

Commit Templates

In MDOpen: **Settings** → **Template Settings** → **Git Commit**

Commit Templates are Attribute Templates that are used to automate the commit of changes to source code or *IFS files after an MDCMS RFP installation is complete.

When an object request in the RFP uses an MDCMS Attribute that is assigned to a commit template, MDCMS will consider the code for staging and committing to the Git Repository defined for the template.

Git Commit Templates are useful to automatically replicate source changes to a Git repository, even when the development team otherwise doesn't use Git.

Git Commit Templates are also useful if further edits to the code occur directly on the IBM i prior to running the RFP, in order to ensure that those changes are also captured and committed back to the Git repository.

During the Post-Installation phase of the RFP, MDCMS will create an MDGIT transaction record for each distinct combination of:

- Git Commit Template - Git Repository
- Target Branch - Object Requester

For each transaction record, the MDGIT Service job named MDGITC will then perform the following steps: 1. Create the target branch if it does not exist.

1. Pull the branch from the origin Git repository.
2. Find the relative path and file naming/extension in the Repository based on the [Attribute Mapping definitions](#).
3. If the file already exists in the repository, generate an SHA-1 signature for the RFP code and an SHA-1 signature for the code in the repository and ignore the file if the signatures match. The signature generation ignores whitespace and line-ending differences, so that files will not be committed if there are no substantive code changes even if there are formatting differences.
4. Stage the files for commit to the branch.
5. Commit the staged changes to the local Git repository in the IFS with the following properties:
 6. a commit message based on the template
 7. the author of the commit set to the Object Requester, and the email set to the email of the Object Requester that is registered in MDSEC
 8. the committer name/email based on the template on the Git server
 9. GPG signing, if enabled for the template
10. Push the Commit to the Git Repository on the Origin Server

Git Commit Template Parameters

Template ID

A unique identifier of up to 10 characters for this template.

Description

A human-readable description of the template.

Repo ID

The Git repository this template applies to. Use content-assist to select from existing repositories.

Commit to CI Branch

- **True** — Commit changes to the same branch that the CI process or local workspace request used to generate the Object Request. If the object request did not originate from the CI process, or through a request from a clone in the local workspace, then the Branch Name parameter value will be used instead.
- **False** — Commit changes to the Branch Name parameter value, regardless of the branch used during request of the Object.

Branch Name

The name of the branch to commit to. This can be defined using a combination of static text and placeholders representing task and project information. For example, the format `feat - ++PRJTSK++` would create branches named `feat - MYPROJ - 123 . 1` for tasks in the MYPROJ project with task number 123 and subtask number 1.

Press F7 to view available placeholders and to insert them at the current cursor position in the Branch Name field.

Content-Assist can be used to view the list of existing branch names and to select an existing branch name.

Crt from Branch

The existing branch that new branches should be created from if the branch defined in Branch Name does not already exist. If left blank or set to *MAIN, new branches are created from the branch designated as the Main Branch for the repository.

Press F7 to view available placeholders and to insert them at the current cursor position in the Crt from Branch field.

Content-Assist can be used to view the list of existing branch names and to select an existing branch name.

Commit Message

The commit message to use for commits made with this template. This can be defined using a combination of static text and placeholders representing RFP, task and project information. For example, the format

`Committing changes for ++PRJTSK++ using RFP ++RFPNBR++` would create commit messages like `Committing changes for MYPROJ-123.1 using RFP 12345` for a commit related to a task in the MYPROJ project with task number 123 and subtask number 1, and an RFP number of 12345.

Committer Name

The name of the committer to use for commits made with this template.

***USER** — the committer name/email will be the same as the author (the Object Requester).

Committer Email

The email of the committer to use for commits made with this template. Ignored if Committer Name is set to *USER.

Upper-Case Git File name for new Source Members

- **True** — new source members created in the Git repository will have upper-case file names.
- **False** — new source members created in the Git repository will have lower-case file names.

Upper-Case Git File type for new Source Members

- **True** — new source members created in the Git repository will have upper-case file types (extensions).
- **False** — new source members created in the Git repository will have lower-case file types (extensions).

End of Line Characters

The type of end of line characters to use for files committed to the Git repository that are converted from source members. - **CR** — Carriage Return only (classic Mac-style) - **CRLF** — Carriage Return + Line Feed (Windows-style) - **LF** — Line Feed only (Unix-style) (recommended) - **LFCR** — Line Feed + Carriage Return (uncommon)

Auto-Link Requests to Local Workspace

- **True** — If the Object Request originated directly within MDCMS or MDOpen, such as from the MDXREF Objects view, check if a local clone of the Repository is in the workspace and link the request to that workspace so that the developer can edit the code locally to take advantage of AI and other IDE features. See [Local Workspace Mapping](#) for more information.
- **False** — Object Requests will not be automatically linked to the repository in the local workspace.

GPG Signing

- **True** — Commits made with this template will be GPG-signed using the MD Service User's GPG key. The Service User must have a GPG key configured on the IBM i for this to work properly.

- **False** — Commits made with this template will not be GPG-signed.

GPG Key

The GPG key to use for signing commits if GPG Signing is set to True. The key is the 40-character hexadecimal key ID.

New GPG Passphrase

The passphrase for the GPG key if GPG Signing is set to True. This is only needed if the GPG key requires a passphrase and the passphrase has not already been stored for the template. The passphrase is stored securely in the MDSEC database and is not retrievable after saving.

*NONE — No passphrase is used for the GPG key

Attributes Assigned to Template

Header Options

Adjust Filter

Click on the Filter icon to filter the list by Application, Level, Object Type, MDCMS Attribute, Library or assigned Template.

Not Assigned to This

Limit the list of attributes to those that are not assigned to this template.

Assigned to Any

Limit the list of attributes to those that are assigned to any template, including this one.

Only Assigned to This

Limit the list of attributes to those that are assigned to this template

All Attributes

View all attributes regardless of assignment.

Row Options

Assign to Template

Assign the attribute to the template. This option will only appear if the attribute is not already assigned to a template, and an [Attribute Mapping definition](#) is found for the attribute. The mapping definition must contain a specific path rather than a naming pattern for the path.

Remove from Template

Remove the attribute assignment from a template. This option will only appear if the attribute is currently assigned to a template.

Map Attribute in Repository

Create an Attribute Mapping definition that maps the attribute to a specific file path in the Git repository. This option will only appear if the attribute is not already assigned to a template, and no [Attribute Mapping definition](#) is found for the attribute.

Set as Default Path for new Mapped Attributes

Set the Mapped Path as the default path for subsequent Attribute Mapping creations. This can help save a lot of time when several attributes are missing Mapping definitions. First click on this option on a row that has a similar mapped path. Then, click the **Map Attribute in Repository** option on a row where the definition doesn't exist yet.