Continuous Integration Definitions

Once a repository is defined, the repository — or specific paths within it — can be configured for CI. Multiple CI definitions can be created for the same repository when more granular control is needed.

A CI definition has the following main purposes: - Define which commits trigger CI processing in MDCMS, based on branch and path filters. - Define how commits are processed, including what gets requested, where it goes, and how it is assigned. - Define Project/Task/Subtask assignment rules based on branch naming conventions or commit message parsing. This is required even when MDCMS requests are not triggered from the push of a commit, but rather from a manually initiated request in the local workspace.

In MDOpen: **Settings → DevOps Settings → Continuous Integrations**

Branch and Path

Repo ID

The ID of the repository this CI definition applies to. Use content-assist to select from defined Git repositories.

Branch Opt

Defines which branch within the repository triggers CI processing in MDCMS.

- **Main Branch** — CI is triggered when a commit is pushed to the branch designated as Main Branch in the repository definition.
- **Not Main Branch** — CI is triggered when a commit is pushed to any branch except the Main Branch.
- **Branch Naming Pattern** — CI is triggered when a commit is pushed to any branch whose name matches the specified wildcard pattern in the Branch parameter. For example, `release-*` matches all branches starting with `release-`.
- **Specific Branch** — CI is triggered only when a commit is pushed to the branch named in the Specific Branch parameter.

Branch

Required when Branch Opt is set to **Branch Naming Pattern** or **Specific Branch**. Enter the branch pattern, branch name or use content-assist to select from a list.

Path

Restricts CI to files committed within a specific directory tree. If left blank, a file in any path committed to the repository will be considered for creating an object request in MDCMS (subject to attribute mapping exclusions). To monitor multiple distinct paths for the same repository, create a separate CI definition for each path. If the repository has already been cloned, content-assist can be used to browse and select the path.

File Naming Pattern

A file name filter applied within the configured path. If blank, all files in the path are considered (subject to attribute mapping exclusions). Use wildcard patterns to restrict by type, for example `*.jar` or by name, for example `MyFile*` or both.

Target Application and Level

Application

The target MDCMS application that pushed objects will be assigned to.

- ***APPL** — The application is determined by the Application field on the task that the object request is assigned to.
- ***TSKT** — The application and level are determined by the Task Type mapping definition.
- **Specific application** — Enter the ID of a defined MDCMS application, or select using content-assist.

Level

The target MDCMS application level that pushed objects will be deployed to. The level must allow checkouts unless **Level is Target of a Merge** is set to true.

If Application is set to `*APPL` and a specific level number is defined here, the Level field on the associated task is used instead. A value of 0 instructs MDCMS to use the task's level number when it is greater than 0; otherwise MDCMS retrieves the lowest level number that allows checkouts for the application. When **Level is Target of a Merge** is true, a specific level number must be provided.

Level is Target of a Merge

Set to true when the target level does not allow checkouts — such as a QA or production level. This situation typically occurs when a Pull Request has been completed, merging changes into a higher-environment branch.

User for Merge Requests

Determines which user is recorded as the requester for object requests generated by a merge.

- **Requester on Source Branch** — The user who pushed commits on the source branch.
- **User that performed Merge** — The user who completed the merge into the target branch.

What to Request

Checkout

Controls how MDCMS determines which files to retrieve from the repository.

- **Diffs** — Push only the differences (additions, changes, and deletions) by comparing the working tree before and after the commit.
- **Contents** — Push the complete contents of the configured path.

Request IFS/Remote Objects

Set to true when the path contains directories and files that will be deployed to the IFS or to a remote Linux or Windows server.

Request Source for IBMi Objects

Set to true when the path contains source files for system objects on the IBM i. Source files may target source members or IFS source files on the partition. The same CI definition can handle both IFS/Remote Objects and Source for IBM i Objects, provided relative paths and file types map correctly.

To disable the automatic triggering of a CI definition after the push of a commit, uncheck both Request parameters.

Default MD Attribute

The MDCMS attribute assigned to pushed objects when no match is found in the Attribute Mapping Table. For IFS/Remote Objects, use an attribute of type `*IFS`, `*PIPE`, or `*REMOTE`. For source objects, this serves as a final fallback when mapping, history, and MDXREF all fail to identify an attribute.

Default User

The default MDCMS user assigned to object requests when the committing Git user is not found in the User Mapping Table.

Relative Path

Defines how the relative path is calculated for object requests of MDCMS object type *IFS or *REMOTE.

- **From Repo Root** — The relative path is the full path from the repository root.
- **From CI Definition Path** — The relative path is only the folders located below the path defined in the Path parameter.
- **From Mapped Attribute Path** — The relative path is the folders below the bottom of the specific repository path for the matched attribute.

- **No Relative Path** — All files are deployed directly to the target folder for the attribute, without any relative path beneath it.

Project and Task Assignment

Project

The project to assign to the object requests. Leave blank if the project will be assigned manually before submitting the RFP. The following special values are also supported:

- ***BRANCH** — The repository branch name is parsed as the reference code. It may be an Azure Workitem, Jira Issue, ServiceNow Task, or an MDCMS Project/Task/Subtask. MDCMS parses the branch name rather than the commit message, and a trailing colon delimiter is not expected. MDCMS will automatically trim any description appearing after the Project-Task.Subtask value when the value is in `*MD` format.

If a constant is used at the beginning of the branch name, a positional value can be appended to `*BRANCH`. For example, if the branch names begin with `release-`, then specify `*BRANCH,9` as the value for the Project paramter.

If the branch naming convention doesn't consistently contain the reference code or `*MD` format, do the following: (1) Create a Custom Project/Task field of type String, up to 160 characters. (2) Add the field to the appropriate Task Types. (3) If using Azure DevOps, Jira, or ServiceNow, map the field to the MDCMS custom field. (4) Enter the branch name into the task field before committing.

- ***AZUR** — The project/task/subtask is derived from an Azure Workitem number placed at the very beginning of the commit message. The workitem must have already been exported into MDCMS.
- ***JIRA** — The project/task/subtask is derived from a Jira Issue Key placed at the very beginning of the commit message. The issue must have already been exported into MDCMS. For example, a commit message of `MDSD-3891 - demo of CI` causes MDCMS to look up issue key `MDSD-3891` and, if it maps to an MDCMS task or subtask, assign it to the object requests.
- ***KACE** — The project/task/subtask is derived from a KACE Service Desk number placed at the very beginning of the commit message. The workitem must have already been exported into MDCMS.
- ***SNOW** — The project/task/subtask is derived from a ServiceNow Task ID placed at the very beginning of the commit message. The task must have already been exported into MDCMS.
- ***LAST** — Uses the Project/Task/Subtask from the most recently installed object with the same name and attribute. Recommended when the CI definition targets a merge level (i.e., the completion of a Pull Request). Useful when higher MDCMS environments are triggered by pull requests merging into higher branches.
- ***MD** — The project/task/subtask is derived from MDCMS-formatted values placed at the very beginning of the commit message. The expected format is `PROJECT-TASK.SUBTASK:` where the colon is the end

delimiter. MDCMS extracts the project ID as everything up to the colon or the final hyphen, the task number as the value between the final hyphen and the colon or final period, and the subtask number as the value between the final period and the colon. Task and subtask numbers are generally optional.

Task

The task within the project to assign to the object requests. Leave blank when assigning directly to a project, or when the Project field is blank or uses a special value.

Subtask

The subtask to assign to the object requests. Leave blank when assigning directly to a project or task, or when the Project field is blank or uses a special value.

Object Request Automation

Generate RFP

Controls whether object requests are placed into a new RFP.

- **True** — Object requests are assigned to a new RFP, provided they are not already checked out. If an object request for the same object already exists from the same branch, it is automatically updated to the latest committed version.
- **False** — Object requests are not assigned to an RFP.

RFP Description

The description applied to the generated RFP, when Generate RFP is true.

- ***COMMITMSG** — Use the Git commit message as the RFP description.
- ***TASK** — Use the task description as the RFP description.

Auto-Request Dependencies

When requesting source for IBM i objects, enabling this option causes MDCMS to automatically request any dependencies that require recompilation — including those affected by file changes, module changes, or copybook changes. The source used for recompilation must exist in the source migration path, based-on path, or source search template on the target partition.

Auto-Create in Dev Lib

When requesting source for IBM i objects, enabling this option causes MDCMS to automatically attempt to compile the requested objects and their dependencies into the development library. If the source target is a source member, the stream file is copied to the source file of the same name in the developer's work library.

Auto-Submit RFP

When true, MDCMS auto-submits the RFP once all objects have been requested, even if the Auto-Submit parameter on the level is set to false.

Pipeline Integration

Pipeline Attribute

An attribute of type `*PIPE`. When specified, an object request is automatically generated for this attribute within the same RFP as the committed objects. This is typically used to trigger build, test, deploy, or approval jobs on pipeline servers. Use content-assist to select from defined pipeline attributes.

