MAY 26, 2017BY ROSEMBERG

Installing PDO Oracle and OCI8 from PHP7 on Ubuntu/Mint (Oracle 11.2)

"Edit – At the end of this post, I added a video showing the step by step of this tutorial"

If you are having or have had a headache installing the OCI8 and PDO modules for Oracle like me, here is a small step-by-step guide of what worked for me, I hope it helps you too.

The steps in this tutorial require the user to have root user privileges.

If you already have Apache and PHP installed, just check if your computer is missing any PHP packages that were installed in step 2.

Step 1 – Install Apache.

Open the terminal and type the following commands:

```
sudo apt-get update
sudo apt-get install apache2
```

With Apache installed, test in the browser ifthe service is running through the url **https:/localhost**

Step 2 – Install PHP.

```
sudo apt-get install php7.0 php7.0-xml php-xml php-common php7.2-cli php7.2-common php7.2-fpm
php7.2-json php7.2-opcache php7.2-readline libapache2-mod-php7.2 php-pear php7.2-dev
```

Step 3- Download the following Oracle Client packages, download *.rpm packages > https://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html

```
 oracle-instantclient11.2-basic-11.2.0.4.0-1.x86_64.rpm

oracle-instantclient11.2-sqlplus-11.2.0.4.0-1.x86_64.rpm

oracle-instantclient11.2-devel-11.2.0.4.0-1.x86_64.rpm
```

Step 4 – Install alien to convert .rpm packages (redhat family) to .deb (debian family).

```
$ sudo apt-get install alien
```

Step 5 – Convert the .rpm files and install.

```
$ sudo alien -i oracle-instantclient11.2-basic-11.2.0.4.0-1.x86_64.rpm
$ sudo alien -i oracle-instantclient11.2-sqlplus-11.2.0.4.0-1.x86_64.rpm
$ sudo alien -i oracle-instantclient11.2-devel-11.2.0.4.0-1.x86_64.rpm
```

Step 6 – Install the libaio1 library.

```
$ sudo apt-get install libaio1
```

Step 7- Create the oracle.conf file in the folder`/etc/ld.so.conf.d/ com o caminho da lib do oracle client, e após criar o arquivo recarregue as configurações de ldconfig, *atenção para mudar o caminho se o seu sistema(PC/SO) não for 64bits(client64).*`

```
$ sudo nano /etc/ld.so.conf.d/oracle.conf
```

paste the lib path in my case it was: **/usr/lib/oracle/11.2/client64/lib** save the file with CTRL+x

Reload settings:

```
$ sudo ldconfig
```

Step 8 – Export the ORACLE_HOME variable

```
$ export ORACLE_HOME=/usr/lib/oracle/11.2/client64/
```

Step 9 – Download the oci8 lib source code , extract the contents of the downloaded file, compile and install it.

```
$ pecl download OCI8
```

```
-------
downloading oci8-2.1.4.tgz...
Starting to download oci8-2.1.4.tgz (191,992 bytes)
.....................................done: 191,992 bytes
File /home/me/oci8-2.1.4.tgz downloaded

oci8-2.1.4.tgz was the downloaded file
--------
```

```
$ tar zxvf oci8-2.1.4.tgz (extraindo o arquivo baixado)
$ cd oci8-2.1.4/
$ phpize
$ ./configure --with-oci8=instantclient,/usr/lib/oracle/11.2/client64/lib
```

```
$ sudo make install
```

After completing the compilation and installation, it will inform you that the file was generated and its path, in my case it was:

```
---------------------------------------------- --------------------
Libraries have been installed in:
 /home/me/Downloads/oci8-2.1.4/modules

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
 - add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
 during execution
 - add LIBDIR to the 'LD_RUN_PATH' environment variable
 during linking
 - use the '-Wl,-rpath -Wl,LIBDIR' linker flag
 - have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
---------------------------------------------- --------------------
Installing shared extensions: /usr/lib/php/20151012/
```

Step 10 – Enable the lib generated in PHP:

Create the oci8.ini file in the PHP available modules folder, in my case it was in the **/etc/php/7.0/mods-available/** folder , and after that create a symbolic link to activate the lib, in my case it was in **/etc/php/7.0/apache2/conf.d** folder

```
$ sudo nano /etc/php/7.0/mods-available/oci8.ini
```

paste the lib name:
extension=oci8.so
save the file with CTRL+x

```
$ cd /etc/php/7.2/apache2/conf.d
$ sudo ln -s /etc/php/7.2/mods-available/oci8.ini oci8.ini
```

Step 11 – Check your PHP version:

```
$ php -v
```

————————
PHP 7.0.8-0ubuntu0.16.04.1 (cli) (NTS)
Copyright (c) 1997-2017 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies
with Zend OPcache v7.0.8-0ubuntu0.16.04.1, Copyright (c) 1999-2017, by Zend Technologies
————————-

Download the PHP source code directly from the github website, download the version corresponding to the version installed on your PC, in my case it was version 7.0.8 (https://github.com/php/php-src/tree/PHP- 7.0.8), if this is not your version, select the corresponding version in the **branches** menu and download the zip, this step is to get the correct version of the **PDO_OCI** lib .

Step 12 – Using the command line, extract the downloaded file, compile and install the lib.

```
$ unzip php-src-PHP-7.0.8.zip #(extraindo o arquivo baixado)
$ cd php-src-PHP-7.0.8/ext/pdo_oci/
```

(in newer versions [7.0.25 for example], this block below is not necessary, you can go straight to the command -> phpize)

```
$ nano config.m4
```

edit the config.m4 file, to add the 11.2 version
of oracle, find the **SUPPORTED_LIB_VERS** variable (more or less on line 5) and add the **11.2**
version number in the version list, also look for **case $PDO_OCI_VERSION** (more or less on line 134) and add **| 11.2 |**
in the version list, save the file with CTRL+X **.**

```
$ phpize
$ ./configure --with-pdo-oci=instantclient,/usr,11.2 #(atente para a versão da lib do oracle)
$ sudo make install
```

Step 13 – Enable the pdo_oci lib generated in PHP:

Create the pdo_oci.ini file in the php available modules folder and the symbolic link as in step 10, just changing the name of the lib:

```
$ sudo nano /etc/php/7.0/mods-available/pdo_oci.ini
```

paste the lib name:
extension=pdo_oci.so
save the file with CTRL+x

```
$ cd /etc/php/7.0/apache2/conf.d
$ sudo ln -s /etc/php/7.0/mods-available/pdo_oci.ini pdo_oci.ini
```

Step 14 – Restart Apache

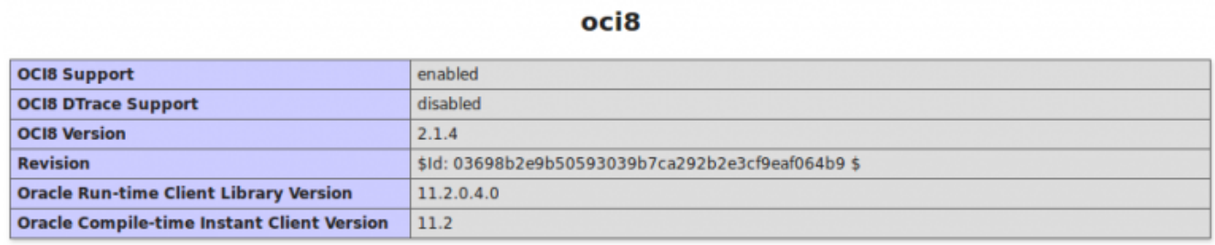
```
$ sudo service apache2 restart
```

Step 15 – Enable the visualization of php information, creating the **info.php file with the phpinfo()** function in the apache **html** folder ; and access the url https://localhost/info.php in the browser and look for the oci8 and pdo_oci libs to see if they are active

```
$ sudo nano /var/www/html/info.php
```

paste the following content:
<?php
phpinfo();
?>
save the file with CTRL+x

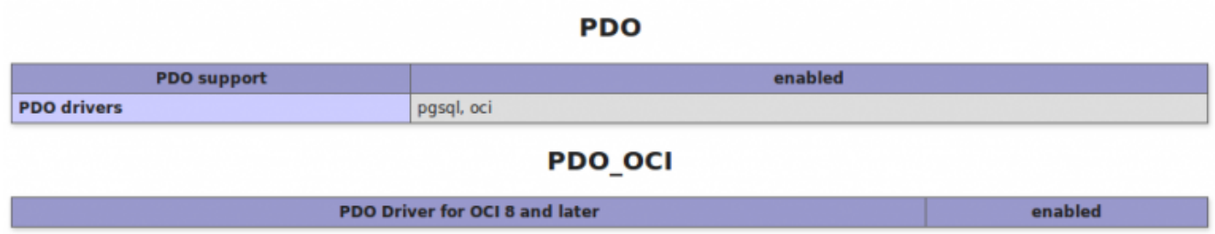
When accessing the page, look for the following excerpts, if they look like the images below, your PHP is working correctly with the oci8 and pdo_oci libs. After verification, delete the created **info.php** file .



oci8

OCI8 Support	enabled
OCI8 DTrace Support	disabled
OCI8 Version	2.1.4
Revision	$Id: 03698b2e9b50593039b7ca292b2e3cf9eaf064b9 $
Oracle Run-time Client Library Version	11.2.0.4.0
Oracle Compile-time Instant Client Version	11.2

PHP information screen stating that oci8 is active.



PDO

PDO support	enabled
PDO drivers	pgsql, oci

PDO_OCI

PDO Driver for OCI 8 and later	enabled

PHP information screen stating that pdo_oci is active.

"Video demonstrating the step by step of this tutorial"

GNU/LINUX , TUTORIAL

GNU , GNU/LINUX , LINUX , MINT , OCI8 , ORACLE , PDO , PHP , TUTORIAL , UBUNTU