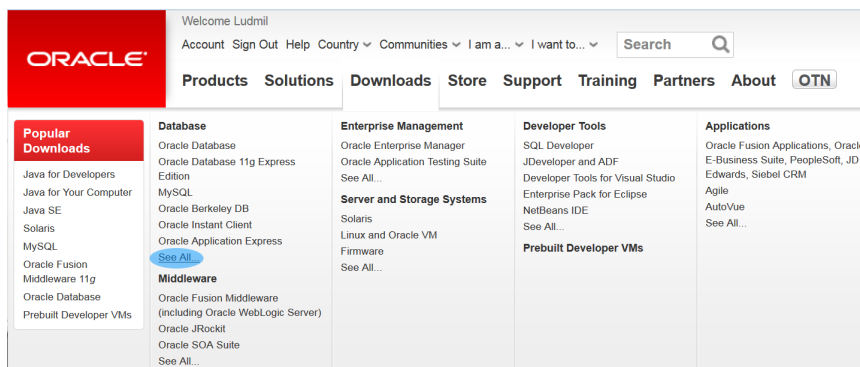
```
x86_64
```

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.

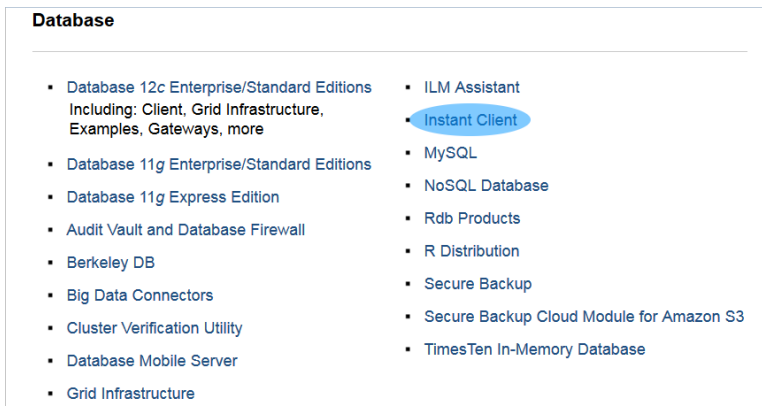
Due to legal reasons, Nagios cannot provide direct download links to these RPMs so you will have to go to [Oracle's website](http://www.oracle.com/) and download them. Navigate to <http://www.oracle.com/> and create an account.

# Nagios XI – How to Install & Configure the Oracle Client & Plugins

Once you've created your account, click the **Downloads** tab, then click on the **See All...** link under the "Database" section.



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.

- Instant Client for Linux x86
- Instant Client for Linux x86-64

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. Also place these `.rpm` files in the same directory as you downloaded the install script, in this document we used the `/tmp` directory.

- Version 12.1.0.2.0**
- Instant Client Package - Basic: All files required to run OCI, OCCI, and JDBC-OCI applications
    - instantclient-basic-linux.x64-12.1.0.2.0.zip (63,352,239 bytes) (cksum - 109893216)
    - oracle-instantclient12.1-basic-12.1.0.2.0-1.x86\_64.rpm (62,587,782 bytes) (cksum - 2840691603)
  - Instant Client Package - Basic Lite: Smaller version of the Basic package, with only English error messages and Unicode, ASCII, and Western European character set support
    - instantclient-basclite-linux.x64-12.1.0.2.0.zip (31,302,454 bytes) (cksum - 3592144768)
    - oracle-instantclient12.1-basclite-12.1.0.2.0-1.x86\_64.rpm (30,940,809 bytes) (cksum - 174277137)
  - Instant Client Package - JDBC Supplement: Additional support for XA, Internationalization, and RowSet operations under JDBC
    - instantclient-jdbc-linux.x64-12.1.0.2.0.zip (1,571,639 bytes) (cksum - 788395887)
    - oracle-instantclient12.1-jdbc-12.1.0.2.0-1.x86\_64.rpm (1,564,558 bytes) (cksum - 1691314891)
  - Instant Client Package - SQL\*Plus: Additional libraries and executable for running SQL\*Plus with Instant Client
    - instantclient-sqlplus-linux.x64-12.1.0.2.0.zip (861,284 bytes) (cksum - 2255047856)
    - oracle-instantclient12.1-sqlplus-12.1.0.2.0-1.x86\_64.rpm (852,271 bytes) (cksum - 1611119156)
  - Instant Client Package - SDK: Additional header files and an example makefile for developing Oracle applications with Instant Client
    - instantclient-sdk-linux.x64-12.1.0.2.0.zip (667,174 bytes) (cksum - 1047596065)
    - oracle-instantclient12.1-devel-12.1.0.2.0-1.x86\_64.rpm (634,803 bytes) (cksum - 2599726994)

## Run The Install Script

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 using yum, 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 my case they look like this:













```
ORACLE_HOME=/usr/lib/oracle/12.1/client64
LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
```

Copy these variable somewhere as you will need them later in the document.

## Configure Oracle Plugin Commands

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:

<input type="checkbox"/>	Command Name	Command Line	Active	Actions	ID
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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 something like this:

## Command Management

Command Name \*

Example: check\_example

Command Line \*

Example: \$USER1\$/check\_example -H \$HOSTADDRESS\$ -P \$ARG1\$ \$ARG2\$

Command Type:

Active ?

Available Plugins

 ?

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 what's already there, but it's 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.

## Commands

▲ Changes detected! **Apply Configuration** for new changes to take effect.

*Displaying 1-3 of 3 results*

<input type="checkbox"/>	Command Name	Command Line	Active	Actions	ID
<input type="checkbox"/>	check_xi_oraclequery	/usr/bin/env LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client/lib ORACLE_HOME=/usr/lib/oracle/12.1/client \$USER1\$/check_oracle_health \$ARG1\$	Yes		73
<input type="checkbox"/>	check_xi_oracleserverspace	/usr/bin/env LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client/lib ORACLE_HOME=/usr/lib/oracle/12.1/client \$USER1\$/check_oracle_health \$ARG1\$	Yes		74
<input type="checkbox"/>	check_xi_oracletablespace	/usr/bin/env LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client/lib ORACLE_HOME=/usr/lib/oracle/12.1/client \$USER1\$/check_oracle_health \$ARG1\$	Yes		75

Results per page 15

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

## Apply Configuration

✔ Configuration applied successfully.

Nagios Core was **restarted** with an updated configuration.

[View configuration snapshots](#)

[Show Written Configs](#)

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>/client
```

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
```

## Finishing Up

You are now ready to use and Oracle Wizard to monitor your Oracle databases. If you have any additional support issues that you cannot resolve, please post your questions on the Nagios Support Forum:

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