One of our customers (a huge European logistics company’s Hungarian headquarter) has offices throughout Europe. However they wanted to federate their multiple JIRA instances into a single one. As part of this huge project, they requested us to migrate a certain projects of a live JIRA instance used in France into their main live instance hosted on their infrastructure in Hungary. JIRA’s Project Restore capabilities seemed to offer a promising route to achieve our goals.

To be more precise on the situation we had to deal with, it is important to mention that the source JIRA system was a JIRA Cloud instance. First of all, we had to migrate from JIRA Cloud to an intermediate JIRA Server instance. This process will be covered by the next article in this series. Here we cover the migration of projects from JIRA Server to JIRA Server.

### Alternatives to Migration

As the target JIRA system was a running live instance with many projects, it wasn't possible to do a full backup and restore operation. This could have been the easiest solution, but not for us here.

We had to rely on JIRA’s Project Restore feature. If one reads through the docs, it will be clear how cumbersome this process is if you need to migrate many projects, using lots of different schemes, workflows, custom fields etc. The doc prescribes to manually create or configure almost all aspects of a JIRA project in the target JIRA system.

Having many projects and metadata all mingled with add-ons hidden in workflow transitions and custom fields, this looked to be several day of manual JIRA configuration. We have desperately searched for an alternative solution and ended up with adding this add-on to our mission: Project Configurator. This add-on has proved to be a life saver for us, see below why.

### The Migration

After a few attempts, we have followed the below steps successfully. We presume the source and target systems are of the same JIRA versions. If this is not the case, we recommend you upgrade the source system to the version of the target one before the migration.

If the target system is of an older version than the source system, we recommend you upgrade the target system to the same version of the source.

### Step 1 - Disable Emailing in the Target System

Before doing anything, we disabled sending and receiving emails in the target system.

You can disable all emailing by setting the below startup options in setenv.sh or setenv.bat:

- `-Datlassian.mail.senddisabled=true`
- `-Datlassian.mail.fetchdisabled=true`
- `-Datlassian.mail.popdisabled=true`
Step 2 - Add-ons

There were add-ons in the source JIRA system that were not available in the target, so we installed them. It is important to have the add-ons installed before the migration as migrated components (workflows elements, fields etc) may come from add-ons. So it is crucial that these components may be created correctly during the migration phase.

Note, the add-ons configuration must be manually duplicated from the source system. Add-on persistent data is not restored during the data load when you restore only projects.

Step 3 - Migrate Metadata using Project Configurator

We were very happy to have Project Configurator add-on available as it proved to save lots of time and efforts.

Resolve Name Collisions
Rename all schemes, dashboards, filters, screens, workflows, custom fields, etc if a corresponding entity exist in the target system with the same name but different definition.

E.g. if there is a Screen Scheme with the same name but different screenmapping, rename the scheme in the source system to prevent name collisions. Name collisions is not a problem for Project Configurator, it will simply overwrite the schemes in the target system. And this is certainly something you will want to avoid.

Export the metadata you want to migrate

Configure the export through Project Configurator’s admin screen:

The export will create an XML file you will need in the next step.

Load the metadata in the target system

Before loading the metadata in the target JIRA instance, we strongly recommend you read at least this page in Project Configurator’s documentation: https://awnaba.atlassian.net/wiki/x/K1AF

Project Configurator allows you to load the XML file in a transaction that is rolled back. This is a simulation of the data load to reveal potential problems without corrupting the database.
if the simulation completes with errors, repeat the data load with "Apply changes?" turned on.

### Important notes

**Project Components and Versions:**

- By default Versions and Components are loaded by Project Configurator. We recommend you leave them out when you import the metadata. It is because when you import project data (see below), JIRA will stop saying, it cannot import project data into projects with existing versions and components. This means, you will have to delete all components and versions from the projects before the import can progress further.

Project configurator will migrate users, but there are a few things to know:

- user passwords are not migrated, users may have to use "Forgotten password" link upon first login to generate a new password
- users are created in the first writeable user directory be it Active Directory / LDAP or JIRA's internal directory. Setting the order of user directories in the target system before loading metadata is important

### Step 4 - Migrate Workflows
Even though recent versions of Project Configurator supports moving workflows, we used JIRA’s Workflow bundle export and restore feature to move the workflows themselves. Workflow bundles are rich in terms of containing all elements of the workflow, including definition of the screens, custom fields used by the transitions, and transition elements (e.g. post functions) added from add-ons. This is why it is important that all add-ons are available in the target system.

See details in the documentation.

It is important to know that both the export and the import are very sophisticated. Warnings, explanations and settings are available. Importing the bundle you may opt for creating the missing statuses.
Preview of import

**Statuses**
The following statuses will be created in your instance of JIRA.

- **TO DO - 2**

---

Workflow Imported

**Workflow has been successfully imported**

Additional Configuration

When this workflow was exported, additional configuration notes were provided by the author. These notes may provide information on how you can configure the imported workflow to its original state.

Please copy these notes for your records.

The items from the following plugins were removed from the workflow:

**Transition:** In Review

- System post functions removed from this transition:
  - Update Issue Field
**Step 5 - Restore Projects**

Restoring projects requires you to create a full export (XML backup) in the source system and use "Project Import" in JIRA Administration / System to start the restore process.

Follow the steps detailed in the [documentation](#).

Project import gives you detailed summary of the import result, and the potential errors allowing you to repeat the import after fixing the problems.
**Important notes**

We experienced that during the project restore process, the Create workflow transition in the issues' workflows is executed. This may have unexpected consequences.

If the Create transition contains post functions that initialize fields (e.g. assign issues, update custom fields etc), the issues that are imported will be reassigned to someone else, or fields will be overwritten.

E.g. if the Create transition has a post function from MISC Workflow Extensions: to assign the issue to the first member of a role, the issue will be reassigned unexpectedly.

Due this fact, you need to remove these post functions from the workflows before importing the project data. The easiest is to copy the workflows, modify the original ones and restore them in the workflow schemes from the copy after the data migration.

---

**Step 6 - Add Mail Handlers**

if the source system had mail handlers, you must add them manually to the target system in Administration / System / Incoming Mail

**Step 7 - Enable Emailing and Restart**

Finally, edit setenv.sh/bat to enable emailing again and restart and reindex JIRA.

**Backups and Restore Points Strategy**

We recommend you do a system backup before each and every step listed above. If a step fails and you will have a backup version to restart from the previous step, it is a real time saver and relief. If JIRA and its database are running in a virtual server environment, creating snapshots of the systems are the best options. These operations are fast and reliable.

**Conclusion**

Due to the complexity and many aspects of Projects in JIRA, moving them from one JIRA instance to another is a cumbersome process that includes many manual configuration too, despite Project Configurator that helps a lot but cannot make everything smooth. However, if done with care and attention, the above steps will most likely let you complete project migrations in a more plannable timeframe.