

# Hook naming in Drupal



Klaus Purer (klausi)  
<http://drupal.org/user/262198>

# What's the problem?

- Hooks are a great way to extend Drupal
- BUT: no conventions for hook naming
- Completely unrelated function definitions may implement a hook by accident
- Thousands of contributed modules that expose hooks themselves and may crash with each other
- A crash easily leads to a totally broken Drupal site

# Solution 1: Add "hook" keyword

- Convention: every hook needs to have the word "hook" in its name
- Now: `mymodule_node_update()`
- Solution: `mymodule_hook_node_update()`
- Good: more awareness in the function name that this will be called as hook
- Bad: function names get longer, not bullet-proof as crashes may happen nevertheless

# Solution 2: annotations

- Add "@hook" to the comment above a function definition (as requirement)

- Now:

```
/**  
 * Implements hook_node_update().  
 */  
function mymodule_node_update() {}
```

- Solution:

```
/**  
 * @hook node_update()  
 */  
function mymodule_node_update() {}
```

# Solution 2: annotations (2)

- Good: no crashes! (you don't write @hook in the comments of your "normal" functions)
- Bad: does not solve the problem of contributed modules exposing hooks with the same name

# Solution 3: info file

- All modules that expose hooks have to state them in the info file
- All modules that implement hooks have to state them in the info file
- Good: Drupal can easily detect naming collisions (2 modules exposing the same hook name), no accidental implementations
- Bad: ?