

Wikiprint Book

Title: Trac Links

Subject: Ecopath Developer Site - TracLinks

Version: 2

Date: 2024-05-05 10:55:33

Table of Contents

Trac Links	3
Where to use TracLinks	3
Advanced use of TracLinks	4
Relative links	4
InterWiki links	4
InterTrac links	5
Server-relative links	5
Quoting space in TracLinks	5
Escaping Links	5
Parameterized Trac links	5
TracLinks Reference	6
attachment: links	6
comment: links	6
query: links	6
search: links	6
ticket: links	6
timeline: links	6
wiki: links	7
Version Control related links	7
source: links	7
export: links	7
log: links	7

Trac Links

[TracLinks](#) are a fundamental feature of Trac, because they allow easy hyperlinking between the various entities in the system?such as tickets, reports, changesets, Wiki pages, milestones, and source files?from anywhere [WikiFormatting](#) is used.

[TracLinks](#) are generally of the form **type:id** (where *id* represents the number, name or path of the item) though some frequently used kinds of items also have short-hand notations.

Where to use [TracLinks](#)

You can use [TracLinks](#) in:

- Source code (Subversion) commit messages
- Wiki pages
- Full descriptions for tickets, reports and milestones

and any other text fields explicitly marked as supporting [WikiFormatting](#).

Some examples:

- Tickets: **#1** or **ticket:1**
- Ticket comments: **comment:1:ticket:2**
- Reports: **{1}** or **report:1**
- Changesets: **r1**, **[1]**, **changeset:1** or (restricted) **[1/trunk]**, **changeset:1/trunk**
- Revision log: **r1:3**, **[1:3]** or **log:@1:3**, **log:trunk@1:3**, **[2:5/trunk]**
- Diffs (requires [?0.10](#)): **diff:@1:3**, **diff:tags/trac-0.9.2/wiki-default/tags/trac-0.9.3/wiki-default** or **diff:trunk/trac@3538/sandbox/vc-refactoring@3539**
- Wiki pages: **CamelCase** or **wiki:CamelCase**
- Parent page: **[..]**
- Milestones: **milestone:1.0**
- Attachment: **attachment:example.tgz** (for current page attachment), **attachment:attachment.1073.diff:ticket:944**

(absolute path)

- Files: **source:trunk/COPYING**
- A specific file revision: **source:/trunk/COPYING@200**
- A particular line of a specific file revision: **source:/trunk/COPYING@200#L25**

Display:

- Tickets: [#1](#) or [ticket:1](#)
- Ticket comments: [comment:1:ticket:2](#)
- Reports: [{1}](#) or [report:1](#)
- Changesets: [r1](#), [\[1\]](#), [changeset:1](#) or (restricted) [\[1/trunk\]](#), [changeset:1/trunk](#)
- Revision log: [r1:3](#), [\[1:3\]](#) or [log:@1:3](#), [log:trunk@1:3](#), [\[2:5/trunk\]](#)
- Diffs (requires [0.10](#)): [diff:@1:3](#), [diff:tags/trac-0.9.2/wiki-default/tags/trac-0.9.3/wiki-default](#) or [diff:trunk/trac@3538/sandbox/vc-refactoring@3539](#)
- Wiki pages: [CamelCase](#) or [wiki:CamelCase](#)
- Parent page: [..](#)
- Milestones: [milestone:1.0](#)
- Attachment: [attachment:example.tgz](#) (for current page attachment), [attachment:attachment.1073.diff:ticket:944](#)

(absolute path)

- Files: [source:trunk/COPYING](#)
- A specific file revision: [source:/trunk/COPYING@200](#)
- A particular line of a specific file revision: [source:/trunk/COPYING@200#L25](#)

Note: The [wiki:CamelCase](#) form is rarely used, but it can be convenient to refer to pages whose names do not follow [WikiPageNames](#) rules, i.e., single words, non-alphabetic characters, etc. See [WikiPageNames](#) for more about features specific to links to Wiki page names.

Trac links using the full (non-shorthand) notation can also be given a custom link title like this:

```
[ticket:1 This is a link to ticket number one].
```

Display: [This is a link to ticket number one](#).

If the title is omitted, only the id (the part after the colon) is displayed:

```
[ticket:1]
```

Display: [1](#)

wiki is the default if the namespace part of a full link is omitted (*since version 0.10*):

```
[SandBox the sandbox]
```

Display: [the sandbox](#)

[TracLinks](#) are a very simple idea, but actually allow quite a complex network of information. In practice, it's very intuitive and simple to use, and we've found the "link trail" extremely helpful to better understand what's happening in a project or why a particular change was made.

Advanced use of [TracLinks](#)

Relative links

To create a link to a specific anchor in a page, use '#':

```
[#Relativelinks relative links]
```

Displays:

[relative links](#)

Hint: when you move your mouse over the title of a section, a "¶" character will be displayed. This is a link to that specific section and you can use this to copy the #... part inside a relative link to an anchor.

To create a link to a [?SubWiki](#)-page just use a '/':

```
WikiPage/SubWikiPage or ./SubWikiPage
```

To link from a [?SubWiki](#) page to a parent, simply use a '..':

```
[..]
```

To link from a [?SubWiki](#) page to a sibling page, use a '../':

```
[../Sibling see next sibling]
```

(*Changed in 0.11*) Note that in Trac 0.10, using e.g. `[../newticket]` may have worked for linking to the `/newticket` top-level URL, but now in 0.11 it will stay in the wiki namespace and link to a sibling page. See [#Server-relativelinks](#) for the new syntax.

[InterWiki](#) links

Other prefixes can be defined freely and made to point to resources in other Web applications. The definition of those prefixes as well as the URLs of the corresponding Web applications is defined in a special Wiki page, the [InterMapTxt](#) page. Note that while this could be used to create links to other Trac environments, there's a more specialized way to register other Trac environments which offers greater flexibility.

InterTrac links

This can be seen as a kind of [InterWiki](#) link specialized for targeting other Trac projects.

Any type of Trac links could be written in one Trac environment and actually refer to resources present in another Trac environment, provided the Trac link is prefixed by the name of that other Trac environment followed by a colon. That other Trac environment must be registered, under its name or an alias. See [InterTrac](#) for details.

A distinctive advantage of [InterTrac](#) links over [InterWiki](#) links is that the shorthand form of Trac links usually have a way to understand the [InterTrac](#) prefixes. For example, links to Trac tickets can be written #T234 (if T was set as an alias for Trac), links to Trac changesets can be written [?\[trac 1508\]](#).

Server-relative links

It is often useful to be able to link to objects in your project that have no built-in Trac linking mechanism, such as static resources, `newticket`, a shared `/register` page on the server, etc.

To link to resources inside the project, use either an absolute path from the project root, or a relative link from the URL of the current page (*Changed in 0.11*):

```
[/newticket Create a new ticket]
[/ home]
```

Display: [newticket home](#)

To link to another location on the server (outside the project), use the *'location' link syntax* (*Changed in 0.11*):

```
[//register Register Here]
```

Display: [Register Here](#)

Quoting space in TracLinks

Immediately after a [TracLinks](#) prefix, targets containing space characters should be enclosed in a pair of quotes or double quotes. Examples:

- `wiki:"The whitespace convention"`
- `attachment:'the file.txt'` or
- `attachment:"the file.txt"`
- `attachment:"the file.txt:ticket:123"`

Escaping Links

To prevent parsing of a TracLink, you can escape it by preceding it with a `!` (exclamation mark).

```
!NoLinkHere.
![42] is not a link either.
```

Display:

NoLinkHere. [42] is not a link either.

Parameterized Trac links

The Trac links target Trac resources which have generally more than one way to be rendered, according to some extra parameters. For example, a Wiki page can accept a `version` or a `format` parameter, a report can make use of dynamic variables, etc.

Any Trac links can support an arbitrary set of parameters, written in the same way as they would be for the corresponding URL. Some examples:

- `wiki:WikiStart?format=txt`
- `ticket:1?version=1`
- `[/newticket?component=module1 create a ticket for module1]`

[TracLinks](#) Reference

The following sections describe the individual link types in detail, as well as several notes advanced usage of links.

attachment: links

The link syntax for attachments is as follows:

- `attachment:the_file.txt` creates a link to the attachment `the_file.txt` of the current object
- `attachment:the_file.txt:wiki:MyPage` creates a link to the attachment `the_file.txt` of the `MyPage` wiki page
- `attachment:the_file.txt:ticket:753` creates a link to the attachment `the_file.txt` of the ticket 753

Note that the older way, putting the filename at the end, is still supported: `attachment:ticket:753:the_file.txt`.

If you'd like to create a direct link to the content of the attached file instead of a link to the attachment page, simply use `raw-attachment:` instead of `attachment:`.

This can be useful for pointing directly to an HTML document, for example. Note that for this use case, you'd have to allow the web browser to render the content by setting `[attachment] render_unsafe_content = yes` (see [TracIni#attachment-section](#)). Caveat: only do that in environments for which you're 100% confident you can trust the people who are able to attach files, as otherwise this would open up your site to [?cross-site scripting](#) attacks.

See also [#export:links](#).

comment: links

When you're inside a given tickets, you can simply write e.g. `comment:3` to link to the third change comment. It's also possible to link to a comment of a specific ticket from anywhere using one of the following syntax:

- `comment:3:ticket:123`
- `ticket:123#comment:3` (note that you can't write `#123#comment:3`!)

query: links

See [TracQuery#UsingTracLinks](#) and [#ticket:links](#).

search: links

See [TracSearch#SearchLinks](#)

ticket: links

Besides the obvious `ticket:id` form, it is also possible to specify a list of tickets or even a range of tickets instead of the `id`. This generates a link to a custom query view containing this fixed set of tickets.

Example:

- `ticket:5000-6000`
- `ticket:1,150`

(since *Trac 0.11*)

timeline: links

Links to the timeline can be created by specifying a date in the [?ISO:8601](#) format. The date can be optionally followed by a time specification. The time is interpreted as being UTC time, but alternatively you can specify your local time, followed by your timezone if you don't want to compute the UTC time.

Examples:

- `timeline:2008-01-29`
- `timeline:2008-01-29T15:48`
- `timeline:2008-01-29T16:48Z+01`

(since Trac 0.11)

wiki: links

See [WikiPageNames](#) and [quoting space in TracLinks](#) above.

Version Control related links

source: links

The default behavior for a [source:/some/path](#) link is to open the directory browser if the path points to a directory and otherwise open the log view.

It's also possible to link directly to a specific revision of a file like this:

- `source:/some/file@123` - link to the file's revision 123
- `source:/some/file@head` - link explicitly to the latest revision of the file

If the revision is specified, one can even link to a specific line number:

- `source:/some/file@123#L10`
- `source:/tag/0.10@head#L10`

Finally, one can also highlight an arbitrary set of lines:

- `source:/some/file@123:10-20,100,103#L99` - highlight lines 10 to 20, and lines 100 and 103.

(since 0.11)

export: links

To force the download of a file in the repository, as opposed to displaying it in the browser, use the `export` link. Several forms are available:

- `export:/some/file` - get the HEAD revision of the specified file
- `export:123:/some/file` - get revision 123 of the specified file
- `export:/some/file@123` - get revision 123 of the specified file

This can be very useful for displaying XML or HTML documentation with correct stylesheets and images, in case that has been checked in into the repository. Note that for this use case, you'd have to allow the web browser to render the content by setting `[browser] render_unsafe_content = yes` (see [TracIni#browser-section](#)), otherwise Trac will force the files to be downloaded as attachments for security concerns.

If the path is to a directory in the repository instead of a specific file, the source browser will be used to display the directory (identical to the result of `source:/some/dir`).

log: links

The `log:` links are used to display revision ranges. In its simplest form, it can link to the latest revisions from the specified path, but it can also support displaying an arbitrary set of revisions.

- `log:/` - the latest revisions starting at the root of the repository
- `log:/trunk/tools` - the latest revisions in `trunk/tools`
- `log:/trunk/tools@10000` - the revisions in `trunk/tools` starting from revision 10000
- `log:@20788,20791:20795` - list revision 20788 and the 20791 to 20795 revision range
- `log:/trunk/tools@20788,20791:20795` - list revision 20788 and the revisions from the 20791 to 20795 range which affect the given path

There are short forms for revision ranges as well:

- `[20788,20791:20795]`
- `[20788,20791:20795/trunk/tools]`
- `r20791:20795` (but not `r20788,20791:20795` nor `r20791:20795/trunk`)

Finally, note that in all of the above, a revision range can be written indifferently `x:y` or `x-y`.

See also: [WikiFormatting](#), [TracWiki](#), [WikiPageNames](#), [InterTrac](#), [InterWiki](#)