Import data from a CSV file

support.atlassian.com/jira-cloud-administration/docs/import-data-from-a-csv-file

When migrating from another issue tracker, you can export data to a comma-separated value (CSV) file and then import that file into your Jira Cloud applications. CSV files are text files representing tabulated data and are supported by most applications that handle tabulated data.

If you're importing a large number of issues into Jira cloud applications, you can split the file into smaller chunks, and then import these CSV files individually.

Preparing your CSV file

When creating a CSV file that can be imported to Jira, make sure that all fields are separated by commas and that any piece of content (including commas and new lines) is enclosed in quotes. Keep on reading to learn more about specific file requirements and data structure.

If you're using Microsoft Excel and OpenOffice, it is not necessary to quote values in cells as these applications handle this automatically.

CSV file requirements

Each CSV file must possess a heading row with a Summary column

The CSV file import wizard uses the CSV file header row to determine how to map data from the file's 2nd row and beyond to fields in Jira.

The header row *should avoid containing any punctuation* (apart from the commas separating each column) or the importer may not work correctly.

The header row *must* contain a column for Summary data.

Commas (as column/field separators) cannot be omitted

For example, this is valid:

```
Summary, Assignee, Reporter, Issue Type, Description, Priority
"Test issue", bob@example.com, bob@example.com, 1, ,
```

... but this is not valid:

```
Summary, Assignee, Reporter, Issue Type, Description, Priority
"Test issue", bob@example.com, bob@example.com, 1
```

CSV file structure

Capturing data that spans multiple lines

Use double-quote marks (") in your CSV file to capture data that spans multiple lines. For example, Jira will treat the following as a valid CSV file with a single record:

```
Summary, Description, Status "Login fails", "This is on a new
line", Open
```

Treating special characters literally

Use double-quote marks (") around a section of text to treat any special characters in that section literally. Once this data is imported, these special characters will be stored as part of Jira's field data. Examples of special characters include carriage returns/enter characters, commas, etc.

To treat a double quote mark literally, you can add another double quote mark character:

```
"Clicking the ""Add"" button results in a page not found error"
```

...which, once imported, will be stored in Jira as:

```
Clicking the "Add" button results in a page not found error
```

Aggregating multiple values into single fields

You can import multiple values into a single Jira field that accepts multiple values (e.g. Fix (for) Version, Affects Version, Component, Labels). To do this, your CSV file must specify the same column name for each value you wish to import to the same field. The number of column names specified must match the maximum number of values to be aggregated into the mapped field.

For example:

```
IssueType, Summary, FixVersion, FixVersion, FixVersion,
Component, Component bug, "First issue", v1, , , Component1, bug, "Second
issue", v2, , , Component1, Component2 bug, "Third issue", v1, v2, v3,
Component1,
```

In the above example, the Component field of the second issue and the Fix Version field of the third issue will generate multiple values in the relevant field after importing.

Importing attachments

You can attach files to issues created from your CSV file. To do this, specify the URL of your attachment in an Attachments column within your CSV file.

```
Assignee, Summary, Description, Attachment, Comment
bob@example.com, "Issue demonstrating the CSV attachment import", "Please
check the attached image below.", "https://jira-
```

```
server:8080/secure/attachment/image-name.png", "01/01/2012 10:10;Admin;
This comment works" bob@example.com, "CSV attachment import with
timestamp,author and filename", "Please check the attached image below.",
"01/01/2012 13:10;Admin;image.png;file://image-name.png", "01/01/2012
10:10;Admin; This comment works"
```

Note that URLs for attachments support the HTTP and HTTPS protocols and can be located anywhere that your Jira Cloud site can access.

Creating sub-tasks

You can create sub-tasks of issues by:

- Adding two columns to your CSV file with the headings "Issue Id" and "Parent Id"
- Ensuring each regular (non sub-task) issue is given a unique (sequential) number in the "Issue Id" column. Do not include any value in the "Parent Id" fields for regular issues.
- Referencing the unique "Issue Id" number of the regular issue in the "Parent Id" column. Do not include any value in the "Issue Id" fields for sub-tasks.

In this example, the fourth issue will be a sub-task of the second issue if you match the "Issue Id" and "Parent Id" fields in your CSV file to the corresponding issue and parent ID fields in Jira:

```
IssueType, Summary, FixVersion, FixVersion, FixVersion,
Component, Component, Issue ID, Parent ID, Reporter Bug, "First issue",
v1, , , Component1, , 1, , abcde-12345-fedcba Bug, "Second issue", v2, ,
, Component1, Component2, 2, , fedcb-12345-edcba Bug, "Third issue", v1,
v2, v3, Component1, , 3, , abcde-12345-fedcba Sub-task, "Fourth issue",
v1, v2, , Component2, , , 2, fedcb-12345-edcba
```

Importing issues into multiple JIRA projects

You can import issues from your CSV file into different Jira projects by:

- Adding two columns to your CSV file with the headings "Project Name" and "Project Key"
- Ensuring that every issue represented in your CSV file contains the name and key in these columns for the corresponding Jira projects

The project name and key data is the *minimum Jira project data* required for importing issues from a CSV file into specific Jira projects.

In this example, the first and second issues will be imported into the "Sample" project (with project key "SAMP") and the third issue will be imported into the "Example" project (with project key "EXAM") if you match the "Project Name" and "Project Key" fields in your CSV file to the corresponding Project name and Project key fields in Jira:

```
IssueType, Summary, Project Name, Project Key bug, "First
issue", Sample, SAMP bug, "Second issue", Sample, SAMP task, "Third
issue", Example, EXAM
```

How to handle unresolved issues

For fields mapping to Resolution, Priority, and Issue Type, you'll see a list with the available values in Jira. Create values that do not exist in Jira by clicking the green plus symbol.

You'll also see a list with available values for fields mapping to Status; however, you will not be able to create new status values.

For the Resolution, Priority, Issue Type, and Status fields, you can choose the following:

- Import as blank — this causes the Jira value to be blank for that field. If you are importing unresolved issues, you should create a field mapping for the Resolution field and change the value from "Unresolved" to "Import as blank".
- No mapping — this attempts to import the value in the CSV file as is. Note that the import will fail if a field value is mapped to a Jira field that doesn't support that value. Fields mapping to Status and Issue Type will use the default value when "Import as blank" is also selected.

Importing worklog entries

Your CSV file can contain worklog entries. For example:

```
Summary,Worklog Only time spent (one hour),3600 With a date and
an author,2012-02-10 12:30:10;wseliga;120 With an additional
comment,Testing took me 3 days;2012-02-10 12:30:10;wseliga;259200
```

Note that you need to use seconds in order to track time spent.

Importing to multi select custom fields

Your CSV file can contain multiple entries for the one Multi Select Custom Field. For example:

```
Summary,Multi Select,Multi Select,Multi Select Sample issue,Value
1,Value 2,Value 3
```

This will populate the Multi Select Custom Field with multiple values.

Importing cascading choice custom fields

You can import values to a cascading choice custom field using the following syntax:

```
Summary, My Cascading Custom Field Example Summary, Parent Value ->
Child Value
```

The '->' separator allows you to import the hierarchy.

Updating existing issues

To update existing issues, your CSV file needs to contain a column that maps to Issue Key. If an issue exists for a given key, it will be updated.

In this example, the second row will create issue TT-1, while the third row will set the number of votes to 7 and add two labels, and the fourth row will change the issue summary. In the next row, issue TT-2 will be created with two labels. The last row removes labels with the <<!clear!>> special marker:

```
issue key,summary,votes,labels,labels TT-1,Original
summary,1,label1,label2 TT-1,,7,label-1,label-2 TT-1,Changed summary,,,
TT-2,Original summary 2,1,label-1,label-2 TT-2,,<<!clear!>>,<<!clear!>>,
```

Importing your CSV file

Once your CSV file has been prepared, log into your Jira Cloud site as a Jira administrator to start the import process.

Upload your file

1. Choose ⚙ > **System**.
2. In the Import and Export section, click **External System Import** and then click **CSV**.
3. Choose your CSV source file.
4. Leave the **Use an existing configuration file** option unchecked if you do not have a configuration file or if you want to create a new one. Configuration files specify a mapping between column names in your CSV file's header row and fields in your Jira site. At the end of the CSV import process, Jira will create a configuration file that you can use for subsequent CSV imports.
5. Expand the Advanced section to change the file encoding (UTF-8 by default) and CSV delimiter. If your CSV file uses a separator character other than a comma, specify that character in the CSV Delimiter field.
6. Click **Next**.

Set up project mappings

1. Choose to import all issues into one (new or existing) Jira project or into multiple Jira projects. To complete this step, fill in the following:

    1. **Import to Jira Project** - You can choose **Select a project** if you're importing all issues into a single Jira project. Start typing the name or key of a project that already exists in Jira, or select **Create New** from the dropdown to add a new project. Note that new projects require a name, key, and lead.

        If you want to import issues into multiple Jira projects, choose **Defined in CSV**. Make sure that every issue in your CSV file includes project name and key data.

    2. **Email Suffix for New Users** - Enter the email address domain for any new users specified in the CSV file who will be added to Jira during the import.

        Date format in import file - Specify the date format used in your CSV file using syntax that complies with the Java SimpleDateFormat.

2. Click **Next**.

▲

Check if the selected project that you're importing the issues into is compatible with the issue key. Otherwise, Jira will import the issues to the selected project and give them new issue keys.

Set up field mappings

1. In the Import column, check each CSV Field (determined by your CSV file's header row) that you want to import into your chosen Jira project. Note that at least one of the selected fields must contain data for the Summary field. If your CSV file contains more than one of the same field name in its header row, the CSV import process will combine these into a single field marked with ▲ .

2. In the Jira field column, select the Jira fields you want to match to fields defined in your CSV file (i.e. each CSV field you selected in the previous step). You must select the Summary field in order to proceed to the next step. For CSV fields that have been aggregated during the import process, you will only be able to select Jira fields that support multiple values.

3. To modify the values of any field in the CSV file *before* importing into Jira, select the **Map field value** checkboxes next to the appropriate fields.

4. Click **Next**.

▲

If your CSV file consists of Jira Service Management projects with comments and is mapped to the **Comments body** in the **Jira fields** column, all the comments from your import file will become public after the CSV import.

ⓘ

For more information about matching CSV fields to Jira fields, see Tips for importing CSV data into Jira fields below.

Set up field value mappings

ⓘ

We don't support mapping company-managed issue types to team-managed issue types. To avoid incomplete or incorrect mappings, we've removed the ability to create a new issue type or map the issue type as is when you import data to a team-managed project. You'll have to map the issue type to an already existing issue type.

1. Specify the Jira field values for each CSV file field value you selected in the previous step. Leave a field blank if you want to import the value as is.
2. To create new Priority or Resolution values in Jira, click **Add new...** next to the appropriate field.
3. Click **Begin Import** when you are ready to begin importing your CSV data into Jira.
4. During the import, you can download a detailed log to reveal more information about the process. You can also save the configuration of this import if you need to import another CSV file with similar settings.

ⓘ

If you have any questions or encounter any problems, please contact Atlassian support.

Tips for importing CSV data into Jira fields

Below are some helpful tips when importing data from your CSV file into specific Jira fields:

Jira Field	Import Notes
Project	CSV data is imported on a per-project basis. You can specify an existing Jira project or let the importer automatically create a new project for you at time of import.
Summary	This is the only required field.
Issue Key	You can set the issue key for an imported issue. If an issue with a given key already exists in Jira, it will be updated instead.

Component	You can import issues with multiple components by entering each component in a separate column.
Affects Version	You can import issues with multiple "Affects Version" values by entering each version in a separate column.
Fix Version	You can import issues with multiple "Fix Version" values by entering each version in a separate column.
Comment Body	You can import issues with multiple comments by entering each comment in a separate column.
Date Created	Please use the date format specified on the second step of the CSV import wizard.
Date Modified	Please use the date format specified on the second step of the CSV import wizard.
Due Date	Please use the date format specified on the second step of the CSV import wizard.
Issue Type	If not specified in your CSV file, imported issues will be given the default (i.e. first) Issue Type as specified in your Jira site.
Issue rank	Issue rank will not be retained, so any manual ranking of Jira Software boards will be lost.
Labels	You can import issues with multiple labels by entering each label in a separate column.
Priority	If not specified in your CSV file, imported issues will be given the default (i.e. first) Priority as specified in your Jira site. You can also create new Jira values on-the-fly during the import process.
Resolution	If not specified in your CSV file, imported issues will be given the default (i.e. first) Resolution as specified in your Jira site. You can also create new Jira values on-the-fly during the import process. See How to handle unresolved issues for helpful tips.
Status	Can only be mapped to existing workflow statuses in Jira. If not specified in your CSV file, imported issues will be given the default (i.e. first) Status as specified in your Jira system.

Original Estimate	The value of this field needs to be specified as number of seconds.
Remaining Estimate	The value of this field needs to be specified as number of seconds.
Time Spent	The value of this field needs to be specified as number of seconds.
Users	**Supported user types** • email address • Atlassian Account ID **Creating users** You can choose to have the importer automatically create Jira users for any values of the Assignee or Reporter field. • Users will be created as active accounts in Jira. Users will need to get their passwords emailed to them the first time they log into Jira. • Users with no real name will get the portion of their email address (login name) before the "@" character as their Full Name in Jira. • If you are using External User Management, the import process will not be able to create Jira users; instead, the importer will give you a list of any new users that need to be created. You will need to create the users in your external user repository before commencing the import. • If you have a user-limited license (e.g. personal license), and the number of required users is larger than the limit, then the import will be stopped. A page will be displayed showing a list of users that can't be created. • If Assignee and Reporter are not mapped, then no users are created. **Updating users** Portal-only customer data in Jira Service Management behaves differently than other user accounts data. If the portal-only customer data is different in the import and the destination, then we use the destination data.
Watchers	If you have users specified as Watchers in your CSV file, and these users do not exist in Jira, they will not be imported. A user must be available in Jira before you can import them as a watcher on a specific issue.

Other fields	If you wish to import any other fields, you can choose to map them to specific Jira custom field. If your custom fields don't yet exist in Jira, the importer can automatically create them for you. If your custom field is a date field, please use the date format specified on the second step of the CSV import wizard.