how to migrate the artifacts from one organization to another in azure devops

**ChatGPT said:**

To **migrate artifacts** (such as build artifacts, NuGet packages, npm packages, Maven packages, or pipeline artifacts) from **one Azure DevOps organization to another**, there is no native "one-click" migration feature. However, you can follow a structured manual or semi-automated process depending on the artifact type.

**🧭 General Steps (High-Level)**

1. **Identify artifact type**
2. **Download from source organization**
3. **Re-upload to target organization**
4. **Update pipeline definitions to point to new feeds**

**📦 Artifact Types & Migration Strategies**

**🔹 1. Pipeline Build Artifacts**

**Scenario:** You want to move .zip, .dll, .tar.gz, etc. created during CI.

**✅ Steps:**

* Go to **Pipelines → Builds** in source org
* Select a build → **Download artifact**
* Save the .zip or files locally or to Azure Storage
* In the target org:
  + Create a new build or release pipeline
  + Upload artifacts to a release pipeline via PublishBuildArtifacts or storage

yaml

CopyEdit

# Re-upload example

- task: PublishBuildArtifacts@1

inputs:

PathtoPublish: '$(Build.ArtifactStagingDirectory)'

ArtifactName: 'drop'

publishLocation: 'Container'

**🔹 2. Azure Artifacts (NuGet/npm/Maven feeds)**

If you're migrating from one Azure Artifacts feed to another (across orgs), follow these steps:

**✅ NuGet Example**

**On source org:**

bash

CopyEdit

nuget sources Add -Name "sourceFeed" -Source "https://dev.azure.com/{org1}/\_packaging/{feed1}/nuget/v3/index.json"

nuget install MyPackage -Source sourceFeed -OutputDirectory ./packages

**On target org:**

bash

CopyEdit

nuget sources Add -Name "targetFeed" -Source "https://dev.azure.com/{org2}/\_packaging/{feed2}/nuget/v3/index.json"

nuget push ./packages/MyPackage.1.0.0.nupkg -Source targetFeed -ApiKey az

⚠️ You must authenticate using az artifacts universal download or Azure Artifacts Credential Provider.

Same applies for:

* npm: Use npm pack + npm publish
* Maven: Use mvn deploy:deploy-file

**🔹 3. Universal Packages**

**✅ Download from source org:**

bash

CopyEdit

az artifacts universal download \

--organization https://dev.azure.com/{org1} \

--feed {feed-name} \

--name {package-name} \

--version {version} \

--path ./downloaded

**✅ Publish to target org:**

bash

CopyEdit

az artifacts universal publish \

--organization https://dev.azure.com/{org2} \

--feed {feed-name} \

--name {package-name} \

--version {version} \

--description "Migrated package" \

--path ./downloaded

**🔐 Prerequisites**

* Azure DevOps CLI (az devops extension add --name azure-devops)
* Correct PAT (Personal Access Token) or Azure login for both orgs
* Permissions on both feeds

**🧠 Best Practices**

* Use consistent **naming conventions** across orgs
* Migrate with **versioning preserved**
* Tag migrated packages/releases as migrated-from-org1
* Set **retention policies** to avoid overgrowth
* Automate using **scripts or pipelines** if you're migrating large volumes

**📦 Bonus: Automate with Script**

Here's a conceptual PowerShell script outline:

powershell

CopyEdit

# Download from org1

az artifacts universal download `

--organization https://dev.azure.com/org1 `

--feed "feed1" --name "mypkg" --version "1.0.0" `

--path "./pkg"

# Upload to org2

az artifacts universal publish `

--organization https://dev.azure.com/org2 `

--feed "feed2" --name "mypkg" --version "1.0.0" `

--path "./pkg"

Would you like:

* A full automation script?
* YAML pipeline to migrate and publish artifacts?
* NuGet/npm-specific migration examples?

Let me know and I’ll tailor it!

**You said:**

A full automation script?

**ChatGPT said:**

Sure! Here's a **full automation script** in **PowerShell** to migrate artifacts between two Azure DevOps organizations using the **Universal Packages** method.

**✅ What it does:**

* Authenticates to both source and target Azure DevOps orgs
* Downloads a universal package from the source org
* Publishes it to the target org

You can customize it to loop through multiple packages/versions.

**📁 Prerequisites**

1. Azure CLI with DevOps extension

bash

CopyEdit

az extension add --name azure-devops

1. PATs (Personal Access Tokens) with **artifact read/write permissions** in both orgs

**📜 PowerShell Script**

powershell

CopyEdit

# === CONFIGURATION ===

# Source org

$srcOrg = "https://dev.azure.com/source-org"

$srcFeed = "source-feed"

$srcPkg = "your-package-name"

$srcVersion = "1.0.0"

# Target org

$targetOrg = "https://dev.azure.com/target-org"

$targetFeed = "target-feed"

# Output path for package download

$downloadPath = "C:\temp\artifact-migration\$srcPkg"

# Azure DevOps PATs

$env:AZURE\_DEVOPS\_EXT\_PAT = "<YOUR\_SOURCE\_PAT>" # Temporary set for login

# === STEP 1: Download from source ===

Write-Host "Downloading package from source org..."

az artifacts universal download `

--organization $srcOrg `

--feed $srcFeed `

--name $srcPkg `

--version $srcVersion `

--path $downloadPath `

--only-show-errors

# === STEP 2: Authenticate with target org ===

$env:AZURE\_DEVOPS\_EXT\_PAT = "<YOUR\_TARGET\_PAT>"

# === STEP 3: Publish to target org ===

Write-Host "Publishing package to target org..."

az artifacts universal publish `

--organization $targetOrg `

--feed $targetFeed `

--name $srcPkg `

--version $srcVersion `

--path $downloadPath `

--description "Migrated from $srcOrg" `

--only-show-errors

Write-Host "✅ Migration complete!"

**🛡️ Security Tip:**

Avoid hardcoding PATs in scripts. Use:

* Azure Key Vault
* Environment variables
* Secure DevOps Library in pipelines

**🧪 Optional Enhancements:**

* Loop over multiple package names/versions
* Validate package existence in target before pushing
* Integrate into Azure DevOps pipeline YAML

Would you like this as:

* A Bash version?
* A YAML pipeline for CI/CD migration?
* Support for other artifact types (NuGet/npm)?

Let me know and I’ll generate it!

**You said:**

A YAML pipeline for CI/CD migration?

**ChatGPT said:**

Here’s a complete **Azure DevOps YAML pipeline** that automates **artifact migration** (Universal Packages) from one Azure DevOps organization to another.

**🧭 What It Does**

This pipeline will:

1. Authenticate to the **source Azure DevOps org**
2. Download a universal package from a **source feed**
3. Authenticate to the **target Azure DevOps org**
4. Upload the package to the **target feed**

**✅ Prerequisites**

1. Install Azure CLI in the agent image (windows-latest or ubuntu-latest)
2. Create two service connections or use **environment variables for PATs**
3. Enable the **Azure DevOps extension**:

bash

CopyEdit

az extension add --name azure-devops

**📄 YAML Pipeline: azure-pipelines.yml**

yaml

CopyEdit

trigger: none # Manually triggered

parameters:

- name: sourceOrg

default: "https://dev.azure.com/source-org"

- name: sourceFeed

default: "source-feed"

- name: sourcePackage

default: "my-universal-pkg"

- name: sourceVersion

default: "1.0.0"

- name: targetOrg

default: "https://dev.azure.com/target-org"

- name: targetFeed

default: "target-feed"

- name: sourcePat

default: "$(sourcePAT)" # Add to variable group or pipeline variable (secret)

- name: targetPat

default: "$(targetPAT)" # Add to variable group or pipeline variable (secret)

jobs:

- job: MigrateArtifact

displayName: "Migrate Universal Package"

pool:

vmImage: 'ubuntu-latest'

steps:

- script: |

echo "Installing Azure DevOps CLI extension..."

az extension add --name azure-devops

echo "Authenticating to source organization..."

export AZURE\_DEVOPS\_EXT\_PAT='${{ parameters.sourcePat }}'

az artifacts universal download \

--organization ${{ parameters.sourceOrg }} \

--feed ${{ parameters.sourceFeed }} \

--name ${{ parameters.sourcePackage }} \

--version ${{ parameters.sourceVersion }} \

--path ./downloaded-pkg

echo "Authenticating to target organization..."

export AZURE\_DEVOPS\_EXT\_PAT='${{ parameters.targetPat }}'

az artifacts universal publish \

--organization ${{ parameters.targetOrg }} \

--feed ${{ parameters.targetFeed }} \

--name ${{ parameters.sourcePackage }} \

--version ${{ parameters.sourceVersion }} \

--description "Migrated from ${{ parameters.sourceOrg }}" \

--path ./downloaded-pkg

displayName: "Download from source and upload to target"

**🔒 Secure PAT Handling**

* Create two pipeline variables: sourcePAT and targetPAT
* Mark both as **“Keep this value secret”**
* Use a **variable group** linked to an Azure Key Vault for better security (optional)

**🧠 Tips**

* Add a loop over versions or packages using a script if needed.
* Add a dependsOn job or environment to promote to higher environments.
* This YAML assumes use of **Universal Packages**. Let me know if you want to convert it for **NuGet**, **npm**, or **Maven**.

Would you like:

* This pipeline adapted to **NuGet/npm** feeds?
* A downloadable .zip with sample folder structure?
* A GitHub Actions version?

Let me know and I’ll help further!

Top of Form

Tools

Bottom of Form

ChatGPT can make mistakes. Check importan