

OSDPL Developer's Primer

! Work in progress.

! Externals are not supported by git.
This strategy will have to be adjusted to work with git.

This Document

This guide is intended to help orient a developer on the customizations and structure of the Open Source Design Pattern library as implemented in Drupal 5.x.

This guide assumes you have some understanding of Drupal or plan on learning it yourself. As such, this document will not cover topics that can be learned elsewhere, instead will focus on OSDPL specific details in Drupal.

Getting a copy of the OSDPL

1. Download a copy of Drupal 5.x from Drupal (<http://www.drupal.org>). At time of writing this, the OSDPL is running v5.15.
2. Once you have Drupal running and installed on a server, get the OSDPL configuration from the Fluid Project github repository
 - Get the OSDPL here: <https://github.com/fluid-project/osdpls>
 - All necessary modules will be downloaded.

The OSDPL Repo Structure

The following is an outline of the repository structure for the OSDPL:

```
./drupal-5.x
  /files           (directory for all uploaded user content)
  /sites
    /default
      /modules     (Contains external references to modules hosted
                   elsewhere or in the ./vendor directory. Does not
                   actually contain any physical files at this time,
                   just references to other resources.)
      /themes      (Contains the theme currently used and modified
                   for the OSDPL)
./vendor
  /drupal-contrib (Contains specific versions of contributed modules
                   used by the OSDPL and not hosted in an SVN repository
                   elsewhere. Externals properties in
  /akismet        /5.x-1.1
  /cck_blocks     /drupal-5.x/sites/default/modules reference these files.)
  ...
```

SVN repositories for Drupal and modules:

- <http://subversible.com/svn/> (Preferred source)
- <http://svn.advantagelabs.com/managed/vendor/> (Secondary source)
- <http://source.fluidproject.org/svn/fluid/pattern-cms/trunk/vendor/> (Managed here if above 2 sites do not have the files required)

Contributing your changes back to the OSDPL

Work done to the OSDPL can be contributed back using two methods:

1. If you do not have commit access to Fluid's OSDPL project repo, create a patch and submit it to the Fluid-Work mailing list for review.
 - If the patch addresses a known bug, also attach the patch to the bug and reference the bug # when emailing the Fluid-work list.
 - To gain Push access to Fluid's OSDPL repo, please see [Process for Granting Commit Access](#).
2. If you have push access to Fluid's OSDPL repo, simply push it up to the repository.

Task list / Bug tracking

All OSDPL related bugs and tasks are tracked in the Fluid Project Jira bug tracker. If you are looking for things to do, this is a good place to start.

- [OSDPL bugs and tasks on Fluid Project Jira](#)

Regarding updating modules and Drupal versions

Caution is required if updating Drupal core or contrib modules, and you will also need commit access to the Fluid OSDPL repository. (If you do not have commit access, then please send a message to the Fluid-work mailing list requesting the update to Drupal).

Using the Repo structure above as a reference:

1. Determine if the module that requires updating is versioned in one of the two public 3rd party SVN repositories.
 - If present in 3rd party repository, then update the Externals property of `./drupal-5.x/files/sites/default/modules` to reference the new version.
 - If not managed in a 3rd party repository, then commit the updated module into <https://github.com/fluid-project/osdpl/tree/master/vendor> following the naming convention laid out for existing modules.

To summarize, Fluid should only manage a Drupal module only if that module does not appear in a SVN repository elsewhere.

Key modules

- **CCK** - Custom entry form for design patterns.
- **CCK Fieldgroup Tabs** - Helps divide the CCK form into tabs to make it more digestible to the user.
- **CCK Validation** - Overrides the default error handling and replaces it with better error messages that match tab names.
- **Panels** - Controls the information displayed of many of the listing pages as well as the layout of the OSDPL front page.
- **Views** - For listing design patterns belonging to certain categories (Taxonomies) and/or workflow states.
- **Workflow-NG** - A design pattern can take on different states. Depending on the state, different access and visibility is granted.

Design Pattern Workflow and Privileges

Design patterns, when authored, can take on 4 different states selected using the CCK form: Private Draft, Open for Comments, Request for Review, and Published. These states are defined using the Workflow_NG module and the different actions taken for each state change can be seen by accessing the Rule Configurations.

Content states

Private Draft: The node is only visible to the content author and administrators. No one else is able to view or edit.

Open for Comments: Registered users or higher can view and comment on the design pattern. Anonymous / unregistered users will not be able to see this content.

Request for Review: Viewable and editable by Pattern Editors or higher. Registered users and anonymous users will not be able to see this content.

Published: Viewable by everyone.

Actions

Four basic actions are done for each event fired in the Workflow:

1. Set the machine state.
2. Set the permissions for the content.
3. Publish/Unpublish the content.
4. Redirect the User.

Items 1 - 3 help implement the access rules for the given state.

Note: When switching states to "Comments", "Review Request", or "Published", the content is actually published in Drupal (i.e. node status is set to 1). This is so that registered users can comment if in the "Comments" state, and pattern editors view and edit in the "Review Request" state.

Related Modules

- Content Access - Gives Workflow_NG control over node access.

Customized Design Pattern CCK template

ImageField

uidp Content Type display rendering

Viewfield

Managing Views, Blocks, and Panels

Navigation, menus, and breadcrumbs

Other high-priority tasks