Build a Site Event Calendar

Release 0

Melissa Anderson

December 31, 2011

CONTENTS

CHAPTER

ONE

BUILD A SITE EVENT CALENDAR

1.1 Ingredients

1.1.1 Staple

1. http://drupal.org/project/ctools - Enable Chaos tools
2. http://drupal.org/project/views - Views: Enable Views and Views UI
3. http://drupal.org/project/date - Date: Enable Date, Date API, Date Repeat, Date Popup
4. http://drupal.org/project/fullcalendar - FullCalendar: Enable FullCalendar, FullCalendar Colors
5. http://drupal.org/project/colorsrs API - Other: Enable Colors (install the dev version)
6. http://drupal.org/project/colorboxbox - Other: Enable Colorbox

Drush users:

```
drush dl ctools views date date_api fullcalendar fullcalendar_colors colors colorbox
drush en ctools views views_ui date date_api date_repeat date_popup
fullcalendar fullcalendar_colors colors colorbox -y
```

1.1.2 Specialty

FullCalendar JQuery Libraries

1. Visit http://arshaw.com/fullcalendar
2. Download the most recent version of the plugin.
3. Unzip fullcalendar/fullcalendar to sites/all/libraries/fullcalendar (e.g., sites/all/libraries/fullcalendar/fullcalendar.min.js).
4. Do no include the demos or jQuery directories.

ColorBox Library

1. Visit http://jacklmoore.com/colorbox/
2. Download Current version
3. Unzip it straight into sites/all/libraries (e.g., sites/all/libraries/colorbox)

1.2 Part 1: Build a way to create events

1.2.1 1.1: Add a new content type called Event

Structure > Content types > +Add new content type

Name: Event

Description: Create an event for display on the site calendar

Submission Form Settings	Title field label: Event
Publishing Options	[] Published (Check only this option; uncheck others)
Display Settings	[ ] Display author and date information (Uncheck this)
Comment Settings	Closed (Select Closed)
Menu Settings	[ ] Uncheck all menus

1.2.2 1.2: Configure Region and Language Settings

Configuration > Regional and language > Regional settings

The module and status pages will show warnings and display instructions for initially configuring Date module.

Regional Settings Locale

1. Default country: United States
2. First day of the week: Sunday
3. [ ] Use ISO-8601 week numbers (Leave unchecked)

Time Zones

For this recipe, the calendar represents events happening in a single geographical location, so we won't be using timezone settings.

1. Default time zone: America/Los Angeles
2. [ ] Users may set their own timezones (Uncheck this)

1.2.3 1.3 Configure Date and Time

Configuration > Regional and language > Date and time

Initial Date Module Configuration

1. Select formats for Long, Medium, or Short (Accept the defaults)
2. Click Save configuration. This provides the initial configuration for the date module and removes the warning on the modules page.

Add a new date format

1. Choose the Formats tab (upper right)
2. +Add format
3. Format string: d M Y
4. Save with: Add format
5. Choose the Types tab (upper right)

6. +Add date type
7. Date type: Block-short <- this is just a label, you can choose any formatting you like.
8. Date format: Choose the example of the new format from the bottom of the select list: 14 Nov 2011
9. Save with: Add date type

Note: If you make a mistake with the format but don't notice until later, you can edit a date format by returning to *Configuration > Regional and Language: Date and Time > Formats* and clicking the edit link for the format in question. Unfortuantely this will cause the format to be unassigned from its Date type. Visit the Types tab and re-select the format from the dropdown list where appropriate. If you forget this last step, the date field won't be automatically updated throughout your site.

1.2.4 1.4: Add a date field

Structure > Content types > Event > Manage fields Add new field

1. Label: Date
2. Field name: date
3. Type of data to store: Date <- plain Date, not the ISO or Unix format
4. Form element to edit the data: Popup calendar with repeat options

Field Settings

1. Date attributes to collect
2. [] Check Collect an end date
3. [] Required field
4. Accept other default field settings
5. Save.

Event Settings

1. [] Check Collect an end date
2. Accept the rest as is.
3. Save settings.

1.2.5 1.5: Rarrange the items in a way that makes sense

Structure > Content types > Event > Manage fields

Use the grabber to drag and drop the fields into order:

1. Event name
2. Date
3. Body
4. URL path settings.

Remember to Save.

1.2.6 1.6: Create proper paths

Configuration > Search and Metadata: URL aliases > Patterns

Later on, we'll be creating a landing page at /calendar, so we're adding that to the path now.

1. Pattern for all Event paths: calendar/[node:title]

1.2.7 1.7: Create an event

Content > +Add content

1. Add an event on today called Training
2. Check that the path is as you expected, e.g. http://eventcalendar.everydaydrupal.com/calendar/training-opensourcery

1.3 Part 2: Create the calendar

1.3.1 2.1: Add a calendar view

Structure > Views > +Add new view

1. View name: Calendar
2. Show content of type Event sorted by Unsorted
3. [] Create a page
4. Page title: Calendar
5. Path: calendar
6. Display format: FullCalendar
7. Items to display: (ignore)
8. [ ] Use pager
9. [] Create menu link:Menu: Main menuLink text: Event calendar
10. [] Include an RSS feed
11. Feed path: calendar.xml
12. Continue & edit

2.1.1 Add the date field

You'll see an error message as soon as you've clicked Continue & edit. You have to add the Date field from your event to the view for the calendar to function.

In the fields section of the left-hand column:

Fields

1. add:

   Content: Date Appears in: node:event.
2. [ ] Remove the check in the box Create a label

3. It's your choice what you set in the other visible areas. Don't change anything in the collapsed field groups.
4. Apply (All displays)

2.1.2 Advanced

Other

For the drag and drop functionality to work when you switch pages, you MUST expancd the Advanced fieldgroup and set Ajax to Yes.

2.1.3 Configure the feed

We want a feed icon to appear on the page with the calendar so that feed users can find the feed URL.

If you create the feed on the initial set screen, this will be set up automatically, but if you create the feed later, you'll need to set it up manually.

In the center column of the view's Feed display, under Feed settings, choose:

> Attach to:
>
> [] Master
>
> [] Page

1.4 Part 3: Create and place an Upcoming events block

We'll create an Upcoming events block for the front page and our calendar pages. We could build this in the same view as the calendar, but since it's substantially different, we'll create it in its own view.

1.4.1 3.1: Create the block

Structure > Views > +Add new view

1. View name: Upcoming events
2. Show content of type Event sorted by Unsorted <- Choosing Newest first here would sort by when the Event node was created, not when the event was scheduled.
3. [ ] Create a page
4. [] Create a block
5. Display fomat: Unformatted list of fields
6. Continue & Edit

3.1.1 Add the date field

Fields

1. add:
2. Content: Date
3. Appears in: node:event.

4. [ ] Remove the check in the box Create a label (Remove check)
5. [ ] Exclude from display (Leave unchecked)
6. Formatter:
7. Date and time
8. Choose how users view dates and times:
9. Block-short
10. Display: Both Start and End dates
11. Don't change anything in the collapsed field groups.
12. Apply (All displays)

3.1.2 Filter to future dates

I filter on the date an event ends so that it remains visible while it's happening, allowing users to reference directions or other information quickly. You might choose to filter by the Start date if it better suits your needs.

Filter Criteria

1. add:
2. Content: Date - end date (field_date:value2)
3. Configure extra settings for filter criterion:
4. Accept the defaults. They won't matter.
5. Configure filter criterion:
6. Operator:
7. Is greater than
8. Enter a relative date
9. Relative date:
10. now

3.1.3 Sort by date

1. add:
2. Content: Date - start date (field_date)
3. Click Add and configure
4. Choose Sort ascending (default)
5. Apply
6. Don't forget to save the view, too!

1.4.2 3.3 Place the block

Structure > Blocks > Views: Upcoming Event > Configure

Block title:Upcoming events

Region Settings: Bartik (default theme) Sidebar Second

Visibility Settings	
Pages	Only the listed pages (Select this, enter the text below) • <front> • calendar/*
Content types	[] Gallery (Check only this option)
Roles	(Leave as is)
Users	(Leave as is)

1.5 Part 4: Categorize events by terms

1.5.1 4.1 Create the vocabulary and add terms to it

Structure > Taxonomy > +Add vocabulary

1. Name: Event Types
2. Description: Categorize events by type.
3. Add two terms (for the Category Event Types
4. Name: Public
5. Description: These events are open to the public
6. Accept the rest of the defaults and save.
7. Name: Staff Meeting
8. Description: These meetings are required meetings for all staff.
9. Accept the rest of the defaults and save.

1.5.2 4.2 Add the term to the Event content type

Structure > Content Types > Event > Manage Fields

1. Add new field
2. LABEL: Add new field: Event type
3. NAME: field_event_type
4. FIELD: Term reference
5. WIDGET: Select List
6. Save
7. Make sure the Event Types category is selected.
8. Save field settings

9. Accept all the defaults
10. Save settings
11. Drag the Event Type term beneath the Event name.
12. Save.

1.5.3 4.3 Create and tag events

Content > +Add Content

1. Create an event in the current month tagged Public
2. Create an event in the current month tagged Staff Meeting
3. Create an event in the current month and don't tag it.

1.6 Part 5: Color code events

Configuration > User Interface: FullCalendar > Colors (tab) > Taxonomy (subtab)

You can color code your events by Node Type, Taxonomy, or User Role.

1.6.1 5.1 Set the default color

1. Choose the color for events that aren't labeled with a term.

1.6.2 5.2 Set the colors for the taxonomy terms

1. Expand Event types.
2. Use the color picker to assign different colors.
3. Save configuration

1.6.3 5.3 Check out the colors. Try dragging. Try dropping.

1.7 Part 6: Create filters based on your categories

There are several ways to maintain a master calendar and then filter it so that you show only certain events.

First, we'll add a filter to the main calendar so users can choose what to see.

1.7.1 6.1: Add drop down filter to the existing calendar

Structure > Views

1. Locate your calendar on the main views page or use the contextual links while viewing it to edit.
2. Filter Criteria

   add Choose Content: Has taxonomy term

3. Add and configure filter criteria
4. For This page (override)
5. [ ] Expose this filter to visitors, to allow them to change it
6. Label: Choose a category
7. Apply (this display)

1.7.2 6.2 Create a pre-filtered option

We're going to filter the calendar so that it only shows one kind of event and the user cannot change this. Organizations with complex structures may prefer to shield their users from the selection porcess and choose instead to place these calendars in department areas.

1. Clone the page display
2. Change the Display name to "Filtered by Page" for your own convenience
3. Path: calendar/%
4. Advanced
5. Contextual Filters

   add Content: Has taxonomy term ID
6. In the configuration for the filter:
7. For: This page (override)
8. When the filter value is NOT in the URL

   Provide default value Type: Taxonomy term ID from URL <- Choose from the select list.
9. [ ] Uncheck Load default filter from term page
10. Click Add (This Display). Note: If the button says Apply (all displays), you missed step 1. You can make that change now.

You might wonder why we've created a second display instead of just adding the contextual filter to the first Page display and putting the % wildcard at the end of its Path.

The wildcard prevents putting the calendar on the menu, so we had a choice. We could add all of the calendar links to the menu manually and create a single display, or we can create the main display and duplicate it. I chose the second because I wanted the main calendar view to contain the drop down filter that I did not want to place on the other views.

1.7.3 6.3 Add our filtered views to the main menu

Structure > Menus > Main Menu > add link

Because we're working with a simple test site, we'll add menu entries for our event types to the main menu. If you need to verify the taxonomy id of a term you can Structure > Taxonomy > Event Types, list terms, and the Term ID is visible in the URL of the edit link (taxonomy/term/#)

1. Menu link title: Public Events
2. Path: calendar/1
3. Save
4. +Add link

5. Menu link title: Staff Meetings
6. Path: calendar/2
7. Save

Using terms vs. term names: Term names are better, really, because they're more memorable. If the name is changed, however, then all the links using the name have to be updated or redirects must be put into place.

1.8 Part 7: Set and test permissions

People > Permissions

If you are not using the Test Kitchen Install Profile or if you are new to the idea of users, roles, permissions or masquerade, see http://training.opensourcery.com/basics

1.8.1 7.1 Set permissions

Set permissions as follows:

Author	**Editor**	**Admin**
[] Event: Create new content	[] Event: Create new content	[] Event: Create new content
[] Event: Edit own content	[ ] Event: Edit own content	[] Event: Edit own content
[ ] Event: Edit any content	[] Event: Edit any content	[] Event: Edit any content
[] Event: Delete own content	[ ] Event: Delete own content	[] Event: Delete own content
[ ] Event: Delete any content	[] Event: Delete any content	[] Event: Delete any content

1.8.2 7.2 Test Author privileges

Masquerade as Test Author and ensure you CAN:

1. Create an Event
2. Edit that Event
3. Delete that Event

Ensure you CANNOT:

1. Edit galleries you didn't create
2. Delete galleries you didn't create

When you're done, remember to Switch back

1.8.3 7.3 Test Editor privileges

Masquerade as Test Editor and ensure you CAN:

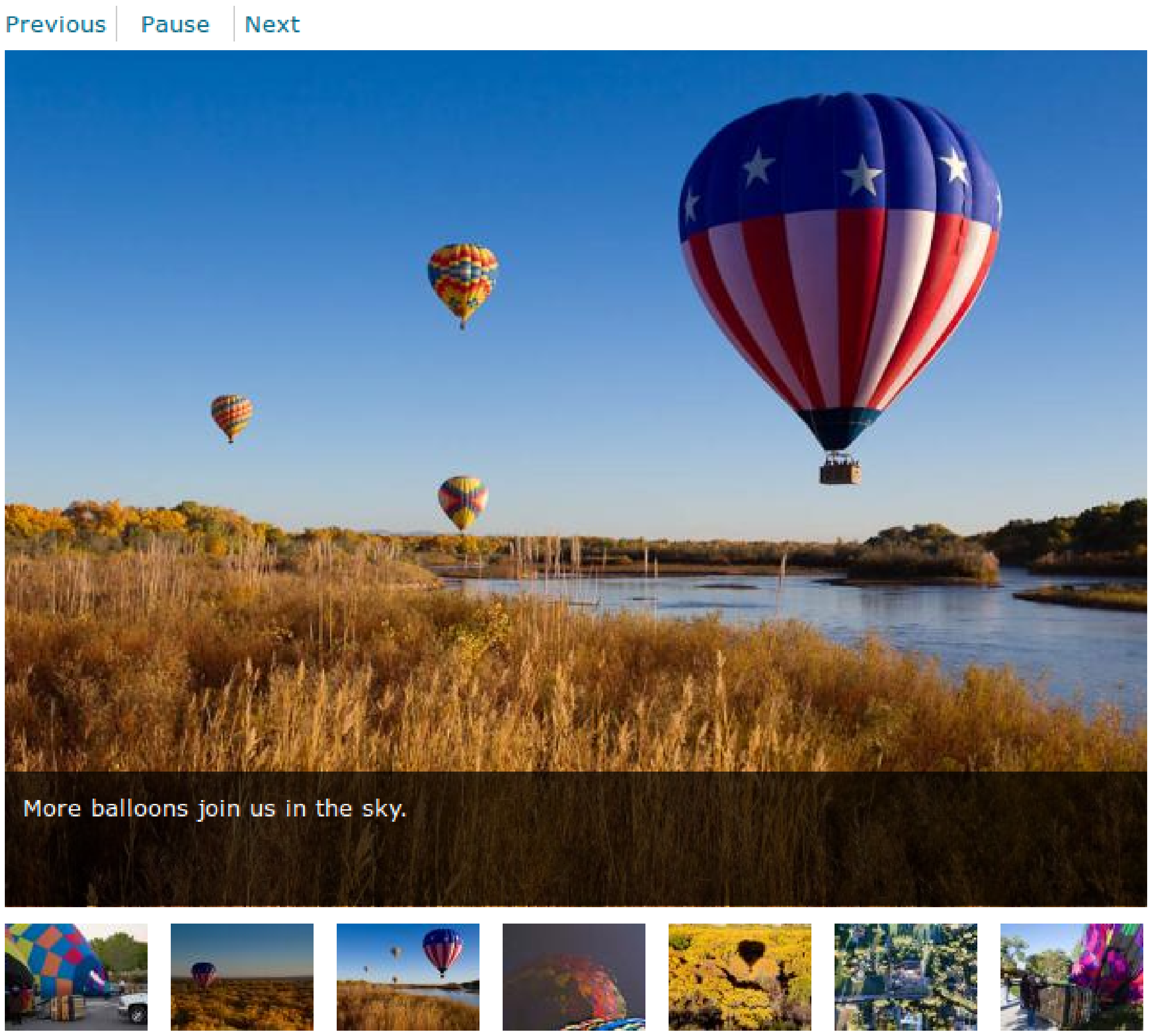
1. Create an Event
2. Edit that Event
3. Delete that Event
4. Edit an Event you didn't create
5. Delete an Event you didn't create

CHAPTER

TWO

BUILD IMAGE GALLERIES AND SLIDESHOWS

Create photo galleries that elegantly display photos and titles with optional slide show viewing controls. Upload an image just once and have a variety of high quality sizes created without ever opening a graphics program. Then, bring them all together into a beautiful index that groups them by category!



2.1 Ingredients

2.1.1 Staple

1. http://drupal.org/project/ctools - Enable Chaos Tools (all others are not needed)
2. http://drupal.org/project/views - Enable Views and Views UI
3. http://drupal.org/project/libraries - Enable Libraries
4. http://drupal.org/project/views_slideshow - Enable Views Slideshow

Drush users:

```
drush dl ctools views libraries views_slideshow
drush en ctools views views_ui libraries views_slideshow views_slideshow_cycle -y
```

2.1.2 Speciality

Views Slideshow module uses two external libraries.

Required for basic slide show functionality:

1. Visit http://malsup.com/jquery/cycle/download.html
2. Click the link to "Cycle Plugin"
3. In your browser, choose File > Save Page as > jquery.cycle.all.js
4. Place in sites/all/libraries/jquery.cycle (you'll have to create that folder) so that the final path looks like sites/all/libraries/jquery.cycle/jquery.cycle.all.js

Required for advanced features, which include setting the speed of the slide show:

1. Visit https://github.com/douglascrockford/JSON-js/downloads and download the appropriate format, .zip or .tar.gz, for your environment.
2. Extract it
3. Rename the directory (which begins with douglascrockford-) to json2
4. Place it in sites/all/libraries so that the final path looks like: sites/all/libraries/json2/json2.js

2.2 Part 1: Build the form for creating galleries

2.2.1 1.1: Add a new content type called Gallery

Structure > *Content types* > *+Add new content type*

Name: Gallery

Description: Create an image gallery with slide show controls.

Submission Form Settings	Title field label: Gallery name
Publishing Options	[] Published (Check only this option; uncheck others)
Display Settings	[ ] Display author and date information (Uncheck this)
Comment Settings	Closed (Select Closed)
Menu Settings	[ ] Uncheck all menus

2.2.2 1.2: Add an image field

Structure > *Content types* > *Gallery* > *Manage fields*

Add new field

1. Label: Gallery Image
2. Field name: gallery_image
3. Type of data to store: Image
4. Form element to edit the data: Image <- the default
5. Save

Field Settings

Leave as is: Public files selected, no default image

Gallery Settings

There are a lot of settings here! Accept the defaults EXCEPT the following:

File directory: galleries

Minimum image resolution: 640 x 480

We're building our gallery around a common 640 x 480 resolution. By requiring images to be at least that large here, we can prevent jarring changes in size and eliminate white space between main images and thumbnails.

[] Enable Title field

Gallery Image Field Settings

Number of values: Unlimited

2.2.3 1.3: Create proper paths

Configuration > Search and Metadata: URL aliases > Patterns

Later on, we'll be creating a landing page at /galleries, so we're adding that to the path now.

Pattern for all Gallery paths: galleries/[node:title]

2.2.4 1.4: Create a test gallery

Content > +Add content > Gallery

1. Gallery name: Test Gallery
2. Body text: Optional
3. Browse and upload at least 4 images, giving each one a title.
4. Check the URL pattern is what you expect, something like: http://tests.l/galleries/test-gallery

2.2.5 1.5: Hide the default display of images

Structure > Content types > Gallery > Manage display

1. Set the Default display setting format for Image to Hidden.
2. Verify the Teaser display format is hidden. It should already be set that way.
3. Go to Content and view the gallery you created in 1.4
4. You should see nothing but body text you entered.

2.3 Part 2: Create custom image sizes

2.3.1 2.1: Main gallery image

Configuration > Media > Image styles > +Add style

We set a minimum resolution for uploading the image, but users can upload higher resolution if the wish. Using this style ensures uniform presentation within the gallery.

1. Image style name: gallery_main
2. In Effect, select the new effect Scale and crop from the drop down, then click Add
3. Width: 640
4. Height: 480
5. Click the Add effect button (Your changes are saved; the button on the next page is just for reordering the effects if there is more than one.)

2.3.2 2.2: Gallery thumbnails

Configuration > Media > Image styles > +Add style

The thumbnail settings are chosen so the images stay proportional to an 640 x 480 main image and so seven thumbnails fit width-wise underneath it, with an allowance for padding.

1. Image style name: gallery_thumb
2. In Effect, choose Scale and crop, then click Add
3. Width: 80
4. Height: 60
5. Click the Add effect button

2.3.3 2.3: Gallery index thumbnails

Configuration > Media > Image styles > +Add style

Add a third style for the index of galleries on the site.

1. Click +Add style again
2. Image style name: gallery_index
3. In Effect, choose Scale and crop, then click Add
4. Width: 160
5. Height: 120
6. Click the Add effect button

2.4 Part 3: Create the galleries

Views delivers extraordinary power to the non-programmer, and the price is a densely-packed interface. We describe the steps below, but there's a place where the screen cast is worth a thousand words!

2.4.1 3.1: Create the actual gallery display

Structure > Views > +Add new view

On the introductory Views page:

1. View name: Gallery
2. Show content of type Gallery sorted by Newest first
3. [ ] Uncheck Page
4. [] Check Block
5. Block title: Gallery
6. Display fomat: Slideshow of fields
7. Items per page: (Make this blank)
8. Continue & Edit

In main Views interface, we'll configure three areas:

3.1.1 - available fields
3.1.2 - row style settings,
3.1.3 - the contextual filter.

We'll set them up in this order because they depend upon each other, even though they're visually in a different order.

3.1.1: Add Fields

First, add the main gallery image:

1. Fields: add
2. From the popup, select: Content: Gallery image
3. [ ] Remove the check in the box Create a label
4. Set the Image style to gallery_main.
5. Multiple Field Values: [ ] Uncheck Display all values in the same row
6. Apply (All displays)

Second, add the thumbnail gallery images. Much like the first step:

1. Fields: add
2. From the popup, select: Content: Gallery image
3. [ ] Remove the check in the box Create a label
4. [] Check Exclude from display
5. Set the Image style to gallery_thumb.
6. Multiple Field Values: [ ] Uncheck Display all values in the same row
7. Apply (All displays)

Next, add the title.

1. Fields: add

2. From the popup, select:Content: Gallery image
3. [ ] Remove the check in the box Create a label
4. Leave other visible settings at their default
5. Multiple Field Values: [ ] Uncheck Display all values in the same row
6. Expand Rewrite Results
7. [] Rewrite the output of this field
8. In the text field, enter [field_gallery_image_2-title] (View the available patterns by expanding Replacement Patterns.)
9. Apply (all displays)

You now have three fields, all named the same thing but configured differently: the main image, the thumbnails, and the image titles.

Finally, remove the Content Title since it redundantly displays the name of the gallery.

1. Click the link Content: Title
2. Scroll and click Remove

3.1.2: Format

1. Format: Settings
2. In the Top Widgets section, check Controls.
3. In the Bottom Widgets section, check Pager and choose the middle instance of Content: Gallery Image. This will provide the thumbnail images for the pager.
4. Apply (All displays)

3.1.3: Advanced

1. Contextual Filter (Add)
2. Content: Nid
3. Provide a default value > Type: Content ID from URL
4. Apply (All displays)

Be sure to save the view!

2.4.2 3.2: Place and configure the block

Structure > Blocks > Views: Galleries > Configure

Configuring the block to display only on Gallery nodes prevents it from being called on a view, and listing it only on specific pages prevents it from appearing on the Edit tab.

Block title: <none>

Region Settings

Footheme (default theme) Content

Visibility Settings

Pages	Only the listed pages (Select this and enter the text below) galleries*
Content types	[] Gallery (Check only this option)
Roles	(Leave as is)
Users	(Leave as is)

2.5 Part 4: Create an index of all galleries

Structure > Views > +Add new view

You could create the gallery index page in the same view as the gallery block by adding a Page display, but the settings are different enough that they don't gain a lot by sharing defaults, so we'll create this as a separate view.

2.5.1 4.1: Add a new view

Intro screen

1. View name: Galleries
2. Show Content of type Gallery sorted by Newest first
3. [] Create a page
4. Page title: Galleries
5. Display format: Grid of fields
6. Items to display: 24
7. [] Create a menu link > Menu: Main menu, Weight: 1
8. Menu link title: Galleries
9. Continue & edit

Fields

1. add
2. Gallery image
3. [ ] Uncheck Create a label
4. Formatter: Image (no change)
5. Image style: gallery_index
6. Link image to: Content
7. Apply (all displays)

Be sure to save the view!

2.5.2 4.2: A brief detour

Structure > Blocks > Main menu: configure

Footheme uses the block system to place menus, and the main menu is not enabled yet.

Region Settings

1. Footheme (default theme). Select Menu Bar.
2. Save block

2.5.3 4.3: Auto-generate galleries

Configuration > Development: Generate content

1. [] Gallery (Uncheck others)
2. Accept the rest of the defaults
3. Generate

2.6 Part 5: Style the Gallery

We'll apply some basic formatting to the galleries.

2.6.1 5.1: Open footheme.css

You can edit your style sheet any way that is comfortable for you. If you need more support than the username and password for your sandbox, see Connecting to your sandbox with sFTP at http://training.opensourcery.com/basics/sftp

2.6.2 5.2: Arrange thumbnails beneath the main image

```
/* Arrange the thumbnails beneath the main image */
.views-content-field-gallery-image {
  float: left;
  padding-right: 13px;
}

.views-slideshow-cycle-main-frame-row-item {
  padding: 5px 20px;
}

.views-slideshow-controls-bottom {
  padding: 0 0 0 20px;
}
```

2.6.3 5.3: Format the controls

```
/* Format the controls */
.views-slideshow-controls-text {
  padding-left: 20px;
}
```

```
.views-slideshow-controls-text span {
  display:  block;
  float: left;
}

.views_slideshow_controls_text_resume,
.views_slideshow_controls_text_pause{
  text-align: center;
  width: 55px;
  border-left: 1px solid #ccc;
  border-right: 1px solid #ccc;
  padding: 0 5px;
  margin: 0 5px;
}
```

2.6.4 5.4 Place title text over the main image

Structure > Views > Gallery: edit

Placing opaque text on a transparent background requires some special setup. If the text is placed inside the transparent container, it will inherit the transparency. Instead, it must be placed parallel to the container and positioned.

1. Edit the third instance of the filed content: Galley image
2. Expand REWRITE RESULTS and edit to match:

   ```
   <div class="transparency"></div>
   <div class="overlay">[field_gallery_image_2-title]</div>
   ```

3. Apply (all displays)
4. Add the styling to footheme.css:

   ```
   /* Overlay the text on the main image */
   .views-field-field-gallery-image {
     position: relative;
   }

   .transparency {
     position: absolute;
     bottom: -10px;
     left: 0px;
     width: 640px;
     height: 75px;
     background: black;
     margin: 20px;
     filter:alpha(opacity=70);
     opacity: 0.7;
     -moz-opacity:0.7;
   }

   .overlay {
     color: white;
     position: absolute;
     bottom : 0px;
     left: 0px;
     height: 75px;
   ```

```
    padding: 0 30px 0 30px;
  }
```

2.7 Part 6: Set and test permissions

People > Permissions

If you are not using the Test Kitchen Install Profile or if you are new to the idea of users, roles, permissions or masquerade, see http://training.opensourcery.com/basics

2.7.1 6.1: Set permissions

Set permissions as follows:

Author	Editor	Admin
[] Gallery: Create new content	[] Gallery: Create new content	[] Gallery: Create new content
[] Gallery: Edit own content	[ ] Gallery: Edit own content	[] Gallery: Edit own content
[ ] Gallery: Edit any content	[] Gallery: Edit any content	[] Gallery: Edit any content
[] Gallery: Delete own content	[ ] Gallery: Delete own content	[] Gallery: Delete own content
[ ] Gallery: Delete any content	[] Gallery: Delete any content	[] Gallery: Delete any content

2.7.2 6.2: Test Author privileges

Masquerade as Test Author and ensure you CAN:

1. Create a gallery
2. Edit that gallery
3. Delete that gallery

Ensure you CANNOT:

1. Edit galleries you didn’t create
2. Delete galleries you didn’t create

When you’re done, remember to Switch back

2.7.3 6.3: Test Editor privileges

Masquerade as Test Editor and ensure you CAN:

1. Create a gallery
2. Edit that gallery
3. Delete that gallery
4. Edit a gallery you didn’t create
5. Delete a gallery you didn’t create

CHAPTER

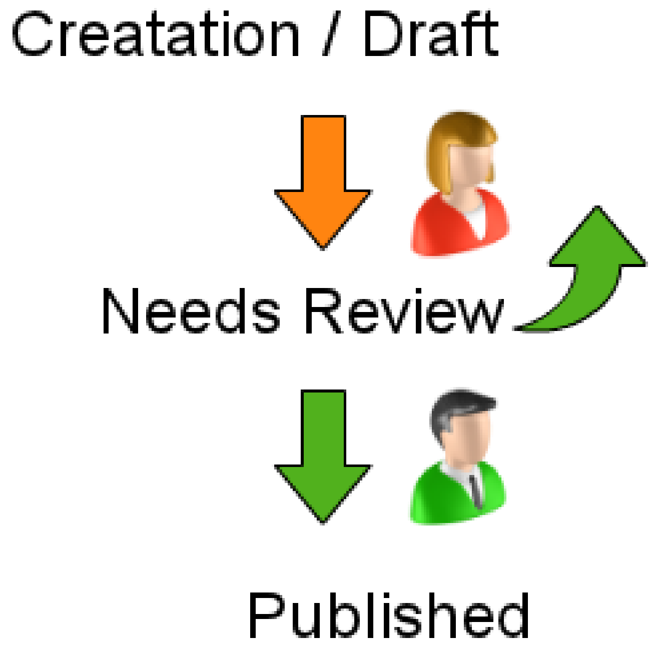
THREE

PERFECTING PERMISSIONS: EDITORIAL WORKFLOW

Does your organization require content review before the content gets published? We'll build a common editorial workflow that allows an author to create and edit his or her own work and submit it for editorial review. Author and editor can collaborate until it's ready to be passed along for final approval and publication.

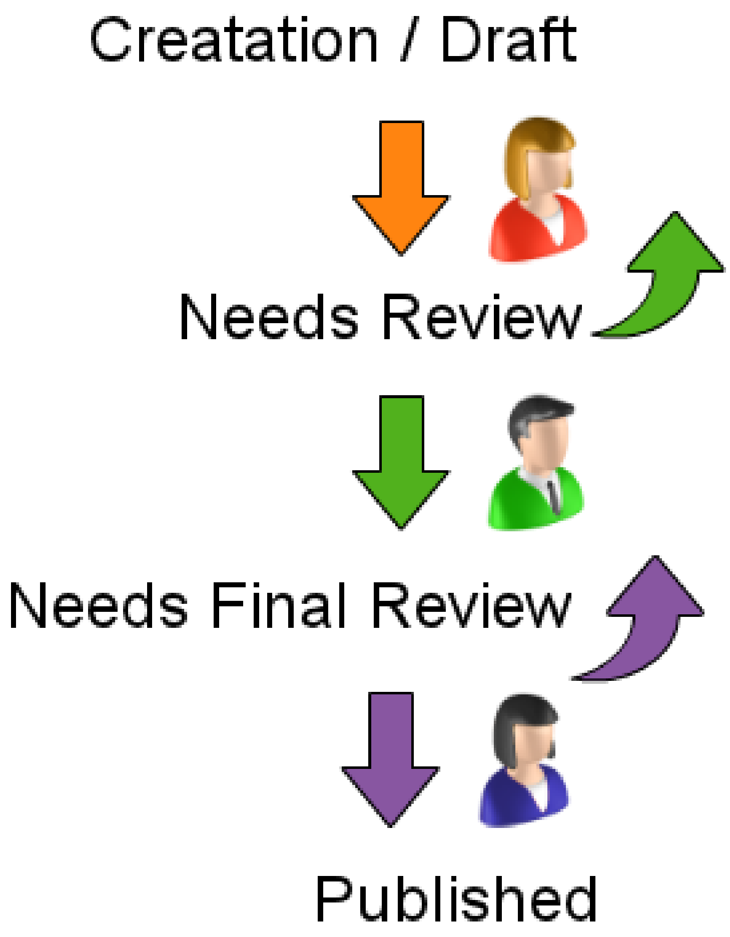
3.1 Base Workflow

With just two roles, we make use of Workbench access out of the box.



3.2 Expanded Workflow

Adding a publisher lets us look at how to define new states, transitions, and views.



3.3 Ingredients

3.3.1 Staple

1. http://drupal.org/project/ctools - Enable Chaos tools
2. http://drupal.org/project/views - Views: Enable Views and Views UI
3. http://drupal.org/project/workbench - Enable Workbench
4. http://drupal.org/project/workbench_moderation - Enable Workbench Moderation
5. http://drupal.org/project/diff - Enable Diff

Drush users:

```
drush dl ctools views workbench workbench_moderation diff
drush en ctools views workbench workbench_moderation -y
```

3.4 Part 1: Users and permissions

People > Permissions

We're working with three pre-created users, each of whom is assigned to one of the three pre-created roles that will be used in the workflow. Creating users and roles is routine, but permissions are one of the trickiest things to get right when creating a workflow, so we're working through that together.

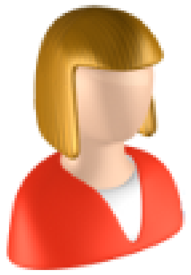


Figure 3.1: Cathy Contributor
Cathy has been assigned the Contributor role so that she will be able to create and edit her own content.

Figure 3.2: Eli Editor
Eli has been assigned the Editor role so that in addition to being able to do anything a contributor can do, he will be able to edit the work of others and advance content for publication.

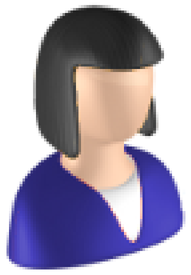


Figure 3.3: Peggy Publisher
Peggy has been assigned the Publisher role. This is the only role that will be able to make content live on the site. Publishers can do everything that editors and contributors can do.

3.4.1 1.1: Node permissions

Before setting up the second layer of permissions provided by Workbench Moderation, we'll set the basic node permissions. These settings are based on the premise that no site content is ever deleted. During development, you might choose to allow the deletion of content.

Permission	Contributor	Editor	Publisher
View own unpublished content	[]	[]	[]
View content revisions	[]	[]	[]
Revert content revisions	[ ]	[]	[]
Delete content revisions	[ ]	[ ]	[ ]
Article: Create new content	[]	[]	[]
Article: Edit own content	[]	[]	[]
Article: Edit any content	[ ]	[]	[]
Article: Delete own content	[ ]	[ ]	[ ]
Article: Delete any content	[ ]	[ ]	[ ]
Basic page: Create new content	[]	[]	[]
Basic page: Edit own content	[]	[]	[]
Basic page: Edit any content	[ ]	[]	[]
Basic page: Delete own content	[]	[ ]	[]
Basic page: Delete any content	[ ]	[ ]	[]

3.5 Part 2: Configure workbench moderation

3.5.1 2.1: Review configuration

Configuration > Workbench: Workbench Moderation

We will start by using the default states provided by Workbench Moderation and later add new ones. The labeling of states is tricky business since the contributor chooses a state before it's been applied and everyone else sees the state afterward. In general, it's best to choose the label that makes the most sense *after* it has been applied, e.g., Published instead of Publish.

3.5.2 2.2: Enable the workflow for content types

Structure > Content types > Article: edit

Although we don't cover this today, it is possible to have different workflows for different content types. This gets complex very quickly! See http://drupal.org/node/1206854#comment-5241230 for how to enable this.

1. Publishing options
2. [ ] Published (Uncheck this)
3. [] Promoted to front page (Leave checked)
4. [ ] Sticky at top of lists (Leave unchecked)
5. [] Create a new revision (Check this!)
6. [] Enable moderation of revisions (Can only be checked once the item above is checked. Check this).
7. Default moderation state: Draft
8. Save content type
9. Repeat for the Basic page content type (although don't promote pages to the front page)

3.5.3 2.3: Set the Workbench Moderation permissions

People > Permissions

At the bottom of the modules page:

Workbench

Permission	Contributor	Editor	Publisher
Administer Workbench settings	[ ]	[ ]	[ ]
Access My Workbench	[]	[]	[]

Workbench Moderation

Permission	Contributor	Editor	Publisher
View all unpublished content	[ ]	[]	[]
Administer Workbench Moderation	[ ]	[ ]	[ ]
Bypass moderation restrictions	[ ]	[ ]	[ ]
View moderation history	[]	[]	[]
View the moderation messages on a node	[]	[]	[]
Use “My Drafts” workbench tab	[]	[]	[]
Use “Needs Review” workbench tab	[ ]	[]	[]
Moderate all content from Draft to Needs Review*	[]	[]	[]
Moderate all content from Needs Review to Draft	[ ]	[]	[]
Moderate all content from Needs Review to Published	[ ]	[]	[]

- It’s oddly worded, but you must grant this permission to the Contributor. It really means “Moderate all content types to Needs review.” It is limited in scope by your particular permissions for each content type. The Contributor role is only allowed to edit its own Articles and Basic Pages, so it Cathy Contributor won’t be able to Moderate other peoples’ work.

3.5.4 2.4 Check your permissions

*Configuration > Workbench: Workbench Moderation > Check permissions (tab)

Workbench has a great feature to help ensure you’ve set up your transitions properly.

3.6 Part 3: Watch Contributor > Editor Moderation in Action

Home

We’re going to set up in two separate browsers, once for Cathy Contributor and one for Eli Editor so that we can watch as content progresses through the workflow.

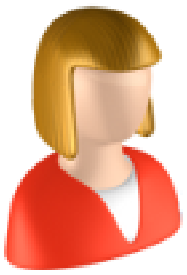
3.6.1 3.1: Create an article as Cathy Contributor

3.6.2 3.2: View Eli Editor’s Worbench

My Workbench > Needs Review (tab)

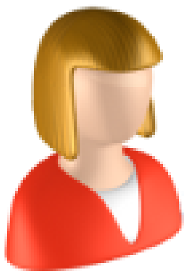
3.6.3 3.3: Move Cathy’s work to Needs Review

Workbench > My Drafts*



1. In the first browser, log in as admin and masquerade as Cathy Contributor.
2. Create a new article.
3. Note the message that the draft will be placed in moderation.

1. In the second browser, log in as admin and masquerade as Eli Editor.
2. View My Workbench > Needs Review (tab)
3. You shouldn't see anything yet.



Cathy Contributor can move her work from Draft to Needs Review in three places:

- On the View draft tab of the node, just as soon as she's saved it
- On the Moderate tab, where the entire revision history is available
- From the My Drafts page

Her work will be visible to no one but an administrator until she moves it to the Needs Review state.

1. Move the article created in 2.4.1 into the Needs Review state.
2. Note that it disappears from Drafts and reappears on the My content page in My Edits. If Cathy bookmarks the page, she can still edit this while it's in the Needs review state, good for fixing that last typo.

3.6.4 3.4: Publish Cathy's article

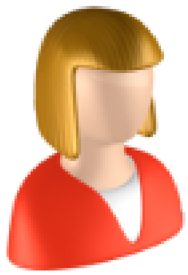
My Workbench > Needs Review (tab)

1. Reload Eli Editor's Needs Review tab.
2. Locate Cathy's new article.
3. Publish it.

What if you'd like to return the article to Cathy because it needs work? You can send it back by setting it to Draft, but out of the box, it's difficult to add a message to the page. We'll fix that as best we can in Part 5: Adjusting the Views. There is a feature request to make it easier to add comments at http://drupal.org/node/1257650.

3.6.5 3.5: As Cathy, review the published work

My Workbench



1. Take a look at Cathy's main workbench page.
2. Edit the Contributor FAQ
3. Note that the officially-approved content stays published while this draft goes into moderation.

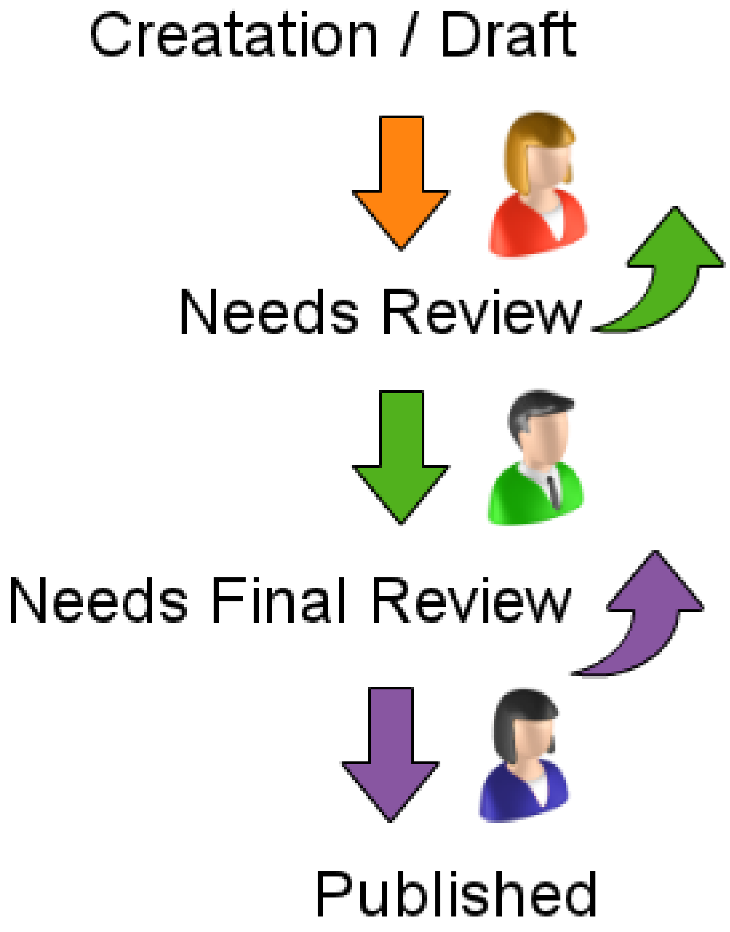
3.6.6 3.6: Experiment

1. Try routing the article between Cathy and Eli in ways your organization might use workflow.
2. Try creating content as Eli.
3. Switch back to the admin account when you're done.

3.7 Part 4: Expanding the Workflow

3.7.1 4.1: Add a new state

Configuration > Workbench: Workbench Moderation



1. On the main screen, add a state:

   Name: Needs Final Review

   Description: Ready for Publication

2. Arrange them in order by using the grabber.

   - Draft
   - Needs Review
   - Needs Final Review
   - Published

3. Save

When you save, you'll see the following warning: Depending on the changes you have made it may be necessary to reconfigure Views that leverage Workbench Moderation such as workbench_moderation. Part 5 will address the customizations necessary to take advantage of our new state.

3.7.2 4.2: Enable the transitions

Configuration > Workbench: Workbench Moderation: Transitions (tab)

Next we define the new transitions. Move to the Transitions tab and:

1. Add Needs Review > Needs Final Review This will let us give editors permission to advance content to publishers.
2. Add Needs Final Review > Needs Review This will allow us to give publishers the ability to send the content back to the editors.
3. Add Needs Final Review > Published This will allow us to give the publisher permission to publish.
4. Delete Needs Review > Published This will removes an editor's ability to publish without final review.

3.7.3 4.3: Assign transition permissions

People > Permissions

Set the new permissions.

Workbench Moderation

Permission	Contributor	Editor	Publisher
Moderate all content from Needs Review to Needs Final Review	[ ]	[]	[]
Moderate all content from Needs Final Review to Needs Review	[ ]	[ ]	[]
Moderate all content from Needs Final Review to Published	[ ]	[ ]	[]

3.8 Part 5: Adjust the views

Structure > Views > workbench_moderation: edit

Since we added our own state, we'll need to create another tab that is visible to the Publisher.

1. Select the display: Needs Review Page
2. Clone it
3. Display Name: Ready to Publish Page
4. Title: Ready to Publish
5. Filter Criteria > Workbench Moderation State: Needs Final Review
6. Page Settings > Path: admin/workbench/ready-to-publish
7. Page Settings > Menu > Title: Ready to Publish
8. Page Settings > Access: Permissions (change to role and select Publisher)

3.9 Part 6: Watch Editor > Publisher Moderation in Action

3.9.1 6.1: Create an article as Eli Editor

Home

1. In the first browser, log in as admin and masquerade as Eli Editor.
2. Create a new article Title: Editorial Policies Body: Just because you *can* move your work without review by another editor may not mean you should.
3. Save
4. Move it directly to Needs Final Review

3.9.2 6.2: View Peggy Publisher's Workbench

My Workbench > Needs Review (tab)

1. In the second browser, log in as adhttp://workflow.everydaydrupal.com/min and masquerade as Peggy Publisher.
2. View My Workbench > Ready for Publication (tab)
3. Publish Eli's Editorial Policies article.

3.9.3 6.3: Move Eli's work to Needs Review

My Workbench > My Drafts

3.9.4 Part 7: Design your own workflow

Extend your understanding by working on one of the following:

If this workflow is exactly what you need, consider:

1. How could the workbench moderation pages do a better job of supporting the process in your organization?
2. What additional features might you want?

If this workflow doesn't meet your needs, design one that does.

CHAPTER

FOUR

KEEPING IT FRESH: SCHEDULING CONTENT

Intro blurb

4.1 Ingredients

4.1.1 Staple

1. http://drupal.org/project/ctools Enable Chaos Tools
2. http://drupal.org/project/views Enable Views and Views UI
3. http://drupal.org/project/date Enable Date and Date Popup
4. http://drupal.org/project/scheduler Enable Scheduler
5. http://drupal.org/project/diff Enable Diff

4.2 Part 1: Create content to work with

Configuration > Development: Generate Content

1. [] Check Article
2. [ ] Uncheck Basic Page
3. Generate 50 articles, going back in time for a year.
4. Generate

4.3 Part 2: The latest content approach

Structure > Views > +Add a new view

Pros: Content is still visible if a user reaches it via a link from elsewhere or a search engine.

Cons: The article is still visible. :)

4.4 Part 3: Schedule articles