

Keep track of important status updates

On this page

[Prerequisites](#) | [Configuration](#) | [Screenshots](#) | [Related use cases](#)

Use case

Status updates are key to successfully managing projects. Usually you need to capture additional information which often gets lost in the comment history.

This use case combines **multiple custom text fields** and populates the automatically using **post functions**.

DECADIS APPS

Dashboards

It's me! Project Shuttle

Projects

Issues

Boards

xPort

Charts

Create

Marketing Creatives / MAC-5

Website - Product Page Graphics

Edit

Comment

Assign

More

Status Update

Ready For Review

Stop Progress

Admin

Details

Type: Task

Priority: Critical

Labels: None

Status: DRAFT (View Workflow)

Resolution: Unresolved

General

Status History

Current status:

2020-07-28 - Alana Grant

Graphics added to website by Ryan

Dashboards
It's me! Project Shuttle
Projects
Issues
Boards
xPort
Charts
Create

Marketing Creatives / MAC-5

Website - Product Page Graphics

Edit
Comment
Assign
More
Status Update
Ready For Review
Stop Progress
Admin

Details

Type: Task
Priority: Critical
Labels: None
Status: DRAFT (View Workflow)
Resolution: Unresolved

General
Status History

Status History:

2020-07-28 - Alana Grant
Graphics added to website by Ryan
2020-06-01 - Alana Grant
Grahpics approved by Cassie
2020-05-20 - Admin Istrator
Graphics currently in review

Prerequisites

Create three custom fields

- 1

Create a custom field of type: **Text Field (multi-line)**
Name: **Status Update**
- 2

Create a second custom field of type: **Text Field (multi-line)**
Name: **Current status**
- 3

Create a third custom field of type: **Text Field (multi-line)**
Name: **Status History**
- 4

Navigate to the **Field configuration** and set the renderer of the 3 custom fields to "**Wiki Style Renderer**".

Create a transition screen

- 1

Create a new **screen**
Name: **Status Update Screen**

2

Add the field **Status Update** to the newly created screen.

Set up a workflow transition

1

Create a new transition with the following parameters:

From Status: **Any**

To Status: **Itself**

Name: **Status update**

Screen: **Status Update Screen**

2

Add a [Fields required or changed validator](#) to the newly created transition.

3

Field

Status Update

4

Condition

Is required

5

Message to show when validation fails

Please enter a status update!

6

Once set up, the validator should look something like this:

Update parameters of the 'Fields required or changed' Validator for this transition.

Update parameters of the 'Fields required or changed' Validator for this transition.

Fields*

Define the fields to be validated.

Field	Validation		
☒ Status Update (12903)	is required	☒ STATUS UPDATE (12903)	 
Add field			

> Validation options

Error message

Add an **optional** error message to show when a validation fails.

Location

Show on top

Define where the error message should be displayed.

Message

  Please enter a status update!

Enter plain text and optionally use **field codes**, e.g. `%issue.summary`, to insert field values.

Add translation

Add **optional** error message translations. Translations will be displayed based on the user's language setting.

Fields required or changed

Documentation 

Check if fields are set/empty or have changed on the transition screen.

Workflow

JWT 3.0.0 Use cases



Examples

Pick one of the following example and get started in no time. More provided upon your selection.

- Make fix version required
- Enforce a re-assignment



Configuration

1

Add an **Update or copy field values** post function to the **workflow transition** created in the **prerequisites** section.

2

Target issue

Choose **Current issue**

3

Configure the field: **Current Status**

1

Field

Choose **Current Status**

2

Expression

Set the parsing mode to **Advanced text** and use the following expression

```
"*" + dateTimeToString({system.currentDateTime}, "yyyy-MM-dd", USER_LANG) + "*"
- "*" + {%system.currentUserFullName} + "*\n" + {%issue.cfnnnnn}
```

{%issue.cfnnnnn} is the field code for the field **Status Update**

4

Configure the field: **Status History**

1

Field

Choose **Status History**

2

Expression

Set the parsing mode to **Advanced text** and use the following expression

```
%{issue.cfaaaaa} + "\n\n" + %{issue.cfbbbbbb}
```

%{issue.cfaaaaa} is the field code for the field **Current Status**

%{issue.cfbbbbbb} is the field code for the field **Status History**

5

Configure the field: **Status Update**

1

Field

Choose **Status Update**

2

Value

Choose **Clear field value**

The field value **must be cleared**, so users will be presented with an empty text field the next time they leave a status update.

6

User

Choose **Current user**



Screenshots

Update parameters of the Update or copy field values Function for this transition.

Update parameters of the Update or copy field values Function for this transition.

Basic configuration *

Define the required parameters needed to execute this post function.

Target issue *

Current issue

Select the issues to be updated.

Fields *

Define the fields to be updated.

Field	Value	Write protection
Current status	Parser expression "" + dateTimeToString ...	
Status History	Parser expression %{issue.cf12903} + "\n ...	
Status Update (12903)	Clear field value	

Add field

- Additional options
- Conditional execution
- Run as

Update or copy field values



Documentation

Update one or multiple issue fields or copy field values across the same issue or related issues (parent / child, Epic / Story, linked issues and many more).

Workflow

JWT 3.0.0 Use cases



Examples

Pick one of the following example configurations and get started in no time. More details will be provided upon your selection.

- Assign issue to current user
- Keep parent's priority in sync
- Assign important issues to the project lead

Related use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Add days skipping weekends and holidays to a Date Picker field		Update or copy field values	addTime()	
Add or remove request participants		Update or copy field values		
Add watchers ignoring inactive users		Update or copy field values	usersInGroup() isActive() toString() filterByPredicate()	
Assign important issues to the project lead		Update or copy field values		
Assign issue to current user		Update or copy field values		
Automatically fill an insight custom field after a transition		Update or copy field values		
Change the assignee to the next evaluator		Update or copy field values	first() toStringList()	

Create an internal Service Management comment on linked issues		Update or copy field values	
Keep parent issue's priority in sync		Update or copy field values	
Keep track of important status updates		Update or copy field values Fields required or changed	dateTimeToString()
Match several values of a list		Update or copy field values	toString() distinct() filterByPredicate()
Prioritize the issues globally		Update or copy field values	indexOf() previousValue() issuesFromJQL()
Set a date on the same week day on alternate weeks		Update or copy field values	addDays() nextDayOfTheWeek() modulus() weekOfTheYear() datePart()
Set a Date Picker field to the nth day of the month		Update or copy field values	dayOfTheMonth() lastDayOfTheMonth() addDays()
Set assignee based on a former assignee		Update or copy field values	previousValue()
Set Due Date with latest value among sub-tasks		Update or copy field values	fieldValue() max() siblingSubtasks()
Set fix version based on its start and release date		Update or copy field values	toString() textOnStringList() unreleasedVersions() startDates() releaseDates() first()
Set the next fix version		Update or copy field values	floor() toNumber() substring() length()
Shorten the summary to a maximum number of characters		Update or copy field values	substring() length()

If you still have questions, feel free to refer to our [support](#) team.