# Table of Contents

**Topic Data Helper Plugin**

- Background ................................................................................................................................. 1
- When to use this plugin .............................................................................................................. 1
- How it works ................................................................................................................................ 2
  - Finding the topics to find data in ........................................................................................... 2
  - Collecting data from the topics ............................................................................................... 3
  - Removing unwanted data ....................................................................................................... 4
  - Sorting the data ...................................................................................................................... 6
  - Limiting the amount of data to show .................................................................................... 7
  - Formatting the data ................................................................................................................ 7
  - Additional functions ................................................................................................................ 8
- Syntax Rules .................................................................................................................................... 8
- Plugin Settings ............................................................................................................................. 8
- Plugin Installation Instructions .................................................................................................. 8
- Plugin Info ...................................................................................................................................... 9
**Topic Data Helper Plugin**

Helper plugin for collecting, filtering and sorting data objects.

This plugin is intended for plugin authors. This plugin is used by TWiki:Plugins.AttachmentListPlugin and TWiki:Plugins.FormFieldListPlugin to process this kind of parameters:

```text
%ATTACHMENTLIST{
    web="*"
    topic="*"
    excludetopic="WebHome, WebPreferences"
    extension="jpg,jpeg,gif,png"
    includefilepattern="(?i)^[A]"
    fromdate="2007/01/01"
    sort="$fileName"
    sortorder="descending"
}%
```

In short, this plugin provides:

- **Collecting**
  - Creation of a web-topic hash to pass one set of topics to process
  - Exclude topics that the current user does not have view permission for
  - Adding your custom data objects to this hash

- **Filtering**
  - Filter your data objects by property (direct match or regular expression)
  - Filter by date range

- **Listing** (for further processing)
  - Get a array of all data objects
  - Get a stringified array for object file storage or caching

- **Sorting**
  - Sorting by property (primary key) and secondary key
  - Sort ascending or descending

**Background**

With extending TWiki:Plugins.FormListPlugin I found I had the same needs as with TWiki:Plugins.AttachmentListPlugin. I needed almost the plugin syntax parameters! I decided to abstract out the collecting, filtering and sorting functions and provide them in a re-usable way.

**When to use this plugin**

Any time you need to process a set of data - filtering, sorting - this plugin may make your life easier.

See for example how filters in AttachmentListPlugin are created:

```perl
# filter attachments by date range
if ( defined $inParams->{'fromdate'} || defined $inParams->{'todate'} ) {
    TWiki::Plugins::TopicDataHelperPlugin::filterTopicDataByDateRange(
        $topicData, $inParams->{'fromdate'},
        $inParams->{'todate'} );
}

# filter included/excluded filenames
if ( defined $inParams->{'file'} || defined $inParams->{'excludefile'} ) {
    
```

**Topic Data Helper Plugin**
There is a relatively small burden for setting up your data for data collection - to enable it for filtering and sorting. After that it is more or less straightforward.

**How it works**

Let us assume the processing order in your plugin would be:

1. Finding the topics to find data in (filtering out unwanted topics)
2. Collecting data from the topics
3. Removing unwanted data
4. Sorting the data
5. Limiting the amount of data to show
6. Formatting the data

TopicDataHelperPlugin does not prescribe any way to write your plugin. But let's follow this order and see how the plugin could help you.

**Finding the topics to find data in**

Almost all functions assume you have a hash object with web-topic-data relations. For AttachmentListPlugin this looks like:

```perl
%topicData = ( 
  Web1 => { 
    Topic1 => { 
      picture.jpg => FileData object, 
      me.PNG => FileData object, 
      ... 
    },
  },
)
```

The first step is to create a hash of web-topic relations, and that is what `createTopicData` does:

```perl
createTopicData( $webs, $excludewebs, $topics, $excludetopics ) -> \%hash
```

And it may be called like this:

```perl
my $webs   = $inParams->{'web'}   || $inWeb   || ''; 
my $topics = $inParams->{'topic'} || $inTopic || ''; 
my $excludeTopics = $inParams->{'excludetopic'} || ''; 
my $excludeWebs   = $inParams->{'excludeweb'}   || ''; 
```
my $topicData =
TWiki::Plugins::TopicDataHelperPlugin::createTopicData(
  $webs, $excludeWebs, $topics, $excludeTopics );

The resulting hash looks like this:

%topicData = (
  Web1 => {  
    Topic1 => 1,
    Topic2 => 1,
    ...
  }
  Web2 => {  
    Topic1 => 1,
    Topic2 => 1,
    ...
  }
)

The 1 values are placeholders for now.

At this stage you will have filtered out unwanted webs and topics as passed in the parameters $webs, $excludewebs, $topics, $excludetopics.

**Collecting data from the topics**

To store the data we will retrieve from the topics need a separate data structure. I find it useful to create a data class. For AttachmentListPlugin I have used the class FileData:

```perl
package TWiki::Plugins::AttachmentListPlugin::FileData;

To filter and sort on object data properties, these properties must be accessible as instance members.

For instance, to filter FileData objects on attachment date, we create a FileData date property that we fill with the $attachment->{'date'} value:

```perl
sub new {  
  my ( $class, $web, $topic, $attachment ) = @_;  
  my $this = {};  
  $this->{'attachment'} = $attachment;  
  $this->{'date'} = $attachment->{'date'} || 0;  
  ...
  bless $this, $class;
}
```

... to be able to write:

```perl
my $fd =  
  TWiki::Plugins::AttachmentListPlugin::FileData->new( $inWeb,  
    $inTopic, $attachment );
my $date = $fd->{date};
```

To add our data objects to the web-topic hash, we call insertObjectData:

```perl
insertObjectData( $topicData, $createObjectDataFunc, $properties )
```

Where $topicData is our hash reference, and $createObjectDataFunc is a reference to a function that will create data objects. You will write that function. Parameter $properties is optional. You may pass a hash reference with custom data to your object creation function.
For AttachmentListPlugin that function looks like:

```perl
sub _createFileData {
    my ( $inTopicHash, $inWeb, $inTopic ) = @_; 

    my $attachments = _getAttachmentsInTopic( $inWeb, $inTopic );
    if ( scalar @$attachments ) {
        $inTopicHash->{$inTopic} = ();
        foreach my $attachment (@$attachments) {
            my $fd =
            TWiki::Plugins::AttachmentListPlugin::FileData->new(
                $inWeb, $inTopic, $attachment );
            my $fileName = $fd->{name};
            $inTopicHash->{$inTopic}{$fileName} = \$fd;
        }
    } else {
        # no META:FILEATTACHMENT, so remove from hash
        delete $inTopicHash->{$inTopic};
    }
}
```

And it is called with:

```perl
TWiki::Plugins::TopicDataHelperPlugin::insertObjectData(
    $topicData, \&_createObjectData
);
```

Now your hash will have the structure:

```perl
%topicData = (
    Web1 => {
        Topic1 => {
            'key a' => object,
            'key b' => object,
            ...
        },
    },
)
```

### Removing unwanted data

TopicDataHelperPlugin provides 4 filter functions:

- `filterTopicDataByViewPermission( $topicData, $wikiUserName )`

Filters topic data objects by checking if the user `$wikiUserName` has view access permissions.

Removes topic data if the user does not have permission to view the topic.

Example:

```perl
# filter topics by view permission
my $user = TWiki::Func::getWikiName();
my $wikiUserName = TWiki::Func::userToWikiName( $user, 1 );
TWiki::Plugins::TopicDataHelperPlugin::filterTopicDataByViewPermission(
    \$topicData, $wikiUserName);
```

- `filterTopicDataByDateRange( $topicData, $fromDate, $toDate, $dateKey )`

Filters topic data objects by date range, from `$fromDate` to `$toDate` (both in epoch seconds).
Removes topic data if:

- the value of the object attribute $dateKey is earlier than $fromDate
- the value of the object attribute $dateKey is later than $toDate

Use either $fromDate or $toDate, or both.

Example:

```perl
# filter attachments by date range
if ( defined $inParams->{'fromdate'} || defined $inParams->{'todate'} ) {
    TWiki::Plugins::TopicDataHelperPlugin::filterTopicDataByDateRange(
        $topicData, $inParams->{'fromdate'}, $inParams->{'todate'});
}

filterTopicDataByProperty( $topicData, $propertyKey, $isCaseSensitive, $includeValues, $excludeValues )
```

Filters topic data objects by matching an object property with a list of possible values.

Removes topic data if:

- the object attribute $propertyKey is not in $includeValues
- the object attribute $propertyKey is in $excludeValues

Use either $includeValues or $excludeValues, or both.

For example, AttachmentListPlugin uses this function to filter attachments by extension. extension="gif, jpg" will find all attachments with extension 'gif' OR 'jpg'. OR 'GIF' or 'JPG', therefore $isCaseSensitive is set to 0.

Example:

```perl
# filter included/excluded field VALUES
my $values = $inParams->{'includevalue'} || undef;
my $excludeValues = $inParams->{'excludevalue'} || undef;
if ( defined $values || defined $excludeValues ) {
    TWiki::Plugins::TopicDataHelperPlugin::filterTopicDataByProperty(
        $topicData, 'value', 1, $values, $excludeValues);
}

filterTopicDataByRegexMatch( $topicData, $propertyKey, $includeRegex, $excludeRegex )
```

Filters topic data objects by matching an object property with a regular expression.

Removes topic data if: - the object attribute $propertyKey does not match $includeRegex - the object attribute $propertyKey matches $excludeRegex

Use either $includeRegex or $excludeRegex, or both.

Example:

```perl
# filter filenames by regular expression
if (   defined $inParams->{'includefilepattern'}
    || defined $inParams->{'excludefilepattern'}
) {
    TWiki::Plugins::TopicDataHelperPlugin::filterTopicDataByRegexMatch(
        $topicData, $excludeRegex);
```
After using the filter functions, your topic data hash will probably be quite some smaller. Next step is sorting the data. Limiting the result set will come after sorting.

**Sorting the data**

Before sorting, we must make an array from our hash. This is what `getListOfObjectData` does:

```perl
getListOfObjectData( $topicData ) -> \@objects
```

Example:

```perl
my $objects =
  TWiki::Plugins::TopicDataHelperPlugin::getListOfObjectData($topicData);
```

Now we can sort the list of data objects with `sortObjectData`:

```perl
sortObjectData( $objectData, $sortOrder, $sortKey, $compareMode, $nameKey )
-> \@objects
```

Function parameters:

- `\@objectData` (array reference) - list of data objects (NOT the topic data!)
- `$sortOrder` (int) - either $TWiki::Plugins::TopicDataHelperPlugin::sortDirections{'ASCENDING'}, $TWiki::Plugins::TopicDataHelperPlugin::sortDirections{'DESCENDING'} or $TWiki::Plugins::TopicDataHelperPlugin::sortDirections{'NONE'}
- `$inSortKey` (string) - primary sort key; this will be a property of your data object
- `$compareMode` (string) - sort mode of primary key, either ‘numeric’ or ‘alphabetical’
- `$nameKey` (string) - to be used as secondary sort key; must be alphabetical; this will be a property of your data object

This function returns a reference to an sorted array of data objects.

To get the primary sort key and the kind of data (alphabetical or integer) we can create a lookup table in our data class:

```perl
my %sortKeys = (
  '$fileDate'      => [ 'date',      'integer' ],
  '$fileSize'      => [ 'size',      'integer' ],
  '$fileUser'      => [ 'user',      'string' ],
  '$fileExtension' => [ 'extension', 'string' ],
  '$fileName'      => [ 'name',      'string' ],
  '$fileTopic'     => [ 'topic',     'string' ]
);
```

```perl
sub getSortKey {
  my ($inRawKey) = @_;
  return $sortKeys{$inRawKey}[0];
}
```

```perl
sub getCompareMode {
  my ($inRawKey) = @_;
  return $sortKeys{$inRawKey}[1];
}
```

Sorting the data
This can be used as follows:

```perl
my $sortKey = &TWiki::Plugins::AttachmentListPlugin::FileData::getSortKey($inParams->{'sort'});
my $compareMode = &TWiki::Plugins::AttachmentListPlugin::FileData::getCompareMode($inParams->{'sort'});
```

Similarly we can create a mapping between user input (for instance the `sortorder` parameter) and the `$sortOrder` value we pass to TopicDataHelperPlugin:

```perl
my %sortInputTable = ('none' => $TWiki::Plugins::TopicDataHelperPlugin::sortDirections{'NONE'},
'sorting' => $TWiki::Plugins::TopicDataHelperPlugin::sortDirections{'ASCENDING'},
'descending' => $TWiki::Plugins::TopicDataHelperPlugin::sortDirections{'DESCENDING'},
);
```

```perl
# translate input to sort parameters
my $sortOrderParam = $inParams->{'sortorder'} || 'none';
my $sortOrder = $sortInputTable{$sortOrderParam} || $TWiki::Plugins::TopicDataHelperPlugin::sortDirections{'NONE'};
```

Now we sort the data:

```perl
$objects = TWiki::Plugins::TopicDataHelperPlugin::sortObjectData($objects, $sortOrder, $sortKey, $compareMode, 'name');
```

## Limiting the amount of data to show

With the final data in the right order, we can simply shorten the array:

```perl
splice @$objects, $inParams->{'limit'} if defined $inParams->{'limit'};
```

## Formatting the data

Formatting is not provided by TopicDataHelperPlugin, but your formatting function would logically have this setup:

```perl
sub _formatData {
    my ( $inObjects, $inParams ) = @_;
    my @objects = @$inObjects;
    my $format = $inParams->{'format'} || $defaultFormat;
    my $separator = $inParams->{'separator'} || "\n";
    my @formattedData = ();
    foreach my $object (@objects) {
        my $s = $format;
        ... perform string substitutions
        push @formattedData, $s;
    }
    my $outText = join $separator, @formattedData;
    return $outText;
}
```

Limiting the amount of data to show
Additional functions

A useful utility function when you need to match values to a comma-separated string, is `makeHashFromString`.

```
makeHashFromString( $text, $isCaseSensitive ) -> %hash
```

For example:

```perl
my $excludeTopicsList = 'WebHome, WebPreferences';
my $excludeTopics = makeHashFromString( $excludeTopicsList, 1 );
...
```

This will create:

```
$hashref = {
    'WebHome' => 1,
    'WebPreferences' => 2,
};
```

To store object data in a file, you may use `stringifyTopicData`:

```
stringifyTopicData( $topicData ) -> @objects
```

This function creates an array of strings from topic data objects, where each string is generated by the object's method `stringify` (to be implemented by your object's data class).

For example, `FormFieldData`'s `stringify` method looks like this:

```perl
sub stringify {
    my $this = shift;
    return
    "1.0\t$this->{web}\t$this->{topic}\t$this->{name}\t$this->{value}\t$this->{date}";
}
```

Call this method with:

```perl
my $list = TWiki::Plugins::TopicDataHelperPlugin::stringifyTopicData($topicData);
my $text = join "\n", @$list;
```

Syntax Rules

None. See the plugin documentation in the .pm file on detailed usage instructions.

Plugin Settings

- Set SHORTDESCRIPTION = Helper plugin for collecting, filtering and sorting data objects.
- Set DEBUG = 0

Plugin Installation Instructions

- Download the ZIP file from the Plugin web (see below)
- Unzip TopicDataHelperPlugin.zip in your root ($TWIKI_ROOT) directory. Content:

<table>
<thead>
<tr>
<th>File:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>data/TWiki/TopicDataHelperPlugin.txt</td>
<td></td>
</tr>
</tbody>
</table>
Optionally, if it exists, run **TopicDataHelperPlugin_installer** to automatically check and install other TWiki modules that this module depends on. You can also do this step manually. Alternatively, manually make sure the dependencies listed in the table below are resolved. None

Visit **configure** in your TWiki installation, and enable the plugin in the **Plugins** section.

### Plugin Info

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Copyright:</td>
<td>© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors. Ideas, requests, problems regarding TWiki? Send feedback</td>
</tr>
<tr>
<td>License:</td>
<td>GPL</td>
</tr>
<tr>
<td>Plugin Version:</td>
<td>19246 (2010-07-25)</td>
</tr>
<tr>
<td>TWiki Dependency:</td>
<td>None</td>
</tr>
<tr>
<td>CPAN Dependencies</td>
<td>none</td>
</tr>
<tr>
<td>Other Dependencies:</td>
<td>none</td>
</tr>
<tr>
<td>Perl Version:</td>
<td>5.005</td>
</tr>
<tr>
<td>Benchmarks:</td>
<td>GoodStyle nn%, FormattedSearch nn%, TopicDataHelperPlugin nn%</td>
</tr>
<tr>
<td>Feedback:</td>
<td><a href="http://TWiki.org/cgi-bin/view/Plugins/TopicDataHelperPluginDev">http://TWiki.org/cgi-bin/view/Plugins/TopicDataHelperPluginDev</a></td>
</tr>
<tr>
<td>Appraisal:</td>
<td><a href="http://TWiki.org/cgi-bin/view/Plugins/TopicDataHelperPluginAppraisal">http://TWiki.org/cgi-bin/view/Plugins/TopicDataHelperPluginAppraisal</a></td>
</tr>
</tbody>
</table>

### Related topics:

- TWikiPlugins
- TWiki:Plugins.AttachmentListPlugin
- TWiki:Plugins.FormFieldListPlugin

This topic: TWiki > TopicDataHelperPlugin

Topic revision: r0 - 2010-07-26 - TWikiContributor

Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors. Ideas, requests, problems regarding TWiki? Send feedback

**Note:** Please contribute updates to this topic on TWiki.org at TWiki:TWiki.TopicDataHelperPlugin