

Contents

Introduction	3
Getting Started with FML.....	4
Page Structure	4
Data Retrieval	4
Customizing the Master Page	5
Sample FML Master Page	5
Master Page FML Tags and Usage	5
Customizing Page Layouts.....	6
Creating Customized Modules	7
Anatomy of a Module.....	7
Data Retrieval	7
Data Presentation.....	7
Examples.....	7
Common FML Elements	8
Overview.....	8
Supported Elements	8
Usage Reference.....	8
Master Page FML Elements	10
Overview.....	10
Supported Elements	10
Container Elements.....	10
Nested Elements	10
Example of Usage	10
Usage Reference.....	11
Container Elements.....	11
Nested Elements	11
Page Layout FML Elements	13
Overview.....	13
Supported Elements	13
Usage Reference.....	13
Data Container FML Elements	14

FML is subject to change and as a result the accuracy of this document is not guaranteed

Single Item Container	14
Item List Container - General.....	15
Item List Container - Topics	17
Item List Container - Comments.....	18
Data Bound FML Elements.....	20
Overview.....	20
Supported Elements	20
Usage Reference.....	20
Data Elements.....	23
General	23
Content – All Types.....	24
Content – Video.....	25
Content – Text (i.e. Post, News, Discussion Topic).....	26
Content – Events	27
Content – Topics.....	28
Content – Albums.....	29
Content – Audio.....	30
People – All Types.....	31
Content Categories – All Content Types.....	32
Content Categories – Topics.....	33
User’s Communities.....	34
Comments	35
Signup/Login FML Elements.....	36
Appendix 1 - FML Enumerations.....	37
Available Picture Sizes	37
Date Formats	37
DAAPIParam values	37
CommunityRole values	37
EventDetail values	37
HTMLCleanerOptions values	38
BannerPlacementId values.....	38

Introduction

The **Flux Markup Language** (FML) allows you to create highly customized community websites for your users. FML enables the Flux platform to be extended to meet your particular needs. FML is a combination of standard HTML tags with additional proprietary Flux-specific tags added. The Flux-specific tags are used to access the social data and layer it into the community pages.

Any web developer who is familiar with HTML can easily get started with FML without needing to have any understanding of the .NET framework upon which the site is built. FML currently allows all standard HTML tags to be used, and CSS inclusion and JavaScript is fully supported by FML.

Getting Started with FML

FML can be used to customize virtually every part of the community. Every FML document always starts with the following declaration:

```
<?fml version="1.0" ?>
```

Each FML tag is always prefixed with “<fx:” to indicate that it is a special tag and must be processed by the FML Interpreter. An example of an FML tag to add a menu to the page is as follows:

```
<fx:menu />
```

The structure of FML pages is below. More detail on customizing each section using FML can be found in later sections of this documentation.

Page Structure

The **Master Page** is the global container for every page on the site. It constructs the basic framework of the site, including the Menu, Ad Placements, Page Content, Footer etc. Each community can have only 1 Master Page.

Each customizable page has a pre-defined **Page Layout** which specifies the structure of content on the page. This can be configured to have any kind of tabular layout. An example of a simple page layout would contain 3 columns of content, but FML can be used to create any custom layout desired.

Within each page on the site, **Modules** are added to specify the actual content for the page. For example, if you want to display a gallery of community members you can easily add a module which displays member thumbnails and metadata. Flux provides a number of preset modules with customizable properties to achieve basic customization of presentation. However, FML allows you to achieve advanced customization by presenting the data in whatever format desired.

Data Retrieval

The Flux **Data Access API** (DAAPI) provides the mechanism for retrieving publicly available community data. FML uses DAAPI queries to make requests for data. The output is offered in multiple formats (XML, Atom, JSON) for convenience. The DAAPI can be used both inside the community and externally (for integration of outbound widgets and promotions). For additional information, see the full DAAPI documentation.

Customizing the Master Page

The Master Page is customized via the Developer Panel which can be accessed from your Community Management page. This section gives an example of a sample Master Page and describes how FML is used to customize it.

Sample FML Master Page

```
<?fml version="1.0" ?>
<fx:masterPage googleTracker id="UA-1162848-6" addressBarIcon=http://www.mydomain.com/addressBar.ico">
<div class="page Flux">
  <div class="pageHeader">
    <fx:quickNavigation />
    <fx:adBanner PlacementId="Right300x250" />
    <fx:menu />
  </div>
  <div class="pageContent">
    <fx:pageContent />
  </div>
  <div class="pageFooter">
    <table width="100%" cellpadding="0" cellspacing="0" border="0">
      <tr>
        <td valign="middle">
          <fx:footer />
        </td>
        <td valign="middle" align="right"><a href="http://community.vinylpulse.com/-
/Messaging/InviteFriends.aspx"> Tell your friends about VinylPulse</a></td>
      </tr>
    </table>
  </div>
</div>
</fx:masterPage>
```

Master Page FML Tags and Usage

The Master Page is always contained within the `<fx:masterPage>` tag. This denotes that all nested elements form part of the Master Page. A limited number of FML tags are currently available for use within the Master Page which are illustrated in the example above. Full documentation can be found in the [Master Page FML Elements](#) section.

Important - It should be noted that the following code is mandatory and should not be removed. This tag is a placeholder for the actual content of the page, and without it the site will not render correctly.

```
<div class="pageContent">
  <fx:pageContent />
</div>
```

Additionally, it should be noted that the Master Page is inside the `<BODY>` of the document, and so cannot be used to customize anything outside of page body. Therefore, the Master Page cannot be used to inject tags such as the `<BODY>`, `<TITLE>` and `<META>` tags.

Customizing Page Layouts

The layout of any customizable page can be modified via the Developer Panel which can be accessed from your Community Management page. Layouts can be customized using standard HTML with the inclusion of specific `<fx:zone>` FML tag to inject the content into each area of the page. See the [Page Layout FML Elements](#) section for more details

```
<?fml version="1.0" ?>
<table border="0" cellpadding="0" cellspacing="0" >
  <tr valign="top">
    <td class="zone zoneLeftWide zoneContent">
      <fx:zone id="zone2" />
    </td>
    <td class="zone zoneRightWide zoneContent">
      <fx:zone id="defaultZone" default="true" />
    </td>
  </tr>
</table>
```

Creating Customized Modules

Anatomy of a Module

An FML module is comprised of a dataset which retrieves data from the Flux DAAPI, and a visual representation of this data. FML provides a means of easily representing the data within the module. Modules generally fall into the category of either a single-element (i.e. Featured Video), or multi-element (i.e. Recent Visitors). Both of these can be easily represented within FML.

Data Retrieval

Data is retrieved in the form of a query to the Flux DAAPI. For example, the following query will return the 5 most active users within the community:

```
datasource="/feeds/people/-/active/?max-result=5"
```

For additional information, see the full DAAPI documentation.

Data Presentation

Data can be presented in single element, or list element format. Single Element allows one specific item to be displayed, and the data retrieved from the DAAPI contains extensive metadata for the item. By contrast, List Element allows lists of multiple items to be displayed, and data retrieved from the DAAPI contains a smaller subset of metadata per item. The FML syntax used in each case differs slightly.

In **single element** modules, the `<fx:single-element>` container element must be used. In **list element** modules, the `<fx:grid>` container element must be used. Both of these containers require a DAAPI query as a parameter to determine the data source and contain nested elements which control data presentation. The [Data Container FML Elements](#) section details usage of `fx:single-element` and `fx:grid`. Also, see the [Data Elements](#) section for full documentation on data elements available when displaying single content items (via `fx:single-element`) and lists of content items (via `fx:grid`).

Examples

You can see an extensive set of FML module examples here: <http://developer.flux.com/examples>

This gives some simple introductory examples to get started using FML to build a customized module. Also, there are some more advanced examples to demonstrate the richer, more heavily customized UI which can be achieved by utilizing FML.

Common FML Elements

Overview

Common FML elements are supported anywhere FML is used (i.e. within Master Page, Page Layouts, Modules)

Supported Elements

- [fx:adBanner](#)
- [fx:url](#)
- [fx:staticUrl](#)
- [fx:visible](#)
- [fx:loggedUserUrl](#)

Usage Reference

fx:adBanner

Description

Renders an advertisement banner in the specified location, targeted to the required placement

Attributes

Name	Required?	Type	Description
PlacementId	Yes	BannerPlacementId enum	Designates size and position of ad banner on the page

Example

```
<fx:adBanner PlacementId="Right300x250" />
```

fx:url

Description

This generates a link to internal community pages. It must use a relative URL as a parameter, and will append the current community domain to build the full URL. It should not be used for external URLs.

Attributes

Name	Required?	Type	Description
Path	Yes	string	Relative URL for link

Example

```
<fx:url path="/Community/BrowseUsers.aspx"/>
```

fx:staticUrl

Description

This generates a link to the Flux static servers to serve content uploaded via the Static Content process. It must use a relative URL as a parameter and the full static content URL will be auto-generated. The result is a URL such as

http://static[n].fluxstatic.com/-/clients/[community_name]/img/img.gif

Attributes

Name	Required?	Type	Description
------	-----------	------	-------------

FML is subject to change and as a result the accuracy of this document is not guaranteed

Path	Yes	String	Relative URL for link
------	-----	------------------------	-----------------------

Example

```
<fx:staticUrl path="/img/img.gif"/>
```

fx:visible

Description

Allows visibility of content on the page to be restricted dependant on user's login state and what permissions (role) they have within the current community. This FML element can be wrapped around any content on the page to control the visibility.

Attributes

Name	Required?	Type	Description
To	No	CommunityRole enum	Pipe () delimited list of roles who CAN view content nested within this FML element
NotTo	No	CommunityRole enum	Pipe () delimited list of roles who CANNOT view content nested within this FML element

Example

The following example will display a link to create a new blog post for Admins only

```
<fx:visible to="Administrator">
  <a href='<fx:url path="/AdminSite/CreatePost.aspx" />'>Create new blog entry</a>
</fx:visible>
```

fx:loggedUserUrl

Description

Displays the profile URL for the currently logged-in user within the scope of the current community.

Attributes

None

Example

```
<fx:loggedUserUrl/>
```

Master Page FML Elements

Overview

All Master Page elements must be nested inside the `fx:masterPage` container as shown below. These FML elements can only be used within the Master Page editor, and cannot be used elsewhere on the site.

Supported Elements

Container Elements

[fx:masterPage](#)

Nested Elements

[fx:pageContent](#)

[fx:menu](#)

[fx:quickNavigation](#)

[fx:footer](#)

[fx:inviteFriendsUrl](#)

[fx:globalContentSearch](#)

[fx:masthead](#)

Example of Usage

```
<fx:masterPage>
<div class="page Flux">
  <div class="pageHeader">
    <fx:globalContentSearch />
    <fx:quickNavigation />
    <div class="clearFloat">&nbsp;</div>
    <fx:masthead />
    <fx:menu />
  </div>
  <div class="pageContent">
    <fx:pageContent />
  </div>
  <div class="pageFooter">
    <table width="100%" cellpadding="0" cellspacing="0" border="0">
      <tr>
        <td align="right"><fx:inviteFriendsUrl /></td>
      </tr>
    </table>
  </div>
</div>
```

Usage Reference

Container Elements

fx:masterPage

Description

This is the container for the Master Page of the site. All HTML and FML markup to customize the site's Master Page must be placed inside of this tag.

Attributes

Name	Required?	Type	Description
GoogleTracker	No	String	Allows Google tracking code to be embedded into the site. This parameter contains the unique Google Analytics ID to be used
AddressBarIcon	No	String	Allows an image icon to be specified to be used in the browser address bar

Example

```
<fx:masterPage googleTracker="UA-1162848-6" addressBarIcon="http://mydomain/mydomain.ico">  
[template]  
</fx:masterPage>
```

Nested Elements

fx:pageContent

Description

This is the placeholder for all content in the page. It is mandatory and must not be removed.

Attributes

None

Example

```
<fx:pageContent />
```

fx:menu

Description

Renders the community menu

Attributes

None

Example

```
<fx:menu />
```

fx:quickNavigation

Description

Renders the Quick Navigation bar, which contains links to all personal pages for logged-in users. For logged-out users it prompts to sign-in or join the community.

Attributes

Name	Required?	Type	Description
ShowLogo	No	Boolean	Determines whether to display community logo. By default it will be displayed.
LogoSize	No	ThumbnailSize enum	Size of the community logo. Default if not specified is 25x25

Example

```
<fx:quickNavigation ShowLogo="true" LogoSize="25x25" />
```

fx:footer

Description

Renders a standard footer for the community containing default links.

Attributes

None

Example

```
<fx:footer />
```

fx:inviteFriendsUrl

Description

Renders the "Tell you friends about [Community name]" text as link to the invite friend page.

Attributes

None

Example

```
<fx:inviteFriendsUrl />
```

fx:globalContentSearch

Description

Renders a search bar on the page.

Attributes

None

Example

```
<fx:globalContentSearch />
```

fx:masthead

Description

Renders the masthead on the page.

Attributes

None

Example

```
<fx:masthead />
```

Page Layout FML Elements

Overview

These FML elements can only be used within the Page Editor when editing/adding custom PBT pages.

Supported Elements

[fx:zone](#)

Usage Reference

fx:zone

Description

Renders an empty container for a “zone” within a PBT page. This allows modules to be dragged and dropped into this area on the page. Each PBT page must have at least 1 zone defined.

Attributes

Name	Required?	Type	Description
Id	Yes	String	Designates unique id for the zone
Default	No	Boolean	Determines which zone new modules will be added to by default. By default this is false. Only 1 zone should be set as default

Example

```
<table id="ModuleTable" border="0" cellpadding="0" cellspacing="0">
  <tr class="doublePageViewer" valign="top">
    <td class="zoneContent">
      <fx:zone id="zone1" />
    </td>
    <td class="zoneContent">
      <fx:zone id="zone2" default="true"/>
    </td>
  </tr>
</table>
```

Data Container FML Elements

Single Item Container

fx:single-element

Description

Container for rendering a single item of content or single user. It requires a DAAPI query to be provided to request the data to be rendered. **Note – it only accepts single item DAAPI queries, not list queries.**

Attributes

Name	Required?	Type	Description
DataSource	Yes	String	DAAPI query to format data request
Master	No	String	An ID can be specified to bind a single element to a master grid element. This allows for “slideshow” style functionality whereby clicking on a thumbnail in the grid updates the content displayed in the single element

Example

```
<fx:single-element datasource="visitor">
  [template]
</fx: single-element>
```

Item List Container - General

fx:grid

Description

Container for rendering item lists of content, people or categories. This element contains 1 or more columns, so can represent a simple list, or a tabular “gallery” layout. It recognizes a number of nested FML elements which are listed below. These nested elements can only be used within the scope of an `<fx:grid>` container. This container requires a DAAPI query to be provided to request the data to be rendered. **Note – it only accepts DAAPI list queries, not single item queries.**

Nested FML Elements

FML Element	Required?	Syntax	Description
fx:header	No	<code><fx:header></code> <code>[template]</code> <code></fx:header></code>	Contains HTML markup and FML elements to render the list’s header
fx:item	Yes	<code><fx:item></code> <code>[template]</code> <code></fx:item></code>	Contains HTML markup and FML elements to render the actual item. This is a repeater which renders n times, depending on the list size
fx:alternatingitem	No	<code><fx:alternatingItem></code> <code>[template]</code> <code></fx:alternatingItem></code>	Provides a mechanism of modifying the visual appearance of alternating items in the dataset
fx:columnSeparator	No	<code><fx:columnSeparator></code> <code>[template]</code> <code></fx:columnSeparator></code>	Allows HTML to be placed as a vertical separator between columns. This is only useful if using a gallery layout with more than 1 column
fx:rowSeparator	No	<code><fx:rowSeparator></code> <code>[template]</code> <code></fx:rowSeparator></code>	Allows HTML to be placed as a horizontal separator between rows in the list
fx:noResult	No	<code><fx:noresult></code> <code>[template]</code> <code></fx:noresult></code>	In the event of no data being returned, this markup will be shown
fx:footer	No	<code><fx:footer></code> <code>[template]</code> <code></fx:footer></code>	Contains HTML markup and FML elements to render the list’s footer

Attributes

Name	Required?	Type	Description
DataSource	Yes	String	DAAPI query to format data request
Columns	No	Int	Number of columns to display. Allows for a grid-based layout. If no specified, defaults to 1
Id	No	Int	Allows for more than 1 list within the same module. This must only be specified if more than 1 <code><fx:grid></code> container is contained within the same FML document
Master	No	String	An ID can be specified to bind a single element to a master grid element. This allows for “slideshow” style functionality whereby clicking on a thumbnail in the grid updates the content displayed in the single element

Example

```
<fx:grid datasource="/feeds/people/-/active/?max-result=5" columns="3">
```

```
<fx:header>[template]</fx:header>
<fx:item>[template]</fx:item>
<fx:alternatingItem>[template]</fx:alternatingItem>
<fx:columnSeparator>[template]</fx:columnSeparator>
<fx:rowSeparator>[template]</fx:rowSeparator>
<fx:noresult>[template]</fx:noresult>
<fx:footer>[template]</fx:footer>
</fx:grid>
```

Item List Container - Topics

fx:topicList

Description

Container for rendering topic lists. This element contains only 1 column. It recognizes a number of nested FML elements which are listed below. These nested elements can only be used within the scope of an `<fx:topicList>` tag. The element checks parameters of the query string and if it contains parameter "tclid=[category id]" it shows topics for the specific category with [category id].

Nested FML Elements

FML Element	Required?	Syntax	Description
fx:header	No	<code><fx:header></code> <code>[template]</code> <code></fx:header></code>	Contains HTML markup and FML elements to render the list's header
fx:item	Yes	<code><fx:item></code> <code>[template]</code> <code></fx:item></code>	Contains HTML markup and FML elements to render the actual item. This is a repeater which renders n times, depending on the list size
fx:alternatingitem	No	<code><fx:alternatingItem></code> <code>[template]</code> <code></fx:alternatingItem></code>	Provides a mechanism of modifying the visual appearance of alternating items in the dataset
fx:noResult	No	<code><fx:noresult></code> <code>[template]</code> <code></fx:noresult></code>	In the event of no data being returned, this markup will be shown
fx:footer	No	<code><fx:footer></code> <code>[template]</code> <code></fx:footer></code>	Contains HTML markup and FML elements to render the list's footer
fx:moderatorBar	No	<code><fx:moderatorBar></code>	Embeds moderation controls for the topics. This is implicitly only visible to users with permission to perform moderation tasks

Attributes

Name	Required?	Type	Description
Pager	No	String Available values: None Top Bottom Both (default)	Enable/disable pagers for the list
numberTopics	Yes	Int	Max number of topics to show on one page in the list
returnUrl	Yes	String	Provides a link to allow breadcrumbs to be built. This is intended to link back to the Discussion Categories page

Example

```
<fx:topiclist pager=" Top" numberTopics="5" returnUrl="/DiscussionCategories.aspx">
  <fx:header>[template]</fx:header>
  <fx:item>[template]</fx:item>
  <fx:alternatingItem>[template]</fx:alternatingItem>
  <fx:noresult>[template]</fx:noresult>
  <fx:footer>[template]</fx:footer>
</fx:topicList>
```

Item List Container - Comments

fx:comment-grid

Description

Container for rendering comments. This element contains only 1 column. It recognizes a number of nested FML elements which are listed below. These nested elements can only be used within the scope of an `<fx:comment-grid>` tag. It requires a DAAPi comments query to be specified to determine which comments to display.

Nested FML Elements

FML Element	Required?	Syntax	Description
fx:header	No	<code><fx:header></code> <code>[template]</code> <code></fx:header></code>	Contains HTML markup and FML elements to render the list's header
fx:item	Yes	<code><fx:item></code> <code>[template]</code> <code></fx:item></code>	Contains HTML markup and FML elements to render the actual item. This is a repeater which renders n times, depending on the list size
fx:alternatingitem	No	<code><fx:alternatingItem></code> <code>[template]</code> <code></fx:alternatingItem></code>	Provides a mechanism of modifying the visual appearance of alternating items in the dataset
fx:columnSeparator	No	<code><fx:columnSeparator></code> <code>[template]</code> <code></fx:columnSeparator></code>	Allows HTML to be placed as a vertical separator between columns. This is only useful if using a gallery layout with more than 1 column
fx:rowSeparator	No	<code><fx:rowSeparator></code> <code>[template]</code> <code></fx:rowSeparator></code>	Allows HTML to be placed as a horizontal separator between rows in the list
fx:noResult	No	<code><fx:noresult></code> <code>[template]</code> <code></fx:noresult></code>	In the event of no data being returned, this markup will be shown
fx:footer	No	<code><fx:footer></code> <code>[template]</code> <code></fx:footer></code>	Contains HTML markup and FML elements to render the list's footer

Attributes

Name	Required?	Type	Description
DataSource	Yes	String	
Id	No	String	Allows for more than 1 comment list within the same module. This must only be specified if more than 1 <code><fx:comment-grid></code> container is contained within the same FML document
showGreetingPanel	No	Boolean	Determines whether to show the "Hi [display name]" greeting. By default it is hidden
showViewSelector	No	Boolean	Determines whether to show links to allow users to switch between "Threaded" and "Most Recent" views. BY default they are hidden.
addButtonStyle	No	String	Optional CSS style to be applied to Add Comment button
addButtonText	No	String	Allows the text in the Add Comment button to be customized.

Example

```
<fx:comment-grid datasource="/feeds/comments/0[content_ucid]/?list_type=threaded&max-results=10">
  <fx:header>[template]</fx:header>
  <fx:item>
    <fx:item-comment-creator-thumbnail size="50x50" MouseOverEffect="highlight"/>
    <a href="<fx:item-comment-creator-profileUrl />"><fx:item-comment-creator-title
  /></a>&nbsp;said:
    <fx:item-comment-text MaxLength="500"/>
    <fx:item-comment-dateCreated format="LongDate"/>
  </fx:item>
  <fx:footer>[template]</fx:footer>
</fx:comment-grid>
```

Data Bound FML Elements

Overview

These FML elements bind themselves to the DAAP query used within an `<fx:grid>` data container. The DAAP query used by the data container is modified dynamically by these elements to change the data delivered to the page. Functionality such as pagination, search, sorting, filtering is available using these elements.

Note - they should only be used with content lists, not when displaying single content items.

Supported Elements

[fx:pager](#)
[fx:searchBar](#)
[fx:dataRequestModifier](#)

Usage Reference

fx:pager

Description

This element is used to render a “pager” which displays next and previous links to allow pagination through items in the list. This FML element implicitly binds itself to the “start-index” parameter in the DAAP query to determine which page of data to be displayed.

Attributes

Name	Required?	Type	Description
Delimiter	No	String	HTML markup to delimit prev and next links
PrevLink	No	String	HTML markup for “prev” link
NextLink	No	String	HTML markup for “next” link
ListID	No	Int	Only necessary when more than 1 <code><fx:grid></code> element is contained within the same FML document. It allows pager to reference the correct grid.

Example

```
<fx:pager delimiter="&nbsp;|&nbsp;" prevlink="Back" nextlink="Forward"/>
```

fx:searchBar

Description

This renders a Search control (text input box and Search button). It implicitly binds itself to the “q” parameter in the DAAPI query to apply a search term filter to the data returned.

Attributes

Name	Required?	Type	Description
MaxLength	No	Int	Specifies maximum input length of search term
ButtonText	Yes	String	Text label for search button
ListID	No	Int	Only necessary when more than 1 <code><fx:grid></code> element is contained within the same FML document. It allows the search bar to reference the correct grid.
ButtonCssClass	No	String	Optional CSS class to apply to search button
InputCssClass	No	String	Optional CSS class to apply to input text box

Example

```
<fx:searchBar MaxLength="35" ButtonText="Search Discussions"/>
```

fx:dataRequestModifier

Description

This renders a dropdown control which can be bound to any DAAPI parameter to allow the DAAPI query to be dynamically manipulated. Examples of its use are to provide Sort or Filter functionality for a content list. The list of available DAAPI parameters which can be controlled can be seen in [Appendix 1](#) within the DAAPIParam enumeration. Note – when providing values for SortOrders, refer to the DAAPI documentation for correct sort order names.

Nested FML Elements

FML Element	Required?	Syntax	Description
fx:listItem	Yes	<code><fx:listItem value="[parameter value]" text="[label]" selected="[boolean]" /></code>	Specifies list of options available. “Value” attribute contains item value. “Text” attribute contains label for item. “Selected” attribute specifies which item is selected by default

Attributes

Name	Required?	Type	Description
Parameter	Yes	DAAPIParam enum	Specifies which DAAPI parameter will be modified to use the selected value in this control.
ListID	No	Int	Only necessary when more than 1 <code><fx:grid></code> element is contained within the same FML document. It allows the search bar to reference the correct grid.

CssClass	No	String	Optional CSS class to apply
----------	----	--------	-----------------------------

Example

```
<fx:dataRequestModifier parameter="SortOrder">
  <fx:listitem value="mostRecentTopicActivity" text="Recent Activity" selected="true" />
  <fx:listitem value="creationDate" text="Date Created" selected="" />
  <fx:listitem value="numberOfComments" text="Number of Comments" selected="" />
  <fx:listitem value="numberOfViews" text="Number of Views" selected="" />
</fx:dataRequestModifier>
```

Data Elements

General

Description	Single Element	List Element
Count of items in the list	N/A	<code><fx:item-count /></code>
Index of the current item being rendered in the list	N/A	<code><fx:item-index /></code>

Content – All Types

Description	Single Element	List Element
UCID of the content	<code><fx:content-ucid /></code>	<code><fx:item-content-ucid /></code>
Content title	<code><fx:content-title maxLength="[int]"/></code>	<code><fx:item-content-title maxLength="[int]"/></code>
Content description	<code><fx:content-description maxLength="[int]"/></code>	
Content permalink	<code><fx:content-permalink /></code>	
Content creator's display name	<code><fx:content-creator-title maxLength="[int]"/></code>	<code><fx:item-content-creator-title maxLength="[int]"/></code>
URL to content creator's profile page	<code><fx:content-creator-profileUrl /></code>	<code><fx:item-content-creator- profileUrl /></code>
Generates HTML markup which contains content creator's thumbnail and title	<code><fx:content-creator-thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]"/></code>	<code><fx:item-content-creator- thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]"/></code>
The URL of the content creator's thumbnail image	<code><fx:content-creator-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]"/></code>	<code><fx:item-content-creator- thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]"/></code>
Date content was created	<code><fx:content-dateCreated format="[dateFormat]"/></code>	<code><fx:item-content-dateCreated format="[dateFormat]"/></code>
The URL (absolute path) to the FLUX content detail page. It can be used for a href attribute of an <a> tag, or <iframe> tag	<code><fx:content-detailUrl /></code>	<code><fx:item-content-detailUrl /></code>
Generates HTML markup which contains content thumbnail and title	<code><fx:content-thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]"/></code>	<code><fx:item-content-thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]"/></code>
The URL of the content thumbnail	<code><fx:content-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]"/></code>	<code><fx:item-content-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]"/></code>
Content's rating	<code><fx:content-rating /></code>	<code><fx:item-content-rating /></code>
Number of comments content has received	<code><fx:content-numberOfComments /></code>	<code><fx:item-content-numberOfComments /></code>
Number of shares for the content	<code><fx:content-numberOfShares /></code>	<code><fx:item-content-numberOfShares /></code>
Number of views for the content	<code><fx:content-numberOfViews /></code>	<code><fx:item-content-numberOfViews /></code>
Displays "Content Action" control. It can be configured to show/hide each item in this control via parameters	<code><fx:content-action showAddToMyProfile="[boolean]" showComments="[boolean]" showVote="[boolean]" showPermalink="[boolean]" showFShare="[boolean]" showFlag="[boolean]" /></code>	<code><fx:item-content-action showAddToMyProfile="[boolean]" showComments="[boolean]" showVote="[boolean]" showPermalink="[boolean]" showFShare="[boolean]" showFlag="[boolean]" /></code>
fShare URL for the content item	<code><fx:content-fshareUrl/></code>	<code><fx:item-content-fshareUrl/></code>

Content - Video

Description	Single Element	List Element
Renders video player. This element is available only for video content.	<code><fx:videoContent-player width="[int]" height="[int]"/></code>	

FML is subject to change and as a result the accuracy of this document is not guaranteed

Content – Text (i.e. Post, News, Discussion Topic)

Description	Single Element	List Element
Displays the text message associated with the content	<pre><fx:textContent-message MaxLength="[int]" Stripping="[HtmlCleanerOptions]"/ ></pre>	<pre><fx:item-textContent-message MaxLength="[int]" Stripping="[HtmlCleanerOptions]"/ ></pre>

Content – Events

Description	Single Element	List Element
Displays details of the event (venue, address, link, other bands)	<code><fx:eventContent-detail name="[eventDetail]" /></code>	<code><fx:item-eventContent-detail name="[venue address link otherbands]" /></code>
Displays the date the event will occur	<code><fx:eventContent-eventdate format="[dateFormat]" /></code>	<code><fx:item-eventContent-eventdate format="[dateFormat]" /></code>

Content – Topics

Description	Single Element	List Element
Date when last comment was created for this topic (or date of topic creation if 0 comments)		<code><fx:item-topicContent-LastPost-dateCreated format="[DateFormat]"/></code>
Display Name of creator of the last comment for this topic (or creator of topic if 0 comments)		<code><fx:item-topicContent-LastPost-creator-title /></code>
Profile URL of creator of the last comment for this topic (or creator of topic if 0 comments)		<code><fx:item-topicContent-LastPost-creator-profileUrl /></code>
If topic is “sticky” then HTML markup within template is rendered		<code><fx:item-topicContent-IsSticky>[HTML template here]</fx:item-topicContent-IsSticky></code>
If category is “locked” then HTML markup within template is rendered		<code><fx:item-topicContent-IsLocked>[HTML template here]</fx:item-topicContent-IsLocked></code>

Content - Albums

Description	Single Element	List Element
Number of content items in the album	<code><fx:albumContent-numberOfElements/></code>	<code><fx:item-albumContent-numberOfElements/></code>

Content – Audio

Description	Single Element	List Element
Running time of the audio track	<code><fx:audioContent-RunningTime /></code>	<code><fx:item-audioContent-RunningTime /></code>
URL to original audio source file	<code><fx:audioContent-sourceUrl /></code>	
Record Label associated with the audio track	<code><fx:audioContent-label /></code>	
Release Date of the audio track	<code><fx:audioContent-releasedDate /></code>	
Artist associated with the audio track	<code><fx:audioContent-artist /></code>	
Producer associated with the audio track	<code><fx:audioContent-producer /></code>	
Album associated with the audio track	<code><fx:audioContent-albumName /></code>	

People – All Types

Description	Single Element	List Element
UCID of the user	<code><fx:user-ucid/></code>	<code><fx:item-user-ucid/></code>
User's display name	<code><fx:user-title/></code>	<code><fx:item-user-title/></code>
User's login name (URL name)	<code><fx:user-name/></code>	<code><fx:item-user-name/></code>
User's first name	<code><fx:user-firstName/></code>	
User's last name	<code><fx:user-lastName/></code>	
User's gender	<code><fx:user-gender/></code>	
User's age	<code><fx:user-age/></code>	
User's birthday in specified format	<code><fx:user-birthday format="[dateFormat]"</code>	
User's city	<code><fx:user-city /></code>	<code><fx:item-user-city /></code>
User's country	<code><fx:user-country/></code>	<code><fx:item-user-country/></code>
User's postal code	<code><fx:user-postalCode/></code>	
User's state code (i.e. CA)	<code><fx:user-stateCode/></code>	<code><fx:item-user-stateCode/></code>
Date user joined community	<code><fx:user-joinedDate/></code>	
User's profile URL	<code><fx:user-profileUrl/></code>	<code><fx:item-user-profileUrl/></code>
Generates HTML markup which contains user's thumbnail and title	<code><fx:user-thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]"/></code>	<code><fx:item-user-thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]"/></code>
The URL of the user's thumbnail image	<code><fx:user-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]"/></code>	<code><fx:item-user-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]"/></code>
About Me text	<code><fx:user-profile-AboutMe/></code>	
Relationship status	<code><fx:user-profile- RelationshipStatus/></code>	
Interested In info	<code><fx:user-profile-InterestedIn/></code>	
Looking For info	<code><fx:user-profile-LookingFor/></code>	
Interests	<code><fx:user-profile-Interests/></code>	
Occupation	<code><fx:user-profile-Occupation/></code>	
Favorite Books	<code><fx:user-profile-FavoriteBooks/></code>	
Favorite Shows or Movies	<code><fx:user-profile- FavoriteShowsOrMovies/></code>	
Favorite Music	<code><fx:user-profile-FavoriteMusic/></code>	
Number of views of this user's profile across the Flux platform	<code><fx:user-numberOfViews /></code>	
Categories user belongs to	<code><fx:user-categories /></code>	
User Type (only applies for "special users")	<code><fx:user-type /></code>	<code><fx:item-user-type /></code>

Content Categories – All Content Types

Description	Single Element	List Element
ID of the category		<code><fx:item-category-id /></code>
Category's title		<code><fx:item-category-title MaxLength="[int]"/></code>
Category's description		<code><fx:item-category-descriptpion MaxLength="[int]"/></code>
Number of content items in the category		<code><fx:item-category- numberOfElements /></code>
The URL of the category's thumbnail image		<code><fx:item-category-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]"/></code>
If category is "official" then HTML markup within template is rendered		<code><fx:item-category- isOfficial>[HTML template here]</fx:item-category- isOfficial ></code>

Content Categories – Topics

Description	Single Element	List Element
Date when last post (topic or comment) was created in the category		<code><fx:item-topicCategory-lastPost-dateCreated format="[date format]"/></code>
Display Name of creator of the last post (topic or comment) in the category		<code><fx:item-topicCategory-lastPost-creator-title /></code>
Profile URL of creator of the last post (topic or comment) in the category		<code><fx:item-topicCategory-lastPost-creator-profileUrl /></code>
Generates HTML markup which contains thumbnail and title of creator of the last post (topic or comment) in the category		<code><fx:item-topicCategory-lastPost-creator-thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]"/></code>
The URL of the thumbnail image of creator of the last post (topic or comment) in the category		<code><fx:item-topicCategory-lastPost-creator-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]"/></code>

User's Communities

Description	Single Element	List Element
UCID of community		<code><fx:item-community-ucid /></code>
Title of community		<code><fx:item-community-title /></code>
Default landing page of community		<code><fx:item-community-landingPageUrl /></code>
Generates HTML markup which contains thumbnail and title of community		<code><fx:item-community-thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]"/></code>
The URL of the thumbnail image of community		<code><fx:item-community-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte] /></code>
Number of members in community		<code><fx:item-community- numberOfMembers /></code>
Date user joined community		<code><fx:item-community-user- JoinedDate /></code>
User's profile URL within community		<code><fx:item-community-user- profileURL /></code>

Comments

Description	Single Element	List Element
Display Name of the user who created the comment		<code><fx:item-comment-creator-title /></code>
Profile URL of the user who created the comment		<code><fx:item-comment-creator-profileUrl /></code>
Generates HTML markup which contains thumbnail and display name of user who created the comment		<code><fx:item-comment-creator-thumbnail mouseOverEffect="[Zoom None]" showTitle="[boolean]" size="[pictureSize]" MaxTitleLength="[int]" /></code>
The Profile URL of the user who created the comment		<code><fx:item-comment-creator-thumbnailUrl size="[pictureSize]" operation="[Crop Resize Matte]" /></code>
Date/Time comment was created		<code><fx:item-comment-dateCreated format="[dateFormat]" /></code>
Comment message body		<code><fx:item-comment-text MaxLength="[int]" /></code>
Displays action bar to allow actions such as "Edit", "Delete", "Flag". These actions will automatically only be available to users with correct permissions.		<code><fx:item-comment-actionBar /></code>

Signup/Login FML Elements

Coming Soon...

Appendix 1 - FML Enumerations

Available Picture Sizes

25x25
50x50
70x70
90x90
120x120
320x320
560x560
70x53
120x90

Date Formats

LongDate	Result format: mm.dd.yyyy hh:mm:ss
ShortDate	Result format: mm.dd.yy
ShortDate2	Result format: mm/dd/yyyy
LongDateTimeAMPM	Result format: mm.dd.yyyy hh:mm AM[PM]
ShortTimeAMPM	Result format: hh:mm AM[PM]
ShortTime	Result format: hh:mm (24 hours)
Elapsed	Result format: n min(s) [hour(s) or day(s)]

DAAPIParam values

SortOrder
StartIndex
MaxResults
ShowOnlyWithPhotos
Category
Editorial

CommunityRole values

Administrator
Contributor
Member
LoggedUser
ContentCreator

EventDetail values

Venue
Address
Link

OtherBands

HTMLCleanerOptions values

PlainText	Strips all HTML/Javascript. Result is plain text only.
MiniRichEdit	Basic HTML tags are allowed (, <p>, <h1> etc) but other tags are stripped
FullRichEdit	Allows <embed> and <object> tags, but strips Javascript and other potentially unsafe HTML
Html	Strips Javascript, but all other HTML is allowed.

BannerPlacementId values

Top728x90
Right160x600
Right300x250
Bottom728x90