

Working with Cockpit projects

PN307 | Posted on August 6, 2026 | Updated on August 6, 2026



Stéphane LOVEJOY

Senior Software Developer

imperix • in

Table of Contents

- [Project management basics](#)
- [Opening and closing projects](#)
 - [Opening an existing project](#)
 - [Closing a project](#)
- [Exporting and importing projects](#)
 - [Exporting projects](#)
 - [Importing projects](#)

When testing various control algorithms or migrating across different converter topologies, the Cockpit interface can quickly become cluttered. Effective project management is essential to maintain an organized workspace, ensure configuration data is safely backed up, and streamline the transition between different hardware setups. This article explains the role of the project file (.ixproj) and details how to effectively handle, store, and transfer complete project environments within the software.

Before reading this guide, it is recommended to consult [Programming and operating imperix controllers](#) to understand how projects are initially created after the code generation process, and the [Cockpit – User guide](#) for the fundamental concepts of the imperix monitoring software

The use of the project files (.ixproj) and the associated project management features are available starting from ACG SDK and CPP SDK version 2026.2.

Project management basics

Cockpit relies on a dedicated project file (.ixproj) to safely store all essential data required to restore a specific working environment. The file is continuously autosaved in the background and contains:

- **Module configurations:** The position and settings of every module in the workspace. For example, it saves the arrangement of the modules on the screen, which user variables are plotted in the Scope module, and the design of the [GUI Builder module](#).
- **Project settings:** The project name, the link to the specific imperix controller where the code is executed, the path to the generated user code (.elf), the path to the snapshot folder, and any offline user variables.

The project file (.ixproj) does not contain the actual Simulink/PLECS model or the compiled user code (.elf). It only stores the file paths pointing to them.

Prior to **SDK version 2026.2**, project files (.ixproj) did not exist. When upgrading from an older version of the SDK, a .ixproj file is automatically generated for every existing project in Cockpit. By default, this new project file is saved in the same directory as the associated user model (Simulink or PLECS).

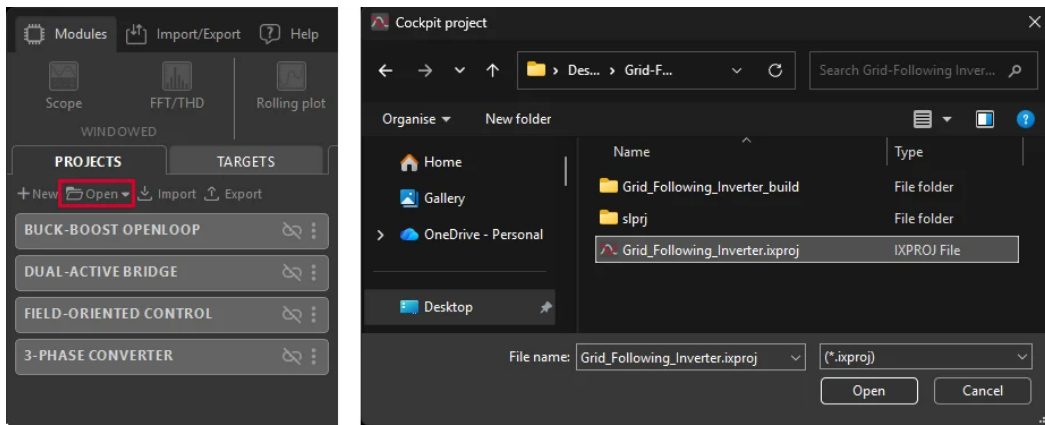
Opening and closing projects

To maintain an organized workspace and manage multiple projects, Cockpit allows existing projects to be easily loaded, or those not currently in use to be closed.

Opening an existing project

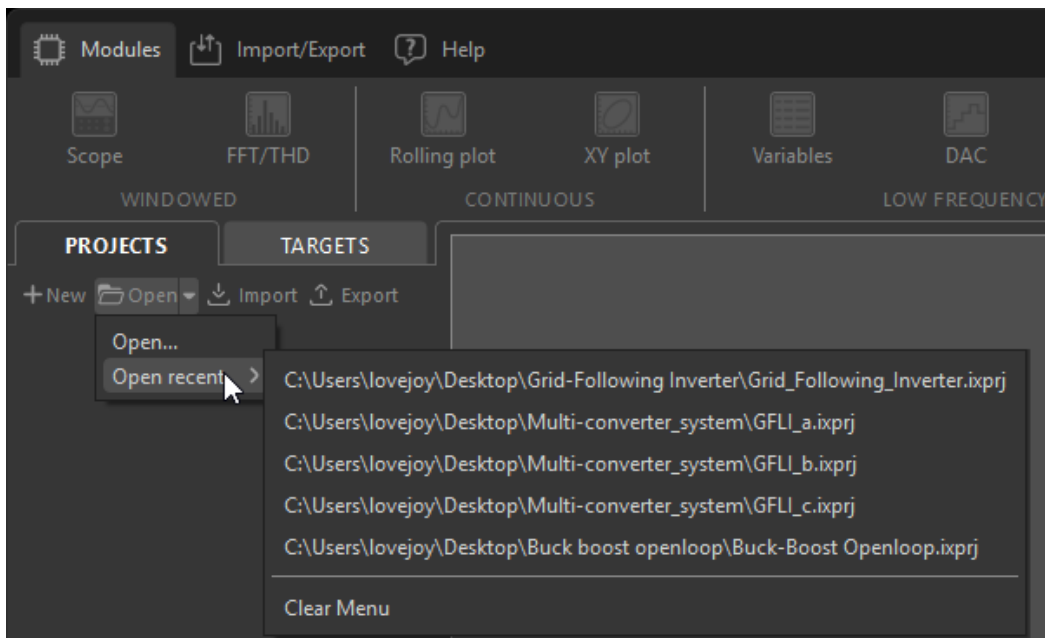
If an existing project needs to be accessed (such as one that was recently closed or developed for a different application), it can be loaded into the Projects perspective using one of the following methods:

- **Using the file browser:** The *Open* button located at the top left of the left bar can be clicked. This launches a file dialog window where the appropriate folder and corresponding .ixproj file can be selected to load the configuration into Cockpit.



Open an existing project from Cockpit

- **Using the recent files list:** For quicker access to recently closed items, the dropdown arrow immediately to the right of the *Open* button can be expanded. Hovering over *Open recent* displays a list of recently closed files, allowing them to be reloaded with a single click.

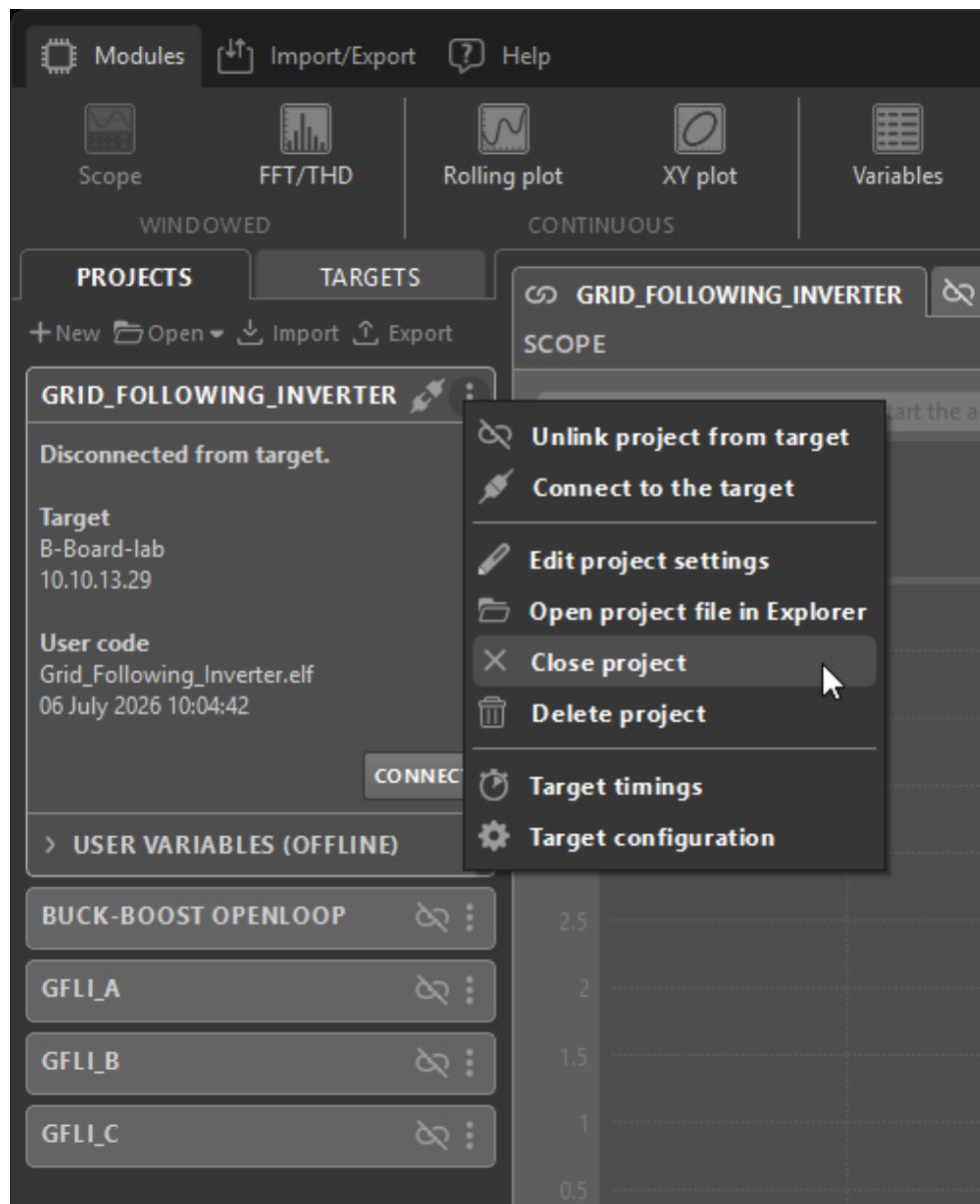


Open a recently opened project from Cockpit

Closing a project

When testing various algorithms or working with multiple setups, the left bar can quickly become crowded with unused project panes. To keep the interface clean and focused, projects that are not actively running can be closed.

To close a project, the three-dot menu (⋮) located in the top right corner of the project pane must be clicked, and *Close project* selected from the dropdown list.



Closing a project from Cockpit

It is important to distinguish between closing and deleting a project within Cockpit:

- **Closing** simply removes the project pane from the left bar. The entire configuration is safely preserved within the .ixproj file, and it can be reopened at any time without losing any settings.
- **Deleting** permanently erases the .ixproj file, meaning the specific Cockpit configuration for that project is lost forever.

Note that whether a project is closed or deleted, external assets or user models (such as Simulink/PLECS models, snapshot folders, and the generated user code .elf file) are never deleted.

Exporting and importing projects

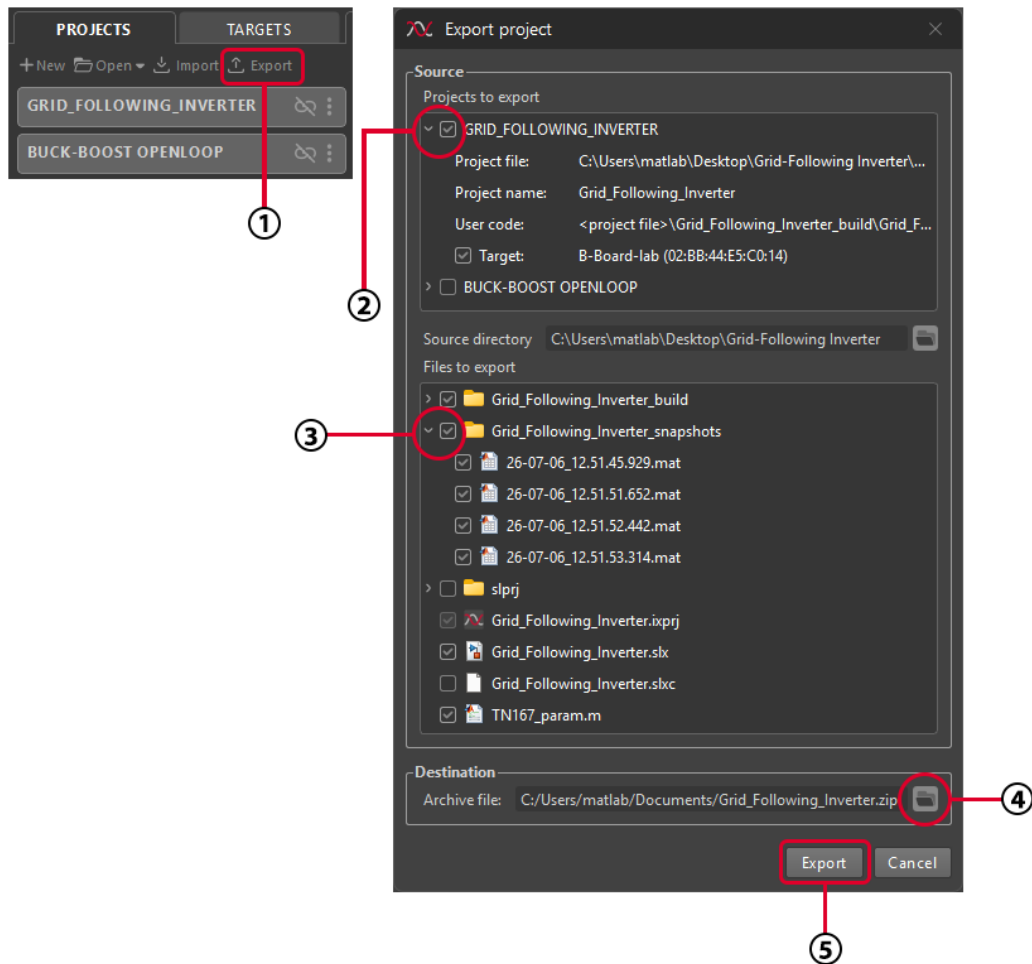
Moving configurations between different computers, backing them up, or sharing them typically requires more than just the .ixproj file. It is usually necessary to also export the user models (Simulink or PLECS), additional scripts, and possibly test results or snapshots. The export and import features allow complete project packages to be easily bundled and transferred, ensuring all necessary dependencies are included.

Exporting projects

The export feature allows one or more projects, along with their associated user models and assets, to be bundled into a single .zip archive.

To export a project package:

1. Click on the *Export* button located in the Projects tab of the left bar. This opens the **Export project** window.
2. In the **Source** section, under **Projects to export**, the box next to the project(s) to be included should be checked. Expanding a project item will display its configuration details, such as the linked target and user code path. Choose whether to maintain the original link to the target by checking the **Target** box.
3. Once an item is selected, the **Files to export** section automatically populates. Cockpit detects and selects essential files by default (such as the .ixproj file, the _build folder, and linked Simulink/PLECS models). Individual folders, such as MATLAB scripts (.m), [Vivado projects](#), or snapshots, can be manually checked or unchecked to customize the export.
4. In the **Destination** section, clicking the folder icon next to **Archive file** allows the location and name of the resulting .zip file to be chosen.
5. Finally, clicking *Export* generates the zip package.



Export project procedure

If several projects are selected, as is typically the case in a multi-master setup, all selected items will be exported into a single, unified .zip archive. For this reason, it is highly recommended that all associated projects share a common *Source directory* before exporting.

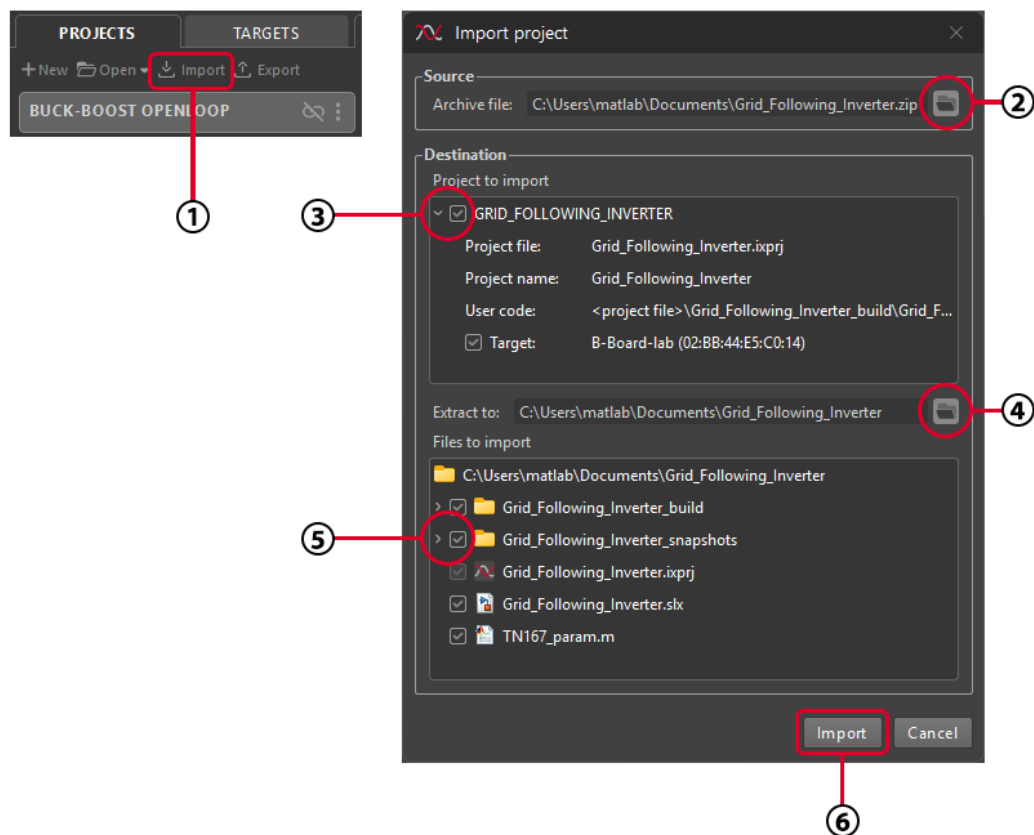
Importing projects

The import projects feature allows a previously exported .zip archive to be loaded, seamlessly unpacking the files and restoring the full project configuration into Cockpit.

To import a project package:

1. The *Import* button located in the Projects tab of the left bar must be clicked. This opens the **Import project** window.
2. In the **Source** section, clicking the folder icon next to **Archive file** allows the exported .zip package to be selected.
3. The **Destination** section will populate with the contents of the archive. Under **Project to import**, the projects to be loaded from the archive should be

- selected. The project item can be expanded to review its details and choose whether to maintain the original link to the target by checking the **Target** box.
4. The folder icon next to **Extract to** allows the extraction directory to be chosen.
 5. Under **Files to import**, the folders and files that will be extracted can be verified. Individual folders, such as MATLAB scripts (.m), [Vivado projects](#), or snapshots, can be manually unchecked to customize the import.
 6. Clicking **Import** extracts the files and loads the project(s) in the left bar of Cockpit.



Import project procedure