

PDO を使う

仕組み

PDO -> PDO_OCI -> Oracle ライブラリ

準備

必要なもの

11g は失敗するかも

- Oracle Instant Client basic 10g
- Oracle Instant Client devel 10g
- php-pdo
- re2c

```
[dag]
name=Dag RPM Repository for Red Hat Enterprise Linux
baseurl=http://ftp.riken.jp/Linux/dag/redhat/el$releasever/en/$basearch/dag
gpgcheck=1
gpgkey=http://ftp.riken.jp/Linux/dag/packages/RPM-GPG-KEY.dag.txt
```

```
yum install re2c
```

インストール

1. Oracle Client をインストール
2. PDO_OCI をインストール ()
3. /etc/php.d/pdo_oci.ini を作成し、extension=pdo_oci.so を追加

接続

```
define ("_dns", "oci:dbname=192.168.10.139/xe");
define ("_db_user", 'test');
define ("_db_pass", 'test');
db = new PDO(_dns, _db_user, _db_pass);
```

な感じ

注意点

日付のデフォルトフォーマット

自分が想定している日付のフォーマットになっていないことがある。

```
alter session set NLS_DATE_FORMAT = 'yyyy/mm/dd hh24:mi
```

みたいな感じで日付のフォーマットを決めておいたほうが良い。

MDB2 を使う

仕組み

MDB2 -> MDB2_Driver_oci8 -> oci8 -> Oracle ライブラリ

準備

必要なもの

- Oracle Instant Client basic 10g
- Oracle Instant Client devel 10g

11g は失敗するかも

インストール

1. Oracle Client をインストール
2. oci8 をインストール ()
3. pear install MDB2
4. pear install MDB2_Driver_oci8
5. php.ini または、 /etc/php.d/oci8.ini を作成し、 extension=oci8.so を追加

接続

```
define ("_dns", "oci8://test:test@//192.168.10.139/xe");  
db = MDB2::factory( _dns );
```

な感じ。または、

```
oci8://test:test@192.168.10.139:1521/?service=xe
```

10g の場合

```
test:test@//192.168.10.139:1521/xe
```

のようにポート指定をすると接続に失敗するので注意

各ライブラリのインストール方法

oci8 インストール

pecl を使う場合

```
pecl install oci8
```

を実行。

```
Please provide the path to the ORACLE_HOME directory. Use  
'instantclient,/path/to/instant/client/lib' if you're compiling with Oracle Instant Client  
[autodetect] :
```

と出たら Oracle instant のライブラリのパスを入力する。

入力例 :

```
instantclient,/usr/lib/oracle/10.2.0.4/client/lib/  
instantclient,/usr/lib/oracle/10.2.0.4/client64/lib/
```

pecl を使わない場合

pecl に存在しない等の理由でエラーが出る場合は手動でインストールする。
<http://pecl.php.net/package/oci8> から をダウンロード。

```
tar zxvf oci8-2.0.8.tgz  
cd oci8-2.0.8  
phpize  
./configure --with-oci8=instantclient,/usr/lib/oracle/10.2.0.4/client64/lib/ <= パスは環境にあわせて  
修正  
make  
make install
```

PDO_OCI インストール

10g

64bit 版は client64 にインストールするが PDO_OCI の configure スクリプトは client 決め打ち
なのでディレクトリー名を変更する

configure スクリプトの方を変更しても良い・・・かも。

1. 64bit の場合、/usr/lib/oracle/ バージョン /client64 を /usr/lib/oracle/ バージョン /client ヘシン
ポリックリンクを貼るか、リネームす
2. 64bit の場合、/usr/include/oracle/ バージョン /client64 を /usr/include/oracle/ バージョン
/client ヘシンポリックリンクを貼るか、リネームする
3. wget http://pecl.php.net/get/PDO_OCI-1.0.tgz
4. tar xvfz PDO_OCI-1.0.tgz
5. cd PDO_OCI-1.0
6. phpize
7. データベース (SJIS) - クライアント (UTF-8) の場合ソースを修正する
 1. -col->maxlen = data_size;
 2. +col->maxlen = data_size*3;
8. ./configure --with-pdo-oci=instantclient,/usr,10.2.0.4 <- Oracle Client のバージョン
9. make
10. make install

12c

<http://www.code-knight.com/suzuking/items/75fad3dacfc7a8b0b7535d68359d554a>

以下パッチファイルを作成

patch.txt

```
### Eclipse Workspace Patch 1.0  
#P pdo_oci  
Index: config.m4  
=====
```

```
--- config.m4 (revision 141)  
+++ config.m4 (working copy)  
@@ -7,6 +7,8 @@  
+ if test -s "$PDO_OCI_DIR/orainst/unix.rgs"; then  
+   PDO_OCI_VERSION= grep "ocommon" $PDO_OCI_DIR/orainst/unix.rgs | sed 's/[ ] [ ]*/:/g' | cut  
-d: -f 6 | cut -c 2-4`  
+ test -z "$PDO_OCI_VERSION" && PDO_OCI_VERSION=7.3  
+ elif test -f $PDO_OCI_DIR/lib/libcIntsh.$SHLIB_SUFFIX_NAME.12.1; then  
+   PDO_OCI_VERSION=12.1  
+ elif test -f $PDO_OCI_DIR/lib/libcIntsh.$SHLIB_SUFFIX_NAME.10.1; then  
+   PDO_OCI_VERSION=10.1
```

```

elif test -f $PDO_OCI_DIR/lib/libclntsh.$SHLIB_SUFFIX_NAME.9.0; then
@@ -57,14 +59,19 @@
    AC_MSG_CHECKING([for oci.h])
    if test -f $PDO_OCI_IC_PREFIX/include/oracle/$PDO_OCI_IC_VERS/client/oci.h ; then
      PHP_ADD_INCLUDE($PDO_OCI_IC_PREFIX/include/oracle/$PDO_OCI_IC_VERS/client)
+     PDO_OCI_LIB_DIR="$PDO_OCI_IC_PREFIX/lib/oracle/$PDO_OCI_IC_VERS/client/lib"
      AC_MSG_RESULT($PDO_OCI_IC_PREFIX/include/oracle/$PDO_OCI_IC_VERS/client)
    elif test -f $PDO_OCI_IC_PREFIX/lib/oracle/$PDO_OCI_IC_VERS/client/include/oci.h ; then
+     PHP_ADD_INCLUDE($PDO_OCI_IC_PREFIX/lib/oracle/$PDO_OCI_IC_VERS/client/include)
+     PDO_OCI_LIB_DIR="$PDO_OCI_IC_PREFIX/lib/oracle/$PDO_OCI_IC_VERS/client/lib"
      AC_MSG_RESULT($PDO_OCI_IC_PREFIX/lib/oracle/$PDO_OCI_IC_VERS/client/include)
+     elif test -f $PDO_OCI_IC_PREFIX/sdk/include/oci.h ; then
+     PHP_ADD_INCLUDE($PDO_OCI_IC_PREFIX/sdk/include)
+     AC_MSG_RESULT($PDO_OCI_IC_PREFIX/sdk/include)
+     PDO_OCI_LIB_DIR="$PDO_OCI_IC_PREFIX"
    else
      AC_MSG_ERROR([I'm too dumb to figure out where the include dir is in your instant client
install])
    fi
  - PDO_OCI_LIB_DIR="$PDO_OCI_IC_PREFIX/lib/oracle/$PDO_OCI_IC_VERS/client/lib"
  - PDO_OCI_VERSION="`echo $PDO_OCI_IC_VERS | cut -d. -f1-2`"
    else
      if test -d "$PDO_OCI_DIR/rdbms/public"; then
@@ -119,6 +126,9 @@
        10.2)
          PHP_ADD_LIBRARY(clntsh, 1, PDO_OCI_SHARED_LIBADD)
          ;;
+       12.1)
+         dn1 PHP_ADD_LIBRARY(clntsh, 1, PDO_OCI_SHARED_LIBADD)
+         ;;
        *)
          AC_MSG_ERROR(Unsupported Oracle version! $PDO_OCI_VERSION)
          ;;
@@ -176,22 +186,24 @@
      -L$PDO_OCI_LIB_DIR $PDO_OCI_SHARED_LIBADD
    ])

  - ifdef([PHP_CHECK_PDO_INCLUDES],
  - [
  -   PHP_CHECK_PDO_INCLUDES
  - ],[
  -   AC_MSG_CHECKING([for PDO includes])
  -   if test -f $abs_srcdir/include/php/ext/pdo/php_pdo_driver.h; then
+dn1   ifdef([PHP_CHECK_PDO_INCLUDES],
+dn1   [
+dn1     PHP_CHECK_PDO_INCLUDES
+dn1   ],[
+   AC_MSG_CHECKING([ PDO includes])
+   if test -f $prefix/include/php/ext/pdo/php_pdo_driver.h; then
+   pdo_inc_path=$prefix/include/php/ext
+   elif test -f $prefix/include/php5/ext/pdo/php_pdo_driver.h; then
+   pdo_inc_path=$prefix/include/php5/ext
+   elif test -f $abs_srcdir/include/php/ext/pdo/php_pdo_driver.h; then
+   pdo_inc_path=$abs_srcdir/ext
+   elif test -f $abs_srcdir/ext/pdo/php_pdo_driver.h; then
+   pdo_inc_path=$abs_srcdir/ext
  -   elif test -f $prefix/include/php/ext/pdo/php_pdo_driver.h; then
  -   pdo_inc_path=$prefix/include/php/ext
    else
      AC_MSG_ERROR([Cannot find php_pdo_driver.h.])
    fi
    AC_MSG_RESULT($pdo_inc_path)
  - ])
+dn1 ])

  PHP_NEW_EXTENSION(pdo_oci, pdo_oci.c oci_driver.c oci_statement.c, $ext_shared,,-I$pdo_inc_path)

```

64bit 版は client64 にインストールするが PDO_OCI の configure スクリプトは client 決め打ちなのでディレクトリー名を変更する

configure スクリプトの方を変更しても良い・・・かも。

1. 64bit の場合、 /usr/lib/oracle/12.1/client64 を /usr/lib/oracle/12.1/client ヘシンのリンクを貼るか、リネームす
2. 64bit の場合、 /usr/include/oracle/12.1/client64 を /usr/include/oracle/12.1/client ヘシンのリンクを貼るか、リネームする

3. wget http://pecl.php.net/get/PDO_OCI-1.0.tgz
4. tar xvfz PDO_OCI-1.0.tgz
5. cd PDO_OCI-1.0
6. patch -p0 < patch.txt
7. phpize
8. データベース (SJIS) - クライアント (UTF-8) の場合ソースを修正する
 1. -col->maxlen = data_size;
 2. +col->maxlen = data_size*3;
9. ./configure --with-pdo-oci=instantclient,/usr,12.1 <- Oracle Client のバージョン
10. make
11. make install

文字コード

DB サーバとクライアントの文字コードが異なる場合は以下のいずれかの方法で設定する。

接続時に文字コードを指定する

```
charset
```

を指定する。

例

```
oci:dbname=IP/ORCL;charset=AL32UTF8
oci8://test:test@192.168.10.139:1521/?service=xe&charset=AL32UTF8
```

環境変数等に文字コードを指定する

```
NLS_LANG=Japanese_Japan.UTF8
```

等の環境変数を設定する

ORA-24408 が発生する場合

```
/etc/hosts
```

に

```
127.0.0.1 自分のホスト名
```

を書く。自分のホスト名から IP が引けないとこのエラーが出てしまうことがあるみたい。

LOB の扱い

MDB2

大きく 2 つの方法がある。

文字列として扱う

特に意識しなければ、普通の varchar2 と同じように扱える。

ただし、末尾のスペースが消えるなどの動作することがあるので注意。

ファイルポインタとして扱う

prepare の第 3 引数に戻り値の型を渡すことで blob の戻り値をファイルポインタが返ってくる。

```
$db->loadModule('Datatype');
....
$sql = "select .... "
$types = null; null か バインド変数の型
$returnTypes = array("blob", "text");
$stmt = db->prepare($sql, $types, $returnTypes);
$rows = $stmt->fetchAll(MDB2_FETCHMODE_ASSOC);
$res = $rows[0]["img"];
if (is_resource($res)) {
    while(!feof($res)) {
        echo fread($res, 8192);
        ob_flush();
        flush();
    }
}
$db->datatype->destroyLOB($res);
```

OCI8 直接

MDB2 同様、大きく 2 つの使い方がる。

文字列として扱う

oci_fetch_all で取得すると文字列として取得できる。MDB2 の時のように末尾のスペースが消えることはない。

OCI-Lob クラスとして扱う

oci_fetch_array で取得すると、OCI-Lob クラスが返ってくる。