Parsing modes

On this page

General Information

There **are multiple parsing modes** available in the **expression parser**. The **two most commonly** used parsing modes are:

- **Basic**: with this simple parsing mode you can write free text and insert field codes with format **%{...somefield}** or **%{...somefield.i}** anywhere in your text. These **field codes will be replaced at runtime with the corresponding field values** of the issue currently being processed.
- **Advanced**: with this parsing mode we can do much more complex text composition thanks to the usage of functions for replacing substrings, changing case, reading fields in linked issues, sub-tasks, JQL selected issues, and much more. It requires the text to be parsed to be written as **string expression** respecting the **parser syntax**.

You can **easily switch between** parsing modes. The available modes **depend on the context**!

Your browser does not support the HTML5 video element

Most functions will accept **string** values so casting values to string is a very **powerful function**. Details can be found below in the **converting data types** section!
Additionally you can directly transform a field value to text using the following syntax: **%{...somefield}**

Parsing modes overview

Mode	Supported features	Return type	Example
Basic	Field codes	STRING	The basic parsing mode supports the usage of field codes. Field codes can be used to access issue field values. **simple text using a field code to read the summary** `This is the issue summary: %{trigger.issue.summary}`
Advanced	Field codes Parser Functions	STRING	The advanced parsing mode has a defined syntax that allows you to write functions to read and manipulate data from any issue in Jira. Field codes are supported as well as clear text, written in quotation marks. **Advanced expression to read the issue summary and use a function to get the assignee mail address** `"This is the issue summary:" + %{trigger.issue.summary} + " and the assignee mail is: " + userEmail(%{trigger.issue.assignee})`

Math /date time	Field codes
	Parser Functions
		NUMBER
		DATE
		DATE_TIME
			The mathematical and date time parsing mode works like the advanced mode but expect a number as result instead of a string. The resulting number is used to updated numeric or date time fields. In case of date or date time fields the number will be cast to a date.
			Time to resolve the issue
			`{trigger.issue.resolutionDate} -`
			`{trigger.issue.createdDate}`

Logical	Field codes
	Parser Functions
		BOOLEAN	The logical parsing mode works like the advanced parsing mode but expression result must return true or false.
			Check if the assignee is equals the reporter
			`{trigger.issue.assignee} = {trigger.`
			`issue.reporter}`

Issue List	Field codes Parser Functions	ISSUE []	
String List	Field codes Parser Functions	STRING []	
Mixed	Field codes Parser Functions written in three curly braces	STRING	

Automatic parsing mode converter: You can write your text in **basic mode**, and then switch to **advanced mode**. The text to be parsed will be automatically rewritten as a string expression. Now, you can simply make the modifications you require, making use of text formatting functions, or inserting math or time expressions where needed.

To update issue fields the parsing result will be cast to the expected value e.g. a user name will be cast to a user to update a user field like the assignee field.