Virtual fields

Automation Toolbox for Jira provides a set of special fields called **virtual fields**, making almost all **properties of issues, projects and users** accessible to every feature in the app.

Virtual fields may be **read** and **written** by Automation Toolbox for Jira in the same way ordinary custom fields are.

Virtual fields and their associated **field codes,** were created to

- provide data accessibility beyond the scope of normal Jira workflow processing
- insure **data integrity** throughout their use.

In Automation Toolbox for Jira you can use virtual fields by searching for and picking their **associated field codes** in the dropdown menus provided wherever a **parser expression** can be inserted.

Your browser does not support the HTML5 video element

The notation listed below references each virtual field code, independent from **it's context**.

> ⓘ **Remember**
>
> The virtual fields represent the **field / data** part of the field code notation.
>
> **%{trigger.parent.summary}**
>
> **%{system.currentUser}**
>
> Some virtual fields are only available when used in the right context!

Readable fields

Issue-related fields

Field name	Field code	Type	Output	Example
Summary	`%{issue.summary}`	STRING	Issue summary.	-
Description	`%{issue.description}`	STRING	Issue description.	-
Assignee	`%{issue.assignee}`	USER	The name of the user who has the issue assigned.	**albert.einstein**
Assignee's full name	`%{issue.assignee.fullName}`	STRING	Name and surname of the user who currently has the issue assigned	**Albert Einstein**

Assignee's email	`%{issue.assignee.email}`	STRING	Email address of the user who currently has the issue assigned	**albert.einstein@yahoo.com**
Reporter	`%{issue.reporter}`	USER	The name of the user who reported the issue. This field is editable, while **Creator** field isn't.	**albert.einstein**
Reporter's full name	`%{issue.reporter.fullName}`	STRING	Name and surname of the user who created the issue	**Albert Einstein**
Reporter's email	`%{issue.reporter.email}`	STRING	Email address of the user who created the issue	**albert.einstein@yahoo.com**
Creator	`%{issue.creator}`	USER	User name of the user who actually creates the issue in JIRA. This field, unlike **Reporter**, can't be edited.	**richard.feynman**
Creator's full name	`%{issue.creator.fullName}`	STRING	Name and surname of the user who actually created the issue in JIRA	**Richard Feynman**
Creator's email	`%{issue.creator.email}`	STRING	Email address of the user who actually created the issue in JIRA	**richard.feynman@gmail.com**
Date and time of creation	`%{issue.created}`	DATE_TIME	Date and time when issue has been created. When cast to string format defined at **jira.date.time.picker.java.format** is used.	**19/Mar/14 1:38 PM** for **jira.date.time.picker.java.format = dd/MMM/yy h:mm a**
Date and time of last update	`%{issue.updated}`	DATE_TIME	Date and time of the most recent issue update. When cast to string format defined at **jira.date.time.picker.java.format** is used.	**19/Mar/14 1:38 PM** for **jira.date.time.picker.java.format = dd/MMM/yy h:mm a**
Date and time of latest status change	`%{issue.lastStatusChange}`	DATE_TIME	Date and time of the most recent status change in the issue. When cast to string format defined at **jira.date.time.picker.java.format** is used.	**19/Mar/14 1:38 PM** for **jira.date.time.picker.java.format = dd/MMM/yy h:mm a** Useful to find out for how much time is the issue in current status using expression: `{00057}` – `{000158}`, where `{00057}` = **Current date and time** and `{000158}` is **Date and time of latest status change**.
Due date	`%{issue.dueDate}`	DATE	Field that can store a date with no time part, and is used to schedule issues. When cast to string format defined at **jira.date.time.picker.java.format** is used.	**19/Mar/14** for **jira.date.time.picker.java.format = dd/MMM/yy h:mm a**, since this field doesn't store time part.

Date and time of resolution	`%{issue.resolutionDate}`	DATE_TIME	Date and time of the most recent update of field *Resolution*. When cast to string format defined at **jira.date.time.picker.java.format** is used.	**19/Mar/14 1:38 PM** for **jira.date.time.picker.java.format = dd/MMM/yy h:mm a**
Priority	`%{issue.priority}`	PRIORITY	Name of the priority in **Default language** configured in the JIRA instance	**Blocker** **Critical** **Major**
Original estimate (minutes)	`%{issue.originalEstimate}`	NUMBER	Time originally estimated for the issue expressed in minutes. It's a real number, so it may have a non-zero fractional part.	-
Remaining estimate (minutes)	`%{issue.estimate}`	NUMBER	Remaining time estimated for the issue expressed in minutes. It's a real number, so it may have a non-zero fractional part.	-
Total time spent (minutes)	`%{issue.timeSpent}`	NUMBER	Work time currently spent at the issue expressed in minutes. It's a real number, so it may have a non-zero fractional part.	-
Components	`%{issue.component}`	COMPONENTS	Comma separated list of component names	**Web Site, Authenticator, Statistics**
Components leaders	`%{issue.component.leads}`	MULTIUSER	Comma separated list of user names	**albert.einstein@yahoo.com, richard.feynman@gmail.com**
Fixed versions	`%{issue.fixVersion}`	VERSIONS	Comma separated list of fixed versions	**1.0, 2.0, 2.1**
Fixed versions with details	`%{issue.fixVersion.details}`	STRING	A text with a line for each fixed versions. Each line contains the following information separated by characters '#': name of version, description, release date and archive situation	**1.0 # First release # RELEASED ON 28/Mar/14 12:00 AM # ARCHIVED**
Number of fixed versions	`%{issue.fixVersion.count}`	NUMBER	Number of fixed versions in current issue.	-
Affected versions	`%{issue.version}`	VERSIONS	Comma separated list of fixed versions	**1.0, 2.0, 2.1**
Affected versions with details	`%{issue.version.details}`	STRING	A text with a line for each affected versions. Each line contains the following information separated by characters '#': name of version, description, release date and archive situation	**1.0 # First release # RELEASED ON 28/Mar/14 12:00 AM # ARCHIVED**

Number of affected versions	`%{issue.version.count}`	NUMBER	Number of affected versions in current issue.	-
Number of votes received	`%{issue.votes}`	NUMBER	Number of votes received by the issue.	-
Keys of subtasks	`%{issue.subtasks}`	STRING	Comma separated list of issue keys.	**CRM-23, CRM-26, CRM-31**
Number of subtasks	`%{issue.subtasks.count}`	NUMBER	Number of subtasks of current issue.	-
Keys of linked issues	`%{issue.linkedIssues}`	STRING	Comma separated list of issue keys.	**CRM-13, HR-12, SDESK-45**
Number of linked issues	`%{issue.linkedIssues.count}`	NUMBER	Number of issues linked to current issue.	-
Environment	`%{issue.environment}`	STRING	Field to describe the environment where the issue has occurred.	-
Attachments	`%{issue.attachment}`	ATTACHMENT	Comma separated list of file names of all files attached to the issue.	**file1.txt, readme.pdf, screenshot.png**
Number of attachments	`%{issue.attachment.count}`	NUMBER	Number of files attached to current issue.	-
Labels	`%{issue.labels}`	LABELS	Blank separated list with all the labels of the issue.	**web customer java mobile**
Number of labels	`%{issue.labels.count}`	NUMBER	Number of labels in current issue.	-
Issue key	`%{issue.key}`	STRING	Issue key	**CRM-25** **HR-52** **SDESK-1**
Issue type	`%{issue.issueType}`	ISSUETYPE	Name of issue's **Issue Type** in Jira's default language	**Bug** **Improvement** **New Feature** **Task**

Issue status	`%{issue.status}`	STATUS	Name of current status of the issue in Jira's default language.	**Open** **In Progress** **Resolved**
Issue status category	`%{issue.statusCategory}`	STATUS_CATEGORY	Name of the category in Jira's default language.	**To Do** **In Progress** **Done**
Previous issue status	`%{issue.status.previous}`	STATUS	Name of the status the issue was just before current one. The name is shown in Jira's default language.	**Open** **In Progress** **Closed**
Previous issue status category	`%{issue.statusCategory.previous}`	STATUS_CATEGORY	Name of the category of the status the issue was just before current one. The name is shown in Jira's default language.	**To Do** **In Progress** **Done**
Available transitions	`%{issue.transitions.achievable}`	STRING	A comma separated list with the **names of transitions** with origin in current issue status.	If current issue is in **Open** status of classic Jira workflow, then field **Available transitions** will return: `Start Progress, Resolve Issue, Close Issue`
Available target statuses	`%{issue.status.achievable}`	STRING	A comma separated list with the **names of target status** of those transitions with origin in current issue status.	If current issue is in **Open** status of classic Jira workflow, then field *Available target statuses* will return: `In Progress, Resolved, Closed`
Resolution	`%{issue.resolution}`	RESOLUTION	The name of the resolution currently set at the issue, or empty string if **Resolution** is not set. The name of the resolution is shown in Jira's default language.	**Resolved** **Won't fix** **Cancelled**
Security level	`%{issue.securityLevel}`	SECURITY	Name of the security level the issue currently has.	**Classified**
Last comment	`%{issue.lastComment}`	STRING	Last comment entered in the issue.	-
Last comment's visibility restriction	`%{issue.lastComment.visibility}`	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)	`%{issue.lastComment.author}`	USER	Name of the user who entered the last comment in the issue.	**galileo.galilei**

Watchers	`%{issue.watcher}`	MULTIUSER	Comma separated list of user names of current watchers of the issue.	**albert.einstein, richard.feynman, galileo.galilei**
Project key	`%{issue.project.key}`	STRING	**Project Key** of the project the issue belongs to.	**CRM, HR, SDESK**
Project name	`%{issue.project.name}`	STRING	Name of the project the issue belongs to.	**Customer Relationship Management**
Project description	`%{issue.project.description}`	STRING	Field **Description** of the project the issue belongs to.	**-**
Project URL	`%{issue.project.url}`	STRING	Field **URL** of the project belongs to.	**https://www.decadis.de**
Project category	`%{issue.project.category}`	STRING	Name of the **Category** of the project the issue belongs to.	**-**
Project leader	`%{issue.project.lead}`	USER	Name of the user who leads the project the issue belongs to.	**galileo.galilei**
Project leader's full name	`%{issue.project.leadFullName}`	STRING	Name and surname of the user who leads the project the issue belongs to.	**Galileo Galilei**
Project leader's email	`%{issue.project.leadEmail}`	STRING	Email address of the user who leads the project the issue belongs to.	**galileo.galilei@me.com**
Rest of issues in the project	`%{issue.remainingIssuesInProject}`	STRING	Comma separated list of issue keys in the project except current issue's key.	**CRM-1, CRM-2, CRM-3, CRM-4**
Workflow scheme	`%{issue.workflowScheme}`	STRING	The name of the workflow scheme of the project current issue belongs to.	**-**

Sprint ID	`%{issue.sprintId}`	STRING	The internal ID for the sprint of current issue. This value can be used for setting field **Sprint** in other issues.	-
Sprint Start Date	`%{issue.sprintStartDate}`	DATE	Sprint's start date.	-
Sprint End Date	`%{issue.sprintEndDate}`	DATE	Sprint's end date.	-

Project-related fields

Project key	`%{trigger.project.key}`	STRING	**Key** of the project from the event.	CRM, HR, SDESK
Project name	`%{trigger.project.name}`	STRING	Name of the project from the event.	**Customer Relationship Management**
Project description	`%{trigger.project.description}`	STRING	Field **Description** of the project from the event.	**-**
Project URL	`%{trigger.project.url}`	STRING	Field **URL** of the project from the event.	**https://www.decadis.de**
Project category	`%{trigger.project.category}`	STRING	Name of the **Category** of the project from the event.	**-**
Project leader	`%{trigger.project.lead}`	USER	Name of the user who leads the project from the event.	**galileo.galilei**
Project leader's full name	`%{trigger.project.leadFullName}`	STRING	Name and surname of the user who leads the project from the event.	**Galileo Galilei**
Project leader's email	`%{trigger.project.leadEmail}`	STRING	Email address of the user who leads the project from the event.	**galileo.galilei@me.com**

Component-related fields

Component id	`%{trigger.component.id}`	STRING	**Id** of the component from the event.	**10000**
Component name	`%{trigger.component.name}`	STRING	Name of the component from the event.	**Customer Relationship Management Component**
Component description	`%{trigger.component.description}`	STRING	Field **Description** of the component from the event.	**-**
Component lead	`%{trigger.component.lead}`	USER	Name of the user who leads the component from the event.	**galileo.galilei**
Component default assignee	`%{trigger.component.defaultAssignee}`	USER	Name of the user who is the default assignee of the component from the event.	**galileo.galilei**

Version-related fields

Version id	`%{trigger.version.id}`	STRING	**Id** of the version from the event.	**10000**
Version name	`%{trigger.version.name}`	STRING	Name of the version from the event.	**Customer Relationship Management Version**

Version description	`%{trigger.version.description}`	STRING	Field **Description** of the version from the event.	-
Version archived	`%{trigger.version.archived}`	STRING	Archived status of the version from the event.	-
Version released	`%{trigger.version.released}`	STRING	Released status of the version from the event.	-
Version start date	`%{trigger.version.startDate}`	DATE	Start date of the version from the event.	-
Version release date	`%{trigger.version.releaseDate}`	DATE	Release date of the version from the event.	-
Version project	`%{trigger.version.project}`	STRING	Project key of the version from the event.	-
Version project id	`%{trigger.version.projectId}`	STRING	Project id of the version from the event.	-
Version sequence	`%{trigger.version.sequence}`	STRING	Sequence of the version from the event.	-

User-related fields

User id	`%{trigger.user.id}`	STRING	**Id** of the user from the event.	**10000**
User name	`%{trigger.user.name}`	STRING	Name of the user from the event.	**Customer Relationship Management Component**
User full name	`%{trigger.user.fullName}`	STRING	Full name of the user from the event.	**Galileo Galilei**
User email	`%{trigger.user.email}`	STRING	Email address of the user from the event.	**galileo.galilei@me.com**
User key	`%{trigger.user.key}`	USER	Key of the user from the event.	**galileo.galilei**
User directory id	`%{trigger.user.directoryId}`	STRING	Directory id of the user from the event.	-
User directory user	`%{trigger.user.directoryUser}`	STRING	Directory user of the user from the event.	-
User is active	`%{trigger.user.isActive}`	STRING	Active status of the user from the event.	**true**

System-related fields

Current user	`%{system.currentUser}`	USER	Name of the user who is executing the event.	**isaac.newton**
Current user's full name	`%{system.currentUserFullName}`	STRING	Name and surname of the user who is executing the event.	**Isaac Newton**
Current user's email	`%{system.currentUserEmail}`	STRING	Email address of the user who is logged in.	**albert.einstein@yahoo.com, isaac.newton@gmail.com**
Current date and time	`%{system.currentDateTime}`	DATE_TIME	Current date and time of JIRA server's clock. When cast to string format defined at **jira.date.time.picker.java.format** is used.	**19/Mar/14 1:38 PM** for **jira.date.time.picker.java.format = dd/MMM/yy h:mm a**
JIRA base URL	`%{system.baseUrl}`	STRING	Returns the base URL of current JIRA instance.	-

Writable fields

When a value is written into these fields, a feature or system field of the issue the field belongs to will be modified. These fields are also readable.

Field Name	Field code	Effect of Writing	Allowed String Values STRING	Allowed Numeric Values NUMBER	Examples
Summary	`%{issue.summary}`	Sets issue **Summary**	text string trimmed to 255 characters	numbers are cast to string	-
Description	`%{issue.description}`	Sets issue **Description**	unlimited text string	numbers are cast to string	-
Assignee	`%{issue.assignee}`	Assigns the issue to a user	**user name** (not user's full name)	n/a	-
Reporter	`%{issue.reporter}`	Updates issue **Reporter** to a different user	**user name** (not user's full name)	n/a	-
Due date	`%{issue.dueDate}`	Sets issue's system field **Due Date**	date-time in format defined at **jira.date.time.picker.java.format**, or as in JQL (yyyy/MM/dd HH:mm, yyyy-MM-dd HH:mm, yyyy/MM/dd, yyyy-MM-dd, or relative to current time using "w" (weeks), "d" (days), "h" (hours) or "m" (minutes)	number of minutes relative to current time.	`2014-03-25` : date example `2015-01-31 14:35` : date with time example `- 4d 1h` : 4 days and 1 hour before current time. `2w, 3d, 4h, 5m` : 2 weeks, 3 days, 4 hours and 5 minutes after current time.
Priority	`%{issue.priority}`	Updates the issue **Priority**	**name of a Priority** (e.g., Blocker, Critical, Major,...)	(0 = highest priority, 1 = second highest,... n = lowest	
Original estimate (minutes)	`%{issue.originalEstimate}`	Sets original estimate	cast from string to number will be attempted	number of minutes	`30`: sets **Original estimate** to 30 minutes. **Original estimate (minutes)** can be increased or decreased using expression parser with formulas like: `{00024} + 60` for increasing remaining estimate in 1 hour, or `{00024} - {10000}` for decreasing remaining estimate by the amount of minutes stored in custom field with code `{10000}`.
Remaining estimate (minutes)	`%{issue.estimate}`	Sets remaining estimate	cast from string to number will be attempted	number of minutes	`30` : sets **Remaining estimate** to 30 minutes. **Remaining estimate (minutes)** can be increased or decreased using expression parser with formulas like: `{00024} + 60` for increasing remaining estimate in 1 hour, or `{00024} - {10000}` for decreasing remaining estimate by the amount of minutes stored in custom field with code `{10000}`.

Total time spent (minutes)	`%{issue.timeSpent}`	Sets time spent.+	cast from string to number will be attempted	number of minutes	`30` : sets **Time spent** to 30 minutes. Total time spent (minutes) can be increased or decreased using expression parser with formulas like: {00024} + 60 for increasing remaining estimate in 1 hour, or {00024} - {10000} for decreasing remaining estimate by the a mount of minutes stored in custom field with code {10000}.
Add to time spent (minutes)	`%{issue.addTimeSpent}`	Adds a number of minutes to current value of field "**Total time spent (minutes)**"	cast from string to number will be attempted	number of minutes	`60` : increases time spent in 1 hour.
Components	`%{issue.component}`	Sets issue **Components**	Comma separated list of **component names**. Prefixes + and - can be used to add or remove (check or uncheck) single values or set of values.	n/a	`+ Web Site, User Management` : adds the components **Web Site** and **User Management** to current issue, provided they exist in the project. `- Library A, Component B` : removes the components **Library A** and **Component B** from current issue, provided they exist in the project.
Fixed versions	`%{issue.fixVersion}`	Sets **Fixed versions**	Comma separated list of **version names.** Prefixes + and - can be used to add or remove (check or uncheck) single values or set of values.	numbers are cast to string	`+ 1.0, 1.1` : adds versions **1.0** and **1.1** to field **Fixed versions**. `- 2.0, 2.2` : removes versions **2.0** and **2.2** from field **Fixed versions**.
Affected versions	`%{issue.version}`	Sets **Affected versions**	Comma separated list of **version names**. Prefixes + and - can be used to add or remove (check or uncheck) single values or set of values.	numbers are cast to string	`+ 1.0, 1.1` : adds versions **1.0** and **1.1** to field **Affected versions**. `- 2.0, 2.2` : removes versions **2.0** and **2.2** from field **Affected versions**.
Environment	`%{issue.environment}`	Sets issue **Environment**	unlimited text string	numbers are cast to string	-
Issue status	`%{issue.status}`	Will make the issue progress through the workflow to the written status, provided there is transition in the workflow from current status to the written status. It requires that all conditions and validations in the transition are satisfied.	**name of a Status** (e.g., Open, In Progress, Resolved,...)	n/a	`Resolved` : will search for a transition from current issue's status to **Resolved** status. If conditions are satisfied, it will try to execute the transition. Then if validators are satisfied in that transition are satisfied, it will be executed, resulting in moving issue to **Resolved** status. Every post-function in the executed transition is also executed.
Resolution	`%{issue.resolution}`	Sets issue **Resolution**	**name of issue Resolution** (e.g., Fixed, Won't Fix, Duplicate, Incomplete,...)	n/a	-
Labels	`%{issue.labels}`	Sets **Labels**	Comma separated list of **labels**. If label doesn't exist it is created. Prefixes + and - can be used to add or remove (check or uncheck) single values or set of values.	numbers are cast to string	`+ jira, plugin` : adds labels "**jira**" and "**plugin**" to the issue. `- scrum, web` : removes labels "**scrum**" and "**web**" from the issue.
Attachments (only new attachments will be added)	`%{issue.attachmentsAdd}`	Adds new attachments coming from another issue keeping current attachments. Rejects duplicated attachments.	Comma or blank separated list of **issue keys** whose attachments will be copied to current issue.	n/a	`CRM-1` : adds to current issue the attachments in issue **CRM-1** except if attachments are already in current issue. `CRM-2, HR-34, HR-50` : adds to current issue the attachments in issues **CRM-2, HR-34** and **HR-50** except if attachments are already in current issue.

Attachments with details	`%{issue.attachment.details}`	Text string	Comma separated list of file names with its mime types and sizes in Kbytes, for every file attached to the issue.		**file1.txt (text/plain, 5.14 KB), readme.pdf (application/pdf, 179.8 KB), screenshot.png (image/png, 5.449 KB)**
Security level	`%{issue.securityLevel}`	Sets issue **Security Level**	**name of a Security Level**	n/a	`Public` : sets security level to **Public**, provided this security level exists in the Security Scheme of the project. `Reserved` : sets security level to **Reserved**, provided this security level exists in the Security Scheme of the project.
Last comment	`%{issue.lastComment}`	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 permits same values as "**Last comment's visibility restriction**".	numbers are cast to string	**WARNING**: When writing into this field, please add your writer action AFTER "**Add Comment**" action since otherwise updated comment will be the one before last, instead of the last one.
Last comment's visibility restriction	`%{issue.lastComment.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` (without doble quotes), or alternatively `jsd_public` and `jsd_internal` to avoid name collision with existing project roles and user groups.	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.
Watchers	`%{issue.watcher}`	Sets watchers of an issue. Can be used to add or remove watchers to linked issues, sub-tasks or issues returned by JQL, but requires to use operator '+' and '-'. Also can be used to copy watchers between issues.	comma separated list of **user names, group names** or **project roles names**. Prefixes + and - can be used to add or remove single users or set of users.	n/a	`albert, richard, john` : sets 3 users as watchers replacing current ones. `+ Developers, Administrators` : adds every user in project roles or groups **Developers** and **Administrators** to current watchers. `- Testers, Administrators, + albert, richard` : removes users in project roles or groups **Testers** and **Administrators** and adds users **albert** and **richard** as watchers.