

Export IFS Content to Git

If your IBM i source code is not yet in Git, MDOpen provides the option to export IFS folder structures and files to Git, including committing the export locally and publishing on the origin server.

Before you Begin

The [Git Credentials](#) and [Git Repository](#) must already be defined in MDOpen and MDGIT must be able to connect to the Repository.

If your source code is currently in source physical files on the IBM i, you must first use the [MDSRC2IFS](#) command to copy the code to the IFS.

Export to Git

In MDOpen: **MDXREF** → **IFS Contents**

1. Navigate to the IFS folder containing the content you wish to export
2. Multi-Select each folder or file that should be exported and then select option **Export to Git**. A selected folder will be part of the path in the Git repo, but any parent folders will not be automatically part of the repo path.

The Export Dialog

The option will open a dialog with the following parameters:

Include Subfolders

- **True** — Include all subfolders and their contents in the export.
- **False** - only files directly within the selected folder(s) will be included in the export. Note - empty folders cannot be exported to Git, as this isn't permitted by the Git commit process.

File Name Pattern

Optionally filter the export to only include files with names that match a specific pattern. For example, would only include files with the extension. This parameter supports the standard wildcard character . If left blank, all files will be included regardless of name.

Git Repository

The Git Repository to export to. This must already be defined in MDOpen under Git Repositories. Use content-assist to select from a list.

Branch

The name of the branch to export to. Use content-assist to select from a list of existing branches on the origin server, or enter the name of a new branch to create. If the branch does not already exist on the origin server, it will be created during the export process.

It is recommended to export to a new branch and then validate the contents. If anything is not as intended, you can then simply delete the branch on the origin server (which triggers the deletion of the branch in the IFS) and then try again. If correct, then create a pull-request to merge the exported code into your main branch.

Create from Branch

The existing branch to create the new branch from. This is only relevant if you are exporting to a new branch that does not already exist on the origin server. Use content-assist to select from a list of existing branches on the origin server.

Target Path in Repository

The path within the Git repository to export to. For example, if you enter `src/sql` then the content will be exported to a folder named `sql` within a folder named `src` in the Git repository. If the specified path does not already exist in the repository, it will be created during the export process. If left blank, the content will be exported to the root of the repository.

Commit Message

The commit message to use for the Git commit that is created during the export process. This should be a meaningful message that describes the content being exported, such as "Initial export of source code from IFS".

Commit Template for Committer / GPG Key

Optionally specify a [Commit Template](#) to determine the committer name/email and GPG key to use for signing the commit created during the export process. If left blank, the committer will be set to the current user's [mapped user in Git](#) and no GPG key will be used.

The Export Process

When you click the **Confirm** button to start the export, the MDGIT service jobs will perform the export. Once complete, if successful, the info message "Export to Git completed" will be displayed. If there are any errors, an actionable message will be displayed with ability view the MDGIT IFS logs. The MDGIT IFS logs can also be viewed during the process to follow the progress of the export.