

FLUIDTEMPLATE (TypoScript cObject) ([data] contains page properties)

template	cObject	extbase.pluginName	Plugin name: string/stdWrap
file	string/strWrap	extbase.controllerExtensionName	Extension name: string/stdWrap
layoutRootPath	filepath/stdWrap	extbase.controllerName	Controller name: string/stdWrap
partialRootPath	filepath/stdWrap	extbase.controllerActionName	Action name: string/stdWrap
format	string/stdWrap	variables	(array of cObjects) -> Access: {variable}
stdWrap	stdWrap	settings	(array of keys) -> Access: {settings.key}
templateRootPaths, layoutRootPaths, partialRootPaths => filepath array (Fallback paths)			

Addressing the view in action (controller class)

\$this->view->assign('key', \$value)	Makes \$key (which is of kind string, array or object) available as {identifier} in fluid. Multiple assign is possible.
\$this->view->assignMultiple(array('key'=>\$value))	Forces rendering of template (default is at actions end)

Templates, Layouts, Partials (in directory Resources/Private/...)

Template (Templates/ [Controller]/ [Action].html)	<f:layout name="default" /> <f:section name="content"> <f:render partial="partName" arguments="..." /> <f:render partial="..." arguments="{_all}" /> </f:section>	Layout (Layouts/Default.html)	<f:render section="content" optional="true" />
		Partial (Partials/PartName.html)	...

Object accessor syntax

{name.property}	Object accessor: Result of getProperty() in model {name}
{name.key} / {name.number}	Associative / Numeric array: Element in array {name} with {key} / at position {number}

ViewHelper syntax

{namespace f=TYPO3}Fluid{ViewHelpers}	Declares the abbreviation f as namespace for TYPO3/Fluid/ViewHelpers ViewHelper with the name {vname}. Corresponding class is found at: typo3/sysext/Fluid/Classes/ViewHelpers/VnameViewHelper.php
ATTRIBUTES:	FLUID-ARRAYS will be listed like: {key1:value1,key2:value2} e.g. each="{0:1,1:2,...}" or values="{0:'odd',1:'even'}" or arguments="{name:object}". Fluid arrays allowed in attributes only!
attr="value" or attr="FLUID-ARRAY"	

Inline syntax:

<f:vname argument="value"... /> could be written as {f:vname(argument:value, ...)} and <f:format.n12><f:format.crop maxCharacters="20">{some.value}</f:format.crop></f:format.n12> could be written as {some.value -> f:format.crop(maxCharacters:20) -> f:format.n12}</f:format.n12>

Boolean expressions - like: condition="XXX operator YYY"

XXX, YYY is of type number, object accessor, array or ViewHelper inline syntax operator is one of: == != % > < =

TagBasedViewHelper (general arguments for tag based viewhelper)

used with viewhelper: form (and all sub viewhelpers), image, link, renderFlashMessages

additionalAttributes	Associative array of additional tag-attributes	style	Individual CSS style
class	CSS classes for this tag	title	Tooltip text for this element
dir	Sets dir attribute - ltr or rtl	accesskey	Defines access key
id	Unique id	tabindex	Tab order for this element
lang	Language of this element (RFC 1766)	onclick	JavaScript for onclick event

alias ViewHelper - define alias of variables

<f:alias map="{x: foo.bar.baz, y: foo.bar.baz.name}">{x.name} or {y}/</f:alias>	
map	Mapping of alias - array - just valid inside alias tags

base ViewHelper - returns a base href tag

cObject ViewHelper - display TypoScript object (access with data & current)

<f:cObject typscriptObjectPath="lib.customHeader" data="{article}" currentValueKey="{article.title}" />	
typscriptObjectPath	TypoScript object path which should be used
data	Data which is used for the rendering - same as <f:cObject>data/</f:cObject>
currentValueKey	Key which sets the stdWrap property current

comment ViewHelper - comment out

count ViewHelper - counts elements

<f:count subject="{0:1, 1:2, 2:3, 3:4}" />	
subject	The array or ObjectStorage to iterated over

cycle ViewHelper - iterates through given values

<f:for each="{0:1, 1:2, 2:3, 3:4}" as="foo"> <f:cycle values="{0: 'odd', 1: 'even'}" as="zebraClass"><li class="{zebraClass}">{foo}</f:cycle> </f:for>	
values	Array of values which is used for iteration
as	Name of iteration variable

debug ViewHelper

<f:debug>{testVariables.array}</f:debug>			
ansiColors	Add color to output (FALSE)	maxDepth	Sets the max recursion depth (8)
blacklistedClassNames	An array of class names (RegEx) to be filtered (NULL)	plainText	Plaintext output or HTML (FALSE)
blacklistedPropertyNames	An array of property names and/or array keys (RegEx) to be filtered (NULL)	title	Optional custom title for the debug output
inline	Inline rendering (FALSE)		

flashMessages ViewHelper - renders the flash messages (if there are any)

<f:flashMessages renderMode="div" />	
renderMode	One of 'div' or 'ul' (ul is default)

for ViewHelper - foreach function

<f:for each="{fruit1: 'apple', fruit2: 'pear', fruit3: 'banana', fruit4: 'cherry'}" as="fruit" key="label"> {label}: {fruit} </f:for>			
<f:for each="{0:1, 1:2, 2:3, 3:4}" as="foo" iteration="fooIterator"> Index: {fooIterator.index} Cycle: {fooIterator.cycle} Total: {fooIterator.total} {f:if(condition: fooIterator.isEven, then: 'Even')} {f:if(condition: fooIterator.isOdd, then: 'Odd')}{f:if(condition: fooIterator.isFirst, then: 'First')} {f:if(condition: fooIterator.isLast, then: 'Last')} </f:for>			
each	Array of values or objects which is used for iteration	key	Key of iteration variable
as	Name of iteration variable	reverse	If TRUE, direction will be reversed
iteration	The name of the variable to store iteration information index, cycle, isFirst, isLast, isEven, isOdd, total		

groupedFor ViewHelper - grouping of results

<f:groupedFor each="{0: {name: 'cherry', color: 'red'}, 1: {name: 'banana', color: 'yellow'}, 2: {name: 'strawberry', color: 'red'}}" as="fruitsOfThisColor" groupBy="color" groupKey="color"> {color} fruits: <f:for each="{fruitsOfThisColor}" as="fruit" key="label">{label}: {fruit.name}</f:for> </f:groupedFor>			
as	Name of iteration variable	groupBy	Group by this property (property path possible)
each	Array or object which is used for iteration	groupKey	Name of variable which stores the grouping

if/then/else ViewHelper - if-then-else (w/o then if there is no else)

<f:if condition="somecondition"><f:then>...</f:then><f:else>...</f:else></f:if>	
Short-hand-Syntax: {f:if(condition: '{rank}' > 100, then: 'rank is > 100', else: 'rank is <= 100')}	
<f:if condition="{rank} == {k.bar()}" /> / <f:if condition="{foo.bar} == 'stringToCompare'" />	
condition	XX Comparator YY (e.g. <f:if condition="{rank} > 100">) Comparator is one of: ==, !=, <, <=, >, >= and % The % operator converts the result of the % operation to boolean. XX and YY can be one of: number / Object Accessor / Array / a ViewHelper / a String

image ViewHelper - displays an image (uri.image VH for link to image)

<f:image src="EXT:myext/Resources/Public/typo3_logo.png" alt="alt text" /> {f:image(src: 'EXT:myext/Resources/Public/logo.png', alt: 'alt text', minWidth: 30, maxWidth: 40)}			
alt	Specifies an alternate text for an image	width	Width of the image ('m'/'c' possible)
ismap	Specifies an image as a server-side image-map.	height	Height of the image ('m'/'c' possible)
longdesc	Specifies the URL to a document	minWidth	Minimum width of the image
usemap	Specifies an image as a client-side image-map	minHeight	Minimum height of the image
src	Source of the image	maxWidth	Maximum width of the image
treatIdAs	Given src argument is a sys_file_reference	maxHeight	Maximum height of the image
Reference	record	image	FAL object

form ViewHelper <f:form ...> - form generation

absolute	Render absolute action URI (FALSE)	name	Name of form
action	Form action (default is current uri)	noCache	Disable caching (FALSE)
actionUri	Override the "action" attribute	noCacheHash	Suppress the cHash (FALSE)
additionalParams	Add. action URI query params (Array)	pluginName	Target plugin (default is current)
addQueryString	Query params will be kept in the URI	pageUid	UID of target page
arguments	Additional arguments (Array)	pageType	Type of target page (default is 0)
controller	Target controller (default is current)	object	Object bound to the form->property
enctype	MIME type for transmission	objectName	Name of the object bound to the form
extensionName	Target extension name (def. is current)	onreset/onsubmit	JavaScript handler
format	The requested format like "html"	section	The anchor to be added to the URI
fieldNamePrefix	Prefix to be added to all field names. If not set: tx_ yourExtension_plugin	argumentsToBeExcludedFromQueryString	Arguments to be removed from the action URI
method	GET or POST (default)	hiddenFieldName	Class name of hidden fields

GENERAL ATTRIBUTES FOR ELEMENTS

errorClass	CSS class to set if there are errors	value	Value of element
name	Name of element	disabled	Displays the element disabled
property	Property of object bound through form	autofocus	for button, textfield & textarea

form.button - displays a button

<button type="submit" name="" value="">Send Mail</button>			
form	1 or more forms the button belongs to	formnovalidate	No validation of form data
formation	Form action	formtarget	Target (e.g. _blank, _self)
formenctype	Encoding	type	Type (e.g. button, reset, submit)
formmethod	Method (POST or GET)		

form.checkbox - displays a checkbox

<f:form.checkbox name="myCheckBox" value="someValue" checked="{object.value} == 5" />			
checked	If TRUE then checkbox is checked	property="myProperty" value="myValue"	if {object.myProperty} == myValue => checked

form.hidden - displays a hidden field

<f:form.hidden name="myHiddenValue" value="42" />	
---	--

form.password - displays a password input field

<f:form.password name="myPassword" />			
maxlength	Maximum length of field	size	Length of input field
readonly	Readonly attribute of field		

form.radio - displays a radio button

<f:form.radio name="myRadioButton" value="someValue" checked="{object.value} == 5" />		
checked	If TRUE then radiobutton is checked	see form.checkbox above (property="...")

form.select - displays a selector box

<f:form.select name="users" options="{userArray}" optionValueField="id" optionLabelField="firstName" />			
selectAllByDefault	Select all options	size	Length of selector box
optionLabelField	Property label field	multiple	Display a multi-select box
optionValueField	Property value field (uid default)	prepend OptionLabel	An option at first position - label
options	Assoc array or object used for options	prepend OptionValue	An option at first position - value
sortByOptionLabel	List will be sorted by label		

form.submit - displays a submit button

<f:form.submit value="Send Mail" />	
-------------------------------------	--

form.textarea - displays a text area

<f:form.textarea name="myTextArea" value="This is shown inside the textarea" rows="5" cols="30" />			
cols	Number of cols	rows	Number of rows
placeholder	Placeholder text		

form.textfield - displays an input field

<f:form.textfield name="myTextBox" value="default value" />			
maxlength	Maximum length of field	required	Required attribute of field
placeholder	Placeholder attribute of field	size	Length of input field
readonly	Readonly attribute of field	type	Type (e.g. text, email, ...)

form.upload - displays an upload field (just works with enctype="multipart/form-data" in form tag)

<f:form.upload name="file" />	
-------------------------------	--

form.validationResults (use attribute ,for' to restrict it to a property, e.g. for="blog.title" and ,as' as error name)

<f:form.validationResults><f:if condition="{validationResults.flattenedErrors}"><ul class="errors"><f:for each="{validationResults.flattenedErrors}" as="errors" key="propertyPath">{propertyPath}<f:for each="{errors}" as="error">{error.code}: {error}</f:for></f:for></f:if></f:form.validationResults>	
---	--

format ViewHelper - formats in different ways (use value attr or child VH)

format.bytes - Formats an integer with a byte count into human-readable form			
<code><format.bytes value="1024" decimals="2" decimalSeparator="," thousandsSeparator=" " /></code>			
decimals	Number of digits after decimal point	thousandsSeparator	Character for thousands
decimalSeparator	Decimal point character		
format.cdata - Outputs an argument/value without any escaping and wraps it with CDATA tags.			
<code><format.cdata value="" /></code>			
format.crop - Use this view helper to crop the text between its opening and closing tags			
<code><format.crop maxCharacters="17" append="&nbsp;[more]" /></code>			
append	String, which is appended at crop position	respectHtml	Cropped string will respect HTML
maxCharacters	Max. characters which are displayed	respectWordBoundaries	Crops only at word boundaries
format.currency - displays currency conversions			
<code><format.currency value="100" thousandsSeparator="," currencySign="€" /></code>			
currencySign	The currency sign (like \$, €, ...)	prependCurrency	Prepend currency sign (FALSE)
decimals	Set decimals places	separateCurrency	Space between signs (FALSE)
decimalSeparator	Character for decimal separation	thousandsSeparator	Character for thousands sep.
format.date - displays dates (use @ if you handle with unix timestamps) - default is ["SYS" "dmmmy"]			
<code><format.date format="H:i" /></code>			
format	Format in date() syntax	date	DateTime object or string
format.htmlEntitiesDecode - Applies <code>html_entity_decode()</code> to a value			
<code><format.htmlEntitiesDecode encoding="ISO-8859-1" /></code>			
encoding	Encoding	keepQuotes	Keep Quotes (FALSE)
format.htmlEntities - Applies <code>htmlEntities()</code> escaping to a value			
<code><format.htmlEntities encoding="ISO-8859-1" /></code>			
encoding	Encoding	keepQuotes	Keep Quotes (FALSE)
format.htmlspecialchars - Applies <code>htmlspecialchars()</code> escaping to a value			
<code><format.htmlspecialchars encoding="ISO-8859-1" /></code>			
encoding	Encoding	keepQuotes	Keep Quotes (FALSE)
doubleEncode	If FALSE html entities won't be encoded		
format.html - Renders strings with TYPO3 parseFunc			
<code><format.html parseFuncPath="lib.parseFunc" /></code>			
parseFuncPath	Path to own parseFunc. Default is "lib.parseFunc RTE"		
format.nl2br - Wrapper for PHP function nl2br			
<code><format.nl2br /></code>			
format.number - Formats numbers			
<code><format.number decimals="1" decimalSeparator="," thousandsSeparator=" " /></code>			
decimals	Numbers after comma. Default is "2"	thousandsSeparator	Character for thousands sep.
decimalSeparator	Character for decimal separation		
format.padding - Wrapper for PHP function str_pad()			
<code><format.padding padLength="10" padString=" " /></code>			
padLength	Length of outputted string	padType	Values: right (default), left, both
padString	String which is used for filling up		
format.printf - Wrapper for PHP function printf()			
<code><format.printf arguments="number: 362525200" /></code>			
arguments	Arguments for printf as array		
format.raw - Outputs an argument/value without any escaping			
<code><format.raw value="" /></code>			
format.stripTags - Removes tags from the given string (applying PHPs strip_tags() function)			
<code><format.stripTags /></code>			
format.urlencode - Encodes the given string (applying PHPs rawurlencode() function)			
<code><format.urlencode /></code>			

translate ViewHelper - (from Resources/Private/Language/locallang.xlf)

<code><translate key="LLL:EXT:myext/Resources/Private/Language/locallang.xlf:key1" /></code>			
arguments	Arguments	htmlEscape	If FALSE, the output will not be escaped (TRUE)
default	Default key if 'key' is not found	id	Key in locallang file (has precedence over key)
extensionName	UpperCamelCased extension key	key	Key in locallang file

link/uri ViewHelper - generates URLs (link.x with tag / uri.x without tag)

name	Specifies the name of an anchor	target	Target parameter
rel	Rel: current => linked document	rev	Rel: linked => current document
link.action / uri.action - generates extbase action links			
<code><link.action action="show" /></code>			
absolute	Render absolute URI	format	The format, e.g. ".html"
action	Target action	linkAccess	Show even access restricted pages
additionalParams	Additional parameters	noCache	Deactivate cache for target page
addQueryString	Query params kept in the URI	noCacheHash	No cHash parameter
addQueryStringMethod	GET POST GET,POST POST,GET		
arguments	Arguments	pageType	Page type (default 0)
argumentsToBeExcluded	Arguments to be removed from the action URI	pageUid	Target page UID (default current)
controller	Target Controller	pluginName	Target Plugin
extensionName	Target Extension	section	Anchor to be added to the URI
link.email / uri.email - generates an email link			
<code><link.email email="foo@bar.tld" /></code>			
email	Email address		
link.external / uri.external - generates links to external targets			
<code><link.external uri="http://www.typo3.org" /></code>			
defaultScheme	Scheme - 'http' is default	uri	Target URL
link.page / uri.page - generates links to TYPO3 pages			
<code><link.page pageUid="1" /></code>			
absolute	Render absolute URI	noCache	Deactivate cache for target page
additionalParams	Additional parameters	noCacheHash	No cHash parameter
addQueryString[Method]	Query params kept in the URI	pageType	Page type (default 0)
argumentsToBeExcluded	Arguments to be removed from the action URI	linkAccess	Show even access restricted pages
section	Anchor	restrictedPages	Show even access restricted pages
		pageUid	Target page UID (default current)
uri.resource - creating URIs to resources			
<code><uri.resource uri="css/stylesheets.css" /></code>			
absolute	Render absolute URI	path	The path & filename of the resource
extensionName	Target extension name		

security ViewHelper

security.isAuthenticated - implements an isAuthenticated/else condition for FE users/groups	
<code><security.isAuthenticated /></code>	
security.ifHasRole - implements an ifHasRole/else condition for FE users/groups	
<code><security.ifHasRole role="Admin" /></code>	
role	Group role (either UID or title)

switch / case ViewHelper

<code><switch expression="person.gender" /></code>	
--	--

be ViewHelper - backend module viewhelper

be.container - View helper which allows you to create extbase based modules in the style of TYPO3 default modules.			
<code><be.container pageTitle="foo" enableJumpToUrl="false" /></code>			
addJsInlineLabels	Custom labels to add JS inline labels	loadExtJS	Specifies whether to load ExtJS
enableClickMenu	If TRUE (default), loads clickmenu.js	loadExtJSTheme	Whether to load ExtJS „grey“ theme
enableExtJSDebug	If TRUE, debug version of ExtJS is loaded.	loadQuery	Whether to load Query library
enableJumpToUrl	If TRUE (default), includes "jumpToUrl.js"	loadPrototype	Specifies whether to load prototype lib
extJSAdapter	Load alternative adapter (ext-base is def.)	loadScriptaculous	Specifies whether to load scriptaculous
includeCssFiles	List of custom CSS files to be loaded	pageTitle	Title tag of the module
includeJsFiles	List of custom JS files to be loaded	scriptaculousModule	Add. modules for scriptaculous
be.pageInfo - return page info icon			
<code><be.pageInfo /></code>			
be.pagePath - current page path, prefixed with „Path.“ and wrapped in a span with the class "typo3-docheader-pagePath"			

be.tableList - renders a record list as known from the TYPO3 list module

<code><be.tableList tableName="fe_users" /></code>			
alternateBackgroundColors	Rows have alt bgcolors	readOnly	Edit icons won't be shown if TRUE
clickTitleMode	"edit", "show" or "info"	recordsPerPage	Amount of records to be displayed
enableClickMenu	Enables context menu	sortDescending	Records sorted in descending order
fieldList	List of fields to be displayed (array)	sortField	Table field to sort the results by
filter	Corresponds to „Search String“ textbox	storagePid	Records are fetched from this PID
levels	Corresponds to the level selector	tableName	Name of the database table

be.buttons.csh - CSH button as known from the TYPO3 backend

<code><be.buttons.csh table="xMOD_csh_corebe" /></code>			
iconOnly	If set, full text will never be shown	styleAttributes	Additional style-attribute
field	Field name (CSH locallang main key)	table	Table name ('_MOD_' + module)
be.buttons.icon - returns button with icon (possible icons in ["coreSpriteImageNames"] typo3/system/core/ext_tables.php)			
<code><be.buttons.icon uri="furi.action(action:new)" /></code>			
icon	One of list above	uri	Target URI
title	Title attribute		

be.buttons.shortcut - returns shortcut button as known from the TYPO3 backend.

<code><be.buttons.shortcut getVars="(0, 'M, 1, myOwnPrefix') setVars="(0, 'function') /></code>			
getVars	List of GET variables to store	setVars	List of SET[] variables to store
be.menus.actionMenu / be.menus.actionMenuItem - select box, used to switch between multiple actions and controllers			
<code><be.menus.actionMenu /></code>			
action	Action to be called	controller	Controller to be called
arguments	Arguments	label	Label of option tag

be.security.isAuthenticated - implements an isAuthenticated/else condition for BE users/groups (like security.isAuthenticated)

be.security.ifHasRole - implements an ifHasRole/else condition for BE users/groups. (like security.ifHasRole)

API: Write own ViewHelper with name [Vhname] -

Put file [Vhname]ViewHelper.php in Classes/ViewHelpers/ with the following class definition:

```

class [Vhname]ViewHelper extends \TYPO3\CMS\Fluid\Core\ViewHelper\AbstractViewHelper {
    $arguments
    $templateVariableContainer
    $controllerContext
    $viewHelperVariableContainer
    render($arg1, $arg2, ...)
    renderChildren()
}

```

initializeArguments() Register arguments within this method with \$this->registerArgument(\$name, \$type, \$description, \$required, \$defaultValue)

initialize() Code before rendering of viewhelper

\$arguments Associative array of arguments of viewhelper. Accessible within render()

\$templateVariableContainer Contains all variables which are accessible in the template. Change with add() & remove(), read with get()

\$controllerContext Context of controller - serves as API for the controller. Supports the following properties: request, response, arguments, uriBuilder, flashMessageContainer, ...

\$viewHelperVariableContainer / Container for exchanging data between viewhelpers. Add with add() & read with get()

render(\$arg1, \$arg2, ...) Render method which is responsible for output of viewhelper. It's required.

renderChildren() Called within render() will return all rendered content from child elements

class [Vhname]ViewHelper extends \TYPO3\CMS\Fluid\Core\ViewHelper\AbstractTagBasedViewHelper { }

\$tagName Name of tag which will be generated by the viewhelper (default is 'div')

\$tag Instance TagBuilder. API: setContent(\$tagContent), getContent(), forceClosingTag(\$forceClosingTag), addAttribute(\$attributeName, \$attributeValue, \$escapeSpecialCharacters = TRUE), addAttributes(array \$attributes, \$escapeSpecialCharacters = TRUE), removeAttribute(\$attributeName), reset(), render()

registerUniversalTagAttributes() Registers all universal tag attributes like class, dir, id, lang, style, title, accesskey, ...

widget ViewHelper

widget.autocomplete - autocomplete for fields (needs Static TS 'Fluid: Default AJAX configuration')			
<code><widget.autocomplete for="name" objects="posts" searchProperty="author" /></code>			
for	Name of field for autocomplete	searchProperty	Property of object to search in
objects	Objects to search in		
widget.link / widget.uri - create links to extbase actions within widgets (link with a-tag / uri without)			
<code><widget.link action="show" /></code>			
action	Same arguments as action.link or uri	ajax	TRUE for link to ajax widget
arguments	Arguments	section	The anchor to be added to the URI
widget.paginate / be.widget.paginate - renders a pagination of objects (FE and BE)			
<code><widget.paginate objects="blogs" as="paginatedBlogs" configuration="{itemsPerPage: 3, insertAbove: 1, insertBelow: 1}" /></code>			
as	Identifier of paginated objects	objects	Objects that will be paginated
configuration	itemsPerPage, insertAbove, insertBelow, maximumNumberOfLinks (FE), recordsLabel (BE)		