

Introduction to Flux Developer Tools

Introduction

When you create a community on the Flux platform, you become an Admin for that community. By default, all Admins are given a basic toolset that is accessible from the Community Manager which allows you to edit many aspects of your community's appearance and content.

As a Flux Developer you gain access to additional tools described in this document. You can use the Flux Markup Language (FML), access a Preview environment and leverage the Static Content tools to perform extensive customization of your site beyond what is available through the regular toolset.

If you are interested in accessing the Developer Tools as described in this document, please email us at support@flux.com to have your account status changed from basic Admin options to Developer mode.

Manage Your Community

Add content to your community and make sure all of the modules and pages look the way you want.

Get started by [inviting people to join](#), adding [photos](#) and [videos](#), and starting some [discussions](#).

Check out our [Partner Documentation](#) for additional information, or [send us an email](#).

Content Approval

Flux moderates all content uploaded to community sites. However, if you've uploaded content that you want to appear immediately use this section to update the approval queue.

Flagged Content

Moderate content that has been flagged by users as inappropriate. You can choose to remove flags or ban flagged content.

Member Management

Manage your site's members. You can ban or feature members from the Member Management tool.

Categories

Set up the categories for your content pages here.

Community settings

Change [Basic Information](#) about your site, such as your community name or logo, choose [Default Thumbnails](#) for users that have not uploaded their own, select [Moderation Settings](#) for your site, and more...

Community Pages & Navigation

See the list of all pages on your community in your [Pages Directory](#). Edit page layouts, [Create New](#) pages, and add or remove pages from your [Site Navigation Editor](#).

Themes

Change the appearance of your site.

Community Media

Add editorial content like [posts](#), [news](#), and [events](#) and [manage content](#) added by members of the community.

Widgets

Add widgets to surface community activity and functionality on your existing site. Add [members online](#), [fShare](#), [photos](#), [videos](#), and more.

Reporting

Find out how many active members you have on your site and what they are doing.

Manage Developer Pages

Manage your static files and master page. Make custom markup changes in preview mode and deploy your changes live.

Advertising

Review advertising information and how to maximize your revenue opportunity.

Partner Documentation

Review our [Partner Documentation](#) for more information on community tools, data exports, [getting Flux on your own domain](#), and more. Still need help? Contact your community manager or [email us](#).

Fig. 1 View of Community Manager, when Advanced/Developer Mode enabled

Terms

Flux Community – a community powered by the Flux Platform

Community Admin – a user with privileges to change the community settings, to manage other users and site content

Custom Markup

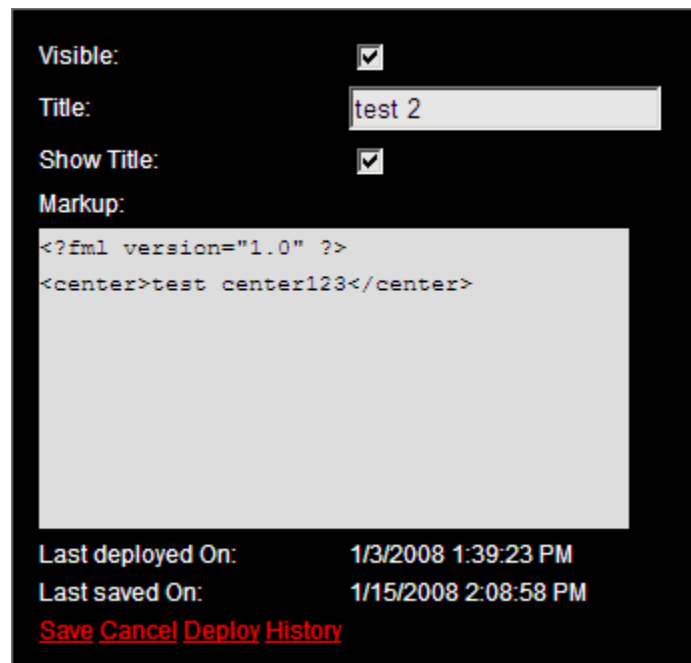
The Flux Markup Language (FML) is only available in Developer mode for communities who want to do extensive customization beyond what is offered in the basic array of configuration properties. It gives the power to directly modify the HTML and server-side code of the pages/modules within the site. The proprietary FML allows any HTML developer to make adjustments to the core behavior and layout of the site without needing to have any understanding of the .NET framework upon which the site is built. FML sits on top of the ASP.NET code and acts as a transparent layer to the developer so that they never need to work with the ASP.NET code directly.

All pages and modules are built as a series of templates. By modifying the HTML and FML of the underlying templates, a very rich level of customization can be achieved. For example, FML can be used to alter the layout of a specific module on the page or to expose additional data which is not exposed by the module by default. Also, the ASP.NET master page which controls the entire site can be adjusted using FML. This allows, for example, the positioning of the Community Menu to be changed, or the items in the footer to be changed.

For additional information, please see the FML documentation.

FML Module Editor

The FML module editor is very similar to the regular module editor, with the addition of a field for editing FML markup and two actions: “Deploy” and “History”.



Visible:

Title:

Show Title:

Markup:

```
<?fml version="1.0" ?>
<center>test center123</center>
```

Last deployed On: 1/3/2008 1:39:23 PM

Last saved On: 1/15/2008 2:08:58 PM

[Save](#) [Cancel](#) [Deploy](#) [History](#)

Fig. 5 FML Module Editor

The FML Module editor includes the following *properties*:

- **Visible** – when selected, the module will be visible to visitors of the page; when the checkbox is unselected, the module will be visible only when viewed via “Edit Mode” - which is available only for the community admin. Use this property when you would like to save work you have done on the module, but are not ready to display it on your live site. When the module is ready to go, simply check the “Visible” box, and it will be displayed to all users.
- **Title & Show Title**– as with all PBT modules on Flux.com, this field and checkbox allow you to edit the title of the module, and determine whether or not that title is displayed to the end user.
- **Markup** – This field contains the FML markup itself. For more information on this topic, see our FML Markup Documentation on developer.flux.com.
- **Last Deployed On** – this is an informational field that displays a time and date stamp of the most recent deployment of your edited module.
- **Last Saved On** – this is an informational field that displays a time and date stamp of the last time when markup was saved (with “Save” button).

Introduction to Flux Developer Tools

The FML Module editor offers the following *actions*:

- **Save** – when you change any property of a module and click “Save” – the entire set of properties will be saved. The only exception to this concerns the **Markup** property. When you change FML markup and click “Save” the markup will be used to render the module only in Preview Mode, as explained above.
- **Cancel** – this option is fairly self-explanatory and acts to close the editor without making any changes.
- **Deploy** – this option acts as a bridge between Preview and Live Modes. When you click this action, the saved FML markup from Preview version (which was saved via the “Save” button) is transferred into FML markup of Live version. In other words, the edited markup is visible to all users when deployed, not just those with the Preview cookie in place.
 - *Note:* “Deploy” circumvents “Save” in the sense that if you have changed markup in **Markup** property without saving it (by clicking “Save”) and instead click “Deploy,” that markup will be presented for both Preview and Live Modes. This is not recommended; it is better to first “Save” it, take a look at the end result in Preview Mode and deploy only when satisfied with your results.
- **History** – this action opens a special window displaying all versions of markup that have been deployed and made visible to Live Mode. History does not include markup edits that have only been saved to Preview Mode, and not deployed. The History list provides a way to revert changes in FML markup to a specific previous version. To do this, simply click the **Rollback** button next to the desired version. This loads the selected FML markup from the **History** to the **Markup** property, so you can then **Save** to Preview Mode or **Deploy** it to Live Mode.

Important restrictions for the FML module editor:

- You cannot edit FML modules unless you have enabled Preview Mode. This requirement is in place to help avoid mistakes that might arise if editing module markup directly on your page as it is immediately rendered for visitors to your community.
- Newly added FML modules will be set to non visible by default – you must actively mark a module as visible in order for it to display on your site. This is to prevent accidental display of an empty module within your community.

Manage Developer Pages

This area of the community manager allows you to keep track of the status of all your customizable elements. You can perform all actions inherent to Preview Mode on this page. You will only see this option if your community has Developer Mode enabled.

Manage Developer Pages

Preview status: **Enabled**

[Disable Preview](#)

To Allow Others To View The Preview...

To show the site in preview mode to someone who does not have admin access, you can provide them the links below. Enable preview mode for the links to appear.

When they copy the "Activate" link into their browser, they will see the site in preview mode.

When they want to see the live site again, they should copy the "Deactivate" link into their browser.

Activate Link: <http://testcomm007.flux-staging.com/-/CommunityPre>

Deactivate Link: <http://testcomm007.flux-staging.com/-/CommunityPre>

Static Content

Static Content includes files such as JS Files, Graphic Images, Flash Files, and Css Files
Upload via FTP to the preview server. You then need to come to this page and deploy the files to live.

[Copy Live to Preview \(Sync\)](#)

[Deploy to Live, No Theme](#)

[Deploy to Live, Include Theme](#)

Master Page

Preview Version	Live Version	Last Saved On	Last Deployed On	Actions
1	1	1/15/2008 2:27:30 PM	1/15/2008 2:27:34 PM	Edit

Custom Markup Modules

Module Title	Page	Preview Version	Live Version	Last Saved On	Last Deployed On	Actions
Custom Markup Module	CLP	1	1	1/15/2008 2:12:31 PM	1/15/2008 2:12:31 PM	Edit
test 2	CLP	20	19	1/15/2008 2:08:58 PM	1/3/2008 1:39:23 PM	Edit Deploy

Fig. 6 Developer Pane

In this example, you can see that Preview Mode is enabled. You can enable and disable it with respective buttons: **Enable Preview** and **Disable Preview** – they toggle depending on whether Preview mode is on or off. The table that appears below the buttons shows an aggregated view of all customizable elements on your pages.

Manage Developer Pages - Static Content

Static content refers to all assets you might have to customize for your community site: images, Jscripts, CSS files, Flash files, etc. You can take the following actions on static content:

1. **Copy Live to Preview (Sync)** – this action copies all static resources from Live Mode to Preview Mode. This means that all static images, Jscripts, CSS files, Flash files, etc. that are used on the site in Live Mode will be copied into Preview Mode and thus will be used also in Preview Mode. This action makes sense when you want to synchronize your Live and Preview Mode before you start to modify static resources in Preview Mode.
2. **Deploy to Live, No Theme** and **Deploy to Live, Include Theme** – these actions allow you to copy all static resources from Preview Mode to Live Mode. These actions should be performed when you have completed tweaking UI or client-side behavior of your community (via Preview Mode) and are ready to push your changes to Live Mode and start using those new static resources on your community website.

There is the following difference between those two actions:

- Your community theme is considered a static resource, as it is a CSS file which is generated via the Community Theme page. You may decide whether the community theme should be applied along with all the rest of the static resources while copying them from Preview to Live (**Deploy to Live, Include Theme** action) or not (**Deploy to Live, No Theme** action). In other words, if you click **Deploy to Live, No Theme**, all resources will be copied excluding the .css file associated with your Community theme. This feature might be used when you have finished tweaking pages with static content (images, Jscripts, Flash files etc.) but still not finished setting up community theme. In this case, it makes sense to deploy everything except that unfinished piece, which you can return to later for completion. On the other hand, if you will click **Deploy to Live, Include Theme** everything, including community theme .css, will be copied, and visible to community visitors.

Manage Developer Pages - Master Page

The master page is a container applied to all pages within your community site. So, markup stored in the master page will appear on all pages of your community. The master page is generally used to place ad banners at the top or at the bottom of the page, and apply navigation menu, quick menu or other elements that you would like to appear on each page within your community. In the developer pane view, the following fields apply to the master page:

- **Preview Version** – displays a version number for the markup used to render master page while in Preview Mode.
- **Live Version** – displays a version number for the markup used to render master page while in Live Mode (what visitors actually see).

Both these numbers are for your convenience to track how far away the markup of Preview Mode is from the markup of Live Mode. If the two numbers are equal, that means the same markup version is being used in Preview Mode and in Live Mode. If **Preview Version** number is greater than **Live Version** number that means Preview version of markup contains some differences, when compared to Live, and serves as a reminder that while you are in Preview Mode you might see your pages differently than regular visitors, who always see the Live markup version.

- **Last Deployed On** – displays a timestamp of when you last updated the markup of Live Mode.
- **Last Saved On** – displays a timestamp of when you last updated the markup of Preview Mode.

Both these timestamps are for your convenience to recall when you last updated Preview or Live version of markup.

Available actions you can perform:

- **Edit** - when you click this link, a sliding text area will open to allow you to customize markup of master page in Preview Mode (you cannot customize markup of master page in Live Mode – you can only customize one of Preview Mode and then deploy it to make it affect Live Mode). In the text area you should specify FML markup for your master page.

Fig. 7 Sliding text area for customizing markup of master page in Preview Mode

Introduction to Flux Developer Tools

- **History** – opens a special window with list of all markups that have ever been deployed to Live Mode. This button acts quite similarly to History button in FML module editor; however the window that is opened contains versions of markup of master page not versions of markup of FML module. The same way you would revert FML module markup you can revert master page markup using History window – markup will loaded into text area so you can modify it and/or save.
- **Deploy** – again, this action is similar to that in the FML module editor. When selected, the markup of current Preview version is pushed to be markup of Live version (causing regular visitors to see new master page once you will have deployed it). This button is only available if **Preview Version** number is greater than **Live Version** number – which lets you know that you might have something to deploy, as some changes actually exist.

Manage Developer Pages – Custom Modules

The information available about each FML module is the same as for master page as are the available actions. The only additional information for FML modules is the title of the module and the name of the page that hosts it – so you can distinguish between modules easily. Please reference the section above to see what information and actions are available.

For more information on custom markup, please see our FML Documentation at developer.flux.com.

Preview Mode

Introduction

Preview Mode is a new approach to customizing markup code for your community. Previously, when customizing the markup of your community’s master page, or changing custom markup of individual PBT modules, your changes appeared on your community site immediately upon saving. This instantaneous update could easily lead to errors in modules or pages. To improve this process, we have introduced a new approach for customizing elements of your community’s site – “Preview Mode”. Like the name suggests, the purpose of Preview Mode is to allow you to customize your markup and take a look at the end results, before actually effecting changes on your live community.

<pre><?fml version="1.0" ?> <div>version 2</div></pre>	Save Cancel History
--	--

How It Works

When you enable Preview Mode, a cookie is passed to your machine. When this cookie is in place, it determines which markup – Preview Mode or Live - should be displayed by the server to your machine. “Live Mode” refers to the normal display of code as available when the Preview cookie is disabled.

Each FML module and master page, along with associated static content (for example, community theme) has two different versions: Preview and Live. When Preview Mode is enabled, the server will display the Preview version of FML control markup, master page markup and static content. Accordingly when Preview Mode is disabled (in other words, when Live Mode is on) opposite versions of markups and set of static content are displayed.

Because it is cookie-based, Preview Mode is machine- specific, so if you enable Preview Mode on one computer and then log in to your community from another computer, Preview Mode will not be enabled for that second machine. Additionally, if you log out and then log in again with a different account on the same computer, Preview Mode will still be shown. To switch Preview Mode off you need to clear the cookie file from your browser. Alternatively if you have access to the Developer Pane, you can click “disable” in this view.

Working with Preview Mode

You can use Preview Mode when editing the following elements of your community:

- Master page markup – this is a markup container for all pages within a community site
- FML module markup– this module is dedicated to render special FML markup which you can specify
- Static assets (e.g. graphic and Flash files, Jscripts, CSS files, etc.) – any static content you might need to customize appearance and behavior of your community

When you are in “Edit” mode as the community admin, you can handle these elements in two places: a special page called “Developer Pane,” where all aggregated information about all customizable elements are gathered; and an FML module editor.