



# STM32CubeProgrammer + Atollic TrueSTUDIO for STM32

## **Integrating STM32CubeProgrammer**



What - why - how

What is STM32CubeProgrammer?

Why integrate it?

How can the tool be integrated?

# What is STM32CubeProgrammer?

STM32CubeProgrammer (STM32CUBEPROG) is an all-in-one multi-OS software tool for programming STM32 microcontrollers and external memories.

**Pre-requisite – download STM32CubeProgrammer here:**

<http://www.st.com/en/development-tools/stm32cubeprog.html>

This provides an environment for reading, writing and verifying device memory through both the debug interface (JTAG and SWD) and the bootloader interface (UART and USB).

STM32CubeProgrammer is delivered in GUI (graphical user interface) and CLI (command-line interface) versions.

This app-note will describe **how to invoke the CLI version from inside Atollic TrueSTUDIO** (or any other Eclipse based tool).

# Why/when integrate STM32CubeProgrammer?

Do you really need to integrate STM32CubeProgrammer into Atollic TrueSTUDIO? Most likely the answer is "No".

BUT STM32CubeProgrammer offers more functionality compared to the Atollic TrueSTUDIO ST-Link GDB-server which manage the programming of STM32.

Integrating STM32CubeProgrammer into TrueSTUDIO can boost your efficiency compared to using ST-Link GDB-server. Here are some use cases:

- STM32CubeProgrammer supports flashing of STM32 devices not yet supported by ST-Link GDB-server
- STM32CubeProgrammer flash operation is sometimes faster than ST-Link GDB-server
- Option bytes must be set/cleared pre/post flash operation
- Flash download must be handled using UART/USB (bootloader)
- STM32CubeProgrammer can program external memories

## How-to use STM32CubeProgrammer GUI?

The process to program the QSPI flash on the STM32F723E-DISCO board using STM32CubeProgrammer is:

1. Connect the board to the PC.
2. Start STM32CubeProgrammer and open Erasing and Programming view.
3. Browse to the file to be programmed. For instance the file can be an elf file or hex file.
4. Select programming options to be used. E.g. Verify Programming.
5. Select the MX25L512G\_STM32F723E-DISCO external loader.
6. Press Connect to connect to the board.
7. Press the “Start Programming” button to Program the QSPI flash.
8. Wait until programming is completed.
9. Select Disconnect so that ST-LINK\_gdbserver can connect to the board.

The project can now be debugged in Atollic TrueSTUDIO. Make sure that the “load” command is commented in the GDB startup script so that GDB does not try to load the program with the ST-LINK\_gdbserver.

# How-to use STM32CubeProgrammer CLI?

STM32\_Programmer\_CLI.exe supports many command line options but normally only a few are required to program flash memory.

## 1. Internal STM32 Flash Memory Programming

STM32CubeProgrammer determines the flash algorithm to use by reading the STM32 Device ID.

Programming can be done by just specifying the connection and the elf-file.

Example: Using '-c' option (port=SWD) and '-d' option (elf-file)

```
>STM32_Programmer_CLI.exe -c port=SWD -d G:/workspace/ApplicationLED1/Debug/ApplicationLED1.elf
```

## 2. External Memory Programming

When an external memory shall be programmed an external memory loader file needs to be specified.

STM32CubeProgrammer includes external loaders, ".stldr" files, for most available STM32 EVAL and Discovery boards.

Example: Using the '-el' option (stldr-file) to program STM32F723E-DISCO Quad-SPI flash

```
>STM32_Programmer_CLI.exe -c port=SWD -el "C:/Program Files/STMicroelectronics/STM32Cube/STM32CubeProgrammer/bin/ExternalLoader/MX25L512G_STM32F723E-DISCO.stldr" -d G:/workspace/ApplicationLED2/Debug/ApplicationLED2.elf
```

The STM32CubeProgrammer User Manual (UM2237) contains information on how to develop customized loaders for external memories.

# How-to integrate STM32CubeProgrammer?

Atollic TrueSTUDIO (Eclipse) supports setting up *external tools configurations*

*External tool configurations* can be chained into a *Launch Group*

We will now setup STM32CubeProgrammer CLI as an *external tool*. This tool will be chained into a *Launch Group* together with your ST-Link debug configuration so that each time the *Launch Group* is run the following happens:

1. STM32Cubeprogrammer is launched to carry out programming related operations. Then disconnects from the ST-Link and exits.
2. A debug session is launched using the ST-Link GDB-server. The debug configuration skips the programming operation (load).

The external tools configuration allows the user to setup multiple configurations using different STM32CubeProgrammer commands.



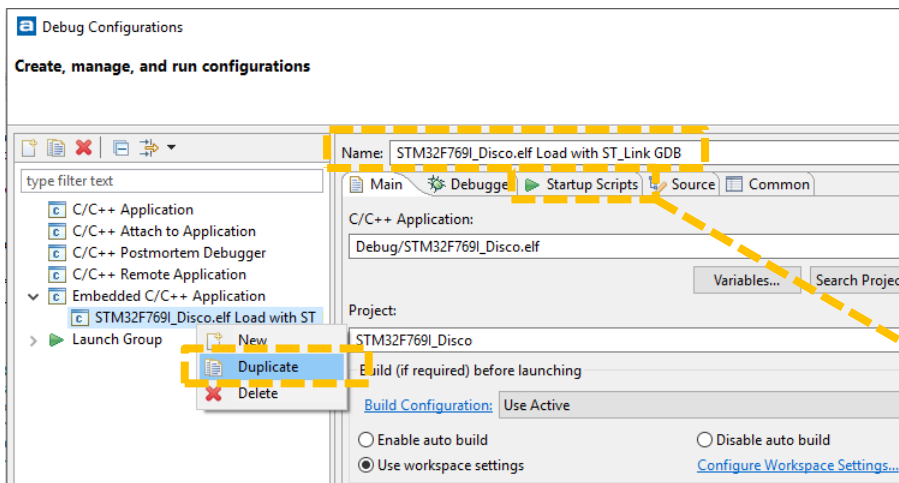
## How-to

This chapter describes how-to setup the integration in Atollic TrueSTUDIO (Eclipse)

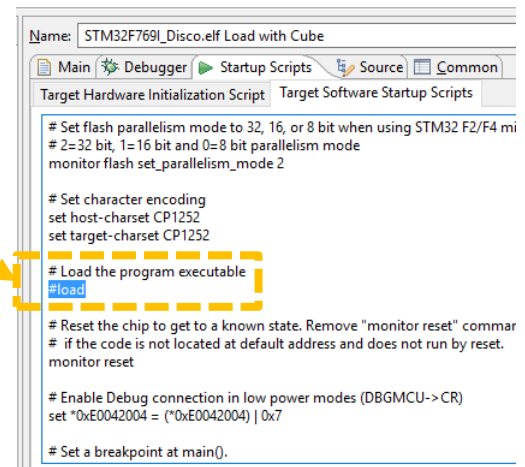
# First duplicate the debug configuration (optional)

Do this for back-up purpose or in case you want to be able to chain up different launch use cases and need one debug configuration for each use case:

1. Run → Debug Configurations...
2. Right-click on your Embedded debug configuration → Duplicate
3. Give the copy a new name

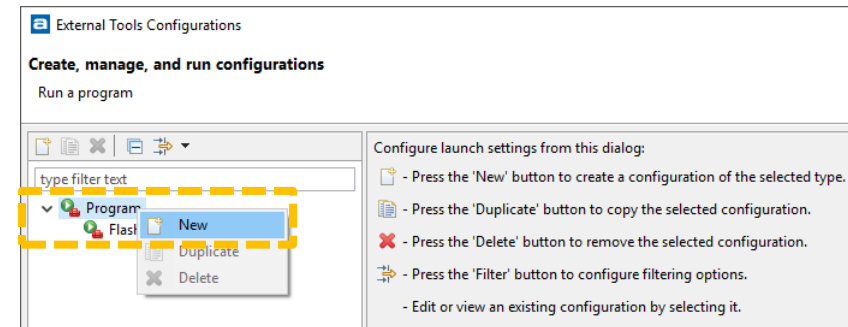
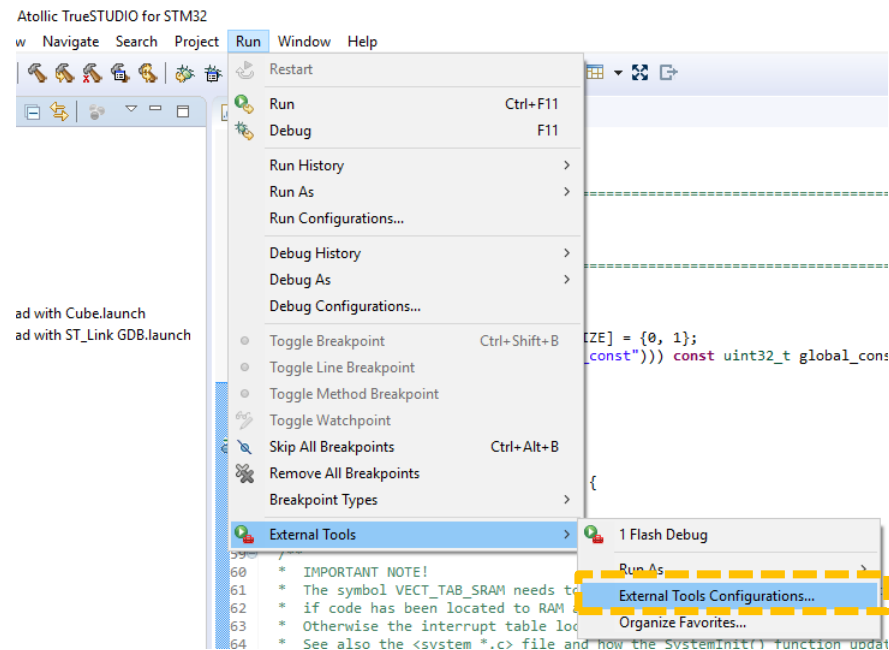


4. Open *Startup Scripts* tab
5. Remove or comment the load command



# Create an ext. tools config...

Run → External Tools → External Tools Configurations...



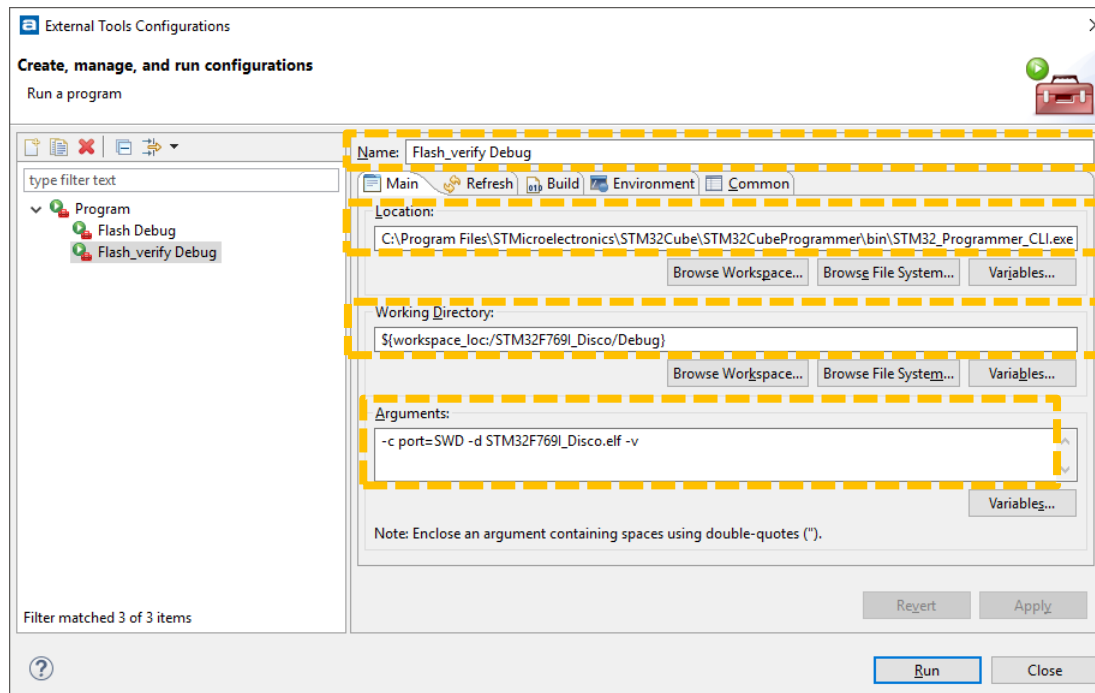
# Ext. tools config for STM32CubeProgrammer

*Name:* Give the configuration a name that describes what it does

*Location:* Should point to the STM32CubeProgrammer CLI executable

*Working directory:* Can be set to project output folder for the relevant build configuration

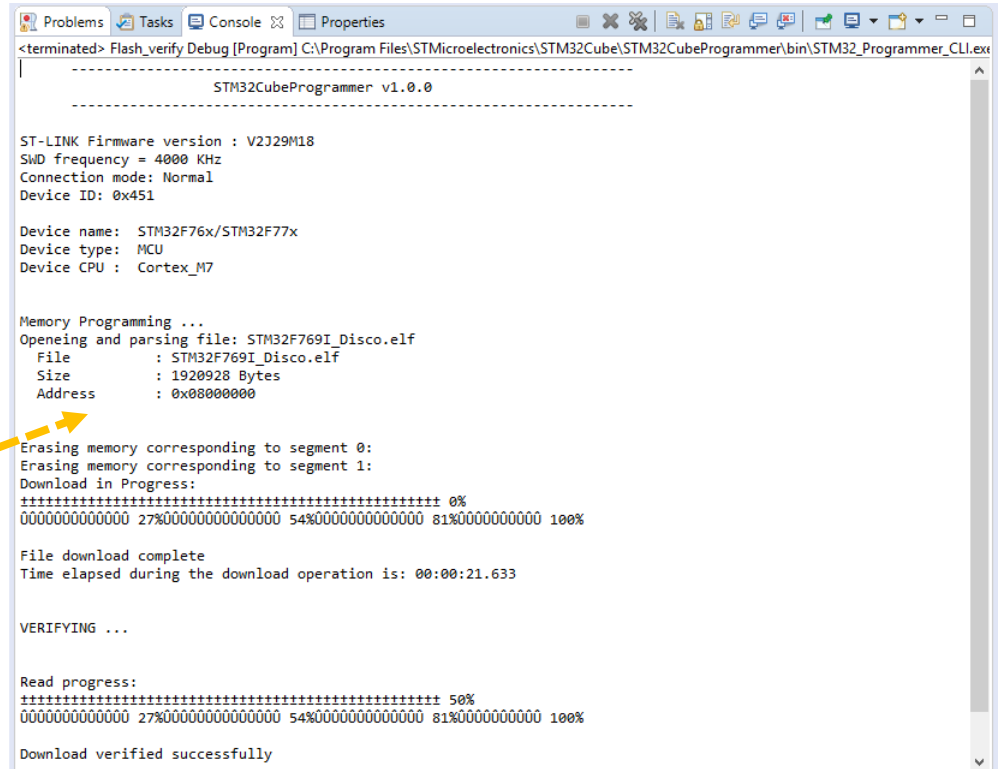
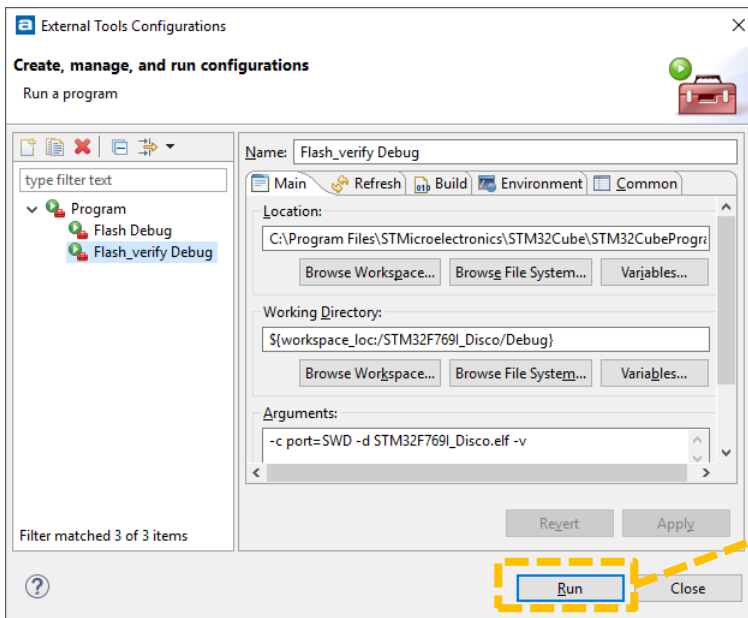
*Arguments:* STM32CubeProgrammer will be called with these arguments/options. The '-el' argument shall be added when programming external memory.



When this step is done you can call STM32CubeProgrammer from within the IDE!

# Test launch of STM32CubeProgrammer

Verify that your ext. tools configuration works as expected by running it!



No error message?

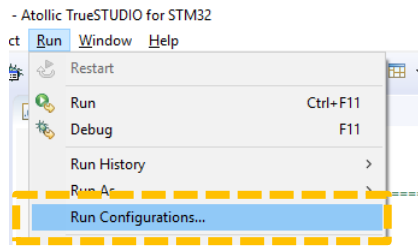
Working ext. tools configuration!

# Create a Launch Group...

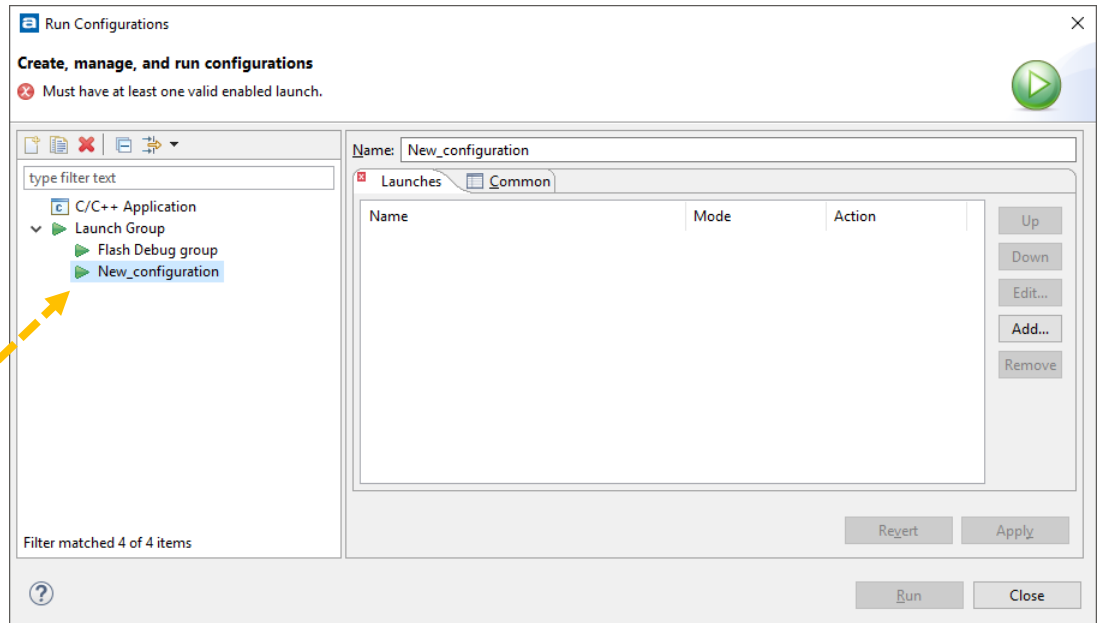
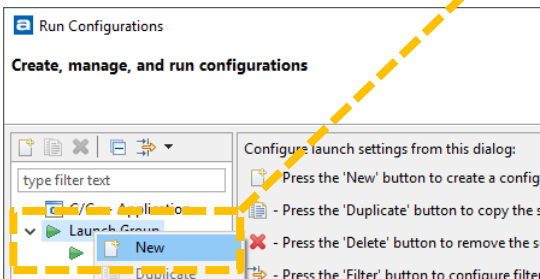
Create a launch group run configuration that will perform:

1. flash with *ext. tool*
2. launch *debug configuration*

Run → *Run Configurations...*



Right-click *Launch Group* → *New*



# Add flash step into Launch Group

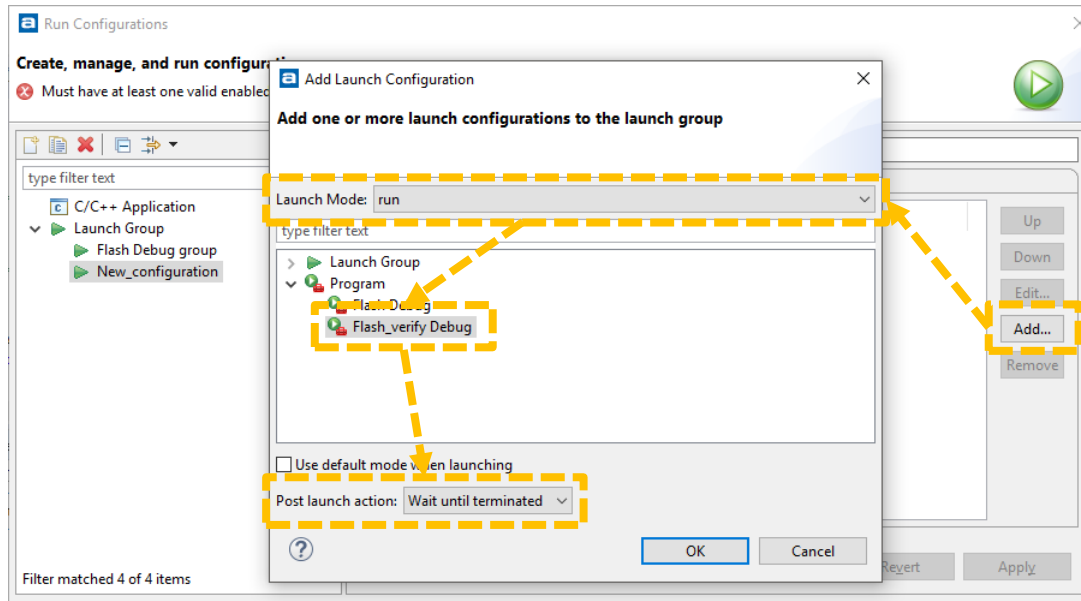
Add STM32CubeProgrammer ext. tool configuration as first step in the launch group:

*Run Configurations* → *Add...*

*Launch mode:* **run**

*Program:* The ext.tool configured on page 9-10

*Post launch action:* **Wait until terminated** ← **IMPORTANT!**



# Add debug configuration into Launch Group

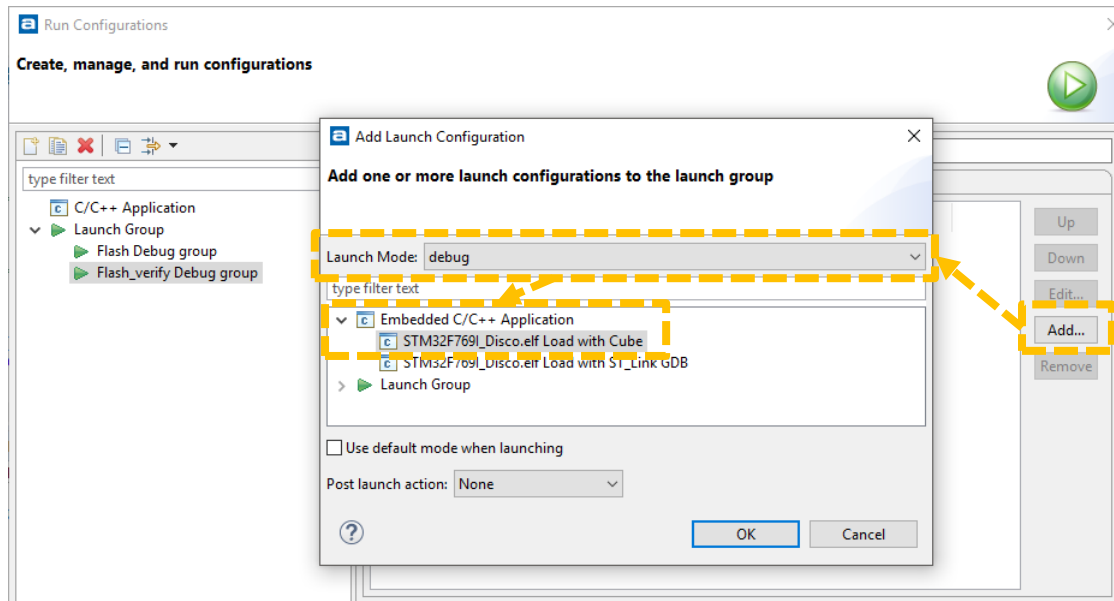
Add a *debug configuration* to the *Launch Group* after STM32CubeProgrammer has finished flash operation:

*Run Configurations* → *Add...*

*Launch mode:* **debug**

*Embedded C Application:* The debug configuration used in TrueSTUDIO to debug the project

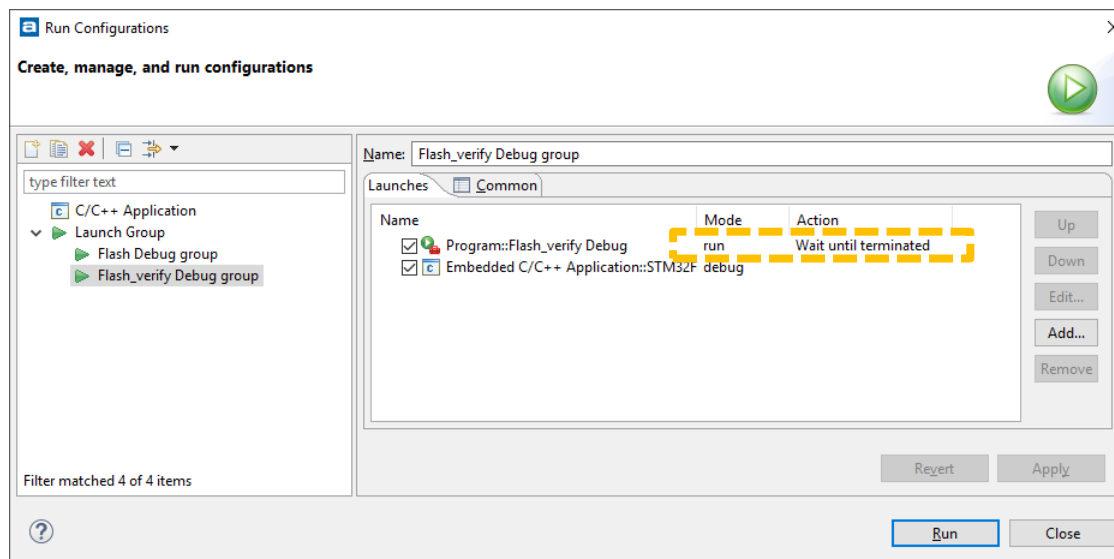
*Post launch action:* **none**





# How-to integrate STM32CubeProgrammer?

The complete *Launch Group* should look like this:



Click *Run* to test that:

1. The Launch Group runs STM32CubeProgrammer to flash the device
2. The Launch Group launches the debug configuration to debug the device

# Done!

We are always interested to hear your feedback!

Drop an e-mail to [support@atollic.com](mailto:support@atollic.com), use the e-mail subject: "*AN1801\_cubeprogrammer\_in\_truestudio*"