

- [Creating a trail](#)
- [Types of trail](#)
- [Trail link syntax](#)
- [Using a trail](#)
 - [Path trail](#)
 - [Circular trail](#)
- [Cross group trails](#)
- [Trail style](#)
- [Trail page lists](#)

The [Wiki Trails](#) feature allows wiki authors to create "trails" through sequences of pages in the wiki. You simply specify pages and their order on a "trail index", and then place the navigation markup on the pages that you will be navigating.

(Don't confuse the [pagelist](#) directive with [Wiki Trails](#) - they are different animals as explained in the [Q and A](#) below.)

Creating a trail

Before you can use a trail through a group of pages, you have to create a "trail index" on a separate page, which we will call the "trail index page".

On that trail index page, you simply create a numbered or bulleted [list](#) of links. (So every numbered or bulleted list of links implicitly creates a trail.)

It is important that each page name ([link](#)) be the first item following each bullet; any text or formatting in front of the page name will exclude it from the trail.

If you want to format your trail (list), you can [include a CSS](#).

An example trail index page might contain the list:

- [Installation](#) how to install
- [The customisation page](#)
- [PmWiki](#) some other text [PmWiki Philosophy](#) (The latter won't be in the trail because it is preceded by text)
- Yet some other text. [PmWiki.WikiStyles](#) (This won't be in the trail because it follows text)
- [Uploads](#) (This won't be in the trail because it is preceded by the %center% style.)
- Some text (This won't be in the trail because it is not a link)
- [PageLists](#) Listing pages by multiple criteria with templated output
- <http://pmwiki.org> (This won't be in the trail because it is not a page link)
 - [PmWiki:InterMap](#) (This won't be in the trail because it is an [InterMap](#) link)
- [Cookbook:Cookbook](#) (This won't be in the trail because it is an [InterMap](#) link)

[PmWiki philosophy](#) [Design notes](#) (The first link will, and the second link won't, be in the trail defined by ([definition list](#)))

- [Security](#) (This won't be in the trail because its preceded by a (hidden) [anchor](#))
- [Links](#) (This won't be in the trail because its preceded by a (hidden) `%newwin%` style)
- [Troubleshooting](#) (This won't be in the trail because its preceded by (hidden) *italic* style markup)

The list above creates the following "wikitrail", displayed using a [pagelist](#):

```
(:pagelist trail={$FullName}#trailstart#trailend fmt={$FullName}#traillist:)
> Pm Wiki.Installation <
> Pm Wiki.Local Customizations <
> Pm Wiki.Pm Wiki <
> Pm Wiki.Page Lists <
> Pm Wiki.Pm Wiki Philosophy <
```

Observations

1. In general, indentation levels in the page list don't matter -- trails are a linear sequence of pages.
2. A page is part of the trail only if the page link immediately follows the list markup.
3. The list itself can be [delineated](#) by the use of [anchors](#), allowing for multiple lists on a page, or for some list items to be excluded.

Trail types

[Pm Wiki](#) defines 2 trail markups:

- `<<|[[Trail Index Page]]|>>` displays as "<< PreviousPage | Trail Index Page | NextPage >>".
- `<|[[Trail Index Page]]|>` displays as "< PreviousPage | Trail Index Page | NextPage >", except the appropriate arrow is omitted at the beginning and end of the trail.

Trail link syntax

The trail link has the same syntax as a standard [link](#), this means for example you can specify

- `<|[[TrailIndexPage | +]]|>`

Trail links can be restricted by [anchors](#) (links to a specific location within a page),

this means you can have more than one trail on a page, or start a trail from a specific location in a page.

- `<|[[Trail Index Page#trailstart#trailend]]|>`

Using the trail

What makes a trail "work" is adding *trail markup* on the pages in the trail (i.e. the pages that are listed in the bullet/numbered list on the trail index page).

To build a trail, add *trail markup* like `<<|[[TrailIndexPage]]|>>` to a page, where `TrailIndexPage` is the page, described above, containing the bulleted list of pages in the trail. Pm Wiki will display the trail markup with links to any previous and next pages in the trail.

The trail markup can be placed anywhere in a page, and a page can contain multiple trail markups. If you are adding a trail to every page in a group, consider setting the trail markup in the [GroupHeader](#) or [Group Footer](#) pages instead of on every individual page in your group.

`^|[[TrailIndexPage]]|^` treats the list levels as a hierarchy and displays the "path" to reach the current page (i.e., a "breadcrumb" trail). In the example trail above, the markup `^|TrailIndexPage|^` on `TrailPage4` would display as "TrailIndexPage | TrailPage2 | TrailPage4".

Wiki administrators can change the trail separator of the "path" trail (`^|[[TrailIndexPage]]|^`) from the default `|` by setting the variable `$TrailPathSep` in the "config.php" file. For instance `$Trail Path Sep? = ' > '`; will output "TrailIndexPage > TrailPage2 > TrailPage4".

Circular trails

Typically, a trail is a linear list with a first and a last page. However, the trail can be made "circular" by repeating the first page as the last item in the trail index:

```
* [[TrailPage1]] * [[TrailPage2]] ... * [[TrailPageN]] * [[TrailPage1]]
```

If the trail index page is intended to be read by others, the last item can be made invisible inside an `(:if false:)` block:

```
* [[TrailPage1]] * [[TrailPage2]] ... * [[TrailPageN]] (:if false:) *  
[[TrailPage1]] (:ifend:)
```

Cross Group Trails

Before version 2.2.1, if your trail contains pages in different groups, it should use full `[[Group.Name]]` links instead of just `[[Name]]`.

Other notes

- There is no space between `and` and `[[link]]` and `;` same for the other trail markups.
- Note that non-existing pages will appear in the [Wiki Trail?](#) as links.

Trail style

Pm Wiki encapsulates the trail with a *wikitrail* css class.
This allows the wiki trail to be [customised](#) by defining CSS for the *wikitrail* in the *local.css* file.

Trail in [page lists](#)

Trails from a single page can only be displayed using the pagelist [trail](#) parameter. For example

```
(:pagelist trail=PmWiki/WikiTrails          > Pm Wiki.Pm Wiki <
fmt=PmWiki.WikiTrails#traillist order=random > Pm Wiki.My Trail Page 3? <
count=3:)                                   > Pm Wiki.Installation <
```

A simple example of a [Wiki Trail?](#)

1) On the [Trail Index Page?](#):

```
* [[MyTrailPage1]] * [[MyTrailPage2]] * [[MyTrailPage3]]
```

2) On the pages [My Trail Page 1?](#), 2, and 3:

```
<<|[[TrailIndexPage]]|>>
```

```
> {=\$FullName}{=\$Groupspaced}.{=\$Namespaced}?<
```

Questions

What's the difference between a [PageList](#) and a [Wiki Trail](#)??

The pagelist directive dynamically generates a list of pages. There are many ways to generate the list, including using a [Wiki Trail](#) as the source. The pagelist directive then displays the pages that match the criteria using an optional template - for example displaying each page name on a separate line as a link or including the entire content. The pagelist directive currently does not have built-in navigation markup that you can put on the pages in the list. By contrast, [Wiki Trails](#) are simply specified via links on an "index" page and you can put previous-next navigation markup on each page. The two serve very different purposes. [Wiki Trails](#) are useful for specifying the pages in [web feeds](#), for creating a "tour" through a predefined set of pages, and many other things.