Custom Ticket Fields

Trac supports adding custom, user-defined fields to the ticket module. Using 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. (Determines relative placement in forms.)
- **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 (Item #, starting at 0).
    - 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 (Item #, starting at 0).
    - order: Sort order placement.
- **textarea**: Multi-line text area.
    - label: Descriptive label.
    - value: Default text.
    - cols: Width in columns.
    - rows: Height in lines.
    - order: Sort order placement.

Sample Config

```
[ticket-custom]

test_one = text
test_one.label = Just a text box

test_two = text
test_two.label = Another text-box
```

```
test_two.value = Just a default value

test_three = checkbox
test_three.label = Some checkbox
test_three.value = 1

test_four = select
test_four.label = My selectbox
test_four.options = one|two|third option|four
test_four.value = 2

test_five = radio
test_five.label = Radio buttons are fun
test_five.options = uno|dos|tres|cuatro|cinco
test_five.value = 1

test_six = textarea
test_six.label = This is a large textarea
test_six.value = Default text
test_six.cols = 60
test_six.rows = 30
```

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

Reports Involving Custom Fields

The SQL required for TracReports to include custom ticket fields is relatively hard to get right. You need a `JOIN` with the `ticket_custom` field for every custom field that should be involved.

The following example includes a custom ticket field named `progress` in the report:

```
SELECT p.value AS __color__,
   id AS ticket, summary, component, version, milestone, severity,
   (CASE status WHEN 'assigned' THEN owner||' *' ELSE owner END) AS owner,
   time AS created,
   changetime AS _changetime, description AS _description,
   reporter AS _reporter,
  (CASE WHEN c.value = '0' THEN 'None' ELSE c.value END) AS progress
 FROM ticket t
    LEFT OUTER JOIN ticket_custom c ON (t.id = c.ticket AND c.name = 'progress')
    JOIN enum p ON p.name = t.priority AND p.type='priority'
 WHERE status IN ('new', 'assigned', 'reopened')
 ORDER BY p.value, milestone, severity, time
```

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

See also: TracTickets, TracIni