Project Properties

On this page

- What are Project Properties?
- Purpose of Project Properties
- Example

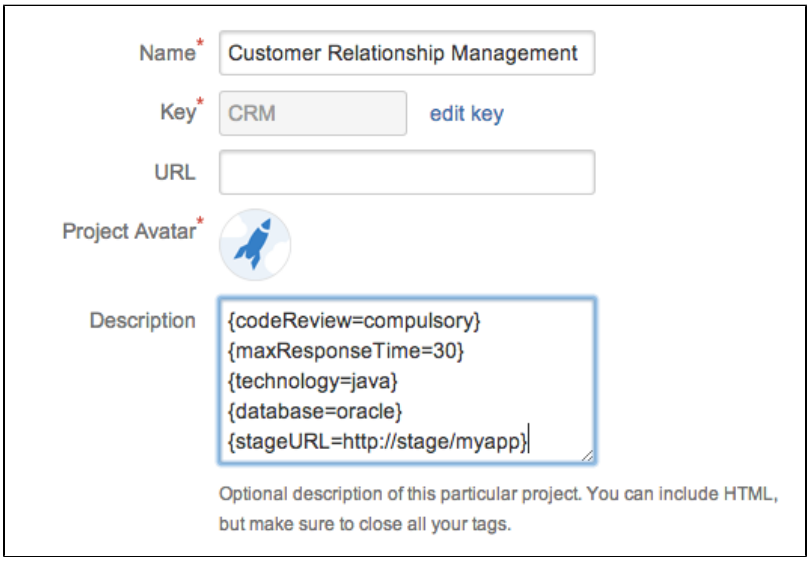
What are Project Properties?

Project Properties is a simple mechanism implemented by **Jira Workflow Toolbox**, to include machine readable **key-value pairs** into **projects description**, using format **{property=value}**, thereby operating as an effective way for **adding metadata to your Jira projects**.

Examples of Project Properties

The following screenshot shows Jira edit project details dialog with 5 examples of project properties:

- **codeReview**: used to enforce or to make optional certain part of the workflow related with code reviews.
- **maxResponseTime**: time elapsed since issue notification to first response. Can be used to set alarms with JQL filters and suscriptions, to increase priority when deadline approaches, etc.
- **technology**: project technology. Can be used in auto-composed texts, workflow customization, JQL searches, etc.
- **database**: database provider. Can be used in auto-composed texts, workflow customization, JQL searches, etc.
- **stageURL**: URL of stage environment. Can be used to compose an auto-comment to inform the user where to test the solution prior to release.



Purpose of Project Properties

Customizing workflow behavior depending on projects

Project Properties are initially aimed to **customize your workflows behavior depending on project features** (metadata added to projects in the form of Project Properties), this way **highly improving workflow reusability**.

For example, you can use project properties to:

- **Enable/disable transitions** depending on project properties values.
- Set project **customized values** for timelines (dates), costs, weights, or any other date or numerical values used in your workflows. Can also be used for implementing non-numerical or non-date attributes.
- **Set priorities among projects**, that maybe inherited to issues, for example by using priority adjustment tables.
- Provide **project related metadata** that may be used by the workflow post-functions to **compose project customized dynamic text**, setting custom or virtual fields with **project customized values**, ..., e.g., customer information, servers URLs, ...

Tools for customizing workflows

The tools available for customizing workflows are:

- **Conditions**:
  - **Check project property**: compare the value of a project property with a literal value, a custom or virtual field value, or a composed text, disabling a transition if comparison fails.
- **Validators**:
  - **Check project property**: compare the value of a project property with a literal value, a custom or virtual field value, or a composed text, showing the user a custom message if comparison fails.
- **Post-functions**:
  - **Set or create project property**: updates the value of a project property, or creates it in case it doesn't exist. Works on current issue's project, or on any project its key can be provided.
  - **Read a project property**: read the value of a project property and store it's value into a custom or virtual field. Works on current issue's project, or on any project its key can be provided.
- **Function in the boolean Expression Parser**, which is used in many features of the plugin:
  Function "**projectProperty(property_name)**" is available in the plugin's parser for reading the value of a project property. The parser is a common module of the plugin available for the following features of the plugin:
  - Condition " **Boolean Condition with math, date-time or text-string terms**"
  - Validator " **Boolean Validator with math, date-time or text-string terms**"
  - Post-function " **Set a field as a function of other fields**" and " **Mathematical and date-time expression calculator**".

Enhancing JQL search capability by project attributes

You can ask for the value of project properties in your JQL queries, by using **3 JQL functions** provided by **Jira Workflow Toolbox**:

- **projectsWhereTextProperty**(project_property, operator, string_literal | regular_expression)
- **projectsWhereNumberProperty**(project_property, operator, numeric_literal)
- **projectsWhereDateProperty**(project_property, operator, date_literal)

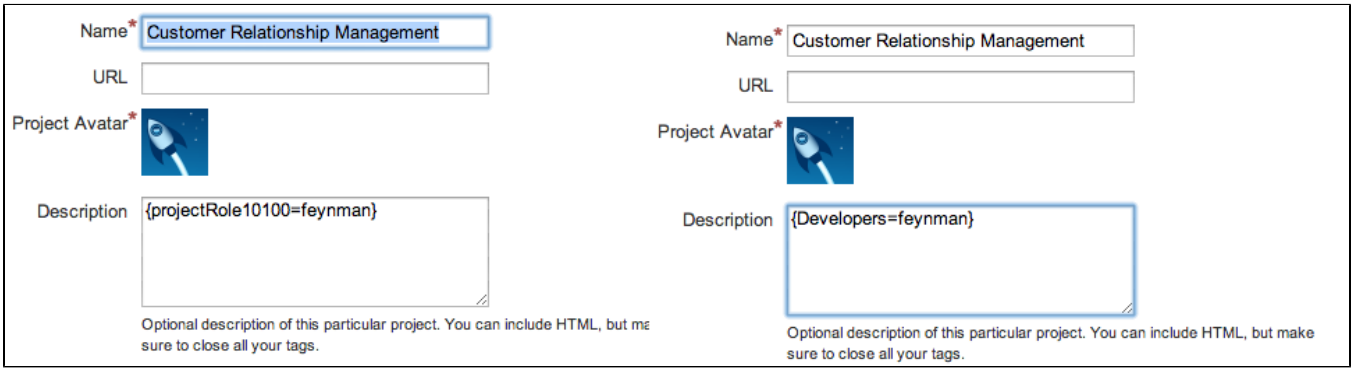
You can use it, among other things, to **extend the Jira native Project Category concept**, by creating many other project classifications by using Project Properties.

Setting default assignee for a Project Role

One of the two ways available in post-function " **Assign to project role**" for **setting default assignee for a project role** uses Project Properties with certain format.

Example

Setting user "**feynman**" as default assignee for project role "**Developers**", with internal identifier **10100**. As you can be seen in the picture, you can use the **name of the project role**, or the **internal identifier** as project property name. The second option has de advantage that it's immune to project role renaming.



Usage Examples

- **Check project property**
- **Validation based on the value of a Date type Project Property**