

Migration from Azure DevOps repositories to GitHub

The description of how the migration is done.

Prerequisites/dependencies

- config.json
- usermapping.txt (azure=github) or let the tool create a mapping file
- GitHub organization
- "Enterprise Cloud Importer" feature enabled in that organization
- GitHub user must be an owner in that organization
- AzureDevOps PAT token

Know issues/limitations

- Pull requests inline comments are not exported
- Pull requests attachments are not exported
- Repository security policies are not exported
- Pull requests have latest commits exported if PR has been squashed and merged
- Exported Repository cannot have an existing Repository in GitHub with the same name
- Package to import cannot be larger than 20GB

Pre-migration

- Identify repositories to migrate
- Verify that GitHub organization exists
- Verify that users have pre-created in GitHub
- Verify if repositories has large files
- Verify if repositories need a prefix

Dry-run

- Validate source and target prerequisites
- Create migration configuration
- Migrate repositories
- Check the logs for errors and warnings
- Validated the migrated Repositories contain the same information

Dry-run validation

-
- Check and validated that selected repositories are migrated with expected data

Migration

- Validate source and target prerequisites
- Create migration configuration
- Migrate repositories
- Check the logs for errors and warnings
- Validated the migrated Repositories contain the same information

Migration validation

- Check and validated that selected repositories are migrated with expected data

Work Instructions

Tool installation

- Get the tool from the Github repository

Setup configuration

- Create a config.json file matching the structure from the below example.
 - when migrating from old Tfs URLs then eg. `myorg.visualstudio.com` then keep in mind that the source should have **DefaultCollection** in the end: **"source":**
"myorg.visualstudio.com/DefaultCollection"
- Create a usermapping.txt file if needed.
- Generate an Azure DevOps PAT for the user running the command

Export repositories

- Start a console/terminal and use the following command: `synchub.exe run -n "ArToGithubExporter" -v "v1" --config config.json -u AzureDevOpsUser -p AzurePAT`
- Once the tool finishes open File Explorer and go to the target path that was configured in config.json

Import repositories

- Go to GitHub and to your organization, make sure you have Enterprise Cloud Importer feature enabled from GitHub.
- In your browser go to <https://eci.github.com/>
- Follow the instruction from the import tool in GitHub

Examples

Example config.json

```
{
  "source": "https://dev.azure.com/organization",
  "target": "Path to export the files",
  "projects": [ { "name": "AZURE_PROJECT_NAME",
  "repositories": [ { "name": "REPOSITORY_NAME" } ] } ],
  "SchemaVersion": "1.2.0", //Required - do not change
  "UserMapping": "PATH_TO_USERMAPPING_FILE",
  "GitHubUserNameExtension": "GH username extension if needed",
  "RepositoryNamePrefix": "ssw-ps-fusion-"
}
```

Example usermapping.txt content

azureemail=githubusername

Example command

```
synchub.exe run -n "ArToGithubExporter" -v "v1" --config config.json -u AzureDevOpsUser  
-p AzurePAT
```