PluginList format control block

The FORMAT control block is used to create individually formatted objects that can then be used in any of the individual OUTPUT methods.

- **Available filters**
- Basic example
- Examples of the use of the FORMAT control block (format and default value)
- Table with check marks instead of content when values found
- Using Font Awesome icons instead of content depending on the value of the field content
- Display an Item Link (tracker item) linked to another item within a template page
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Boolean</th>
<th>Sample</th>
<th>Tiki Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>Display a default value or a default file when no value is available</td>
<td>no</td>
<td>&quot;empty&quot;, &quot;1140&quot;</td>
<td>-</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Boolean</td>
<td>Sample</td>
<td>Tiki Version</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>---------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>format</td>
<td>format the displayed value to something different that the db value (human readable)</td>
<td>no</td>
<td>&quot;trackerrnder&quot;, &quot;date&quot;, &quot;objectlink&quot;</td>
<td>-</td>
</tr>
</tbody>
</table>
Note that in some cases the field that will be displayed has its own behavior and it can clash with the format you selected.

IE: A tracker field that is a "title" is pre-formatted as a link. If you set the format as "objectlink" (or you set the display inside a link) the display will end broken as Tiki will try to display a link inside a link.

For the default parameter, it's important to remember that the tracker field "Files" works a bit differently than the others with this parameter. The tracker field Files expect an Id from the File Gallery. When most of the other fields type will display the text of the default parameters the default for "Files" must be an existing file in a gallery. The file can be an "anonymous" portrait file or a plain white image or an image with only the text "Not Available", etc. Whatever you feel right for your usage.
Basic example

In most of the examples shown in PluginList output control block and PluginList advanced output control block you will see a FORMAT control block have been used. E.g. in the example shown for Table output as below, two FORMAT control blocks are used:

**Sample using the default table formatting**

```plaintext
{LIST()}
{filter type="trackeritem"}
{filter content="water"}
{OUTPUT(template="table")}
  {column field="title_link" sort="title" label="Title" mode="raw"}
  {column field="description" label="Description"}
  {column field="event_date" sort="tracker_field_18" label="Event Date"}
{OUTPUT}
{FORMAT(name="title_link")}{display name="title" format="objectlink"}{FORMAT}
{FORMAT(name="event_date")}{display name="tracker_field_18" format="date"}{FORMAT}
{LIST}
```

The first FORMAT control block defines an object with the name `title-link` that is displayed in a specific way - this object reference is then used in the column control block in the body of the table OUTPUT. Similarly the second FORMAT control block defines the `event_date` object which is also used in a column control block.

A similar use of the FORMAT control block allows the normal set of Smarty variables, eg `{user}` etc, to be made available in Smarty templates - for example:

**Format example to reference a normal smarty variable**

```plaintext
{FORMAT(name="theuser")}{user}{FORMAT}
```
The double brackets ( {{}} ) around user should have no spaces between them. Spaces are used here only to keep the user variable from being interpreted.

This creates a reference object theuser with the userId of the current user which can then be invoked in a server stored smarty template using the variable {$row.theuser}.
Examples of the use of the FORMAT control block (format and default value)

Default parameter applied on the "photo" field will display a default generic image if not file found. Default parameter applied on the "location" field will display the text "Unknown" if the field is empty.
Table with check marks instead of content when values found

In case you wanted a table with check marks (ticks) for if certain fields were filled in, rather than showing the actual data showing, you can use this code:

```
{LIST()}
{filter field="tracker_id" content="42"}
{OUTPUT(template="table")}
{column label="Title" field="title" sort="title"}
{column label="Has Data" field="hasData"}
{OUTPUT}
{FORMAT(name="hasData")}{DIV(class={display name="tracker_field_data" default="hidden"} title="Has some data"}{icon name="check-circle"}{DIV}{FORMAT}
{LIST}
```

So the 2nd column uses a FORMAT plugin where the contents of the field in question, tracker_field_data, is used as the class for a div but the default (used if the data field is empty) is set to be the bootstrap class "hidden" so the contents, the check mark, only appears if there is some data.

Obviously if the data is the name of some other class weird stuff might happen, but for most usual cases, it works as expected and saves having to make a new smarty template just for this simple list.

(Tip and example taken from a message from JonnyB in the developers list - thanks!)
Using Font Awesome icons instead of content depending on the value of the field content

Tweaking the sample above and using font awesome (integrated in Tiki) you can display corresponding icons for a content.

In this demo case we have a tracker with a "gender" field that hold 2 values, "male" or "female". There is a "male" and a "female" icon in Font Awesome and as both use the same term we can assign it to a div type icon and class.

```plaintext
{LIST()
  {filter field="tracker_id" content="42"}
  {OUTPUT(template="table")}
    {column label="Title" field="title" sort="title"}
    {column label="Gender" field="gender"}
  {OUTPUT}
{FORMAT(name="gender")}{DIV(type=1 class=fa fa-{display name="tracker_field_gender"})}{DIV}{FORMAT}
{LIST}
```

It will display a ♂ if the content of the field = male and a ♀ if the content of the field = female.
Display an Item Link (tracker item) linked to another item within a template page

When you use a template to display your plugin list you want other item related or linked to the item to open using the same template (or another it doesn't matter) but not to open the trackers built-in system. For this you need to enable the Alias feature and set it to use the template to display the tracker item, "record-" for this sample.

When you use an item link to link an item to other item (related, friend of, etc.) this is a kind of a problem. Item link has 2 option to display and item link. Value and Link. Value display the id of the content (ie: 6). Link display the value of the content linked to the tracker item (ie: Bernard).

Your link should be like:

```
[record-6|Bernard]
```

So it open the item "Id6" in the "record-" template showing the value "Bernard" for the link. To achieve this you need to create and use 2 row.

```
{FORMAT(name="name")}{display name="tracker_field_contact" format=trackerrender}{FORMAT}
{FORMAT(name="id")}{display name="tracker_field_contact"}(FORMAT)
```

The first one with format=trackerrender will display the content of the tracker Item Link field as it is shown in
The second one **without** the format=trackerrender will show the real value for this field (6).

The assemble your link as follow:

```
[record-{display name="id"}|{display name="name"}]
```

It will complete the trick and open the item 6 using the record- template while the link displayed will show the right value, Bernard.

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 aggregate control block
- PluginList overview about control blocks parameters and values
- LIST - Troubleshooting The List Plugin
- LIST - Graphical User Interface
- LIST - Hacks and Fun