Hey Ben,

Here are the three things Ted and I worked on

Groups Calendar Context

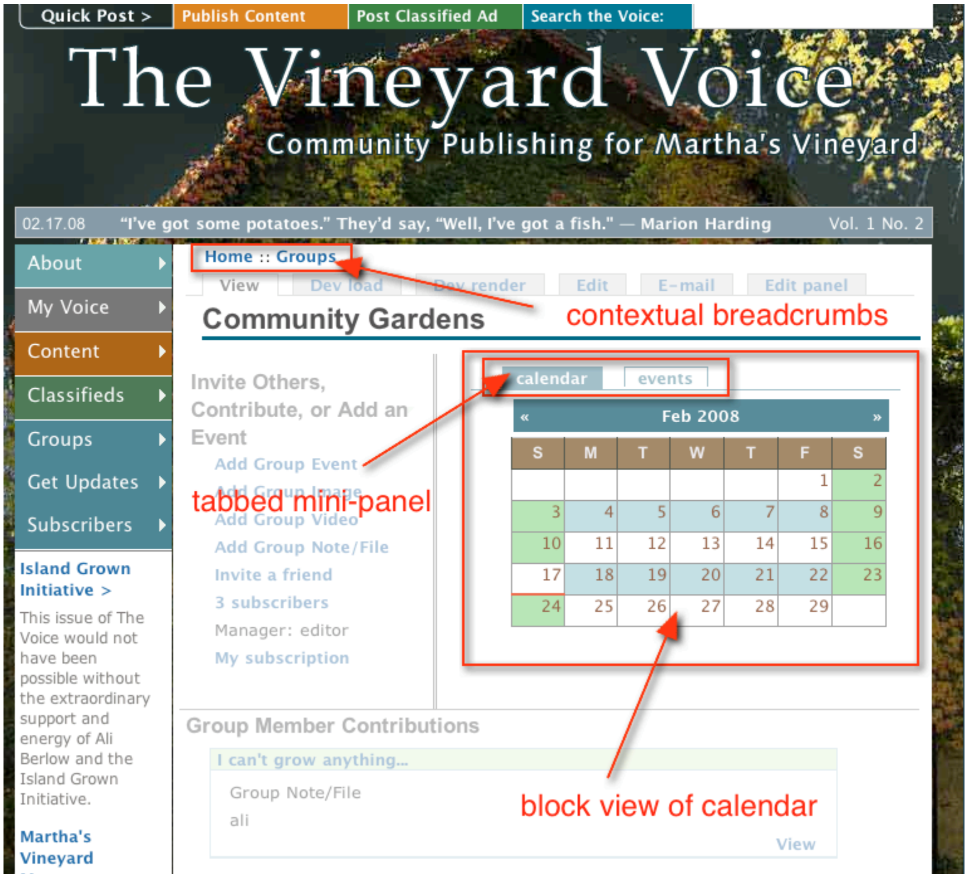
→ Organic Groups/CCK-Views/Panels 2 Calendar using calendar and date module (not event and og_calendar)
→ Tabbed mini-panel implementation of block — passing arguments through to the view.
→ Organic Groups Breadcrumb Context management

I will try to walk you through in detail.

I provide all the steps and notes, even though you are familiar with many of them. (I will share this document with others less familiar with the process.)

Contextual Group Calendar Using Calendar module, Views and Panels 2

From a UI perspective, this is what we're after:



02.17.08 “I've got some potatoes.” They'd say, “Well, I've got a fish.” — Marion Harding Vol. 1 No. 2

Community Gardens

Invite Others, Contribute, or Add an Event

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	

Island Grown Initiative >

This issue of The Voice would not have been possible without the extraordinary support and energy of Ali Berlow and the Island Grown Initiative.

Martha's Vineyard

Group Member Contributions

I can't grow anything...

Group Note/File

ali

contextual breadcrumbs

tabbed mini-panel

block view of calendar

Work/Project Summary

I wanted full control over the evolution and design of calendaring in my site, but as importantly, I wanted to maintain group home-page context as users schedule and review events.

As you may know, using Event module takes control of calendaring AND disrupts the context — it moves the user from the groups area of the site with the calendar displayed into the event module breadcrumb. This is confusing for users, and I wanted to eliminate that confusion.

In order to do this I had to eliminate the Event and og_calendar modules, but I also had to have a way for the group node id (not the group id) to be read and maintained by Drupal. Once I understood what argument handling code to pass through (not just arguments), all as pretty straightforward.

The step that complicated this is that I am using mini panels to place a calendar block on the page. This is not hard, but the panel must recognize the argument handling code in order to "know" where it is — in order to maintain group context. By simply passing arguments through the mini panel to the view the whole set of 1) how to display events and 2) how to maintain context was all synched up, and I could move on to clarifying the breadcrumb.

The reason breadcrumbing was so important is that this is the only way to present navigational options with "Groups/"the-group-you-are-in"/calendar context at the top of each content area — following the trail from the group home page block-view to calendar-view to event listing to node. To manage the breadcrumbs a template page snippet had to be developed.

All this likely convoluted to the average user. It was to me prior to diving into it. I hope my breakdown of the steps helps others avoid the convolution and get right to clean, contextual group calendaring.

Many of the steps implemented here can be applied to other scenarios, and I hope too that the presentation of the steps helps people understand how Views, CCK and Panels 2 "talk" to each other through field types, argument types and argument handling code. It's really not that hard.

I worked with Ted Sirbinsky of Lullabot and MothersClick fame on solving this problem.

Views-based Calendar Configuration Procedure

The challenge of building a Views-based, tabbed mini-panel calendar has several parts:
Check here for the concise, Drupal.org generic walkthrough of this process.

I turning off the event and og_calendar modules, if they are installed,

II installing calendar module and making sure date module are installed and enabled,

III creating a simple event content type with a date-stamp

IV configuring a group calendar view

V installing panels 2 and tabs panel-style modules:
note: panels 2 is in beta and my current configuration does not work with panels 2 beta 2. It works well with panels 2 alpha 13. I don't know why, and I don't have time to figure it out right now.

Make sure you run update.php when you install Panels 2. There are some database additions that are critical for proper Panels 2 functionality.

VI creating a **block view calendar** with the right arguments and argument handling code,

VII creating the calendar mini panel and passing argument handling code through to the view in the **mini-panel**,

VIII creating the "group_tab" mini panel and styling the mini panel as "**tabbed style**" and placing the tabbed mini panel into Group home panel page

IX clarifying breadcrumbs through **template page code**.

I & II — Easy part. I just removed event and og_calendar from my configuration. I did not have event nodes already on my site, so I didn't have to worry about management of those dates. Download the calendar and date modules, install and enable them.

III — **Creating an Event Content Type**

Karen S. wrote a very good, step by step for using Calendar and Date with views can be found here. There is also a step by step on creating the event content type. The one difference between the content type Karen S. creates and mine is that I use a Datestamp field instead of a Date field. (The view Karen S. provides had to be modified for my application.

VI — **Creating a Calendar Block View**

Installing and activating Calendar module makes available a calendar view. Once the view is **added** the basic template for a calendar is available -- it's just a matter of configuring it well.

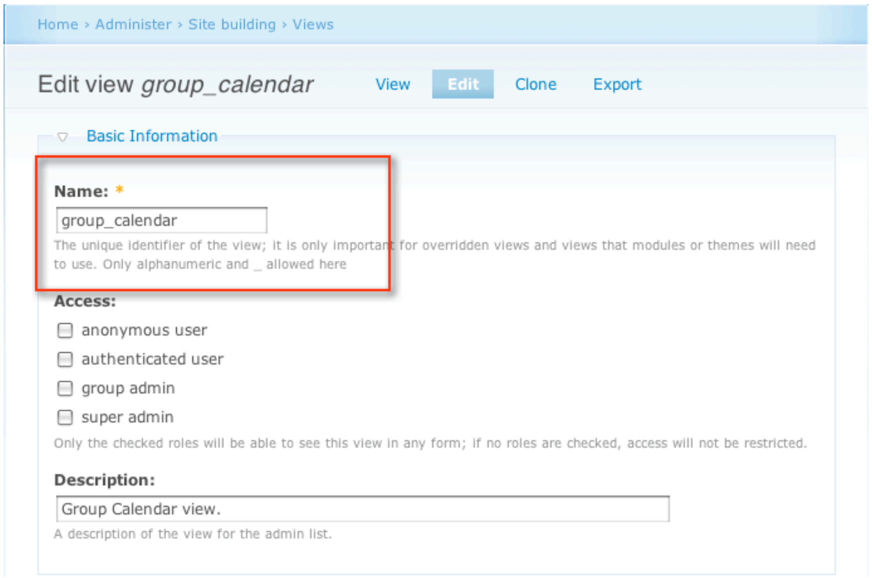
This "simple" thing is where so many people spend too much time. The steps here are important to follow, and it's easy to miss a couple of them:

1) **Editing the Calendar View:**

Once you have added the Calendar View, choose "edit":

→ Give the View a good **name**. Mine's "group_calendar". You may add calendars in the future; a clear name will make them all easy to keep track of all calendars. This

group name is machine readable and is used by the your database.
Note: It is NOT the View "Title". The View Title will be discussed below.



Edit view *group_calendar*

Name: *

group_calendar

The unique identifier of the view; it is only important for overridden views and views that modules or themes will need to use. Only alphanumeric and _ allowed here

Access:

- □ anonymous user
- □ authenticated user
- □ group admin
- □ super admin

Only the checked roles will be able to see this view in any form; if no roles are checked, access will not be restricted.

Description:

Group Calendar view.

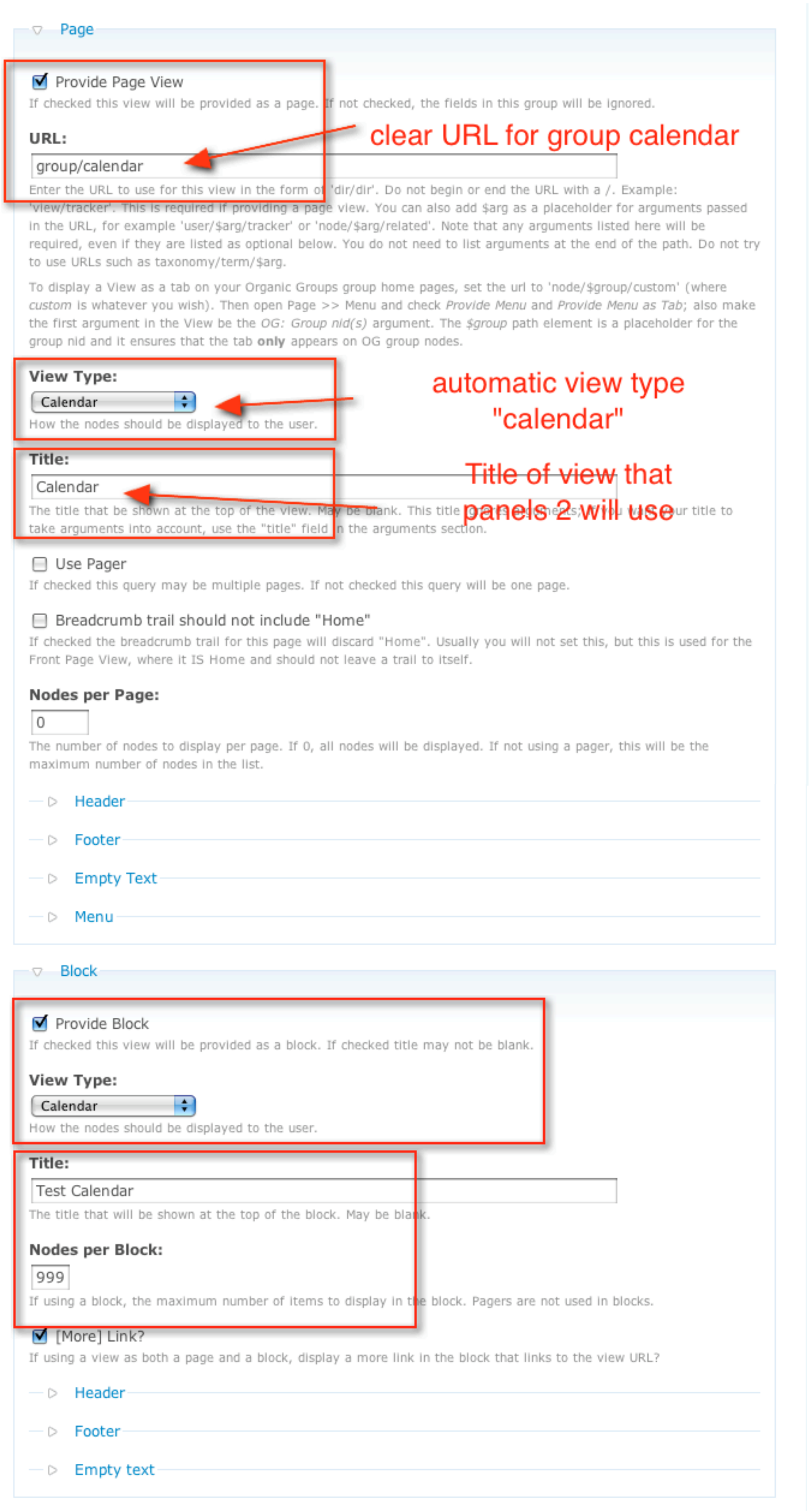
A description of the view for the admin list.

2) **Page View & Block View:**

You will need both a Page view and a Block view for the group_calendar.

The reason is clear when you think about it. In terms of workflow or user experience, I wanted to have the block calendar available on the individual group home page that would link users to events in a compact space and would give them an overview of their specific group's events. They can click on an event in that calendar block and go to the page view calendar. [Creating a Groups home page is another whole step -- some of the process can be found here.]

→ Make sure "Provide Page View" is selected.
It is important that the URL entered is clear and simple.
→ Because the Calendar View was added, the "View Type" will be automatically selected as **Calendar**.
→ The "Title" of the calendar view in my site is "Calendar."
Note: Be clear here. In order for you to place this block into a panel in the next section, Panels 2 will need **the title of this view**. The "View Title" helps **YOU** keep track of and use the view. "Name" helps the **database** keep track of and use the view.
→ Make sure "Provide Block" is checked.
→ Calendar View Type will automatically be selected.
If for some reason you have changed it, make sure it is "Calendar."
I have titled the block "Test Calendar" — title it what will make the most sense.
→ Display 999 nodes per block -- These are events, and chances are you may need a lot of linked references to nodes/events created by users. It helps to just max this field out.



▽ Page

☑ Provide Page View

If checked this view will be provided as a page. If not checked, the fields in this group will be ignored.

URL:

group/calendar

clear URL for group calendar

Enter the URL to use for this view in the form of 'dir/dir'. Do not begin or end the URL with a /. Example: 'view/tracker'. This is required if providing a page view. You can also add $arg as a placeholder for arguments passed in the URL, for example 'user/$arg/tracker' or 'node/$arg/related'. Note that any arguments listed here will be required, even if they are listed as optional below. You do not need to list arguments at the end of the path. Do not try to use URLs such as taxonomy/term/$arg.

To display a View as a tab on your Organic Groups group home pages, set the url to 'node/$group/custom' (where *custom* is whatever you wish). Then open Page >> Menu and check *Provide Menu* and *Provide Menu as Tab*; also make the first argument in the View be the *OG: Group nid(s)* argument. The *$group* path element is a placeholder for the group nid and it ensures that the tab **only** appears on OG group nodes.

View Type:

Calendar

How the nodes should be displayed to the user.

automatic view type "calendar"

Title:

Calendar

Title of view that panels 2 will use

The title that be shown at the top of the view. May be blank. This title ignores arguments; if you want your title to take arguments into account, use the "title" field in the arguments section.

☐ Use Pager

If checked this query may be multiple pages. If not checked this query will be one page.

☐ Breadcrumb trail should not include "Home"

If checked the breadcrumb trail for this page will discard "Home". Usually you will not set this, but this is used for the Front Page View, where it IS Home and should not leave a trail to itself.

Nodes per Page:

0

The number of nodes to display per page. If 0, all nodes will be displayed. If not using a pager, this will be the maximum number of nodes in the list.

▽ Block

☑ Provide Block

If checked this view will be provided as a block. If checked title may not be blank.

View Type:

Calendar

How the nodes should be displayed to the user.

Title:

Test Calendar

The title that will be shown at the top of the block. May be blank.

Nodes per Block:

999

If using a block, the maximum number of items to display in the block. Pagers are not used in blocks.

☑ [More] Link?

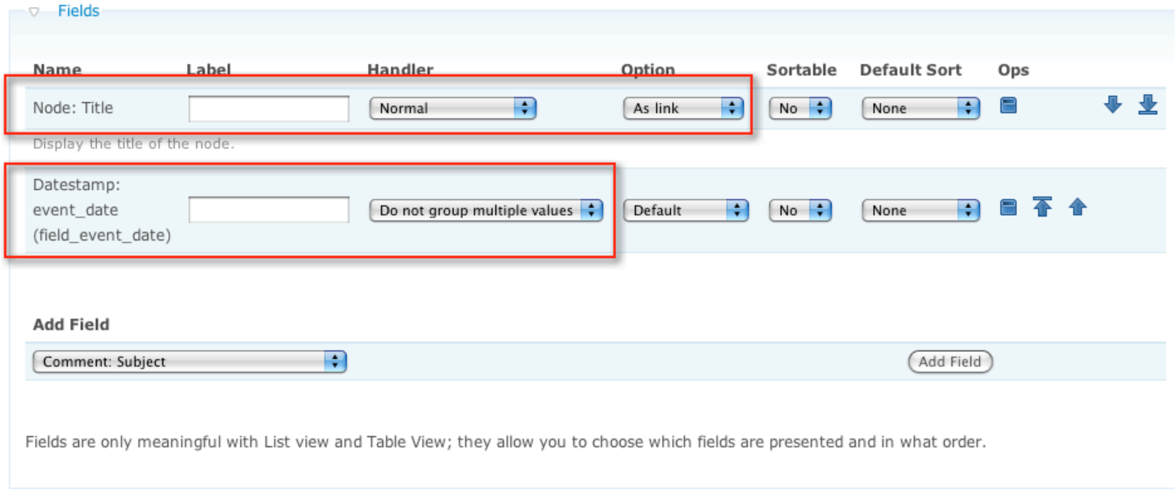
If using a view as both a page and a block, display a more link in the block that links to the view URL?

3) **Fields:**

The fields for this Calendar view are simple:

→ Create a Node **Title** field as a link

→ Create a **Datestamp** (name of content type) field. The name of this datestamp must be the same as the CCK event_date content type. Don't group multiple values.



▽ Fields

Name	Label	Handler	Option	Sortable	Default Sort	Ops
Node: Title Display the title of the node.		Normal	As link	No	None	
Datestamp: event_date (field_event_date)		Do not group multiple values	Default	No	None	

Add Field

Comment: Subject

Fields are only meaningful with List view and Table View; they allow you to choose which fields are presented and in what order.

4) **Arguments:**

Once the fields are created you must create the proper arguments for each of ways your events are presented. This is the heart of OG context control.

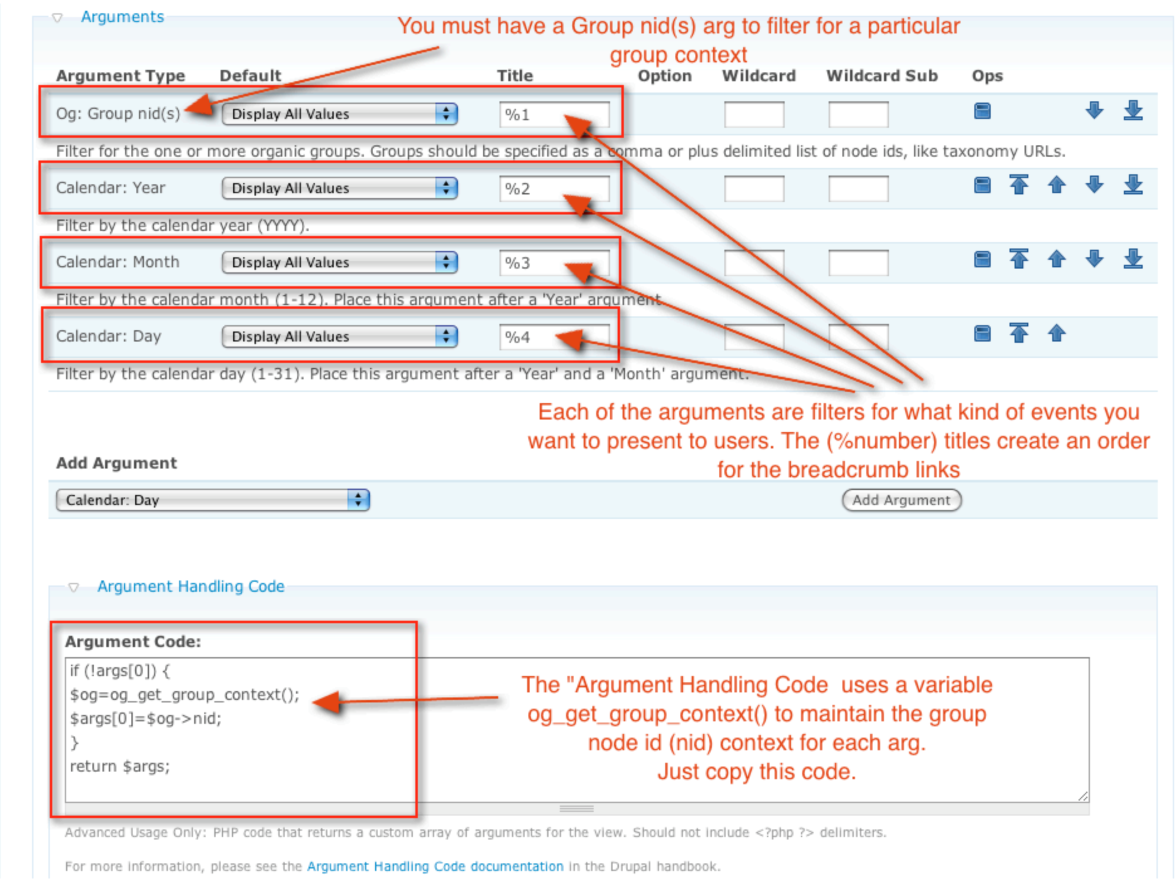
→ Calendar module creates a view template and Calendar:Year, Month, Day should be already present in your view.

→ You will have to add the **Og: Group nid(s)** argument type in order to use the argument handling code.

→ These argument Titles will present the user with clear breadcrumbs which clarify where the user is at any time,

→ The **Og: Group nid(s)** arg type will establish a Node Id context which ties each of the args together with a og_get_group_context() argument code.

→ It is important that these arguments be set up just this way. This is the heart of OG context control.



Arguments

Argument Type	Default	Title	Option	Wildcard	Wildcard Sub	Ops
Og: Group nid(s)	Display All Values	%1				
Filter for the one or more organic groups. Groups should be specified as a comma or plus delimited list of node ids, like taxonomy URLs.						
Calendar: Year	Display All Values	%2				
Filter by the calendar year (YYYY).						
Calendar: Month	Display All Values	%3				
Filter by the calendar month (1-12). Place this argument after a 'Year' argument.						
Calendar: Day	Display All Values	%4				
Filter by the calendar day (1-31). Place this argument after a 'Year' and a 'Month' argument.						

You must have a Group nid(s) arg to filter for a particular group context

Each of the arguments are filters for what kind of events you want to present to users. The (%number) titles create an order for the breadcrumb links

Add Argument

Calendar: Day

Argument Handling Code

Argument Code:

```
if (!args[0]) {
$og=og_get_group_context();
$args[0]=$og->nid;
}
return $args;
```

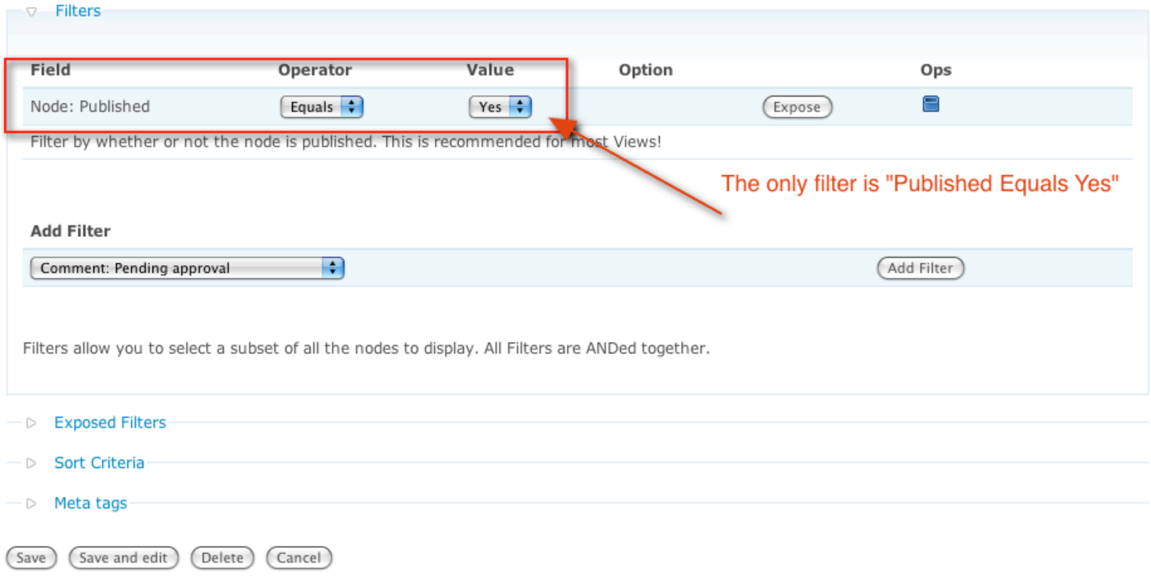
The "Argument Handling Code uses a variable og_get_group_context() to maintain the group node id (nid) context for each arg. Just copy this code.

Advanced Usage Only: PHP code that returns a custom array of arguments for the view. Should not include <?php ?> delimiters.

For more information, please see the Argument Handling Code documentation in the Drupal handbook.

5) **Filters**:

→ make sure to set your filter to Published Equals Yes -- otherwise you won't get any published events.



Filters

Field	Operator	Value	Option	Ops
Node: Published	Equals	Yes	Expose	
Filter by whether or not the node is published. This is recommended for most Views!				

The only filter is "Published Equals Yes"

Add Filter

Comment: Pending approval

Filters allow you to select a subset of all the nodes to display. All Filters are ANDed together.

Congratulations.

You now will have a Views Calendar Page view. You will also have a calendar block to place into your panel. Calendar module comes with a css-styled calendar-mini that will reference all event dates as links that will connect you with a day-view of the node, and to the node itself.

But much more on this next.

VII — Placing Views Block in Panels 2 mini-panel

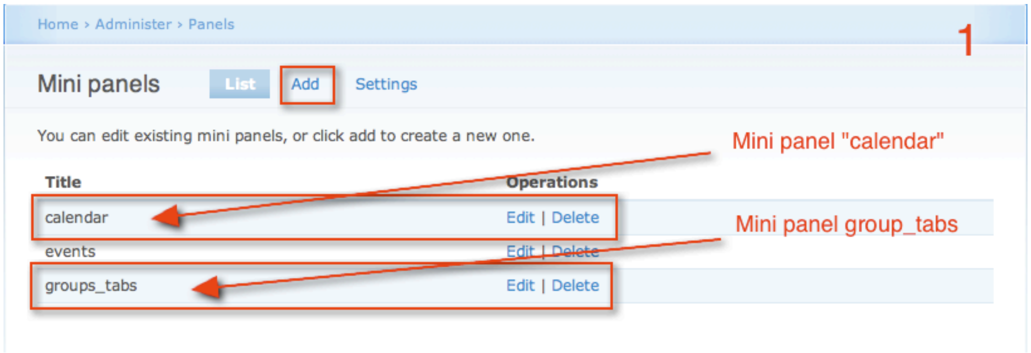
1) **Mini Panel Content panel (not the tabbed panel, but its content.)**

In section V you downloaded and enabled the Panels 2 and Tabs Panel-style modules.

In this segment of section IV, you will create one content mini panel that will contain your Block View Calendar. We start with three mini panels already created.

Note: I have created a "calendar" mini panel, an "events" mini panel and a "groups_tabs" mini panel. The last, "groups_tabs" mini panel will contain the other two mini panels and will display them with tabs. The first two don't need tabs.

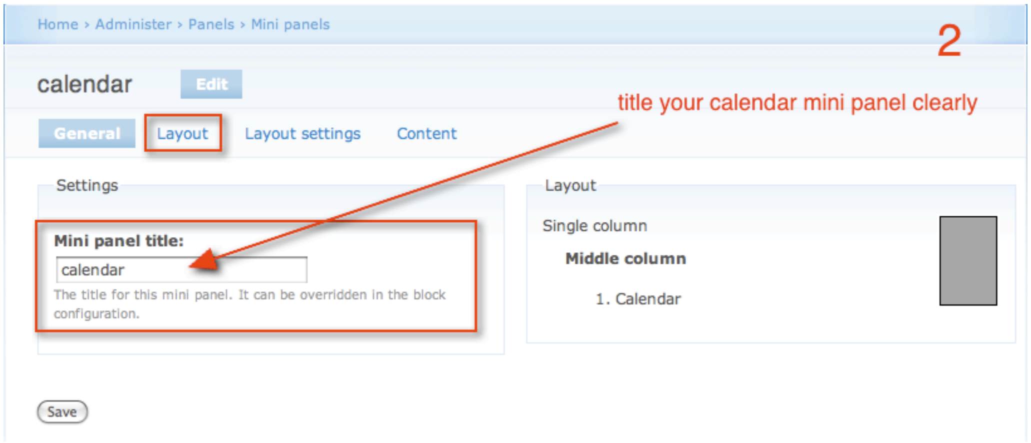
→ First click Add to create a new mini-panel



You can edit existing mini panels, or click add to create a new one.

Title	Operations
calendar	Edit | Delete
events	Edit | Delete
groups_tabs	Edit | Delete

→ Once you have clicked Add, provide a clear title -- "calendar" works well. Then click Layout.


Home › Administer › Panels › Mini panels
2
calendar
Edit
title your calendar mini panel clearly
General
Layout
Layout settings
Content
Settings
Layout
Save

Mini panel title: calendar

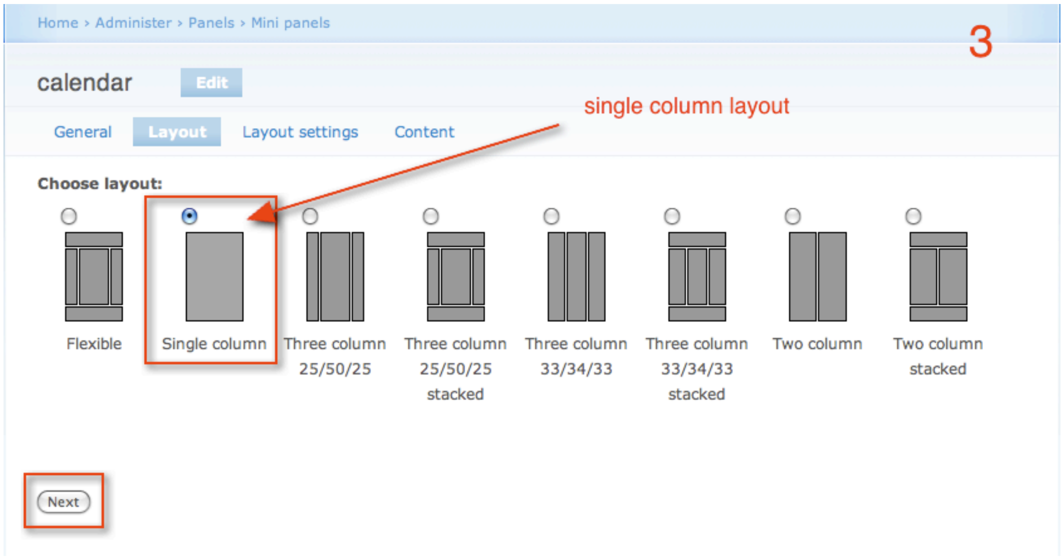
The title for this mini panel. It can be overridden in the block configuration.

Single column

Middle column

1. Calendar

→ For this mini panel we want a Single column layout. Then click Layout Settings, or Next.



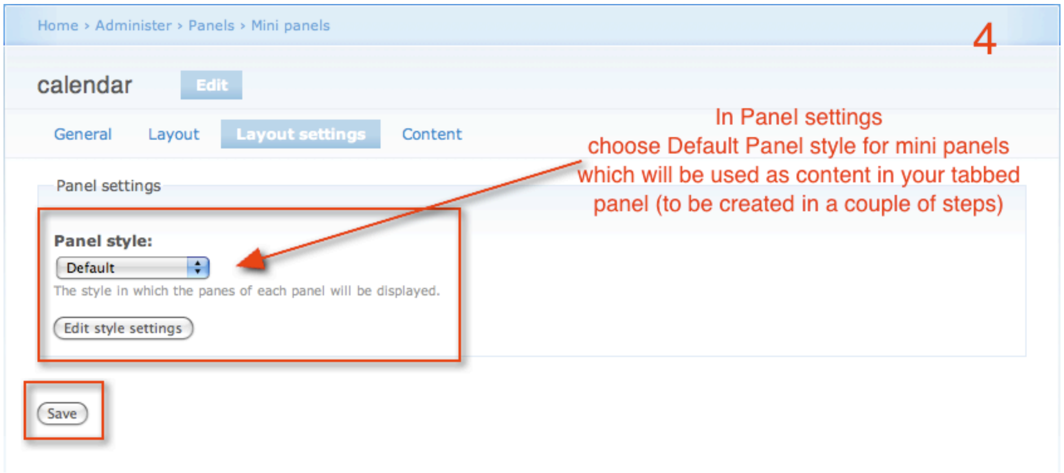
3

single column layout

Choose layout:

○ Flexible
● Single column
○ Three column 25/50/25
○ Three column 25/50/25 stacked
○ Three column 33/34/33
○ Three column 33/34/33 stacked
○ Two column
○ Two column stacked

→ **Layout settings** provides a form to set the Panel style. Because this mini panel will go inside another, tabbed mini panel, set this to the default/no tabs style. Save.



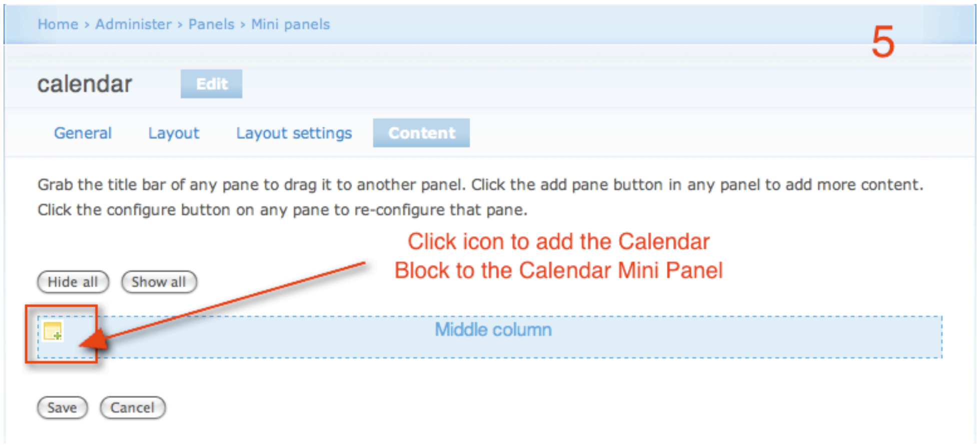
4

In Panel settings
choose Default Panel style for mini panels
which will be used as content in your tabbed
panel (to be created in a couple of steps)

Panel style: Default

The style in which the panes of each panel will be displayed.

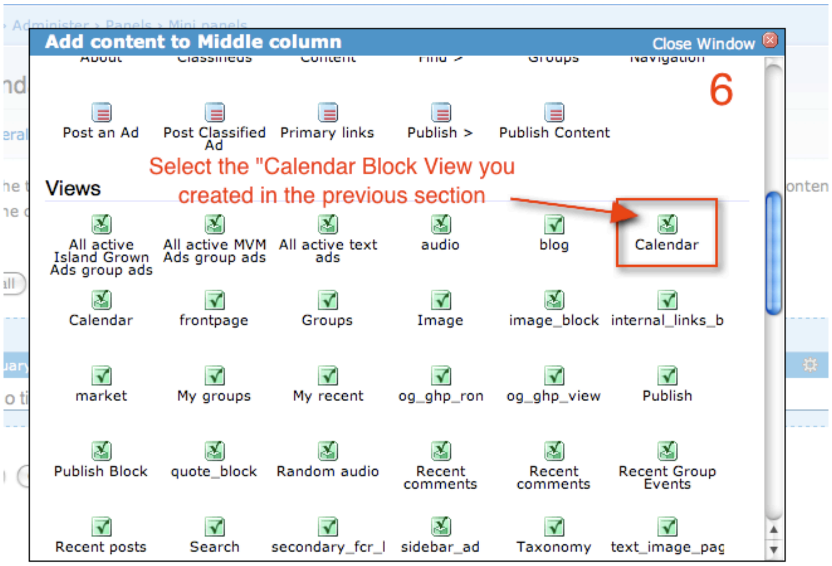
→ **Content:** Once you have set the layout, it's time to add your calendar block view.
Click the icon to add content.



Grab the title bar of any pane to drag it to another panel. Click the add pane button in any panel to add more content. Click the configure button on any pane to re-configure that pane.

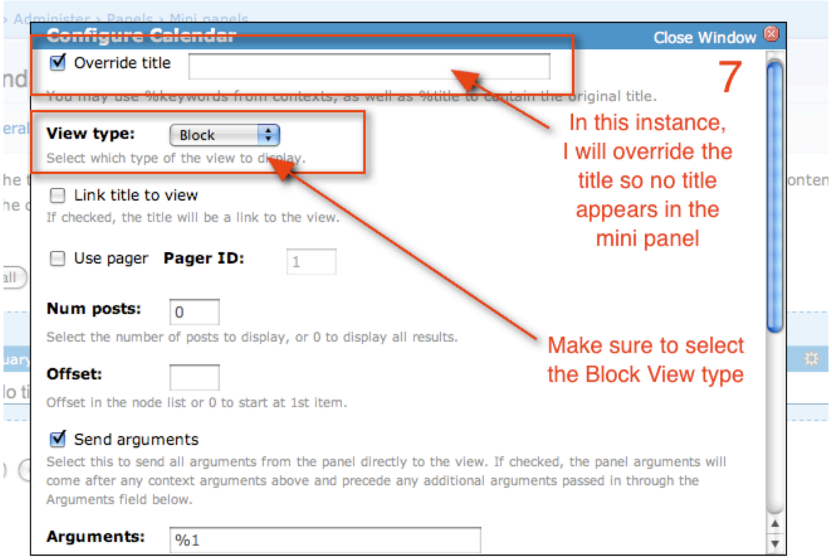
Click icon to add the Calendar Block to the Calendar Mini Panel

→ **Add Content to mini panel**. Here is the reason you were clear when naming your calendar view. In Panels 2 beta you can restrict the clutter of views available to panels, so you don't have to pick through a bunch of blocks and views. Here we have found our Calendar View.



Select the "Calendar Block View you created in the previous section

→ **Configure Calendar** is an important form. It displays as a pop-up. I want to override the title of the block, so no title appears. We want the calendar to appear in the tabbed mini panel as a block, so select View type to be Block.



› Administer › Panels › Mini panels

Configure Calendar Close Window

☑ Override title

You may use %keywords from contexts, as well as %title to contain the original title.

View type: Block

Select which type of the view to display.

□ Link title to view

If checked, the title will be a link to the view.

□ Use pager **Pager ID:** 1

Num posts: 0

Select the number of posts to display, or 0 to display all results.

Offset:

Offset in the node list or 0 to start at 1st item.

☑ Send arguments

Select this to send all arguments from the panel directly to the view. If checked, the panel arguments will come after any context arguments above and precede any additional arguments passed in through the Arguments field below.

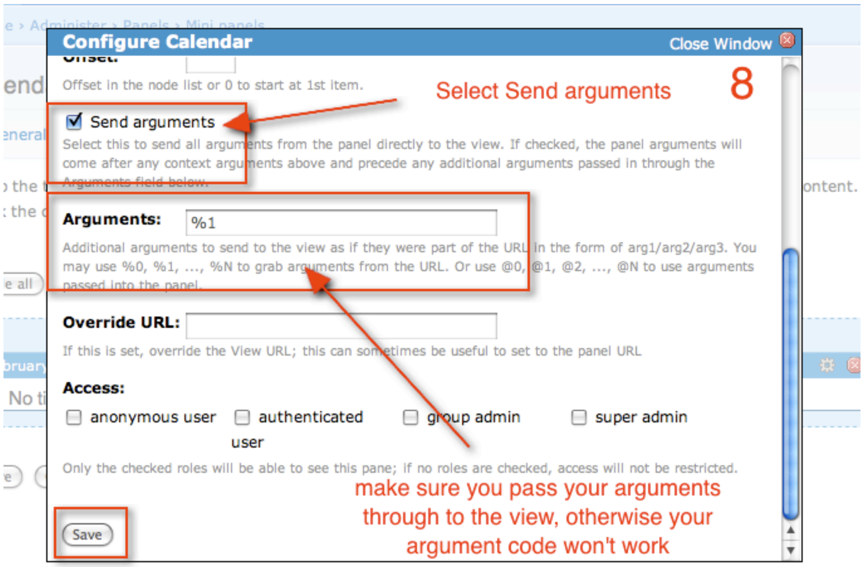
Arguments: %1

7

In this instance, I will override the title so no title appears in the mini panel

Make sure to select the Block View type

→ **Arguments:** **Don't miss this step!** This is where our argument handling code will be passed through to the view from the panel. You must click **Send arguments** and enter **%1** in the Arguments field. Otherwise our code will go unnoticed.



› Administer › Panels › Mini panels

Configure Calendar Close Window

Offset in the node list or 0 to start at 1st item.

☑ Send arguments

Select this to send all arguments from the panel directly to the view. If checked, the panel arguments will come after any context arguments above and precede any additional arguments passed in through the Arguments field below.

Arguments: %1

Additional arguments to send to the view as if they were part of the URL in the form of arg1/arg2/arg3. You may use %0, %1, ..., %N to grab arguments from the URL. Or use @0, @1, @2, ..., @N to use arguments passed into the panel.

Override URL:

If this is set, override the View URL; this can sometimes be useful to set to the panel URL

Access:

□ anonymous user □ authenticated user □ group admin □ super admin

Only the checked roles will be able to see this pane; if no roles are checked, access will not be restricted.

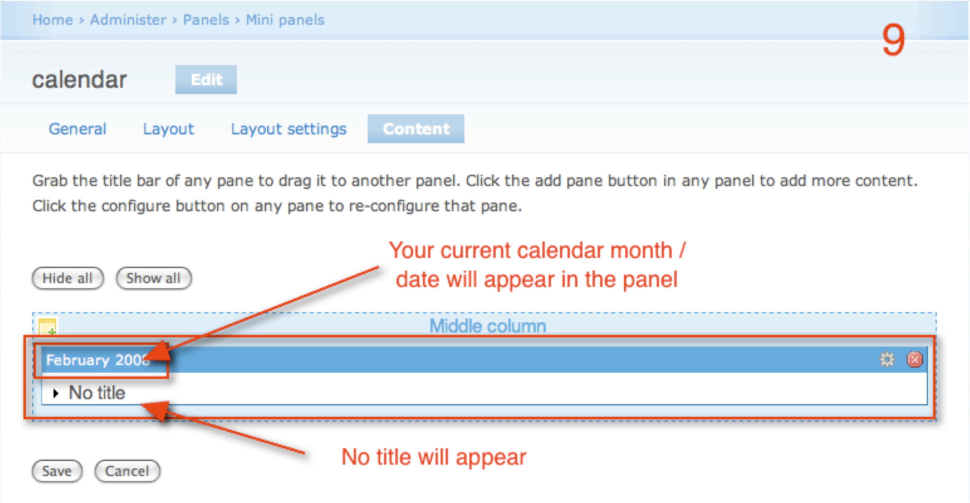
Save

Select Send arguments

8

make sure you pass your arguments through to the view, otherwise your argument code won't work

→ **Content placed in configured mini panel:** If you have not missed a step, you will see your working block inside your single column mini panel.

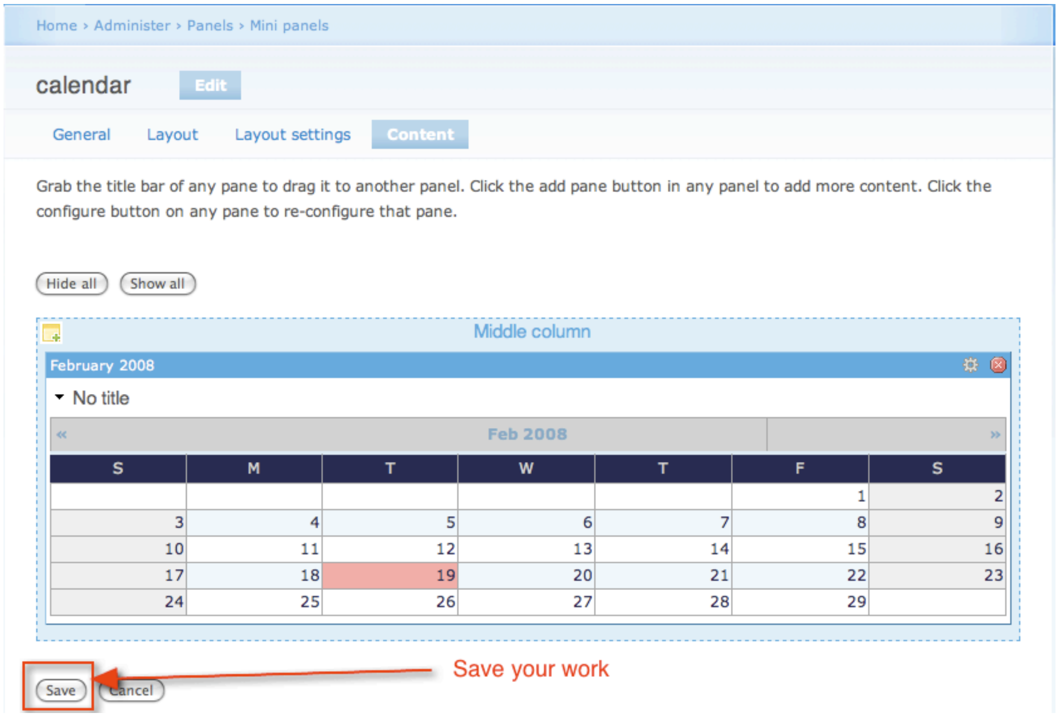


Grab the title bar of any pane to drag it to another panel. Click the add pane button in any panel to add more content. Click the configure button on any pane to re-configure that pane.

Your current calendar month / date will appear in the panel

No title will appear

→ **Save your mini panel configuration.**



Grab the title bar of any pane to drag it to another panel. Click the add pane button in any panel to add more content. Click the configure button on any pane to re-configure that pane.

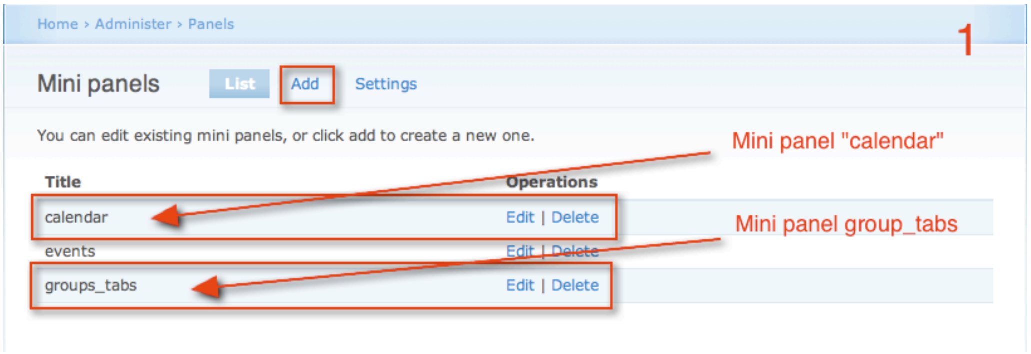
S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	

Save your work

VIII <u>Tabbed Mini Panel implementation</u>

This step by step will be quick and largely visual.

→ Again, I have created three panels in advance. Here, we will focus on the "groups_tabs" mini panel. We will use Tabs panel style module by Wim Leers to style "groups_tabs" and to display the "calendar" mini panel we just created.



Mini panels

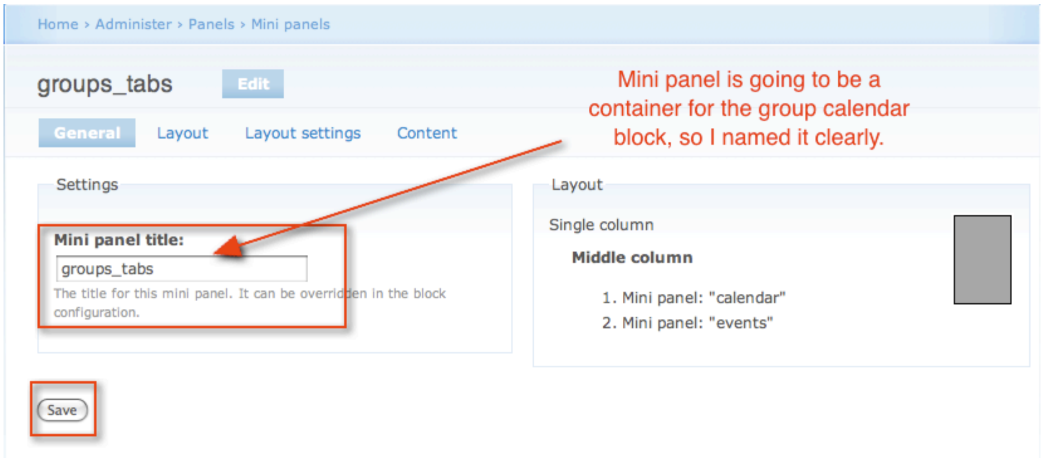
You can edit existing mini panels, or click add to create a new one.

Title	Operations
calendar	Edit | Delete
events	Edit | Delete
groups_tabs	Edit | Delete

Mini panel "calendar"

Mini panel group_tabs

→ Title your tabbed mini panel clearly so it can be selected easily when placing it in your Panels page layout (see below.)



groups_tabs

Settings

Mini panel title: groups_tabs

The title for this mini panel. It can be overridden in the block configuration.

Layout

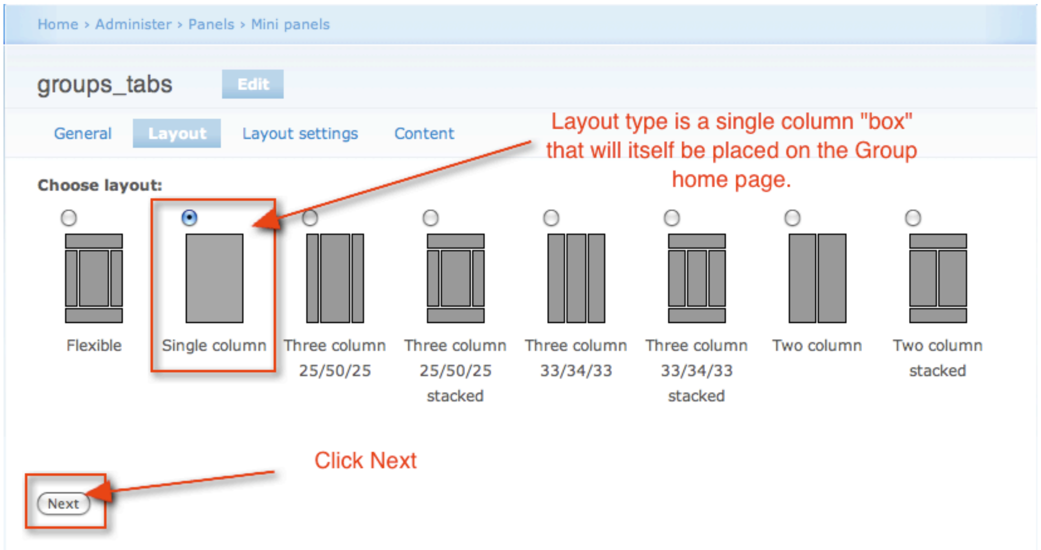
Single column

Middle column

1. Mini panel: "calendar"
2. Mini panel: "events"

Mini panel is going to be a container for the group calendar block, so I named it clearly.

→ Again, select a Single column layout. This is a "box" to contain your other panels, and it will be placed into two-column stacked panel later on.



groups_tabs

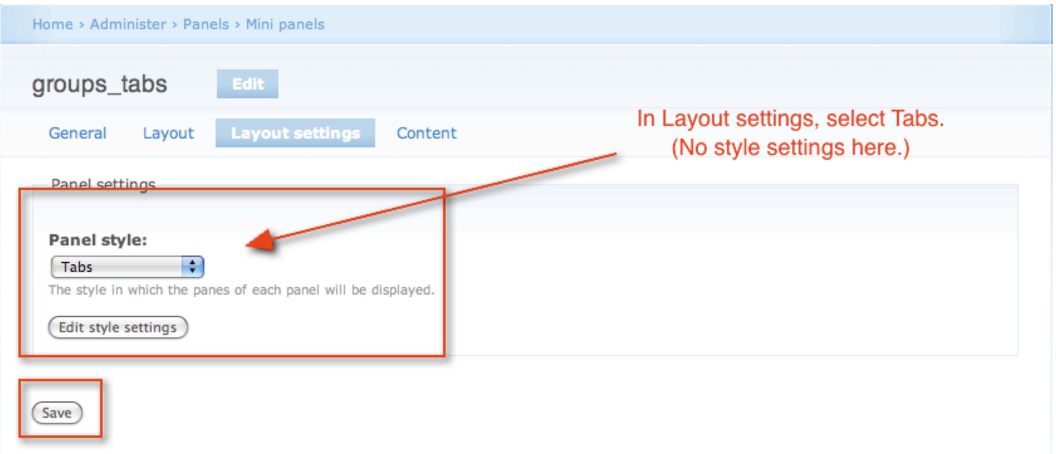
Choose layout:

○ Flexible
● Single column
○ Three column 25/50/25
○ Three column 25/50/25 stacked
○ Three column 33/34/33
○ Three column 33/34/33 stacked
○ Two column
○ Two column stacked

Layout type is a single column "box" that will itself be placed on the Group home page.

Click Next

→ Because your mini panel will have more than one block of content, you can select Tabs in the Layout settings section of mini panel configuration. (You must have the Tabs Style module loaded and enabled to see this selection option. If you don't have it installed you will not see this.

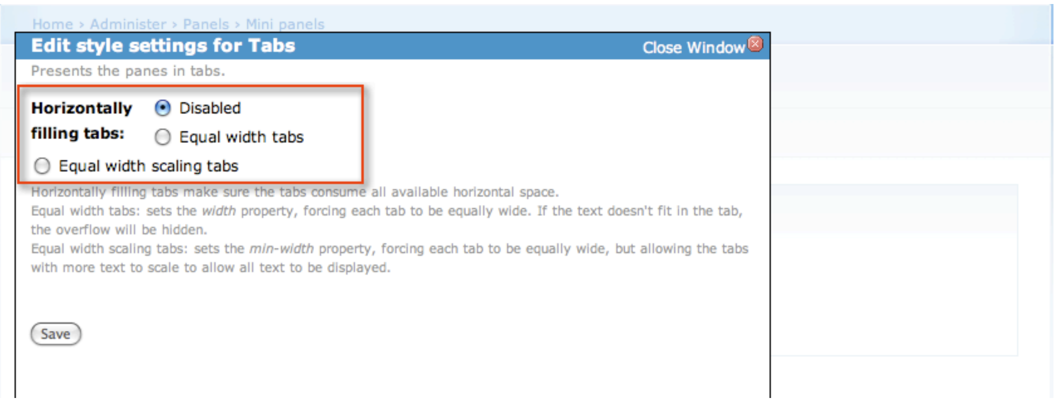


In Layout settings, select Tabs.
(No style settings here.)

Panel style: Tabs

The style in which the panes of each panel will be displayed.

→ I have no style settings set for my mini panel, but here is the edit pop-up, so you know it's options.



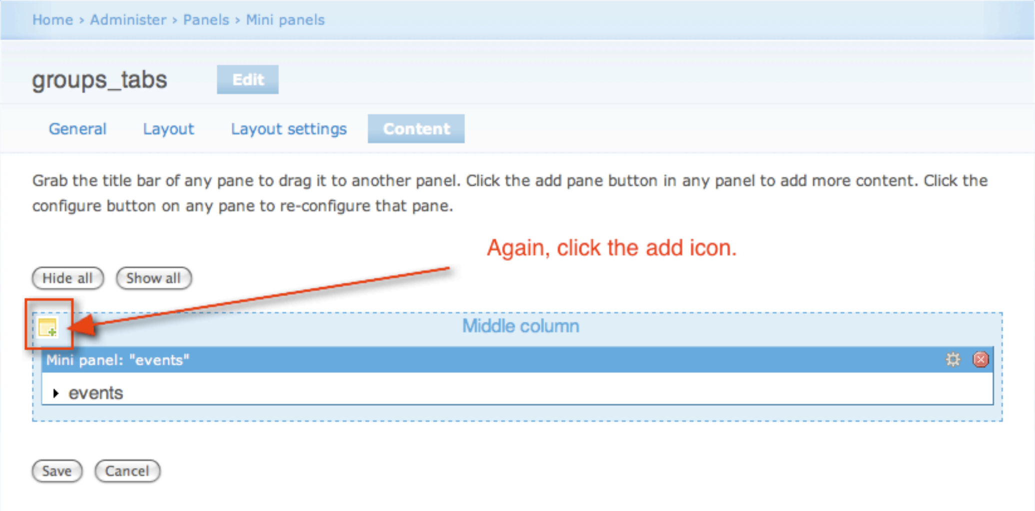
Edit style settings for Tabs

Presents the panes in tabs.

Horizontally filling tabs: ● Disabled ○ Equal width tabs ○ Equal width scaling tabs

Horizontally filling tabs make sure the tabs consume all available horizontal space.
Equal width tabs: sets the *width* property, forcing each tab to be equally wide. If the text doesn't fit in the tab, the overflow will be hidden.
Equal width scaling tabs: sets the *min-width* property, forcing each tab to be equally wide, but allowing the tabs with more text to scale to allow all text to be displayed.

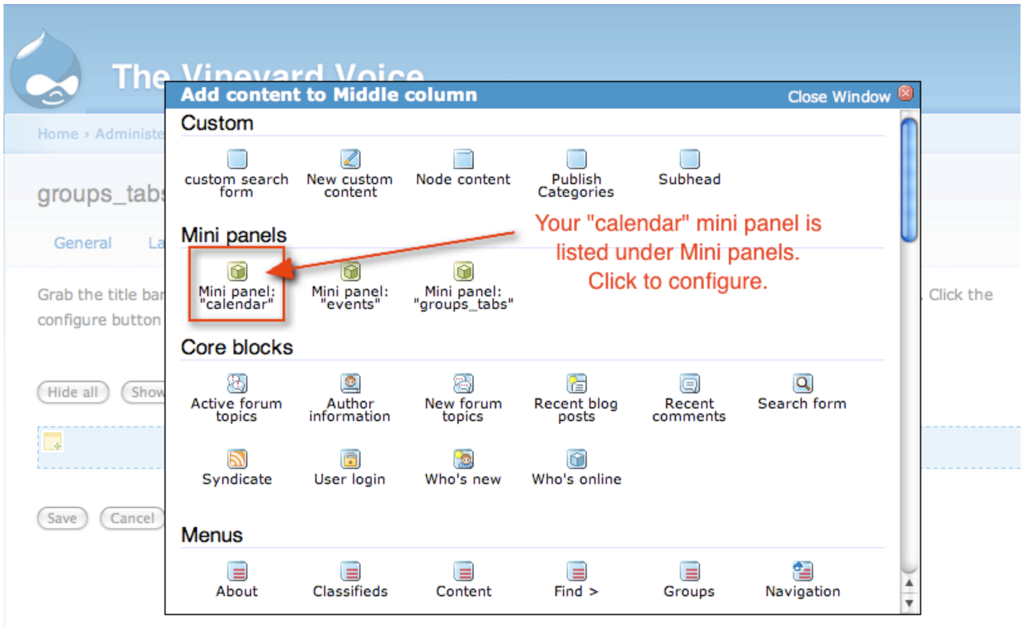
→ Add your "calendar" mini panel by clicking the icon.



Grab the title bar of any pane to drag it to another panel. Click the add pane button in any panel to add more content. Click the configure button on any pane to re-configure that pane.

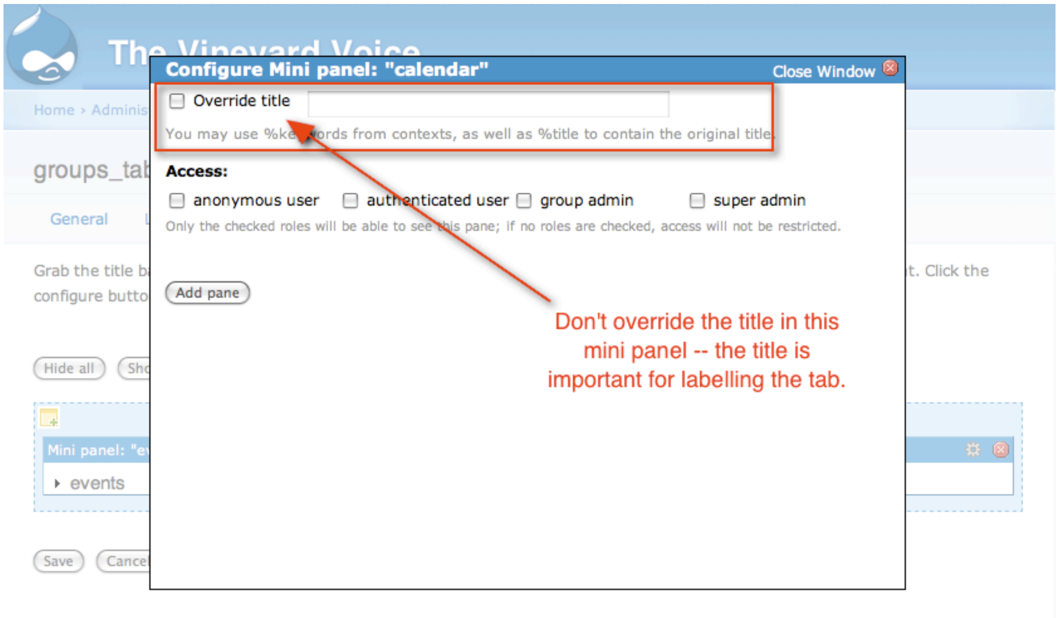
Again, click the add icon.

→ There is a special section in the Add content pop-up. In Panels 2 there is a Panels node selection option that allows you to filter the types of nodes that will show up in this window. If you have created multiple mini panels (I have "events" too) select those as well.



Your "calendar" mini panel is listed under Mini panels. Click to configure.

→ I have not overridden the title here. Click "Add pane".

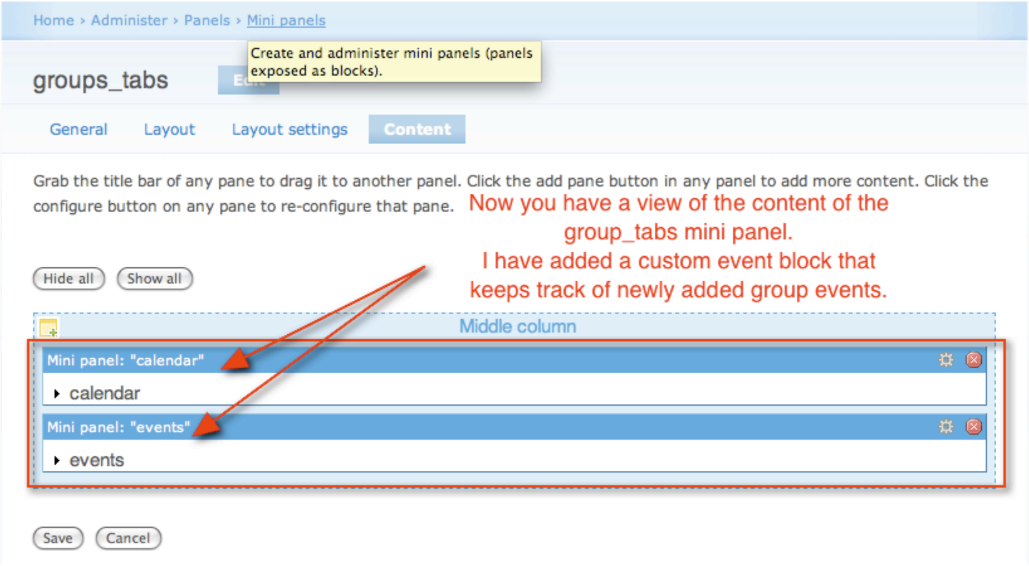


You may use %keywords from contexts, as well as %title to contain the original title.

Only the checked roles will be able to see this pane; if no roles are checked, access will not be restricted.

Don't override the title in this mini panel -- the title is important for labelling the tab.

→ Once you have added the mini panel(s), they will be shown in your groups_tabs Content area. You can click the arrow/bullet next to the Mini panel title to preview its contents.



Home › Administer › Panels › Mini panels

Create and administer mini panels (panels exposed as blocks).

groups_tabs

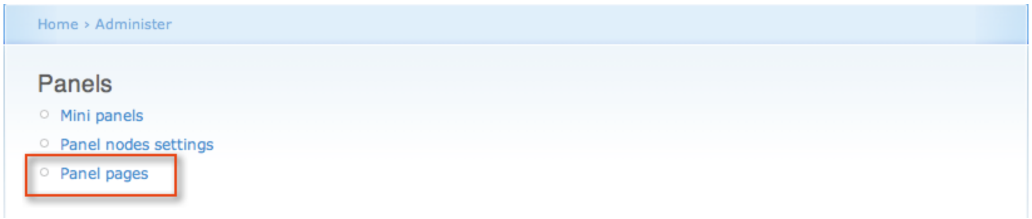
Grab the title bar of any pane to drag it to another panel. Click the add pane button in any panel to add more content. Click the configure button on any pane to re-configure that pane.

Now you have a view of the content of the group_tabs mini panel.
I have added a custom event block that keeps track of newly added group events.

VIII (Part Two) __**Placing Tabbed Mini Panel into your page layout**__

This section will not be outlined in detail. It is more of a bonus section to see how the panels page handles nested mini panels and to return us to the first UI image now with a little more understanding.

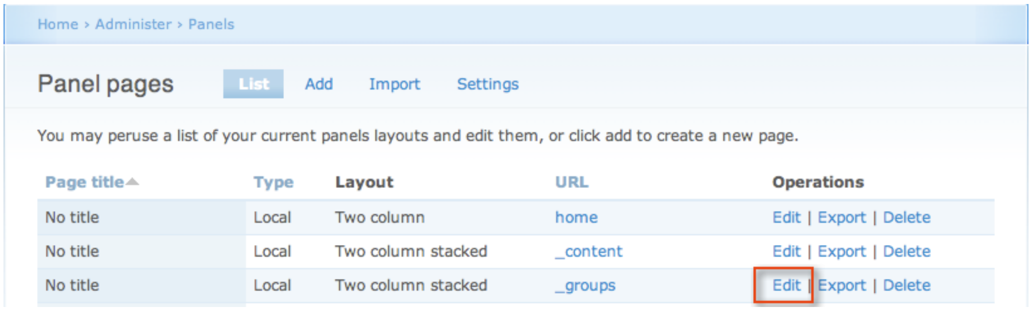
→ Go to your Panels admin, and select panels pages



Home › Administer

Panels

→ You must first create a panel. I have many and **Edit** the _groups panel page.



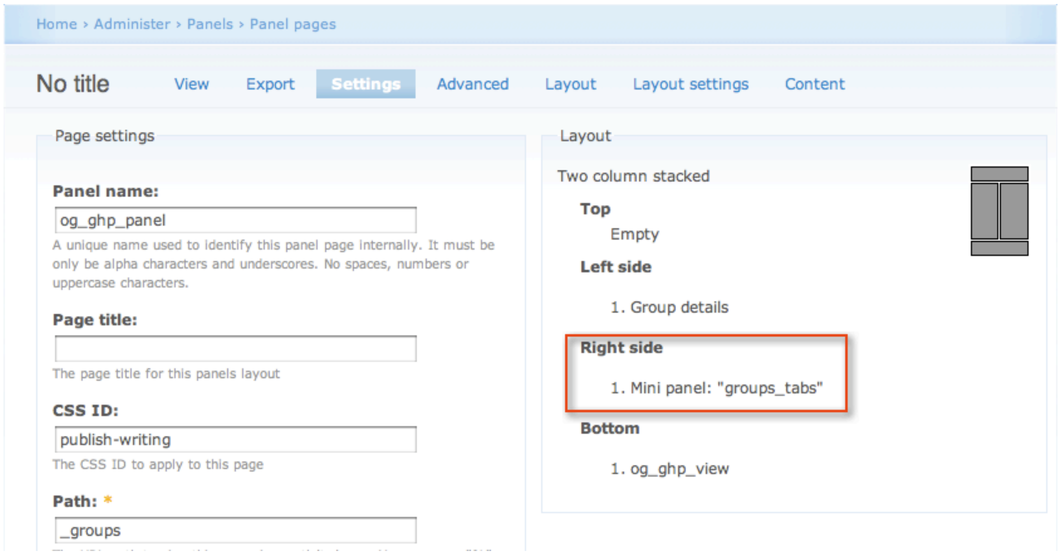
Home › Administer › Panels

Panel pages

You may peruse a list of your current panels layouts and edit them, or click add to create a new page.

Page title ▲	Type	Layout	URL	Operations
No title	Local	Two column	home	Edit | Export | Delete
No title	Local	Two column stacked	_content	Edit | Export | Delete
No title	Local	Two column stacked	_groups	Edit | Export | Delete

→ In the layout area of the Settings section, you can see here I have already placed the "groups_tabs" mini panels in the right side of a Two column stacked layout.



Page settings

Panel name:

og_ghp_panel

A unique name used to identify this panel page internally. It must be only be alpha characters and underscores. No spaces, numbers or uppercase characters.

Page title:

The page title for this panels layout

CSS ID:

publish-writing

The CSS ID to apply to this page

Path: *

_groups

Layout

Two column stacked

Top

Empty

Left side

1. Group details

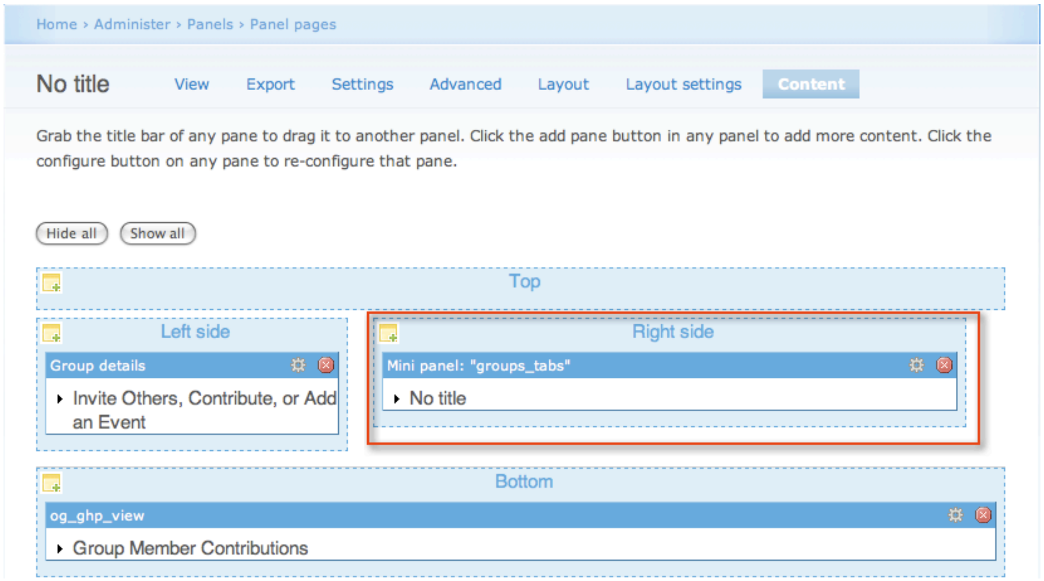
Right side

1. Mini panel: "groups_tabs"

Bottom

1. og_ghp_view

→ This is where you would Add the tabbed mini panel "groups_tabs" as you have done in earlier steps.



Grab the title bar of any pane to drag it to another panel. Click the add pane button in any panel to add more content. Click the configure button on any pane to re-configure that pane.

Group details

Invite Others, Contribute, or Add an Event

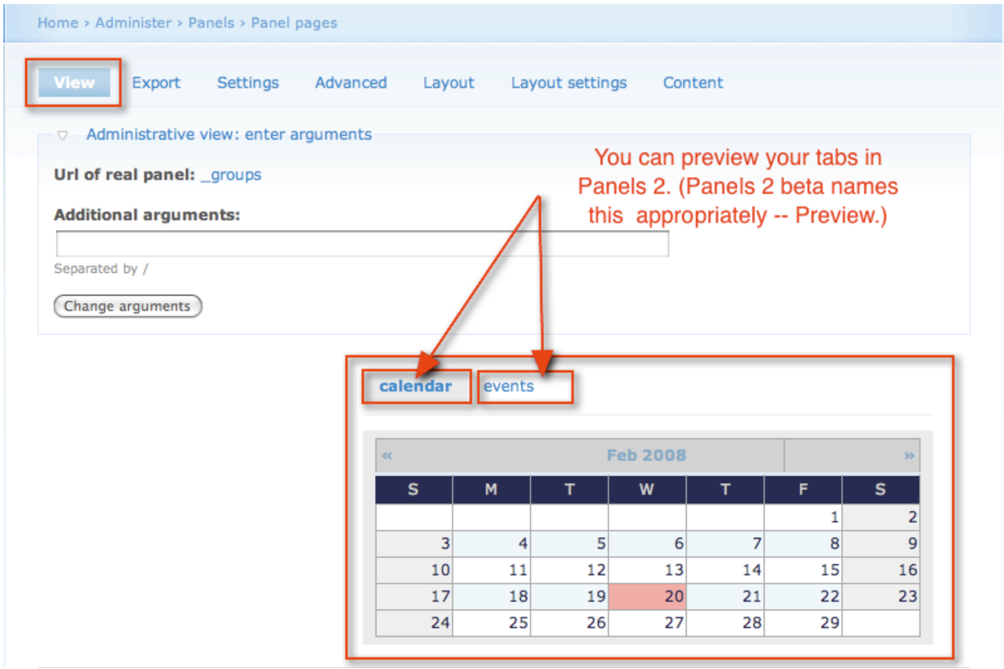
Mini panel: "groups_tabs"

No title

og_ghp_view

Group Member Contributions

→ If you select Preview (in Panels 2 beta, it is "View" in my alpha release) you can now see your tabbed mini panel display, and can toggle between calendar and events. (I have not added any additional arguments.)



Url of real panel: _groups

Additional arguments:

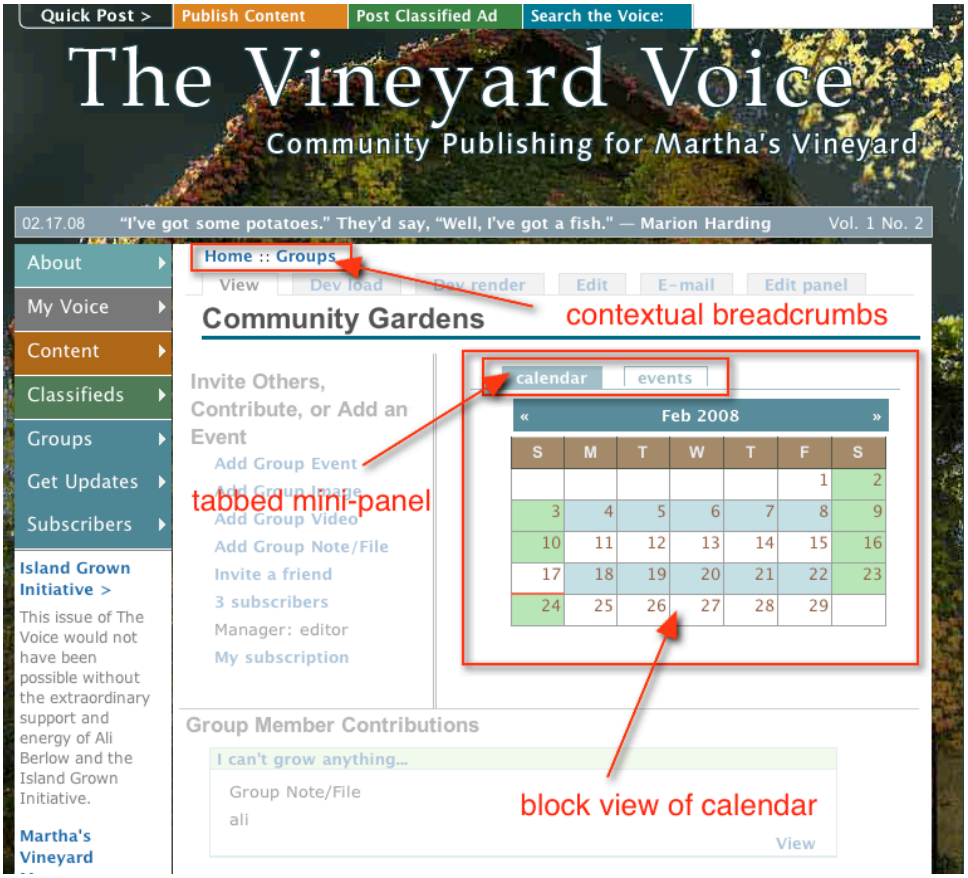
Separated by /

You can preview your tabs in Panels 2. (Panels 2 beta names this appropriately -- Preview.)

Feb 2008

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	

→ Once again, the themed "final result."



02.17.08 "I've got some potatoes." They'd say, "Well, I've got a fish." — Marion Harding Vol. 1 No. 2

Island Grown Initiative >

This issue of The Voice would not have been possible without the extraordinary support and energy of Ali Berlow and the Island Grown Initiative.

Martha's Vineyard

Community Gardens

Invite Others, Contribute, or Add an Event

Manager: editor

contextual breadcrumbs

tabbed mini-panel

Feb 2008

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	

Group Member Contributions

I can't grow anything...

Group Note/File

ali

block view of calendar

IX Clarifying and making uniform the Breadcrumb Context

All the above "functionality" development enables the user to create context sensitive group calendars. But Drupal's existing breadcrumbing system doesn't talk to the group context very well. So we created a template.php theme function override that tells the breadcrumbs to behave.

I worked with Ted Sirbinsky of Lullabot and MothersClick fame on solving this problem too. The issue was that the Groups home page recognizes the URL passed through the view. But calendar module and Drupal's default theme_breadcrumb() breadcrumb variable aren't set to interact. The Theme function override helps hook the breadcrumb into the group's view, node id and specific arguments.

I wrote this note to Ted when we were working this out:

I like this structure.

Home :: **Groups** :: **The Vineyard Voice**

This structure allows for:
    1) "Groups": group directory selection
    2) "The Vineyard Voice": would take you back to the group you are in -- (When it is also a calendar name it is confusing. I'd like to eliminate that confusion.)
This structure also serves as a real breadcrumb.

I would like to use this basic breadcrumb unit as the prefix for all deeper group calendar breadcrumbs.

For example, this current breadcrumb:

Home :: **Test Calendar** :: **The Vineyard Voice** :: **2008** :: **January 2008**

would be replaced with this structure:

Home :: **Groups** :: **The Vineyard Voice** :: **Test Calendar** :: **January** :: **2008**

To accomplish this,

Ted wrote a theme_breadcrumb() override and called it vv_fix_breadcrumb().
The code is well commented and should be self-explanatory, even to non-phpers.

I will simply add the theme override code and bold the comments so you can understand what is happening:

```
/**
 * Fix breadcrumbs to work the way we want them to, overriding Drupal.
 *
 * @breadcrumb - Currently breadcrumb variable from Drupal.
 */

function vv_fix_breadcrumb($breadcrumb = array()) {

  // if a user is looking at a calendar
  // arg(2) is the group id
  if (arg(1) == 'calendar' && is_numeric(arg(2))) {
    $breadcrumb = array(); // reset to empty so we can rebuild them
    $group = node_load(arg(2));
    $view = views_load_view(88); // the views ID of the calendar view
    $breadcrumb[] = l('Home', '');
    $breadcrumb[] = l('Groups', 'og');
    $breadcrumb[] = l($group->title, 'node/'. $group->nid);
    $breadcrumb[] = l($view->page_title, $view->url .'/'. arg(2));

    // if the view is showing a specific year
    if (is_numeric(arg(3))) {
      $breadcrumb[] = l(arg(3), $view->url .'/'. arg(2) .'/'. arg(3));
    }

    // if the view is showing a specific month
    if (is_numeric(arg(4))) {
      $date = mktime(0, 0, 0, arg(4), 1, arg(3));
      $month = date('F', $date);
      $breadcrumb[] = l($month, $view->url .'/'. arg(2) .'/'. arg(3) .'/'. arg(4));
    }

    drupal_set_breadcrumb($breadcrumb);
  }
  // This section of the breadcrumb override is not for the Group pages, but for the main content types section of the Vineyard Voice
  if (arg(0) == 'publish') {
    $breadcrumb = array(); // reset to empty so we can rebuild them
    $breadcrumb[] = l('Home', '');

    if (arg(1) == 'audio' || arg(1) == 'blog' || arg(1) == 'video' || arg(1) == 'writing' || arg(1) ==
'image') {
      $breadcrumb[] = l(ucfirst(arg(1)), 'publish/'. arg(1));
    }
    drupal_set_breadcrumb($breadcrumb);
  }

  return $breadcrumb;
}
```