

Add a comment

This function has been **renamed** with the **JWT 3.0** release.

Find the new documentation at:

[Add comment](#)

On this page

- [Purpose](#)
- [Example: Create an automatic issue closing comment](#)
- [Other resources related to comments](#)
- [Usage Examples](#)
- [Related Features](#)

Purpose

This post-function creates a comment in current issue using the **content of a field**, setting the **visibility** by **project role**, by **user group** or Jira Service Desk's **internal** / **public** visibility modes. Comment **author** can be selected among user fields, or leave as **anonymous comment** using an empty field.

Text of the comment can be introduced directly in the post-function, or be composed previously using [Copy parsed text to a field](#) post-function, and stored into an **ephemeral string field**.

Comments can also be created in related issues using [Write field on linked issues or sub-tasks](#) or [Update issue fields](#) for writing into virtual fields "New comment" and "New comment (sends email notifications)".

Example: Create an automatic issue closing comment

Using [Jira Workflow Toolbox 2.2.1](#) or higher versions

We use **Add a comment** post-function for composing the text of the message and creating the comment with the desired visibility restrictions:

Comment's text: 	<input checked="" type="radio"/> Parsed Text Entered text will be used as comment's body.	1 Issue closed by <code>%{00021}</code> on <code>%{00057}</code> with <code>%{00028}</code> resolution.
	<input type="radio"/> Field Value of selected field will be used as comment's body.	Field code injector: Resolution - [Issue resolution] - %{00028}  <small>Field codes with format <code>%{nnnnn}</code> will be replaced with the corresponding values. Specific levels of Cascading Select fields can be referenced with <code>%{nnnnn.0}</code> for parent level, and <code>%{nnnnn.1}</code> for child level.</small>
Comment's author:	Assignee - [User]  <small>User that will appear as comment's authot. For anonymous comment you should select a non-initialized field.</small>	
Comment visibility:	Jira Service Desk Internal Comment (not visible to customers)  <small>Comment's visibility can be limited to users in a group, in a project role. For Jira Service Desk, internal comment and public comment visibility options are also available.</small>	
Group:	jira-administrators 	
Project Role:	Administrators 	
Notify by email:	<input type="checkbox"/> An Issue Commented event will be triggered.	

Conditional execution:
Optional boolean expression that should be satisfied in order to actually execute the post-function.
(Syntax Specification)

1

Leave the field empty for executing the post-function unconditionally.

Collection of Examples

[Line 1 / Col 1]

[Logical connectives](#): and, or and not. Alternatively you can also use &, | and !.
[Comparison operators](#): =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and != can be used with *strings*, *multi-valued fields* and *lists*.
[Logical literals](#): true and false. Literal null is used with = and != to check whether a field is initialized, e.g. {00012} != null checks whether Due Date is initialized.

String Field Code Injector:
Summary - [Text] - %{00000}

Numeric/Date Field Code Injector:
Original estimate (minutes) - [Number] - {00068}

Check Syntax

Creating comments in other issues:
Comments can also be created in related issues using ["Write field on linked issues or subtasks"](#) or ["Write field on issues returned by JQL query"](#) for writing into virtual field **"New comment"**.

Note that:

- **%{00021}** is field code for **"Current user's full name"**
- **%{00057}** is field code for **"Current day and time"**
- **%{00028}** is field code for **"Resolution"**

Using Jira Workflow Toolbox 2.2 or lower versions

We will use 2 post-function to implement this usage example:

- 1) We use [Copy parsed text to a field](#) post-function for composing the text of the comment, and storing it into **"Ephemeral string 1"**:

Target field:

Ephemeral string 1 - [Text]

Field to be written with the resulting parsed text.

☐ Don't overwrite target field if it's already set.

Parsing Mode:

☒ **Basic**
Basic mode: Insert field codes anywhere in the text, and they will be replaced with corresponding field values. Field code formats are **%{nnnnn}**, and **%{nnnnn.i}** for Cascading Select fields (i = 0 for base level).

☐ **Advanced**
Advanced mode: Strings literals are written in double quotes (*"This is a string."*). Operator ***+*** is used to concatenate strings, and field codes are like in basic mode, e.g., *"Issue key is " + %{00015} + "."*. More information at [parser syntax documentation](#).

Text to be parsed and then copied to target field:

Syntax Specification

1 Issue closed by %{00021} on %{00057} with "%{00028}" resolution.

Note that:

- **%{00021}** is field code for **"Current user's full name"**
- **%{00057}** is field code for **"Current day and time"**
- **%{00028}** if field code for **"Resolution"**

- 2) We us **Add a comment** post-function for creating a comment using the text stored in field **"Ephemeral string 1"**:

Comment's text:	<input type="text" value="Ephemeral string 1 - [Text]"/> <small>Field containing the text of the comment.</small>
Comment's author:	<input type="text" value="Current user - [User]"/> <small>User that will appear as comment's author. For anonymous comment you should select a non-initialized field.</small>
Comment visibility:	<input type="text" value="Everybody"/> <small>You can restrict comment's visibility to users in a certain group or playing a certain project role.</small>
Group:	<input type="text" value="jira-administrators"/>
Project Role:	<input type="text" value="Administrators"/>
<input type="button" value="Add"/> <input type="button" value="Cancel"/>	

Once configured, the transition looks like this:

IN PROGRESS
Done
DONE

Screen: CRM: Pantalla de resolución y seguimiento sencillos de incidencias

Triggers 0
Conditions 1
Validators 0
Post Functions 7

The following will be processed after the transition occurs Add post function

1. The following text parsed in **basic** mode will be copied to **Ephemeral string 1**:
Issue closed by %{Current user's full name} on %{Current date and time} with "%{Resolution}" resolution.
This feature will be run as user in field **Current user**.
2. Add a comment with text in field **Ephemeral string 1**, visible to **everybody**, and **Current user** as author.

Other resources related to comments

[Jira Workflow Toolbox](#) provides a series of virtual fields related with comments that can be used with all the features of the plugin:

Writable virtual fields

Field Name	Effect of Writing	Allowed String Values	Allowed Numeric Values	Examples
Last comment	Updates the last comment added to the issue if it exists	unlimited text string Optionally parameter visibility can be added for setting comment visibility. To do it add ": {visibility: visibility_value}" at the end of the text. This parameter admits the same values as " Last comment's visibility restriction ". (See example Creating a Jira Service Desk internal comment) Since version 2.2	numbers are cast to string	WARNING: When writing into this field, please add your writer post-function AFTER " Add a comment to an issue if one is entered during a transition. " post-function, since otherwise updated comment will be the one before last, instead of the last one. Example: Add current user to comment

Last comment's visibility restriction Formerly called "Last comment's visibility"	Updates the visibility of the last comment in the issue if it exists. Can be used to set the visibility of a comment introduced with virtual field " New comment "	name of a Project Role or name of a Group . In Jira Service Desk it also admits values public and internal , or alternatively jsd_public and jsd_internal to avoid name collision with existing project roles and user groups. Since version 2.2	n/a	Developers: for setting visibility to project role or group with name Developers. jsd_internal : for setting JSD internal visibility. jira-developers : for setting visibility to "jira-developers" user group.
New comment	Inserts a new comment into the issue	unlimited text string Optionally parameter visibility can be added for setting comment visibility. To do it add ": { visibility: visibility_value }" at the end of the text. This parameter admits the same values as " Last comment's visibility restriction ". (See example Creating a Jira Service Desk internal comment) Since version 2.2	numbers are cast to string	<ul style="list-style-type: none"> This is a comment only visible to developers group.: {visibility=jira-developers} This is a comment only visible to Admin project role.: {visibility=Administrators} This is a comment only visible to Jira Service Desk agents and collaborators.: {visibility=jsd_internal} This is a comment also visible to Jira Service Desk customers.: {visibility=jsd_public}

Virtual field "**New comment**" and "**New comment (sends email notifications)**" can be used in combination with [Write field on linked issues or sub-tasks](#) or [Update issue fields](#) for adding comments to other issues (linked issues, sub-task, JQL selected issues, or issues returned by an issue list expression).

Optional Parameter {visibility=visibility_value}

Since version **2.2** parameter **visibility** can be added at the end of comment's text in order to set visibility restrictions. To do it simply add ": {**visibility=visibility_value**}" at the end of the comment's text. Let's see some examples:

- This is a comment only visible to developers group.: {**visibility=jira-developers**}
- This is a comment only visible to Admin project role.: {**visibility=Administrators**}
- This is a comment only visible to Jira Service Desk agents and collaborators.: {**visibility=jsd_internal**}
- This is a comment also visible to Jira Service Desk customers.: {**visibility=jsd_public**}

Values returned when reading virtual fields

Field name	Type	Value returned when cast to string	Examples
Last comment	Text string	Last comment entered in the issue.	-
Last comment's visibility restriction	Text string	Name of a group or project role the visibility of the last comment entered in the issue is restricted to. If no restriction is applied to the comment, this field is returns an empty string . For Service Desk's special Internal visibility restriction, string Internal is returned. This field can be used to check visibility restriction of field entered in transition when it exists.	-
Last commenter (since version 2.1.33)	User	Name of the user who entered the last comment in the issue.	galileo.galilei
Transition's comment	Text string	Comment entered in transition screen. This field can be used to make comment mandatory in transition screen, or to validate the content of comment entered in a transition.	-

Usage Examples

Page: [Compose dynamic text by inserting field values in a text template](#)
Page: [Creating a Jira Service Desk internal comment](#)
Page: [Creating a Jira Service Desk internal comment on linked issues](#)

Related Features

- [Copy parsed text to a field](#)
- [Write field on linked issues or sub-tasks](#)
- [Update issue fields](#)