layout: page
title: "Pipelines and Task Groups"
author: alexander-hjelm_solidify
last_updated: 2022-07-01
search_exclude: false

This document shows how to migrate pipelines and task groups using SyncHub.

Prerequisites/dependencies

- Config.json
- Azure DevOps PAT
- Pipeline Export tool

The following resources used in the definition need to be mapped:

- Service connections

- Deployment groups

- Agent pools

- Git repos

- Variable groups

- Order of migration

    1. Task Groups
    2. Build Definitions
    3. Release Definitions

Permissions

- **required PAT scopes**:

    - Build (Read & execute)
    - Release (Read, write, execute & manage)

Know issues/limitations

- Definitions and TaskGroups need to have unique names
- Migrated pipelines need to have a valid state

Notes on pipeline migration

In order for the Task Group mapping to persist, you must include your Task Groups in the pipeline import run. Removing them from the folder will break the mapping and cause the pipeline import to fail.

Pre-Migration

- Verify that has valid PAT
- Verify that Service Connections mappings are created
- Verify that App Pool and Deployment Group mappings is created
- Verify that variable groups are migrated and mapping exists
- You need to replace source url and id:s with the correct data from the target

Pre-migration, import job, configuration

- set correct target projectID
- set correct target organization
- set correct target queue name
- set correct target agent pool identifier if it a hosted build queue
- set correct imported git repo

Dry-run

- Validate that Connected Services, Deployment groups, Variable Groups are migrated and mapped

- Validate source and target prerequisites
- Create an export configuration
- Decide which Pipelines to migrate in the dry-run(if not all)
- Migrate Task Groups and Pipelines
- Check the logs for errors and warnings

Dry-run validation

- Check that the exported task groups match the selected count
- Check that the exported Pipelines match the selected count
- Check that the exported Deployment Groups match the selected count

Migration

- Validate that Connected Services, Deployment groups, Variable Groups are migrated and mapped
- Validate source and target prerequisites
- Create an export configuration
- Migrate Pipelines and Task groups
- Check the logs for errors and warnings
- Validate that selected Pipelines/task groups count matches in source and target

Migration validation

- Check that the exported task groups match the selected count
- Check that the exported Pipelines match the selected count
- Check that the exported Deployment Groups match the selected count

Work Instructions

Get the export tool

- Checkout export tool from GitHub

Setup configuration

- If the user who is running the tool does not have an Azure DevOps PAT, one needs to be generated.
- Create config.json based on the example below.
- Fill in the necessary info

Run Export Tool

- Run the tool with the example command filling in the missing information.

Run the import tool

- Run the tool with the example command filling in the information

Examples

Example export config

```
{
  "configuration":
  {
    "source": "Source Url",
    "target": "Local path where to save",
    "orgName": "Organization name",
    "projectName": "Project Name",
    "buildDefinitionFilters": [
      {"path": "\\*"}
    ],
    "releaseDefinitionFilters": [
      {"path": "\\*"}
    ],
    "onPrem": false,
    "apiVersion": "6.0-preview.1"
  }
}
```

Example export command

```
synchub.exe run -n "Pipeline-Export" -v "v1" --config config.json -u useremail -p PAT --license-file
C:\path\to\license-file.json
```

Exmaple import config

```
{
    "source": "source url",
    "target":  "target url",
    "sourceFilePath": "",
    "targetProject": "",
    "orgName": "",
    "IsRepoInGithub": true,
    "connectedServiceId": "Github Service connection id",
    "githubOrg": "",
    "repositoryNamePrefix": "Optional if prefix is used epos",
    "migrateDeploymentGroupTags": false,
    "omitDisabledTasksForReleases": false,
    "definitionPathPrefix": "leave empty if definitions saved to root folder",
    "queueIdMapping": "D:\\MigrationWorkspace\\mappings\\queueid_map.csv",
    "serviceConnectionMapping": "D:\\MigrationWorkspace\\mappings\\serviceconnection_map.csv",
    "deploymentGroupMapping": "D:\\MigrationWorkspace\\mappings\\deploymentgroup_map.csv",
    "repositoryMapping": "D:\\MigrationWorkspace\\mappings\\repository_map.csv",
    "variableGroupsMapping": "D:\\MigrationWorkspace\\mappings\\variablegroup_map.csv",
    "taskNameMapping": "D:\\MigrationWorkspace\\mappings\\taskname_map.csv",
    "taskVersionMapping": "D:\\MigrationWorkspace\\mappings\\taskversion_map.csv",
    "onPrem": false,
    "apiVersion": "6.0-preview.1"
}
```

Feature flags

- **IsRepoInGithub**: [true/false] Set to `true` if you have previously migrated the git repositories to GitHub. If `true`, the following configuration values must also be set:

  - **connectedServiceId**: The GUID of the GitHub App service connection (in Azure DevOps -> Project Settings).
  - **githubOrg**: The name of the GitHub organization where the new repositories reside.

- **repositoryNamePrefix**: Include a prefix in the reference to the repository names (optional).
- **migrateDeploymentGroupTags**: [true/false] Should we migrate Tags for deployment groups?
- **omitDisabledTasksForReleases**: [true/false] Should we omit disabled tasks inside release pipelines?
- **definitionPathPrefix**: Use this property if the source pipelines are inside a folder.

Mapping files

- **taskNameMapping**: Map task IDs for build pipelines and release pipelines. Example:

```
b10d92e9-3d97-42d1-a4a3-57e1aa3e0bf3,7a45b82d-fc56-4d98-a3a2-19e22cc486f7
af8e394b-5f07-4e7b-b8e0-df5b43f70e7d,06a30a0b-9c1a-4685-b134-3b9d1e18d99a
c5e642d1-7984-42c4-a6ca-3b5a1f67e89d,f057e50f-70ca-4d2a-89c4-df2e3ebc8a27
91a641bb-63b7-44f3-a556-2e617e646b5e,bc9b34b7-c0d7-4b82-ba24-3433e2b78c20
e57d8c23-7ff7-4d7b-9bca-49a04f89b932,fb9d980b-ea3b-4dd9-88cc-5e84a17c2b79
```

- **taskVersionMapping**: Set the version of a specific task (by task ID) for build pipelines and release pipelines. Example:

```
b10d92e9-3d97-42d1-a4a3-57e1aa3e0bf3,1.*
af8e394b-5f07-4e7b-b8e0-df5b43f70e7d,2.*
c5e642d1-7984-42c4-a6ca-3b5a1f67e89d,2.*
91a641bb-63b7-44f3-a556-2e617e646b5e,3.*
e57d8c23-7ff7-4d7b-9bca-49a04f89b932,2.0
```

Example import command

```
synchub.exe run -n "Pipeline-Import" -v "v1" --config config.json -u useremail -p PAT --license-file
C:\path\to\license-file.json
```

Build definition JSON example

```
{
```

```json
    "options": [
      {
        "enabled": true,
        "definition": {
          "id": "5d58cc01-7c75-450c-be18-a388ddb129ec"
        },
        "inputs": {
          "branchFilters": "[\"+refs/heads/*\"]",
          "additionalFields": "{}"
        }
      },
      {
        "enabled": false,
        "definition": {
          "id": "a9db38f9-9fdc-478c-b0f9-464221e58316"
        },
        "inputs": {
          "workItemType": "Bug",
          "assignToRequestor": "true",
          "additionalFields": "{}"
        }
      }
    ],
    "variables": {
      "BuildConfiguration": {
        "value": "release",
        "allowOverride": true
      },
      "BuildPlatform": {
        "value": "any cpu",
        "allowOverride": true
      },
      "system.debug": {
        "value": "false",
        "allowOverride": true
      }
    },
    "properties": {},
    "tags": [],
    "_links": {
      "self": {
        "href": "https://dev.azure.com/maneric2/6c541cf6-d884-481a-9872-
f3b5368cfde7/_apis/build/Definitions/10?revision=1"
      },
      "web": {
        "href": "https://dev.azure.com/maneric2/6c541cf6-d884-481a-9872-f3b5368cfde7/_build/definition?
definitionId=10"
      },
      "editor": {
        "href": "https://dev.azure.com/maneric2/6c541cf6-d884-481a-9872-f3b5368cfde7/_build/designer?
id=10&_a=edit-build-definition"
      },
      "badge": {
        "href": "https://dev.azure.com/maneric2/6c541cf6-d884-481a-9872-f3b5368cfde7/_apis/build/status/10"
      }
    },
    "buildNumberFormat": "$(date:yyyyMMdd)$(rev:.r)",
    "jobAuthorizationScope": "projectCollection",
    "jobTimeoutInMinutes": 60,
    "jobCancelTimeoutInMinutes": 5,
    "process": {
      "phases": [
        {
          "steps": [
            {
              "environment": {},
              "enabled": true,
              "continueOnError": false,
              "alwaysRun": false,
```

```
          "displayName": "Use NuGet 4.4.1",
          "timeoutInMinutes": 0,
          "condition": "succeeded()",
          "task": {
            "id": "2c65196a-54fd-4a02-9be8-d9d1837b7c5d",
            "versionSpec": "0.*",
            "definitionType": "task"
          },
          "inputs": {
            "versionSpec": "4.4.1",
            "checkLatest": "false"
          }
        },
        {
          "environment": {},
          "enabled": true,
          "continueOnError": false,
          "alwaysRun": false,
          "displayName": "NuGet restore",
          "timeoutInMinutes": 0,
          "condition": "succeeded()",
          "task": {
            "id": "333b11bd-d341-40d9-afcf-b32d5ce6f23b",
            "versionSpec": "2.*",
            "definitionType": "task"
          },
          "inputs": {
            "command": "restore",
            "solution": "$(Parameters.solution)",
            "selectOrConfig": "select",
            "feedRestore": "",
            "includeNuGetOrg": "true",
            "nugetConfigPath": "",
            "externalEndpoints": "",
            "noCache": "false",
            "disableParallelProcessing": "false",
            "packagesDirectory": "",
            "verbosityRestore": "Detailed",
            "searchPatternPush":
"$(Build.ArtifactStagingDirectory)/**/*.nupkg;!$(Build.ArtifactStagingDirectory)/**/*.symbols.nupkg",
            "nuGetFeedType": "internal",
            "feedPublish": "",
            "publishPackageMetadata": "true",
            "allowPackageConflicts": "false",
            "externalEndpoint": "",
            "verbosityPush": "Detailed",
            "searchPatternPack": "**/*.csproj",
            "configurationToPack": "$(BuildConfiguration)",
            "outputDir": "$(Build.ArtifactStagingDirectory)",
            "versioningScheme": "off",
            "includeReferencedProjects": "false",
            "versionEnvVar": "",
            "requestedMajorVersion": "1",
            "requestedMinorVersion": "0",
            "requestedPatchVersion": "0",
            "packTimezone": "utc",
            "includeSymbols": "false",
            "toolPackage": "false",
            "buildProperties": "",
            "basePath": "",
            "verbosityPack": "Detailed",
            "arguments": ""
          }
        },
        {
          "environment": {},
          "enabled": true,
          "continueOnError": false,
          "alwaysRun": false,
```

```json
        "displayName": "Build solution **\\\*.sln",
        "timeoutInMinutes": 0,
        "condition": "succeeded()",
        "task": {
          "id": "71a9a2d3-a98a-4caa-96ab-affca411ecda",
          "versionSpec": "1.*",
          "definitionType": "task"
        },
        "inputs": {
          "solution": "$(Parameters.solution)",
          "vsVersion": "latest",
          "msbuildArgs": "",
          "platform": "$(BuildPlatform)",
          "configuration": "$(BuildConfiguration)",
          "clean": "false",
          "maximumCpuCount": "false",
          "restoreNugetPackages": "false",
          "msbuildArchitecture": "x86",
          "logProjectEvents": "true",
          "createLogFile": "false",
          "logFileVerbosity": "normal"
        }
      },
      {
        "environment": {},
        "enabled": true,
        "continueOnError": false,
        "alwaysRun": false,
        "displayName": "VsTest - testAssemblies",
        "timeoutInMinutes": 0,
        "condition": "succeeded()",
        "task": {
          "id": "ef087383-ee5e-42c7-9a53-ab56c98420f9",
          "versionSpec": "2.*",
          "definitionType": "task"
        },
        "inputs": {
          "testSelector": "testAssemblies",
          "testAssemblyVer2": "**\\$(BuildConfiguration)\\*test*.dll\n!**\\obj\\**",
          "testPlan": "",
          "testSuite": "",
          "testConfiguration": "",
          "tcmTestRun": "$(test.RunId)",
          "searchFolder": "$(System.DefaultWorkingDirectory)",
          "resultsFolder": "$(Agent.TempDirectory)\\TestResults",
          "testFiltercriteria": "",
          "runOnlyImpactedTests": "False",
          "runAllTestsAfterXBuilds": "50",
          "uiTests": "false",
          "vstestLocationMethod": "version",
          "vsTestVersion": "latest",
          "vstestLocation": "",
          "runSettingsFile": "",
          "overrideTestrunParameters": "",
          "pathtoCustomTestAdapters": "",
          "runInParallel": "False",
          "runTestsInIsolation": "False",
          "codeCoverageEnabled": "False",
          "otherConsoleOptions": "",
          "distributionBatchType": "basedOnTestCases",
          "batchingBasedOnAgentsOption": "autoBatchSize",
          "customBatchSizeValue": "10",
          "batchingBasedOnExecutionTimeOption": "autoBatchSize",
          "customRunTimePerBatchValue": "60",
          "dontDistribute": "False",
          "testRunTitle": "",
          "platform": "$(BuildPlatform)",
          "configuration": "$(BuildConfiguration)",
          "publishRunAttachments": "true",
```

```
      "failOnMinTestsNotRun": "False",
      "minimumExpectedTests": "1",
      "diagnosticsEnabled": "false",
      "collectDumpOn": "onAbortOnly",
      "rerunFailedTests": "False",
      "rerunType": "basedOnTestFailurePercentage",
      "rerunFailedThreshold": "30",
      "rerunFailedTestCasesMaxLimit": "5",
      "rerunMaxAttempts": "3"
    }
  },
  {
    "environment": {},
    "enabled": true,
    "continueOnError": true,
    "alwaysRun": false,
    "displayName": "Publish symbols path",
    "timeoutInMinutes": 0,
    "condition": "succeeded()",
    "task": {
      "id": "0675668a-7bba-4ccb-901d-5ad6554ca653",
      "versionSpec": "2.*",
      "definitionType": "task"
    },
    "inputs": {
      "SymbolsFolder": "$(Build.SourcesDirectory)",
      "SearchPattern": "**\\bin\\**\\*.pdb",
      "IndexSources": "true",
      "PublishSymbols": "false",
      "SymbolServerType": "TeamServices",
      "SymbolsPath": "",
      "CompressSymbols": "false",
      "DetailedLog": "true",
      "TreatNotIndexedAsWarning": "false",
      "SymbolsMaximumWaitTime": "",
      "SymbolsProduct": "",
      "SymbolsVersion": "",
      "SymbolsArtifactName": "Symbols_$(BuildConfiguration)"
    }
  },
  {
    "environment": {},
    "enabled": true,
    "continueOnError": false,
    "alwaysRun": true,
    "displayName": "Copy Files to: $(build.artifactstagingdirectory)",
    "timeoutInMinutes": 0,
    "condition": "succeededOrFailed()",
    "task": {
      "id": "5bfb729a-a7c8-4a78-a7c3-8d717bb7c13c",
      "versionSpec": "2.*",
      "definitionType": "task"
    },
    "inputs": {
      "SourceFolder": "$(system.defaultworkingdirectory)",
      "Contents": "**\\bin\\$(BuildConfiguration)\\**",
      "TargetFolder": "$(build.artifactstagingdirectory)",
      "CleanTargetFolder": "false",
      "OverWrite": "false",
      "flattenFolders": "false",
      "preserveTimestamp": "false"
    }
  },
  {
    "environment": {},
    "enabled": true,
    "continueOnError": false,
    "alwaysRun": true,
    "displayName": "Publish Artifact: drop",
```

```json
                "timeoutInMinutes": 0,
                "condition": "succeededOrFailed()",
                "task": {
                  "id": "2ff763a7-ce83-4e1f-bc89-0ae63477cebe",
                  "versionSpec": "1.*",
                  "definitionType": "task"
                },
                "inputs": {
                  "PathtoPublish": "$(build.artifactstagingdirectory)",
                  "ArtifactName": "drop",
                  "ArtifactType": "Container",
                  "TargetPath": "\\\\my\\share\\$(Build.DefinitionName)\\$(Build.BuildNumber)",
                  "Parallel": "false",
                  "ParallelCount": "8",
                  "FileCopyOptions": ""
                }
              }
            ],
            "name": "Agent job 1",
            "refName": "Job_1",
            "condition": "succeeded()",
            "target": {
              "executionOptions": {
                "type": 0
              },
              "allowScriptsAuthAccessOption": false,
              "type": 1
            },
            "jobAuthorizationScope": "projectCollection"
          }
        ],
        "target": {
          "agentSpecification": {
            "identifier": "vs2017-win2016"
          }
        },
        "type": 1
      },
      "repository": {
        "properties": {
          "cleanOptions": "0",
          "labelSources": "0",
          "labelSourcesFormat": "$(build.buildNumber)",
          "reportBuildStatus": "true",
          "gitLfsSupport": "false",
          "skipSyncSource": "false",
          "checkoutNestedSubmodules": "false",
          "fetchDepth": "0"
        },
        "id": "459065f3-1078-4c8a-b503-fd5f24551cd9",
        "type": "TfsGit",
        "name": "MyFirstProject",
        "url": "https://dev.azure.com/maneric2/MyFirstProject/_git/MyFirstProject",
        "defaultBranch": "refs/heads/master",
        "clean": "false",
        "checkoutSubmodules": false
      },
      "processParameters": {
        "inputs": [
          {
            "aliases": [],
            "options": {},
            "properties": {},
            "name": "solution",
            "label": "Solution",
            "defaultValue": "**\\*.sln",
            "required": true,
            "type": "filePath",
            "helpMarkDown": "The path to the Visual Studio solution file or NuGet packages.config. Wildcards
```

```
can be used.  For example, `**\\\\*.sln` for all sln files in all sub folders."
        }
      ]
    },
    "quality": "definition",
    "authoredBy": {
      "displayName": "Manuel Ericstam",
      "url": "https://spsprodweu1.vssps.visualstudio.com/A66ee456f-0938-4694-b50f-
aa2cff71051a/_apis/Identities/a2360fe7-2f0a-48c6-afe6-364e8f39df46",
      "_links": {
        "avatar": {
          "href":
"https://dev.azure.com/maneric2/_apis/GraphProfile/MemberAvatars/aad.MGQyYTViOWEtMmY2Ni03M2ZkLWIyOTItMmY5N
GQ1ODdlNWJj"
        }
      },
      "id": "a2360fe7-2f0a-48c6-afe6-364e8f39df46",
      "uniqueName": "manuel.Ericstam@gmail.com",
      "imageUrl":
"https://dev.azure.com/maneric2/_apis/GraphProfile/MemberAvatars/aad.MGQyYTViOWEtMmY2Ni03M2ZkLWIyOTItMmY5N
GQ1ODdlNWJj",
      "descriptor": "aad.MGQyYTViOWEtMmY2Ni03M2ZkLWIyOTItMmY5NGQ1ODdlNWJj"
    },
    "drafts": [],
    "queue": {
      "_links": {
        "self": {
          "href": "https://dev.azure.com/maneric2/_apis/build/Queues/26"
        }
      },
      "id": 26,
      "name": "Azure Pipelines",
      "url": "https://dev.azure.com/maneric2/_apis/build/Queues/26",
      "pool": {
        "id": 11,
        "name": "Azure Pipelines",
        "isHosted": true
      }
    },
    "id": 10,
    "name": "MyFirstProject-.NET Desktop-CI",
    "url": "https://dev.azure.com/maneric2/6c541cf6-d884-481a-9872-f3b5368cfde7/_apis/build/Definitions/10?
revision=1",
    "uri": "vstfs:///Build/Definition/10",
    "path": "\\",
    "type": "build",
    "queueStatus": "enabled",
    "revision": 1,
    "createdDate": "2020-08-13T21:06:24.327Z",
    "project": {
      "id": "6c541cf6-d884-481a-9872-f3b5368cfde7",
      "name": "MyFirstProject",
      "url": "https://dev.azure.com/maneric2/_apis/projects/6c541cf6-d884-481a-9872-f3b5368cfde7",
      "state": "wellFormed",
      "revision": 64,
      "visibility": "private",
      "lastUpdateTime": "2020-09-22T13:26:54.373Z"
    }
}
```