Jira Test Management migrator (JTMM)

Table of Contents

- Jira Test Management migrator (JTMM)
  - Product overview
  - IMPORTANT
  - Configuration
    - Map-Fields
    - User-Config-File
    - Global-Variables
  - Permission-Requirements
    - Jira-Required-Permissions
    - Azure-DevOps-required-permissions
    - Jira-Authentication
  - Features
    - Jira-Key-Tracking
  - How-to-run-JTMM
    - CLI-parameters
    - Direct-Migration
    - Automated-Migration
  - Individual-documentation
    - QMetry
    - Xray
    - Zephyr-Scale-|-Zephyr-Squad

Product overview

The **Jira Test Management migrator (JTMM)** developed by **Solidify** is a powerful and robust tool designed to help you easily migrate your Jira test management data to Azure DevOps. Detailed features specific to each test framework are outlined in their respective step guides.

Our tool supports the following Jira test frameworks:

- QMetry (Cloud)
- Zephyr Scale (Cloud)
- Zephyr Squad (Server)
- Xray (Cloud, Server)
- More to come soon!

Jira Server vs Jira Cloud

The only test frameworks which supports Jira Server is Zephyr Squad and Xray. The rest frameworks only supports Jira Cloud.

ADO Cloud vs ADO Server

The tool only supports Azure DevOps Cloud. We are working on adding support for Azure DevOps Server.

IMPORTANT

Remember, use the **Jira to Azure DevOps Migrator** to migrate your Jira issues first, before using this migrator. Otherwise, the issue links may not be recreated properly, and the code will throw errors. If you don't care about links, then ignore the messages.

Configuration file

In order to run the program, a configuration file needs to be provided. The file must be in JSON format. When running the CLI, you must specify the path to the config file using the `-c` or `--config` parameter.

The configuration file contain some required values for example your environment values and some optional values for example if you need to map some fields. If you don't specify the mapping field values the migrator will fall back to a default configuration of those fields.

The supported fields are:

Variable	Description	Required
`azureOrganization`	Azure DevOps Organization	*
`jiraAuthMethod`	Jira Authentication Method (`basic`/`oauth2`)	*
`jiraAccountEmail`	Jira Account Email or Username for authentication (if JIRA_AUTH_METHOD = `basic`)	*
`jiraDomain`	Jira Domain	*
`jiraProtocol`	Jira Protocol (`https`/`http`)	*
`jiraApiVersion`	Jira API Version (`1`/`2`/`3`)	*
`jiraOriginWiField`	The name of your custom Jira key field	
`Mapper`	Mapping the following field values: **state**, **priority**, **IterationPath**, **Link_Types**	

Example from Zephyr Scale:

```
{
    "azureOrganization": "my-azure-org",
    "jiraAuthMethod": "Basic",
    "jiraAccountEmail": "my.email@gmail.com",
    "jiraDomain": "Your-Jira-Domain.atlassian.net",
    "jiraProtocol": "https",
    "jiraApiVersion": "2",
    "jiraOriginWiField": "Custom.YourCustomField",
    "Mapper": {
        "Priority": {
            "High" : 1,
            "Normal" : 2,
            "Low" : 3
        },
        "State": {
            "test_plan": {
                "Draft": "Inactive",
                "Deprecated": "Inactive",
                "Approved": "Active",
            },
            "test_suite": {
                "Not Executed": "In Planning",
                "In Progress": "In Progress",
                "Done": "Completed"
            },
            "test_case": {
                "Draft": "Design",
                "Deprecated": "Closed",
                "Approved": "Ready",
            }
        },
        "IterationPath":{
            "Sprint-1": "YourTargetProject\\YourMappedPath-1",
            "Sprint-2": "YourTargetProject\\YourMappedPath-2"
        },
    }
}
```

User Config File

A user config file is required to do custom user mappings. The user mapping should be in the form of `email@jira.com=email@ado.com`, the first email being the user email from Jira, and the second one being their Azure DevOps user email. If the `user_config.txt` file isn't provided, the tool will use the same email in Azure DevOps.

For user mapping to work, all users in the Jira organization must make sure that their email is publicly visible to everyone. To do this, go to https://id.atlassian.com/manage-profile/profile-and-visibility (https://id.atlassian.com/manage-profile/profile-and-visibility) -> Contact -> Email Address -> Who can see this? -> "Anyone"

user_config.txt example

```
email1@jira.com=email1@ado.com
email2@jira.com=email2@ado.com
email3@jira.com=email3@ado.com
```

Credentials values set in a .env file

This tool requires an `.env` where you can set your secret values. The file needs to be provided in the root directory. The value in the file depends on which test adapter you are using. Below is a table of each required values for each test framework.

Required for all platforms

Variable	Description
`AZURE_API_TOKEN`	Azure DevOps API Token
`JIRA_API_TOKEN`	Jira API Token or Password (if JIRA_AUTH_METHOD = `basic`) or Jira OAuth2 Token (if JIRA_AUTH_METHOD = `oauth2`)

Required for Zephyr

Variable	Description
`ZEPHYR_API_TOKEN`	Zephyr API Token (platform = `zephyr-squad` or `zephyr-scale`)

Required for QMetry

Variable	Description
`QMETRY_API_TOKEN`	QMetry API Token

Required for Xray Cloud

Variable	Description
`XRAY_CLIENT_ID`	Xray Client ID (platform = `xray`)
`XRAY_CLIENT_SECRET`	Xray Client Secret (platform = `xray`)

Example of .env file

```
AZURE_API_TOKEN = 'XXXX'
JIRA_API_TOKEN = 'XXXX'
QMETRY_API_TOKEN = 'XXXX'
XRAY_CLIENT_ID = 'xxxx'
XRAY_CLIENT_SECRET = 'xxxx'
ZEPHYR_API_TOKEN = 'XXXX'
```

All data in the above example are fictitious.

Permission requirements

Jira required permissions

The Account who will run the migration will require at least **Administrator** permission in each relevant Jira project.

Specifics for Zephyr

For **Zephyr Squad**, the account will need Administrator permission in Zephyr.

Azure DevOps required permissions

Required license

The Account who will run the migration will require any of the following licenses:

- Basic + Test Plans
- Visual Studio Professional
- Visual Studio Enterprise

Required project permissions

The Account who will run the migration will require the following permissions under **Project Settings** -> **Permissions**:

- Create Test Runs

- Delete Test Runs
- Manage Test Plans

Required area path permissions

Additionally, you must configure the permissions of the **area path** where the Test Plan is located. Under **Project Settings** -> **Project configuration** -> **Areas** -> **[Select the area path of the Test Plan]** -> **Security**: You must add the Account and **Allow** the following permissions:

- Manage test plans
- Manage test suites

If the area permissions are configured incorrectly, you will see the following error message in the pipeline log:

```
Task failed, error message: Error: You do not have the appropriate permissions to manage test suites under
this area path.: xxx/xxx/xxx
```

Jira Authentication

We support various methods of authenticating with the Jira Rest API:

Basic authentication

Authentication with Jira username and password/Jira API token. You will need to specify the following variables in the `.env` file:

```
JIRA_AUTH_METHOD = Basic
JIRA_API_TOKEN= *JIRA PASSWORD/API TOKEN*
```

For troubleshooting common authentication issues with **Basic authentication**, check out our guide on github: https://github.com/solidify/jira-azuredevops-migrator/blob/master/docs/faq.md#2-why-i-am-getting-unauthorized-exception-when-running-the-export (https://github.com/solidify/jira-azuredevops-migrator/blob/master/docs/faq.md#2-why-i-am-getting-unauthorized-exception-when-running-the-export)

OAuth2-based authentication

Authentication with an OAuth token. You will need to specify the following variables in the `.env` file:

```
JIRA_AUTH_METHOD = Oauth2
JIRA_OAUTH_TOKEN= YOUR_JIRA_OAUTH2_TOKEN
```

Authentication for Jira Server

When you are using Jira Server, you need to add different credentials to authenticate. The credentials in your `.env` file needs to looks like this:

```
JIRA_API_TOKEN = "your-jira-password"
JIRA_ACCOUNT_EMAIL = "your-jira-username"
```

Features

Jira Key Tracking

We've introduced a functionality that allows you to monitor your migrated Work Items effectively. This feature enables you to query all Work Items migrated from Jira preventing the duplication of Work Items during the migration process.

If you want to disable the prevention of duplications you can pass in `-f, --force` as a parameter to the CLI. This will force in all Work Items even if they exists in the target project.

To implement this feature, you must incorporate custom fields into your process template within Azure DevOps. These fields should be added to the work item types: Test Case, Test Plan, and Test Suite. Optionally, these fields can be hidden to prevent users from altering the values.

When a test has been migrated the the tool will automatically append the Jira issue key to the designated custom field. `This feature is only available for Zephyr Scale and Xray at this moment.`

Enable it by passing in `-j, --jira-key-tracking` as a parameter to the CLI. FYI this feature is only available if you add the `JIRA_ORIGIN_WI_FIELD` in your configuration file.

```
"jiraOriginWiField": "Custom.YourCustomField"
```

```
./jira-test-mgmnt-migrator.exe --platform <zephyr-scale/zephyr-squad/xray/xray-server> --config
<path_to_config/config.json> --source source_project --target target_project --jira-key-tracking
```

How to run JTMM

CLI parameters

The parameters currently supported by the CLI are:

Flag	Description	Required
`-p, --platform`	The platform that you wish to migrate from: `qmetry`, `zephyr-scale`, `zephyr-squad`, `xray` or `xray-server`	*
`-s, --source`	Jira source project id (e.g. 10001)	*
`-t, --target`	ADO target project name	*
`-c, --config`	Path to the configuration file	*
`--license-file-path`	Provide a software license by specifying the path to the license file	*
`-l, --list`	List all Jira or Azure DevOps projects (`ado/jira`)	
`-a, --auto`	Automate multiple migrations; provide an automation file (format: *jira_project_id=azure_devops_project_name*)	
`-i, --ignore-errors`	Ignore errors and keep running even when errors occur	
`-j, --jira-key-tracking`	Preserved the connection to the Jira Issue by adding the Jira Issue ID to a given field in ADO	
`-f, --force`	If set, the Jira Issues will be migrated even if the corresponding Work Items already exist in the target project	
`-v, --log-level`	Set log level to either 'debug' or 'info' to get different amounts of information about your migration. Default = info	
`-n, --jira-key-title`	Insert the the Jira Issue Key to all Work Item Titles where applicable, on the following format: [JIRA-123] Original Title	
`-o, --save-issues`	Save a JSON object locally for every issue downloaded. This is useful for troubleshooting your work. Does not affect the import.	

Migration types

There are 2 ways to run a migration in JTMM.

Direct Migration

The first way is a simple straight forward one-time migration, simply specify the platform (zephyr-scale/zephyr-squad/qmetry/xray), the source (The Jira Project ID, for example 10001, 10002, 10003), path to your config file and the Azure DevOps target project name.

This is the command you will issue:

```
.\jira-test-mgmnt-migrator.exe --platform <zephyr-scale/zephyr-squad/qmetry/xray/xray-server> --config
<path_to_config/config.json> --source <Jira_Project_ID> --target <Azure_DevOps_project_name>
```

Automated migration

The second way is "Automation mode". There are 2 things you need to specify here. The platform (zephyr-scale/zephyr-squad/qmetry/xray) and the config file (auto_file_name.txt). The config file should be in the format of "jira_project_id=azure_devops_project_name". For example:

```
10000=my_first_ado_project
10001=my_second_ado_project
10002=my_third_ado_project
```

The command you will be using to run this migration is:

```
.\jira-test-mgmnt-migrator.exe --platform <zephyr-scale/zephyr-squad/qmetry/xray/xray-server> --config
<path_to_config/config.json> --auto <auto_file_name.txt>
```

The program should now be looping through your config file, migrating each project as it goes.

Logging

If you wish to troubleshoot your execution, you can set the parameter `--log-level` to `debug` in the CLI. This will make the CLI run in debug mode, which will print out more information about the execution.

The default log level is `info`.

Individual documentation

QMetry

Supported migrated data for QMetry

Fields	Test Case	Test Cycle	Test Plan	ADO Value
Description	N	Y	N	Description
Precondition	N	N/A	N/A	Description
Folder Path	Y	Y	Y	AreaPath
Priority	Y	Y	Y	Priority
Status	Y	Y	Y	State
Assignee	N	N	Y	Assignee
Reporter	N	N	Y	N/A
Components	N	N	N/A	N/A
Labels	Y	N	Y	Tags
Sprint	N	N	Y	IterationPath
Fix Versions	N	N	N	N/A
Estimated Time	N	N/A	Y	N/A
Created By	N	N	N	N/A
Created On	N	N	N	N/A
Updated By	N	N	N	N/A
Updated On	N	N	N	N/A
Test Steps	Y	N/A	N/A	Test Steps
Attachments	N	N	N	Attachments
Executions	N	N/A	N/A	N/A
Story	N	N/A	N/A	N/A
Comments	N	N	N	Comments
Audit Logs	N	N	N	History
Test Case Links	N/A	Y	N/A	Parent Link
Test Cycle Links	N/A	N/A	Y	Parent Link
Planned Start Date	N/A	N	Y	Parent Link
Planned End Date	N/A	N	Y	Parent Link

Xray

Supported migrated data for Xray

The Xray migration adapter only supports the following test types:

- Test Plan
- Test

Fields	Test	Test Plan	ADO Value
Description	Y	Y	Description
Priority	Y	Y	Priority
Status	Y	Y	State
Sprint	Y	Y	Iteration Path
Reporter	Y	Y	Assignee
Components	N	N	Components
Labels	Y	Y	Tags
Created On	Y	N	Created Date
Updated On	Y	N	Updated Date
Test Steps	Y	N/A	Test Steps
Attachments	Y	Y	Attachments
Comments	Y	N	Comments
Issue Links	Y	Y	Relation Links
Parent / child link	N	N	Parent / Child Link
Planned Start Date	N/A	N	End Date
Planned End Date	N/A	N	Start Date
Key	Y	Y	Your_Custom_Field
Revisions	N	N	Revisions

Generate Xray secret and client id

Please follow Xray's official documentation (https://docs.getxray.app/display/XRAYCLOUD/Global+Settings%3A+API+Keys) on how to generate a client id and secret.

Zephyr Scale | Zephyr Squad

Supported migrated data for Zephyr Scale

Fields	Test Case	Test Cycle	Test Plan	ADO Value
Description	Y	Y	N	Description
Precondition	Y	Y	N	Description
Objective	Y	N/A	Y	Description
Folder	Y	Y	Y	AreaPath
Priority	Y	N	N	Priority
Status	Y	Y	Y	State
Labels	Y	Y	Y	Tags
Sprint	N	N	N	IterationPath
Estimated Time	N	N	N	N/A
Owner	Y	Y	Y	Assignee
Test Steps	Y	N/A	N/A	Test Steps
Attachments	N	N	N	Attachments
Comments	N	N	N	Comments
Audit Logs	N	N	N	History
Issues	Y	Y	Y	Related Link
Web Links	Y	N	Y	Description
Test Cases	N/A	Y	N/A	Child Link
Test Plans	N/A	Y	N/A	Parent Link
Test Cycles	N/A	N/A	Y	Child Link

Supported migrated data for Zephyr Squad

Fields	Test Case	Test Cycle	ADO Value
Description	Y	Y	Description
Precondition	Y	Y	Description
Objective	Y	N/A	Description
Folder	Y	Y	AreaPath
Priority	Y	N	Priority
Status	Y	Y	State
Labels	Y	Y	Tags
Sprint	N	N	IterationPath
Estimated Time	N	N	N/A
Owner	Y	Y	Assignee
Test Steps	Y	N/A	Test Steps
Attachments	Y	N	Attachments
Comments	Y	N	Comments
Audit Logs	N	N	History
Issues	Y	Y	Related Link
Web Links	Y	N	Description
Test Cases	N/A	Y	Child Link
Test Cycles	N/A	N/A	Child Link