A black and white logo

Description automatically generated with low confidence

|  |
| --- |
| **Migration from one organization to another organization**  **(Using Azure DevOps Migration Tool)** |

**Migration from one organization to another organisation**

# Table of Contents

[Table of Contents 2](#_Toc133395540)

[1. Overview (Tool) 3](#_Toc133395541)

[2. PRE-REquisites 4](#_Toc133395542)

[3. Server configuration and setup 4](#_Toc133395543)

[4. Processors: 7](#_Toc133395544)

# Overview (Tool)

Azure DevOps Migration Tools allow you to migrate Teams, Backlogs, Tasks, Test Cases, and Plans & Suits from one Project to another in Azure DevOps / TFS both within the same Organisation, and between Organisations.

Azure DevOps Migration Tools allow you to bulk edit and migrate data.

**We can do with this tool:**

* Migrate Work Items, TestPlans & Suits, Teams, Shared Queries, & Pipelines from one Team Project to another**.**
* Migrate Work Items, TestPlans & Suits, Teams, Shared Queries, & Pipelines from one Organisation to another.
* Bulk edit of Work Items across an entire Project.

**Versions of Azure DevOps & TFS support:**

* Work Item Migration Supports all versions of TFS 2013+ and all versions of Azure DevOps (Many tools also work all the way back to TFS 2010 but are not supported).
* You can move from any Tfs/AzureDevOps source to any Tfs/AzureDevOps target.
* Process Template migration only supports XML based Projects.

**Uses of this tool:**

* Merge many projects into a single project
* Split one project into many projects
* Assistance in changing Process Templates
* Bulk edit of Work Items
* Migration of Test Suites & Test Plans
* new Migration of Builds & Pipelines

# PRE-REquisites

**Below are the pre-requisites needs to be ready:**

* Windows 7+ / Windows Server 2003+
* PowerShell v2+ (minimum is v3 for install from this website due to TLS 1.2 requirement)
* .NET Framework 4+ (the installation will attempt to install .NET 4.0 if you do not have it installed)(minimum is 4.5 for install from this website due to TLS 1.2 requirement)
* Chocolatey
* vsts-sync-migrator

**Install:**

* Install Chocolatey from [Chocolatey Software | Installing Chocolatey](https://chocolatey.org/install)
* Run choco install vsts-sync-migrator to install the tools [Chocolatey Software | Azure DevOps Migration Tools 12.8.9.0](https://community.chocolatey.org/packages/vsts-sync-migrator)

The tools are now installed. To run them you will need to switch to c:\tools\MigrationTools\ and run migration.exe.

**Upgrade :**

Run choco upgrade vsts-sync-migrator to upgrade the tools source

**Default configuration file:**

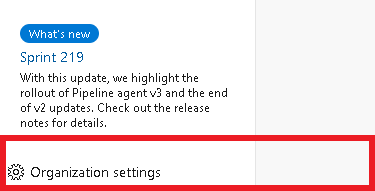
* Open a command prompt or PowerShell window at C:\tools\MigrationTools\
* Run ./migration.exe init to create a default configuration
* Open configuration.json from the current directory

You can now customise the configuration depending on what you need to do. However a basic config that you can use to migrate from one team project to another with the same process template.

# Server configuration and setup

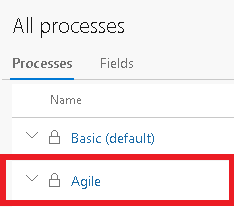
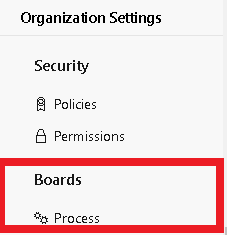
Follow the setup instructions to make sure that you can run the tool against your environments and importantly add the required custom field **‘ReflectedWorkItemId’**.

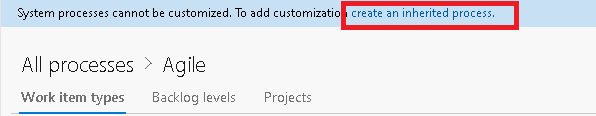
1. Open Azure DevOps Services
2. Click on organization settings



1. We need to create a custom process (inherited proces).

Under Boards – Process >> Agile >> create an inherited process.

****

****

Give a process name and create an inherited process.

Move project from agile process to custom process.

1. Create a new field (Custom.ReflectedWorkItemId) under work items.

Definition (Name = ReflectedWorkItemId)

Layout(Lable = Custom.ReflectedWorkItemId)

**Follow same steps also in Target Project**

1. Configure the default configuration file (C:\tools\MigrationTools\ configuration.json)

Add

* Collection
* Project
* ReflectedWorkItemIDFieldName
* PersonalToken



# Processors:

* **WorkItemMigrationConfig.**

WorkItemMigrationConfig is the main processor used to Migrate Work Items, Links, and Attachments. Use WorkItemMigrationConfig to configure.



* **"WIQLQueryBit": "AND [System.WorkItemType] NOT IN ('Test Suite', 'Test Plan','Shared Steps','Shared Parameter')"**

Above query will migrate all the workitem present in source except test suite,test plan,shared query,shared parameter.

* **"WIQLQueryBit": "AND [System.AreaPath] UNDER 'ProjectName\\TeamName' AND [System.WorkItemType] NOT IN ('Test Suite', 'Test Plan')"**

**"NodeBasePaths": [**

**"ProjectName\\DesignerTeam",**

**"ProjectName\\March23",**

**"ProjectName\\April23",**

**]**

Above query will migrate all the workitem present in Source project for specifically Team except test suite, test cases, shared parameters, shared steps.

In **“nodebasepath”** along with teams we will also need to mention the iteration and spirint in which team is working.

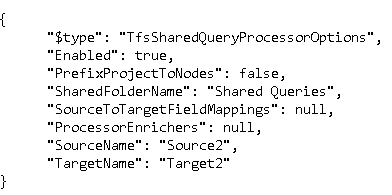
* **"WIQLQueryBit": "AND [System.WorkItemType] IN ('Test Suite', 'Test Plan', 'Shared Steps','Shared Parameter')"**

Above query will migrate all the workitem(Test suite,Test cases,shared parameters,shared steps) present in source project.

* **TfsSharedQueryProcessor.**

The TfsSharedQueryProcessor enabled you to migrate queries from one location to another.

In configuration file **“processors”** need to add TfsSharedQueryProcessor



Also need to add **“endpoint”** for TfsSharedQueryProcessor



* **AzureDevOpsPipelineProcessor**.

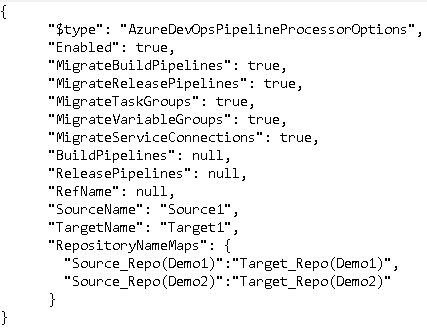
Azure DevOps Processor that migrates Taskgroups, Build- and Release Pipelines.

Migrates variable groups

Migrates task groups

Migrates classic and yml build pipelines

Migrates release pipelines



Also need to add **“endpoint”** for TfsSharedQueryProcessor

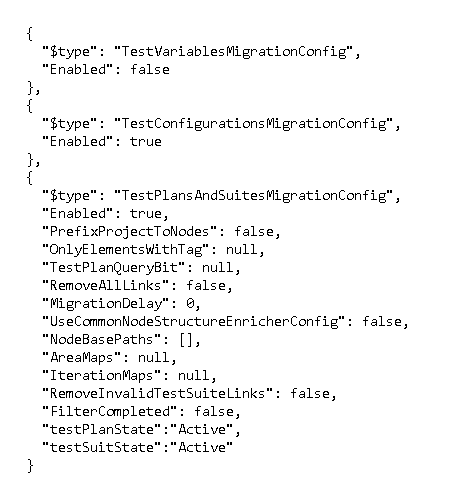


For pipeline migration we need to add manual repo name in target project as same as source project.

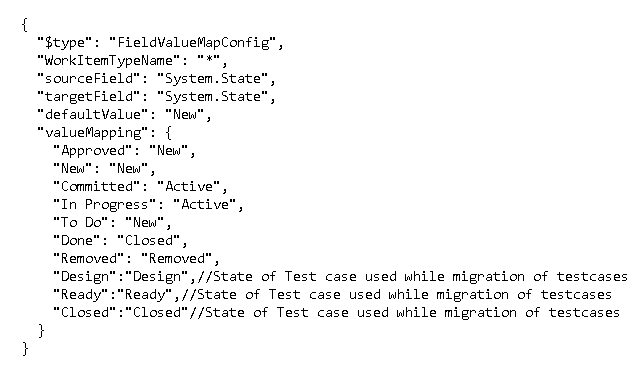
Also, the pipelines which contain marketplace task (sonarqube,sonarcloud) does not migrate.

* **TestPlansAndSuitesMigrationContext.**

To run a full plans and suits you should run the three processors in this order below. TestVariablesMigrationConfig and TestConfigurationsMigrationConfig only need run once.



Also we need to add FieldValueMapConfig



While executing the test plans, test cases, test suites “Migrator throws an exception state unavailable exception( Active, Closed, New, Ready, Design).

To overcome with error or exception we need to explicitly add that missing state in process in custom process in test cases.

Organisation >> Process >> custom process >> test cases workitem >> state >> add missing states (Active - Inprogress, Ready- Inprogress, Closed-Closed, Design-Inprogress)

**Step 5.**

Now you are able to see deployed application on specific port.