

4

Exporting & Importing Project Data

This section discusses the export and import features of FlexNet Code Insight.

Exporting and Importing

Introduction

FlexNet Code Insight provides export and import functionality for project data. The export/import can be performed on the same server or across servers. Project data can be quickly imported into an empty project to create live inventory, or imported into a scanned project to create live inventory with file associations. The functionality can be useful in any of these following scenarios:

- **Backup of project and audit data:** Use export to create a snapshot of a project at a given point in time. The exported backup data file will include all project and scan details (owner, dates, scan/policy settings, etc.), project inventory (containing inventory details, field values, file associations, and inventory status) and a list of scanned files and their review status. The project data snapshot can be stored for backup and restored if/when recovery of a project is needed.
- **Copying a project:** Use export/import to create an exact copy of a project for future scanning and audit work. To do this, export the data from the source project, scan the target project (pointed to the same codebase) and import the data into the target project. The target project can be used for continued scanning and analysis work while the source project remains unchanged.
- **Branching a project and Audit work reuse:** Use export/import to apply analysis work performed on one project, to another project containing a similar codebase. For example, you can apply the analysis work performed on project “foo-v1” to project “foo-v2”. To do this, export the data from “foo- v1”, scan “foo-v2”, import the data into “foo-v2”. By default, only inventory that is relevant between the source and target project based on file matches, is processed during import. For example, “foo-v1” inventory without a file match to “foo-v2” is considered an empty inventory item and therefore will not be imported. If you wish to change the default behavior to process all inventory (including empty inventory) set the flag *createEmptyInventory=false* during import.
- **Sharing of live audit results between teams:** Use export/import to share live audit results between teams or with Flexera Professional Services. For example, you can export data from “project 1” on instance 1 and import the data into “project 2” on instance 2. Results can be imported into an empty project for a live view of inventory or imported into a scanned project for a live view of inventory with file associations and access to the codebase file-tree. Use the *createEmptyInventory=true* flag during import to make sure all inventory is processed.

- **Migrating audited projects from FNCI v6 to FNCI v7:** Use import to create live project inventory in Code Insight from an FNCI v6 project/workspace data file. After exporting data from FNCI v6, run the **Audit Data Migration Tool** (available for download in the [Flexera Product and License Center](#)) to map inventory fields from v6 to v7 and to convert the exported data to the proper JSON data format required for import into FNCI v7. Then perform an import into an FNCI v7 project. Import into an empty project for a live view of inventory or into a scanned project for a live view of inventory with file associations.
- **Creating inventory from an external system:** Use import to create live project inventory in Code Insight from a data file containing legacy or external data. This type of import requires conversion of the legacy data to the import data JSON format prior to importing into the new project.

Exporting Project Data

Introduction

Project data export is accessible via Web UI or via a REST interface in one of the following ways:

- Select **Export Project Data** from the **Manage Project** dropdown menu on the **Project Summary** page. See [Exporting Project Data Using the FlexNet Code Insight UI](#).
- Call the **Export Project Data API**. Invoke the Export Project Data REST API from any REST client or command line tool that supports cURL.

The project data is exported into a JSON data file, and then compressed into a zip archive. The archive file can be stored for backup purpose or imported into a project using the Import Project Data API.

Project export runs as a background process that does not interfere with scanning or analysis work.



Note • These instructions are specific to the version of FNCI containing the data to be exported and may not apply to other versions. If exporting from a different version of FNCI, please be sure to consult the documentation included with that version.

What Data is Exported?

The project data export is always processed in full. In other words, there is no way to limit the exported data. The export data file includes the following content:

- **Export Information:** date and time, owner, Code Insight version, CL and Electronic Update version, etc.
- **Project Details and Scan Settings:** name, description, scan profile, policy profile, scanroot path.
- **Inventory:** all inventory data, fields, publish and review status, associated files, associated repository item.
- **Reviewed Files:** absolute files paths and MD5s for all files marked as reviewed.
- **Custom Data:** custom versions, licenses and vulnerabilities and custom mappings between these entities.



Note • Some data contained in the exported data file is for informational purpose only and will not necessarily be processed during an import.

Prerequisites for Export

- A running instance of Code Insight v7.
- An existing, non-empty project (containing at least one inventory item).
- If exporting via REST API, a valid JWT token for the project owner of the project to be exported is required. For more information, see “Generating a JWT Authorization Token” in the *FlexNet Code Insight Installation and Configuration* guide. The token is not required if exporting via Web UI. A JWT token is not required if exporting via WebUI.
- If exporting via REST API, a REST client or command line interface with cURL support. Visit <https://curl.haxx.se/download.html> to download cURL. Once installed, ensure that cURL is added to your PATH environment variable so that you can use it with batch or PowerShell scripts and call it from the command prompt of any working directory. cUrl is not required if exporting via WebUI.

Exporting Project Data Using the Web UI



Task *To export project data via Web UI, do the following:*

1. Log into Code Insight as the project owner of the project you want to export.
2. Navigate to the **Project Summary** page. See [Project Summary Page](#) for information about the fields on the page.
3. Open the **Manage Project** dropdown and select **Export Project Data**.
4. When prompted, choose a name for the data file or keep the default name (the default name includes the project Id and timestamp).
5. Select a location to store the data file.
6. When complete, Code Insight generates a zip archive containing the project data in JSON format, and saves it to the specified location, for example:
 - C:/fnci/project_export/MyProject.zip/5-export-12-22-2020_10-42.zip
 - OR
 - C:/fnci/project_export/MyProject.zip/MyProject-export-12-22-2020_10-42.json



Note • Export via Web UI is currently only supported for export from projects of type “Standard”. To export from “Inventory Only” projects, you must use the Export Project Data REST API.

Exporting Project Data Using the REST API



Task *To export project data using the Export Project Data REST API, do the following:*

1. Determine the ID of the project that you want to export. You can do this visually via UI or programmatically via API.
 - Find the project ID in the **URL** of the **Project Details** page of a given project. For example:
http://localhost:8888/codeinsight/FNCI#myprojectdetails/?id=1.
 - Call the **Get Project Id API**. For example:
http://localhost:8888/codeinsight/api/project/id?projectName=eportal-1.3



Note • When using the Get Project Id API to look up the name, if the project name contains a space or special character, replace the character with its encoded version. For example, use “project%20foo” for a project named “project foo”.

2. Modify the following cURL command by substituting the highlighted variables based on your server and project information. Specify the name of the zip file to redirect the command output to.

```
curl -X GET "HOST:PORT/codeinsight/api/project/exportProjectData?projectId=PROJECT_ID" -H "accept: application/json" -H "Authorization: Bearer JWT_TOKEN" > PROJECT_DATA_FILE.zip
```

For example:

```
curl -X GET "http://localhost:8888/codeinsight/api/project/exportProjectData?projectId=55" -H "accept: application/json" -H "Authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJqcjV1c2V5SWQjE1MTA5NjM2NzZ9" > MyProjectData file.zip
```

3. Execute the above statement using a command line utility that supports cURL.



Note • If copying the command from above, use a Text Editor to remove formatting and any line breaks or extra spaces.

- The status of the export process will appear in the command prompt window as such:

```
Export Zip
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           0 33917   0     0    11305    0  --:--:--  0:00:03  --:--:-- 11253
```

- When complete, Code Insight generates a zip archive containing the project data in JSON format, and saves it to directory from which the statement was executed, for example:

- C:/fnci/project_export/MyProject.zip/5-export-12-22-2020_10-42.zip
- OR
- C:/fnci/project_export/MyProject.zip/MyProject-export-12-22-2020_10-42.json



Note • If an archive with the same name already exists in the export data directory, the archive will be overwritten.

- Unzip the archive.
- Verify that the export process completed successfully by opening the project data file with a utility that supports JSON format such as Textpad or Notepad++.



Note • If the export process is not successful, the zip archive will contain a file with the status code and error message. After you resolve the error, delete the invalid archive file, and re-run the script.

Importing Project Data

Introduction

Project data import is available through a REST interface using the **Import Project Data API** which can be invoked from any REST client or command line tool that supports cURL.

The input to the API is an archived JSON data file containing project data. The archive is first decompressed and the included data file is then applied to the specified target project.

You can create the JSON data file for input, by performing an export from an FNCI v7 project (according to the instructions in [Exporting Project Data](#)), by performing an export from an FNCI v6 project and using the Audit Migration Tool to convert the data into the required format, or manually from data exported from an external system. In all cases, the input to the API must be a data file of the expected JSON data format.

Import is typically performed on a target project of type “Standard” pointed to an already-scanned codebase. A scan of the target project codebase is required prior to import in order to build the codebase file tree and to prepare the project for storing file information. This type of import creates live inventory with file associations and provides access to the codebase files via Analysis Workbench. Alternatively, an import may be performed into a project of type “Inventory Only” without a scanned codebase. This type of import creates live inventory without any file associations and does not provide access to the codebase files via Analysis Workbench. See the section **Types of Import** for additional information.



Note • These instructions are specific to the version of FNCI to which data will be imported and may not be applicable for other versions. Please be sure to consult the documentation provided with your FNCI version.



Note • A Standard Import assumes that the codebase has been scanned on the destination instance.

What Data is Imported

The data that is imported is controlled by the **Project Type** of the target project (the project to which data is imported to). For example, if the **Project Type** of the target project is “Standard”, then the import will be a standard import. If the Project type of the target project is “Inventory Only”, then the import will be an inventory only import. Differences are discussed below.

Standard Import:

1. Provides a full import (**processes both inventory and files**). Processes all inventory items (published and unpublished). Includes inventory details, inventory status, associated repository item (component/version/license) and lists associated file paths.
2. Processes all files that are marked as reviewed. Includes their absolute files paths and MD5s. Unreviewed files are not imported.
3. Processes custom data. Imports custom versions, custom licenses and custom mappings.
4. Requires a target project of type “Standard” for import.
5. Requires a scanned codebase in the target project.
6. Suitable for full copying of projects, branching, and ongoing scanning and analysis.
7. Provides direct access to inventory via Inventory page and to codebase files via Analysis Workbench.

Inventory Only Import:

1. Provides a partial import (**processes only inventory, without processing files**). Processes published inventory only. Includes inventory details, inventory status, associated repository item (component/version/license) and lists file paths if available.
2. Does not process files (reviewed or unreviewed).
3. Processes custom data. Imports custom versions, custom licenses and custom mappings.
4. Requires a target project of type “Inventory Only”.
5. Does not require a scanned codebase (or associated codebase) on the target project.
6. Suitable for creating live inventory from an external system, or from an FNCI Inventory Only project that is configured to scan remotely using a Jenkins or other Scan plugin.
7. Provides direct access to inventory via Inventory page, without access to codebase files via Analysis Workbench.



Note • Import does not process custom components. Instead, Code Insight creates work-in-progress (WIP) or license-only type inventory to represent the item.

Prerequisites for Import

- A running instance of Code Insight v7.
- An existing project to which data will be imported. The project can be empty or non-empty depending on the desired outcome. Likewise, the project should be of type “Standard” with a scanned codebase or “Inventory Only” without a scanned codebase depending on the desired outcome.
- A valid zip archive containing the JSON data file with the project data to be imported.

- A completed Electronic Update.
- A valid JWT token for the project owner of the project to be used for import is required. For more information, see “Generating a JWT Authorization Token” in the *FlexNet Code Insight Installation and Configuration* guide.
- A REST client or command line interface with cURL support. Visit <https://curl.haxx.se/download.html> to download cURL. Once installed, ensure that cURL is added to your PATH environment variable so that you can use it with batch or PowerShell scripts and call it from the command prompt of any working directory.



Note • *Running an import on a project currently being scanned is not recommended.*

Additional Information about Import

How Files & Inventory Are Processed During Import

During import, the file path (and optionally the MD5) is the key between data in the data file and the codebase files on the target server. File paths are matched between the data file and codebase files by subtracting the root path from the absolute path. For example:

- absolute file path: /home/fnci/scanRoot/1/ePortal-1.3/src/gettext.c
- root path: /home/fnci/scanRoot/1/
- file path: /ePortal-1.3/src/gettext.c

The following default logic applies during the import process:

- Only files that match based on file path are processed for inventory-to-file association in the target project.
- Only files that match based on file path and MD5 are processed when marking files as reviewed in the target project.
- Only Inventory containing files that match based on file path is processed during import into the target project (empty inventory is not created).

Import Options

The following options may be applied during import to override the default settings as needed:

checkInventory (default: false)

If enabled, only files with matching MD5 in the import data file and the scanned codebase will be associated to inventory in the target project.

checkReviewed (default: true)

If enabled, only files with matching MD5 in the import data file and the scanned codebase will be marked as reviewed.

createEmptyInventory (default: false)

If enabled, all inventory (with or without matched files based on file path) is processed during import.



Note • If the `checkInventory`, `checkReviewed`, and `createEmptyInventory` options are not set, the default values apply.

Importing Project Data Using the REST API



Task To import project data using the Import Project Data REST API

1. Create a Standard or Inventory Only project for import, depending on the type of import. For information on creating a project, see [Creating a Project](#).



Note • You are most likely to have an Inventory Only project if the project was created as part of a remote scan on an external system using the Jenkins plugin or another Scan Agent plugin.

2. For Standard import only, upload and scan the same or a similar codebase. A scan is not required for an Inventory-only project.
3. Determine the ID of the project to which the data will be imported. You can do this visually via UI or programmatically via API.
 - Find the project ID in the **URL** of the **Project Details** page of a given project. For example: `http://localhost:8888/codeinsight/FNCI#myprojectdetails/?id=1`.
 - Call the **Get Project Id API**. For example: `http://localhost:8888/codeinsight/api/project/id?projectName=portal-1.3`



Note • When using the *Get Project Id API* to look up the name, if the project name contains a space or special character, replace the character with its encoded version. For example, use *“project%20foo”* for a project named *“project foo”*.

4. Modify the following cURL command by substituting the variables based on your server and project information:

```
curl -X POST --data-binary "@PROJECT_DATA_FILE.zip" "HOST:PORT/codeinsight/api/importer/importProjectData?projectId=PROJECT_ID" -H "accept: application/json" -H "authorization: Bearer JWT_TOKEN" -H "cache-control: no-cache" -H "Content-Type: application/octet-stream"
```

The following is an example:

```
curl -X POST --data-binary "@MyProjectData file.zip" "http://localhost:8888/codeinsight/api/importer/importProjectData?projectId=1" -H "accept: application/json" -H "authorization: Bearer eyJzdWIiOiJqcnViaW4iLCJ1c2VySWQiOjEwLjE1MTA5NjM2NzZ9" -H "cache-control: no-cache" -H "Content-Type: application/octet-stream"
```



Note • By default, empty inventory items without file associations (by file path between source and target project) are not imported. If prefer to import empty inventory (for example if importing from a scanned project into an inventory only project) you must use the *“createEmptyInventory=true”* option for inventory to be generated in the new project.



Note • The *PROJECT_DATA_FILE* parameter refers to the name of the project data file containing the data you wish to import. The *PROJECT_ID* refers to the ID of the project you want to import the data into.

5. Execute the above statement using a command-line utility that supports cURL.



Note • If copying the command from above, use a Text Editor to remove formatting and any line breaks or extra spaces.

6. When the import is done, a status message with the term **OK** will appear in the command prompt window. If the import is not successful, a status code and error message will appear.
7. To verify that the import completed successfully, open the target project in Code Insight and navigate to the **Inventory** page. See [Searching for Projects](#). Confirm the total number of inventory items based (only inventory with matched files is imported by default) and that the inventory items contain accurate inventory details and file path associations.

Special Considerations for Export/Import

Empty Inventory

An empty inventory item is one that does not contain any associated files. By default, empty inventory is not processed during import and no empty inventory items are created in the target project.

Duplicate Inventory

Two inventory items are considered identical between the source and target project if they are associated with the same repository item (by component ID). Duplicate inventory is resolved during import and all fields in the target project are overridden by the data in the data file, with the exception of empty data in the data file.

Manually Unreviewed Files

Files that are manually *unreviewed* in the source project, do not retain the *unreviewed* status after standard import to the target project. This occurs because the project data file stores information only about reviewed files.

If you wish to retain both reviewed and unreviewed file status after standard import, manually mark all of the codebase files in the target project as *unreviewed* before importing the data file.