

3.3 Launching STM32CubeMX

When running STM32CubeMX behind a proxy, see [Section 3.4.1](#)

3.3.1 Running STM32CubeMX as a standalone application

To run STM32CubeMX as a standalone application on Windows select STM32CubeMX from Program Files > ST Microelectronics > STM32CubeMX or double-click STM32CubeMX icon on your desktop.

To run STM32CubeMX as a standalone application on Linux, launch the STM32CubeMX executable from STM32CubeMX installation directory.

To run STM32CubeMX as a standalone application on macOS, launch the STM32CubeMX application from the launchpad.

Note: There is no STM32CubeMX desktop icon on macOS.

3.3.2 Running STM32CubeMX in command-line mode

To facilitate its integration with other tools, STM32CubeMX provides a command-line mode. Using a set of commands (listed in [Table 1](#)) it is possible to:

- load an MCU
- load an existing configuration
- save a current configuration
- set project parameters and generate corresponding code
- generate user code from templates
- load a board identified through its part number
- refresh the list of embedded software packages (packs and STM32Cube MCU packages) and install/remove a package
- select additional software (packs) components to add to the project.

Three command-line modes are available:

- To run STM32CubeMX in interactive command-line mode, use the following command lines:

– On Windows:

```
cd <STM32CubeMX installation path>
jre\bin\java -jar STM32CubeMX.exe -i
```

– On Linux and macOS:

```
STM32CubeMX -i
```

The “MX>” prompt is then displayed to indicate that the application is ready to accept commands.

- To run STM32CubeMX in command-line mode getting commands from a script, use the following command lines:

– On Windows:

```
cd <STM32CubeMX installation path>
jre\bin\java -jar STM32CubeMX.exe -s <script filename>
```

- On Linux and macOS:

```
STM32CubeMX -s <script filename>
```

All the commands to be executed must be listed in the script file. An example of script file content is shown below:

```
load STM32F417VETx
project name MyFirstMXGeneratedProject
project toolchain "MDK-ARM v4"
project path C:\STM32CubeProjects\STM32F417VETx
project generate
exit
```

- To run STM32CubeMX in command-line mode getting commands from a script and without UI, use the following command lines:

- On Windows:

```
cd <STM32CubeMX installation path>
jre\bin\java -jar STM32CubeMX.exe -q <script filename>
```

- On Linux and macOS:

```
STM32CubeMX -q <script filename>
```

Here again, the user can enter commands when the MX prompt is displayed.

Table 1. Command line summary

Command line	Purpose	Example
help	Displays the list of available commands.	help
swmgr refresh	Refreshes the list of embedded software package versions available for download.	swmgr refresh
swmgr install stm32cube_<series> _<version> ask	Installs the specified STM32Cube MCU package version.	swmgr install stm32cube_f1_1.8.0 ask
swmgr remove stm32cube_<series> _<version>	Removes the specified STM32Cube MCU package version.	swmgr remove stm32cube_f1_1.8.0
swmgr install <packVendor>.<packName>. <packVersion> ask	Installs the specified pack version.	swmgr install STMicroelectronics. X-CUBE-NFC4.1.4.1 ask
swmgr remove <packVendor>.<packName>. <packVersion>	Removes the specified pack version.	swmgr remove STMicroelectronics. X-CUBE-BLE1.4.2.0
swmgr install <filename path> <license-mode (accept ask)>	Installs an embedded software package.	swmgr install "C:\repo\packs\STMicroelectronics. X-CUBE-BLE1.4.2.0.pack" accept

Table 1. Command line summary (continued)

Command line	Purpose	Example
pack enable <vendor> <pack>[/bundle] <version> <class> <group>[/subgroup] [variant]	Selects a software pack component to add in the project. The presence of the "/" in the second and/or the fifth parameter(s) indicates, respectively, the explicit mention of a bundle and/or a subgroup (reference: Arm CMSIS pack pdsc format). To find out the pack / bundle / class / group / subgroup names of the component to enable, select the component and click "Hide/Show details" from the Additional Software window.	pack enable STMicroelectronics "X-CUBE-BLE1/BlueNRG-MS" 1.0.0 "Wireless" "Controller"
pack validate	Applies in the project all pack components enabled since the "pack validate" command was last called.	pack validate
load <mcu>	Loads the selected MCU.	load STM32F101RCTx load STM32F101Z(F-G)Tx
load <board part number> <allmodes nomode>	Loads the selected board with all peripherals configured in their default mode (allmodes) or without any configuration (nomode).	loadboard NUCLEO-F030R8 allmodes loadboard NUCLEO-F030R8 nomode
config load <filename>	Loads a previously saved configuration.	config load C:\Cube\ccmram\ccmram.ioc
config save <filename>	Saves the current configuration.	config save C:\Cube\ccmram\ccmram.ioc
config saveext <filename>	Saves the current configuration with all parameters, including those for which values have been kept to default (unchanged by the user).	config saveext C:\Cube\ccmram\ccmram.ioc
config saveas <filename>	Saves the current project under a new name.	config saveas C:\Cube\ccmram2\ccmram2.ioc
csv pinout <filename>	Exports the current pin configuration as a csv file. This file can be (later) imported into a board layout tool.	Csv pinout mypinout.csv
script <filename>	Runs all commands in the script file. There must be one command per line.	script myscript.txt
project couplefilesbyip <0 1>	This option allows the user to choose between 0 (to generate the peripheral initializations in the main) and 1 (to generate each peripheral initialization in dedicated .c/.h files).	project couplefilesbyip 1

Table 1. Command line summary (continued)

Command line	Purpose	Example
setDriver <Peripheral Name> <HAL LL>	For supported Series, STM32CubeMX can generate peripheral initialization code based on LL or on HAL drivers. This command line allows the user to choose, for each peripheral, between HAL-based and LL-based code generation. By default code generation is based on HAL drivers.	setDriver ADC LL setDriver I2C HAL
generate code <path>	Generates only “STM32CubeMX generated” code and not a complete project that would include STM32Cube firmware libraries and toolchains project files. To generate a project, use “project generate”.	generate code C:\mypath
set tpl_path <path>	Sets the path to the source folder containing the .ftl user template files. All the template files stored in this folder are used for code generation.	set tpl_path C:\myTemplates\
set dest_path <path>	Sets the path to the destination folder that will hold the code generated according to user templates.	set dest_path C:\myMXProject\inc\
get tpl_path	Retrieves the path name of the user template source folder.	get tpl_path
get dest_path	Retrieves the path name of the user template destination folder.	get dest_path
SetStructure <Advanced/Basic>	Selects the project structure to generate.	SetStructure Basic
SetCopyLibrary <copy all / copy only / copy as reference>	Selects how the reference libraries are copied to the projects.	SetCopyLibrary "copy all"
project setCustomFWPath <CustomFwLocation>	Specifies a path to STM32Cube MCU software libraries different from STM32Cube repository path (specified under Help > Updater settings).	project SetCustomFwPath "F:/SharedRepository/STM32Cube_FW_F0_V1.11.0"
project toolchain <toolchain>	Specifies the toolchain to be used for the project. Use the “project generate” command to generate the project for that toolchain.	project toolchain EWARM project toolchain “MDK-ARM V4” project toolchain “MDK-ARM V5” project toolchain TrueSTUDIO project toolchain SW4STM32

Table 1. Command line summary (continued)

Command line	Purpose	Example
project name <name>	Specifies the project name.	project name ccmram
project path <path>	Specifies the path where to generate the project.	project path C:\Cube\ccmram
project generate	Generates the full project.	project generate
exit	Ends STM32CubeMX process.	exit

3.4 Getting updates using STM32CubeMX

STM32CubeMX implements a mechanism to access the Internet and to:

- download embedded software packages: STM32Cube MCU packages (full releases and patches) and third-party packages (.pack) based on the Arm® CMIS pack format
- manage a user-defined list of third-party packs
- check for STM32CubeMX and embedded software packages updates
- perform self-updates of STM32CubeMX
- refresh STM32 MCUs descriptions and documentation offer.

Installation and update related submenus are available under the **Help** menu and from the home page as well.

Off-line updates can also be performed on computers without Internet access (see [Section 3.4.3](#)). This is done by browsing the filesystem and selecting available STM32Cube MCU packages.

If the PC on which STM32CubeMX runs is connected to a computer network using a proxy server, STM32CubeMX needs to connect to that server to access the Internet, get self-updates and download firmware packages. Refer to [Section 3.4.2](#) for a description of this connection configuration.

To view Windows default proxy settings, select Internet options from the Control panel and select LAN settings from the **Connections** tab (see [Figure 5](#)).