

## FORMTOOLS

### General attributes

<i>type</i>	boolean   date   longchar   numeric   string
<i>ftSeq</i>	position of field in edit form
<i>ftFieldSet</i>	legend for fieldset (optional)
<i>ftWizardStep</i>	name of wizard step (optional)
<i>ftLabel</i>	field label
<i>ftHint</i>	help text displayed under field

### Array / UUID

<i>ftJoin</i>	list of valid content types
<i>ftAllowSelect / ftAllowCreate / ftAllowEdit</i>	
<i>ftLibraryDataSQLWhere</i>	SQL where clause, filters selectable content
<i>ftLibraryData / ftLibraryDataTypename</i>	function (and function location), get's passwed primary object's object as first argument, should return query of objectid's

### ArrayUpload

Similar properties to array (i.e. *ftJoin*, *ftAllowSelect*), but must join ONE other content type

<i>ftFileProperty</i>	property to upload to (extension and size restrictions on that property are respected)
<i>ftEditableProperties</i>	if <i>ftAllowEdit</i> ="true", restrict to specified properties
<i>ftListWebskin</i>	defaults to librarySelected
<i>ftView</i> ="tiled" + <i>ftTileWidth</i> + <i>ftTileHeight</i>	tiled UI

### Boolean

### Category

<i>ftAlias</i>	category alias for branch to display
----------------	--------------------------------------

### Country

<i>ftCommon</i>	list of countries to display at top of list (codes or names)
<i>ftCountries</i>	restrict to specific countries (country code or full name)
<i>ftValue</i>	code   name, defaults to full name

### DateTime

<i>ftRenderType</i>	jquery   dropdown
<i>ftToggleOffDateTime</i>	boolean, allow user to disable field (e.g. expiry)
<i>ftShowTime</i>	boolean, field includes time, defaults to true

### Email

### File

<i>ftAllowedFileExtensions</i>	e.g. "pdf,doc,ppt"
<i>ftMaxSize</i>	maximum file size in bytes
<i>ftSecure</i>	defaults to false
<i>ftDestination</i>	storage directory

### Image

<i>ftAutoGenerateType</i>	none   center   fitinside   pad
<i>ftImageHeight / ftImageWidth</i>	used in automatic resizing

### Integer

### List

<i>ftList</i>	e.g. "opt1:Option 1,opt2:Option 2,opt3:Option 3"
<i>ftListData</i>	function that returns a list (ala <i>ftList</i> ) or a value+name query
<i>ftSelectMultiple</i>	

### LongChar

<i>ftLimit</i>	character limit
<i>ftLimitOverage</i>	truncate   warn

### Navigation

<i>ftAlias</i>	navigation alias for branch to display
----------------	--

### RichText

<i>ftImageArrayField</i>	property containing insertable content
<i>ftTemplateTypeList</i>	list of types that can be inserted (will still only list content already attached)

### State

<i>ftCountries</i>	list of country codes or names
<i>ftValue</i>	code   value
<i>ftWatch</i>	country property, updates states if user selects a country

## TAGS

```
<cfimport taglib="/farcry/core/tags/webskin" prefix="skin" />
```

### Insert view

```
<skin:view objectid="..." typename="..." webskin="..." />
```

### Link to another object

```
<skin:buildLink objectid="...">link text</skin:buildLink>
```

### Add pagination

```
<skin:pagination query="#qNews#" r_stObject="stObject">
  <skin:view stObject="#stObject#" webskin="displayTeaser" />
</skin:pagination>
```

### Load JavaScript into the head

```
<skin:loadJS id="jquery" />
<skin:loadJS id="myJS" baseHref="/js" lFiles="a.js,b.js" />
<skin:loadJS id="hello">alert("Hello world");</skin:loadJS>
<skin:loadJS id="external-jquery"
  lFiles="http://code.jquery.com/jquery-1.6.4.min.js" />
```

### Register a JavaScript library

Same attributes as `<skin:loadJS />`  
`<skin:registerJS ... />`

### Load CSS into the head

Supports the same attributes as `<skin:loadJS />`, plus 'media' and 'condition' (e.g. `condition='if IE'`)  
`<skin:loadCSS ... />`

### Insert site menu

Outputs nested UL lists  
`<skin:genericNav navID="#application.navid.home#"
 startLevel="2" depth="1"
 id="main"
 bIncludeHome="false" />`

### Alerts

Schedules a message to be displayed as a bubble on this page. All attributes are optional.

```
<skin:bubble title="Hello" message="world" pause="3000"
  sticky="false" tags="hi" />
```

### Custom alert handling

Alerts are displayed as Growl-like bubbles by default. Use this tag for custom output.

```
<skin:pop tags="hi" variable="stB" start="<ul>" end="</ul>">
  <li>#stB.title#: #stB.message#</li>
</skin:pop>
```

Inserts alert into top of page ala StackOverflow.

```
<cfimport taglib="/farcry/core/tags/formtools" prefix="ft" />
```

### Create a form

`<ft:object>` and `<ft:button>` should be enclosed by this tag.

```
<ft:form />
```

### Insert edit fields

```
<ft:object objectid="..." typename="..." lFields="..." />
```

### Add buttons to form

```
<ft:buttonPanel>
  <ft:button value="Cancel" validate="false" />
  <ft:button value="Save" />
</ft:buttonPanel>
```

### Process form post

```
<ft:processform action="Save">
  <ft:processformobjects typename="...">
    <cfset stProperties.label = stProperties.firstname />
    <cfif not isvalid("email",stProperties.email)>
      <ft:advice objectid="#stProperties.objectid#"
        field="email"
        message="That is an
  invalid email address"
        value="#stProperties.email#"
  />
    <ft:break />
  </cfif>
```

```
<cfimport taglib="/farcry/core/tags/container" prefix="con" />
```

### Add a container to a webskin

`<ft:object>` and `<ft:button>` should be enclosed by this tag.

```
<con:container label="#stObj.objectid#_top"
  defaultMirrorLabel="top"
  lRules="ruleSelectedNews,ruleRichText" />
```

## API

### application.fapi

*getContent*(*typename*) returns an instantiated component of the type  
*getContentObject*(*objectid*, *typename*) returns the property struct; *typename* is not required, but does reduce DB requests  
*getContentObjects*(*typename*, [*Properties*], [*orderBy*], [*maxRows*]) returns query on type, *objectid* and *typename* by default; results are automatically filtered by the status a user is allowed to see; the following additional arguments are also supported:  
*propertyname* [*eq*|*neq*|*gt*|*gte*|*lt*|*lte*|*in*|*notin*|*like*|*isnull*] these filters automatically detect the property type, account for null dates and FarCry faux null dates (i.e. 2050), and support any-category (catNews\_in) and every-category filters (catNews\_eq)  
*setData*(*stProperties*) saves the provided content struct to the DB  
*getNavID*(*alias*, *alternativeAlias*) returns the required *objectid*  
*getLink*(*objectid*, [*view*], [*bodyView*], [*urlParameters*]) returns the friendly URL (FUs) for the object; falls back to /index.cfm?objectid=... if FUs are disabled; the default page webskin and body webskin can be overridden; *urlParameters* are added as /key/value if FU functionality and the URL allow it;

This information is displayed in the FarCry tray when profile=1 is in the URL.

*addProfilePoint*(*section*, *label*) adds a timestamp point to the request profile  
*addRequestLog*(*text*) adds text to the request log

### application.security

*isLoggedIn*() is a user logged in  
*checkPermission*(*permission*, [*object*], [*type*]) returns true if the current user has the permission; if *object* is provided, the object permission (e.g. navigation view) is checked; if *type* is provided, the type permission (e.g. edit news) is checked

*session.dmProfile* profile information for the logged in user

/anyurl?logout=1 logs the user out

## Webskin Decorators

Decorators are properties set on a webskin using the form

```
<!-- @@decoratorname: value --->
```

### FU decorators

These decorators impact how friendly URLs containing the webskin will be generated and parsed. Currently they are not used elsewhere.

*viewStack* page | body | fragment | ajax | any  
 Defaults: displayPage\*: page, displayBody\*: body, all others: any  
*viewBinding* type | object | any  
 Defaults: displayType\*: type, all others: object  
*fuAlias* getLink will use this instead of the full webskin name

### Caching

Caching is enabled when a type has bObjectBroker=1 as a component attribute. Webskin caching is hierarchical - if a webskin is flushed, any webskin that uses it also gets flushed.

*cacheStatus* 1 | 0 | -1  
 1 = put the webskin into the cache, -1 = don't put this webskin or any webskin that contain it into the cache (trumps 1)  
*cacheTimeout* Seconds to cache  
*cacheTypeWatch* This webskin should be flushed from the cache if any content in the listed types is changed or created  
*cacheByURL* Set to true to include the full URL in the cache key  
*cacheByRoles* Set to true if the page is different for different user roles  
*cacheByVars* List of fully scoped variables to include in the cache key. Functions can be included in this list provided they, and their arguments, are fully scoped. Default values can be specified by adding =value, e.g. url.page=1

These decorators effect headers. The smallest value on a page is used.

*browserCacheTimeout* Sets the maxage header of the response  
*proxyCacheTimeout* Sets the s-maxage header of the response