



STM32CubeIDE quick start guide

Introduction

This document is a brief guide for users to get started quickly with STMicroelectronics [STM32CubeIDE](#).

STM32CubeIDE is an integrated development environment (IDE) based on the Eclipse® framework. It is aimed at users developing embedded software in C/C++ for STMicroelectronics STM32 MCUs and MPUs.

This manual provides basic information about:

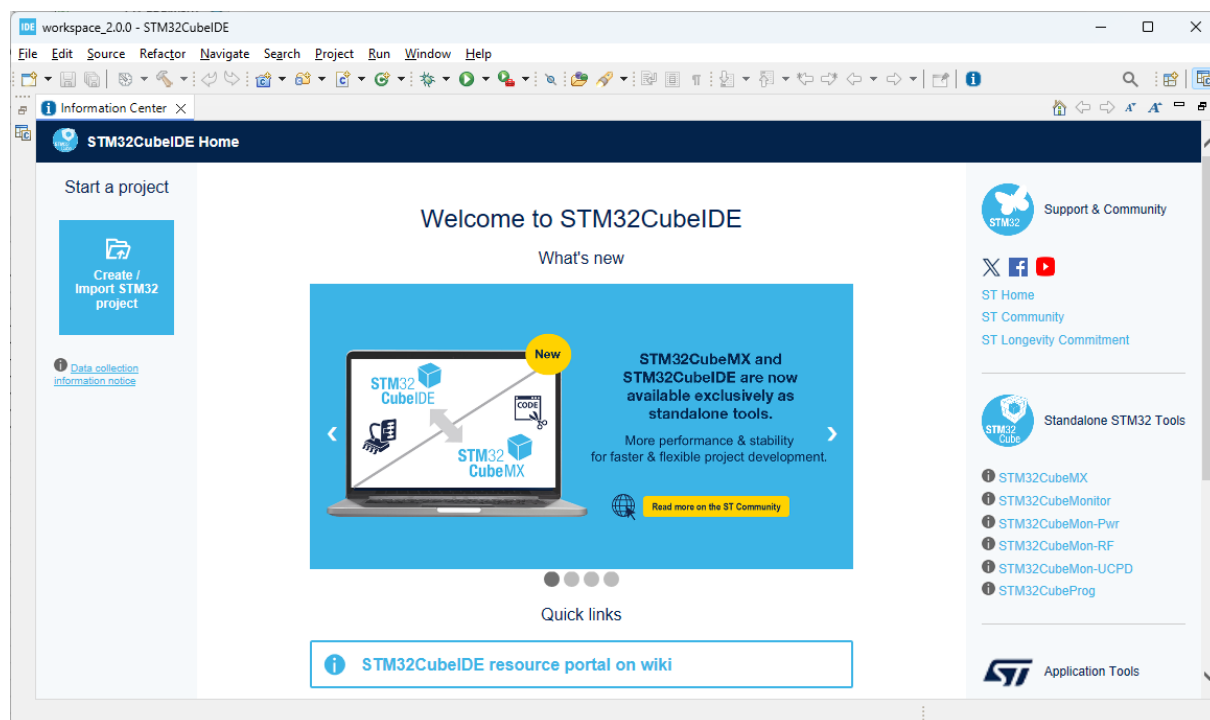
- [Information Center](#)
- [Workspace and projects](#)
- [Project information](#)
- [Debugging](#)



1 Information Center

When STM32CubeIDE is started the first time, the *Information Center* opens.

Figure 1. STM32CubeIDE with *Information Center*



The *Information Center* enables the user to quickly reach information about the product and how to use it. By clicking on the corresponding hypertext links, manuals are opened and latest information is obtained from the STMicroelectronics website at www.st.com.

It is not required to read all material before using STM32CubeIDE for the first time. Rather, it is recommended to consider the *Information Center* as a collection of reference information to return to.

The *Information Center* window can be reached at any time via the [Help]>[Information Center] command. When the *Information Center* tab is closed, the standard views in the C/C++ perspective are opened.

STM32CubeIDE supports STM32 32-bit products based on the Arm® Cortex® processor.

arm

Note: Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

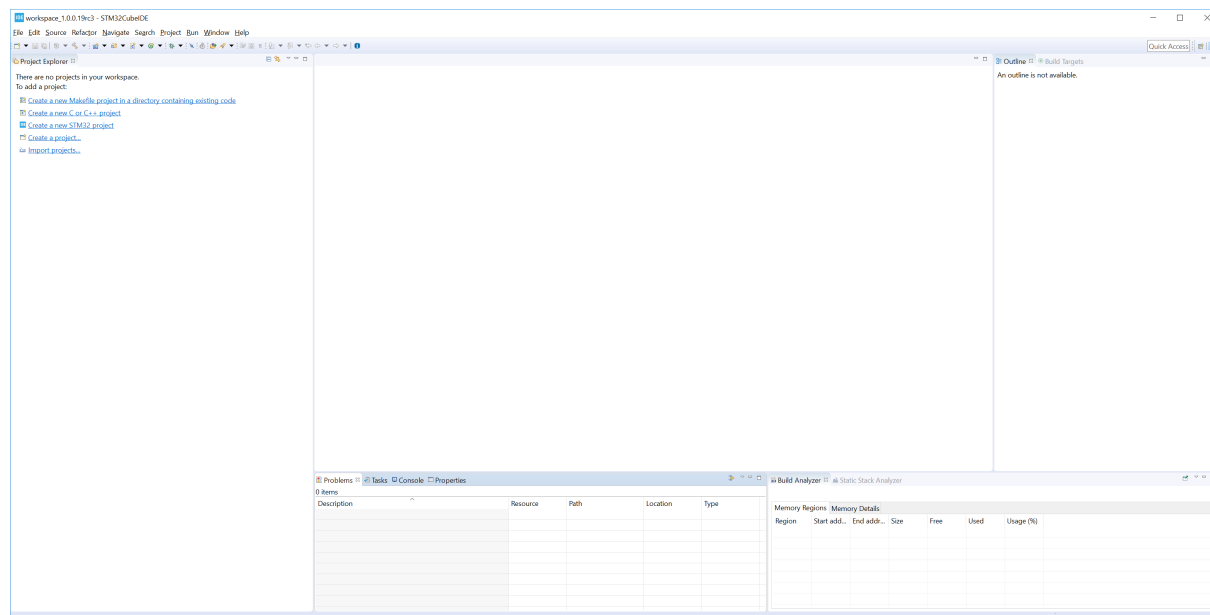
The Arm word and logo are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.

2 Workspace and projects

STM32CubeIDE is built using the Eclipse® framework; it inherits characteristics that might be unfamiliar to new users.

The Eclipse® editing environment uses perspectives. A *perspective* is a set of windows dedicated to a purpose. The C/C++ *perspective* is dedicated to writing and editing code, and navigating through projects. Users spend a lot of time in this perspective.

Figure 2. STM32CubeIDE with C/C++ perspective open, no project defined yet

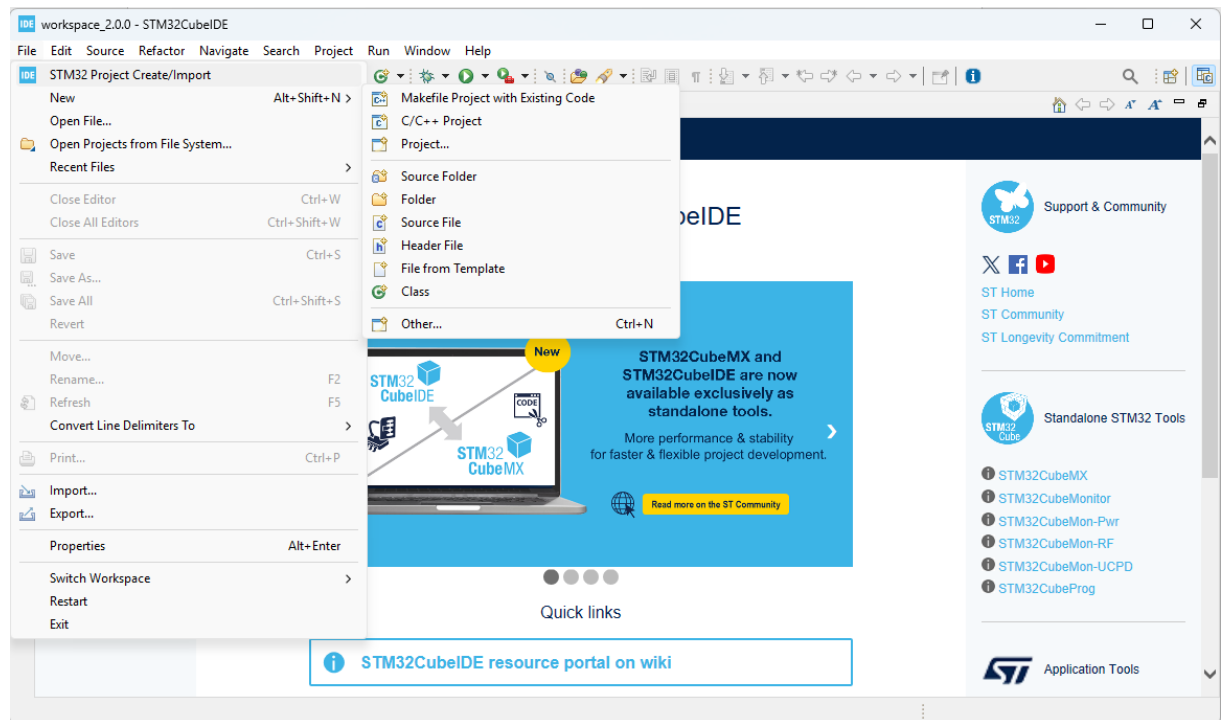


Another popular perspective is the *Debug perspective*, where users debug and test. Switching back and forth between perspectives is easy, and can even be automatic in some cases.

Workspace and projects

The next thing to do is create a project, which requires to introduce firstly the concept of workspace. A workspace is a container that includes project folders or information about project folders, and a `.metadata` folder that contains information about the projects. A workspace is simply a folder on a hard drive, which can be located anywhere in the storage media. When STM32CubeIDE starts up, it asks which workspace must be used. This may be changed at any time by selecting [File]>[Switch Workspace] and navigating to another folder.

Figure 3. STM32 project creation



Note: Workspace and project names must contain only ascii characters. This is also valid for the path to the workspace.

2.1 Project types in STM32CubeIDE

STM32CubeIDE enables users to:

- Create new projects from scratch for STM32 microcontrollers like empty projects or Empty CMake projects. This process allows full customization of the project configuration, including device selection, firmware package management, and peripheral initialization.
- Import existing projects, including those generated by STM32CubeMX or STM32CubeMX2, as well as projects using Makefile or CMake-based build systems.

2.2 Start from scratch with STM32CubeIDE

STM32CubeIDE allows users to create empty projects and empty CMake projects.

Start from scratch - STM32CubeIDE empty project

To create an empty project, proceed as follows:

1. Select the [File]> [STM32 Project Create/Import]> [Create New STM32 Project]] command. The pop-up window is illustrated in Figure 4.
2. Select STM32CubeIDE project from the *Create New STM32 Project* menu.
3. Follow the wizard displayed to select MCU and enter the required data: project name, *Targeted Language* and *Targeted Binary Type*. This is illustrated in Figure 5.

4. Click on [Finish]. Illustrated in Figure 6

Figure 4. Create new STM32CubeIDE project wizard

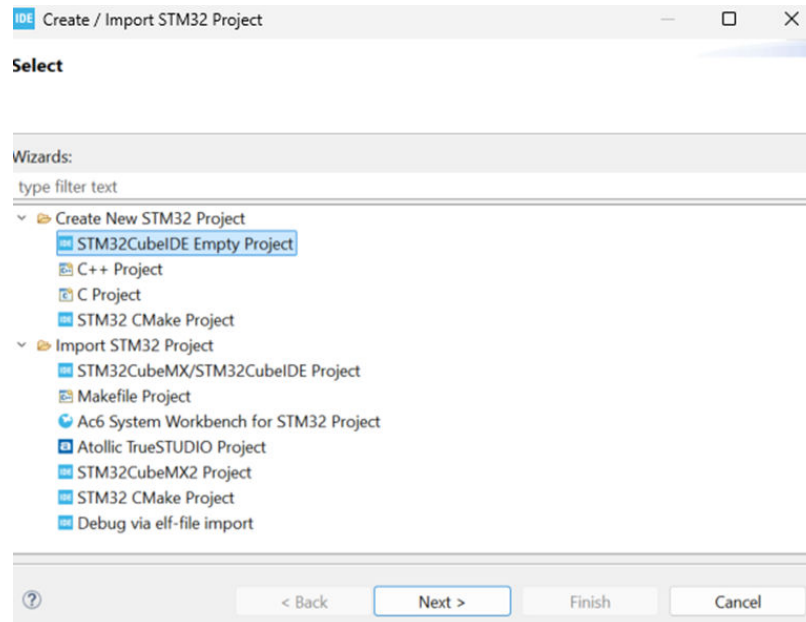


Figure 5. STM32CubeIDE MCU selection wizard

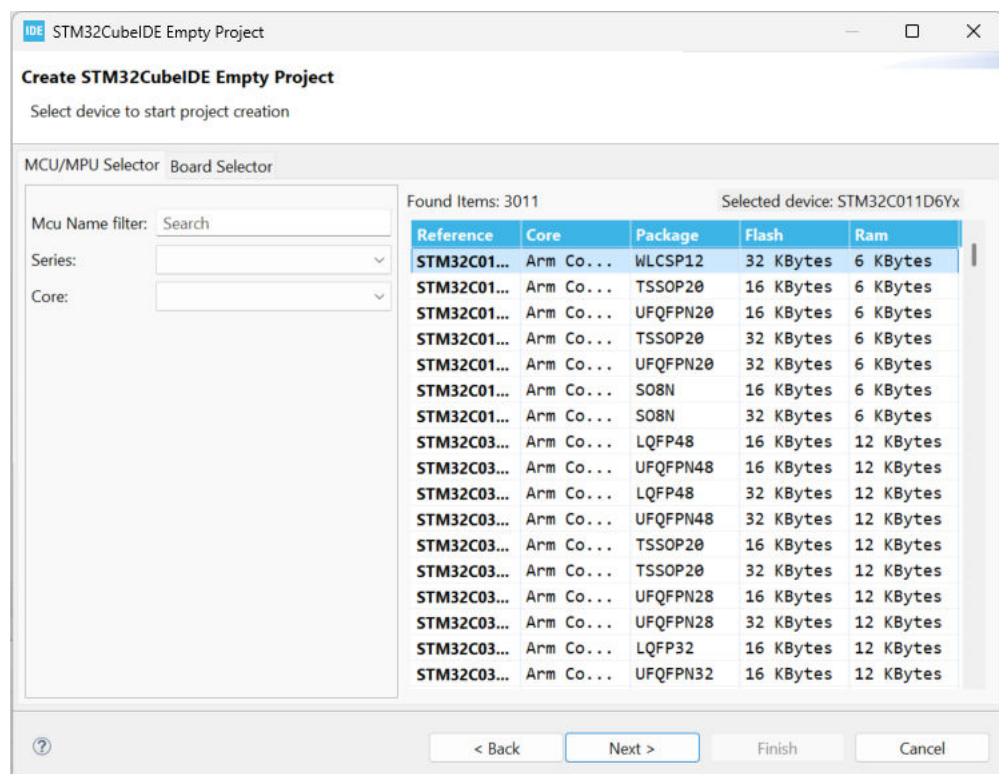
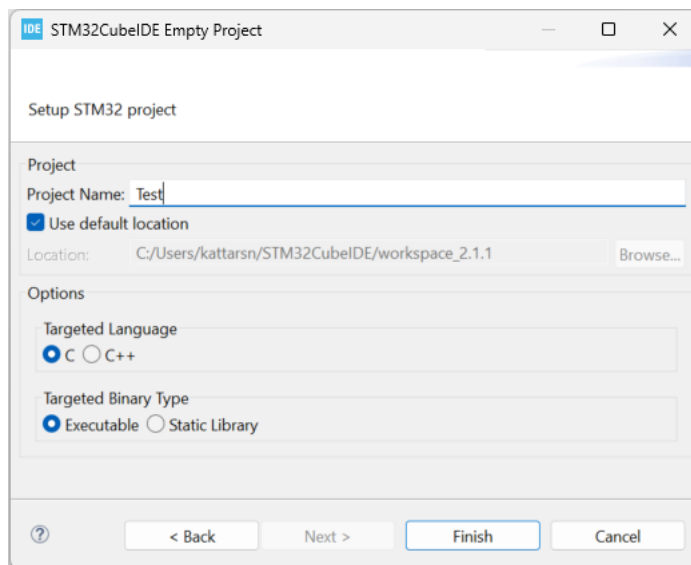


Figure 6. STM32CubeIDE empty project wizard



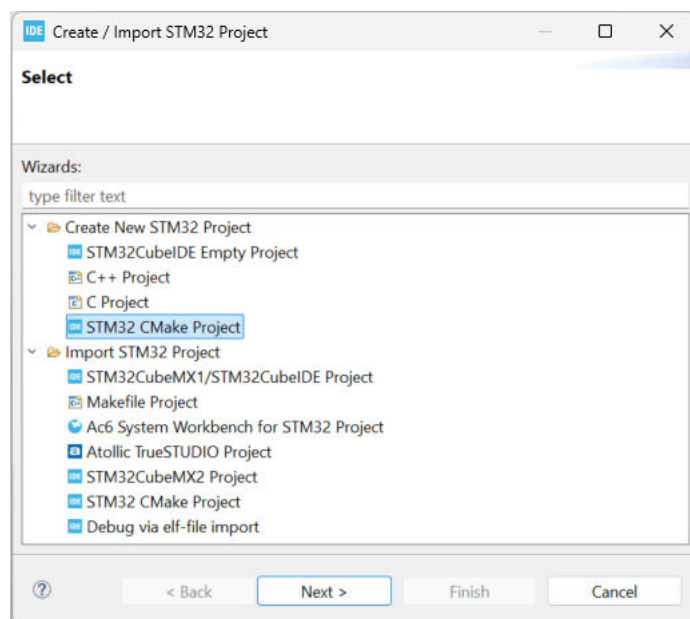
DT78322V1

Start from scratch – STM32CubeIDE empty CMake project creation

To create an empty CMake project, proceed as follows:

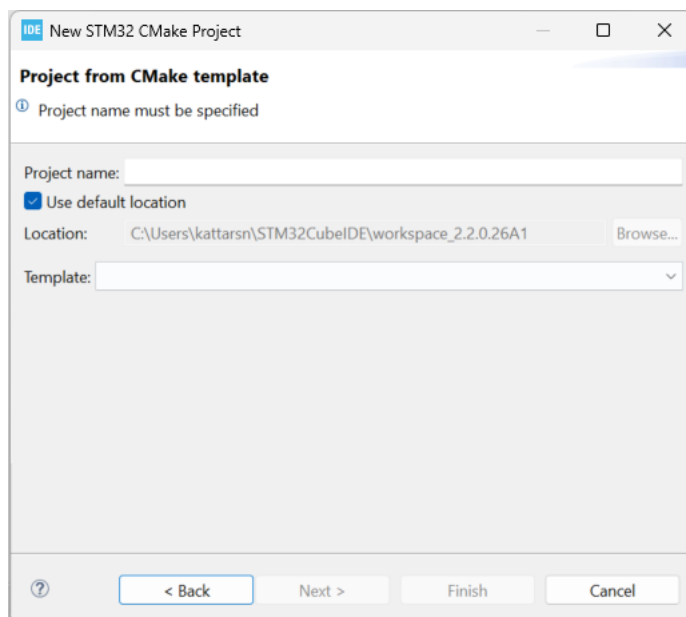
1. Select the **[File]> [STM32 Project Create/Import]>[Create New STM32 Project]** command.
2. Select STM32 CMake project as illustrated in Figure 7
3. Follow the wizard displayed to enter the following information as illustrated in Figure 8:
 - Project name
 - Select template such as *Executable* or *Static Library*
 - Select the required MCU
4. Click on [Finish]. Illustrated in Figure 9

Figure 7. Create new STM32 CMake project wizard



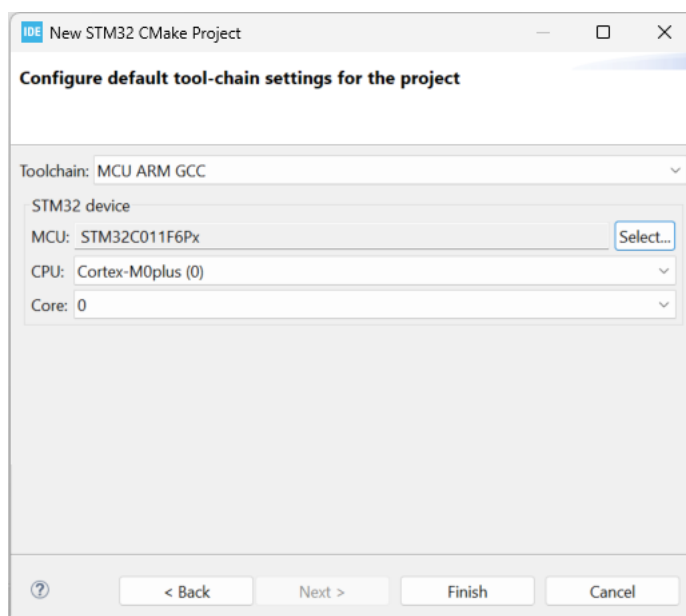
DT78323V1

Figure 8. New STM32 CMake project wizard



DT78324V1

Figure 9. CMake MCU selection wizard



DT78325V1

2.3 Start from existing project with STM32CubeIDE

STM32CubeIDE allows users to import projects generated by STM32CubeMX or STM32CubeMX2.

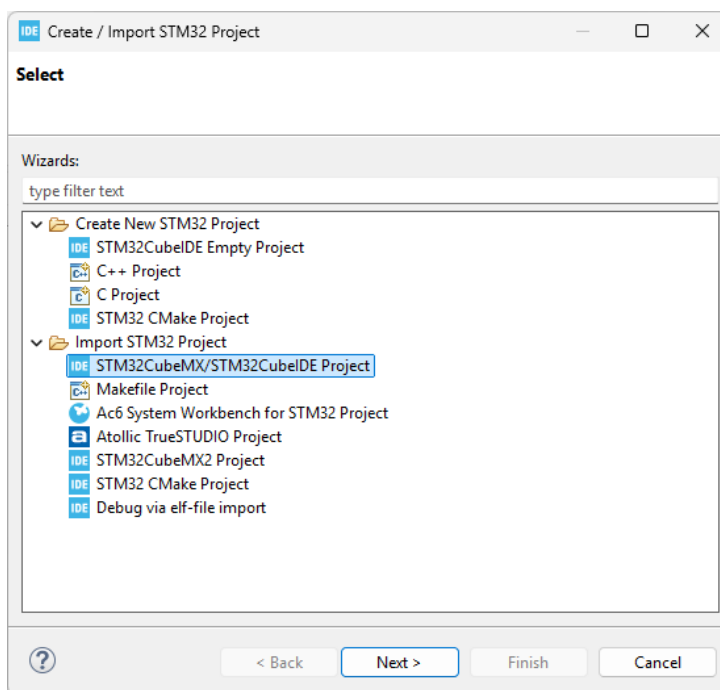
Start from existing project: Import STM32CubeMX or STM32CubeMX2 project

To import an STM32CubeMX project use the STM32 project wizard and follow the steps below:

1. Select the **[File] > [STM32 Project Create/Import] > [Import STM32 Project]** menu command. The wizard is illustrated in Figure 10
2. Select STM32CubeMX or STM32CubeMX2 project

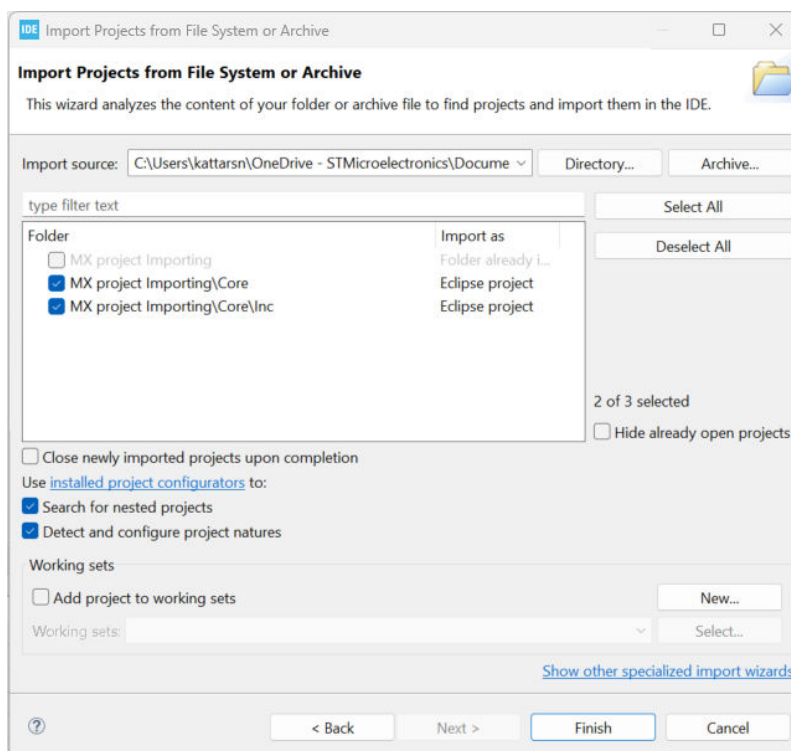
3. Follow the wizard instructions displayed to enter the requested data:
 - Project name
 - STM32CubeMX directory location. Illustrated in Figure 11.
4. Click on [Finish].

Figure 10. Import existing STM32CubeMX project wizard



DT78326V2

Figure 11. Import project from file system



DT78327V1

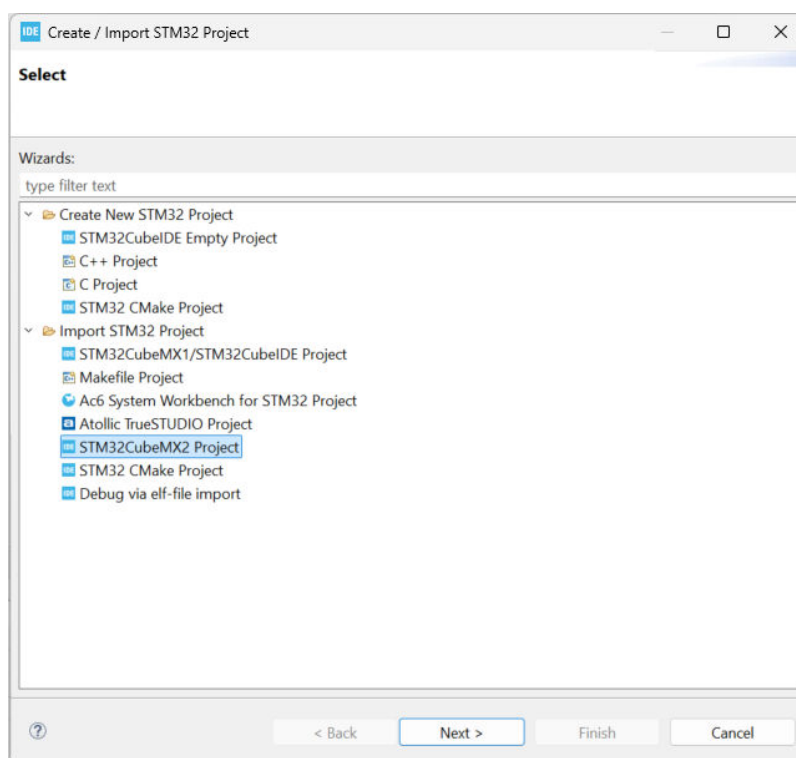
New projects can also be created through **[File]>[Import STM32 Project]> STM32CubeMX/STM32CubeIDE]** menu command.

Start from existing project: Import STM32CubeMX2 or STM32CubeIDE project

To import an STM32CubeMX2 project use the STM32 project wizard and follow the steps below:

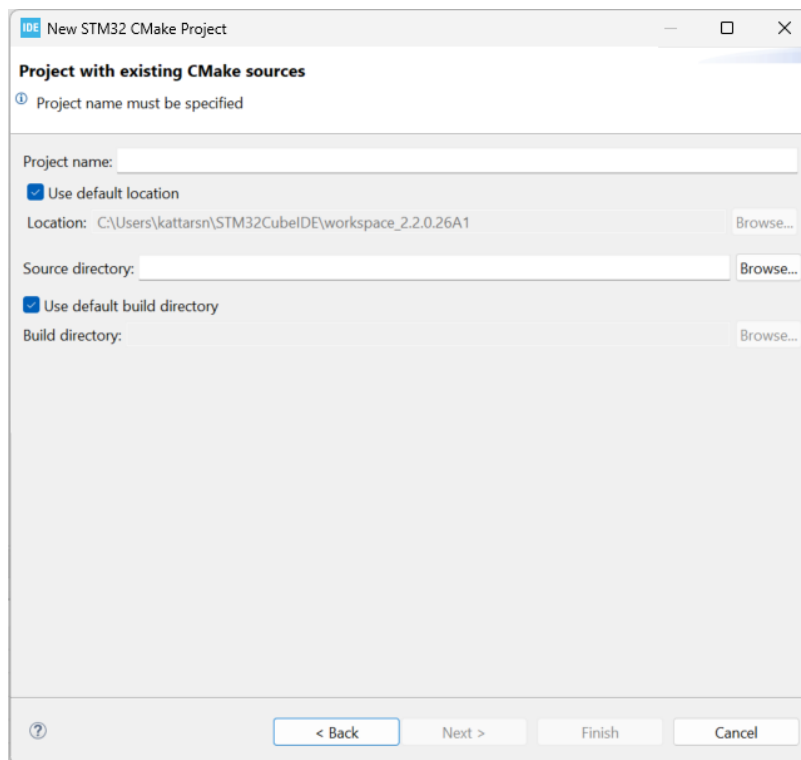
1. Select the **[File]> [STM32 Project Create/Import]> [Import STM32 Project]]** menu command. Illustrated in [Figure 12](#)
2. Select STM32CubeMX2 project
3. Follow the wizard displayed to enter the requested data, as illustrated in [Figure 13](#):
 - Project name
 - STM32CubeMX2 directory.
4. Click on [Finish].

Figure 12. Import STM32CubeMX2 project wizard



DT78328V1

Figure 13. Project used with existing CMake source wizard



DT78329V1

New projects can be created as well through **[File]>[Import STM32 Project]> STM32CubeMX2]** project menu command.

Project importing can also be initiated through the added icon in the toolbar illustrated in Figure 14.

Figure 14. Project import icon



DT78330V1

Please refer to the following links for an initiation with STM32CubeIDE:

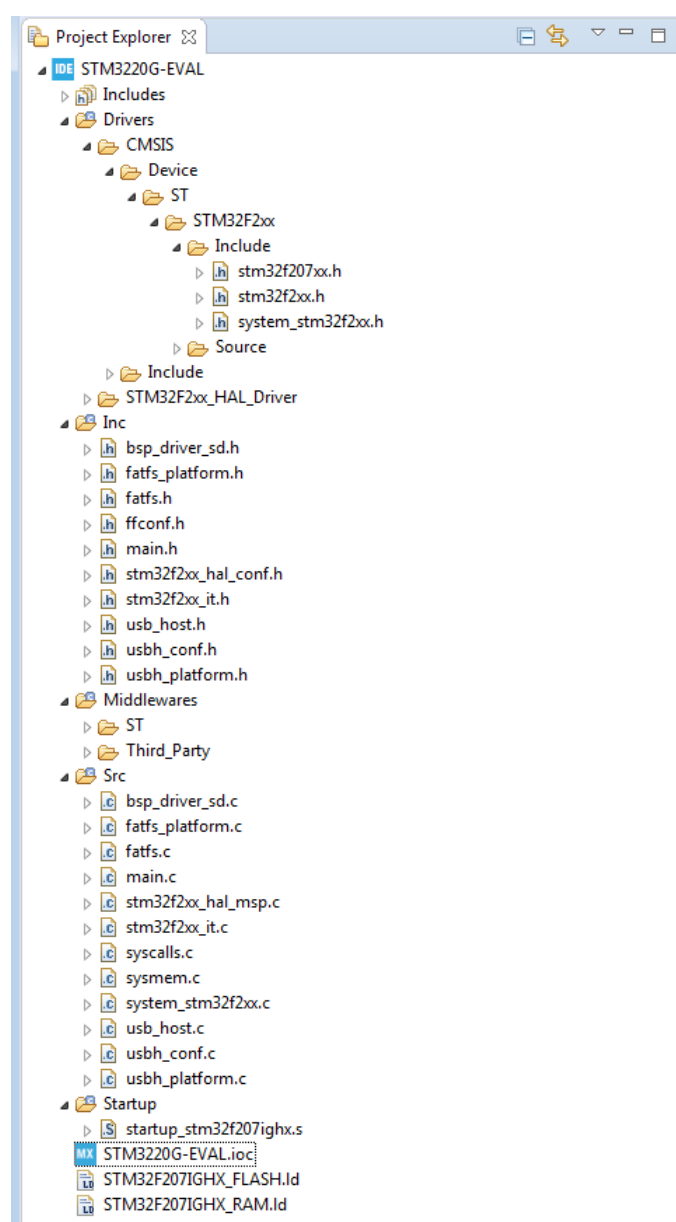
- [Getting Started with STM32Cube Ecosystem](#)
- [Introduction to STM32CubeIDE](#)

3 Project information

Once the information is entered in the wizard, it is possible to work in the project itself:

- The *Project Explorer* docking view is populated with the selected project. In the *Project Explorer* window, the subfolders can be expanded to any desired sublevel.
- The kind of project created by the *Project Wizard* depends on the device family. In any case, the project contains all the necessary elements of a complete embedded application, ready to program into the memory of the part and run.
- When several projects exist in a workspace, it is possible to close some of them. The actions controlled by the icons such as run, or launch debug session then apply to the open projects.
- Projects can be closed or opened by first right-clicking on the project name in the *Project Explorer* docking view and then selecting **[Open Project]** or **[Close Project]**.

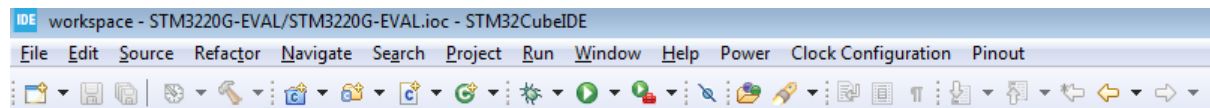
Figure 15. Project Explorer window populated with a project



Controls

All control icons have tool tips activated by hovering the mouse over the icon. The icons are shortcuts for the same functions that can be activated from the main menu bar. These icons control specific functions that pertain to code editing, building, and project management; they are unique to the *C/C++ perspective*.

Figure 16. Main control icons for editing, building, and project management



Use this icon to create new C source code module, header files, or new object such as project, library, or repository ([**File**]>[**New**] from the main menu options).



Use this icon to build your project.



Use this icon to launch a specific debug configuration, or configure a debug configuration by clicking on the arrow (Functions can be activated in the [**Run**] option in the main menu).



The flashlight icon launches various search utilities; the arrows navigate among recently visited places in the project ([**Search**] and [**Navigate**] in the main menu options).

4 Debugging

Once the project is built without errors, use the  icon to launch the program code into the part together with the debugger. The first time a debug session is launched, [STM32CubeIDE](#) builds the project and then shows the debug launch configuration menu. This gives the user the opportunity to verify settings and make changes if desired.

Important: *If the Serial Wire Viewer (SWV) is to be used during debugging, this must be explicitly enabled in the dialogue box:*

1. Click on the **[Debugger]** tab to reach the SWV settings and enable SWV (SWV is not enabled by default).
2. If changes are made to the default settings in the launch configuration properties dialog box, they must be saved by clicking on the **[Apply]** button.

When ready to launch the debug session, click on **[OK]**. Several things happen “under the hood” including the launch of the debugger driver and GDB server, as well as the programming of the application into the part, and any other actions called for in the startup script. At this point, the application is normally halted at the first line of `main()`.

The main control icons of the debugger are as follows:

	Resume full speed execution of the application on the target (greyed out when running).
	Halt execution (greyed out when halted).
	Step into a function, over a function, or out of a function.
	Toggle between C and instruction stepping.
	Reset the chip and restart execution.
	Terminate the debug session.
	Terminate and relaunch the debug session.

Basic debugging techniques

- To set a breakpoint, click on the blue horizontal bar next to the statement line number.
- To view memory locations, use the *Variables* (variables in focus or global variables), *Memory*, or *Expressions* window.

Configuring and using debugging tools, especially with SWV, is frequently more complicated than using the IDE for C/C++ code development. More details on this topic are provided by associated documents available on the [STM32CubeIDE](#) web page, which must be consulted for anything beyond the most basic operations.

Revision history

Table 1. Document revision history

Date	Revision	Changes
18-Apr-2019	1	Initial version.
02-Nov-2020	2	Updated window screenshots and icons.
09-Nov-2021	3	Updated the <i>Information Center</i> screenshot.
06-Nov-2025	4	Updated the <i>Information Center</i> screenshot (<i>Figure 1</i>). Updated the new project creation: - Added <i>Figure 3. STM32 project creation</i> - Updated <i>Figure 4. New project wizard</i>
18-Feb-2026	5	Updated <i>Figure 4. New project wizard</i> .
22-Jun-2026	6	Added: Section 2.1: Project types in STM32CubeIDE Section 2.2: Start from scratch with STM32CubeIDE Section 2.3: Start from existing project with STM32CubeIDE Updated: Section 2: Workspace and projects

Contents

1	Information Center.....	2
2	Workspace and projects	3
2.1	Project types in STM32CubeIDE.....	4
2.2	Start from scratch with STM32CubeIDE.....	4
2.3	Start from existing project with STM32CubeIDE.....	7
3	Project information	11
4	Debugging	13
	Revision history	14
	List of figures.....	16

List of figures

Figure 1.	STM32CubeIDE with <i>Information Center</i>	2
Figure 2.	STM32CubeIDE with <i>C/C++ perspective</i> open, no project defined yet	3
Figure 3.	STM32 project creation	4
Figure 4.	Create new STM32CubeIDE project wizard	5
Figure 5.	STM32CubeIDE MCU selection wizard	5
Figure 6.	STM32CubeIDE empty project wizard	6
Figure 7.	Create new STM32 CMake project wizard	6
Figure 8.	New STM32 CMake project wizard	7
Figure 9.	CMake MCU selection wizard	7
Figure 10.	Import existing STM32CubeMX project wizard	8
Figure 11.	Import project from file system	8
Figure 12.	Import STM32CubeMX2 project wizard	9
Figure 13.	Project used with existing CMake source wizard	10
Figure 14.	Project import icon	10
Figure 15.	<i>Project Explorer</i> window populated with a project	11
Figure 16.	Main control icons for editing, building, and project management	12

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 STMicroelectronics – All rights reserved