

# Custom Ticket Fields

Trac supports adding custom, user-defined fields to the ticket module. With custom fields you can add typed, site-specific properties to tickets.

## Configuration

Configuring custom ticket fields is done in the `trac.ini` file. All field definitions should be under a section named `[ticket-custom]`.

The syntax of each field definition is:

```
FIELD_NAME = TYPE
(FIELD_NAME.OPTION = VALUE)
...
```

The example below should help to explain the syntax.

## Available Field Types and Options

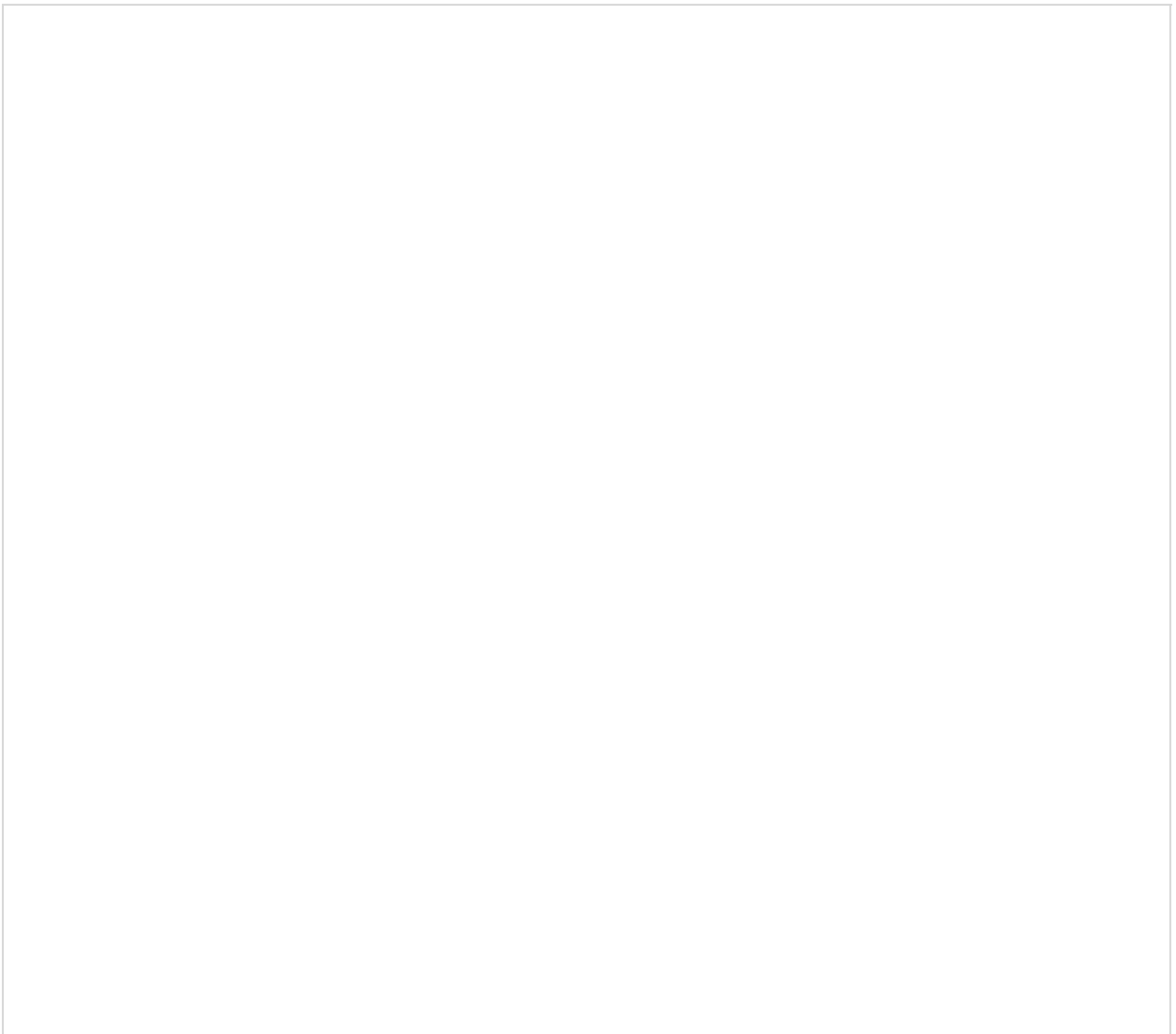
- **text**: A simple (one line) text field.
  - ◆ label: Descriptive label.
  - ◆ value: Default value.
  - ◆ order: Sort order placement; this determines relative placement in forms with respect to other custom fields.
  - ◆ format: One of:
    - ◇ `plain` for plain text
    - ◇ `wiki` to interpret the content as [WikiFormatting](#)
    - ◇ `reference` to treat the content as a queryable value (*since 1.0*)
    - ◇ `list` to interpret the content as a list of queryable values, separated by whitespace (*since 1.0*)
- **checkbox**: A boolean value check box.
  - ◆ label: Descriptive label.
  - ◆ value: Default value, 0 or 1.
  - ◆ order: Sort order placement.
- **select**: Drop-down select box. Uses a list of values.
  - ◆ label: Descriptive label.
  - ◆ options: List of values, separated by | (vertical pipe).
  - ◆ value: Default value (one of the values from options).
  - ◆ order: Sort order placement.
- **radio**: Radio buttons. Essentially the same as **select**.
  - ◆ label: Descriptive label.
  - ◆ options: List of values, separated by | (vertical pipe).
  - ◆ value: Default value, one of the values from options.
  - ◆ order: Sort order placement.
- **textarea**: Multi-line text area.
  - ◆ label: Descriptive label.
  - ◆ value: Default text.
  - ◆ cols: Width in columns. (*Removed in 1.1.2*)

- ◆ rows: Height in lines.
- ◆ order: Sort order placement.
- ◆ format: Either `plain` for plain text or `wiki` to interpret the content as [WikiFormatting](#).
- **time**: Date and time picker. (*Since 1.1.1.*)
  - ◆ label: Descriptive label.
  - ◆ value: Default date.
  - ◆ order: Sort order placement.
  - ◆ format: One of:
    - ◇ `relative` for relative dates.
    - ◇ `date` for absolute dates.
    - ◇ `datetime` for absolute date and time values.

If the `label` is not specified, it will be created by capitalizing the custom field name and replacing underscores with whitespaces.

Macros will be expanded when rendering `textarea` fields with format `wiki`, but not when rendering `text` fields with format `wiki`.

## Sample Configuration





**Note:** To make a `select` type field optional, specify a leading `|` in the `fieldname.options` option.

## Reports Involving Custom Fields

Custom ticket fields are stored in the `ticket_custom` table, not in the `ticket` table. So to display the values from custom fields in a report, you will need a join on the 2 tables. Let's use an example with a custom ticket field called `progress`.

```
pvalue __color__
  id ticket summary value progress
  ticket t enum p ticket_custom
  status tid ticket name
pname tpriority p
  pvalue
```

**Note:** This will only show tickets that have `progress` set in them. This is **not the same as showing all tickets**. If you created this custom ticket field *after* you have already created some tickets, they will not have that field defined, and thus they will never show up on this ticket query. If you go back and modify those tickets, the field will be defined, and they will appear in the query.

However, if you want to show all ticket entries (with `progress` defined and without), you need to use a `JOIN` for every custom field that is in the query:

```
pvalue __color__
  id ticket summary component milestone severity
  status
  time created
  changetime _changetime description _description
  reporter _reporter
  value value progress
  ticket t
  ticket_custom tid ticket name
  enum p pname tpriority p
  status
  pvalue milestone severity time
```

Note in particular the `LEFT OUTER JOIN` statement here.

Note that if your config file uses an **uppercase** name:



you would use **lowercase** in the SQL: `AND c.name = 'progress_type'`.

---

See also: [TracTickets](#), [TracIni](#)