

Migrating only specific repositories from Nexus3 to Artifactory is explained in this document.

In order to do so, you would have to run the migration in multiple steps:

[Run the migration tool in multiple stages](#)

This document assumes the latest version of the tool is [jfrog-nexus-migrator-0.9.1.sh](#) ( check for the latest version in [Migrating from Sonatype Nexus Repository Manager to Artifactory](#) )

Let's go over these steps one by one:

1)Run

```
C/C++  
./jfrog-nexus-migrator-0.9.1.sh config
```

This will ask how to connect to Artifactory ( and saves the details to /opt/migration/nexus-migrator/jfrog-cli.conf.v5) and Nexus ( and saves the details to /opt/migration/nexus-migrator/migrationConfig.yaml) .

Note: In the migrationConfig.yaml the default threads is 12. You can monitor the migration and increase the threads if monitoring shows everything is fine in Nexus and Artifactory during the migration .

2) Run

```
C/C++  
./jfrog-nexus-migrator-0.9.1.sh getConfig
```

3) After running (1) and (2), a migration folder would be created (under the path where the script is located), you should navigate into the "migration" folder -> and then into the nexus-migrator folder.

```
varunm@varunm-nexus3:/opt/migration/nexus-migrator$ ls -ltr  
total 296  
-rw-r--r-- 1 root root 868 Apr 24 13:54 migrationConfig.yaml  
-rw-r--r-- 1 root root 266671 Apr 24 13:54 nexus.json  
-rw-r--r-- 1 root root 570 Apr 24 13:54 repositories.list  
-rw-r--r-- 1 root root 3 Apr 24 13:54 securityRegexMap.list  
-rw-r--r-- 1 root root 214 Apr 24 13:54 security.json  
drwxr-xr-x 6 root root 4096 Apr 24 14:11 artifact-cache  
-rw-r--r-- 1 root root 1484 Apr 24 14:11 repositories.yaml  
-rw-r--r-- 1 root root 15 Apr 24 14:11 assetmap.json  
-rw-r--r-- 1 root root 2 Apr 24 14:11 artifact.list  
varunm@varunm-nexus3:/opt/migration/nexus-migrator$
```

4) In the nexus-migrator folder, you should edit the file named "repositories.yaml" and delete any repository which you do not want to be created in Artifactory, an example from my end:

```
Repositories:
  localRepositories:
    docker-local:
      repoLayout: simple-default
      type: docker
    maven-releases:
      repoLayout: maven-2-default
      type: maven
      handleReleases: "true"
    maven-snapshots:
      repoLayout: maven-2-default
      type: maven
      handleReleases: "true"
    nuget-hosted:
      repoLayout: simple-default
      type: nuget
  remoteRepositories:
    maven-central:
      type: maven
      repoLayout: maven-2-default
      url: https://repo1.maven.org/maven2/
      handleReleases: "true"
    nuget.org-proxy:
      nuget:
        downloadContextPath: api/v2/package
        feedContextPath: api/v2
      type: nuget
      repoLayout: simple-default
      url: https://api.nuget.org/v3/index.json
  virtualRepositories:
    maven-public:
      type: maven
      repositories:
        - maven-releases
        - maven-snapshots
        - maven-central
      repoLayout: maven-2-default
    nuget-group:
      type: nuget
      repositories:
        - nuget-hosted
        - nuget.org-proxy
      repoLayout: simple-default
Blobstores:
- name: default
  type: File
  path: default
```

The picture above shows the repositories.yaml file after running the getConfig step.

If you wish to only migrate the "docker-local" repository, delete everything else in the repositories.yaml :

```
Repositories:
  localRepositories:
    docker-local:
      repoLayout: simple-default
      type: docker
Blobstores:
- name: default
  type: File
  path: default
```

Note: if you want the target repository in artifactory to be a different name change that in the /opt/migration/nexus-migrator/repositories.list where we map the source and target repos.

For example here is mapping of a source repo npm ( in Nexus) which will be migrated to target repo npm ( in Artifactory):

```
C/C++
```

```
- source: npm  
  target: npm
```

Here is the repositories.list reponame syntax:

```
C/C++
```

```
repoName:
```

```
- source: <repo name in nexus>  
  target: <repo name to be created in artifactory>
```

repositories.yaml reponame syntax:

```
C/C++
```

```
Repositories:
```

```
  localRepositories:
```

```
    <repo name in nexus>:
```

```
      repoLayout: simple-default
```

```
      type: generic
```

Note: the value of target (in repositories.list) is used to indicate the name in artifactory.

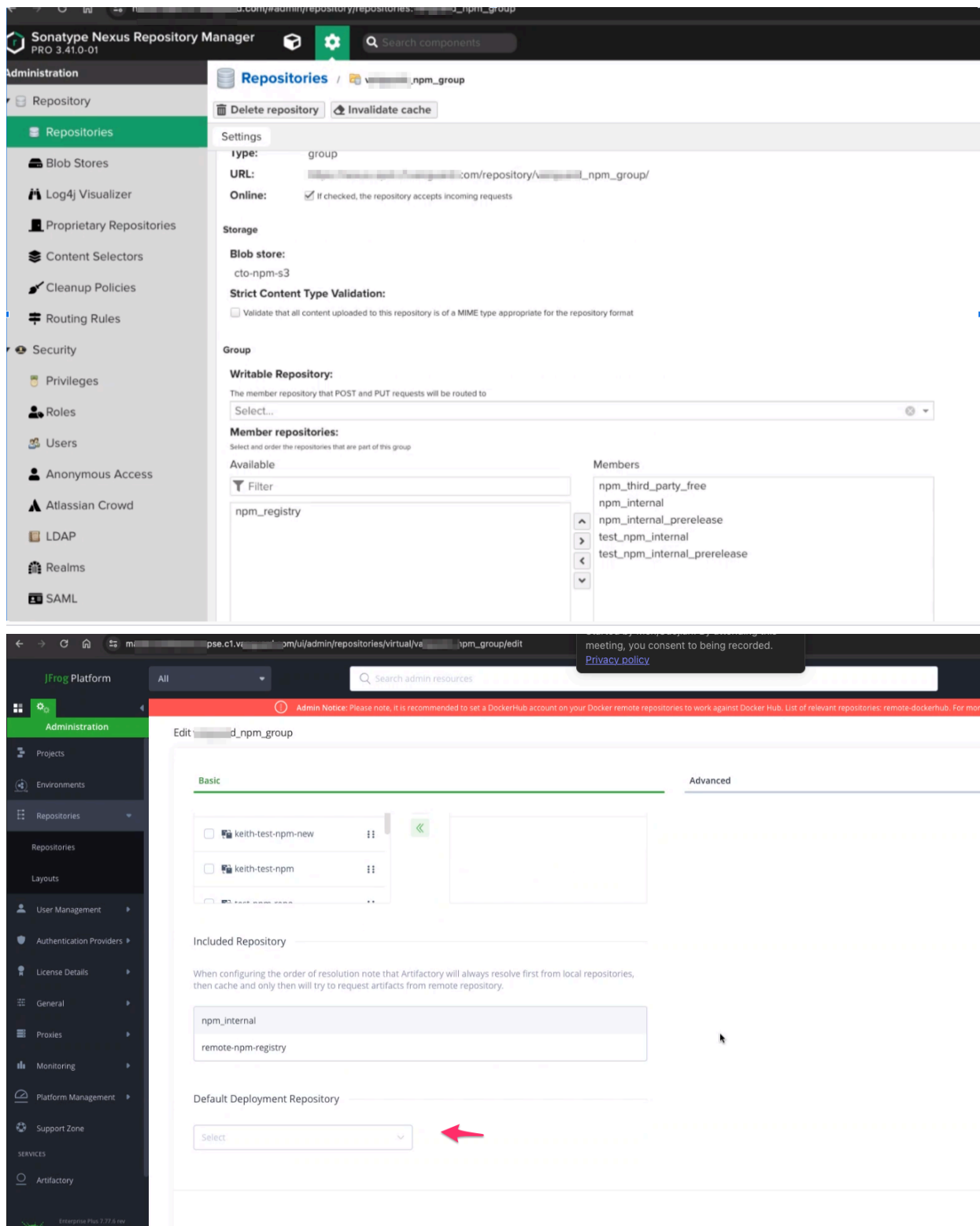
repositories.list file is the only place where new reponame of artifactory is used . In other files such as repositories.yaml the original repo name in Nexus should be used for reference.

5) After editing the repositories.yaml file, you may run the next step:

[./jfrog-nexus-migrator-0.9.1.sh](#) createRepo

This will only create the repositories specified in the **repositories.yaml** file in your target Artifactory instance ( based on the repo name you want to use for the target repo in **repositories.list**)

Note: The nexus group will map to a virtual repo in Artifactory. You can choose to configure the “Default Deployment Repository” in the virtual repo as that is not configured in the **repositories.yaml**



6) Now we can migrate the Artifacts to the specific repositories that you created with the createRepo step, in my example, I only created a "docker-local" repository, hence on this step -> I will only migrate the "docker-local" repository content:

```
./jfrog-nexus-migrator-0.9.1.sh migrateArtifact --repos="docker-local"
```

This is explained in detail in [Run the migration tool in multiple stages](#)

Note: The syntax for `--repos` with `ma` is,

```
./jfrog-nexus-migrator-0.9.2.sh ma --repos="<repo name in nexus>"
```

Another option is to use the asset file as explained in later steps .

7) Now, you may finish the migration by running the last step to migrate the users , permissions etc::

```
./jfrog-nexus-migrator-0.9.1.sh migrateSecurity
```

8) If you want to redo the migration for any repo ( say repo named npm) or do a delta migration then change the status of the repo in

/opt/migration/nexus-migrator/repositories.yaml

```
C/C++
From
  npm:
    repoLayout: simple-default
    type: npm
    status: completed

to

  npm:
    repoLayout: simple-default
    type: npm
    status: pending
```

And do a transfer using the asset file which just has the files you want to transfer to the target npm repo:

Note: An asset file with name as `<repo-name>_assetmap.json` should be created in /opt/migration/nexus-migrator folder . See steps in [nexus\\_asset\\_file\\_creator/blob/master/README.md](#) . Here the **repo-name** ( in the `<repo-name>_assetmap.json`) is name of your **target** repo ( based on log below) .

C/C++

```
2024/05/21 18:56:10 [DEBUG] : Source repository name - npm, Target repository
name - sv-npm-local
2024/05/21 18:56:10 [DEBUG] : Getting the repository migration status for
sv-npm-local
2024/05/21 18:56:10 [INFO] : Starting artifact migration for sv-npm-local [5/6]
2024/05/21 18:56:10 [ERROR] :sv-npm-local_assetmap.json does not exist in
migration/nexus-migrator/ directory.
```

When running migrate with the "--repos" option internally we generate the asset file using the Nexus public API to get the file list -

<https://help.sonatype.com/en/assets-api.html>

For example , if you run the API against any npm repos with scoped packages in it we could see it.

example :

C/C++

```
curl -u admin:Jfrog@2022 -X GET
'http://localhost:8081/service/rest/v1/assets?repository=npm' >
npm_assetmap.json
```

You will get:

C/C++

```
{
  "items" : [ {
    "downloadUrl" :
"http://localhost:8081/repository/npm/@parcel/markdown-ansi",
    "path" : "@parcel/markdown-ansi",
    "id" : "bnBtOmQ4MzMwMDVkdNDljMzQ4ZThjZmUyOTNiM2EwYzZiOGJi",
    "repository" : "npm",
    "format" : "npm",
    "checksum" : {
      "sha1" : "f732ac870b0887f6788aa9ab43e611ad2358d2a3"
    },
    "contentType" : "application/json",
    "lastModified" : "2024-05-21T16:29:20.030+00:00",
```

```

    "lastDownloaded" : null,
    "uploader" : "admin",
    "uploaderIp" : "172.16.1.6",
    "fileSize" : 832,
    "blobCreated" : null,
    "blobCreated" : "2024-05-21T16:29:20.030+00:00"
  }, {
    "downloadUrl" :
"http://localhost:8081/repository/npm/@parcel/markdown-ansi/-/markdown-ansi-2.1
2.0.tgz",
    "path" : "@parcel/markdown-ansi/-/markdown-ansi-2.12.0.tgz",
    "id" : "bnBt0jI3YjkwNGJhNDMxZD1lNWVkdExZDYyZjRlZjI4NTIw",
    "repository" : "npm",
    "format" : "npm",
    "checksum" : {
      "sha1" : "a4301321fa784a28ba817e65e41432fe8b3b3192",
      "sha512" :
"599cf7af32fc9341f7591e2a4c75fa21cf03944b35ee478ef60b43e0c346c8c3506ea404bd0eb5
95625a207d2702d8047adbb448e213895a9d66b6fccf1fccef",
      "sha256" :
"7bdc6c5f379e9f5b257a3856a52be90be4e335018f805a156f6ad382d5d1b050",
      "md5" : "6d3c05d7040971e18ab1804f6edb9a9d"
    },
    "contentType" : "application/octet-stream",
    "lastModified" : "2024-05-21T16:29:20.051+00:00",
    "lastDownloaded" : null,
    "uploader" : "admin",
    "uploaderIp" : "172.16.1.6",
    "fileSize" : 2056,
    "blobCreated" : null,
    "blobCreated" : "2024-05-21T16:29:20.051+00:00",
    "npm" : {
      "name" : "@parcel/markdown-ansi",
      "version" : "2.12.0"
    }
  } ],
  "continuationToken" : null
}

```

But when running the `migrateasset` command using ``ma --use-existing-asset-file="true"`` the `<repo-name>_assetmap.json` the asset list should be in below format:

Unset

```
{
  "assets": [
    {
      "source": "SOURCE_REPO_NAME_IN_NEXUS/testgp/testad/v12/testad-v12.jar",
      "fileblobRef": ""
    },
    {
      "source": "SOURCE_REPO_NAME_IN_NEXUS/testgp/testad/v12/testad-v12.jar.sha256",
      "fileblobRef": ""
    },
    ...
  ]
}
```

Note: the "fileblobRef": "" and SOURCE\_REPO\_NAME\_IN\_NEXUS  
Example:

C/C++

```
{
  "assets": [
    {
      "source": "npm/@parcel/markdown-ansi/-/markdown-ansi-2.9.3.tgz",
      "fileblobRef": "",
      "lastDownloaded": "null",
      "lastUpdated": "1693320901754",
      "createdDate": "1693320901754"
    }
  ]
}
```

You can run it as:

C/C++

```
./jfrog-nexus-migrator-0.9.2.sh ma --use-existing-asset-file="true" && tail -F migration.log
```

You should see "Success - 1" as shown in migration.log snippet below:



C/C++

```
2024/05/21 23:00:47 [DEBUG] : Source repository name - npm, Target repository
name - npm
2024/05/21 23:00:47 [DEBUG] : Getting the repository migration status for npm
2024/05/21 23:00:47 [INFO] : Starting artifact migration for npm [5/6]
2024/05/21 23:00:47 [INFO] : Reading assets from
migration/nexus-migrator/npm_assetmap.json
2024/05/21 23:00:47 [DEBUG] : Creating temporary directories for artifacts
migration
2024/05/21 23:00:47 [INFO] : Generated final artifact list for migration
2024/05/21 23:00:47 [INFO] : Starting artifacts migration
2024/05/21 23:00:47 [DEBUG] : Setting the number of threads for migration
2024/05/21 23:00:47 [INFO] : [Thread 3] [1/1] Fetching artifact :
npm/@parcel/markdown-ansi/-/markdown-ansi-2.9.3.tgz
2024/05/21 23:00:47 [INFO] : [Thread 3] [1/1] Deploying artifact :
npm/@parcel/markdown-ansi/-/markdown-ansi-2.9.3.tgz
2024/05/21 23:00:47 [INFO] : Artifact migration got completed for npm
```

#### Artifact Migration Report

npm : Failed - 0 | Success - 1

---