

## Using the Zend Framework Cache

Friday, 02 September 2011

It has bothered me for some time that I don't have some kind of caching in place for the PHP programs/sites that I work on. I ended up using the Zend Framework Cache. I think this is a great set of software that is easy to use and implement.

The steps are easy and simple:

- Download the Zend Framework library to your site. I just left it in the versioned name directory as downloaded. I had previously been bitten when I abbreviated such a download to the nice name, in this case 'ZendFramework'.
- Decide on a caching mechanism. I am using the File caching, so create a directory to store the cache. This is controversial: the tradeoff is disk access versus overhead in connecting to mysql. The O/S may have cached your file access. MySQL will likely cache your table access. I guess I just found it much easier to code the File access. Laziness has proven to be a bad design paradigm.
- Make the corresponding changes to your code.

Referring to the example coding below:

- Set the `include_path` to have a reference to the Zend Framework. I had thought about putting the library in a generic location on the server and referencing as needed. Also curious that there should be a way to use a CDN out there somewhere that will provide this access.
- include the specific parent object we want to use, in this case `Cache.php`
- tell the cache how to setup. ZF uses the concept of a frontend and a backend. I really don't see the point of splitting these up. It always irks me when people use fine grain philosophical points when coding. Java is rife with this.
- I think I am using the most common configuration options: `lifetime`, `auto serialize` and `cache_dir`.
  - `lifetime` can be set to null to store all of your cache items forever.
  - `auto serialize` needs to be specifically set. I am curious how often just Strings are stored in a cache.
  - `cache_dir`. Where to put the cache on disk. Again, also curious if you had several sites accessing the same data this could be in a common location and shared amongst the caches?
- store something in the cache
- retrieve it

```
<?php
```

```
set_include_path('/:homepages/...../ZendFramework-1.11.10-minimal/library');
```

```
include ('Zend/Cache.php');
```

```
$frontendOptions = array(
    'lifetime' => 7200, // cache lifetime of 2 hours - null==forever
    'automatic_serialization' => true //needed for non-string data
);

$backendOptions = array(
    'cache_dir' => '/homepages/...../tmp/' // Directory where to put the cache files
);

// getting a Zend_Cache_Core object
$cache = Zend_Cache::factory('Core',
    'File',
    $frontendOptions,
    $backendOptions);

$cache->save(1, 'test');

$test = $cache->load('test');

var_dump($test);

?>
```

My next steps:

- Simple standard steps for loading from cache:

```
-- $value = $cache->load($name)
```

```
-- if ($value===false) load from database and store in cache
```

- I think most of my cache items are always needed and will not change, lookup tables and the like for pull-down lists. I like the idea of setting the lifetime to null - meaning never update this cache item.

- I have other items that may change slightly. This is a little more difficult. How to clear just those items and reload the cache for them. A couple of articles point to using a version number in the name of the cache item. So, something like:

- make sure we have a cache item named 'version' in the cache starting with '1'. It lives forever.

- when we change something in the cache
- clear the current items in the cache with the current version
- update the version in the cache
- change the naming used in storing items in the cache to include the current version