

facebook FBML Reference Guide

Consolidated by: yapish.com

Last updated: 03/10/2008

Table of Contents

Table of Contents	2
FBML	5
From Facebook Developers Wiki.....	5
Valid HTML Elements	5
Facebook tags	6
Summary:.....	6
User/Groups.....	6
<i>Fb:name</i>	6
<i>Fb:user</i>	8
<i>Fb:pronoun</i>	9
<i>Fb:profile-pic</i>	10
<i>Fb:grouplink</i>	12
<i>Fb:networklink</i>	13
<i>Fb:is-in-network</i>	14
<i>Fb:if-can-see</i>	15
<i>Fb:if-can-see-photo</i>	16
<i>Fb:if-is-app-user</i>	17
<i>Fb:if-is-friends-with-viewer</i>	18
<i>Fb:if-is-group-member</i>	19
<i>Fb:if-is-user</i>	20
<i>Fb:if-user-has-added-app</i>	21
Deprecated Tags	22
<i>Fb:userlink</i>	22
<i>Fb:if-is-own-profile</i>	23
Profile-specific	24
<i>Fb:wide</i>	24
<i>Fb:narrow</i>	25
<i>Fb:profile-action</i>	26
<i>Fb:user-table</i>	28
<i>Fb:user-item</i>	29
<i>Fb:subtitle</i>	30
Visibility on Profile	33
<i>Fb:visible-to-owner</i>	33
<i>Fb:visible-to-user</i>	34
<i>Fb:visible-to-friends</i>	35
<i>Fb:visible-to-app-users</i>	36
<i>Fb:visible-to-added-app-users</i>	37
<i>Fb:visible-to-connection</i>	38
<i>Fb:18-plus</i>	39
<i>Fb:21-plus</i>	40
Embedded media	41
<i>Fb:iframe</i>	41
<i>Fb:photo</i>	42
<i>Fb:mp3</i>	43
<i>Fb:swf</i>	44

<i>Fb:flv</i>	47
<i>Fb:silverlight</i>	48
Tools	49
<i>Fb:board</i>	49
<i>Fb:comments</i>	51
<i>Fb:friend-selector</i>	53
<i>Fb:google-analytics</i>	55
<i>Fb:if</i>	57
<i>Fb:else</i>	58
<i>Fb:mobile</i>	59
<i>Fb:multi-friend-input</i>	60
<i>Fb:random</i>	62
<i>Fb:random-option</i>	63
<i>Fb:switch</i>	64
<i>Fb:default</i>	65
<i>Fb:user-agent</i>	66
<i>Fb:typeahead-input</i>	68
<i>Fb:typeahead-option</i>	70
Misc	71
<i>Fb:js-string</i>	71
<i>Fb:fbml</i>	73
<i>Fb:fbmlversion</i>	74
<i>Fb:redirect</i>	75
<i>Fb:ref</i>	76
<i>Fb:share-button</i>	78
<i>Fb:time</i>	79
<i>Fb:title</i>	80
<i>Fb:page-admin-edit-header</i>	81
Forms	82
<i>Fb:submit</i>	82
Message/Wall attachments	83
<i>Fb:attachment-preview</i>	83
Notifications and requests	86
<i>Fb:notif-subject</i>	86
<i>Fb:notif-page</i>	87
<i>Fb:notif-email</i>	88
<i>Fb:req-choice</i>	89
<i>Fb:request-form</i>	90
<i>Fb:multi-friend-selector</i>	97
<i>Fb:multi-friend-selector (condensed)</i>	99
<i>Fb:request-form-submit</i>	102
<i>Fb:application-name</i>	103
Status messages	104
<i>Fb:error</i>	104
<i>Fb:explanation</i>	105
<i>Fb:message</i>	107
<i>Fb:editor</i>	108
<i>Fb:editor-button</i>	110

<i>Fb:editor-buttonset</i>	111
<i>Fb:editor-cancel</i>	112
<i>Fb:editor-custom</i>	113
<i>Fb:editor-date</i>	114
<i>Fb:editor-divider</i>	115
<i>Fb:editor-month</i>	116
<i>Fb:editor-text</i>	117
<i>Fb:editor-textarea</i>	118
<i>Fb:editor-time</i>	119
Page navigation	120
<i>Fb:dashboard</i>	120
<i>Fb:action</i>	122
<i>Fb:create-button</i>	124
<i>Fb:help</i>	125
<i>Fb:header</i>	126
<i>Fb:mediaheader</i>	128
<i>Fb:header-title</i>	129
<i>Fb:owner-action</i>	130
<i>Fb:tabs</i>	131
<i>Fb:tab-item</i>	132
Dialog	134
<i>Fb:dialog</i>	134
<i>Fb:dialog-title</i>	136
<i>Fb:dialog-content</i>	137
<i>Fb:dialog-button</i>	138
Wall	139
<i>Fb:wall</i>	139
<i>Fb:wallpost</i>	140
<i>Fb:wallpost-action</i>	141
Dynamic FBML attributes	142
Summary	142
Details	142
<i>DynamicFBML/Visibility</i>	142
<i>DynamicFBML/MockAJAX</i>	144
<i>DynamicFBML/Forms</i>	145
<i>Requirelogin</i>	146
<i>FBJS</i>	147
Usage notes	156
Invalid CSS attributes	157
See also	158
External links	159

FBML

From Facebook Developers Wiki

Jump to: [navigation](#), [search](#)

Facebook Markup Language (FBML) enables you to build full Facebook Platform applications that deeply integrate into a user's Facebook experience. You can hook into several Facebook integration points, including the profile, profile actions, Facebook canvas, News Feed and Mini-Feed.

FBML is an evolved subset of [HTML](#) with some elements removed, and others which have been added that are specific to Facebook. You set the FBML for a profile box by calling [profile.setFBML](#) through the [API](#). The FBML is cached on Facebook's server until [profile.setFBML](#) is called again through a canvas page. For a high-level technical spec for FBML, please see [FBMLspec](#).

Valid HTML Elements

a	fieldset	ol	textarea
abbr	font	optgroup	tfoot
acronym	form	option	th
address	h1	p	thead
b	h2	pre	tr
bdo	h3	q	tt
big	h4	s	u
blockquote	h5	samp	ul
br	h6	script	var
caption	hr	select	
center	i	small	
cite	img	span	
code	input	strike	
dd	ins	strong	
del	kbd	style	
dfn	label	sub	
div	legend	sup	
dl	li	table	
dt	link	tbody	
em	meta	td	

Facebook tags

Summary:

- User/Groups
- Deprecated Tags
- Profile-specific
- Visibility on Profile
- Embedded media
- Tools
- Misc
- Forms
- Message/Wall attachments
- Notifications and requests
- Status messages
- Editor display
- Page navigation
- Dialog
- Wall

User/Groups

Fb:name

Description

Renders the name of the user specified, optionally linked to his or her profile.

This also works for Facebook Pages with the ID of the Page passed as the uid parameter.

You can use this tag for both the subject and the object of a sentence describing an action. For example, if a user with the user ID \$stagger tags a photo of a user with the user ID \$tagee, you could say:

```
<fb:name uid="$stagger" capitalize="true" /> tagged a photo of <fb:name subjectid="$stagger" uid="$tagee" />
```

Attributes

Required	Name	Type	Description
required	uid	uid	The ID of the user or Page whose name you want to show. You can also use " loggedinuser " or " profileowner ".
optional	firstnameonly	bool	Show only the user's first name. <i>(default value is false)</i>
	linked	bool	Link to the user's profile. <i>(default value is true)</i>
	lastnameonly	bool	Show only the user's last name. <i>(default value is false)</i>

possessive	bool	Make the user's name possessive (e.g. Joe's instead of Joe). <i>(default value is false)</i>
reflexive	bool	Use "yourself" if useyou is true. <i>(default value is false)</i>
shownetwork	bool	Displays the primary network for the uid. <i>(default value is false)</i>
useyou	bool	Use "you" if uid matches the logged in user . <i>(default value is true)</i>
ifcantsee	string	Alternate text to display if the logged in user cannot access the user specified. <i>(default value is [empty string])</i>
capitalize	bool	Capitalize the text if useyou==true and loggedinuser ==uid. <i>(default value is false)</i>
subjectid	uid	The Facebook ID of the subject of the sentence where this name is the object of the verb of the sentence. Will use the reflexive when appropriate. When <i>subjectid</i> is used, <i>uid</i> is considered to be the object and <i>uid</i> 's name is produced.

Examples

```
<fb:name uid="12345" />
```

When logged in as user 1160:

```
<fb:name subjectid="219770" uid="219770" /> = himself
```

```
<fb:name subjectid="219770" uid="1160" /> = you
```

```
<fb:name subjectid="1160" uid="219770" /> = Josh Gibson
```

```
<fb:name subjectid="1160" uid="1160" /> = yourself
```

```
<fb:name uid="profileowner" useyou="true" possessive="true" reflexive="true" /> = your own
```

```
<fb:name uid="1160" capitalize="true" /> tagged a photo of <fb:name subjectid="1160" uid="219770" />
```

See Also

You can see a real world example of this tag in all three demo applications (Footprints, Restaurants and Who's Showing Up). Go to Demos for more details and to download and install the demo apps.

Fb:user

Description

Forces the enclosed content to be restrained to site privacy checks[verify] based on the viewer of the information. The content within the tag "belongs" to the user specified.

Note: This tag does not appear to work at all when I test it, it does nothing. See bug report number 404.

Attributes

Required	Name	Type	Description
required	uid	int	The user ID of the user whose information will be contained in the tag.

Examples

```
<fb:user uid="12345">
```

This is user 12345's information. It should only be shown to people who can see user 12345.

Fb:pronoun

Description

Renders a pronoun for a specific user.

Attributes

Required	Name	Type	Description
required	uid	int	The user ID for whom to generate the pronoun. You can substitute actor for the user ID so you can more easily aggregate feed stories.
optional	useyou	bool	Use the word "you" if uid is viewing the page. <i>(default value is true)</i>
	possessive	bool	Use the possessive form (his/her/your/their). <i>(default value is false)</i>
	reflexive	bool	Use the reflexive form (himself/herself/yourself/themselves). <i>(default value is false)</i>
	objective	bool	Use the objective form (him/her/you/them). <i>(default value is false)</i>
	usethey	bool	Use "they" if gender is not specified. <i>(default value is true)</i>
	capitalize	bool	Force a capital letter for the pronoun. <i>(default value is false)</i>

Examples

```
<fb:pronoun uid="12345" />
```

```
<fb:pronoun uid="actor" possessive="true" useyou="false"/>
```

Fb:profile-pic

Description

Turns into an img tag for the specified user's or Facebook Page's profile picture. The tag itself is treated like a standard img tag, so attributes valid for img are valid with fb:profile-pic as well.

Attributes

Required	Name	Type	Description
required	uid	int	The user ID of the profile or Facebook Page for the picture you want to display.
optional	size	string	The size of the image to display. (<i>Default value is thumb</i>). Other valid values are thumb (t) (50px wide), small (s) (100px wide), normal (n) (200px wide), and square (q) (50px by 50px).
	linked	bool	Make the image a link to the user's profile. (<i>default value is true</i>)

Examples

```
<fb:profile-pic uid="12345" linked="true" />
```

```
<fb:profile-pic uid="12345" size="square" linked="true" />
```

See Also

You can see a real world example of this tag in all three demo applications (Footprints, Restaurants and Who's Showing Up). Go to Demos for more details and to download and install the demo apps.

Fb:eventlink

Description

Prints the specified event name and formats it as a link to the event's page.

Attributes

Required	Name	Type	Description
required	eid	int	Event ID for the event whose name and link you want to retrieve.

Examples

```
<fb:eventlink eid="21150032416" />
```

Fb:grouplink

Description

Prints the specified group name and formats it as a link to the group's page.

Attributes

Required	Name	Type	Description
required	gid	int	Group ID for the group whose name and link you want to retrieve.

Examples

```
<fb:grouplink gid="2541896821" />
```

Fb:networklink

Description

Prints the specified network name and formats it as a link to the network's page.

Attributes

Required	Name	Type	Description
required	nid	int	Network ID for the network whose name and link you want to retrieve.

Examples

```
<fb:networklink nid="67108896" />
```

Fb:is-in-network

Description

Displays content inside the tag only if the user is in a given network.

Note: You can use fb:else with fb:is-in-network, even though the tag does not start with fb:if-.

Attributes

Required	Name	Type	Description
required	network	int	The network ID to check. You can check one network at a time.
optional	uid	int	The user ID to check. (default value is loggedinuser)

Examples

```
<fb:is-in-network network="16777229" uid="1230541">User 1230541, you are in the <fb:networklink  
nid="16777229" /> network!</fb:is-in-network>
```

Notes

This tag renders content on canvas pages and for notifications and feed stories. It doesn't work with profiles, email and notifications.

You can use the fb:networklink FBML to display the name/link of the network.

Fb:if-can-see

Description

Displays the enclosed content if the logged in user can see the specified what attribute of the specified user.

You can use these settings to provide content or links with relevance to Facebook privacy or to implement your own privacy using Facebook's current controls.

You can use this tag in conjunction with fb:else.

Attributes

Required	Name	Type	Description
required	uid	int	The user ID to check.
optional	what	string	The privacy setting to check. (<i>Default value is search</i>). Other valid values are profile, friends, not_limited, online, statusupdates, wall, groups, courses, photosofme, notes, feed, contact, email, aim, cell, phone, mailbox, address, basic, education, professional, personal, seasonal.

Examples

```
<fb:if-can-see uid="12345" what="profile">  
  Check out how cool <fb:name uid="12345" /> is.  
  <fb:else>Sorry, no dice.</fb:else>  
</fb:if-can-see>
```

Notes

As of FBML v1.1 you can use this tag only on a canvas page.

See Also

You can see a real world example of this tag in the Footprints demo application. Go to Demos for more details and to download and install the demo apps.

Fb:if-can-see-photo

Description

Displays the enclosed content only if the logged in user can see the photo specified.

You should use this tag^[who says?] when specifying content that should only be shown when using fb:photo.

You can use this tag in conjunction with fb:else.

Attributes

Required	Name	Type	Description
required	pid	int	An API-supplied pid of the photo, or any pid found in the query string of a photo URL on Facebook.
optional	uid	int	If pid is not an API-supplied pid, this should be the id parameter in the query string used to find the pid.

Examples

```
<fb:if-can-see-photo pid="12345">  
  <fb:photo pid="12345" />  
  Check out this cool photo!  
</fb:if-can-see-photo>
```

```
<fb:if-can-see-photo pid="543212" uid="6789">  
  <fb:photo pid="54321" uid="6789" />  
  This one is cool too!  
</fb:if-can-see-photo>
```

Notes

As of FBML v1.1 you can use this tag only on a canvas page.

Fb:if-is-app-user

Description

Displays the enclosed content only if the specified user has accepted the terms of service of the application. Use `<fb:if-user-has-added-app>` to determine whether the user has added the application to their account.

Attributes

Required	Name	Type	Description
optional	uid	int	The user ID to check. (default value is loggedinuser)

Examples

```
<fb:if-is-app-user uid="12345">
  Here's the user's score: 55555!
</fb:if-is-app-user>
```

```
<fb:if-is-app-user>
  Your score: 55555!
</fb:if-is-app-user>
```

Notes

As of FBML v1.1 you can use this tag only on a canvas page.

Fb:if-is-friends-with-viewer

Description

Displays the enclosed content only if the specified user is friends with the logged in user.

Attributes

Required	Name	Type	Description
optional	uid	int	The user ID to check. <i>(default value is profile owner^{verify})</i>
	includeself	bool	Return true if viewer and uid specified are the same. <i>(default value is true)</i>

Examples

```
<fb:if-is-friends-with-viewer uid="12345">  
  Hey you guys are friends!  
<fb:else>Did you know that you have something in common?</fb:else>  
</fb:if-is-friends-with-viewer>
```

Notes

As of FBML v1.1 you can use this tag only on a canvas page.

Fb:if-is-group-member

Description

Displays the enclosed content only if the specified user is a member of the specified group.

Attributes

Required	Name	Type	Description
required	gid	int	The group ID.
optional	uid	int	The user ID to check. (default value is loggedinuser)
	role	string	The role to check. (Default value is <i>member</i>). Other valid values are officer, admin.

Examples

```
<fb:if-is-group-member gid="54321" uid="12345" role="admin">  
  Hey, you're an admin of that group!  
<fb:else>Sorry, admins only!</fb:else>  
</fb:if-is-group-member>
```

Notes

As of FBML v1.1 you can use this tag only on a canvas page.

Fb:if-is-user

Description

Only renders the content inside the tag if the viewer is one of the specified user(s).

Attributes

Required	Name	Type	Description
required	uid	int	The user ID of the user that is allowed to see the content. To match multiple users, pass in a comma-delimited list of uids.

Examples

```
<fb:if-is-user uid="uid1,uid2">This is secret 4 u 2!</fb:if-is-user>
```

Notes

As of FBML v1.1 you can use this tag only on a canvas page.

Fb:if-user-has-added-app

Description

Displays the enclosed content only if the specified user has added the application to their account.

Attributes

Required	Name	Type	Description
optional	uid	int	The user ID to check. (default value is loggedinuser)

Examples

```
<fb:if-user-has-added-app uid="12345">  
  This user has put this app into his account.  
<fb:else>That user hasn't put this app into his account.</fb:else>  
</fb:if-user-has-added-app>
```

```
<fb:if-user-has-added-app>  
  Hey, welcome!  
<fb:else>You must add this to your account to use it</fb:else>  
</fb:if-user-has-added-app>
```

Notes

As of FBML v1.1 you can use this tag only on a canvas page.

This tag seems to work inside `<fb:subtitle>` on a profile page.

Deprecated Tags

Fb:userlink

Description

Prints the specified user's full name linked to their profile along with their network, optionally (as is normally displayed on the Wall, for example).

This tag is **deprecated**. It is still useful in one context, however: using it in `feed.publishActionOfUser` (with the user ID on whose behalf the action is being published) prevents the Mini-Feed from prepending the actor's name to your feed title.

Attributes

Required	Name	Type	Description
required	uid	int	The User ID for the user whose name and link you want to retrieve.
optional	shownetwork	bool	Indicates whether to display the network name. <i>(default value is true)</i>
	ifcantsee	string	The alternate text to display if the user cannot see the specified user ID.

Examples

```
<fb:userlink uid="123445" />
```

See Also

You can see a real world example of this tag in the Footprints and Restaurants demo applications. Go to Demos for more details and to download and install the demo apps.

fb:name

Fb:if-is-own-profile

Description

This tag is deprecated, since if- tags are no longer allowed on profile pages.

It was used to display content inside only if the viewer of the profile matches the profile owner.[verify]

See also/instead fb:visible-to-user

Attributes

Examples

```
<fb:if-is-own-profile>  
  Welcome back to your profile!  
  <fb:else>This is not your profile.</fb:else>  
</fb:if-is-own-profile>
```

Notes

As of FBML v1.1 you can use this tag only on a canvas page.
error=Not allowed in profile. How to replace this?

See Also

You can see a real world example of this tag in the Who's Showing Up demo application. Go to Demos for more details and to download and install the demo apps.

Profile-specific

Fb:wide

Description

The enclosed content appears only when profile box is in the wide column of the profile. See fb:narrow for the opposite.

Attributes

Examples

```
<fb:wide>  
This will only appear in the wide column.  
</fb:wide>
```

Notes

If no fb:narrow or fb:wide tag is specified, all content appears in either column.

The wide profile box is 388 pixels wide, not including margins. The left margin is 8 pixels, but there is no right margin. For balanced margins, make your content 380 pixels wide.

See Also

You can see a real world example of this tag in the Footprints demo application. Go to Demos for more details and to download and install the demo apps.

Fb:narrow

Description

Renders the content contained by the tag only if the profile box is in the narrow column of the profile.

Attributes

Examples

```
<fb:narrow>  
This text appears only in the narrow column.  
</fb:narrow>
```

Notes

If no **fb:narrow** or [fb:wide](#) tag is specified, all content is displayed in either column.

The narrow profile box is 190 pixels wide, not including margins. The left margin is 10 pixels, but there is no right margin. For balanced margins, make your content 180 pixels wide.

If you want your **fb:narrow** information to appear when installed, please remember to set the **Default Profile Box Column** value to "Narrow" in *Developer Application » My Apps » Application's Settings » Installation section*.

Fb:profile-action

Description

Renders a link on the user's profile under their photo (such as "View More photos of..").

Attributes

Required	Name	Type	Description
required	url	string	The URL to which the user is taken after clicking. <i>Must be an absolute link.</i>

Examples

```
<fb:profile-action url="http://www.mysite.com/action/">
  Perform Action
</fb:profile-action>
```

NOTE: 'if-is-own-profile' deprecated! See that function for details

```
<fb:if-is-own-profile>
  <fb:profile-action url="http://apps.facebook.com/myapplication/status/">
    View Your Status
  </fb:profile-action>
<fb:else>
  <fb:if-is-app-user uid="profileowner">
    <fb:profile-action url="http://apps.facebook.com/myapplication/status/">
      View this person's status
    </fb:profile-action>
  <fb:else>
    <fb:profile-action url="http://apps.facebook.com/myapplication/invite/">
      Invite this person to MyApplication
    </fb:profile-action>
  </fb:else>
</fb:if-is-app-user>
</fb:else>
</fb:if-is-own-profile>
```

NOTE: example with fb:visible-to-user instead of the deprecated fb:if-is-own-profile

```
<fb:visible-to-user uid="loggedinuser">
  <fb:profile-action url="http://apps.facebook.com/application/my_page">
    View Your Application
  </fb:profile-action>
<fb:else>
  <fb:profile-action url="http://apps.facebook.com/application/">
```

```

        Check Out Application
    </fb:profile-action>
  </fb:else>
</fb:visible-to-user>

```

Notes

Use [profile.setFBML](#) to add the button to a user's profile.

If you add this to the profile of a user who has not added your application, it will appear only to those users who have added your application. By setting the default FBML (in Edit Settings, under [My Applications](#)) to include an fb:profile-action, this action appears on every profile the user of your application sees.

There is a limit of about 30 characters in the action label. (**Note:** Leading and trailing white-space counts against this limit, so if your link is being elided, check for and remove any extra spaces between the tags and the label text.)

For users for whom you have not called [profile.setFBML](#), the actions are read from the content in "Default FBML" section of your application settings. For the most part, this applies to any user who has not added your application.

For users for whom you have called [profile.setFBML](#), the actions are read from whatever content you set when you last called [profile.setFBML](#) for that user.

Example: There are three friends -- Larry, Curly and Moe. Both Larry and Curly add your application. When Larry views Curly's profile, the profile actions that Larry sees on Curly's profile come from the content you posted to Curly's profile. When Larry views Moe's profile, the profile actions Larry sees on Moe's profile come from the default FBML content, because you haven't called [profile.setFBML](#) on Moe, because you don't necessarily know he exists

See Also

You can see a real world example of this tag in the Footprints demo application. Go to [Demos](#) for more details and to download and install the demo apps.

[Application Visibility in Profiles](#)

Errors

fb:profile-action: Required fbml_env var missing: "profile"

- If you get this error, it means you tried to use the [fb:visible-to-owner](#), [fb:visible-to-app-users](#), and the other fb:visible-to-*** elements on a Canvas page.

fb:if-is-own-profile

- Not allowed in profile. Use [fb:visible-to-owner](#), [fb:visible-to-app-users](#), and the other fb:visible-to-*** elements instead.

running example above through the [Facebook FBML validator] returns

- RUNTIME ERROR: fb:if-is-own-profile: "ifs" are not allowed in flavor ProfileBoxFBMLFlavor --[719255867](#) 11:05, 19 November 2007 (PST)

Most punctuation is stripped from the profiles box.

Can someone write a working example that reproduces this behavior WITHOUT if-tags?

Fb:user-table

Description

Renders a table, each cell of which contains a thumbnail and name for a particular user, similar to the Mutual Friends table on profile pages. Inside this tag, use [fb:user-item](#) tags to specify the set of users. This tag only works on profile pages. Applications **cannot** use this on their canvas pages.

Attributes

Required	Name	Type	Description
optional	cols	int	The number of columns in the table. (default value is 6 for fb:wide , 3 for fb:narrow)

Examples

```
<fb:user-table cols="3">
  <fb:user-item uid="12345" />
  <fb:user-item uid="123456" />
  <fb:user-item uid="123457" />
  <fb:user-item uid="123458" />
  <fb:user-item uid="123459" />
  <fb:user-item uid="123450" />
</fb:user-table>
```

Fb:user-item

Description

Renders a single cell of a table, which contains a thumbnail and name for a particular user, similar to the Mutual Friends table on profile pages. Must be used inside a [fb:user-table](#) tag. This tag only works on profile pages. Applications **cannot** use this on their canvas pages.

Attributes

Required	Name	Type	Description
required	uid	uid	The ID of the user whose name and photo you want to show.

Examples

```
<fb:user-table cols="3">
  <fb:user-item uid="12345" />
  <fb:user-item uid="123456" />
  <fb:user-item uid="123457" />
  <fb:user-item uid="123458" />
  <fb:user-item uid="123459" />
  <fb:user-item uid="123450" />
</fb:user-table>
```

Fb:subtitle

Description

Defines the subtitle for the profile box. The enclosed content is interpreted purely as plain text. This tag may contain [fb:action](#), an optional tag that renders an action link on the right-hand side of the subtitle.

Attributes

Required	Name	Type	Description
optional	seeallurl	string	The URL for a "See all" link. This URL must be a canvas page.

Examples

```
<fb:subtitle seeallurl="http://apps.facebook.com/myapp/seeall" >  
  <fb:action href="http://apps.facebook.com/wall/walltowall.php">Wall-to-wall</fb:action>  
  Displaying 10 wall posts  
</fb:subtitle>
```

Notes

This tag always renders the HTML title attribute with *Learn More About This Application* for seeallurl. [\[possible bug?\]](#) At present there doesn't seem to be any way to prevent this.

This tag does not function within fb:visible-to- tags. [\[possible bug?\]](#)

See Also

[fb:title](#)

Fb:action

Description

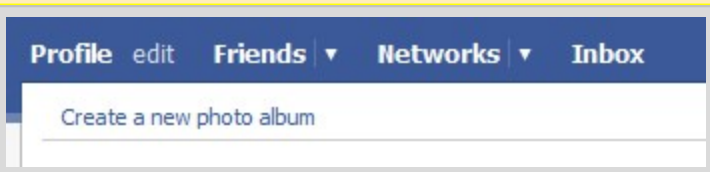
Renders a link, usually for navigational purposes. Its appearance depends on its container. The tag **must** be a child of either [fb:dashboard](#) or [fb:subtitle](#).

Attributes

Required	Name	Type	Description
required	href	string	The URL for the link. The URL must be a canvas page. For example, href="http://apps.facebook.com/<appname>/<filename>.php".
optional	title	string	Specifies the text to display as a tool tip for this link. (<i>seems to do nothing at the moment</i>)
	onclick	string	Call a FBJS function

Examples

```
fb:dashboard>
  <fb:action href="new.php">Create a new photo album</fb:action>
</fb:dashboard>
```



```
<fb:dashboard>
  <fb:action href="new.php">Create a new photo album</fb:action>
  <fb:action href="you.php">Photos of You</fb:action>
</fb:dashboard>
```



This article or section needs expansion

Notes

The link appears in the top left of the fb:dashboard element and the top right of the fb:subtitle element. ~~You cannot use FBJS onclick, since you cannot have the href attribute set to # or JavaScript.~~ You *can* use the onclick attribute: use any absolute URL in the href and make sure onclick returns false to prevent the link from being followed.

See Also

You can see a real world example of this tag in the Who's Showing Up demo application. Go to [Demos](#) for more details and to download and install the demo apps.

Visibility on Profile

Fb:visible-to-owner

Description

Displays content inside only if the viewer of the profile matches the profile owner. This also works for Facebook Pages with the content only displayed to an admin of the Page.

Attributes

Required	Name	Type	Description
	bgcolor	color	The color of the blank box shown to users who are not the owner.

Examples

```
<fb:visible-to-owner>
  Welcome back to your profile!
</fb:visible-to-owner>
```

Notes

This is new to [FBML v1.1](#), make sure you wrap your code in `<fb:fbml version="1.1">` and `</fb:fbml>`!. **Do not use this tag to display private or sensitive information.** Content inside this tag is rendered to all users' browsers, including those who are not the profile owner. For those who are not the owner, the content is shown as white space on the page but it is still visible by viewing the page source. This tag cannot be used to for controlled display elements like fb:subtitle

Fb:visible-to-user

Description

Only displays the content inside the tag if the viewer is the specified user.^[verify]

This also works for Facebook Pages with the content only displayed to an admin of the Page and the viewer is not on a Page profile.

Attributes

Required	Name	Type	Description
required	uid	int	The uid of the user that is allowed to see the content. ^[verify]
	bgcolor	color	The color of the blank box shown to users who are not the owner or the user specified.

Examples

```
<fb:visible-to-user uid="12345">This is secret 4 u 2!</fb:visible-to-user>
```

Notes

This is new to [FBML v1.1](#).

This does not appear to accept a comma-delimited list of user ids like [fb:if-is-user](#) does. [Request for such.](#)

Do not use this tag to display private or sensitive information. Content inside this tag is rendered to all users' browsers, including those who are not one of the specified users. For those who are not one of those users, the content is shown as white space on the page but it is still visible by viewing the page source.

Content in this tag is also visible to the owner.

Fb:visible-to-friends

Description

Use this tag to display the content inside the tag on a user's profile only if the viewer is a friend of that user. While this tag is still valid, you may find it more versatile to use [fb:visible-to-connection](#), since it applies equally to user profiles and Facebook Pages.

Attributes

Required	Name	Type	Description
optional	bgcolor	color	The color of the blank box shown to viewers who are not friends of the user.

Examples

```
<fb:visible-to-friends uid="12345">Hello my friend!</fb:visible-to-friends>
```

Notes

This is new to [FBML v1.1](#).

Do not use this tag to display private or sensitive information. Content inside this tag is rendered to all users' browsers, including those who are not one of the specified users. For those who are not one of those users, the content is shown as white space on the page but it is still visible by viewing the page source.

Content in this tag is also visible to the owner.

Fb:visible-to-app-users

Description

Displays the enclosed content only if the viewer has granted full permissions to the application. [\[verify\]](#)

Attributes

Required	Name	Type	Description
	bgcolor	color	The color of the blank box shown to users who do not own the app.

Examples

```
<fb:visible-to-app-users>  
  Welcome users!  
</fb:visible-to-app-users>
```

Notes

This is new to [FBML v1.1](#).

Do not use this tag to display private or sensitive information. Content inside this tag is rendered to all users' browsers, including those who have not granted full permissions to the application. For those who have not done so, the content is shown as white space on the page but it is still visible by viewing the page source.

Fb:visible-to-added-app-users

Description

Displays the enclosed content only if the viewer has added the application to their account. [\[verify\]](#)

Attributes

Required	Name	Type	Description
	bgcolor	color	The color of the blank box shown to users who are not the owner or have added the app.

Examples

```
<fb:visible-to-added-app-users>  
Hey, welcome! Thanks for adding my app!  
</fb:visible-to-added-app-users>
```

Notes

This is new to [FBML v1.1](#).

Do not use this tag to display private or sensitive information. Content inside this tag is rendered to all users' browsers, including those who have not added the app to their account. For those who have not added the app, the content is shown as white space on the page but it is still visible by viewing the page source.

Fb:visible-to-connection

Description

Use this tag to display the content inside the tag on a user's or a [Facebook Page's](#) profile only if the viewer is a friend of that user or is a fan of that Facebook Page.

Attributes

Required	Name	Type	Description
optional	bgcolor	color	The color of the blank box shown to viewers who are not friends of the user or fans of the Page.

Examples

```
<fb:visible-to-connection uid="12345">Welcome, fans!</fb:visible-to-connection>
```

Notes

This is new to [FBML v1.1](#).

Do not use this tag to display private or sensitive information. Content inside this tag is rendered to all users' browsers, including those who are not one of the specified users. For those who are not one of those users, the content is shown as white space on the page but it is still visible by viewing the page source.

Content in this tag is also visible to the owner.

Fb:18-plus

Description

Restricts content to users who are age 18 or older.

Examples

```
<fb:18-plus>Oooh la la<fb:else>Barney?</fb:18-plus>
```

Fb:21-plus

Description

Restricts content to users who are age 21 or older.

Examples

```
<fb:21-plus>Oooh la la<fb:else>Barney?</fb:21-plus>
```

Embedded media

Fb:iframe

Description

Inserts an <iframe> tag into the page.

Attributes

Required	Name	Type	Description
required	src	string	The URL of the iframe. Signed GET parameters are appended to the URL to prove that the frame was loaded through Facebook, as described in the forms section. These parameters also include one named fb_sig_in_iframe to indicate this context.
optional	smartsize	bool	This parameter smartly sizes the iframe to fit the remaining space on the page and disables the outer scrollbars. If you include more than one smartizing iframe, they automatically distribute the size appropriately. <i>(default value is false)</i>
	frameborder	int	Indicates whether to show (1) or hide (0) an iframe border. <i>(default value is 1)</i>
	scrolling	string	Indicates whether to show scrollbars. <i>(default value is yes)</i> - use "yes", "no", or "auto" (not "true" or "false")
	style	string	Indicates a custom inline style for the iframe.
	width	int	Indicates the width of the IFRAME.
	height	int	Indicates the height of the IFRAME.

Notes

You cannot use the <fb:iframe> code on the profile page.

You must use your server's URL as the src for your iframe. Otherwise, apps.facebook.com wraps your page with the Facebook layout.

You cannot use FBML inside an iframe.

Fb:photo

Description

Renders a Facebook photo.

Attributes

Required	Name	Type	Description
required	pid	int	An API-supplied pid of the photo, or any pid found in the query string of a photo URL on Facebook.
optional	uid	int	If the pid is not an API-supplied pid, this should be the ID parameter in the query string used to find the pid.
	size	string	The size of the photo to display. (<i>default value is normal</i>). Other valid values are thumb (t), small (s), normal (n), and square (q).
	align	string	The image's alignment. (<i>default value is left</i>) and the only other valid value is right.

Examples

```
<fb:photo pid="12345" />
```

```
<fb:photo pid="54321" uid="6789" />
```

Notes

The pid returned after making an FQL call is not the same pid found in the string of a photo URL. In order to use the API-supplied pid, do **not** include the uid.

Any content that belongs with the photo should be wrapped^[who says?] by [fb:if-can-see-photo](#).

Fb:mp3

Description

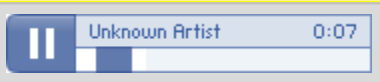
Renders a flash-based audio player.

Attributes

Required	Name	Type	Description
required	src	string	The URL of the audio file. The URL must be absolute.
optional	title	string	The name of the song.
	artist	string	The name of the artist performing the song.
	album	string	The title of the album.
	width	int	The width of the player in pixels. <i>(default value is 300)</i>
	height	int	The height of the player in pixels. <i>(default value is 29)</i>

Examples

```
<fb:mp3 src="http://host.com/file.mp3" title="Song Name" artist="Song Artist" />
```



Notes

The src attribute must point directly to the mp3 file, [\[possible bug?\]\[who says?\]](#) and cannot be played by a query string (for example, "songs.php?songid=35911").

Supported formats include .mp3.

.wav files are **not** supported [\[possible bug?\]](#).

File's bitrate must be in increments of 11KHz (11KHz, 22KHz, 44.1 KHz all work).

File must end in .mp3 (for example, /location/1385097 will not work).

Song name, artist, and album must be in ASCII characters or does not display correctly. i.e. Song names in Chinese do not display in Chinese but in gibberish or not at all [\[possible bug?\]](#).

See Also

[fb:flv](#)

[fb:swf](#)

Fb:swf

Description

Renders a Shockwave Flash (SWF) object. On profile pages, an image appears first. When the user clicks the image, it turns into the Flash object. On canvas pages, the image is ignored, and the Flash object is directly included.

Attributes

Required	Name	Type	Description
required	swfsrc	string	The URL of the Flash object. The URL must be absolute.
optional	imgsrc	string	The URL of the image (.gif and .jpg formats only). <i>(default value is [];</i> <i>Note that this renders the Flash object unusable and invisible on profile pages.)</i>
	height	int	The height of the image and the Flash object.
	width	int	The width of the image and the Flash object.
	imgstyle	string	The style attribute for the image.
	imgclass	string	The class attribute for the image.
	flashvars	string	The URL-encoded Flash variables. Also passes the fb_sig_ values as described in the section on Forms .
	swfbgcolor	string	The hex-encoded background color for the Flash object.
	waitforclick	bool	Indicates whether to autoplay the Flash object (false) when allowed. false does not work in profiles for security and aesthetic reasons, except after an AJAX call. <i>(default value is true)</i>
	salgin	string	The salgin attribute; this is an <embed> parameter.
	loop	string	Indicates whether to play the Flash object continuously.
	quality	string	Indicates the quality of the object. Specify high, medium or low.
	scale	string	The scaling to apply to the object.
	align	string	Indicates how the browser aligns the object. Specify left, center or right
	wmode	string	Indicates the opacity setting for the object. Specify transparent, opaque or window. <i>(default value is transparent)</i>

Flash Variables

Facebook passes the following parameters to the SWF when it is loaded:

Required	Name	Type	Description
	allowScriptAccess	string	This string is always set to "never."

fb_sig_profile	int	The uid of the user into whose profile the Flash object is being loaded; this is blank when loaded in the canvas .
fb_sig_time	int	The time when the signature was generated.
fb_sig_user	int	The uid of the currently logged in user.
fb_sig_session_key	string	The Facebook session key.
fb_sig_expires	int	The session expiration time.
fb_sig_api_key	string	Your application's API key.
fb_sig_added	bool	Indicates whether the user has added your application.
fb_sig	string	An MD5 hash of all the parameters with names that start with fb_sig_ plus your application secret. This way, the Flash object can confirm that it is being loaded into a Facebook page. (But be careful embedding your secret in your Flash application; hackers can get at it if you do.) See notes below.

Examples

```
<fb:swf
  swfbgcolor="000000"
  imgstyle="border-width:3px; border-color:white;"
  swfsrc='http://www.youtube.com/v/<c:out value="{nutshotData.episodeNumber}"/>'
  imgsrc='http://img.youtube.com/vi/{param.nutshot}/2.jpg'
  width='340' height='270' />
```

You can see it in action at [The nutshOt network](#)

The screenshot shows the 'nutshOt network' website. At the top left is the logo 'nutshOt network' with the tagline 'nuttin' but nutshOts!'. Below the logo are buttons for 'nutshOt a friend', 'rate this nutshOt', 'submit a nutshOt', and 'VIP Section'. The main content area features a large video player titled 'Wallclimber nutshOt' showing a person climbing a wall. Below the video is a rating system with five stars and an 'Average Rating: 4.0' label. To the left of the video are four smaller video thumbnails with titles and descriptions: 'Nine Iron nutshOt', 'Dodgeball nutshOt', 'Holeman nutshOt', and 'Soup Can nutshOt'. At the bottom, there is a 'more nutshOts >>' button and a row of four more video thumbnails.

Notes

Currently, Facebook requires Flash version 9.0.0 for all `<fb:swf>` tags.

Make sure that the `flashvars` parameter is all lowercase. Some sources of embedded links capitalize the `V`, which does not work on Facebook.

Currently, Facebook wraps the resultant Flash object in a `<div>` tag, so despite the fact that the `embed/object` tag is not block-level, consecutive `<fb:swf>` tags will appear one above the other instead of side-by-side.

To verify that your Flash object was loaded from a Facebook page, do the following. For security, this technique does not embed your secret key in your Flash app:

Get all the parameters whose names start with `fb_sig_`. (Do not include the `fb_sig` parameter itself.) In Flex use `Application.application.parameters` to do this.

Strip the `fb_sig_` prefix from each, and make sure the keys are lowercase.

Create a string of the form `param1=value1param2=value2param3=value3`, etc., sorted by the names (not the values) of the parameters. **Note:** Do not use ampersands between the parameters.

Pass this string to your server, where your secret key is stored.

On your server, append your application secret key to the string that was passed in. The following is returned:

```
param1=value1param2=value2param3=value3myappsecret
```

On your server, create an MD5 hash of this string.

Return the MD5 hash from your server to your Flash object.

In your Flash object, compare the returned hash with the `fb_sig` parameter that was passed in. If they are equal, then your Flash object was loaded by Facebook. (Or by someone who stole your secret key.)

While the above technique doesn't embed your secret in the Flash object, what you're doing is making a public Web service to sign parameter strings with your secret and then embedding its address in your Flash object. This is just as bad as publishing your secret key (except you do the MD5 computation for any malicious clients).

What you want to do is send all the parameters to the Web server (including `fb_sig`) and have it verify the signature internally and respond with either `OK` or `NOT_OK`.

See Also

[fb:flv](#)

[fb:mp3](#)

Fb:flv

Description

Renders a Flash-based FLV player that can stream arbitrary FLV (video/audio) files on the page.

Attributes

Required	Name	Type	Description
required	src	string	The URL of the FLV file. The URL must be absolute.
optional	height	int	The height of the video container in pixels.
	width	int	The width of the video container in pixels.
	title	string	The name of the video.

Examples

Notes

The file supplied must already be encoded as FLV – you **cannot** pass an .avi file into the player, for example.

When using fb:flv and Internet Explorer, width and height may be need to be specified. Otherwise the Flash object renders as a single grey pixel.

See Also

[fb:mp3](#)

[fb:swf](#)

Fb:silverlight

Description

Renders a Microsoft Silverlight control. On profile pages, an image appears first. When the user clicks the image, it turns into the control. On canvas pages, the image does not appear, and the Silverlight control is directly included.

Attributes

Required	Name	Type	Description
required	silverlightsrc	string	The URL of the Silverlight control.
optional	imgsrc	string	The URL of the image (.gif and .jpg formats only). <i>(default value is []. Note that this renders the Silverlight control unusable and invisible on profile pages.)</i>
	height	int	The height of the image and Silverlight control.
	width	int	The width of the image and Silverlight control.
	imgstyle	string	The style attribute for the image.
	imgclass	string	The class attribute for the image.
	swfbgcolor	string	The hex-encoded background color for the Silverlight control.

Examples

```
<fb:silverlight silverlightsrc="http://www.somesite.com/silverlight" />
```

Notes

See Also

Tools

Fb:board

Description

This tag is currently in beta. It is available for all applications. Please post any bugs in bugzilla.

Displays a discussion board for a unique identifier. Facebook handles see all page, topic display, posting and storage.

Fb:board is Facebook discussion board for developers to drop on canvas pages easily. It is not designed to be fully extensible or for the developer to get the data in the posts. Using the tag implies a board exists that can be posted on identified by the passed xid. Please note the following:

Every pageload for posting, see all page, and so forth refetches the configuration from the callbackurl supplied. The callbackurl defaults to the page where it was originally found, so the tag should work without any extra coding. However, you can point a callbackurl to a special page that can be more efficient, if you so desire.

An fb_sig_xid parameter is passed to these pages. Facebook looks for this xid when using this board. You can use it to short-circuit your application logic and only output the fb:board tag with the proper parameters.

When an action occurs, the page is also passed an fb_sig_xid_action parameter. If you return true for the action requested in the fb:board tag, you can assume the action was performed. Possible values are: new_topic, new_reply, edit_topic, edit_post, delete_topic, delete_post, mark_irrelevant, mark_relevant. These actions correspond respectively to these parameters: cancreatetopic, canpost, cancreatetopic, canpost, candelete, candelete, canmark, canmark.

The refetch is in the form of a POST to the URL supplied in the callbackurl, mimicking a canvas page. Note that some attributes may be ignored for the user acting on his or her own post.

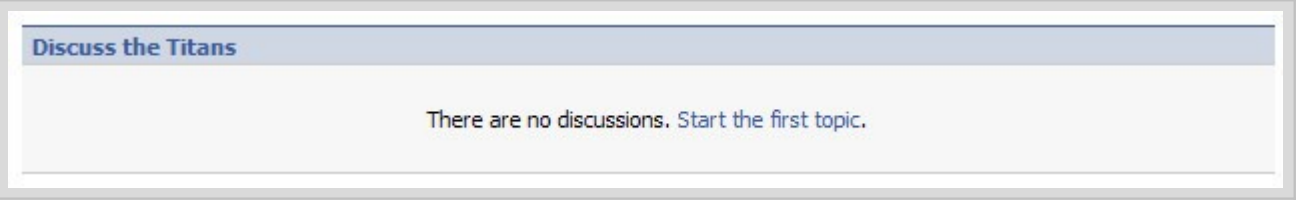
Attributes

Required	Name	Type	Description
required	xid	string	The unique identifier for this board. The board name can contain alphanumeric characters (Aa-Zz, 0-9), hyphens (-) and underscores (_) only.
optional	canpost	bool	Indicates whether the viewing user can post on this board. <i>(default value is true)</i>
	candelete	bool	Indicates whether the viewing user can delete any post or topic on this board. <i>(default value is false)</i>
	canmark	bool	Indicates whether the viewing user can mark a post as relevant or irrelevant. <i>(default value is false)</i>
	cancreatetopic	bool	Indicates whether the viewing user can create a topic on this board. <i>(default value is true)</i>
	numtopics	int	The maximum number of topics to show in the box. <i>(default value is 3)</i>
	callbackurl	string	The URL to refetch this configuration. <i>(default value is the current page)</i>

returnurl [string](#) The URL where the user is returned after selecting a "back" link. *(default value is the current page)*

Examples

```
<fb:board xid="titans_board"
  canpost="true"
  candelete="false"
  canmark="false"
  cancreatetopic="true"
  numtopics="5"
  returnurl="http://apps.facebook.com/myapp/titans/">
  <fb:title>Discuss the Titans</fb:title>
</fb:board>
```



Discuss the Titans

There are no discussions. Start the first topic.

Notes

This tag may contain [fb:title](#)

History

[Nov 7 2007] fb:board went into production as a semi-public beta.

[Dec 10 2007] fb:board is now a fully public beta.

See Also

Fb:comments

Description

Displays a set of comments for a unique identifier. Facebook handles posting, drawing, and see all page. Fb:comments is essentially a wall for developers to drop on canvas pages easily. Using the tag implies a wall-like comments set exists that can be posted or identified by the passed xid. Please note the following:

Every pageload for posting, see all page, and so forth refetches the configuration from the callbackurl supplied. The callbackurl defaults to the page where it was originally found, so the tag should work without any extra coding. However, you can point a callbackurl to a special page that can be more efficient.

An fb_sig_xid parameter is passed to these pages. Facebook looks for this xid. You can use it to short-circuit your application logic and only output the fb:comments tag with the proper parameters.

When an action occurs, the page is also passed an fb_sig_xid_action parameter. Currently this can be "post" or "delete". If you return true for the action requested in the fb:comments tag, you can assume the action was performed.

Note: Facebook redirects immediately to the same page again after passing the post to that page - but without the post information.

The refetch is in the form of a POST to the URL supplied in the callbackurl, mimicking a canvas page.

Attributes

Required	Name	Type	Description
required	xid	string	The unique identifier for this set of comments. Comments can contain alphanumeric characters (Aa-Zz, 0-9), and underscores (_) only.
	canpost	bool	Indicates whether the viewing user can post on this comment set.
	candelele	bool	Indicates whether the viewing user can delete any post on this comment set.
	numposts	int	The maximum number of posts to display.
optional	callbackurl	string	The URL to refetch this configuration. <i>(default value is the current page)</i>
	returnurl	string	The URL where the user is returned after selecting a "back" link. <i>(default value is the current page)</i>
	showform	bool	Boolean whether to show the form (canpost "true" only) for inline posting. Posts using this form will not go to a see-all page after posting, but rather refresh the page.
	send_notification_uid	int	User ID to send a notification to upon someone posting a comment. (Only one uid allowed).

Examples

```
<fb:comments xid="titans_comments" canpost="true" candelete="false"
returnurl="http://apps.facebook.com/myapp/titans/">
<fb:title>Talk about the Titans</fb:title>
</fb:comments>
```



[Code Snippet - Tracking the Comment Count](#)

Notes

This tag may contain [fb:title](#).

Fb:friend-selector

Description

Renders a predictive friend selector input for a given person. You can use this tag inside an [fb:request-form](#) to select users for whom a request can be sent.

Attributes

Required	Name	Type	Description
optional	uid	int	The user whose friends you can select. <i>(default value is the uid of the currently logged-in user)</i>
	name	string	The name of the form element. <i>(default value is friend_selector_name)</i>
	idname	string	The name of the hidden form element that contains the user ID of the selected friend. If you are using this tag inside fb:request-form , do not override the default. <i>(default value is friend_selector_id)</i>
	include_me	bool	Indicates whether or not to include the logged in user in the suggested options. <i>(default value is false)</i>
	exclude_ids	array	A list of user IDs to exclude from the selector. (comma-separated)
	include_lists	bool	Indicates whether or not to include friend lists in the suggested options. <i>(default value is false)</i>

Examples

```
<fb:friend-selector uid="123445" name="uid" idname="friend_sel" />
```



Notes

If the user enters a name that does not match any of his friends, idname gets set to an empty string (instead of a uid number) and name gets set to the entered text.

If no value is entered then idname is not added as a hidden field in the POST.

Bugs

If a valid friend is entered and selected from the drop-down completion, the idname gets set to that person's uid and remains as that value unless another valid friend is entered. This means, if I enter a valid friend and then delete the entry to make it blank (or a non-friend), it sends that previous friend's uid in idname, but sends a blank string (or the non-friend's name) in name.

See Also

You can see a real world example of this tag in the Footprints demo application. Go to [Demos](#) for more details and to download and install the demo apps.

Fb:google-analytics

Description

Inserts appropriate Google Analytics code into a canvas page.

Attributes

Required	Name	Type	Description
required	uacct	string	Your Urchin/Google Analytics account ID.
optional	page	string	The argument given to the urchinTracker() function, either a page or a virtual page.
	ufsc	bool	Sets client info flag, where 1=on and 0=off. <i>This is a UTM user setting. (default value is 1)</i>
	udn	string	Sets the domain name for cookies. Specify auto, none or domain. <i>This is a UTM user setting. (default value is auto)</i>
	uhash	string	Specify whether the unique domain hash for cookies is on or off. <i>This is a UTM user setting. (default value is on)</i>
	utimeout	int	Sets the inactive session timeout in seconds. <i>This is a UTM user setting. (default value is 1800)</i>
	ugifpath	string	Set the Web path to the __utm.gif file. <i>This is a UTM user setting. (default value is /__utm.gif)</i>
	utsp	string	The transaction field separator. <i>This is a UTM user setting. (default value is (a pipe character))</i>
	uflash	bool	Sets the Flash version detection option, where 1=on and 0=off. <i>This is a UTM user setting. (default value is 1)</i>
	utitle	bool	Sets the document title detection option, where 1=on and 0=off. <i>This is a UTM user setting. (default value is 1)</i>
	ulink	bool	Enables linker functionality, where 1=on and 0=off. <i>This is a UTM user setting. (default value is 0)</i>
	uanchor	bool	Indicates whether the use of anchors for campaigns is enabled, where 1=enabled and 0=disabled. <i>This is a UTM user setting. (default value is 0)</i>
	utcp	string	Specifies the cookie path for tracking. <i>This is a UTM user setting. (default value is /)</i>
	usample	int	Represents the sampling percentage of visitors to track, which is a whole number from 1 to 100. <i>This is a UTM user setting. (default value is 100)</i>
	uctm	bool	Sets the campaign tracking module state, where 1=on and 0=off. <i>This is a UTM campaign tracking setting. (default value is 1)</i>
	ucto	int	Sets the timeout in seconds. <i>This is a UTM campaign tracking setting. (default value is 15768000, or 6 months)</i>

uccn	string	The name of the campaign. <i>This is a UTM campaign tracking setting. (default value is utm_campaign)</i>
ucmd	string	Represents the campaign medium. Specify cpc, cpm, link, email or organic. <i>This is a UTM campaign tracking setting. (default value is utm_medium)</i>
ucsr	string	Represents the campaign source. <i>This is a UTM campaign tracking setting. (default value is utm_source)</i>
uctr	string	The campaign term or keyword. <i>This is a UTM campaign tracking setting. (default value is utm_term)</i>
ucct	string	Represents the campaign content. <i>This is a UTM campaign tracking setting. (default value is utm_content)</i>
ucid	int	Represents the campaign ID number. <i>This is a UTM campaign tracking setting. (default value is utm_id)</i>
ucno	string	Indicates whether or not to override the campaign. <i>This is a UTM campaign tracking setting. (default value is utm_nooverride)</i>

Examples

Inserting

```
<fb:google-analytics uacct="UA-9999999-99" />
```

Yields:

```
<script src="https://ssl.google-analytics.com/urchin.js" type="text/JavaScript">
</script>
<script type="text/JavaScript">
_uacct = "UA-9999999-99";
urchinTracker();
</script>
```

Notes

This tag can only be used on canvas pages.

The urchinTracker function provided by Google Analytics can be accessed in [FBJS](#) through Facebook.urchinTracker. To enable, include something similar to the following:

```
<a href="http://www.example.com" onclick="Facebook.urchinTracker('/outgoing/example.com')">
```

Fb:if

Description

Only renders the content inside the tag if value tag is set to true.

Attributes

Required	Name	Type	Description
optional	value	bool	Displays content if true. (<i>default value is false</i>)

Examples

```
<fb:if value="true">Show this content.</fb:if>
```

Fb:else

Description

Handles the else case inside any fb:if, fb:if-* or fb:is-in-network tag.

Applicable tags include:

[fb:if](#)

[fb:if-can-see](#)

[fb:if-can-see-photo](#)

[fb:if-is-app-user](#)

[fb:if-is-friends-with-viewer](#)

[fb:if-is-group-member](#)

[fb:if-is-own-profile](#)

[fb:if-is-user](#)

[fb:if-user-has-added-app](#)

[fb:is-in-network](#)

Examples

```
<fb:if-is-own-profile>  
  Welcome back to your profile!  
  <fb:else>This is not your profile.</fb:else>  
</fb:if-is-own-profile>
```

See Also

You can see a real world example of this tag in the Who's Showing Up demo application. Go to [Demos](#) for more details and to download and install the demo apps.

Fb:mobile

Description

Renders the contained content only when viewed on the mobile website (<http://m.facebook.com>). Any content outside these tags does **not** get rendered on the mobile website.

Examples

This text appears on the regular site, but not on m.facebook.com.
<fb:mobile>This text appears only on m.facebook.com.</fb:mobile>

Fb:multi-friend-input

Description

Renders a multi-friend form entry field like the one used in the message composer. You can use the field inside an [fb:request-form](#) to select users for whom a request can be sent.

Output Adds the following hidden tag inside the form for each user with their [user ID](#) for the value.

```
<input type="hidden" value="123456789" name="ids[]"/>
```

Here is an example, in PHP, of how to access these IDs.

```
$friends = (isset($_REQUEST["ids"])) ? $_REQUEST["ids"] : 0;
```

If users were entered into the multi-friend-input, \$friends becomes an array of IDs; otherwise, \$friends is 0.

Attributes

Required	Name	Type	Description
optional	width	int	The width of entry field. <i>(default value is 350px)</i>
	border_color	string	The color of entry field border. <i>(default value is #8496ba)</i>
	include_me	bool	Indicates whether or not to include the logged-in user in the form. <i>(default value is false)</i>
	max	int	The maximum number of people that can be selected. <i>(default value is 20)</i>
	exclude_ids	array	A comma separated list of user IDs to exclude from the selector.
	prefill_ids	array	A comma separated list of user IDs to pre-populate in the selector. Note that this cannot be used inside an <code><fb:request-form></code> .
	prefill_locked	bool	Set to true to prevent editing of the pre-populated IDs.

Examples

```
<fb:multi-friend-input width="350px" border_color="#8496ba" exclude_ids="4,5,10,15" />
```



Notes

When used inside a [mock AJAX](#) form, the ID array contains only the last UID in the multi-friend-input. [\[possible bug?\]](#)

You can access all of the values of multi-friend-input using form serialize.

When used, it appears to disable the recipient input box on the popup for fb:share-button. [\[possible bug?\]](#)
When more than one instance of this tag appears on a page, only the first instance works. [\[possible bug?\]](#)

Fb:random

Description

Randomly chooses an item inside the tags based on the weights provided.

fb:random allows the developer to input a series of tags, of which one or more are shown randomly. Each item can have a weight and the tag can be specified to show more than one choice. Each option should be wrapped in an [fb:random-option](#) tag.

Attributes

Required	Name	Type	Description
optional	pick	int	The number of items to choose from the random subset. <i>(default value is 1)</i>
	unique	bool	Indicates whether to force uniqueness if pick > 1. <i>(default value is true)</i>

Examples

```
<fb:random>
  <fb:random-option weight="2">A: This will be shown 2 times as often as B.</fb:random-option>
  <fb:random-option weight="1">B: This will be show half as often as A</fb:random-option>
</fb:random>
```

a screenshot is not applicable for this tag

Notes

```
<fb:random>
  <fb:random-option weight="2">A: This will be shown 2 times as often as B.</fb:random-option>
  <fb:random-option weight="1">B: This will be show half as often as A</fb:random-option>
</fb:random>
```

a screenshot is not applicable for this tag

Fb:random-option

Description

Contains code to be output when selected by the [fb:random](#) tag. You can control the frequency of this pick with the weight attribute.

The fb:random-option tag must be inside an [fb:random](#) tag.

Attributes

Required	Name	Type	Description
optional	weight	float	Allows for controlling the frequency of a choice. <i>(default value is 1.0)</i>

Examples

```
<fb:random>
  <fb:random-option weight="2">A: This text appears twice as often as B.</fb:random-option>
  <fb:random-option weight="1">B: This text appears half as often as A.</fb:random-option>
</fb:random>
```

a screenshot is not applicable for this tag

Notes

Weights of options are relative to the number of options left, which shrinks if `unique==true` and `pick>1` in the parent [fb:random](#) tag.

Weights of options can be any float > 0 . The weights are relative to each other, not necessarily to 1.

Fb:switch

Description

Evaluates every fb: tag inside and returns the first one that evaluates to anything other than an empty string. You can use [fb:default](#) to specify a default that gets rendered if nothing else does before it was executed.

Examples

```
<fb:switch>  
  <fb:photo pid="12345" />  
  <fb:profile-pic uid="54321" />  
  <fb:default>You can't see either the photo or the profile pic</fb:default>  
</fb:switch>
```

Note: fb:default always returns its content. Thus, if it is the first tag within fb:switch, then no other tags get tested.

Fb:default

Description

This article or section needs expansion

For a group of fb: tags contained within an [fb:switch](#) tag, the fb:default tag renders any content inside itself if no other fb: tag inside the [fb:switch](#) tag renders code before it.

Examples

```
<fb:switch>
  <fb:photo pid="12345" />
  <fb:profile-pic uid="54321" />
  <fb:default>You can't see either the photo or the profile pic</fb:default>
</fb:switch>
```

a screenshot is not applicable for this tag

Fb:user-agent

Description

Displays the contents wrapped inside the tag to the specified user-agents. You can use fb:user-agent on the canvas page and the profile box. It serves as a tool to deal with browser idiosyncrasies.

User-Agent Resources

<http://www.user-agents.org/index.shtml> - a list of common user agent strings

<http://whatsmyuseragent.com/> - what is your user agent string?

Attributes

Required	Name	Type	Description
required	includes	string	A comma-delimited list of strings to test for inclusion against the HTTP request's user-agent string. If a given include string matches the user-agent string, the content inside the include tag appears; otherwise it does not.
optional	excludes	string	A comma-delimited list of strings to test for exclusion against the HTTP request's user-agent string. If a given include string matches the user-agent string, the content inside the include tag does not appear; otherwise it does appear.

Examples

```
<fb:user-agent includes="ie 6,ie 7">  
Hi IE 6/7 users.  
</fb:user-agent>
```

```
<fb:user-agent excludes="firefox">  
Hi people not using Firefox.  
</fb:user-agent>
```

```
<fb:user-agent includes="mozilla" excludes="firefox/2.0">  
Hi Mozilla (but not Firefox 2.0) users.  
</fb:user-agent>
```

```
<fb:user-agent includes="ie" excludes="ie 6">  
Hi IE (but not IE 6) users.  
</fb:user-agent>
```

Notes

If both the includes and excludes attributes exist, the order of operation is to evaluate the includes then the excludes. For example, you can state includes="MSIE" excludes "MSIE 6.0". The matching is case-insensitive.

Fb:typeahead-input

Description

This tag is currently in beta. Please post bugs in the [Bug Tracker](#).

Creates a type-ahead tool (as suggested) that will give you the results that you specify. To add options in the input box, use [fb:typeahead-option](#).

Attributes

Required	Name	Type	Description
optional	name	string	The variable name that is sent in the POST request when the form is submitted.

Examples

```
fb:fbml version="1.1">
<fb:typeahead-input name="your_input_name">
  <fb:typeahead-option value="me"><fb:name uid=1160 capitalize=yes linked=no /></fb:typeahead-option>
  <fb:typeahead-option value="ex">x</fb:typeahead-option>
  <fb:typeahead-option>why</fb:typeahead-option>
</fb:typeahead-input>
</fb:fbml>
```

Notes

The tag must be wrapped in a [fb:fbml](#) tag with version 1.1.

You must name the input.

You must set autocomplete="off" to prevent the browser's autocomplete from overriding this tag's.

Known Issues

In profile boxes, the autocomplete options cannot extend beyond the edges of the box the application owns.

HTML and FBML that generates HTML inside of fb:typeahead-option is being interpreted as literal HTML

Bug Report

A bug report was filed concerning the wrong values of the input posting.

http://bugs.developers.facebook.com/show_bug.cgi?id=1055

A bug report was filed regarding the number of inputs the typeahead can handle and the rendering of type-ahead suggestions.

http://bugs.developers.facebook.com/show_bug.cgi?id=1145

unable to use indexed name in fb:typeahead value

http://bugs.developers.facebook.com/show_bug.cgi?id=1371

needs ability to set the currently 'selected' item

http://bugs.developers.facebook.com/show_bug.cgi?id=1372

Fb:typeahead-option

Description

This tag is currently in beta. Please post bugs in the [Bug Tracker](#).

This tag specifies the values for a typeahead form input. You must use it in conjunction with [Fb:typeahead-input](#).

Attributes

Required	Name	Type	Description
optional	value	string	The value assigned to the fb:typeahead-input element, used in the POST request when the form is submitted.

Examples

```
<fb:fbml version="1.1">
  <fb:typeahead-input name="your_input_name">
    <fb:typeahead-option value="me"><fb:name uid=1160 capitalize=yes linked=no /></fb:typeahead-option>
    <fb:typeahead-option value="ex">x</fb:typeahead-option>
    <fb:typeahead-option>why</fb:typeahead-option>
  </fb:typeahead-input>
</fb:fbml>
```

Misc

Fb:js-string

Description

This tag renders a block of FBML into an FBML block variable instead of rendering it on the page. You can use this variable in your JavaScript with `setInnerFBML`. See [FBJS](#) for more information.

Attributes

Required	Name	Type	Description
required	var	string	A valid JavaScript identifier.

Examples

```
<fb:js-string var="example">Pre-rendered FBML content.</fb:js-string>
```

a screenshot is not applicable for this tag

```
<fb:js-string var="name.element1">This is the first element.</fb:js-string>
<fb:js-string var="name.element2">This is the second element.</fb:js-string>
```

This allows you to access name as if it were a JavaScript array.

The array can only be one level deep, and associative names must be prefixed by at least 1 alphabetical character (name.5, name.6, etc. will not work).

Real example (PHP):

```
<fb:js-string var="articles.id{$article['article_id']}">
  Do you really want to delete article #{$article['article_id']}?<br /><br />
  Associated contributors to this article will remain credited.
</fb:js-string>
```

```
<a href="#" onclick="removeArticle( {$article['article_id']})">Delete Article</a><span
id="removeSpan{$article['article_id']}"></span>
```

```
<script>
function removeArticle(articleId) {
  var dialog = new Dialog(Dialog.DIALOG_CONTEXTUAL);
  dialog.setContext(document.getElementById("removeSpan"+articleId));
  dialog.showChoice('Confirm Removal', articles["id"+articleId], 'Yes', 'Cancel');
  dialog.onconfirm = function() {
    document.setLocation('http://apps.facebook.com/facebookdocs/removearticle.php?article='+articleId);
  };
}
```

```
</script>
```

Fb:fbml

Description

Defines a block of FBML to be rendered in a specific version of [FBML](#).

Attributes

Required	Name	Type	Description
optional	version	float	The version of FBML with which to render the content. <i>(default value is [the current version])</i>

Examples

```
<fb:fbml version="1.0">  
  The tags that follow rely completely on FBML in version 1.0.  
  This is being rendered in version: <fb:fbmlversion />  
</fb:fbml>
```

```
<fb:fbml version="1.2">  
  The tags here rely on FBML 1.2.  
  This is being rendered in version: <fb:fbmlversion />  
</fb:fbml>
```

Errors

RUNTIME ERROR: fb:fbml: FBML version does not exist: 1.2

See Also

[fb:fbmlversion](#)

Fb:fbmlversion

Description

Prints the version of FBML currently in scope. You should use this tag only for debugging purposes.

Attributes

Required	Name	Type	Description
optional	version	float	The version of FBML with which to render the content. <i>(default value is [the current version])</i>

Examples

```
<fb:fbml version="1.0">  
  The tags that follow rely completely on FBML in version 1.0.  
  This is being rendered in version: <fb:fbmlversion />  
</fb:fbml>
```

```
<fb:fbml version="1.2">  
  The tags here rely on FBML 1.2.  
  This is being rendered in version: <fb:fbmlversion />  
</fb:fbml>
```

Errors

```
RUNTIME ERROR: fb:fbml: FBML version does not exist: 1.2
```

Notes

As you can in the above code example, the second code snippet does not work because there is no version 1.2 of FBML at this time.

See Also

[fb:fbml](#)

Fb:redirect

Description

Redirects a user's browser to a new URL within the Facebook canvas.

Attributes

Required	Name	Type	Description
required	url	string	The URL where you are redirecting the browser. Note this does not work within a profile box.

Examples

```
<fb:redirect url="http://apps.facebook.com/myapp/?not_in_group" />
```

a screenshot is not applicable for this tag

Fb:ref

Description

Fetches and renders [FBML](#) from a given ref source – either a ref string "handle" you've created using [fbml.setRefHandle](#) or a URL that serves FBML. You can use this ref to publish identical FBML to a large number of user profiles and subsequently update those profiles, **without** having to republish FBML on behalf of each user (that is, using [profile.setFBML](#) for each user). For a high level discussion of the benefits of fb:ref and how to use it, [read this forum post](#)

Attributes

Required	Name	Type	Description
required	url	string	The URL from which to fetch the FBML. Facebook caches the content retrieved from that URL until you call fbml.refreshRefUrl . You must specify either url or handle, but not both.
	handle	string	The string previously set by fbml.setRefHandle that identifies the FBML. You must specify either url or handle, but not both.

Examples

Using Key/Value Pairs

When using key/value pairs, Facebook stores developer-specified FBML on its servers in the form of a hashtable.

To set the FBML code of YourHandle, you must call `fbml.setRefHandle` at least once.

```
fbml.setRefHandle("YourHandle", "FBML_Content")
```

Next, include the FBML referenced by the handle. Insert the `<fb:ref>` tag into the appropriate place in your markup:

```
<fb:ref handle="YourHandle" />
```

Using the URL Attribute

When using fb:ref with the url attribute, Facebook downloads the specified URL, caches the result, and parses the FBML, passing it to the user.

To include the specified FBML, insert the `<fb:ref>` tag into the appropriate place in your markup:

```
<fb:ref url="http://www.mysite.com/someurl.php" />
```

To clear the cache, call [fbml.refreshRefUrl](#):

```
fbml.refreshRefUrl("http://www.mysite.com/someurl.php")
```

Notes

Performance

If you have to update a large number of refs, you want to be using handle refs, since you can do it with a single push, rather than URL refs, which will hammer your server on the callback.^{[[verify](#)]} You may also want to stream out your updates in parallel, which can be easily accomplished by using `curl_multi` instead of `curl` in the PHP5 facebook rest client. There's a [guide available](#) for this.

Cache Expiration

As mentioned within a [forum thread](#) by a member of the Facebook team: "You only need to publish on update. We'll keep it otherwise."

So the cache never expires and you will not need to periodically update your application cache. Facebook will keep the cache indefinitely. However, developers should not rely on the cache always keeping their data – i.e. it is not safe to assume that once data has been cached (in the case of a URL ref), it can be deleted from your server.

Nested Refs

The contents of a ref handle can contain FBML, including other `<fb:ref>` tags.

Warning: An infinite nesting can be created by creating two ref handles that refer to each other. Do **not** do this.

FBML Anomalies in Refs

Some FBML tags do not currently appear to operate as expected when nested inside a ref handle. ^[possible bug?]

These include: `<fb:if-is-own-profile>`. ^[verify]

- (RobRoy) I can use `<fb:if-is-own-profile>` inside a ref handle just fine. Anyone else still experiencing this?

This writer has not been able to successfully create mock-Ajax forms inside of ref handles ^[possible bug?] (but that does not necessarily mean they don't work).

See Also

[Changing profile content](#)

Fb:share-button

Description

Renders a standard Share button in a profile for the specified URL or content.

Attributes

Required	Name	Type	Description
required [<i>verify</i>]	class	string	The type of share. Valid values are url, to render a share of the URL specified with the href attribute, and meta, to render a share with the given data.
required	href	string	The reference URL to share. This attribute is required for the url class only.
optional	meta	string	The metadata about the shared item. See descriptions of the necessary data. The meta class may contain this attribute.
	link	string	The content (such as image thumbnails) for the shared item. See descriptions of the necessary data. The meta class may contain this attribute.

Examples

```
<fb:share-button class="url" href="http://apps.facebook.com/example" />
```



```
<fb:share-button class="meta">
  <meta name="medium" content="blog"/>
  <meta name="title" content="Leonidas in All of Us"/>
  <meta name="video_type" content="application/x-shockwave-flash"/>
  <meta name="video_height" content="345"/>
  <meta name="video_width" content="473"/>
  <meta name="description" content="That's the lesson 300 teaches us."/>
  <link rel="image_src"
href="http://9.content.collegehumor.com/d1/ch6/f/6/collegehumor.b38e345f621621dfa9de5456094735a0.jpg"/>
  <link rel="video_src"
href="http://www.collegehumor.com/moogaloop/moogaloop.swf?clip_id=1757757&autoplay=true"/>
  <link rel="target_url" href="http://www.collegehumor.com/video:1757757"/>
</fb:share-button>
```



Fb:time

Description

Renders the date and time in the user's time zone.

Attributes

Required	Name	Type	Description
required	t	int	The time to display in epoch seconds.
optional	tz	string	The time zone in which to display t. Acceptable formats include PHP's List of Supported Timezones and +/- formats such as <i>Etc/GMT-7</i> . Note: Due to a bug (see below) when using a timezone in the <i>Etc/GMT</i> format the time is relative to the timezone. (default value is loggedinuser's timezone)
	preposition	bool	Indicates whether to automatically insert prepositions as appropriate into the time, where "at" prepends the time and "on" prepends the date if it appears. (default value is false)

Examples

```
<fb:time t='1191648451'/>
```

```
<fb:time t="1180502413" tz="America/New_York" />
```

Notes

The time expands as needed. If t took place in the same day, only *hour:minutes[am|pm]* appears. If t is in the same year, *Month Day hour:minutes[am|pm]* appears. Otherwise, *Month Day, Year hour:minutes[am|pm]* appears.

When timezone is entered in the *Etc/GMT* format the time is interpreted as relative to that timezone. This bug has been [reported](#) (and may be a smaller subset of [this bug](#)). The same behaviour does not occur when the timezone is in the preferred *Continent/City* format.

See Also

You can see a real world example of this tag in the Footprints demo application. Go to [Demos](#) for more details and to download and install the demo apps.

Fb:title

Description

Sets the page's <title> tag to its contents. Alternatively, when used inside [fb:comments](#), sets the title for the Wall.

Examples

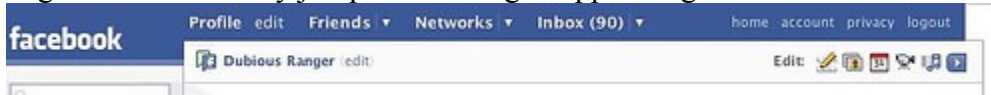
```
<fb:title>This is the canvas page window title</fb:title>
```

```
Facebook | App Name | This is the canvas page window title
```

Fb:page-admin-edit-header

Description

For apps that can be added to Facebook Pages, this adds a standardized edit header for canvas pages so that the Page owner can easily jump to their Page's app configuration.



Notes

This tag should be placed at the top of all pages that the Page admin can use to edit their application on their page.

This tag only works on a canvas page.

This tag should only be displayed to an admin of a Facebook Page that has your application added.

See Also

Forms

Fb:submit

Description

Creates a JavaScript submission mechanism for a form, which makes image or text links act as Submit buttons. Markup contained by this tag is surrounded with an `<a>` tag that includes a submit onclick action.

This article or section needs expansion

Attributes

Required	Name	Type	Description
optional	form_id	string	The ID of the form to be submitted.

Examples

```
<fb:submit></fb:submit>
```

Message/Wall attachments

Fb:attachment-preview

Description

Renders a link in a wall or message attachment that, when clicked, replaces that attachment with newly fetched content. The URL from which the content is fetched is a hidden input in the current content, and all other inputs in the content are posted.

This article or section needs expansion

Attributes

Examples

This is a basic outline of how to setup attachments.

Basically, set the Attachment callback URL to some page on your server. For instance

<http://yourserver.com/attach.php>

attach.php will handle both the attachments, and the selection of the attachments.

```
<?php
if ($_POST['message_sent'] < 1)
{
// The user is selecting the attachment, in other words, preview mode
// Print out form elements that allow the user to select the attachment
echo '<fb:editor-text label="Sample" name="sample" value="'
. htmlspecialchars($_POST['sample']) . "'/>';

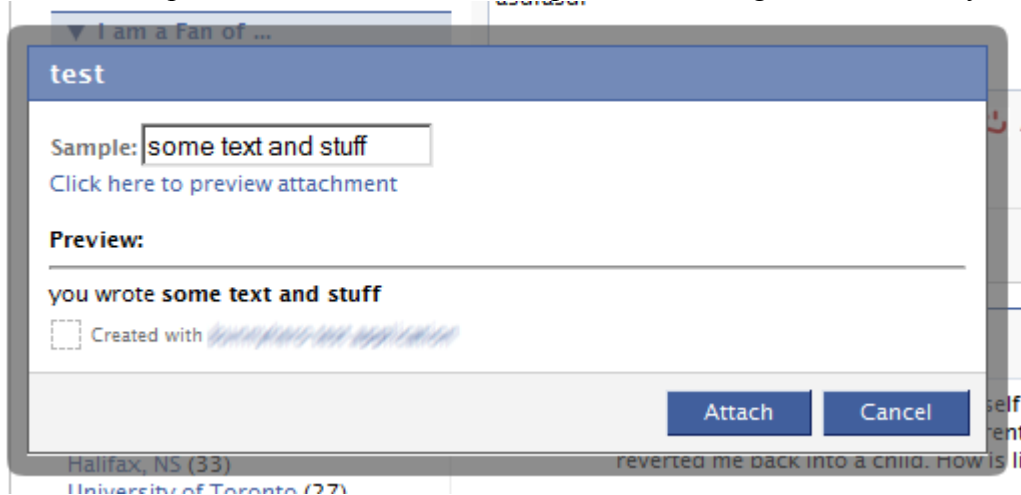
// The following line tells facebook to use the same script for attachments
echo '<input type="hidden" name="url" value="http://yourserver.com/attach.php" />';

// The following line lets your user preview the attachment they've selected
echo "<fb:attachment-preview>Click here to preview attachment</fb:attachment-preview><br /><br />";

echo "<b>Preview:</b>";
echo "<hr />";
}
else
{
// Do anything here you want to do for attached objects only
echo "(The object is attached)<br/><br/>";
}

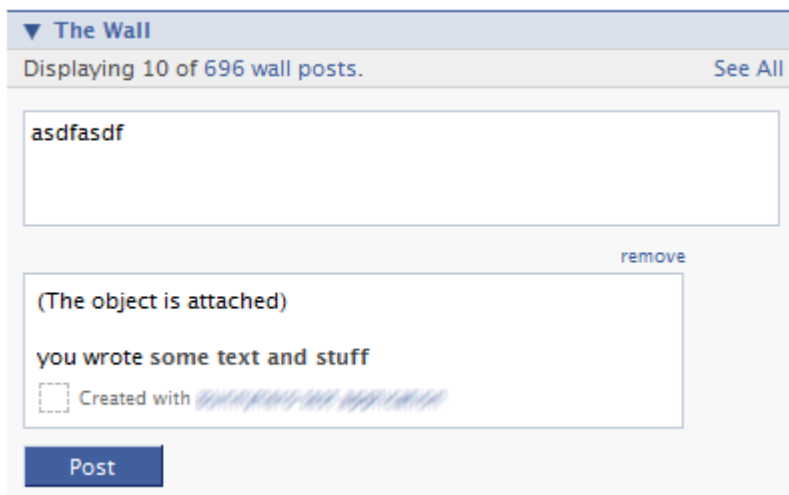
// Display your attachment here. This will show up both for preview and live attachments
echo "you wrote <b>" . htmlspecialchars($_POST['sample']) . "</b><br />";
?>
```

When the user clicks your "attach" link, Facebook calls your script and displays it in a pop-up dialog like the one below. Each time the user clicks the "Preview" link, Facebook calls your attach script again with the values of the form inputs as POST variables, and updates the dialog with whatever your script returns.



"Preview" screenshot from the sample code

When the users clicks the "Attach" button, your script will be called with the POST variable message_sent set to 1, and uses your script's output as the attachment:



"Attached" screenshot from the sample code

You're given the user doing the attachment as fb_sig_user, but remember that users who don't have your app installed can still make attachments on their friends' walls (in this case you won't be given fb_sig_user).

Elements Needed/Steps

1) Go to My Applications and click on Edit Settings for your app.

a) At the bottom, under Attachments, for "Attachment Action:" type in the text you'd like to appear in the wall attachment area. For example "Attach Photo" or "Slap x"

b) In "Callback URL:" type in the URL to where you will house your code, i.e. attach.php. Include the full URL and have it be the URL of your site, not the apps.facebook.com URL. For example,

<http://www.yourserver.com/attach.php>

2) On your callback for attachment page, you will need to have field inputs where the name is 'url' and the value is the URL to the page where you will display what they chose. For example:

```
<table width="100%" border="0" cellspacing="0" cellpadding="3">
  <tr>
    <td><input name="url" type="radio"
value="http://yourserver.com/facebook/attachment.php?actiontotake=choice1"> Choice One </td>
  </tr>
  <tr>
    <td><input name="url" type="radio"
value="http://yourserver.com/facebook/attachment.php?actiontotake=choice2"> Choice Two</td>
  </tr>
  <tr>
    <td><input name="url" type="radio"
value="http://yourserver.com/facebook/attachment.php?actiontotake=choice3"> Choice Three </td>
  </tr>
</table>
```

NOTE: Do not wrap this in a form tag, Facebook is placing the above in their own.

3) On your attachment.php page, just test for which choice they made and display what the attachment should be accordingly.

NOTE: Wall attachments used to be displayed in a collapsed mode on page load, with a preview image specified by the fb:wall-attachment-img tag. However, wall attachments are currently displayed in full (just like message attachments), so the fb:wall-attachment-img tag has been deprecated.

Notifications and requests

Fb:notif-subject

Description

Specifies the content of the email subject line for a notification sent with the [notifications.send](#) call.

Examples

```
<fb:notif-subject>You have a new message!</fb:notif-subject>
```

See Also

You can see a real world example of this tag in the Footprints demo application. Go to [Demos](#) for more details and to download and install the demo apps.

Fb:notif-page

Description

Specifies content of a notification that appears on a user's [Notifications page](#). This tag is used with the [notifications.send](#) call.

Examples

```
<fb:notif-page>Please check out <a href="http://apps.facebook.com/developer">this awesome app!</a></fb:notif-page>
```

See Also

[fb:notif-email](#)

[fb:notif-subject](#)

Fb:notif-email

Description

Specifies content of the email body for a notification sent with the [notifications.send](#) call.

Examples

```
<fb:notif-email>You have a new message!</fb:notif-email>
```

See Also

[fb:notif-page](#)

[fb:notif-subject](#)

Fb:req-choice

Description

Specifies a button to be shown at the bottom of a [request](#) on the user's [requests page](#).

Attributes

Required	Name	Type	Description
required	url	string	The URL to which the button should take the user upon click. <i>must be an absolute</i>
	label	string	Specifies the text to display on this button.

Examples

```
<fb:req-choice url="http://apps.facebook.com/example/confirm.php" label="Confirm" />
```

Notes

An Ignore button is added by default and will dismiss the request. Including Ignore as a req-choice often results ^{[[who says?](#)]}_{[[possible bug?](#)]} in the loss of all buttons. Using the same URL for multiple buttons will cause only the last button with the URL to be shown.

Fb:request-form

Description

Creates a form that sends requests to the selected users. To send requests/invitations to users, create one of these forms and include either an [fb:multi-friend-selector](#) or any combination of [fb:multi-friend-input](#), [fb:friend-selector](#) and [fb:request-form-submit](#). The form is submitted after the user confirms the sending of the request.

In the case where you are not using an [fb:multi-friend-selector](#), the only way for a user to submit a request/invitation through this form is via an [fb:request-form-submit](#) button.

In general, use [fb:multi-friend-selector](#) in a nearly full-page invitation interface where the user is intended to select a large number of people, and [fb:friend-selector](#) or [fb:multi-friend-input](#) in situations where the user is selecting a smaller number of users and you want to integrate it into the context of your own page. As a middle-ground alternative, you can use the [fb:multi-friend-selector \(condensed\)](#) for places where the user might select a medium-sized list of people without needing a full-page interstitial invitations interface.

For developers using iframes, you cannot get the full flexibility of the **fb:request-form** system. However, you can still send requests using the [Multi_friend_selector](#), or you can use the [fb_force_mode](#) parameter to render a particular canvas page with FBML.

For developers wishing to utilize more form parameters, typical form input can be used with the attribute, `fb-protected="true"`, within any `fb:request-form-submit` tag.

Attributes

Required	Name	Type	Description
required	<code>type</code>	string	The type of request or invitation to generate. This corresponds to the word that is displayed on the home page. For example, "event."
	<code>content</code>	string	The contents of the request or invitation to be sent. It should use FBML formatting that contains only links and the special tag <code><fb:req-choice url="" label="" /></code> to specify the buttons to be included in the request. Make sure that you properly encode this attribute. For example, use the PHP "htmlentities" function.
optional	<code>invite</code>	bool	Set this to true if you want to send an invitation or false if you want to send a request. The difference between them is in the content that the user sees. <i>(default value is false)</i>
	<code>action</code>	string	The place where a user gets redirected after submitting the form through the fb:request-form-submit button or when they click Skip this Step . By default the user is directed to http://apps.facebook.com/yourapp/null . <i>Note: This default behavior may not be preserved.</i> ^[verify]
	<code>method</code>	string	Set it to either GET or POST, as you would with a form.

Post Variables

Required	Name	Type	Description
	<code>typeahead</code>	string	The string containing any content the user typed into the multi-selector box.

`ids` [array](#) A zero-based array containing all of the [user IDs](#) of the people the user invited.

Examples

Example FBML Invite Page

```
<fb:fbml>
<fb:request-form
action="index.php"
method="POST"
invite="true"
type="Librarian"
content="Your text goes here. <?php echo htmlentities("<fb:req-choice url=\"YOUR URL\" label=\"Add My
APP!\" />"); ?>">

<fb:multi-friend-selector
showborder="false"
actiontext="Invite your friends to use Librarian.">

</fb:request-form>
</fb:fbml>
```

Example Java Invite Page

Here is the FBML:

```
<fb:request-form
  action="start.htm"
  method="POST"
  invite="true"
  type="nutshOt network"
  content="nutshOt network is the best place on Facebook for viewing, sharing and giving
  friends the highest quality nutshOts. Join me on the nutshOt network!
  <fb:req-choice url='http://www.facebook.com/add.php?api_key=<c:out value='${apiKey}' />'
  label='Check out the nutshOt network!' />
">

  <fb:multi-friend-selector
    showborder="false"
    actiontext="Invite your friends to the nutshOt network."
    exclude_ids="<c:out value='${appUserFriends}' />"
    max="20" />
</fb:request-form>
```

Here is the server side Java:

```
protected Map referenceData(HttpServletRequest request) throws Exception {
    logger.info("InviteFriendsFormController.referenceData ...");

    FacebookRestClient facebookRestClient = getFacebookUtils().getFacebookRestClient(request);

    //get the list of friends who are app users so they can be excluded from the invite page
    String appUserFriends = getFacebookUtils().getAppUsersFriendsString(facebookRestClient);

    HashMap referenceData = new HashMap();
    referenceData.put("appUserFriends", appUserFriends);
    referenceData.put("apiKey", PropertyUtils.getInstance().getPropertyValue("API_KEY"));
    getFacebookUtils().loadUrlData(referenceData);

    return referenceData;
}
```

You can see it in action at [The nutshOt network](#)













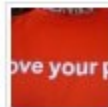


nutshOt network

Invite your friends to the nutshOt network.

Add up to 8 of your friends by clicking on their pictures below.

Find Friends: Filter by Network

View All
Selected (0)
Unselected

 Ablo Shawn Bristol	 Andy Chang Stanford	 Arlene Oca	 Brad English Edmonton, AB
 Brandon Bidewell UBC	 Brent Tatonka Arcand	 Brian Coghlan Edmonton, AB	 Camilla Westberg Los Angele...
 Candace Benson	 Chris Bedrosian Los Angele...	 Chris Stevens Dunn Edmonton, AB	 Christina Baker Diederich Edmonton, AB
 Christina Little Given Edmonton, AB	 Christine Buckius	 Corey Clements Edmonton, AB	 Craig Martin Edinburgh
 Crighton Nichols Australia	 Curtis Andreotti Calgary, AB	 Cynthia Pavana Oracle	 Daniel Sanders Sony Pictu...

Send nutshOt network Invitation

Skip this Step

[1](#) [2](#) [3](#) Next

Example PHP Invite Page

```
<?php
```

```
require_once('config.php');
```

```
require_once('init.php');
```

```
require_once('lib.php');
```

```
$user = $facebook->require_login();
```

```
// Get list of friends who have this app installed...
```

```
$rs = $facebook->api_client->fql_query("SELECT uid FROM user WHERE has_added_app=1 and uid IN (SELECT uid2 FROM friend WHERE uid1 = $user)");
```

```
$arFriends = "";
```

```
// Build an delimited list of users...
if ($rs)
{
    for ( $i = 0; $i < count($rs); $i++ )
    {
        if ( $arFriends != "" )
            $arFriends .= ",";

        $arFriends .= $rs[$i]["uid"];
    }
}

// Construct a next url for referrals
$sNextUrl = urlencode("&refuid=".$user);

// Build your invite text
$invfbml = <<<FBML
You've been invited to join the PickPocket™ Guild!
<fb:name uid="$user" firstnameonly="true" shownetwork="false"/> wants you to add PickPocket™ so that you
can join <fb:pronoun possessive="true" uid="$user"/> wily band of thieves!
<fb:req-choice url="http://www.facebook.com/add.php?api_key=$appapikey&next=$sNextUrl" label="Join the
Guild!" />
FBML;

?>



















<fb:request-form type="PickPocket" action="index.php?c=skipped" content="<?=htmlentities($invfbml)?>"
invite="true">
    <fb:multi-friend-selector max="20" actiontext="Here are your friends who don't have PickPocket™.
Invite them to play with you!" showborder="true" rows="5" exclude_ids="<?=$arFriends?>">
</fb:request-form>
```

Pp PickPocket
+ Recruit Your Friends!

Here are your friends who don't have PickPocket™. Invite them to play with you!

Add up to 15 of your friends by clicking on their pictures below.

Find Friends: Filter by Network

View All	Selected (0)	Unselected	
 Adam Rifkin Silicon Va...	 Alan Lewis eBay	 Alec Peters	 Alex Mehr Berkeley
 Andrea B Newman Indiana	 Anne Richard-... Edmonton, AB	 Bees Knees San Franci...	 Brian Vanderhoff Atlanta, GA
 Cami Searles London	 Chrissy Hopper	 Cornelius Geary San Franci...	 Dan Ablan Chicago, IL
 Dan Tudor	 Devin Hendricks Washington...	 Donovan McGroovy Sonoma / N...	 Douglas Dij Duerring Pittsburgh...
 Eileen Montelione	 Fred Maidment Atlanta, GA	 Gary DuVall Chicago, IL	 Guy Kawasaki Silicon Va...

Send PickPocket Invitation
Skip this Step

1 2 3 Next

Pp Page built by PickPocket

[about](#) [developers](#) [jobs](#) [advertisers](#) [polls](#) [terms](#) [privacy](#) [help](#)

Example Overloaded Form

```
<fb:fbml>
```

```
<fb:request-form method="POST" action="compare.php" content="some content" type="" invite="false">
```

```
  <input id="first_overloaded_id" type="hidden" fb_protected="true" value="first_overloaded_value"
  name="first_overloaded_name"/>
```

```
  <input id="second_overloaded_id" type="hidden" fb_protected="true"
  value="second_overloaded_value" name="second_overloaded_name"/>
```

```
<div class="clearfix" style="padding-bottom: 10px;">
  <fb:multi-friend-selector condensed="true" style="width:80%;" />
</div>
<fb:request-form-submit />
</fb:request-form>
</fb:fbml>
```

Notes

Make sure to include whatever attributes you would normally include if you were creating a `<form>` tag, such as `action` and `method`, as mentioned above.

The [fb:multi-friend-selector](#) tag does not show up in Internet Explorer 6. [\[verify\]](#)

The `type` attribute appears to have a hard limit of 20 characters. Anything else is truncated.

See Also

Fb:multi-friend-selector

Description

There are actually two versions of this button - the full version and the condensed version. This page describes the full version. For information about the condensed version, see [fb:multi-friend-selector \(condensed\)](#).

This is a nearly full-page interface intended to be used on canvas pages to allow the user to send a "large" number of requests or invitations (where "large" is generally some number more than 4). This tag must be used inside an [fb:request-form](#) tag. This interface includes a series of `<input type="hidden" name="ids[]" value="[friend id]">` which are included for selected users in the form that gets submitted to your `<fb:request-form>` action URL. Both the **Skip this Step** button and the **Submit** button take the user to the parent `fb:request-form` action URL.

For developers using iframes, a full page version of this is available. Please see [Multi_friend_selector](#) for information about using it.

A user may invite a number of people each day equal to the maximum number of requests allocated to your application. If a user attempts to go over this limit, the following message appears:



Attributes

Required	Name	Type	Description
required	actiontext	string	An instructional message to display to users at the top of the multi-friend-selector.
optional	showborder	bool	Indicates whether you want a border around the outside of the multi-friend-selector.
	rows	int	The number of rows of friends to show in the multi-friend-selector. <i>(default value is 5 and the value must be between 3 and 10)</i>
	max	int	The maximum number of users that can be selected. This value ranges from 1 to 35, and is capped at the number of friend requests the user has remaining under their limit. This attribute is ignored if it is greater than the number of requests your application is able to send.
	exclude_ids	array	A comma-separated list of user IDs to exclude from the multi-friend-selector.
	bypass	string	The version of the Bypass button you want to use. Set this attribute to "step", "cancel", or "skip", which results in Skip This Step , Cancel , or Skip , respectively. <i>(default value is skip)</i>

Post Variables

Required	Name	Type	Description
	ids	array	An array of the user IDs chosen by the user.
















Examples

```
<fb:multi-friend-selector actiontext="Select the friends you want to invite. (All of them.)" rows="3"/>
```

Select the friends you want to invite. (All of them.)
Add up to 15 of your friends by clicking on their pictures below.

Find Friends: Filter by Network

View All Selected (0) Unselected

 Calvin Gan New Zealand	 Chris Loeffen Hogeschool...		 Chris Peters New Zealand	 Christina Sweet
 Francis Szekeres Franciscan...	 Ian David Fraser San Diego,...	 Jacquie Mckee Tri-Cities...		 James Alan Davis
 James Mullen C. Washing...		 Jamie Ellis	 Jewels V Arizona	 Joel Slater

Fb:multi-friend-selector (condensed)

Description

There are actually two versions of this button - the full version and the condensed version. This page describes the condensed version. For information about the full version, see [fb:multi-friend-selector](#).

This is a more condensed version of an interface which must be used inside an [fb:request-form](#) to select some number of friends to send an invitation or request. This interface includes a series of `<input type="checkbox" name="ids[]" value="[friend id]">` which is included for selected users in the form that gets submitted to your `<fb:request-form>` action URL. For more information, check out [fb:request-form](#).

There are two versions of the condensed multi-friend-selector - a one box version and a two box version. With `selected_rows` set to 0, the one box version appears; it works like a normal check box where clicking a user selects that user. With `selected_rows` set to anything larger, the two box version is used; this results in an upper list containing unselected users and a lower list containing selected users.

The condensed multi-friend-selector behaves like a normal block level element, which is to say that you can style it with a style attribute or by giving it a CSS class, and its width stretches to fit into the containing block.

Attributes

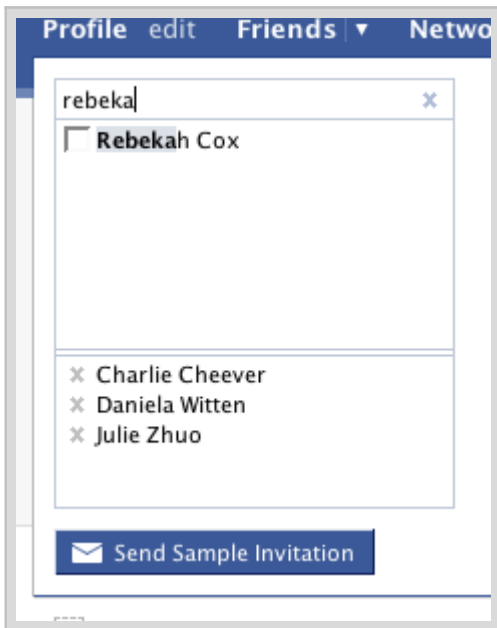
Required	Name	Type	Description
required	<code>condensed</code>	bool	Set this to true to use the condensed version of the multi-friend-selector. If not set, the full fb:multi-friend-selector is used.
optional	<code>max</code>	int	The maximum number of users that can be selected. This value ranges from 1 to 35, and is capped at the number of friend requests the user has remaining under their limit. This attribute is ignored if it is greater than the number of requests your application is able to send.
	<code>exclude_ids</code>	array	A comma-separated list of user IDs to exclude from the condensed multi-friend-selector.
	<code>unselected_rows</code>	int	The number of rows of friends to display in the unselected part of the condensed multi-friend-selector. <i>(default value is 6, and must be a number between 4 and 15)</i>
	<code>selected_rows</code>	int	The number of rows of friends to display in the selected part of the condensed multi-friend-selector. <i>(default value is 5, and must be a number between 5 and 15; or set it to 0 to indicate that you want only a single box for both selected and unselected friends)</i>

Post Variables

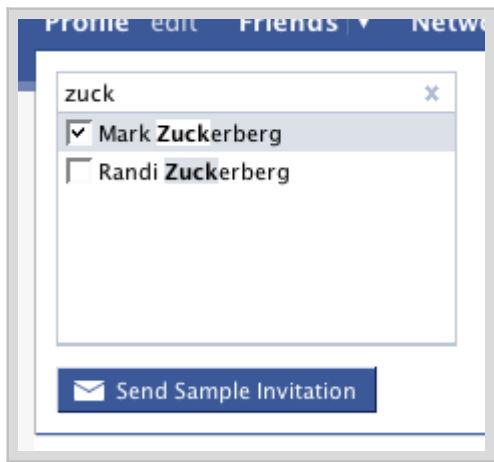
Required	Name	Type	Description
	<code>ids</code>	array	An array of the user IDs chosen by the user.

Examples

```
<div style="padding: 10px;">
  <fb:request-form method="post" action="index.php" content="hey" type="sample" invite="true">
    <div class="clearfix" style="padding-bottom: 10px;">
      <fb:multi-friend-selector condensed="true" style="width: 200px;" />
    </div>
    <fb:request-form-submit />
  </fb:request-form>
</div>
```



```
<div style="padding: 10px;">
  <fb:request-form method="post" action="index.php" content="hey" type="sample" invite="true">
    <div class="clearfix" style="padding-bottom: 10px;">
      <fb:multi-friend-selector condensed="true" selected_rows="0" style="width: 200px;" />
    </div>
    <fb:request-form-submit />
  </fb:request-form>
</div>
```



Fb:request-form-submit

Description

Creates a button that submits an [fb:request-form](#). Use this along with any combination of [fb:multi-friend-input](#) or [fb:friend-selector](#) tags inside an [fb:request-form](#) if you would like to have the user send a request or invitation. When this button is clicked, a confirmation dialog appears that allows the user to confirm the sending of the request or invitations.

You can also use a single [user ID](#) as the label for the button. This means that instead of using other elements in the [fb:request-form](#), the button results in a request being sent to the user that you specify for the label.

Note: When using a user ID as the button label, the user ID must belong to a friend of the current user. Using either the user ID of the current user or a user ID that does not belong to a friend of the current user results in a dialog with a message of "Please select some friends first" when the button is clicked. ([see bug](#))

Attributes

Required	Name	Type	Description
optional	uid	int	Set this to the user ID of a person you would like a request to be sent to.
	label	string	If you are using uid, use this tag to specify the text you want to appear as the label for the button. The text must include "%n" or "%N" which gets replaced with the first name or full name for the user ID, respectively.

Examples

```
<fb:request-form-submit />
```

Fb:application-name

Description

Renders the application name. Use this tag to render the name of an application. This can also be used when you cannot use the application name directly, e.g. to include the application name in a Mini-Feed or News Feed story when that application contains the word "message".

Examples

```
<fb:application-name>  
  MessageMe!  
</fb:application-name>
```

Status messages

Fb:error

Description

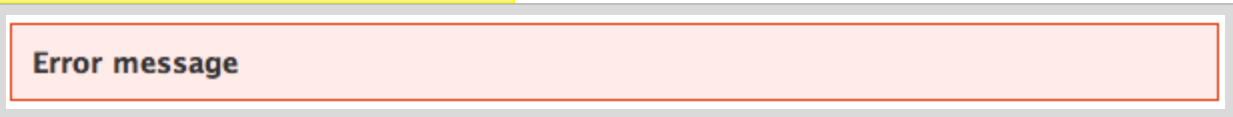
Renders a standard Facebook error message. The content within the <fb:error> tags is used as the error message text.

Attributes

Required	Name	Type	Description
required	message	string	The heading text to display in the error box. Alternately, you can specify the heading text using an fb:message child.
optional	decoration	string	Customize the appearance of the error by choosing among two styles: no_padding - Removes the 20 pixels of padding that surrounds the error message by default. shorten - Removes the 20 pixels of padding from the bottom of the error message.

Examples

```
<fb:error message="Error message" />
```



```
<fb:error>
  <fb:message>Error message</fb:message>
  This is the error message text.
</fb:error>
```



Notes

The error must contain at least one [fb:message](#) child or the message attribute as the heading text.

See Also

You can see a real world example of this tag in the Who's Showing Up demo application. Go to [Demos](#) for more details and to download and install the demo apps

Fb:explanation

Description

Renders a standard Facebook explanation message. The content contained within the <fb:explanation> tags is used as the explanation text.

Attributes

Required	Name	Type	Description
required	message	string	The heading text to display in the explanation box. Alternately, you can specify the heading text using an fb:message child.
optional	decoration	string	Customize the appearance of the explanation by choosing from two styles: no_padding - Removes the 20 pixels of padding that surrounds the explanation by default. shorten - Removes the 20 pixels of padding from the bottom of the explanation.

Examples

```
<fb:explanation message="Explanation message" />
```

Explanation message

```
<fb:explanation>
  <fb:message>Explanation message</fb:message>
  This is the explanation message text.
</fb:explanation>
```

Explanation message

This is the explanation message text.

Notes

The explanation must contain at least one [fb:message](#) child or the message attribute as the heading text.

Fb:success

Description

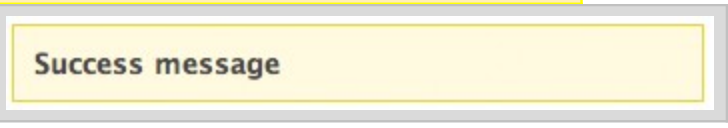
Renders a standard Facebook success message. The content within the <fb:success> tags is used as the success explanation.

Attributes

Required	Name	Type	Description
required	message	string	The heading text to display in the success box. Alternately, you can specify the heading text using an fb:message child.
optional	decoration	string	Customize the appearance of the success message box by choosing among two styles: no_padding - Removes the 20 pixels of padding that surrounds the message by default. shorten - Removes the 20 pixels of padding from the bottom of the message.

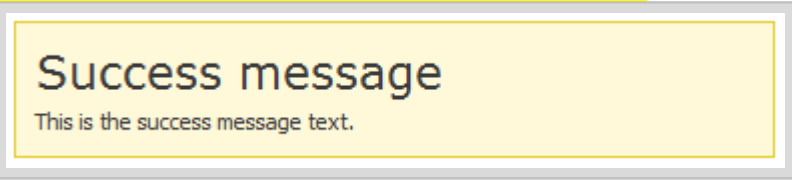
Examples

```
<fb:success message="Success message" />
```



Success message

```
<fb:success>
  <fb:message>Success message</fb:message>
  This is the success message text.
</fb:success>
```



Success message

This is the success message text.

Notes

The message must contain at least one [fb:message](#) child or the message attribute as the heading text.

See Also

You can see a real world example of this tag in the Restaurants and Who's Showing Up demo applications. Go to [Demos](#) for more details and to download and install the demo apps.

Fb:message

Description

Renders the heading text for an error, explanation or success message.

Examples

```
fb:message>  
  This is the heading text for the message.  
</fb:message>
```

Notes

You must use this tag inside [fb:error](#), [fb:explanation](#), or [fb:success](#).

Fb:editor

Description

Creates a form with two columns, just like the form on the edit-profile page. The children of fb:editor specify the rows of the form. For example, an [fb:editor-text](#) child adds a row with a text field in the right column. The label attribute of the fb:editor-* child specifies what text appears in the left column of that row.

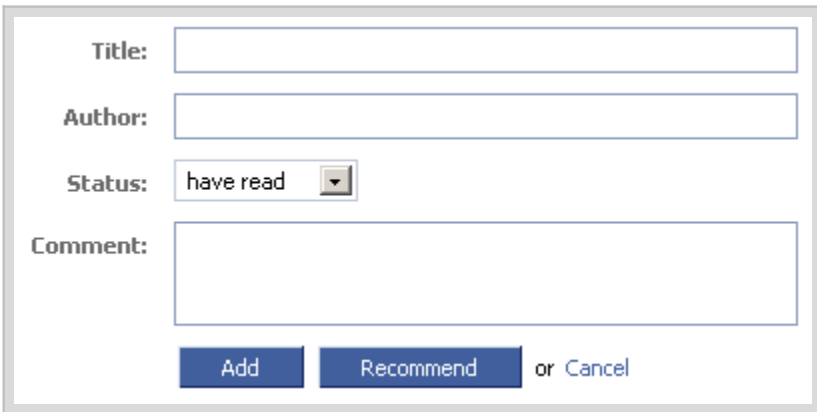
Warning: This form is submitted via POST method.

Attributes

Required	Name	Type	Description
required	action	string	The URL to which the form's data is posted.
optional	width	int	The width of the form/table, in pixels. <i>(default value is 425)</i>
	labelwidth	int	The width of the first column of the form/table, in pixels. <i>(default value is 75)</i> . Note: This value cannot be 0 as it is ignored; use 1 instead.

Examples

```
<fb:editor action="?do-it" labelwidth="100">
  <fb:editor-text label="Title" name="title" value=""/>
  <fb:editor-text label="Author" name="author" value=""/>
  <fb:editor-custom label="Status">
    <select name="state">
      <option value="0" selected>have read</option>
      <option value="1">am reading</option>
      <option value="2">want to read</option>
    </select>
  </fb:editor-custom>
  <fb:editor-textarea label="Comment" name="comment"/>
  <fb:editor-buttonset>
    <fb:editor-button value="Add"/>
    <fb:editor-button value="Recommend"/>
    <fb:editor-cancel />
  </fb:editor-buttonset>
</fb:editor>
```



The image shows a screenshot of a Facebook editor form. It contains the following elements:

- Title:** A text input field.
- Author:** A text input field.
- Status:** A dropdown menu with the text "have read" and a downward arrow.
- Comment:** A large text area for entering a comment.
- Buttons:** At the bottom, there are three buttons: "Add", "Recommend", and "or Cancel".

Notes

Mock AJAX There is currently no way to implement [Mock AJAX](#) using fb:editor. In order to use Mock AJAX, you must mimic the HTML equivalent of fb:editor and combine it with Mock AJAX as described at [\[1\]](#).

This tag may contain any of the following tags:

- [fb:editor-text](#)
- [fb:editor-textarea](#)
- [fb:editor-time](#)
- [fb:editor-month](#)
- [fb:editor-date](#)
- [fb:editor-divider](#)
- [fb:editor-buttonset](#)
- [fb:editor-button](#)
- [fb:editor-cancel](#)
- [fb:editor-custom](#)

Fb:editor-button

Description

Renders a button of type submit inside an [fb:editor](#) tag.

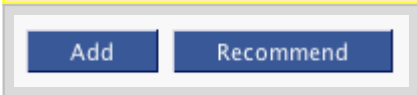
This tag can be a child of an [fb:editor-buttonset](#) container to render multiple buttons next to each other. If the button is not in a fb:editor-buttonset, the button still renders, but is not styled as an editor-button. [\[possible bug?\]](#)

Attributes

Required	Name	Type	Description
required	value	string	The text label for the button.
optional	name	string	The variable name that is sent in the POST request when the form is submitted.

Examples

```
<fb:editor>
  <fb:editor-buttonset>
    <fb:editor-button value="Add" name="add"/>
    <fb:editor-button value="Recommend" name="recommend"/>
  </fb:editor-buttonset>
</fb:editor>
```



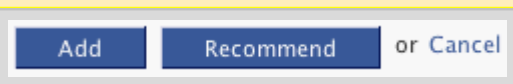
Fb:editor-buttonset

Description

A container for one or more [fb:editor-button](#) tags, which are rendered next to each other with some space between each button.

Examples

```
<fb:editor-buttonset>
  <fb:editor-button value="Add"/>
  <fb:editor-button value="Recommend"/>
  <fb:editor-cancel />
</fb:editor-buttonset>
```



Notes

This tag must contain at least one [fb:editor-button](#) as a child.

Fb:editor-cancel

Description

Renders a Cancel button inside an [fb:editor](#) tag.

Attributes

Required	Name	Type	Description
optional	value	string	The caption for the button. <i>(default value is "Cancel")</i>
	href	string	The URL to redirect to upon clicking. <i>(default value is "#")</i> This doesn't actually cancel anything, so you should set the href to at least reload the page.

Examples

```
<fb:editor>  
  <fb:editor-cancel value="Cancel This" href="http://www.somesite.com" />  
</fb:editor>
```

Fb:editor-custom

Description

Allows you to put any content into an [fb:editor](#) block, as long as it is valid FBML.

Attributes

Required	Name	Type	Description
optional	label	string	The label text for left hand side.
	id	string	The identifier tag for fb:editor.

Examples

This creates a drop down selector.

```
<fb:editor-custom label="Status">
  <select name="state">
    <option value="0" selected>have read</option>
    <option value="1">am reading</option>
    <option value="2">want to read</option>
  </select>
</fb:editor-custom>
```

You can also put regular text into the form.

```
<fb:editor-custom>
  Here is text on the right side.
</fb:editor-custom>
```

Notes

Password fields are currently supported.

Fb:editor-date

Description

Creates two drop down list boxes that let a user select a date. The month is listed using a three-letter abbreviation, and the day is listed as a numerical digit from 1 to 31 (with no leading zeroes). You can only use one date selector per page.

Attributes

Required	Name	Type	Description
required	label	string	The label to display to the left of both selectors.
optional	value	int	The Unix timestamp of the date to display when the page loads. (<i>default value is Dec 31</i>)

Post Variables

Required	Name	Type	Description
	date_month	int	A value in the range of [1-12] representing the selected month.
	date_day	int	A value in the range of [1-31] representing the selected day.

Examples

```
<fb:editor-date label="Date:" />
```

The screenshot shows a form with three input fields. The first field is labeled 'Title:' and is empty. The second field is labeled 'Date:' and contains two dropdown menus; the first is set to 'Jan' and the second is set to '1'. The third field is labeled 'scription' and is empty.

Notes

Using more than one fb:editor-date results in only the last date object being submitted in the postback. You need to validate the data this control provides as it can generate illegal date combinations such as February 31.

If you post back to the same page, then the previous selection does not persist (appearing as if a validation error occurred).

You cannot apply a name attribute to this tag; otherwise, no data is returned in the postback.

If you need more than one date selector, use [fb:editor-custom](#) and insert the markup yourself.

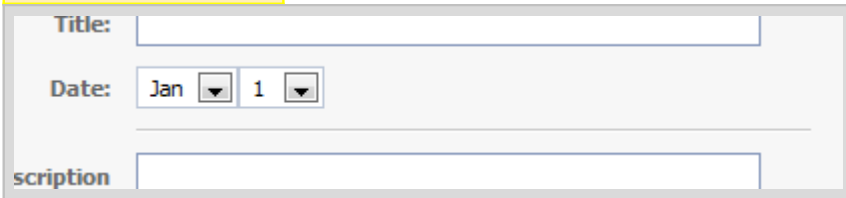
Fb:editor-divider

Description

Renders a horizontal line separator in the column containing the form elements. Despite appearing like an `<hr>` element, it is actually a `<div>` with a class of `divider`.

Examples

```
<fb:editor-divider/>
```



The screenshot shows a form with a light gray background and a thin border. It contains three input fields: a text field labeled "Title:", a date field labeled "Date:" with dropdown menus for "Jan" and "1", and a text field labeled "scription". A horizontal line separator is positioned between the "Date:" field and the "scription" field.

Fb:editor-month

Description

Creates a form selector element displaying the month.

This article or section needs expansion

Attributes

Required	Name	Type	Description
optional	name	string	The name for the control. <i>(default value is month)</i>
	value	int	The value to pre-fill (the number of the month, 1 for Jan, 2 for Feb, etc). <i>(default value is "Month:")</i>

Post Variables

Required	Name	Type	Description
	[name]	int	The value specified for the name parameter in the tag), represented by the numbers 1-12, corresponding to the appropriate month, or -1 if nothing is selected.

Examples

```
<fb:editor action="?do-it" labelwidth="100">
  <fb:editor-month value="3" name="mon"/>
  <fb:editor-buttonset>
    <fb:editor-button value="Add"/>
    <fb:editor-button value="Recommend"/>
    <fb:editor-cancel />
  </fb:editor-buttonset>
</fb:editor>
```

Fb:editor-text

Description

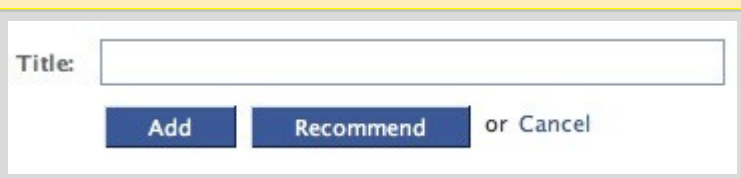
Creates an <input> of type text.

Attributes

Required	Name	Type	Description
optional	label	string	The label to display on the left side of the text box.
	name	string	The name for the control.
	value	string	The default text that populates the text box.
	maxlength	int	The maximum length of the input allowed in the text box.

Examples

```
<fb:editor action="?do-it" labelwidth="100">  
  <fb:editor-text label="Title" name="title" value=""/>  
  <fb:editor-buttonset>  
    <fb:editor-button value="Add"/>  
    <fb:editor-button value="Recommend"/>  
    <fb:editor-cancel />  
  </fb:editor-buttonset>  
</fb:editor>
```



The screenshot shows a rendered Facebook editor component. It features a text input field with the label "Title:" to its left. Below the input field, there are three buttons: "Add", "Recommend", and "Cancel". The "Add" and "Recommend" buttons are blue with white text, while "Cancel" is a plain text label. The entire editor is enclosed in a light gray border.

Notes

This tag must be included within [Fb:editor](#).

Fb:editor-textarea

Description

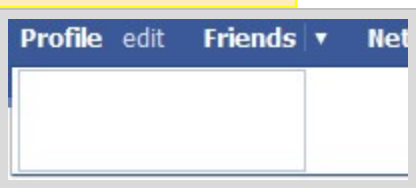
Creates a <textarea> element.

Attributes

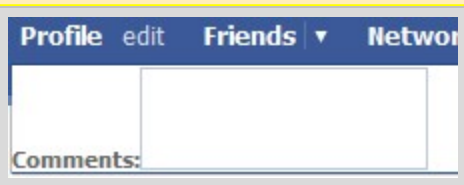
Required	Name	Type	Description
optional	label	string	The label to display on the left side of the text box.
	name	string	The name of the field that is passed when the form is submitted.
	rows	int	The height of the text area in number of lines of text. This is identical to the HTML textarea tag.

Examples

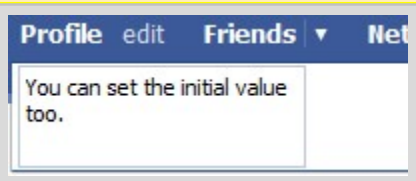
```
<fb:editor-textarea />
```



```
<fb:editor-textarea label="Comments" />
```



```
<fb:editor-textarea>
  You can set the initial value too.
</fb:editor-textarea>
```



Notes

While this field is not a required child of [fb:editor](#), it looks best^{[[who says?](#)]} when it is.

Fb:editor-time

Description

Creates a series of form selector elements showing the time in hours and minutes, and an AM/PM indicator.

Attributes

Required	Name	Type	Description
optional	label	string	The label to display on the left side of the selector. <i>(default value is blank)</i>
	name	string	The name of the field that is passed when the form is submitted. <i>(default value is "time")</i>
	value	int	The default value in epoch seconds with which to populate the selector.

Examples

```
<fb:editor action="?do-it" labelwidth="100">
  <fb:editor-time value="1185930724" name="time"/>
  <fb:editor-buttonset>
    <fb:editor-button value="Add"/>
    <fb:editor-button value="Recommend"/>
    <fb:editor-cancel />
  </fb:editor-buttonset>
</fb:editor>
```

The screenshot shows a time selector interface. It consists of three vertically stacked dropdown menus. The first dropdown shows '6', the second shows '00', and the third shows 'pm'. Below these dropdowns are three buttons: 'Add', 'Recommend', and 'Cancel'. The 'Add' and 'Recommend' buttons are blue, while 'Cancel' is a lighter blue. The text 'or' is placed between 'Recommend' and 'Cancel'.

Notes

This tag returns three values, *name_hour*, *name_min* and *name_ampm* where *name* is the name for the control.

The time value that you enter is rounded to the nearest 15 minutes. Also, this selector displays only the time, **not** the time and date.

Page navigation

Fb:dashboard

Description

Renders a standard Facebook dashboard header. Dashboards can contain the following elements:

[fb:action](#)

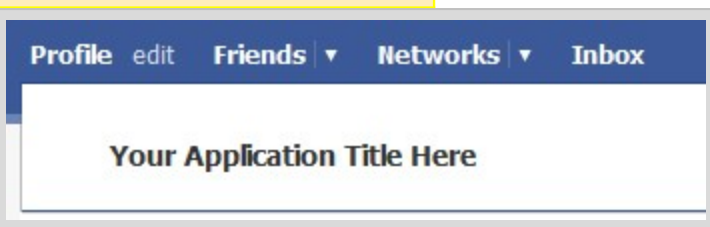
[fb:create-button](#)

[fb:help](#)

Attributes

Examples

```
<fb:dashboard></fb:dashboard>
```



Notes

For an application with an icon, the icon appears next to the application title.

You cannot include tags like [fb:if-user-has-added-app](#) inside a dashboard. You can do the equivalent as illustrated below to create a dashboard that determines when a user has or has not added an application. This snippet mixes in Smarty code. While you can refactor this code, the detail is included for those unfamiliar with Smarty.

```
<fb:if-user-has-added-app>
{if $isOwnSchedule == 'true'}
    <fb:dashboard>
        {include file='header-actions-internal.tpl'}
    </fb:dashboard>
<fb:else>
    <fb:dashboard>
        {include file='header-actions-internal.tpl'}
        <fb:action href="http://apps.facebook.com/add.php?api_key=">Add this app</fb:action>
    </fb:dashboard>
{else}
    <fb:dashboard>
        {include file='header-actions-external.tpl'}
```

```
        </fb:dashboard>
    <fb:else>
        <fb:dashboard>
            {include file='header-actions-external.tpl'}
            <fb:action href="http://apps.facebook.com/add.php?api_key=">Add this app</fb:action>
        </fb:dashboard>
    {/if}
</fb:else>
</fb:if-user-has-added-app>
```

See Also

You can see a real world example of this tag in the Restaurants demo application. Go to [Demos](#) for more details and to download and install the demo apps.

Fb:action

Description

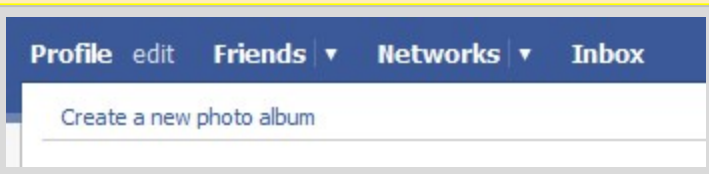
Renders a link, usually for navigational purposes. Its appearance depends on its container. The tag **must** be a child of either [fb:dashboard](#) or [fb:subtitle](#).

Attributes

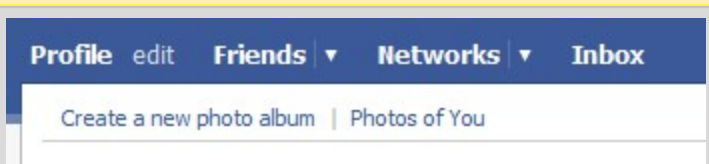
Required	Name	Type	Description
required	href	string	The URL for the link. The URL must be a canvas page. For example, href="http://apps.facebook.com/<appname>/<filename>.php".
optional	title	string	Specifies the text to display as a tool tip for this link. (<i>seems to do nothing at the moment</i>)
	onclick	string	Call a FBJS function

Examples

```
<fb:dashboard>
  <fb:action href="new.php">Create a new photo album</fb:action>
</fb:dashboard>
```



```
<fb:dashboard>
  <fb:action href="new.php">Create a new photo album</fb:action>
  <fb:action href="you.php">Photos of You</fb:action>
</fb:dashboard>
```



This article or section needs expansion

Notes

The link appears in the top left of the `fb:dashboard` element and the top right of the `fb:subtitle` element. ~~You cannot use FBJS onclick, since you cannot have the href attribute set to # or JavaScript.~~ You *can* use the onclick attribute: use any absolute URL in the href and make sure onclick returns false to prevent the link from being followed.

See Also

You can see a real world example of this tag in the Who's Showing Up demo application. Go to [Demos](#) for more details and to download and install the demo apps.

Fb:create-button

Description

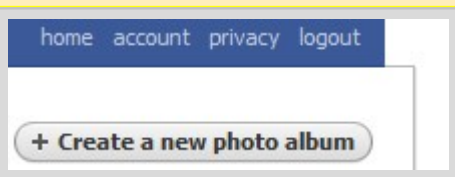
Renders a Create button for adding user-generated content. This tag must be a child of [fb:dashboard](#).

Attributes

Required	Name	Type	Description
required	href	string	The URL where the link for the button takes the user. <i>The URL must be a canvas page.</i>
optional	title	string	Specifies the text to display as a tool tip for this link.
	onclick	string	Call a FBJS function

Examples

```
<fb:dashboard>
  <fb:create-button href="new.php">Create a new photo album</fb:create-button>
</fb:dashboard>
```



Notes

- This button appears in the top right of the [fb:dashboard](#) element, next to the application title.
- Each page can contain only one Create button.
- The (+) symbol appears on all buttons and cannot be altered.

Fb:help

Description

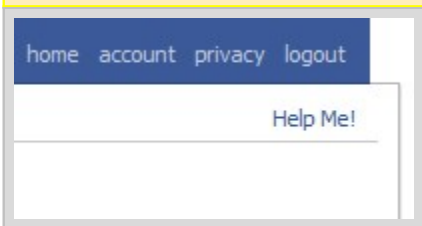
Renders a help link. This tag must be a child of [fb:dashboard](#).

Attributes

Required	Name	Type	Description
required	href	string	The URL for the help page.
optional	title	string	Specifies the text to display as a tool tip for this link.

Examples

```
<fb:dashboard>  
  <fb:help href="help.php">Help Me!</fb:help>  
</fb:dashboard>
```



Notes

This button appears in the top right of the [fb:dashboard](#) element.

Fb:header

Description

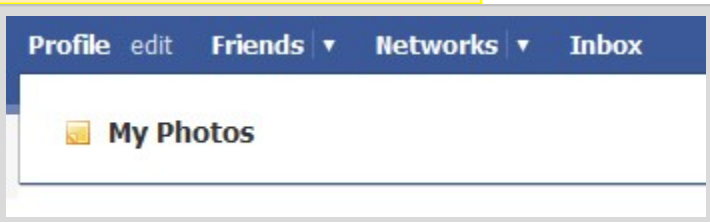
Renders a standard Facebook title header.

Attributes

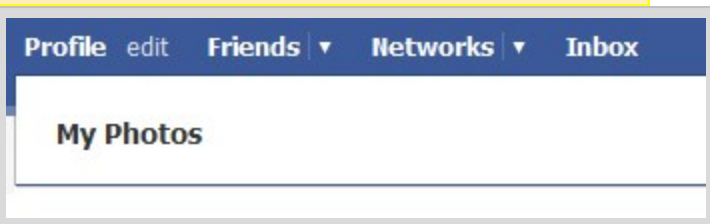
Required	Name	Type	Description
optional	icon	bool	Toggles whether the application icon is displayed. No other icon can be displayed. <i>(default value is true)</i>
	decoration	string	Customize the appearance of the title by choosing among three styles: add_border - Adds a 1px solid #ccc border to the bottom of the header. Useful for pages with gray backgrounds below the header. no_padding - Removes the 20px of padding that is on the header by default. shorten - Removes the 20px of padding from the bottom of the header.

Examples

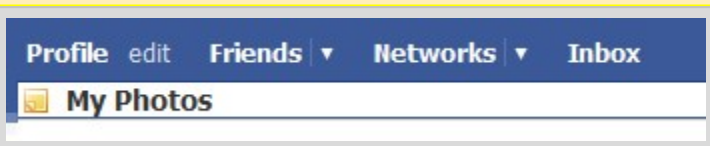
```
<fb:header>My Photos</fb:header>
```



```
<fb:header icon="false">My Photos</fb:header>
```



```
<fb:header decoration="no_padding">My Photos</fb:header>
```



See Also

You can see a real world example of this tag in the Restaurants demo application. Go to [Demos](#) for more details and to download and install the demo apps.

Fb:mediaheader

Description

Renders a standard media header, intended mainly for displaying content contributed by a particular user. The media header is shown at the top of **See All** pages throughout Facebook. It contains a photo of the media owner and links to actions on that user.

The links shown to the content owners must be specified using [fb:owner-action](#) tags. Links to non-owners are always **Profile**, **Send a Message** and **Poke**, subject to standard privacy controls.

This tag may contain:

[fb:header-title](#)

[fb:owner-action](#)

Examples

:This article or section needs expansion

Notes

This tag assumes it has the full width of the canvas to work with, otherwise it does not display correctly. For example, the media image appears above the links with a gray background. You can put the header inside a div with negative margins if you need to compensate.

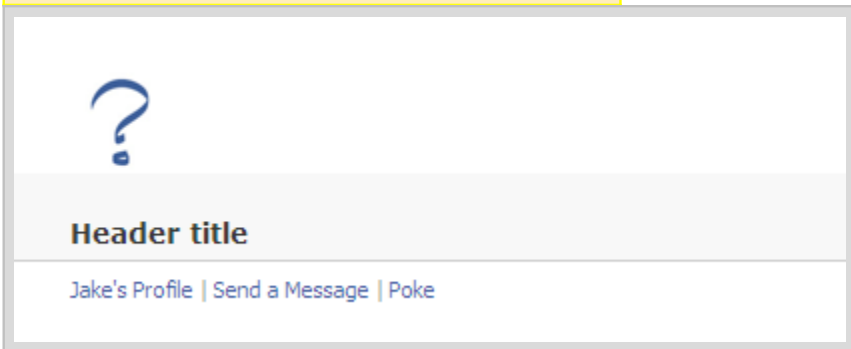
Fb:header-title

Description

Specifies the header title for a [fb:mediaheader](#).

Examples

```
<pre><fb:mediaheader uid="123456">
<fb:header-title>Header title</fb:header-title>
</fb:mediaheader></pre>
```



Fb:owner-action

Description

Specifies an action link to be displayed inside a [fb:mediaheader](#) when the viewer is the owner of the content

Attributes

Required	Name	Type	Description
required	href	string	the URL corresponding to the action.

Examples

```
<fb:mediaheader uid="123456">  
<fb:header-title>Content header</fb:header-title>  
<fb:owner-action href="http://google.com">Go to Google</fb:owner-action>  
</fb:mediaheader>
```

Notes

[fb:tabs](#) and [fb:tab-item](#) do not render correctly within the [FBML Test Console](#).

Requirements: The title of the link must be enclosed between `<fb:owner-action>` and `</fb:owner-action>` tags.

Fb:tabs

Description

Renders a group of standard Facebook navigation tabs. Must contain at least one [fb:tab-item](#).

Examples

```
<fb:tabs>  
<fb:tab-item href='http://apps.facebook.com/yourapp/myphotos.php' title='My Photos' selected='true'/>  
<fb:tab-item href='http://apps.facebook.com/yourapp/recentalbums.php' title='Albums recents' />  
</fb:tabs>
```



Fb:tab-item

Description

Renders a standard Facebook tab. Must always be a child of [fb:tabs](#).

Attributes

Required	Name	Type	Description
required	href	string	The URL to which the link should take the user upon click. <i>must be absolute</i>
	title	string	Specifies the text to display on the tab.
optional	align	string	Specify alignment for this tab item. (<i>default value is left</i>). Other valid value is right
	selected	bool	Indicates whether this tab item has the selected state. (<i>default value is false</i>)

Examples

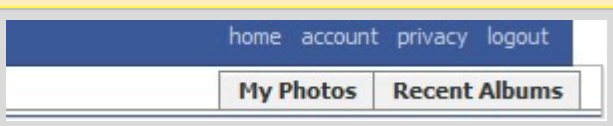
```
<fb:tabs>
  <fb:tab_item href="myphotos.php" title="My Photos" />
  <fb:tab_item href="recent.php" title="Recent Albums" />
</fb:tabs>
```



```
<fb:tabs>
  <fb:tab_item href="myphotos.php" title="My Photos" selected="true"/>
  <fb:tab_item href="recent.php" title="Recent Albums" />
</fb:tabs>
```



```
<fb:tabs>
  <fb:tab_item href="myphotos.php" title="My Photos" align="right"/>
  <fb:tab_item href="recent.php" title="Recent Albums" align="right"/>
</fb:tabs>
```



Notes

[fb:tabs](#) and **fb:tab-item** do not render correctly within the [FBML Test Console](#). Dynamic FBML, clicktohide, clicktoshow, etc., does not work with this FBML tag.

Dialog

Fb:dialog

Description

Important: The fb:dialog tag is in beta mode right now, which means there is a possibility it can change. When it is fully stable, Facebook will update the announcements page and remove this note. In the meantime, we would appreciate any feedback/issues you have about fb:dialog.

Bug Reports and Feature Requests: [Talk:fb:dialog](#)

The **fb:dialog** tag displays a standard popup dialog box when a user clicks on some element. The dialog box can then show some specified content or form. Clicking on the dialog buttons can post the form or use mock-ajax to display the results back into the dialog with the use of fb:dialogresponse.

The **fb:dialog** tag **must** contain:

[fb:dialog-content](#) - the FBML contents of your dialog. You can now style this section of the dialog.

The **fb:dialog** tag **may** contain:

[fb:dialog-title](#) - which displays a title on your dialog

[fb:dialog-button](#) - which adds buttons to the bottom of your dialog

[edit] Invoking the Dialog

To invoke the dialog, simply add the attribute clicktoshowdialog="<FB:DIALOG ID>" to any element.

Note: You can also use [FBJS](#) to create dialog boxes with greater flexibility than **fb:dialog**.

Attributes

Required	Name	Type	Description
required	id	string	The unique identifier for your dialog, which is used to invoke your dialog.
optional	cancel_button	bool	Indicates whether to display a Cancel button to close the dialog.

Examples

Sample Code with [Mock AJAX](#)

An [fb:dialog-button](#) should have the clickrewriteform, clickrewriteid and clickrewriteurl attributes. The clickrewriteid specified is the id attribute of fb:dialog.

```
<fb:dialog id="my_dialog">
  <fb:dialog-title>My Little Dialog</fb:dialog-title>
  <fb:dialog-content><form id="my_form">Do you like my little dialog?</form></fb:dialog-content>
  <fb:dialog-button type="submit" value="Yes" clickrewriteurl="http://www.someurl.com/response.php"
clickrewriteid="my_dialog" clickrewriteform="my_form" />
</fb:dialog>
```

I don't know what link to put, how about `this one?`

The contents of <http://www.someurl.com/response.php> should print out FBML wrapped in [fb:dialogresponse](#).

```

<fb:dialogresponse>
  <fb:dialog-title>My Little Dialog Part II</fb:dialog-title>
  <fb:dialog-content>I'm glad you like my dialog</fb:dialog-content>
  <fb:dialog-button type="button" value="Close" close_dialog=1 />
</fb:dialogresponse>

```

Sample Code with Form Submission

To have a [fb:dialog-button](#) submit a form upon click, you must specify a `form_id` attribute on the [fb:dialog-button](#).

```

<fb:dialog id="my_dialog" cancel_button=1>
  <fb:dialog-title>My Little Dialog</fb:dialog-title>
  <fb:dialog-content>Do you like my little dialog?
    <form id="my_form">
      <p/>
      <input type="radio" name="like_dialog" value="yes"/>Yes
      <br/>
      <input type="radio" name="like_dialog" value="no"/>No
      <p/>
    </form>
  </fb:dialog-content>
  <fb:dialog-button type="submit" value="Yes" form_id="my_form" />
</fb:dialog>

```

Would you like to see a [dialog?](#)

Sample Code with Two-Button Choice

```

<fb:dialog id="my_dialog" cancel_button=1>
  <fb:dialog-title>My Little Dialog</fb:dialog-title>
  <fb:dialog-content><form id="my_form">Do you like my little dialog?</form></fb:dialog-content>
  <fb:dialog-button type="button" value="Yes" href="http://some_url.com/awesome.php" />
  <fb:dialog-button type="button" value="No" href="http://some_url.com/lame.php" />
</fb:dialog>

```

Would you like to see a [dialog?](#)

Fb:dialog-title

Description

fb:dialog-title is a child of [fb:dialog](#) and renders a title for what is displayed inside the popup dialog. Requirements: must be inside a [fb:dialog](#) tag.

Examples

```
<fb:dialog id="my_dialog">
  <fb:dialog-title>My Little Dialog</fb:dialog-title>
  <fb:dialog-content><form id="my_form">Do you like my little dialog?</form></fb:dialog-content>
  <fb:dialog-button type="submit" value="Yes"
clickrewriteurl="http://www.some_url_here.com/dialog_return.php"
clickrewriteid="my_dialog" clickrewriteform="my_form" />
</fb:dialog>
```

Fb:dialog-content

Description

The fb:dialog-content tag is a child of [fb:dialog](#) and represents the content that gets displayed inside the popup dialog when it appears.

This section can be styled like any other part of your app.

Examples

```
<fb:dialog id="my_dialog">
<fb:dialog id="my_dialog">
  <fb:dialog-title>My Little Dialog</fb:dialog-title>
  <fb:dialog-content><form id="my_form">Do you like my little dialog?</form></fb:dialog-content>
  <fb:dialog-button type="submit" value="Yes"
clickrewriteurl="http://www.some_url_here.com/dialog_return.php" clickrewriteid="my_dialog"
clickrewriteform="my_form" />
</fb:dialog>
```

Fb:dialog-button

Description

Renders a button for the [fb:dialog](#) popup.

The fb:dialog-button tag is a child of [fb:dialog](#) and must be contained within it.

Attributes

Required	Name	Type	Description
required	type	string	The type of button. Specify button for a general button or submit for a Submit button to submit the form.
	value	string	The label text for the button.
optional	close_dialog	bool	Indicates whether to close the popup dialog when the user clicks the button.
	href	string	The URL where the user is taken after clicking the button.
	form_id	string	The ID passed with the form when it is submitted after the user clicks the button.
	clickrewriteurl	string	This attribute is used for mock-AJAX with the dialog. See Mock AJAX
	clickrewriteid	string	This attribute is used for mock-AJAX with the dialog. See Mock AJAX
	clickrewriteform	string	This attribute is used for mock-AJAX with the dialog. See Mock AJAX

Examples

```
<fb:dialog id="my_dialog">
  <fb:dialog-title>My Little Dialog</fb:dialog-title>
  <fb:dialog-content><form id="my_form">Do you like my little dialog?</form></fb:dialog-content>
  <fb:dialog-button type="submit" value="Yes"
clickrewriteurl="http://www.some_url_here.com/dialog_return.php"
clickrewriteid="my_dialog" clickrewriteform="my_form" />
</fb:dialog>
```

a screenshot is not applicable for this tag

There is no "cancel" type. To specify a Cancel/No button, use:
 <fb:dialog-button type="button" value="No" close_dialog="1" />

a screenshot is not applicable for this tag

Notes

You can have multiple buttons in a dialog.

If you get the type wrong, the button doesn't appear at all. Facebook does not draw it.

Wall

Fb:wall

Description

Emulates a wall environment.

Examples

```
<fb:wall>  
<fb:wallpost uid="12345">This is a wall post from uid 12345</fb:wallpost>  
</fb:wall>
```

See Also

[fb:wallpost](#)

Fb:wallpost

Description

Renders a wall-style post. You should use it inside [fb:wall](#) tags, but it renders fine without them. It can also contain an [fb:wallpost-action](#) tag, which places a link at the bottom of the post.

Attributes

Required Name Type Description

required	uid	int	The user ID of the author of the post. FBML cannot parse without it, resulting in a runtime error. If the user ID is invalid (for example, the account has been deleted) then an image of a question mark and no name appear.
optional	t	int	The current time, which is displayed in epoch seconds. <i>(default value is empty, no time/date is displayed with the wall post)</i>

Examples

```
<fb:wallpost uid="10" t="1180502413">
```

This tag exists outside <fb:wall> tags.

```
<br />
```

It has the t attribute set to 1180502413

```
</fb:wallpost>
```

```
<fb:wall>
```

```
<fb:wallpost uid="11">
```

This tag exists inside <fb:wall> tags.

It has no t attribute set

```
</fb:wallpost>
```

```
</fb:wall>
```



Marcel Laverdet wrote
at 10:20pm on May 29

This tag exists outside <fb:wall> tags.
It has the t attribute set to 1180502413

Soleio wrote

This tag exists inside <fb:wall> tags. It has no t attribute set

Fb:wallpost-action

Description

Displays a link at the bottom of a wallpost (even if it appears before other text within the [fb:wallpost](#) tag).

Attributes

Required	Name	Type	Description
required	href		the URL of the link. must be absolute

Examples

```
<fb:wall/>
```

```
<fb:wall>  
<fb:wallpost uid="12345">This is a wall post from uid 12345.  
<fb:wallpost-action href="linktohere">This is a small blue link in the post</fb:wallpost-action>  
</fb:wallpost>  
</fb:wall>
```

Dynamic FBML attributes

Summary

[Visibility](#) (clicktoshow, clicktohide,clicktotoggle)

[Mock AJAX](#) (clickrewriteid, clickrewriteurl,clickrewriteform)

[Forms](#) (clicktoenable, clicktodisable,clickthrough)

[Requirelogin](#)

[FBSJS](#)

Details

DynamicFBML/Visibility

Description

These elements all control the visibility of an element or a set of elements.

Format is clickto*="id1,id2,id3".

[clicktoshow](#) - Sets the specified IDs to visible.

[clicktohide](#) - Hides the specified IDs

[clicktotoggle](#) - Will flip the visibility of the specified elements.

[clickthrough](#) - If true, allows checkboxes to change state, buttons to submit forms, etc.; if false, normal behavior of the element is inhibited.

Examples

A live demonstration can be found [here](#) (Facebook login required, app login is not)

```
<a id="show_text" href="#" clicktohide="show_text" clicktoshow="History,hide_text">
Show History</a>

<a id="hide_text" href="#" clicktohide="History, hide_text" clicktoshow="show_text"
style="display:none">Hide History</a>

<div id="History" style="display:none">
THE STUFF TO DISPLAY
</div>
```

Notes

You can create elements that are hidden by default by giving them a style of display: none;. Note that it must be inline, so attach *style="display:none;"* to your targeted tag, and don't declare it within an externally called CSS file. It won't be switched on otherwise.

To use clicktohide and clicktoshow in a form with input type="radio" or input type="checkbox", make sure you set clickthrough="true" as well otherwise the selection will not change when the user clicks on them.

When used on a profile box or on any scrollable page, clicking the links as shown above will cause the scroll position to jump to the top of the page (which is where the unnamed "#" bookmark is considered

to be.) To prevent this from happening, use a bookmark name that does not exist, such as href="#nonexistantbookmark".

DynamicFBML/MockAJAX

Attributes

clickrewriteid - This is the id of the div that will be replaced by the returned data.

clickrewriteurl - This is the (non-canvas) url from which it will grab the data.

clickrewriteform - This is the form to use when submitting.

Usage examples Sample code, with more details, is available at [Mock Ajax](#) page.

Examples

A live demonstration can be found [here](#). (Facebook login required, app login not required) (and it has no explanation, can some be provided please?)

Slightly more explanation can be found [in the old documentation](#).

More discussion found here: <http://utexas.facebook.com/topic.php?uid=2205007948&topic=6584>

Notes

Make the calls directly to your application, i.e. <http://myapp.com/url> **NOT** to <http://apps.facebook.com/myapp/url> . If you call apps.facebook.com you'll get the element removed, because facebook returns a login page, not valid FBML.

Note: The script echoing the value to be displayed by a mockAJAX call **must** be on the callback page (as specified in your app settings). Any other page will fail.

One useful tip in debugging this: You can inspect the JS object FBML.mockAjaxResponse. Use the Firebug Firefox extension to access this.

Some of the things contained in it:

ok - is a boolean telling whether the mock AJAX call was successful. True means everything went OK.

False means something went wrong.

error_code - is the HTTP response code if there is a problem, like 404 if not found, etc.

error_message - is the text of the error message if there was an error.

html - is the rendered HTML that was returned back that will be inserted into the DIV

These only show up if you are the developer of the app:

parse_errors - parse errors in turning the FBML into HTML

render_errors - errors rendering the FBML into HTML

fbml - the raw FBML returned from your URL

url - the URL the FBML was fetched from

Retrieved from "<http://wiki.developers.facebook.com/index.php/DynamicFBML/MockAJAX>"

See Also

DynamicFBML/Forms

Attributes

clicktoenable - Will enable the form elements specified by id.

clicktodisable - Will disable the form elements specified by id.

clickthrough - Will allow clicks on elements using Dynamic FBML to go through to the element.

Examples

```
<form>
  <input type="text" id="firstname" name="firstname" />
  <input id="sendbutton" type="submit" value="Submit"
    clicktodisable="firstname" clicktohide="sendbutton" clickthrough="true" />
</form>
```

Notes

Combining clickthrough and clicktodisable does not allow a form to submit in Safari 3, bug has been submitted to Safari

Disabled form elements do not submit values.

Multiple ids should be separated by commas.

Requirelogin

Description

You can add the attribute requirelogin to anchor tags to ensure that the user will be logged in before accessing the anchor link.

A user who is not logged in to your application and clicks on a link with requirelogin=1 will see a popup dialog prompting them to log into your application. A user who is already logged in to your application (has a valid session) will be able to follow through to the link as they normally would.

Examples

```
<a href="http://www.somewheretogo.com/loggedinusersonly.php" requirelogin=1>  
You must be logged in to follow me</a>
```

FBJS

Description

FBJS is Facebook's solution for developers who want to use JavaScript in their Facebook applications. We built FBJS to empower developers with all the functionality they need, and to protect our users' privacy at the same time.

How It Works

Most providers who allow developers to embed JavaScript within their domain force developers to use iframes to sandbox their code. Facebook has taken a different approach to this problem. JavaScript that you give us gets parsed, and any identifiers (function and variable names) get prepended with your application ID. For example, the following code block:

```
function foo(bar) {  
  var obj = {property: bar};  
  return obj.property;  
}
```

becomes:

```
function a12345_foo(a12345_bar) {  
  var a12345_obj = {property: a12345_bar};  
  return a12345_obj.property;  
}
```

This creates a virtual scope for every application that runs within Facebook. From there we expose certain functionality through a collection of JavaScript objects that allow you to modify your content on Facebook. Our objects are made to mimic the functionality of JavaScript as closely as possible, but it may take some getting used to for people who are already adept with JavaScript.

The Basics

The JavaScript syntax you've come to know and love (or hate) is exactly the same. You can create objects, use anonymous functions, create timeouts and almost any other thing you can think of. Modifying the DOM tree is slightly different, however.

Take this example FBML code, for instance:

```
<a href="#" onclick="hello_world(this); return false;">Hello World!</a>  
<script>  
<!--  
function random_int(lo, hi) {  
  return Math.floor((Math.random() * (hi - lo)) + lo);  
}  
  
function hello_world(obj) {  
  var r = random_int(0, 255), b = random_int(0, 255), g = random_int(0, 255);  
  var color = r+', '+g+', '+b;  
  obj.setStyle('color', 'rgb('+color+')');  
}
```

```
//-->
</script>
```

As you can see, creating FBJS is very similar to JavaScript. Note, however, that this example may not work as expected:

```
<a href="#" id="hello">Hello World!</a>
<script>
<!--
function random_int(lo, hi) {
  return Math.floor((Math.random() * (hi - lo)) + lo);
}

function hello_world(obj) {
  var r = random_int(0, 255), b = random_int(0, 255), g = random_int(0, 255);
  var color = r+', '+g+', '+b;
  obj.setStyle('color', 'rgb('+color+')');
}

hello_world(document.getElementById('hello'));
//-->
</script>
```

In profile boxes, inline scripts are deferred until the first "active" event is triggered by a user. An active event is considered either onfocus, onclick, onmousedown, and so forth. Basically anything that requires a mouse click is an "active" event. On a canvas page, however, this example works just fine.

Also, please note it's very important that you use the syntax in the example above with your `<script>` tag hugging HTML comments. Otherwise `<`'s will be stripped out which makes coding very difficult ;). In the future we plan to modify our FBML parser to accept FBJS code without HTML comment wrappers, but for now it's required. -- Seems this is no longer true and you should be fine without them (though it doesn't hurt to include them and they'll be removed by the FBJS parser anyway).

FBJS DOM Objects

Retrieving Objects

A handle to an FBJS DOM object can be retrieved by either calling `document.getElementById`, or `document.createElement`. Additionally, the "this" pointer in DOM events also points to the target of the event.

Manipulating Objects

FBJS DOM objects implement most of the same methods regular JavaScript objects implement including: `appendChild`, `insertBefore`, `removeChild`, and `cloneNode`. Properties like `parentNode`, `nextSibling`, `src`, `href` (and many others) have been redefined as a couplet of getters and setters.

Instead of `obj.parentNode` just call `obj.getParentNode()`, and so on. Most of the properties are easy to figure out, but here's an exhaustive list of properties in JavaScript and how they translate to FBJS:

JavaScript	FBJS getter	FBJS setter	Description
<code>parentNode</code>	<code>getParentNode</code>		
<code>nextSibling</code>	<code>getNextSibling</code>		
<code>previousSibling</code>	<code>getPreviousSibling</code>		
<code>firstChild</code>	<code>getFirstChild</code>		

lastChild	getLastChild		
childNodes	getChildNodes		Returns a snapshot array of childNodes
innerHTML	n/a	setInnerFBML	Note that this can throw an error if you pass a string directly. Use Fb.js-string to create the string first then pass that variable.
innerHTML	n/a	setInnerXHTML	Beta feature. Allows you to set the innerHTML of an element by passing in a string of XHTML. The XHTML is sanitized according to FBML rules and then placed into the document.
innerText/textContent	n/a	setTextValue	Not exactly like setInnerFBML as this will only allow text (no HTML)! It will remove all childNodes of the element it is called on.
form	getForm		Doesn't work, use <code>document.getElementById('formid')</code> instead
action	getAction	setAction	
value	getValue	setValue	
href	getHref	setHref	
target	getTarget	setTarget	
src	getSrc	setSrc	
className	getClassName	setClassName	
tagName	getTagName		
id	getId	setId	
dir	getDir	setDir	
checked	getChecked	setChecked	
clientWidth	getClientWidth		
clientHeight	getClientHeight		
offsetWidth	getOffsetWidth		
offsetHeight	getOffsetHeight		
n/a	getAbsoluteTop		Returns the elements absolute position relative to the top of the page. Useful because of lack of <code>offsetParent</code> support.
n/a	getAbsoluteLeft		Same as <code>getAbsoluteTop</code> , but horizontally.
scrollTop	getScrollTop	setScrollTop	
scrollLeft	getScrollLeft	setScrollLeft	
scrollHeight	getScrollHeight		
scrollWidth	getScrollWidth		
tabIndex	getTabIndex	setTabIndex	
title	getTitle	setTitle	
name	getName	setName	
cols	getCols	setCols	
rows	getRows	setRows	
accessKey	getAccessKey	setAccessKey	
disabled	getDisabled	setDisabled	
readOnly	getReadOnly	setReadOnly	

type	getType	setType	
selectedIndex	getSelectedIndex	setSelectedIndex	
selected	getSelected	setSelected	
location	n/a	setLocation	
style	getStyle	setStyle	
n/a	getRootElement		used as document.getRootElement - returns the top-level element of your profile box or canvas page

Manipulating Styles

Styles are set with the `setStyle` method and queried with the `getStyle` method. `setStyle` can set multiple styles using the syntax:

```
obj.setStyle({color: 'black', background: 'white'});
```

Or one style at a time using:

```
obj.setStyle('color', 'black');
```

Beware you need to camelize style names. This works:

```
obj.setStyle('textDecoration', 'underline')
```

But this won't:

```
obj.setStyle('text-decoration', 'underline')
```

You must also remember to use 'px' notation when referring to positions or height/width, and so forth.

This works:

```
obj.setStyle('width', '340px')
```

But this doesn't:

```
obj.setStyle('width', '340')
```

This is important to remember when you're using algorithms to calculate those values. You can't just use the calculated variable `x` like: `setStyle('left', x)`, but rather like `setStyle('left', x+'px')`.

Additional functionality for manipulating CSS classes has been added to FBJS DOM nodes.

`addClassName(className)`

Adds a class name to the `className` string if it isn't already present.

`removeClassName(className)`

Removes a class name from the `className` string if it present.

`toggleClassName(className)`

If a class name exists, it removes it. If it doesn't exist it adds it.

`hasClassName(className)`

Returns true if the class name exists or false otherwise.

Setting Content

`innerHTML` isn't implemented for security reasons. Three alternatives exist.

1. `obj.setTextValue(newText)` can be used to set a literal text value inside of your DOM object (no HTML or FBML accepted).
2. `obj.setInnerFBML(fbJsStringVar)` can be used to put HTML or FBML inside of your DOM object. Note that you need to create a [Fb.js-string](#) object first and pass it in as passing a string literal will result in an error.
3. `obj.setInnerXHTML(string)` is a beta feature that allows you to place a string of XHTML directly into the document. The XHTML is sanitized in JavaScript before being rendered.

Working with Text Fields

Textbox selections have been implemented with the methods `getSelection` and `setSelection`. `getSelection` returns an object with properties `start` and `end` which correspond to the W3C-style attributes `selectionStart` and `selectionEnd`. `setSelection` takes two arguments, `start` and `end` (optional). This abstraction was added because Internet Explorer does not support `selectionStart` and `selectionEnd`. Since it is quicker in IE to retrieve both values together, they were coupled together into a single getter and setter. This function should work the same in all browsers with no extra work from you.

Creating FBML Elements

You can also use `createElement` to create FBML elements, although this is currently limited to [fb:swf](#). Once it's created, it works just like any other DOM object does, however, once it is attached to the DOM you cannot move it and `obj.getElementsByTagName('fb:swf')` does not work.

```
var newSwf = document.createElement('fb:swf');
```

Events

Events can be added to FBJS DOM objects using the W3C-style [addEventListener method](#). The third parameter, `useCapture`, is not supported. `removeEventListener` is also supported. In addition to the W3C event methods, we've also added `listEventListeners` and `purgeEventListeners`.

`listEventListeners(eventName)`

Returns an array of handles of all events that have been added to this event. Events that were added in FBML using the `on<event>` attributes will also be included

`purgeEventListeners(eventName)`

Removes all event listeners for a given event. This also removes events that were added as attributes in FBML. Event handlers are called with one parameter, which is an object with information about the event. In the case of event handlers added as attributes, this object will be accessible through the "event" variable (just as it is in regular JavaScript). The event will have attributes `target`, `type`, `pageX`, `pageY`, `ctrlKey`, `keyCode`, `metaKey`, and `shiftKey`. It also implements two methods:

`stopPropagation`

Prevents this event from propagating to any more elements further up in the DOM.

`preventDefault`

Cancels the default behavior of this event without stopping propagation. For instance, `preventDefault` on an `onfocus` event will prevent that element from getting focus.

`getId` on event object of a listener function

When using the `getId()` method inside an event listener function on the event object, the following syntax may be used to retrieve the ID of the object that fired the event:

```
<div id="firedByDescription"></div>
```

```
<div id="foo"></div>
```

```
<div id="bar"></div>
```

```
<script>
```

```
//disclaimer: sample code block meant only to demonstrate functionality
```

```
function myEventHandler(evt) {
```

```
  //we'll use this div later to drop stuff into it
```

```
  firedByDescription = document.getElementById('firedByDescription');
```

```
  if (evt.type == 'mouseout') {
```

```
    //if the event is a mouseout, empty out the description div, and exit the event listener
```

```
firedByDescription.setTextValue("");
return true;
}

//otherwise... do some processing:

/**VERY IMPORTANT*: note that the object, which fired the event is located two nodes up in the DOM tree
//See note below
//eventFiredBy_ObjectId = evt.target.getParentNode().getParentNode().getId();

//On newer versions, it seems that there is no need to go up two levels into the DOM tree, hence
eventFiredBy_ObjectId = evt.target.getId();
//works, whereas the first does not!

/**NOTE** My testing of this suggests that when you call addEventListener() it adds it to the element, AND
all its descendants
// This can then cause the event to be fired multiple times, as it is fired for the element and its descendant
elements.
// When fired by a descendant element, you will probably have to do some kind of getParent()-ing
// I'm raising this as a bug, as it does make things a little unworkable!

//once you have the ID, you may, for example, drop its id into the firedByDescription div:
firedByDescription.setTextValue(eventFiredBy_ObjectId);

//... or do some conditional processing:
if (eventFiredBy_ObjectId == 'foo') {
    //do something if the event was fired by 'foo'
} else {
    //do something if the event was fired by 'bar'
}
}

//add event listener to 'foo' div (mouseover & mouseout)
document.getElementById('foo').addEventListener('mouseover',myEventHandler);
document.getElementById('foo').addEventListener('mouseout',myEventHandler);

//add *the same* event listener to 'bar' div (mouseover & mouseout)
document.getElementById('bar').addEventListener('mouseover',myEventHandler);
document.getElementById('bar').addEventListener('mouseout',myEventHandler);
</script>
```

This functionality is very useful in cases where you have one event handler for multiple objects of the same type. Take for instance a shopping cart of some sort, or any type of object browser. When a user moves her mouse over one of the items in the cart, you may want to conditionally display information about the item -- by finding its ID, you may associate a description text for that item and display it to the user in another div. As you can see, using event listeners can be a very powerful way to display useful additional information to the user, based on where they move their mouse within your Facebook application. Happy coding and creativity!

AJAX

FBJS supplies a very powerful AJAX object for developers. Facebook proxies all AJAX requests and optionally runs useful post-processing on the data returned, such as JSON, or FBML parsing. To use it, just instantiate a new AJAX class. It supports the following properties:

`ondone(data)`

An event handler which fires when an AJAX call returns. Depending on `.responseType`, `data` is an object, a raw string, or an FBML string.

`onerror`

An event handler that fires when an error occurs during an AJAX call.

`requireLogin`

If you set this to true the AJAX call will require the user to be logged into your application before the AJAX call will go through. The AJAX call will then be made with the regular `fb_sig` parameters containing the user's identity. If they refuse to login, the AJAX call will fail.

`responseType`

This can be one of `Ajax.RAW`, `Ajax.JSON`, or `Ajax.FBML`.

`useLocalProxy`

Beta. If this is true and you are using RAW or JSON type, the Ajax object will attempt to use [fb:local-proxy](#) to make a direct call to your app server. See [FBJS_LocalProxy](#) for more details.

`Ajax.RAW`

The response from your server is returned to your callback in its original form.

`Ajax.JSON`

The response from your server is parsed as a JSON object and returned to your callback in the form of an object. Properties of your JSON object that are prefixed with "fbml_" are parsed as individual FBML strings and returned as FBML blocks. These blocks can be used on a DOM object with the `setInnerFBML` method. Each variable and its value in the response is limited to a combined length of 5000 characters. **Note:** be sure to use `json_encode` or else you may see odd results with large data sets. See [Bugzilla #363](#) for more information. `json_encode` is available by default in most PHP5 installations, and implementations for many other languages are available at [json.org](#).

`Ajax.FBML`

The response from your server is parsed as FBML and returned as an FBML block. This block can be used on a DOM object with the `setInnerFBML` method.

And two methods:

`post(url, query)`

Start an AJAX post. `url` must be a remote address, and `query` can be either a string or an object that is automatically converted to a string.

`abort()`

Aborts an AJAX post.

Here's an example showing most of the functionality of AJAX: [Ajax Example](#)

Dialogs

Dialog is an object we've created to allow you to hook into our base dialog abstractions. It allows you to create rich and fully dynamic dialogs for your application.

`Dialog(type)`

(constructor) type can be either Dialog.DIALOG_POP or Dialog.DIALOG_CONTEXTUAL.

Dialog.DIALOG_POP

This is the type of dialog that shows up when you delete a wall post.

Dialog.DIALOG_CONTEXTUAL

This is type of dialog that shows up when you delete a minifeed story.

onconfirm

An event handler that fires when the user selects the button designed as "confirm" (left most button). If this event doesn't return false the dialog will be hidden.

oncancel

An event handler that fires when the user selects the button designed as "cancel" (right most button). If this event doesn't return false the dialog will be hidden.

setStyle

Allows you to set the style of the parent dialog node

showMessage(title, content, button_confirm = 'Okay')

Displays a dialog with only a confirm button. title and content can be either strings or pre-rendered FBML blocks.

showChoice(title, content, button_confirm = 'Okay', button_cancel = 'Cancel')

Displays a dialog with Confirm and Cancel buttons. title and content can be either strings or pre-rendered FBML blocks.

setContext

(only applicable for DIALOG_CONTEXTUAL). Sets the context of a dialog, which basically means where the cursor arrow is pointing.

hide

Hides this dialog if it is visible.

FBML Blocks

Blocks of pre-rendered FBML can be exported into your JavaScript scope on page load. To do this, simply wrap a block of FBML inside an `<fb:js-string var="variable_name">` tag (see [Fb:js-string](#) for more information). Instead of rendering the block of FBML on the page it is put into a FBML block variable, which you can then use in your JavaScript with `setInnerFBML`. This is useful, because tags like `<fb:swf>` get rendered without `waitforclick` restrictions. FBML blocks can also be retrieved from AJAX calls, as explained above.

Animation

Facebook has provided a powerful animation library in FBJS. See [Animation](#) for more details.

Examples

[Hello World!](#)

[Typeahead widget](#)

[Ajax Typeahead widget](#)

[Dialogs](#)

[Dynamic Dialogs using Ajax](#)

[Ajax](#)

[Dynamic Tables](#)

[Slider Widget](#)

[Tabs](#)

Tips

Don't create JavaScript which depends on a sensitive DOM structure. Code like `this.getElementsByTagName('div')[1].getFirstChild().getLastChild().setStyle('color', 'white')` is very fragile and may randomly break if we change the way certain elements are rendered.

Most FBJS DOM methods are chainable. For instance, instead of:

```
var obj = document.createElement('div');
obj.addEventListener('click', click);
obj.addEventListener('mousemove', mousemove);
obj.setStyle('color', 'black');
```

You can do:

```
document.createElement('div').addEventListener('click', on_click).addEventListener('mousemove',
mousemove).setStyle('color', 'black');
```

You aren't allowed to extend base objects like Function or Array, however we do provide a typical "bind" implementation on the Function prototype.

FBJS objects don't contain handles to any of their actual DOM objects, however if you use [Firebug](#), the console can show you exactly to what an object is referring. Try `console.dir` on an FBJS DOM object. In your console you'll see a `PRIV_obj` attribute which is the actual DOM node represented by your FBJS DOM handle. This can help you figure out what FBJS is doing behind the curtains. This trick also works with all other FBJS objects such as AJAX and FBML blocks.

If you want to use timed fading/unfading of elements, creating 'panes' with drop-shadows or generic dragging, check out the small FBJS effects library Backface, at

<http://supercodex.com/backface/backface.zip>, with a demo at <http://apps.facebook.com/backface>.

Use [Firebug](#) to troubleshoot and diagnose anything that isn't working with your FBJS.

Consider using the [Include files](#) support to save processing/load times and bandwidth

Usage notes

[Forms](#)

[Images](#)

[CSS, Design, and the DOM](#)

[JavaScript](#)

[Flavors](#)

[Facebook User IDs](#)

[Default FBML](#)

[Public Canvas Pages](#)

Invalid CSS attributes

Certain CSS properties are stripped by the Facebook platform for security. *This section needs updating. We'll get to it soon.*

As of Dec 7, "position: absolute" maybe cause compatibility problems as mentioned here:

<http://developers.facebook.com/news.php?blog=1&story=56>

See also

[Requested FBML Tags](#)
[FBML DTD](#)

External links

[FBML Test Console](#)

Retrieved from "<http://wiki.developers.facebook.com/index.php/FBML>"