

Fixing PHP namespaces in Drupal 8

Jonathan Brown, Blue Droplet Media

<http://bluedroplet.com/>

Problem

PHP function name collisions

Collisions with other PHP libraries

namespace drupal;

**every function (and class) is prefixed with
drupal**

drupal_set_message()

declare as set_message()

implies drupal\set_message()

called as set_message()

Larry Garfield's new Database API

not really part of Drupal

- standalone PHP library

```
namespace drupaldb;
```

```
namespace drupal;
```

```
$query = \drupaldb\db_select('node');
```

```
namespace drupal;  
use drupaldb as db;
```

```
$query = db\select('node');
```

Modules (and themes)

taxonomy_select_nodes()

hook implementations

ambiguity of where the module name ends and the hook name begins, e.g.

views_comment_load() - is the hook
comment_load
or **load** ?

```
namespace drupal\taxonomy;  
use Drupaldb as db;
```

```
select_nodes()
```

\drupal\set_message()

\drupal\taxonomy\select_nodes()

```
namespace drupal\mymodule;  
use drupal, drupaldb as db;  
use drupal\taxonomy;
```

```
drupal\set_message()  
taxonomy\select_nodes()
```

PHP 5.3 required - released 2009-06-30

Drupal 7 released this year?? wouldn't bet on it

how long til D8 released? 2012 / 2013?

PHP 5.3 will be 3/4 years old

Private / public functions

_node_rankings()

_rankings()

```
/**  
 * @public  
 */  
function select_nodes() {
```

coder module
api.drupal.org

Hook collisions

**module functions can inadvertently become
hook implementations for other modules**

**multiple modules can 'invent' the same hook,
expecting it to do different things**

use

INVENTOR__HOOK()

e.g.

system__menu()

block__info()

Dynamic hook names

MODULE_form_FORM_ID_alter()

becomes

system__form_alter__FORM_ID()

e.g.

system__form_alter__search_form()

```
/**  
 * @implements system__form_alter__FORM_ID()  
 */  
function system__form_alter__search_form() {
```

**the function name is the same in every module
that implements the hook**

- helps code portability

Theme implementations

the module that declares a theme hook in a **system__theme()** is considered the inventor of that hook

theme_user_list(\$variables)
becomes **user__theme__list(\$variables)**
in both the module that invented it and in themes that are overriding it