

## Plugin Tracker

### Overview

Use this [wiki plugin](#) to create a form for inputting an item into a tracker. You can specify what fields will be included in the form. Since [Tiki6](#), this plugin can also be used as a simple way to create custom forms which output the result as pre-formatted wiki pages instead of tracker items.

### Parameters

Embed a form to populate a tracker

Introduced in Tiki 1. Required parameters are in **bold**.

[Go to the source code](#)

Preferences required: feature\_trackers, wikiplugin\_tracker

Parameters	Accepted Values	Description	Default	Since
(body of plugin)		Confirmation message after posting form		
<b>trackerId</b>	digits	Numeric value representing the tracker ID		1
<b>ajax</b>		Use ajax to create and update tracker items instead of form submission via request variables.	n	
<b>email</b>		To send an email once the tracker item has been created. Format: <code>from to template</code> For <code>from</code> and <code>to</code> , use an email address (separate multiple addresses with a comma), a username, a fieldId of a field containing either an email address or a username, a fieldId of a UserSelector or GroupSelector field, or "createdBy" or "lastModifBy" for the item creator or modifier. When username is being used, the email will be sent to the email address of the user on file. When sending to several emails using different template, provide the template name for the message body for each email; I.e., the first template will be used for the first to, the second template if exists will be used for the second from (otherwise the last given template will be used). Each template needs two files, one for the subject one for the body. The subject will be named <code>template_subject.tpl</code> . All the templates must be in the <code>templates/mail</code> directory. Example: <code>webmaster@my.com a@my.com,b@my.com templatea.tpl,templateb.tpl</code> ( <code>templates/mail/tracker_changed_notification.tpl</code> is the default from which you can get inspiration).		2.0
<b>emailformat</b>		Choose between values text or html, depending on the syntax in the template file that will be used	text	6.1
<b>preview</b>		To add a preview button with the label set by this parameter. Default: <code>Preview</code> . Useful to preview the computed fields of an item.	Preview	2.0
<b>reset</b>		Label for the reset button, to return all fields to their default values.	reset	4.2
<b>tpl</b>		Name of the template used to display the tracker items. In the template, the smarty variable <code>{ \$f_id }</code> will be replaced with the appropriate input tag, with <code>id</code> representing the field ID. The form tag and the submit button are generated by Tiki outside the template		2.0
<b>values</b>		Colon-separated list of default values corresponding to the <code>fields</code> parameter. First value corresponds to first field, second value to second field, etc. Default values can be set by using <code>autosavefields</code> and <code>autosavevalues</code> as URL parameters.		2.0
<b>action</b>	<code>separator: :</code>	Colon-separated labels for form submit buttons. Default is <code>Save</code> . When set to <code>NONE</code> , the save button will not appear and values will be saved dynamically.	Save	1
<b>action_style</b>	<code>separator: :</code>	Sets button style classes for action buttons. If multiple buttons have been set in the <code>action</code> parameter, the same number of colon-separated styles must be set here. Example: <code>btn btn-secondary:btn btn-success:btn btn-primary float-end</code>	btn btn-secondary	14.1
<b>rulesparent</b>	text	JQuery selector for the parent object to show or hide when executing field rules.	.tracker-field-group:first	

<code>fieldtofilter</code>		Field to filter user tracker items to: <code>view="user&amp;field"</code>		15.3
<code>fieldtofiltervalue</code>		Value to filter user tracker items to: <code>view="user&amp;field"</code>		15.3
<code>fieldsfilldefaults</code>		Colon-separated list of default values for Multiple Fill Fields.		9.0
<code>fieldsfillseparator</code>		Choose separator between fields in each line of the Multiple Fill text area. Default is pipe ( <code> </code> ).		9.0
<code>fieldsfilldescription</code>	text	Explanatory text below the Multiple Fill Fields text area. Default is the list of the fields which should be in each line. Put a space for no visible text below the Multiple Fill Fields text area.		24.4
<code>fieldsfilllabel</code>	text	Explanatory text on the left of the Multiple Fill Fields text area. Default is "Insert one item per line:".		24.4
<code>chosenGroup</code>	text	The user enters this group via the registration (only a single group name is supported)	Registered	15.0
<code>status</code>		Status of the item used in combination with: <code>view="user"</code> or <code>view="user&amp;field"</code>		6.0
<code>userfieldtofilter</code>		User field to filter user tracker items to: <code>view="user&amp;field"</code>		15.3
<code>fieldtofiltercriteria</code>	(blank) creationAsc creationDesc	If more than one item found, will choose under this criteria. Used in combination with: <code>view="user&amp;field"</code>		15.3
<code>showmandatory</code>	(blank) y n	Indicate mandatory fields with an asterisk (shown by default).	y	1
<code>newstatus</code>	(blank) o p c	Default status applied to newly created or saved items.		2.0
<code>outputwikinamespace</code>	pagename	Name of namespace that is used for the wiki page that is created when outputting to a wiki page.		13.0
<code>registration</code>	(blank) y n	Add registration fields such as Username and Password for use in registration trackers	n	6.0
<code>showdesc</code>	(blank) y n	Show the tracker's description (not shown by default)	n	1
<code>showfieldsdesc</code>	(blank) y n	Show the tracker's field descriptions (shown by default)	y	12.1
<code>showstatus</code>	(blank) y n	Show the status of the items (not shown by default)	n	5.0
<code>showtitle</code>	(blank) y n	Display the title of the tracker at the top of the form (not shown by default)	n	1
<code>sort</code>	(blank) y n	Display columns in the order listed in the fields parameter instead of by field ID (field ID order is used by default)	n	2.0
<code>transactionStep</code>	digits	Transaction step number specifying the order of the transaction steps. The first step must be .	0	15.0
<code>transactionName</code>	alpha	The transaction identifier. This identifier connects the various trackers into a single transaction. Must be unique per transaction. The multiple steps in a single transaction must share the same transaction name.		15.0
<code>transactionPreviousURL</code>	url	The page to go back to when the "previous" button is clicked.		18.4
<code>transactionPreviousLabel</code>	text	Text for the "previous" button label.		18.4

<code>url</code>	url <i>separator:</i> :	URL the user is sent to after the form is submitted. The string <code>itemId</code> will be replaced with <code>itemId=xx</code> where <code>xx</code> is the new (or current) itemId. This parameter can be used in combination with the TABS plugin using <code>?cookietab=1&amp;itemId</code> (change the cookietab parameter's value depending on the tab you want the user to be sent to).		1
<code>target</code>	(blank) _blank _parent _self _top	Set the target parameter for the url (determines whether target will open in a new page, etc.)		4.0
<code>rules</code>	(blank) y n	Set up rules on field definitions to show and hide fields conditionally.	n	
<code>validateusers</code>	(blank) y n	Here one can overrule the default validate users by e-mail preference.		15.0
<code>view</code>	(blank) group page user userandfield	Determine which items will be affected by the form. If set to <code>user</code> and <code>trackerId</code> is not set, then the user tracker associated with the default group will be affected. If <code>trackerId</code> is set, then the item associated with the user in that tracker will be affected. If set to <code>page</code> , the item associated with that page will be affected ( <code>trackerId</code> must be set in this case).		1
<code>colwidth</code>	## or ##%	Specify the width in pixels or percentage of the first column (the labels) in the tracker form.		3.0
<code>wiki</code>	pagename	Name of the wiki page containing the template to display the tracker items. This page must have the permission <code>tiki_p_use_as_template</code> assigned to the Anonymous group to be used as a template.		2.0
<code>autosavevalues</code>	text <i>separator:</i> :	Colon-separated values corresponding to <code>autosavefields</code> . Special syntax cases: <code>categories(x)</code> - selects the first child category under a category with ID <code>x</code> for use in a category field <code>category(x)</code> - selects a category with this ID <code>x</code> for use in a category field <code>preference(x)</code> - inserts the value of the preference with <code>x</code> being the preference name.		5.0
<code>discarditem</code>	(blank) y n	Used with <code>outputtowiki</code> - whether to discard the tracker item itself once the wiki page is created, so that, in effect, the tracker is just a vehicle to create form fields to facilitate creating wiki pages.		6.0
<code>formtag</code>	(blank) y n	If set to Yes (y), the tracker is contained in a <code>form</code> tag and has action buttons	y	6.4
<code>embedded</code>	(blank) y n	Embedded	n	1
<code>transactionFinalStep</code>	(blank) y n	Indicate whether this is the final transaction step		15.0
<code>ignoreRequestItemId</code>	(blank) y n	Do not filter on the parameter <code>itemId</code> if in the url (default is to filter)	n	6.0
<code>urlparams</code>	url <i>separator:</i> :	Input form parameters (or field IDs) separated by a colon (:) to pass them in the URL after the form is submitted and redirected to the URL specified in the <code>url</code> param. Example: <code>urlparams="1:2:3:trackit"</code> . Enter * to keep all the submitted parameters in the URL.		19.0
<code>overwrite</code>	(blank) y n	Overwrite current field values of the item with the input values. Does not overwrite wiki pages and does not work when the <code>discarditem</code> parameter is set to Yes (y).	n	6.0

<code>outputwikirelation</code>	(blank) y n	Store <code>tiki.wiki.linkeditem</code> and <code>tiki.wiki.linkedfield</code> relation from the created wiki page when outputting to a wiki page. Optionally, (separate feature to be turned on in admin panel) these relations are used to sync page renames with the field specified in <code>outputtowiki</code> , and also optionally to redirect page viewing to the tracker item instead (where you can then include the page if needed).	n	13.0
<code>outputwiki</code>	pagename	Name of the wiki page containing the template to format the output to wiki page. Must be set for <code>outputtowiki</code> to work. The template can contain variables to represent fields, for example <code>{f_6}</code> would result in the value of fieldId 6. Also <code>{f_itemId}</code> can be used for the itemId and if you have set <code>register</code> to yes, you can use <code>{register_login}</code> and <code>{register_email}</code> .		6.0
<code>fields</code>	<i>separator:</i> :	Colon-separated list of field IDs to be displayed in the form as input fields. If empty, all fields will be shown. Example: <code>2:4:5</code>		1
<code>itemId</code>	digits	ItemId identifying the item to be edited.		3.0
<code>fieldsfill</code>	<i>separator:</i> :	Colon-separated list of field IDs to be filled with multiple values, to create multiple items in one save. If empty, only one item will be created. Only for item creation. Example: <code>2:4:5</code>		9.0
<code>outputtowiki</code>	digits	Output result to a new wiki page with the name taken from the input for the specified fieldId		6.0
<code>autosavefields</code>	digits <i>separator:</i> :	Colon-separated list of field IDs to be automatically filled with values upon save.		5.0
<code>levelupfields</code>	digits <i>separator:</i> :	Used with the <code>autosavefields</code> and <code>autosavevalues</code> parameters. Colon-separated list of field IDs being auto-saved where the specified auto-save value will not take effect if it is less than or equal to the current value of the field		8.0

## Administration

- You can limit the tracker fields that can be used in this plugin in the field admin panel of the tracker: "Field is public? for use with [trackerlist plugin](#) option  
For Tiki 9.x, the User Selector field (the "userId" field in your tracker, perhaps), edit the field, Permissions, Visibility, Visible by All (even though that's counter-intuitive (you want those items private to that use)).
- If you have a tracker that has only one item per user/IP or if you use the plugin with the param `view=user`, the plugin will automatically prefill the item if that already exists.

## Examples

```
{TRACKER(trackerId="1" fields="1:3:4" action="Send")} Thanks for your contribution! {TRACKER}
```

## How to pre-fill some fields

In some situation you want to go to a TRACKER page (in our example named SubmitOrder) where fields are automatically pre-filled. In this case you can use an URL like this:

```
tiki-index.php?page=SubmitOrder&autosavefields=1&autosavevalues=tikiwiki
```

When in the page SubmitOrder, you click on Save, and the field 1 will be filled with the value `tikiwiki`. If the field 1 is multivalue field (ex: category), the value will be added to the potentially additional value the user has selected whatever the user does.

### Tip:

The construction of such URL can use the feature 'Wiki argument variables'

(see [Advanced Wiki Syntax usage examples](#))

```
tiki-index.php?page=SubmitOrder&autosavefields=2:3&autosavevalues={f_{itemId}}:{f_{page}}
```

You could also use the *url* parameter "**prefills**" with the corresponding "**values**". See more information [here](#):

- [PluginTracker Advanced](#)
- [PluginTrackerPrefill](#)

See also [PluginTrackerList](#) to see how to use the parameter `url` to transfer an `itemId` to an URL or [PluginJQ](#) if you want to use jQuery.

### How to fill an item in more than one step

If you want to fill some fields of an item in a page and to fill some other ones in another page, you need to use the param *url* with the *itemId* option.

Example (uses [SEFUrl](#) enabled links):

Page1

---

```
{tracker trackerId="1" fields="1:2:3" url="Page2?itemId"}
```

Page2

---

```
{tracker trackerId="1" fields="4:5:6" url="Page3?itemId"}
```

Page3 - the final one that will display the currently submitted *itemId*:

---

```
{trackerlist trackerId="1" fields="1:2:3:4:5:6"}
```

It is also possible to use tabs and link to each tab using "cookietab".

Note that the tracker item will be saved by sequence so you better check that no mandatory field rules are on the way.

---

```
{TABS(tabs="First Tab|Second Tab|Third Tab")} {tracker trackerId="1" fields="1:2:3" action="Next" url="pageName?cookietab=2&itemId"} ///// {tracker trackerId="1" fields="4:5:6" action="Next" url="pageName?cookietab=3&itemId"} ///// {TRACKER(trackerId="1" fields="7:8:9" action="Save")}{Saved!}{TRACKER} {TABS}
```

### How to put multiple tracker plugins into a single transaction?

#### *New in Tiki15*

Plugin tracker's transaction feature enables a sequence of trackers to be chained into a single transaction, which is submitted only after the user submits the last tracker form. Otherwise the transaction is cancelled.

The easiest way to use the feature is to create a wiki page for each of the steps.

Here is an example.

#### 1. wiki page: Register+Step+1

---

```
{tracker trackerId="1" fields="2" action="Next" ignoreRequestItemId="y" registration="y" chosenGroup="Registered" transactionName="RegistrationTransaction" transactionStep="0" transactionFinalStep="n" url="Register+Step+2"}
```

## 2. wiki page: Register+Step+2

---

```
{tracker trackerId="2" fields="4" action="Finish" ignoreRequestItemId="y"
transactionName="RegistrationTransaction" transactionStep="1" transactionFinalStep="y"}
```

The registration and the tracker records of `trackerId` 1 and 2 will only get saved after the user presses Finish on Register+Step+2.

How to write a template to display the form elements on the wiki page where the plugin is saved

As mentioned, in a template, use the `{$_id}` syntax to insert the value of *fieldId*=**id**.

It is also possible and nowadays recommended to use the `{$_permanentname}` syntax to insert the value of **"permanent name"** of the tracker field **instead of the id**.

It is also recommended to use a specific custom prefix for all fields of a tracker (one prefix per tracker) to allow to reuse and to provide existing trackers easily with a profile. Syntax:

`{$_prefix_permanentname}`. The prefix can be set in the tracker preference settings.

It is also possible to insert the *ItemId* itself, which is useful to create new wiki pages with tracker plugins inside. To do this you'll have to add an [Auto-Increment Field](#) to your tracker, with the appropriate options to auto-fill it with the *itemId*. This new field content (*itemId* value) can then be inserted in the template with the syntax seen above.

Since [Tiki 15.1](#), wiki page templates can be (optionally) used for email notifications using the **"wikiage name tpl"** format instead of `template_name.tpl` files in the `templates/mail` dir. Tiki will use a page "page name subject tpl" if found.

Please mind, that wiki pages are more for quick and easy use cases, for testing and for proof of concept, whilst `.tpl` files in the server directory are more reliable and much better to handle in larger projects or for advanced use cases.

Mind: `.tpl` templates for emails are stored in the folder `templates/mail`

How to go from TRACKERLIST to TRACKER on this item

---

```
{trackerlist trackerId="1" fields="1:2:3:4:5:6" url="Page1?itemId"}
```

### Email

How to send email to a recipient when new tracker item is created

You can use the parameter `email` to send an email once the tracker item has been created.

The format for the email parameter is: from|to|template.

- **from** is the sender email that will be used for the notification. Multiple addresses can be set (separate multiple addresses with a comma)
- **to** is the recipient(s) to whom the email address will be sent.  
Multiple addresses can be set (separate multiple addresses with a comma).  
Since Tiki 11 (maybe earlier) you can set a tracker field (same item) being used in the tracker plugin template (required).
- **template** is the template(s) to be used for email subject and body. (separate templates can be used)  
Using several emails and several templates it is possible to send different email (different subject & body) to different people. (see examples below)
- The following values can be used for **from** and **to**:  
**email(s)**, **username(s)**, **fieldId(s)** or **createdBy**

`createdBy` was added in [Tiki 17](#) and might be back-ported to 16.

`username` sends to the email of the registered user account and `createdBy` sends to the email of the currently logged in creator of the item.

Since [Tiki 15.1](#), Tracker plugin allows [Items List fields](#) to be used as the **to** or **from** fields in the email parameter. They will be shown in the form as read only labels. This might be useful in cases such as:

- One tracker defines business shops, with its members to be notified of changes related to that shop. And...
- A second tracker handles data of products from shops, and they have:
  - one field as *Item link* to relate the product with their corresponding shop where they sell it.
  - another field from the second tracker is an *Items list*, to indicate the person to be notified, as user defined in the first tracker for that shop.
- You use the email parameter in a plugin tracker call to tracker 2, using the field corresponding to the person (as field id in tracker 2) as the recipient or sender of the email.

Since [Tiki 16.3](#), email recipients option allows to set fieldId of a [UserSelector](#) (including multiple users selected) and [GroupSelector](#).

Basic plugin usage

**single recipient, single template**

---

```
{TRACKER(trackerId="1" fields="1:2:3" action="Send"
email="sender@domain.com|recipient@domain.com|templatename.tpl")}{TRACKER}
```

**single recipient (creator), default site sender, basic default template (for quick proof of concept)**

---

```
{TRACKER(trackerId="15" action="Send" email="|createdBy|")}
```

 Thank you for submitting an issue.  
We will come back to you as soon as possible. What do you want to do now: ((Issue Tracker|submit another issue)) or ((HomePage|browse the website))? {TRACKER}

**single recipient (creator), custom sender, custom template for body, default basic subject**

---

```
{TRACKER(trackerId="15" action="Send" email="info@example.com|createdBy|mytemplate.tpl")}
```

 Thank you for submitting an issue. We will come back to you as soon as possible. What do you want to do now: ((Issue Tracker|submit another issue)) or ((HomePage|browse the website))? {TRACKER}

**single recipient (creator), custom sender, custom template for body, custom subject template**

---

```
{TRACKER(trackerId="15" action="Send"
email="info@example.com|createdBy|mytemplate.tpl,mytemplate_subject.tpl")}
```

 Thank you for submitting an issue. We will come back to you as soon as possible. What do you want to do now: ((Issue Tracker|submit another issue)) or ((HomePage|browse the website))? {TRACKER}

### multiple recipients, single template

---

```
{TRACKER(trackerId="1" fields="1:2:3" action="Send"
email="sender@domain.com|recipient_1@domain.com,recipient_2@domain.com,recipient_3@domain.c
om|templatename.tpl")}{TRACKER}
```

In this case it will send the same notification to recipient\_1, recipient\_2 and recipient\_3.

### multiple recipients, multiple templates

---

```
{TRACKER(trackerId="1" fields="1:2:3" action="Send"
email="sender@domain.com|recipient_1@domain.com,recipient_2@domain.com,recipient_3@domain.c
om|templatename_a.tpl,templatename_b.tpl,templatename_c.tpl")}{TRACKER}
```

In this case each recipient will receive a different notification. The recipient\_1 will receive email using template\_a, recipient\_2 will receive email using template\_b and recipient\_3 will receive email using template\_3.

### creator plus second recipient, two different body templates and two different subjects

---

```
{TRACKER(trackerId="15" action="Send"
email="info@example.com|admin@mysite.com,createdBy|admin-template.tpl,creator-
template.tpl,admin-template_subject.tpl,creator-template_subject.tpl")}{ Thank you for submitting an
issue. We well come back to you as soon as possible. What do you want to do now: ((Issue
Tracker|submit another issue)) or ((HomePage|browse the website))? {redirect page="Test-Tracker-
Creator-Email"} {TRACKER}
```

### single recipient (field), custom sender, custom template to fill the tracker plugin, custom template for body, custom subject template

Wiki page

---

```
{TRACKER(trackerId="15" action="Send" email="info@exemple.com|12|mytemplate.tpl"
wiki="Template page")}{Thank you!}{TRACKER}
```

Template page

---

```
{TR()}{Name}{TR}: {$f_11} {TR()}{Email address}{TR}: {$f_12}
```

When filling the form (using the template) the email address from the field 12 will be used as recipient.



## Templates (to create the email subject and email body)

Each template needs two files, one for the subject one for the body. All the templates must be in the templates/mail directory or (since Tiki15.1) in wiki pages and specified as "**wiki** **page name tpl**" (you have to include the "**tpl**" exact word in the name without a dot, and place it, for instance, at the end).

### Email body and subject naming conventions

The naming of these files needs to be slightly different depending on whether you use template files on disk or wiki pages as templates.

## Template files on disk

If you want to use smarty template files on disk:

- body: "**mytemplate.tpl**"
- subject: "**mytemplate\_subject.tpl**"

Example:

---

```
{TRACKER(trackerId="1" fields="1:2:3" action="Send"
email="sender@domain.com|recipient@domain.com|mytemplate.tpl")}{TRACKER}
```

Note that you don't specify the subject template in the tracker plugin call, but the subject needs to exist with the corresponding name ("mytemplate\_subject.tpl" in this case) if you want to have the subject message customized.

## Wiki pages as Smarty templates

If you want to use wiki pages as smarty templates:

- body: "**page name tpl**"
- subject: and "**page name subject tpl**"  
note that the tpl word (without a dot) needs to be prepended with the word "**subject**" (subject **and a space**)

Example:

---

```
{TRACKER(trackerId="1" fields="1:2:3" action="Send"
email="sender@domain.com|recipient@domain.com|wiki:page name tpl")}{TRACKER}
```

Note that you don't specify the subject template in the tracker plugin call, but the subject needs to exist with the corresponding name ("page name subject tpl" in this case) if you want to have the subject message customized.

### Subject template

---

```
Hello, the {tr}item{/tr} "{$f_name}" {tr}was modified at{/tr} {$server_name} in
{$mail_trackerName}
```

Where `{ $f_name }` will be the content of the field name, `{ $server_name }` is the domain name and `{ $mail_trackerName }` is the tracker name.

## Body template

---

```
{tr}Hello,{/tr} {tr}Title:{/tr} { $f_name } {tr}Description:{/tr} { $f_description } {tr}View the tracker item at:{/tr} { $mail_machine_raw }/tiki-view_tracker_item.php?itemId= { $mail_itemId }
```

## Template - variable list

Not all wiki variables work (pretty tracker syntax won't work).

`$mail_date`

The current timestamp

`$mail_user`

User login

`$mail_itemId`

The tracker itemId

`$mail_item_desc`

The item "title"

`$mail_trackerId`

Tracker id

`$mail_trackerName`

Tracker name

`$mail_machine`

URL of the request (example: `https://doc.tiki.org/PluginTracker` )

`$mail_machine_raw`

URL of the site root (example: `https://doc.tiki.org` )

`$server_name`

Server name (example: `doc.tiki.org` )

`$status`

The item's new status (e.g. Open, Pending or Closed, translated)

`$f_123` or `$f_permanentName`

Values of the fields

## Related pages

See also:

- [PluginTrackerlist](#) - which displays a given tracker items.
- [PluginTracker Advanced](#)

- [Pretty Tracker](#)
- [PluginList](#)