

## Plugin DataChannel

Use this [wiki plugin](#), introduced in [Tiki4](#), to display a form to access data channels. This plugin is similar to [PluginProfile](#) but for [Data Channels](#). Please also see [the data channels profile page](#).

Parameters:

The general format for this plugin is:

```
{DATACHANNEL(various_parameters="")} body input to capture variables passed to the profile  
{DATACHANNEL}
```

Display a form to access data channels

Introduced in Tiki 4. Required parameters are in **bold**.

[Go to the source code](#)

Preferences required: `wikiplugin_datachannel`

Parameters	Accepted Values	Description	Default	Since
(body of plugin)		List of fields to display. One field per line. Comma delimited: <code>fieldname,label</code>  To use values from other forms on the same page as parameters for the data-channel use <code>fieldname, external=fieldid</code> or <code>fieldname, external=fieldid,text</code> can be used where a value is simply listed on the same page where each value has the <code>fieldid</code> as an html id tag. Where <code>fieldid</code> is the id (important) of the external input to use, and <code>fieldname</code> is the name of the parameter in the data-channel . To use fixed hidden preset values use <code>fieldname, hidden=value</code>		
<b>channel</b>	text	Name of the channel as registered by the administrator.		4.0
<code>buttonLabel</code>		Label for the submit button. Default: "Go".	Go	6.0
<code>class</code>		CSS class for this form		6.0
<code>paymentlabel</code>	text		6.0	
<code>returnURI</code>	url	URL to go to after data channel has run. Defaults to current page. Can contain placeholders <code>%reference%</code> or <code>%reference:urlencode%</code> , where reference matches a profile object ref, allowing to redirect conditionally to a freshly created object.		6.0

<code>returnErrorURI</code>	url	URL to go to after data channel has run and failed. If not specified, use the success URL.		12.0
<code>template</code>	text	Template to be used to render the form, instead of the default template		15.0
<code>quietReturn</code>	(blank) y n	If set to <code>y</code> , will return quietly after data channel has run which would be needed if plugin is used in non-wiki page context.	n	6.2
<code>emptyCache</code>	(blank) all templates_c temp_cache temp_public prefs none	Which caches to empty. Default "Clear all Tiki caches"	all	6.0
<code>price</code>	text	Price to execute the data channel (USD).		6.0
<code>debug</code>	(blank) y n	Be careful, if debug is on, the page will not be refreshed and previous modules can be obsolete (not on by default)	n	5.0
<code>array_values</code>	(blank) y n	Accept arrays of multiple values in the POST. e.g. <code>itemId[]=42&amp;itemId=43</code> etc. (multiple values not accepted by default)	n	6.0

Body input options:

Profile Requests - typed in input

The most typical body input option allows data to be typed in to an input field prior to the datachannel being executed and the profile being run and has the format:

`variablename, some text to describe the input required for this variable`

Where `variablename` can then be used as a `profilerequest` variable in the profile that is run and/or in wiki page content templates used to populate new wiki pages. The format for use in profiles is:

`$profilerequest:variablename$VARIABLE NOT SET$`

(the second set of text between the \$ characters provides default text if for some reason the variable has not been set - it doesn't have to be CAPS ! )

But one small problem is that any input made this way will always end up as text that is passed to whatever Tiki object is used in the profile - and this will be a problem if the end storage is a tracker date field. The input process therefore needs to be 'amended' so that the resultant input is the appropriate Linux integer representing the date.

This can only be done by a JS 'hack' at present but this is easily done by placing the following JS code in the "Custom javascript:" area of the Customization tab of the Look & Feel admin screen:

```
$(document).ready(function () { var $dateField = $("<input type='text' class='form-control'>");
$("[name=confirmdate]").hide().parent().append($dateField); $dateField.tiki("datepicker", "jscalendar",
{altField:"[name=confirmdate]"}); });
```

Where you will need to change the `name` elements from `confirmdate` to whatever you have called your `variablename`.

This will then automagically make the input entry field into a datepicker function and this then makes the data entered into the appropriate Linux integer.

### Profile Requests - Using a manual template

From [Tiki 15](#) onwards, it is possible to use a custom Smarty (ie. HTML) template to create the form that is used to collect user input for the datachannel. This provides for complete flexibility. To do this, specify the "template" parameter as the filename of the template file (relative to the templates folder), and start off with a copy of `template/wiki-plugins/wikiplugin_datachannel.tpl`. Remove the "foreach" section that creates the fields normally and then simply put your fields in. A very basic example:

```
<form method="post"
action="#{$datachannel_execution}"{$form_class_attr}{$datachannel_form_onsubmit}> Select your
favorite juice: <select name="juicetype"> <option value="apple">Apple</option> <option
value="orange">Orange</option> </select> <div class="submit_row"> <input type="hidden"
name="datachannel_execution" value="{ $datachannel_execution|escape}"> <input type="submit"
class="btn btn-default btn-sm" value="{tr}{ $button_label}{/tr}"> </div> </form>
```

Note that you still have to provide the names of the fields you are passing in the body of the DATACHANNEL plugin, even though the labels are not needed, for example:

`juicetype`

instead of something like the following which would generate a text field with the "Juice Type" label:

`juicetype, Juice Type`

but even if you did that it would not matter. It would just ignore the label provided.

### Profile Requests - 'hidden' input:

Available from [Tiki9](#), if you want to provide fixed input that is hidden, you can simply specify:

`variablename,hidden=value`

This could be useful when used in conjunction with [Wiki Argument Variables](#), for example:

`creator,hidden=`

### Profile Requests - 'external' input

Available from [Tiki6](#), a very useful option for automatically generating profilerequest variable content is to use the `external` option which has a plugin body format as follows:

`variablename,external=tagId`

This allows data captured in input forms 'earlier' on the page to be automatically assigned to the `variablename`. For example if a TRACKERFILTER is used on the page prior to the DATACHANNEL plugin any of the selected filter fields can be used as 'externals' since the filter fields are all given Id tags of `f_nnn` where `nnn` is the tracker field#.

An example of this might look like:

```
{TRACKERFILTER(trackerId=8,filters=100/d, fields=98:99:100, noflipflop=y, status="opc",
showstatus="y", action="Show filter results", showlinks="n")} {TRACKERFILTER}
```

Followed by:

```
{DATACHANNEL(channel="channelname" buttonLabel="run profile" emptyCache="none")}  
variablename,external=f_100 {DATACHANNEL}
```

Profile Requests - 'external' input: further extended

An extension of the 'external' option was introduced in Tiki15 (and backported to Tiki 12.4 and 14.0) to allow any data previously displayed on the page, as long as it had an individual html Id tag assigned to it. This then allows simple constructs that tabulate data with html tags or more useful filtering of trackers can be carried out using TRACKERFILTER, TRACKERLIST and LIST etc., with appropriate templating, to display sets of data with Id tags embedded that then allow 'external' data to be very flexibly passed to the profile.

The format for this type plugin body input has a modifier added to the end of the standard 'external' format as follows:

```
variablename,external=tagId,text
```

Additional information

It should be noted that all types of input can be mixed and matched in a single datachannel usage eg you can use something like this to capture various 'external' data as well as typed in data

---

```
{DATACHANNEL(channel="channelname" buttonLabel="run profile" emptyCache="none")}  
variablename01,external=var01Id,text variablename02,external=var02Id,text  
variablename03,external=f_100 variablename04,external=var04Id,text  
variablename05,external=var05Id,text variablename06,type in some text for this variable06  
variablename07,external=var06Id,text {DATACHANNEL}
```

Related pages

See in action:

[http://profiles.tiki.org/Project\\_Management&highlight=datachannel](http://profiles.tiki.org/Project_Management&highlight=datachannel)

[http://profiles.tiki.org/Learning\\_Center&highlight=datachannel](http://profiles.tiki.org/Learning_Center&highlight=datachannel)