PEAR DB: What's New in 1.6.0

New York PHP Monthly Meeting
February 24, 2004

by Daniel Convissor
The Analysis and Solutions Company
Introduction to PEAR DB

- PEAR: PHP Extension and Application Repository
- Object oriented API
- Handles 13 of PHP's database extensions
  - dBase, FrontBase, InterBase, Informix, mSQL, MS SQL Server, MySQL, MySQLi,
  - Oracle, ODBC (tested DB2 & Access), PostgreSQL, SQLite, Sybase

Please Note:

- Sample code has been significantly simplified
- If viewing on the web: set browser to full screen / kiosk mode and change font size / zoom until this page fits comfortably in the whole screen
- Opera 7 Win32 users: our style sheet may cause rendering delays, even though it's valid
- Footer placement works right in Mozilla on multiple OS's, but IE and Opera only on Windows
Improvements in DB 1.6.0

- Really works for more than just MySQL.

- New portability features, making it possible to write applications which can be easily ported between DBMS's.

- Prepare/execute works the same way for all DBMS's and allows escaping of placeholder characters.

- PHP 5 compatibility.

- PHP must be at version 4.2.0 or higher.
Improvements in DB 1.6.0 cont'd

- Introduced *experimental* mysqli driver
- Deployed tableInfo() in more drivers and finalized the move from DB_result to DB_common.
- Added quoteSmart() and quoteSimple(), deprecating the inconsistent quote() and quoteString().
- Improved error reporting.
- Countless documentation corrections.
<?php
require_once 'DB.php';

// User, password and db variables defined elsewhere for obvious reasons.

$dbvars = array('mysql' => array(), 'oracle' => array(), 'pgsql' => array(), 'sqlite' => array());

function DB::connect($driver, $params = null)
{
    if (isset($driver)) {
        if (isset($params)) {
            if ($driver == 'mysql') {
                $mysql = & DB::connect("mysql://$muser:$mpw/$mdb");
                if (DB::isError($mysql)) {
                    die($mysql->toString());
                }
            }
            if ($driver == 'oracle') {
                $oracle = & DB::connect("oci8://$ouser:$opw@$odb");
                if (DB::isError($oracle)) {
                    die($oracle->toString());
                }
            }
            if ($driver == 'pgsql') {
                $pgsql = & DB::connect("pgsql://$puser:$ppw@$pdb");
                if (DB::isError($pgsql)) {
                    die($pgsql->toString());
                }
            }
            if ($driver == 'sqlite') {
                $sqlite = & DB::connect("sqlite:///$sdb?mode=0666");
                if (DB::isError($sqlite)) {
                    die($sqlite->toString());
                }
            }
        }
    }
}
?>
Portability: Lowercasing

```
require_once './examples/connect.inc';

$$dbms->query('CREATE TABLE tbl (Cf CHAR(10))');
$$dbms->query("INSERT INTO tbl VALUES ('DOH!')");
$$dbms->setFetchMode(DB_FETCHMODE_ASSOC);
$row =& $$dbms->getRow('SELECT Cf FROM tbl');

if (isset($row['cf'])) {
    echo 'mmm... sweet candy...';
} else {
    print_r($row);
}

$$dbms->query('DROP TABLE tbl');
```

Default Behavior

<table>
<thead>
<tr>
<th>$dbms = 'mysql'</th>
<th>$dbms = 'pgsql'</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Array ( [Cf] =&gt; DOH! )</strong></td>
<td><strong>mmm... sweet candy...</strong></td>
</tr>
</tbody>
</table>

```$dbms->setOption('portability', DB_PORTABILITY_LOWERCASE);```

<table>
<thead>
<tr>
<th>$dbms = 'mysql'</th>
<th>$dbms = 'pgsql'</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mmm... sweet candy...</strong></td>
<td><strong>mmm... sweet candy...</strong></td>
</tr>
</tbody>
</table>
require_once './examples/connect.inc';

$dbms->query('CREATE TABLE tbl (c CHAR(10))');
$dbms->query("INSERT INTO tbl VALUES ('one')");
c = "\$dbms->getOne('SELECT c FROM tbl');
if ($c == 'one') {
    echo 'Time for a Scooby Snack!';
} else {
    echo 'ZOINKS! \$c='$c' strlen=' . strlen($c);
}
$dbms->query('DROP TABLE tbl');

### Default Behavior

<table>
<thead>
<tr>
<th>$dbms = 'oracle';</th>
<th>$dbms = 'mysql';</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOINKS! $c='one       ' strlen=10</td>
<td>Time for a Scooby Snack!</td>
</tr>
</tbody>
</table>

$\$dbms->setOption('portability', DB_PORTABILITY_RTRIM);

<table>
<thead>
<tr>
<th>$dbms = 'oracle';</th>
<th>$dbms = 'mysql';</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time for a Scooby Snack!</td>
<td>Time for a Scooby Snack!</td>
</tr>
</tbody>
</table>
Portability: Null to Empty

```php
require_once './examples/connect.inc';

$dbms->query('CREATE TABLE WMDs (c CHAR(10) NULL)');
$dbms->query("INSERT INTO WMDs VALUES ('')");
$c = & $dbms->getOne('SELECT c FROM WMDs');

if (isset($c)) {
  echo 'Vote! And organize others to vote!';
} else {
  echo "$W's scruples: " . gettype($c);
}

$dbms->query('DROP TABLE WMDs');
```

### Default Behavior

<table>
<thead>
<tr>
<th>$dbms = 'oracle';</th>
<th>$dbms = 'pgsql';</th>
</tr>
</thead>
<tbody>
<tr>
<td>W's scruples: NULL</td>
<td>Vote! And organize others to vote!</td>
</tr>
</tbody>
</table>

```php
$dbms->setOption('portability', DB_PORTABILITY_NULL_TO_EMPTY);  
$dbms->query('CREATE TABLE WMDs (c CHAR(10) NULL)');
$dbms->query("INSERT INTO WMDs VALUES ('')");
$c = & $dbms->getOne('SELECT c FROM WMDs');

if (isset($c)) {
  echo 'Vote! And organize others to vote!';
} else {
  echo "$W's scruples: " . gettype($c);
}
```

<table>
<thead>
<tr>
<th>$dbms = 'oracle';</th>
<th>$dbms = 'pgsql';</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vote! And organize others to vote!</td>
<td>Vote! And organize others to vote!</td>
</tr>
</tbody>
</table>

---

Portability: Null to Empty

Portability: Right Trim

Table of Contents

< Portability: Null to Empty (Previous) Table of Contents (Next) Portability: Error Mapping >
Portability: Error Mapping

- MySQL unique/primary key violations:
  already exists -> constraint violation

- MySQL not-null violations:
  constraint violation -> null value violates not-null constraint

- MS Access ODBC bogus field:
  mismatch -> no such field

- In 1.6.0 this must be set during connect().

- 1.6.1 will allow configuration via setOption().
require_once './examples/connect.inc';

$dbms->query('CREATE TABLE tbl (c CHAR(10) NOT NULL)');
$result =& $dbms->query('INSERT INTO tbl VALUES (NULL)');

if ($result->getMessage() == 'DB Error: null value violates not-null constraint')
    { echo 'Phew, the pointy-haired guy went away!'; }
else
    { echo 'AAAAGH! ' . $result->getMessage(); }

$dbms->query('DROP TABLE tbl');

Default Behavior

<table>
<thead>
<tr>
<th>$dbms = 'mysql'</th>
<th>$dbms = 'oracle'</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAAGH! DB Error: constraint violation</td>
<td>Phew, the pointy-haired guy went away!</td>
</tr>
</tbody>
</table>

$dbms->setOption('portability', DB_PORTABILITY_ERRORS);

<table>
<thead>
<tr>
<th>$dbms = 'mysql'</th>
<th>$dbms = 'oracle'</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phew, the pointy-haired guy went away!</td>
<td>Phew, the pointy-haired guy went away!</td>
</tr>
</tbody>
</table>
Portability: Delete Count

require_once './examples/connect.inc';
$$dbms->query('CREATE TABLE tbl (c CHAR(10))');
$$dbms->query("INSERT INTO tbl VALUES ('one')");
$$dbms->query('DELETE FROM tbl');
$count = $$dbms->affectedRows();
if ($count != 0) {
    echo 'Yes!';
} else {
    echo 'Does everything have to be funny?';
}
$$dbms->query('DROP TABLE tbl');

Default Behavior

<table>
<thead>
<tr>
<th>$dbms = 'sqlite';</th>
<th>$dbms = 'pgsql';</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does everything have to be funny?</td>
<td>Yes!</td>
</tr>
</tbody>
</table>

$$dbms->setOption('portability', DB_PORTABILITY_DELETE_COUNT);

<table>
<thead>
<tr>
<th>$dbms = 'sqlite';</th>
<th>$dbms = 'pgsql';</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes!</td>
<td>Yes!</td>
</tr>
</tbody>
</table>
Portability options can be set when connecting...

```php
$options = array('portability' => DB_PORTABILITY_ALL);
$db =& DB::connect('pgsql:///foo', $options);
if (DB::isError($db)) {
    die($db->getMessage());
}
```

... or during runtime...

```php
$db->setOption('portability', DB_PORTABILITY_ALL);
```
Portability: Constants are Bitwised

- Portability mode constants are bitwised.
- Can combine them using `|` and remove them using `^`.
- Turn on all portability options:
  ```php
  $db->setOption('portability', DB_PORTABILITY_ALL);
  ```
- Enable lowercasing and trimming:
  ```php
  $db->setOption('portability', DB_PORTABILITY_LOWERCASE | DB_PORTABILITY_RTRIM);
  ```
- Enable all options except trimming:
  ```php
  $db->setOption('portability', DB_PORTABILITY_ALL ^ DB_PORTABILITY_RTRIM);
  ```
Portability: Backwards Compatibility

- Portability used to be handled by the `optimize` option.
- `optimize` is now deprecated.
- Old scripts work under the new system.

- If `optimize` gets set to `portability` the following drivers go into these modes:
  - `oci8`: `DB_PORTABILITY_LOWERCASE` and `DB_PORTABILITY_DELETE_COUNT`
  - `fbsql, mysql, mysqli, sqlite`: `DB_PORTABILITY_NUMROWS`

- If `optimize` gets set to `performance` new portability system is switched off.
Important tableInfo() Changes

- Finalized move to the DB_common class.
- Added to Sybase and Informix.

Examine a table by passing a table name:

```php
<?php
$info = $db->tableInfo('tablename');
print_r($info);
?>
```

Probe a query result by passing a DB_result object:

```php
<?php
$result = $db->query('SELECT * FROM tablename');
$info = $db->tableInfo($result);
print_r($info);
?>
```
Error Code Fixes

- **mssql**: `errorCode()` returns DB's code instead of SQL Server's code. Added `errorNative()` to get Server's code.

- Added mappings in several drivers to provide consistency between ibase, ifx, mssql, oci8, odbc(db2), pgsq1, sqlite and sybase.

- If error code portability is turned on, that consistency is expanded to mysql, mysqli and odbc(access).

- **Added** `DB_ERROR_CONSTRAINT_NOT_NULL` for handling null values in NOT NULL columns.
More Information About PEAR DB

- Manual
- Change Log
- Installation Instructions
- CVS:
  
  ```bash
  cvs -d:pserver:cvsread@cvs.php.net:/repository login
  # password is: phpfi
  cvs -d :pserver:cvsread@cvs.php.net:/repository co pear/DB
  ```