

PluginList output control block

The OUTPUT control block defines what the output 'template' will be for the results.

Parameters

parameter	value	example	since
downloadable	<i>filename.csv</i>	<code>downloadable="MyData.csv"</code>	18.0
downloadable-position	<i>top/bottom/both</i>	<code>downloadable-position="both"</code>	25.0
inline	?	n/a	
report	<i>y/n</i>		20.0? (future)
template	<i>file.tpl</i> or <i>built-in template</i> (see below)	<i>my.tpl</i> <code>path/to/template/file/my.tpl</code>	
tplwiki	<i>wiki page</i> (parsed the same as Smarty tpl)	<code>tplwiki="MyTPL"</code>	
wiki	<i>wiki page</i> (parsed including wiki syntax)	<code>wiki="MyTPL"</code>	

The jQuery Sortable Tables feature must be activated for the sort feature to work.

Built-in templates

name	purpose	since
carousel	display the data in a carousel (slider)	14.0
count	just count the number of results	15.0
medialist		13.0
table	display the data in a table format	
debug	display all available data for debugging purposes	

The jQuery Sortable Tables feature must be activated for the sort feature to work.

Below the simple Wiki Formatting Within Plugin is discussed as well as the standard/built-in templates, e.g., table, medialist and carousel. For user defined wiki or smarty templates see [PluginList advanced output control block](#).

Basic Wiki Formatting Within Plugin

By placing Wiki syntax directly in the OUTPUT body content, along with the required objects to be displayed, the control block allows a template for each row of output to be constructed within the LIST plugin itself.

In the example below a table is constructed with the table 'header' row placed before the LIST and the table closing characters placed after the LIST. Each row of the table is then created by the LIST plugin

with each table cell element generated by the display control block and separated by the usual table separator, i.e., a `|`

Wiki Output Sample

```
||Type|Object {LIST()} {list max="10"} {OUTPUT()}{display name="object_type"}|{display
name="object_id"} {OUTPUT} {ALTERNATE()}This will display if no results are found{ALTERNATE}
{LIST} ||
```

Type	Object
wiki page	System Tracker Field
wiki page	PluginSlideshow
wiki page	PluginDl
wiki page	PluginJabber
wiki page	PluginPageTabs
wiki page	PluginArchiveBuilder
wiki page	PluginUserInGroup
wiki page	Features,ca
forum post	454
forum post	452

Table Formatting

A simple built-in templating method can be used by setting the OUTPUT template parameter to `table` and then using a column control block in the OUTPUT body content to define the individual columns of the table.

Table Column Control

Column Parameters

editable	<i>inline</i>	11.0
field	tracker field (permanent name) or FORMAT name reference	
label	text	
mode	<i>raw</i>	
sort	tracker field (permanent name) or FORMAT name reference	
translatelabel	<i>y/n</i>	18.0
class	text, use existing CSS class	21.0

The jQuery Sortable Tables feature must be activated for the sort feature to work.

Column Mode Parameter

When you use table output with a column generating HTML output (like `objectlink` and `trackerrender` formatted values), you should include the `mode="raw"` in the column specification. In the sample below,

the column "title" mode is set to **raw** as in the format we want to display an **objectlink**. Other formats are available; see more about it here: [PluginList display control block](#)








Format parameters

Added in Tiki 11+ `editable=inline` will set the field to be editable inline.

Required: "Inline editing (found in trackers)", `format="tracker render"` and column should be set to `mode=raw`.

Sample using the default table formatting

```
{LIST()} {pagination max="12"} {filter field="tracker_id" content="17"} {filter type="trackeritem"}
{OUTPUT(template="table")} {column label="Title" field="title_link" sort="title" translatelabel="y"
mode="raw" class="text-nowrap"} {column label="Started" field="started" mode="raw" class="text-
info"} {column label="Completed" field="completed" sort="tracker_field_18"} {OUTPUT}
{FORMAT(name="title_link")}{display name="title" format="objectlink"}{FORMAT}
{FORMAT(name="started")}{display name="tracker_field_started" format="trackerrender"
editable=inline}{FORMAT} {FORMAT(name="tracker_field_completed")}{display
name="tracker_field_completed"}{FORMAT} {LIST}
```

Title	Started	Completed
March	321 	566
October	111 	666
November	900 	800
September	766 	344
July	878 	566
August	321 	456
December	799 	276

Note that the `column` controls (within OUTPUT) are converted to a data array sent to the template. The table formatting uses [Smarty](#) formatting and provides the same capabilities.

(Tiki 18+) `translatelabel="y"` will translate the column label provided the translation is known to Tiki.

The use of the FORMAT control blocks allows individual objects to be 'formatted' and then referenced in the various OUTPUT methods - see [PluginList format control block](#) for more details.

Download CSV of Table Results

Since [Tiki18](#) the `downloadable=filename.csv` parameter can be used to add a Download button which generates a CSV file with all data found in the table. Header row is based on the table headers formatted via `column` mode parameter. This option is available if you use the "table" output (see: [Download CSV of Table Results](#) . What is interesting in this option is that it will export ALL the results found for the search query and not only the list displayed (pagination may limit the visible list of results).

This option is also available when using the [CustomSearch plugin](#) and the [List Execute plugin](#).

This option has been improved in Tiki25 offering an option to position the download button when using the table.tpl template.

Output formatting with download option

```
{filter content="22" field="tracker_id"} {OUTPUT(template="table" downloadable="Sample file.csv")}
{column label="Firstname" field="firstname"} {column label="Lastname" field="lastname"}
{OUTPUT} {FORMAT(name="firstname")}{display
name="tracker_field_contactsFirstName"} {FORMAT} {FORMAT(name="lastname")}{display
name="tracker_field_contactsLastName" format="trackerrender"} {FORMAT} {CUSTOMSEARCH}
```

Since Tiki25 the downloadable-position="top|bottom|both" parameter controls position of the Download button has been added. Values are top, -+bottom+- or both. Default is "bottom".

Output formatting with download option

```
{OUTPUT(template="table" downloadable="Sample file.csv" downloadable-position="top")} ...column
specification... {OUTPUT}
```

Table formatting with TableSorter

Starting with Tiki15, PluginList can be used with tablesorter. Use a {tablesorter} tag to add the tablesorter parameters. See below, which uses the same syntax as above except for the addition of the tablesorter tag:

Sample with tracker list and calculation

```
{LIST()}{pagination max="12"} {filter field="tracker_id" content="17"}
{OUTPUT(template="table")} {column label="Month" field="month" mode="raw"} {column
label="Started" field="started"} {column label="Completed" field="completed"} {column
label="Status" field="status"} {tablesorter server="n" sortable="type:reset"
tsortcolumns="type:text|type:none|type:none|type:text"
tsfilters="type:nofilter|type:nofilter|type:nofilter|type:dropdown;empty:empty" tstotals="|
|type:col;label:Total hours|" tstotaloptions="ignore|ignore| |ignore" tstotalformat="#.###."}
{OUTPUT} {FORMAT(name="month")}{display name="tracker_field_month"
format="objectlink"} {FORMAT} {FORMAT(name="started")}{display
name="tracker_field_started"} {FORMAT} {FORMAT(name="completed")}{DIV(type="span"
class="float-right")}{display name="tracker_field_completed"} {DIV} {FORMAT}
{FORMAT(name="status")}{display name="tracker_field_status" default=""}} {FORMAT} {LIST}
```

Will produce the following (note that a span and bootstrap class are added to align the numbers as by default tablesorter calculation are numbers aligned right)

Month	Started	Completed	Status
March	321	566	Behind
October	111	666	Completed
November	900	800	

Month	Started	Completed	Status
September	766	344	Completed
July	878	566	Ahead
August	321	456	Behind
December	799	276	

Sample adding tableorter functionality

```
{LIST()} {filter type="trackeritem"} {filter content="water"} {OUTPUT(template="table")} {column
label="Title" field="title_link" sort="title" mode="raw"} {column label="Description"
field="description"} {column label="Event Date" field="event_date" sort="tracker_field_18"}
{tableorter server="n" sortable="type:reset" tsortcolumns="type:text|type:text|type:digit"
tsfilters="type:text|type:dropdown|type:digit" tspaginate="max:10" tscolselect="critical|5|6"}
{OUTPUT} {FORMAT(name="title_link")}{display name="title" format="objectlink"}{FORMAT}
{FORMAT(name="event_date")}{display name="tracker_field_18" format="date"}{FORMAT} {LIST}
```

No results for query.

Please note that in some cases, if you want to use the tsfilters, you will need to take into account an extra column at the beginning for the:

- status field
- checkbox to perform actions on the listed items through the [PluginListExecute](#) you can safely pre-pend, for instance, a "type:nofilter|" at the begining of the parameters of the tsfilter in such cases.

Table: expand table, number of items found, optional title

Starting with [Tiki20](#), PluginList has a few optional parameters that work in conjunction with the **table** built-in template. Use a `{tableparams}` tag to add these parameters:

- **allowtableexpansion**: for allowing table expansion (a toggle button to allow wide tables to overflow to the right of the content area rather than be scrollable in a limited-width area) (backported to 19.1 and 18.4)
- **shownbitems**: to show the number of items found. (backported to 19.1 and 18.4)
- **title**: whatever string you use here will be shown above the results table, with tiki syntax. The main purpose of this parameter is to allow for the title *not to show* when there are no results found. (backported to 19.2 and 18.4)

See usage example below:

Table showing items count and with an expand button

```
{LIST()} {filter content="1" field="tracker_id"} {OUTPUT(template="table")} {tableparams
allowtableexpansion="y" shownbitems="y" title="!This is the title And this is the subtitle"} {column
label="Book Title" field="book_title"} {column label="Author" field="author"} {column label="Pages"
field="pages"} {column label="Year" field="year"} {OUTPUT}
{FORMAT(name="book_title")}{display name="tracker_field_bookTitle"}{FORMAT}
```

```
{FORMAT(name="author")}{display name="tracker_field_bookAuthor"}{FORMAT}
{FORMAT(name="pages")}{display name="tracker_field_bookPages"}{FORMAT}
{FORMAT(name="year")}{display name="tracker_field_bookYear"}{FORMAT} {LIST}
```

Displaying some tracker fields in a popover

The table output allow to display selected fields in a popover when hovering over the title of an item displayed through the plugin list.

First, from the tracker interface go to tracker properties, display and in the "List detail popup" multi-select field select the fields you want to be display in the popover. Check in the field list the field that is set for item title.

Create your plugin List with mode raw for the title field column (-+mode="raw"+-) and in the format parameter for the same field be sure the format is set to trackerrender (-+format="trackerrender"+-). When hovering over the title it should display the popover.

This code:

Table sample with popover

```
{LIST()} {pagination max="5"} {filter field="tracker_id" content="20"} {filter type="trackeritem"}
{sort mode="modification_date_ndesc"} {OUTPUT(template="table")} {column label="Id"
field="itemid" class="text-left" mode="raw"} {column label="Title" field="title" sort="title"
class="text-nowrap"} {column label="Version" field="version"} {column label="User level"
field="userLevel"} {OUTPUT} {FORMAT(name="itemid")}{display
name="tracker_field_videoresourceVideoId" format="trackerrender"}{FORMAT}
{FORMAT(name="title")}{display name="tracker_field_videoresourceVideoName"}{FORMAT}
{FORMAT(name="version")}{display name="tracker_field_videoresourceTikiVersion"}{FORMAT}
{FORMAT(name="userLevel")}{display
name="tracker_field_videoresourceTikiExperienceLevel"}{FORMAT} {LIST}
```

Should produce (move your mouse over the id in the first column to see the popover) :

Id	Title	Version	User level
2740	How to become translator and translate using the i18n feature	Tiki 19	Intermediate
2741	Comment installer Tiki sur windows	Tiki 19	Beginner
2743	Tiki and Git workflow	Tiki	Developer
		18,Tiki	
		19,Tiki	
		20,Tiki	
		21,Tiki	
		22,Tiki	
		23,Tiki	
		24,Tiki	
		25,Tiki	
		26,Tiki	
		27,master	

Id	Title	Version	User level
2742	Comment installer Tiki avec un environnement développeur sous Windows	Tiki 19	Developer
2735	Contact Management using Trackers	Tiki 18,Tiki 19,any	Beginner

«
1
2 (current)
3
4
5
»

Media List Formatting

[Tiki13+](#)

This sample presents a list of [badges](#) obtained by the current user.

Sample using the default media list formatting
--

```
{LIST()} {filter relation="" objecttype="user" qualifier="tiki.badge.received.invert"}
{OUTPUT(template="medialist")} {icon field="tracker_field_image"} {body
field="tracker_field_description"} {OUTPUT} {LIST}
```

No results for query.

Carousel Formatting

[Tiki14+](#)

Parameters

Parameter	Values	default (empty)
mode	carousel-slide, carousel-fade, ...	carousel-slide
indicators	y, n	y
controls	y, n	y

This sample presents agile gallery of images as a Bootstrap Carousel.

Sample using the default Bootstrap Carousel formatting
--

```
{LIST()} {filter field="gallery_id" content="1"} {OUTPUT(template="carousel")} {carousel
interval="2000" wrap="true" pause="hover" id="mycarousel"} {body field="pic" mode="raw"}
{caption field="caption"} {OUTPUT} {FORMAT(name="pic")}{display format="wikiplugin"
name="wikiplugin_img" fileId="object_id" styleimage="width:100%"}{FORMAT}
{FORMAT(name="caption")} !!! {display name="title" default="Untitled"} {display
name="description" default=""}{FORMAT} {LIST}
```

Carousel parameters

New parameters are added regularly and you can check them by looking at "templates/search/list/carousel.tpl"

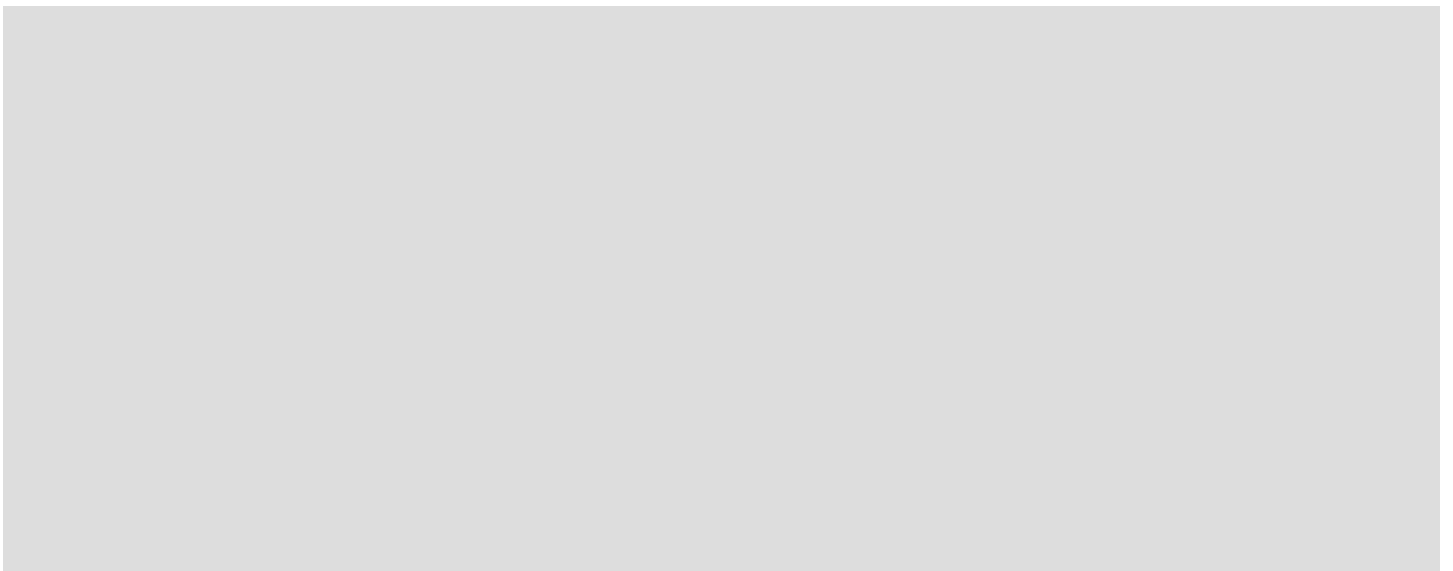
- interval
- pause
- wrap
- indicators (y/n; to hide or show the small indicators below the content)
- controls
- pagination

Example



Fade Example

In Tiki 18 you can use a fade transition by adding `mode="fade"` to the carousel tag, and after Tiki 19+ you should use `mode="carousel - fade"`





Count Formatting

[Tiki15+](#)

This sample shows how to return a simple count of the number of tracker items in tracker 7. For best performance, it is advised to set the max returned results to 1 to prevent having the index to send all the info for all the results (which will slow the request down unnecessarily).

```
{LIST()} {pagination max=1} {filter type="trackeritem"} {filter field="tracker_id" content="7"}  
{OUTPUT(template="count")} {OUTPUT} {LIST}
```

112

Report Formatting

[Tiki20+](#)

This allows you to have specific output format in a wiki page using the CALC extension (extend the wiki plugin List) to execute a formula on the whole result set returned by the list plugin, thus - do aggregations on the data. Useful for reports on more than one tracker item or the whole search result set.

(also see [Calculations Syntax](#))

Example:

```
{LIST()} {filter field="tracker_id" content="45"} {filter type="trackeritem"} {OUTPUT(report="y")}  
There are {CALC()}(count results){CALC} entries. The total amounts are {CALC()} (add (for-each (list  
results) (formula (if (equals tracker_field_trCurrency (str USD)) tracker_field_trAmount 0)))) {CALC}  
USD and {CALC()} (add (for-each (list results) (formula (if (equals tracker_field_trCurrency (str MOP))  
tracker_field_trAmount 0)))) {CALC} MOP. {OUTPUT} {LIST}
```

The new subtotal function aggregates data based on a single column (username in our case) and applies an aggregate function to the rest of the columns (you can choose which ones and what functions to apply). The example aggregates time sheet entries based on the username and applies the 'add' function to the time sheet minutes and 'count' function to the invoice number field. At the end, we have the data in this format:

user1|1250|3

user2|140|1

which is suitable for [PluginFancyTable](#). As you can see in the example, we put the results in a fancytable, so a nice-looking report is done. You can add as many filters to the LIST plugin as you want, so time-constrained data is returned.

```
{LIST()} {filter field="tracker_id" content="22"} {filter type="trackeritem"} {OUTPUT(report="y")}
{FANCYTABLE()} Username|Duration|Invoice {CALC()} (subtotal (list results) (group
tracker_field_TimeSheetWorkerUsername) (aggregate tracker_field_TimeSheetMinutesWorked
tracker_field_TimeSheetInvoiceFromFreelancer) (formula add count) (separators (str |) (str ~nl~)))
{CALC} {FANCYTABLE} {OUTPUT} {LIST}
```

Show the tracker status as a word using "if"

```
{FORMAT( name="stat")}{CALC()} (if (equals tracker_status "o") "open" (if (equals tracker_status "p")
"pending" "closed")){CALC} {display name="tracker_status" format="trackerrender"} {FORMAT}
```

Wiki Formatting (Within Plugin)

For each result, the plugin allows to apply a template. See [PluginList advanced output control block](#).

All the LIST Plugin control blocks

- [PluginList pagination or list control block](#)
 - [PluginList filter control block](#)
 - [PluginList output control block](#)
 - [PluginList format control block](#)
 - [PluginList display control block](#)
 - [PluginList sort control block](#)
 - [PluginList advanced output control block](#)
 - [PluginList multisearch output control block](#)
 - [PluginList aggregate control block](#)
 - [PluginList overview about control blocks parameters and values](#)
 - [Troubleshooting](#)
 - [GUI](#)
 - [Hacks and Fun](#)
 - [PluginList and Metatags - SEO](#)
-