Plugin List Execute

Introduced in Tiki11, the LISTEXECUTE plugin is an extensible plugin allowing administrators to set-up listings and custom actions that can be executed on objects, allowing to support complex workflows and perform some level of automation.

The LISTEXECUTE plugin uses the Search and List from Unified Index to extract lists of objects. All filters available in PluginList can also be used with LISTEXECUTE. Results are presented as a list with checkboxes for each row. Actions can be performed on selected objects.

Parameters

Set custom actions that can be executed on a filtered list of objects

Introduced in Tiki 11.

Go to the source code

Preferences required: wikiplugin_listexecute, feature_search

Parameters
(body of plugin) - List configuration information
no parameters

Sample usage

{LISTEXECUTE()} {filter type="trackeritem"} {filter content="p" field="tracker_status"} {ACTION(name="Expire Issue" group="Triage Team")} {step action="change_status" from="p" to="c"} {step action="email" subject="Issu closed" content_field="email_content" to_field="tracker_field_senderEmail"} {ACTION} {FORMAT(name="email_content")} Hello, We are sorry to announce that we have closed issue #{display name="object_id"} automatically. Summary: {display name="tracker_field_summary"} The issue has been pending additional information for too long. Feel free to answer pending questions and re-open it. -- Acme Software Team {FORMAT} {LISTEXECUTE}

As you can see, the plugin uses a structure of fake plugins to define the behavior.

- Filters are used to select which objects will be displayed
- FORMAT is used to defined composed blocks of content, the email body in the above example
- ACTION defines sequences of steps to be performed. The name is required. Group

name is optional.

- NB: Only one group name can be indicated per action. If two or more groups are to be allowed to execute the action, duplicate the action, having one action per group as in the snippet below:

Multiple groups

```
{ACTION(name="Close") group="Site Managers"} {step action="change_status" from="o" to="c" } {ACTION}
{ACTION(name="Close") group="Triage Team"} {step action="change_status" from="o" to="c" } {ACTION}
```

Each step must defined the action attribute, which is associated with a specific behavior. Each action defines a set of parameters it can accept. The value for those parameters can be set statically in the definition or taken dynamically from the search results, or custom FORMAT blocks by using the `_field` suffix.
In the example, the `content` parameter for the `email` action is set using `content_field` to `email_content` , which is defined in the FORMAT block. The `subject` parameter is set directly, but it would be possible to define a custom, dynamic subject using `subject_field` . In the following snippet, the summary field is set directly into the email `subject_field` :

Sample usage: custom subject

```
{ACTION(name="Expire Issue" group="Triage Team")} {step action="change_status" from="p" to="c"} {step action="email" subject_field="subject_content" content_field="email_content" to_field="tracker_field_senderEmail"
{ACTION} {FORMAT(name="subject_content")} Summary: {display name="tracker_field_summary"} {FORMAT}
```

Additionally, multiple fields can be provided.

- Using the `_field_coalesce` suffix: A comma-separated list of fields can be provided. The first non-empty field will be used.
- Using the `_field_multiple` suffix: A comma-separated list of fields can be provided. All

non-empty fields will be provided. When used on a field that does not allow multiple values, the behavior will be the same as using coalesce.

Actions

To set an action as `default`

To set the default action to be executed, you need to use the `default` parameter, which can be set to either `'y'` or `'1'` (as shown in the example below). The action set as default will automatically be selected from the list of actions. However, using `default` **is not mandatory**.
Any action that does not have the `default` parameter, or has it set to something other than `'y'` or `'1'`, will automatically be considered as not set as default (e.g., default='yes', default='Y', default='default', default='n', default='0', and so on).

Sample usage: setting an action as default using y

```
{LISTEXECUTE()} {pagination max=25} {filter type="trackeritem"} {filter content="6" field="tracker_id"} {filter content="o" field="tracker_status"} {ACTION(name="change tracker item status from Open to Pending" group="Registered" default='y')} {step action="change_status" from="o" to="p"} {ACTION} {ACTION(name="change tracker item status from Open to Closed" group="Registered")} {step action="change_status" from="o" to="c"} {ACTION} {LISTEXECUTE}
```

In this example, `default='y'` means that the action named `change tracker item status from Open to Pending` will be selected by default. You can also set default to `'1'`, as shown below:

Sample usage: setting an action to default using 1

```
{LISTEXECUTE()} {pagination max=25} {filter type="trackeritem"} {filter content="6" field="tracker_id"} {filter content="o" field="tracker_status"} {ACTION(name="change tracker item status from Open to Pending" group="Registered" default='1'))} {step action="change_status" from="o" to="p"} {ACTION} {ACTION(name="change tracker item status from Open to Closed" group="Registered")} {step action="change_status" from="o" to="c"} {ACTION} {LISTEXECUTE}
```

You understand that logically, only one action should be set as default, but if multiple actions are set as default, the last one in descending order (from top to bottom) will be selected as default.

What to do:

- The default action can be set on any action: on the first, on one in the middle, or on the last action.

Sample usage: setting an action to default anywhere

{ACTION(name="change tracker item status from Open to Pending" group="Registered")} {step action="change_status" from="o" to="p"} {ACTION} {ACTION(name="change tracker item status from Open to Closed" default="1" group="Registered")} {step action="change_status" from="o" to="c"} {ACTION}

NB :

You can use `'` character or `"`, it means 'y' equals "y"

What not to do:

- Do not set default to a value other than `'y'` or `'1'`. Example:

Sample usage: setting an action as default anywhere

{ACTION(name="change tracker item status from Open to Pending" group="Registered")} {step action="change_status" from="o" to="p"} {ACTION} {ACTION(name="change tracker item status from Open to Closed" group="Registered" default='yes')} {step action="change_status" from="o" to="c"} {ACTION}

- Do not set multiple actions as default; otherwise, only the last one will be considered:

Sample usage: setting multiple actions as default

{ACTION(name="change tracker item status from Open to Pending" default='y' group="Registered")} {step action="change_status" from="o" to="p"} {ACTION} {ACTION(name="change tracker item status from Open to Closed" default='y' group="Registered")} {step action="change_status" from="o" to="c"} {ACTION}

In this case, the action named `change tracker item status from Open to Closed` will be considered the default!

Fields with a `+` sign allow multiple values to be provided using the `_field_multiple` suffix.

change_status

Applies to tracker items. Allows to change the status from one state to an other.

- `object_type` : Taken directly from the search results
- `object_id` : Taken directly from the search results
- `tracker_status` : Taken directly from the search results. Used to validate the initial state before attempting to change the status.
- `from` : Must be set, defined the initial state (o=open, p=pending, c=closed)
- `to` : Must be set, defined the final state (o=open, p=pending, c=closed)

Sample usage: closing items

{LISTEXECUTE()} {filter field="tracker_id" content="3"} {filter categories="2"} {filter field="tracker_status" content="o"} {OUTPUT(template="table")} {column label="Date" translatelabel="y" field="date"class="text-nowrap"} {column label="Moment" translatelabel="y" field="moment" mode="raw" class="text-nowrap"} {column label="Member" field="member" class="text-nowrap"} {OUTPUT} {FORMAT(name="date")}{display name="tracker_field_momentDate" format="date"}{FORMAT} {FORMAT(name="moment")}{display name="tracker_field_momentMomentLabel_en" format="objectlink"}{FORMAT} {FORMAT(name="member")}{display name="tracker_field_momentFamilyMember_text"}{FORMAT} {ACTION(name="Close")} {step action="change_status" from="o" to="c"} {ACTION} {LISTEXECUTE}

delete

Removes the object from Tiki.

- `object_type` : Taken directly from the search results
- `object_id` : Taken directly from the search results

Supported object types:

Type	Version
file	Tiki12
trackeritem	Tiki18
aggregate	Tiki18 - see PluginList for more info on aggregating results

Sample usage: delete

```
{LISTEXECUTE()} {filter type="trackeritem"} {filter field="tracker_id" content="4"} {filter content="c" field="tracker_status"} {OUTPUT(template="table")} {column label="First Name" field="contactFirstName"} {column label="Last Name" field="contactLastName"} {OUTPUT} {FORMAT(name="contactFirstName")}{display name="tracker_field_contactFirstName" default=" " }{FORMAT} {FORMAT(name="contactLastName")}{display name="tracker_field_contactLastName" default=" "}{FORMAT} {ACTION(name="Delete closed items" group="Registered")} {step action="delete"} {ACTION} {LISTEXECUTE}
```

email

Sends an email to the specified recipient.

- `subject` : Must be set.
- `to_field` (if email in a field of the tracker item) or `to` (if email in plain text) must be set as the recipient of the email message. Since Tiki18.4, you can indicate a user field in the `to_field`

Optional fields:

- `pdf_page_attachment` : Since Tiki17, to be able to send pdf versions of wiki pages as email attachments.
- `from_field` (or `from`): Since Tiki18, to be able to set the **from** field of the email based or the email associated to a user field in the tracker
- `cc_field` (or `cc`): Since Tiki18, like the from_field above but for the **cc** of the email
- `bcc_field` (or `bcc`): Since Tiki18, like the from_field above but for the **bcc** of the email
- `is_html`: Allow html content for the email body

If email value comes from field that are itemLink, itemList or DynamicItemList type you'll need to add the parameter _text at the end of the permname.

---

tracker_field_email_text instead of tracker_field_email

filegal_image_overlay

Added in Tiki17. Overlays some text on top of jpeg images attached to a tracker item which are stored in a file gallery

Requirements:

Ubuntu/Debian sudo apt-get install imagemagick php-imagick sudo service apache2 restart # CentOS (install as root) yum install ImageMagick yum install [--enablerepo=remi] php-pecl-imagick service httpd restart

Parameters:

- `field` : Must be set. Permanent name of the tracker field where the file was uploaded to/selected in.
- `value` : Must be set. Some key(s) from the available list below.

Replacement keys for filegal_image_overlay:

%file_name% %file_id% %parts_filename% %parts_extension% %gallery_name% %gallery_id% %tracker_id% %item_id% %field_id% %field_perm_name% %field_name% %exif_date% %exif_gps% %exif_gps_lat% %exif_gps_lon% %exif_gps_dms% %exif_gps_dms_lat% %exif_gps_dms_lon%

Example of usage:

{step action="filegal_image_overlay" field="_2PhotoAttachments" value="%exif_date% - %exif_gps%"}

filegal_change_filename

Added in Tiki17. Renames files attached to a tracker item which are stored in a file gallery.

- `field` : Must be set. Permanent name of the tracker field where the file was uploaded to/selected in.
- `value` : Must be set. Some key(s) from the available list below.
- `in_place+` : Must be set. y/n. If y, the renamed file replaces the original file. If n, a new file version is created with the new filename, leaving the source file intact.

Replacement keys for filegal_change_filename:

%file_name% %file_id% %parts_filename% %parts_extension% %gallery_name% %gallery_id% %tracker_id% %item_id% %field_id% %field_perm_name% %field_name%

Example of usage:

{step action="filegal_change_filename" field="_2PhotoAttachments" value="%item_id%-%file_id%-%file_name%" in_place="y"}

File and Email

Emailing a link to the file is not currently possible via ItemsList field and it may be not advised (due to security and anti-spam consideration). Instead, you can insert a link to the file with the "&display" parameter to display an image o without to initiate a download (if required user must be logged), something like this to the email content:

http://your-tiki-site/tiki-download_file.php?fileId={display name="tracker_field_picture"}&display

You may also consider to insert a link that will display your image or file or any other information. (see below)

Dynamic Link in the Email

Thanks to Wiki Argument Variables it is possible to include a link to a page to display part of the information related to

the object_id or any wiki dynamic variable you set and that will display filtered tracker item information. It can also inforce security or add validation options (page to display the data may require login and it can be tracked)
IE: You want to display a member ID card that include text (name, adresse and a photo).

1. Enable "Wiki argument variables", Control Panel; Editing and Plugins; Wiki syntax (tiki-admin.php?page=textarea)
2. Create a page that will be used as template with a plugin List.
    - Set a plugin List with all the information you want to display including sorting and filtering
    - Add a filter on the variable like {filter field="object_id" content=""} (you can name it how you want)
3. Set your plugin ListExecute
    - Insert in the email content an url that will set the Wiki Argument Variables and send the value to your template to display the specific member item fields:

```
http://domain.com/memberIdCard?memberId={display
name="tracker_field_memberid"}
```

This is one example and there are many ways for you to use it.

snapshot

Added in Tiki23: http://sourceforge.net/p/tikiwiki/code/78296

tracker_item_clone

Added in Tiki22 New action tracker_item_clone has been added.
This gives the ability to clone one or more selected tracker items.
Cloning tracker items via ListExecute keeps auto-assigned UserSelector values instead of resetting that to none or current user.
The function was enhanced to allow emptying fields and to set a field with default value (like a date field)
It is related to the "Duplicate tracker items" option (was "Clone tracker items" before Tiki25) and use the same code.
You can set the option in the Settings, Trackers control panel.
This code is also checking if the item has children through item link, if items in other trackers are linked to the parent item using an item link tracker field. If items are found **they are also duplicated** silently. See Trackers; Duplicate a trackeritem on an item used in another tracker will create duplication for all the items where the original item is used

Cloning tracker item and resetting selected fields

In this sample we clone an item and modify some values of the new created item;

- Status is changed to "pending"
- Checkbox (productsboughtEmailSent, a flag) is reseted to 'n' using calc.
- Checkbox (productsboughtSMSSent, a flag) is reseted to 'n' using calc.
- Relation field (productsboughtPaymentId) is emptied
- Date field (productsboughtDate) is reseted to the default (today)

Duplicate an item, change status, reset productsboughtDate to today (default) and reset the values of 3 fields using calculations and empty value

{ACTION(name="Duplicate to date and reset fields")} {step action="tracker_item_clone"} {step action="change_status" from="o" to="p"} {step action="tracker_item_modify" field="productsboughtEmailSent" calc="(str n)"} {step action="tracker_item_modify" field="productsboughtSMSSent" calc="(str n)"} {step action="tracker_item_modify" field="productsboughtPaymentId" value=" "} {step action="tracker_item_modify" field="productsboughtDate" calc="(date)"} {ACTION}

tracker_item_modify

Modifies a single field out of a tracker item.

×

⚠ Performance and field checking

To preserve performance and usability the previous and nature of the fields are not continuously checked. When you add a value using this method it is added as hardcoded string without checking the parameters and options of the existing field.

For exemple, if you have a dropdown field with specified values (value1, value2 and value3) it is possible to insert a value that is not in the existing options (value4) without warning or error about it. This can lead to weird behaviour if you have other tools (search, filtering, relations, etc.) depending on the value of this field.
This is advanced usage and it is expected that you know what you are doing (still triple check for typos. □)

- `object_type` : Taken directly from the search results
- `object_id` : Taken directly from the search results
- `field` : Must be set, permanent name of the field to modify, **without the "tracker_field_" prefix** at the permanent name.
    - This is counterintuitive since elsewhere in the list plugin you need to set that prefix to the permanent names when filtering for those fields, etc, but this is the way it works as of version Tiki 15.2 in July 2016 .
- `value` : Optional, a value (value or calc) can be set for different fields as hardcoded string.
- `calc` :Optional (but either value or calc must be set, see above). target value, as a

result of a mathematical calculation (based on the Calculations syntax, used also in the Mathematical Calculation Tracker Field. It is very useful when you intend to execute the action through the console command where there is no human manual input. For exemple `calc="(date)"` on a tracker field will set the value for the default expected as if you were manually saving the item (see the Cloning tracker item sample for the date). Check there the list of available functions and logic operators, etc.

- `ignore_errors` (New in Tiki21 and backported to 20.3 and 18.6) allows modifications of items ignoring mandatory and other validation failures.

```
    - {step action="tracker_item_modify" field="OwnerUser" ignore_errors="y"}
```

- New in Tiki20 (and later backported to 19.2): `add`/ `remove` items This is useful for multivalue fields like User Selector, Category, Relation, Freetags, etc.
    - Initial commit: https://sourceforge.net/p/tikiwiki/code/68197
    - Example:

```
{step action="tracker_item_modify" field="TimeSheetVisibleTo" add="rodrigo"}
{step action="tracker_item_modify" field="TimeSheetVisibleTo" add="jonny"} {step
action="tracker_item_modify" field="TimeSheetVisibleTo" remove="luci"}
```

To let the end user pick a value to be removed

```
{step action="tracker_item_modify" field="TimeSheetVisibleTo" method="remove"}
```

To let the end user pick a value to be added

{step action="tracker_item_modify" field="TimeSheetVisibleTo" method="add"}

To set directly a value (it will replace anything else)

{step action="tracker_item_modify" field="TimeSheetVisibleTo" value="Bernard"}

To exchange a value (as seen above) you can also use several action

{step action="tracker_item_modify" field="TimeSheetVisibleTo" add="jonny"} {step action="tracker_item_modify" field="TimeSheetVisibleTo" remove="luci"}

tracker_item_insert

Adding in Tiki24 the action tracker_item_insert gives the possibility to add a new entry in a tracker of your choice. on the other hand if the field and value are not prescribed the insertion to insert a new empty line which can be modified by the action tracker_item_modify.

options:

- `object_type` :Extract directly from the search results
- `object_id` :Extract directly from the search results
- `to_tracker` :which takes the Identifier of the tracker you want to act on
- `field` :Optional, the permanent name of the field without any prefix.
- `value` :Optional, carries the value that will be initially.

Note: At the time of insertion it is only possible to insert one value.
if the tracer carries a field identifying an auto-increment it will be filled in automatically.

the values of other fields that you want to insert will be filled with the tracker_item_modify action.

---

example of usage with optional field

{ACTION(name="Birthday celebration")} {step action="tracker_item_insert" to_tracker="3" field="actuTitre" value_field="title_content"} {step action="tracker_item_modify" field="actuContenu" value_field="item_content"} {ACTION} {FORMAT(name="title_content")} Happy Birthday {display name="tracker_profilPrenom"} {display name="tracker_profilNom"} {FORMAT} {FORMAT(name="item_content")} The whole team wishes you a happy birthday {display name="tracker_profilPrenom"} {display name="tracker_profilNom"} ! Have a wonderfull day ! {FORMAT}

example of use without optional fields

{ACTION(name="Birthday celebration")} {step action="tracker_item_insert" to_tracker="3"} {step action="tracker_item_modify" field="actuTitre" value_field="title_content"} {step action="tracker_item_modify" field="actuContenu" value_field="item_content"} {ACTION} {FORMAT(name="title_content")} Happy Birthday {display name="tracker_profilPrenom"} {display name="tracker_profilNom"} {FORMAT} {FORMAT(name="item_content")} The whole team wishes you a happy birthday {display name="tracker_profilPrenom"} {display name="tracker_profilNom"} ! Have a wonderfull day ! {FORMAT}

user_group_modify

Added in Tiki24. This allow to assign (add) or unassigned (remove) a user.
You can specify param user="some_user" or use user_field="tracker_field_perm_name" to get the user from the tracker field (user selector) of the item you execute the action against. Note that UserSelector tracker field can hold multiple users, so the action will be executed on all of the stored users in that item.
The same permissions are enforced as the other places in the Tiki changing user groups - current user (the one logged when executing the action) should have permission to group_add_member or group_remove_member OR user should have permission to join group of the target group.

The table template (templates/search/list/table.tpl) has been updated to insert manually a value for a group.

Assign or unassign a user to/from a group that you can enter in the list execute interface.

The field tracker_field_userid point to a user selector.

Assign user to a group

{ACTION(name="Assign user to group" group="Registered")} {step action="user_group_modify" user_field="tracker_field_userid" operation="add"} {ACTION}

un-assign user to a group

{ACTION(name="Unassign user from group" group="Registered")} {step action="user_group_modify" user_field="tracker_field_userid" operation="remove"} {ACTION}

Assign or un-assign a user to/from a specific group

The field tracker_field_userid point to a user selector.

Assign user to the group Legal

{ACTION(name="Assign user to group" group="Registered")} {step action="user_group_modify" user_field="tracker_field_userid" add="Legal"} {ACTION}

Remove a user from the group Legal

{ACTION(name="Unassign user from group" group="Registered")} {step action="user_group_modify" user_field="tracker_field_userid" remove="Legal"} {ACTION}

wiki_approval

Marks the page as approved.

- `object_type` : Taken directly from the search results
- `object_id` : Taken directly from the search results
- `wiki_approval_state` : Taken directly from the search results

categorize_object

Added in Tiki26. This allow to categorize (add) or uncategorize (remove) an object from Category.
Example 1: Add the wiki pages to be selected to the category with ID 1.

```
{LISTEXECUTE()} {filter type="wiki page"} {sort mode="date_desc"} {pagination max="50"}
{OUTPUT(template="table")} {column label="Title" field="title_link" mode="raw"} {column label="Type"
field="object_type"} {column label="Categories" field="cats" mode="raw"} {OUTPUT}
{FORMAT(name="title_link")}{display name="title" format="objectlink"}{FORMAT}
{FORMAT(name="cats")}{display name="categories" format="categorylist"}{FORMAT}
{ACTION(name="Categorize (cat)")} {step action="categorize_object" add="1"} {ACTION} {LISTEXECUTE}
```

Example 2: Remove the wiki pages to be selected to the category with ID 1.

```
{LISTEXECUTE()} {filter type="wiki page"} {sort mode="date_desc"} {pagination max="50"}
{OUTPUT(template="table")} {column label="Title" field="title_link" mode="raw"} {column label="Type"
field="object_type"} {column label="Categories" field="cats" mode="raw"} {OUTPUT}
{FORMAT(name="title_link")}{display name="title" format="objectlink"}{FORMAT}
{FORMAT(name="cats")}{display name="categories" format="categorylist"}{FORMAT} {ACTION(name="Un-
Categorize (un-cat)")} {step action="categorize_object" remove="1"} {ACTION} {LISTEXECUTE}
```

Example 3: Add/delete wiki pages to be selected from the categories (multiple categories) indicated in the tree category. In this example, *PluginListexecute* combines the two actions (add/remove).

```
{LISTEXECUTE()} {filter type="wiki page"} {sort mode="date_desc"} {pagination max="50"}
{OUTPUT(template="table")} {column label="Title" field="title_link" mode="raw"} {column label="Type"
field="object_type"} {column label="Categories" field="cats" mode="raw"} {OUTPUT}
{FORMAT(name="title_link")}{display name="title" format="objectlink"}{FORMAT}
```

{FORMAT(name="cats")}{display name="categories" format="categorylist"}{FORMAT}
{ACTION(name="Categorize generally")} {step operation="add" action="categorize_object"} {ACTION}
{ACTION(name="Un-Categorize generally")} {step operation="remove" action="categorize_object"} {ACTION}
{LISTEXECUTE}

Screenshot:

Click to expand

Batch processing

You may want to use this plugin to automate jobs on your site. To do so, you can use the command line cURL tool to post a request to the page. Or since Tiki17, the action can be run through console.php avoiding to use http authentication and the credentials in clear in the command line.

Before Tiki17:

```
curl "http://example.com/YourActionPage" --form "list_action=ActionName" --form "objects~0=ALL"
```

If your action requires elevated access rights, you can enable HTTP Basic Authentication in Tiki, create a user in tiki with enough permissions to run the action from listexecute, and use the authentication options in curl, so that the command would be like:

```
curl -u myusername:mypassword "http://example.com/YourActionPage" --form "list_action=ActionName" --form
```

"objects~0=ALL"

Since Tiki15 - Backported from Tiki17:

```
php console.php list:execute "Page Name" "Action Name"
```

If you see this type of error message:

```
[root@server]# php console.php list:execute "Batch Sync Users" "SyncUsersTrackers" Command not available at thi
stage. Complete required installation steps. [root@server]#
```

You may need to update the database schema first, with the usual:

```
[root@server]# php console.php d:u
```

Automatisation using Tiki Scheduler

Since Tiki17 you can use Tiki Scheduler to automatise plugin List Execute actions.
Note if you use the url parameter you have to use a shell command and curl;

```
curl -u admin:12345 "http://example.com/tiki-index_raw.php?page=Expiration-Notification&alert=1&days=30" --forn
"list_action=Alert" --form "objects~0=ALL"
```

Since Tiki18 you can use the url parameter with the console command;

```
php console.php list:execute "listexecute profile test" Alert --request="days=30&alert=1"
```

If you use tracker fields output with Email action and it contains relative links (to tracker items, for example), you can
work around it from Tiki18, this way:

HTTP_HOST=domain.com php console.php list:execute WikiPage ActionName

Output

Since Tiki16 (backported to Tiki 15.3) you can use the same syntax to define columns, output, alternative results, etc as it's possible with PluginList to customize the columns you want to display for each record shown.
You can see and play with a working example by means of applying the profile Execute on List.

Customising the output template (advanced)

You can make a copy of `templates/search/list/table.tpl` in your theme's templates directory, e.g. `{OUTPUT(template="themes/mytheme/templates/mytable.tpl")}` and customise how the columns are handled there, but please note it needs to have the word ***table*** still in the filename for the list execute actions to work.

See also

- This plugin can also be used in Revision approval, Flagged Revisions or Files Tracker Field

Some examples

Modify the value of a tracker field

n this case we set the value of the dropdown tracker field "paymentsStage" from anything it was (empty or a differen
value) to "Completed". As per Tiki23, note that this is hardcoded string, meaning that even if the value doesn't exist in
the dropdown options it will be set as is.

{step action="tracker_item_modify" field="paymentsStage" value="Completed"}

Display a list base on expiry date, with mass action to send an email

Before Tiki17:

n this sample we used a tracker with a field "endDate" (contract expiration), a flag field "alert1Sent" to set if the ema
was sent or not yet (so it move away the item from this list if it was sent), a field "email_to" to set to who should be
send the alert email.

n the listExecute plugin we list all the contract where "endDate" is in the date range we want (30 days from now) and
where the field "alert1Sent" is not equal to "y".

The output give us a display of the results. This to test and monitor that all work ok or to manually select items on
which the action will be applied.

In the format we add the listExecute action and steps. Step 1 will set the "alert1Sent" flag to "y" (so the item will be
move away from list and the email won't be sent each time action is initiate). Step 2 will send an email with a proper
subject and body (set below) to "alertTo" destination mail.

Note: I tried to add a variable in the subject like the "contractTitle" or "objectId" but it is not working.

Also in the format we have the "email_content", the body of your email that work fine with text and variables.

{LISTEXECUTE()} {filter field="tracker_id" content="4"} {filter type="trackeritem"} {filter
range="tracker_field_endDate" from="now" gap="2592000"} {filter field="tracker_field_alert1Sent" content="NOT
y"} {sort mode="tracker_field_endDate_asc"} {OUTPUT(template="table")} {column label="Contract Title"
field="contract_title" mode="raw"} {column label="Contract Expiration" field="contract_exp"} {column label="To"

field="email_to"} {OUTPUT} {FORMAT(name="contract_title")}{display name="tracker_field_contractTitle" format="objectlink"}{FORMAT} {FORMAT(name="contract_exp")}{display name="tracker_field_endDate" format="date" default=""}{FORMAT} {FORMAT(name="email_to")}{display name="tracker_field_alertTo" default=""}{FORMAT} {ACTION(name="Alert 1")} {step action="tracker_item_modify" field="alert1Sent" value="y"} {step action="email" subject="Contract expiration alert" content_field="email_content" to_field="tracker_field_alertTo"} {ACTION} {FORMAT(name="email_content")} Hello, This is an automatic email from the Tiki Contract Management system. The contract {display name="tracker_field_contractTitle"} will expire in 30 days from now. You can check the item at: http://example.org/tiki-view_tracker_item.php?itemId={display name="object_id"} -- Signature {FORMAT} {LISTEXECUTE}

You can use this page manually (opening the page and checking item then applying action) or you can set a cronjob (like explained above).

Using a cronjob to automatise the alert email process

To use this listExecute within a cronjob you need to go to "tiki-admin.php?page=login" and to enable "HTTP Basic Authentication" (Preference name: login_http_basic). Then in a terminal you can use the following;

```
curl -u admin:adminpassword "http://example.org/YourActionPage" --form "list_action=ActionName" --form "objects~0=ALL"
```

List of all tickets, with mass action to re-open

```
{LISTEXECUTE()} {filter content="4532 4512 2919" field="object_id"} {filter type="trackeritem"} {ACTION(name="Re-open")} {step action="change_status" from="c" to="o"} {ACTION} {LISTEXECUTE}
```

Display open items with empty date + set current timestamp or change category + change status

The syntax below for filtering on empty fields (exact="") assumes you are using MYSQL as unified search index engine

```
{LISTEXECUTE()} {pagination max=25} {filter type="trackeritem"} {filter content="6" field="tracker_id"} {filter content="o" field="tracker_status"} {filter field="tracker_field_trAppMyApprovalDate" exact=""} {ACTION(name="Add my pre-approval + change tracker item status from Open to Pending" group="Registered")}
```

{step action="tracker_item_modify" field="trAppMyApprovalDate" calc="(date)"} {step action="change_status" from="o" to="p"} {ACTION} {ACTION(name="Set ApprovalStatus as Denied + change tracker item status from Open to Closed" group="Registered")} {step action="tracker_item_modify" field="trAppApprovalStatus" value="4"} {step action="change_status" from="o" to="c"} {ACTION} {LISTEXECUTE}

Display custom columns with table sorter and inline edition

Example taken from the profile Execute on List.

Display custom columns with table sorter and inline edition

{LISTEXECUTE()} {filter type="trackeritem"} {filter content="1" field="tracker_id"} {filter content="o OR p" field="tracker_status"} {filter content="NOT " field="tracker_field_trAppApprovalDate"} {sort mode="tracker_field_trAppID_asc"} {OUTPUT(template="table")} {column field="trAppID" label="ID" mode="raw"} {column field="trAppEmail" label="Email" mode="raw"} {column field="trAppFullName" label="Full Name" mode="raw"} {column field="trAppGCPreApprovalDate" label="GC Pre-approval" mode="raw"} {column field="trAppCEOPreApprovalDate" label="CEO Pre-approval" mode="raw"} {column field="trAppYouCanTradeUntil" label="You can trade until" mode="raw"} {column field="trAppApprovalDate" label="Approval date" mode="raw"} {column field="trAppApprovalStatus" label="Approval status" mode="raw"} {tablesorter server="n" sortable="y" tsfilters="type:nofilter|type:nofilter|type:text|type:text|type:date;format:yy-mm-dd|type:date;format:yy-mm-dd|type:date;format:yy-mm-dd|type:dropdown" tscolselect="critical|critical|4|6|6|2|6|5"} {OUTPUT} {FORMAT(name="trAppID")}{display name="tracker_field_trAppID" format="objectlink" default=""}{FORMAT} {FORMAT(name="trAppEmail")}{display name="tracker_field_trAppEmail" default=""}{FORMAT} {FORMAT(name="trAppFullName")}{display name="tracker_field_trAppFullName" default=""}{FORMAT} {FORMAT(name="trAppGCPreApprovalDate")}{display name="tracker_field_trAppGCPreApprovalDate" format="datetime" default=""}{FORMAT} {FORMAT(name="trAppCEOPreApprovalDate")}{display name="tracker_field_trAppCEOPreApprovalDate" format="datetime" default=""}{FORMAT} {FORMAT(name="trAppYouCanTradeUntil")}{display name="tracker_field_trAppYouCanTradeUntil" format="datetime" default=""}{FORMAT} {FORMAT(name="trAppApprovalDate")}{display name="tracker_field_trAppApprovalDate" format="datetime" default=""}{FORMAT} {FORMAT(name="trAppApprovalStatus")}{display

name="tracker_field_trAppApprovalStatus" format="trackerrender" default=""}{FORMAT}
{ALTERNATE()}{TR()}There are no requests pending your processing.{TR}{ALTERNATE} {ACTION(name="Set as
APPROVED + send email + close item status" group="Registered")} {step action="tracker_item_modify"
field="trAppApprovalStatus" value="2"} {step action="change_status" from="p" to="c"} {step action="email"
subject="Request Approved" content_field="email_content_approved" to_field="tracker_field_trAppEmail"} {step
action="email" subject="Request Approved" content_field="email_content_cs_approved" to="info@example.com"}
{ACTION} {ACTION(name="Set as DENIED + send email + close item from pending status" group="Registered")}
{step action="tracker_item_modify" field="trAppApprovalStatus" value="3"} {step action="change_status"
from="p" to="c"} {step action="email" subject="Request Denied" content_field="email_content_denied"
to_field="tracker_field_trAppEmail"} {ACTION} {ACTION(name="Set as DENIED + send email + close item from
open status" group="Registered")} {step action="tracker_item_modify" field="trAppApprovalStatus" value="3"}
{step action="change_status" from="o" to="c"} {step action="email" subject="Request Denied"
content_field="email_content_denied" to_field="tracker_field_trAppEmail"} {ACTION}
{FORMAT(name="email_content_approved")} Dear {display name="tracker_field_trAppFullName"
format="trackerrender"} Your request (internal Id #{display name="tracker_field_trAppID"}, submitted on {displa
name="creation_date" format="datetime"}) has been approved. Yours sincerely, -- ACME Ltd. Platform to Request
XXX http://example.com {FORMAT} {FORMAT(name="email_content_denied")} Dear {display
name="tracker_field_trAppFullName" format="trackerrender"} Your request submitted on {display
name="creation_date" format="datetime"} is not approved. Please contact acme@example.com for more
information. Yours sincerely, -- ACME Ltd. Platform to Request XXX http://example.com {FORMAT}
{FORMAT(name="email_content_cs_approved")} Dear Management Department, Please find below details from a
person who has been approved a request in ACME Ltd.: {FANCYTABLE(head="Name of Employee (Employee ID) |
Request Date | Approval Date | Last Exercise Date")} {display name="tracker_field_trAppFullName"
format="trackerrender"} | {display name="creation_date" format="date"} | {display
name="tracker_field_trAppApprovalDate" format="date" default=""} | {display
name="tracker_field_trAppYouCanTradeUntil" format="datetime" default=""} {FANCYTABLE} ''Please note that Las
Window Open Date is {display name="tracker_field_trAppTradingPeriodEndDate" format="raw" default=""}'' If you

have any questions, please contact acme@example.com . Yours sincerely, -- ACME Ltd. Platform to Request XXX http://example.com {FORMAT} {LISTEXECUTE}

Related pages

- Initial commit