AI Workflow Skills for MDChange

from Midrange Dynamics

for Version 8.6

Published October 19, 2026

Use this guide to install and run the **mdchange-request** skill from Bob, Cursor, or Visual Studio Code. The skill is for developers who already edit IBM i source in a Git repository and want an agent to create the matching MDCMS object requests, add dependencies, and compile the objects into the development library.

The skill is shipped with MDOpen for MDCMS 8.6. It follows your MDCMS authority. A request, lock, or compile that you cannot perform in MDOpen is also refused when the skill calls MDCMS.

What the skill does

For the files you select, the skill:

1. Finds an existing open object request for that file on the current branch, or creates one.
2. Sends new or changed source to MDCMS. A deleted file becomes a delete request.
3. Assigns the requests in that run to the same RFP.
4. Adds missing impacted objects to the RFP as recompile requests.
5. Optionally creates the objects in the development library.
6. Shows the creation results. For a compile error, you can open the spool file and have the agent propose a source fix.

Run the skill again after a source fix so the corrected source is sent to MDCMS and the objects are created again.

Before you start

- MDCMS **8.6.22** or later, with the MDCMS REST API server running for your environment.
- MDOpen for MDCMS 8.6 version 0.0.125 or later installed in Bob, Cursor, or VS Code, and a HTTP connection defined for the IBM i environment you will use.
- MDPipeline must be licensed on the same IBM i partitions.
- A valid MDSEC API token that you have registered in the HTTP connection in MDOpen.

© Copyright by Midrange Dynamics 2026

- The Git repository registered in MDCMS, with attribute mapping that resolves each file path and file type to an attribute. See Recommended Git Repository Structure.
- A checked-out branch whose name MDCMS can map to a project and task, or an open MDCMS project, task, and subtask you can enter yourself.
- Authority in MDCMS to request objects for the target application and level of the Git Repository.
- On Windows, Git for Windows installed in the default location so Git Bash is available at `C:\Program Files\Git\bin\bash.exe`. On macOS or Linux, `bash` must be available.

The skill does not start by itself. You invoke it by name from an Agent chat.

Put the skill in the repository

Open the Git repository in the IDE. In the Explorer, right-click a folder inside that repository, open **MDOpen**, and choose one of these commands:

Command	What it does
Copy AI Skills to default folder	Copies the skill to the folder this IDE searches for project skills. Use this command.
Copy AI Skills to this folder	Copies the skill into the folder you right-clicked, under `mdchange-request`. Use this only when you need a location other than the default. The IDE discovers the skill from the default folder.

Copy AI Skills to default folder is available when the folder is inside a Git repository. The destination depends on the IDE:

IDE	Skill folder
Bob	`.bob/skills/mdchange-request/`
Cursor	`.cursor/skills/mdchange-request/`
VS Code	`.agents/skills/mdchange-request/`

© Copyright by Midrange Dynamics 2026

The copy includes `SKILL.md` and the shell scripts the skill runs. If any of those files are already present, MDOpen lists them and asks you to confirm before it overwrites them. Confirm when you want to pick up a newer copy of the skill from MDOpen. The skill creates `mdchange.properties` in that folder the first time it runs.

Commit the skill folder so the rest of the team uses the same workflow. `mdchange.properties` stores the connection and the answers from the first run. It does not store the API token. Teams that share one application, level, and repository can commit those values. A developer who uses a different connection can keep a local copy of the properties file.

Start a new Agent chat after the copy so the IDE loads the skill.

Start the skill

Open the repository that contains the skill. Check out the branch the object requests should use. Either leave the source changes uncommitted, or attach the files to the chat before you start the skill. Attached files are the ones the skill requests. With no attachments, the skill lists the uncommitted changes and asks you which files to include.

IDE	How to start
Bob	Switch to Agent mode and tell Bob to use the `mdchange-request` skill. Bob asks before it activates a skill, unless **Auto-Approve** for Skills is turned on in Bob Settings. You can also specifically activate the `mdchange-request` in Settings/Skills.
Cursor	In Agent chat, type `/mdchange-request`.
VS Code	In GitHub Copilot agent chat, invoke `mdchange-request`.

Approve the commands the agent shows. The API token appears as `***` in those commands. Leave the token out of the chat.

Questions on the first run

The skill asks these questions only when `mdchange.properties` does not already contain a valid answer. Later runs reuse the saved values.

1. **MDOpen connection.** You pick a connection nickname from the connections saved in the IDE. Cancel stops the run and saves nothing. The skill builds the MDCMS API address from that connection and stores the host, port, HTTPS setting, and environment.

2. **Application and level.** The list contains applications and levels where checkout is allowed. The application is stored in upper case, up to 6 characters. The level is a number of up to 3 digits.
3. **Git repository.** You pick the MDCMS repository id that matches this Git repository. The id is stored in upper case, up to 10 characters.
4. **Branch mapping.** The skill asks **Map Repository Branch name to MDCMS Tasks?**
5. **Yes.** MDCMS derives the project, task, and subtask from the branch name. You then enter the starting position of the project id in that name, from 1 to 99. Use `1` when the project id is at the start of the name. Use a later position when a fixed prefix comes first. A branch that starts with `release-` uses position `9`. The same rules as Project and Task Assignment for `*BRANCH` apply.
6. **No.** You enter the MDCMS project (upper case, up to 12 characters), task (1 to 7 digits, greater than 0), and subtask (1 to 7 digits, or 0 when there is no subtask). The skill checks that the task exists and is open. A closed task, or a task MDCMS does not accept, returns you to these three fields.

To change a saved answer, edit or delete that line in `mdchange.properties` and run the skill again. Deleting a value makes the skill ask for it.

What happens for each file

Git status	MDCMS request
New or changed file	Modify request. The local file is sent as the source.
Deleted file	Delete request. The file is not compiled.

When an open request for that file and branch already belongs to you, the skill updates the source on that request.

When open requests exist and none of them belong to you, the skill shows the first request and asks you to choose:

Choice	Result
Skip Object	Leave that file unchanged and continue with the next file.
Update Request for other User	Update the source on that existing request.
Create New Request	Create a separate request for you.

© Copyright by Midrange Dynamics 2026

Choice	Result
Abort	Stop the run.

A new request is created with the object locked. If another request already locks the object, you can retry, retry without the lock, ignore the file, or abort.

The first request that succeeds supplies the RFP number. Every later file in the same run is assigned to that RFP.

If a request fails for another reason, you can retry, ignore that file, or abort.

Dependencies and object creation

After the files have been requested, the skill adds dependencies for the RFP when at least one modify request succeeded. Existing dependency requests on the RFP are left in place. A failure at this step stops the run.

The skill then asks **Should the objects be created for new/changed requests in the RFP?**

- **No** ends the run. The object requests and dependencies remain on the RFP.
- **Yes** submits creation to batch, starting at the first object that has not been created, and stops when an error occurs. The skill waits until the job finishes and then shows one row per object:

Column	Meaning
Object Name	Object MDCMS created or attempted
Attribute	MDCMS attribute
Status	Creation result, such as success, error, or ignored
Spoolfile	Whether a spool file is available
Git Path	Repository folder for the source
Git File	Source file, linked so you can open it in the editor
Ignore Reason	Why an object was ignored, when MDCMS provides one

© Copyright by Midrange Dynamics 2026

For an object that has a spool file, you can view the spool file, go to the next object, skip ahead to the next object in error, or finish. When the status is an error and the row has a Git path and file, you can also have the agent try to correct that source from the errors in the spool file. Review the proposed change, then run the skill again to send the corrected source and create the objects.

A creation job that ends with a timeout or an abnormal end can be submitted again or cancelled.

Saved settings

`mdchange.properties` sits in the skill folder. The names below are the lines the skill writes.

Setting	Purpose
`MDCMS_API_URL`	MDCMS REST API address for the selected connection
`MDCMS_CONNECTION_HOST`	Host from the MDOpen connection
`MDCMS_CONNECTION_SECURE`	Whether the connection uses HTTPS
`MDCMS_CONNECTION_PORT`	REST API port, blank when the connection uses the default port
`MDCMS_CONNECTION_ENVIRONMENT`	MDCMS environment suffix, blank for the default instance
`APPLICATION`	Application code
`LEVEL`	Promotion level
`REPOSITORY_ID`	MDCMS Git repository id
`MAP_BRANCH_TO_TASK`	`true` to derive the task from the branch name, `false` to use the project and task below
`PROJECT_STARTING_POSITION`	Character position where the project id starts in the branch name
`MDCMS_PROJECT`	Project, used when branch mapping is off

© Copyright by Midrange Dynamics 2026

Setting	Purpose
`MDCMS_TASK`	Task, used when branch mapping is off
`MDCMS_SUBTASK`	Subtask, used when branch mapping is off

The API token stays in MDOpen. The skill reads it for the connection above and does not write it into this file or into the chat.

Troubleshooting

What you see	What to do
The skill is not offered in chat	Confirm the skill folder is the default folder for that IDE, then start a new Agent chat. Invoke it by name.
Copy to the default folder is cancelled because the IDE is not supported	Bob, Cursor, and VS Code are the supported IDEs. On another editor, use **Copy AI Skills to this folder** and place `mdchange-request` where that product reads project skills.
The skill tells you to check the MCP port values	In MDOpen Settings, confirm **MDOpen MCP Server Port minimum** and **maximum**. The default range is 37821 through 37830, and the maximum can be at most 10 above the minimum. Restart the IDE window so MDOpen

© Copyright by Midrange Dynamics 2026

What you see	What to do
	can register the `mdopen-86` MCP server again.
The agent cannot call MDOpen tools, or `mdopen-86` does not appear	Confirm the MCP server is running in the IDE you are using. See Viewing/Restarting MCP Servers.
The skill tells you to configure the token and user	Sign in to the MDOpen connection. MDOpen stores the API token and user the skill reads. A connection with no stored token cannot be used.
Git Bash or a script cannot be started on Windows	Install Git for Windows so `bash.exe` is at `C:\Program Files\Git\bin\bash.exe`.
The application or level you need is missing	The list includes only levels where checkout is allowed. Confirm your authority and the level settings in MDCMS.
The task is closed	Enter an open project, task, and subtask. This applies when branch mapping is off.
HTTP status above 200	MDCMS refused the call. Read the message in the response. Typical causes are authority, a missing

© Copyright by Midrange Dynamics 2026

What you see	What to do
	attribute mapping for the file path, a repository id that does not match this repo, or the REST API server being unavailable. See MDCMS/MDOPEN Connectivity Troubleshooting.
Target object already locked by another request	Retry, retry without locking, ignore that file, or abort.
Creation stays pending	The skill polls the batch job until it completes. A `jobq-timeout` or `abnormal-jobend` status can be submitted again.

Viewing/Restarting MCP Servers

MDOpen registers an MCP server named `mdopen-86` when the extension starts. Each IDE window runs its own server, on a port from the range in MDOpen Settings. The skill calls MDCMS through that server. Confirm it is running, then start a new Agent chat.

IDE	How to
Bob	In the Bob chat panel, open the gear icon and select the **MCP** tab. **Use MCP Servers** must be checked. `mdopen-86` is running when it is listed and shows a connected status. Expand the server to see its tools. Restart the server from that list when the status is not connected.
Cursor	Open **Customize** in the sidebar and find `mdopen-86`. The server is running when it is enabled and shows a connected status. On an older Cursor release, the same list is under Cursor Settings, **Tools & MCP**, where a green status means connected. Open the Output panel and select **MCP Logs** to confirm the server started. A connection error or

© Copyright by Midrange Dynamics 2026

IDE	How to
	crash is written there. Toggle the server off and on if it stays disconnected, then start a new Agent chat.
VS Code	Open the Command Palette and run **MCP: List Servers**. `mdopen-86` is running when a green indicator is shown next to its name. If the server is stopped, select it and choose **Start Server**. If the status is an error, select **Show Output**, then **Restart Server** after the problem is corrected. In GitHub Copilot agent chat, select **Configure Tools** and confirm the `mdopen-86` tools are listed.

© Copyright by Midrange Dynamics 2026