



STM32 Ecosystem workshop

T.O.M.A.S Team





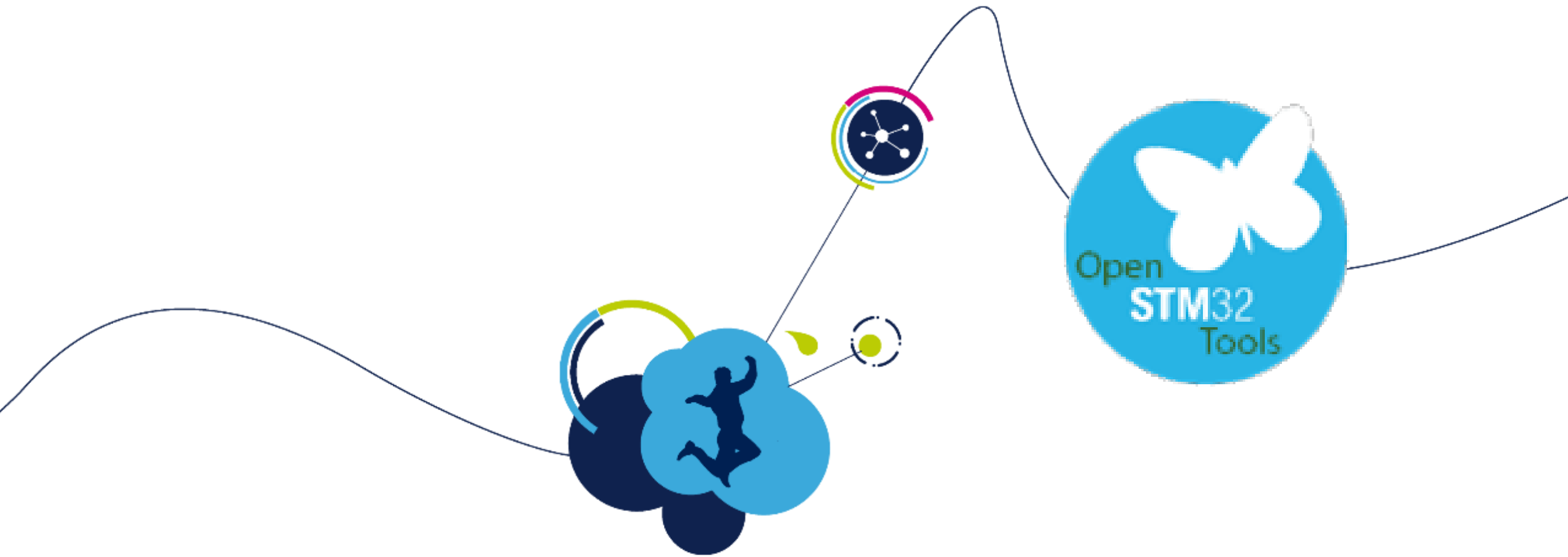
- Now it is a right time for some activities again 😊
- Our goal is to adjust the configuration of SW4STM32 environment.
- You can follow next steps directly on your “L4_Blinky” project.



Our goals for this session

3

- Get knowledge about some features of SW4STM32 toolchain:
 - Selection C dialect and parallel build options
 - Configuration of the indexer
 - How to run an external application
 - How to set autosave before build option
 - How to use “come back later” messages within the code
 - How to use and configure autocompleting option
 - Managing different output files (.hex, .bin files generation)
 - How to run debug session and handling most common issues related to this operation



Some tips in SW4STM32



Code generation strategies for SW4STM32

generation of project files under root or in dedicated folder

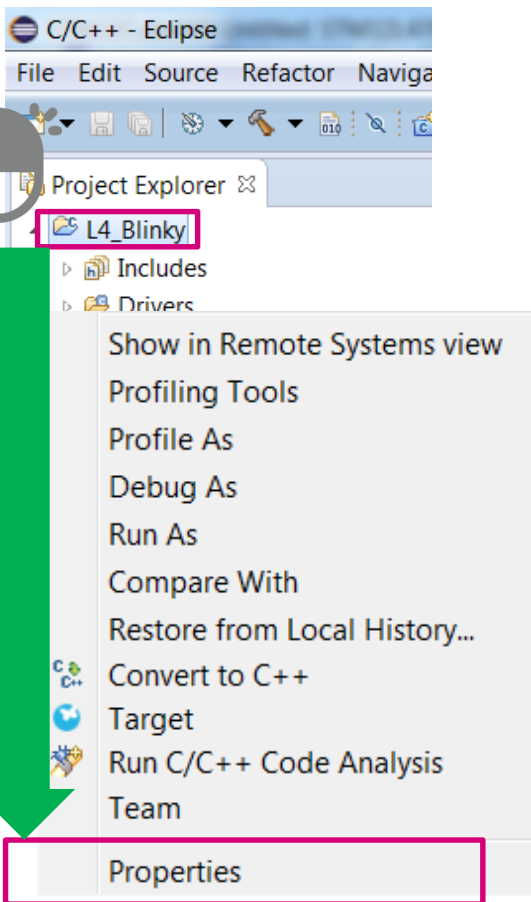
5

- There are two ways how to generate project files for SW4STM32 from STM32CubeMX application:
 1. Generate the toolchain project files in STM32CubeMX user project root folder (**Generate under root** checkbox should be **selected**) -> **we will use this option during this session**
 2. Generate the toolchain project files in dedicated toolchain folder (**Generate under root** checkbox should be **not selected**)
- STM32CubeMX project generation under the root folder allows to:
 - Optional copy of the project into the Eclipse workspace when importing a project.
 - Use of source control systems such as GIT or SVN from the Eclipse workspace.
- **Warning:** Choosing to copy the project into workspace will prevent any further synchronization between changes done in Eclipse and changes done in STM32CubeMX as there will be 2 different copies of the project.



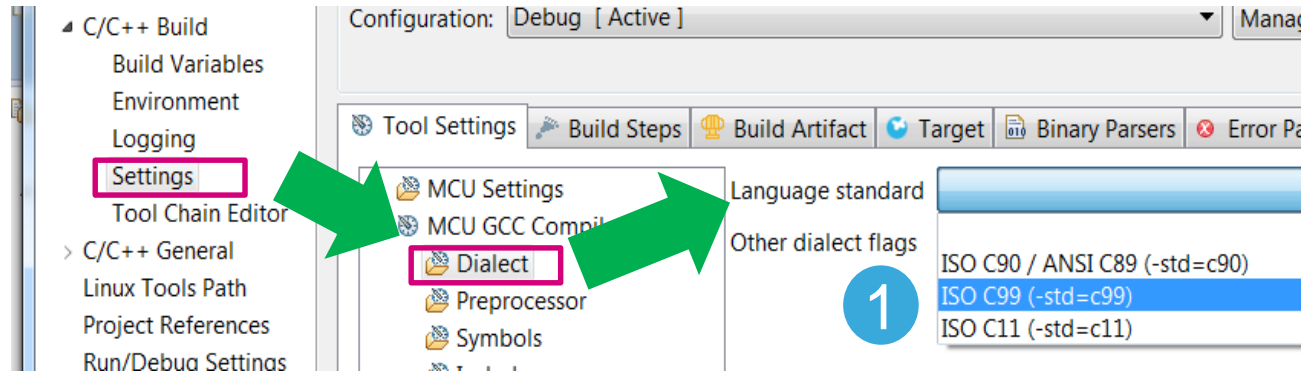
Useful project settings in SW4STM32 configuring C dialect and parallel build

Project->Properties



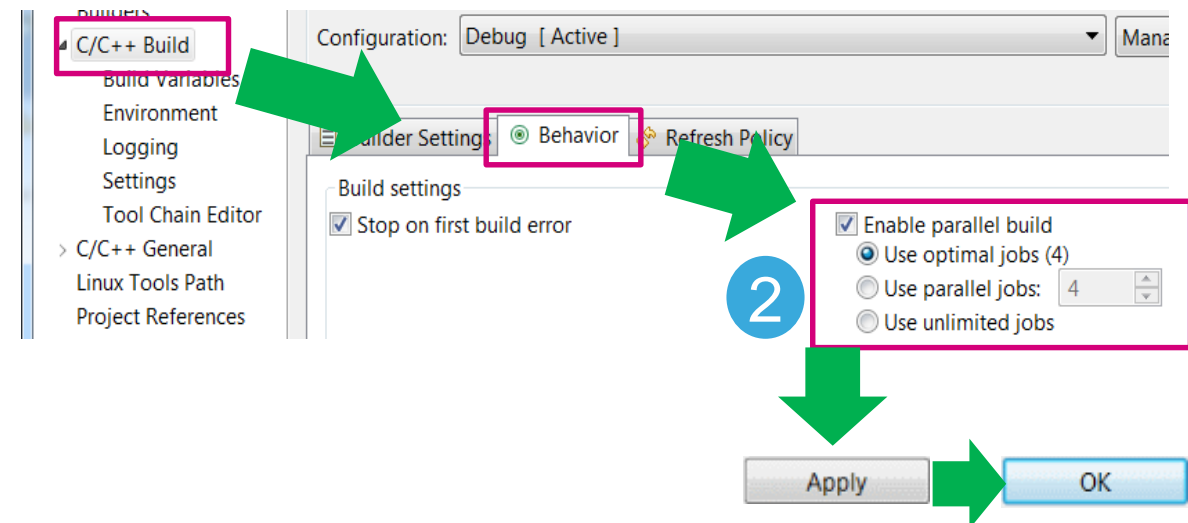
C/C++ Build->Settings->Tools Settings tab → MCU GCC Compiler->Dialect

1. Configure C standard to **C99** to avoid possible compilation errors



C/C++ Build->Behavior tab

2. Check **Enable parallel build** to make use of your machine potential and to shorten compilation time



NOTE: These steps were already mentioned in a previous part, here duplicated to keep narration consistency.



Indexer in SW4STM32

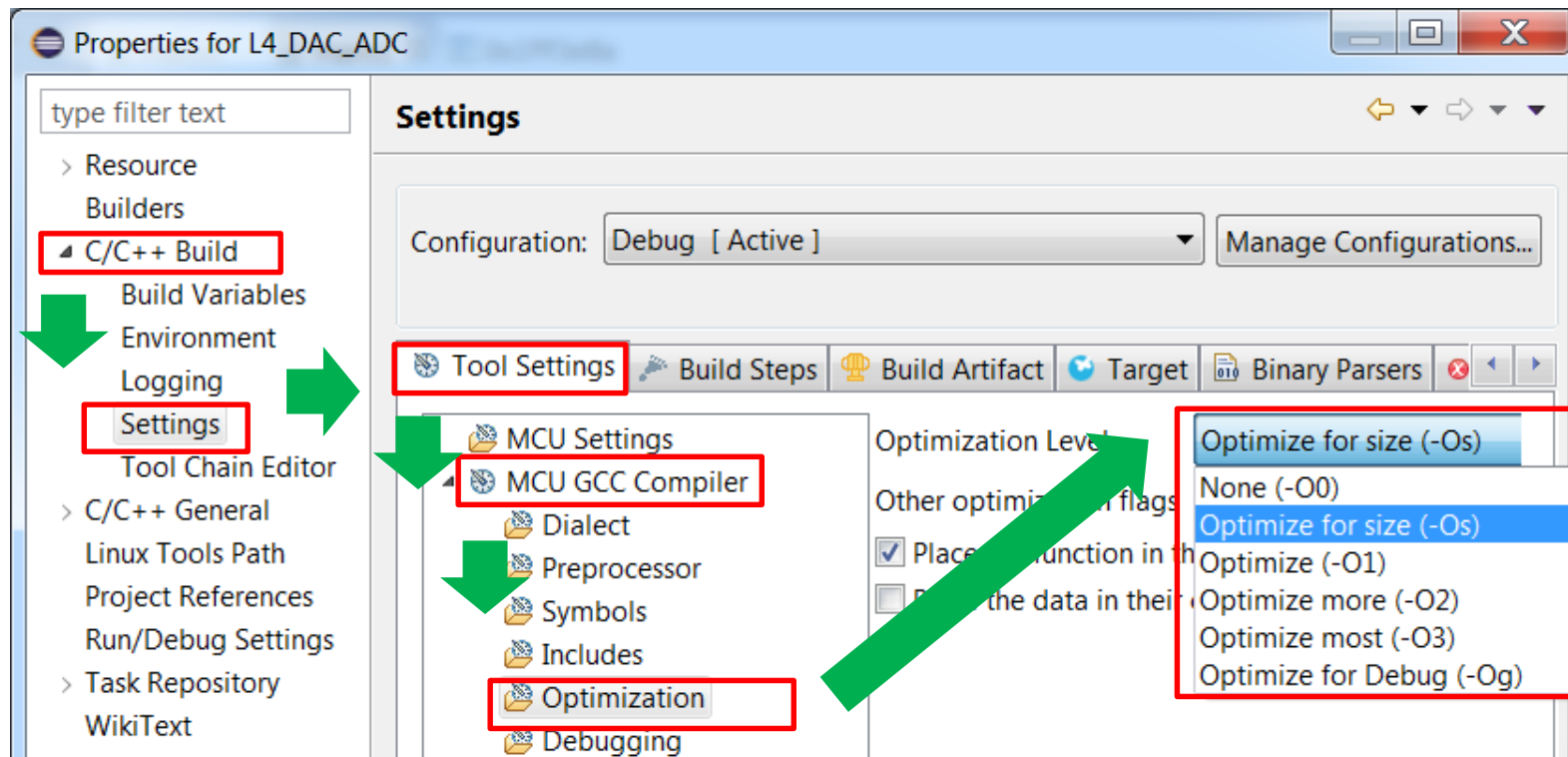
7

- The indexer uses the parser to create a database of project files that provides the basis for search, navigation features and parts of content assist (Ctrl+Space feature).
- The indexer runs on a background and reacts to resource change like:
 - Project creation/deletion
 - Source files creation/deletion
 - File imports
 - Source file content changes
- It is possible to customize the behavior of the indexer (or turn it off) for each project within the workspace separately.

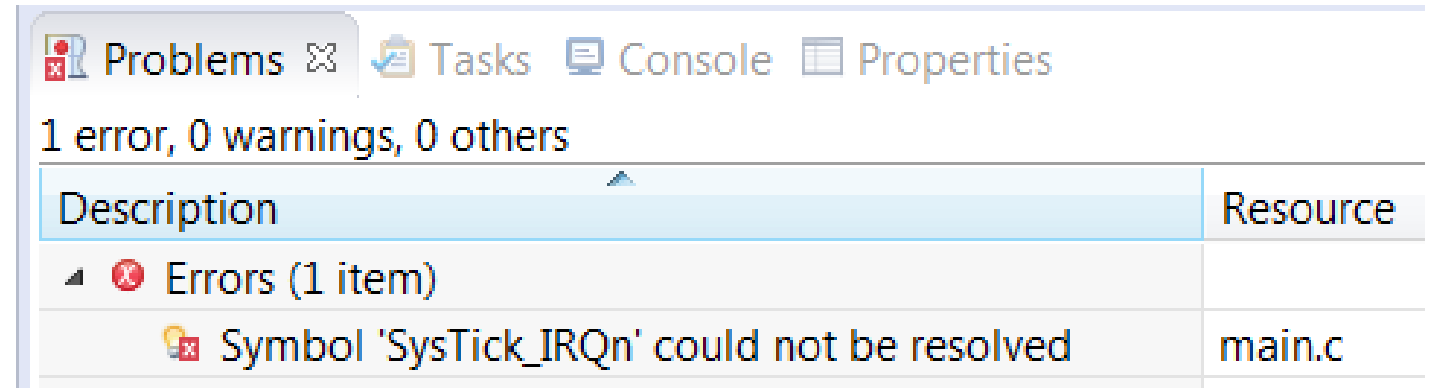


Compiler optimization levels

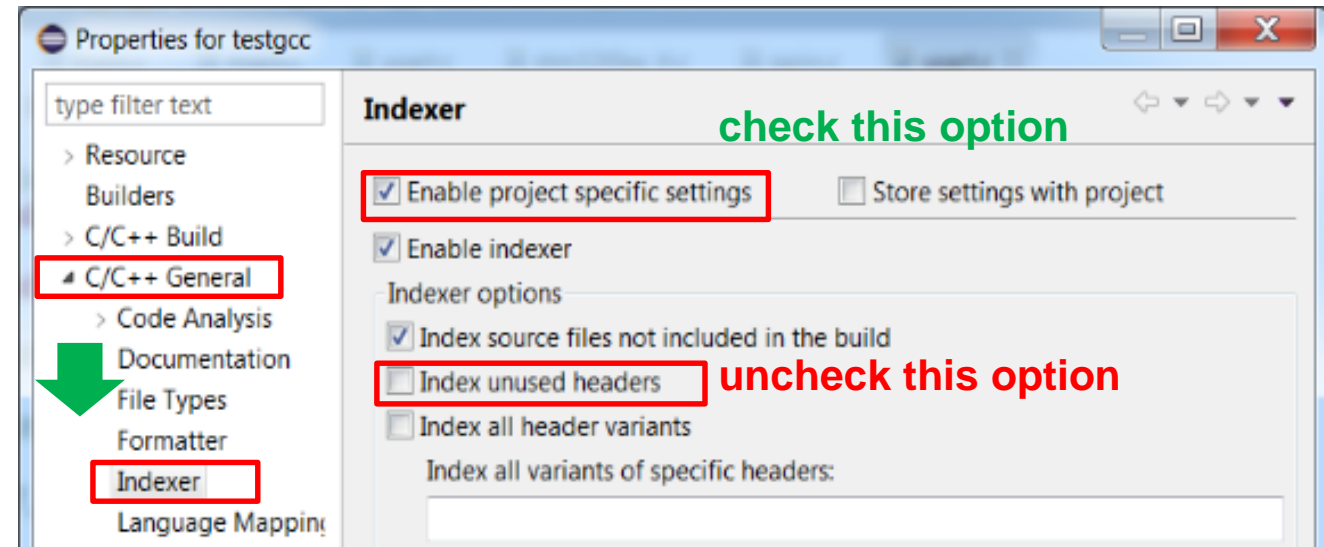
- Optimization levels are accessible within project properties (right click the project name)
- Within project Properties select:
 - C/C++ Build -> Settings ->Tool Settings->MCU GCC Compiler->Optimization From
- From drop-down menu select required optimization level
- Click **Apply->OK** buttons when completed



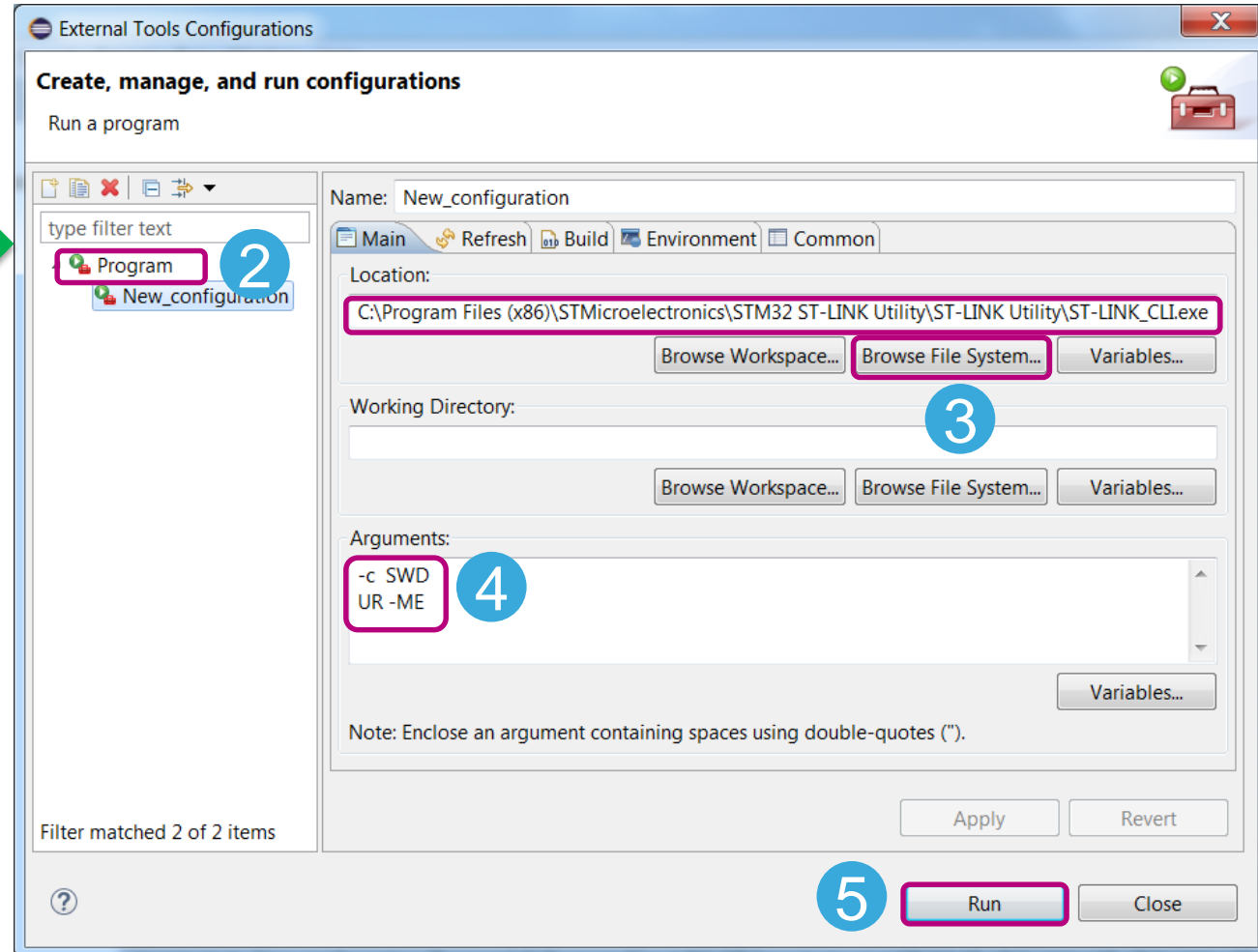
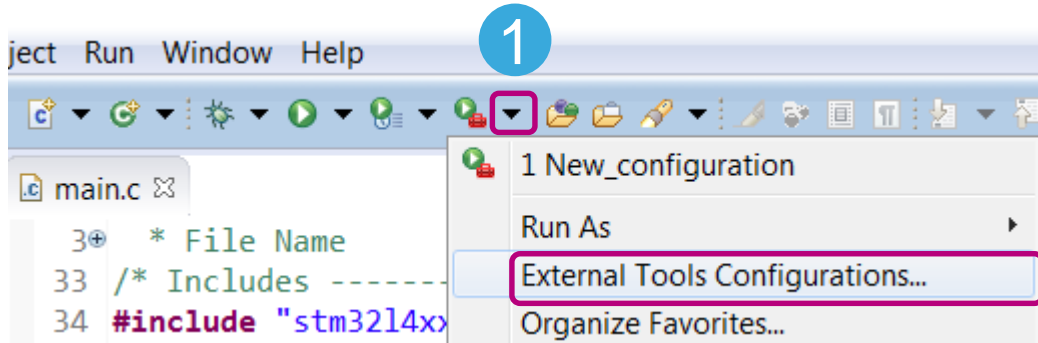
- During code generation with “Generate Under Root” option selected, all header files for a complete family are added to the project causing an error with unresolved parameter (too many sources with the same definition) → this is necessary to change indexer settings.



Project Properties->C/C++ General->Indexer



Example: How to erase Flash memory using command line STLink Utility application from SW4STM32



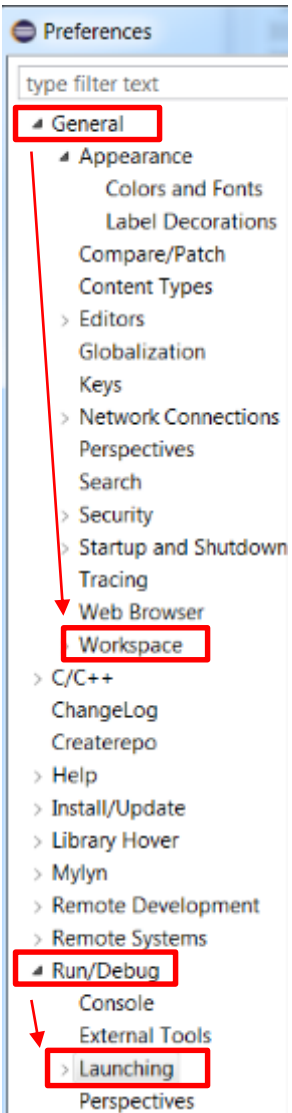
1. Select **External Tools Configuration** from Run configuration option
2. Double click **Program** to create new configuration
3. Browse **ST-LINK_CLI.exe** within File System
4. Add **-c SWD UR -ME** arguments
 - **-c SWD** → use SWD to connect to the target
 - **UR** → connect under reset
 - **-ME** → mass erase of the FLASH memory
5. Run the program
(**ST-LINK_CLI.exe -c SWD UR -ME** will be executed)



Useful configuration

auto save on build and launch

Editor settings are available in **Windows**→**Preferences** menu



General->Workspace



Run/Debug->Launching



Workspace

See '[Startup and Shutdown](#)' for workspace startup and shutdown preferences.

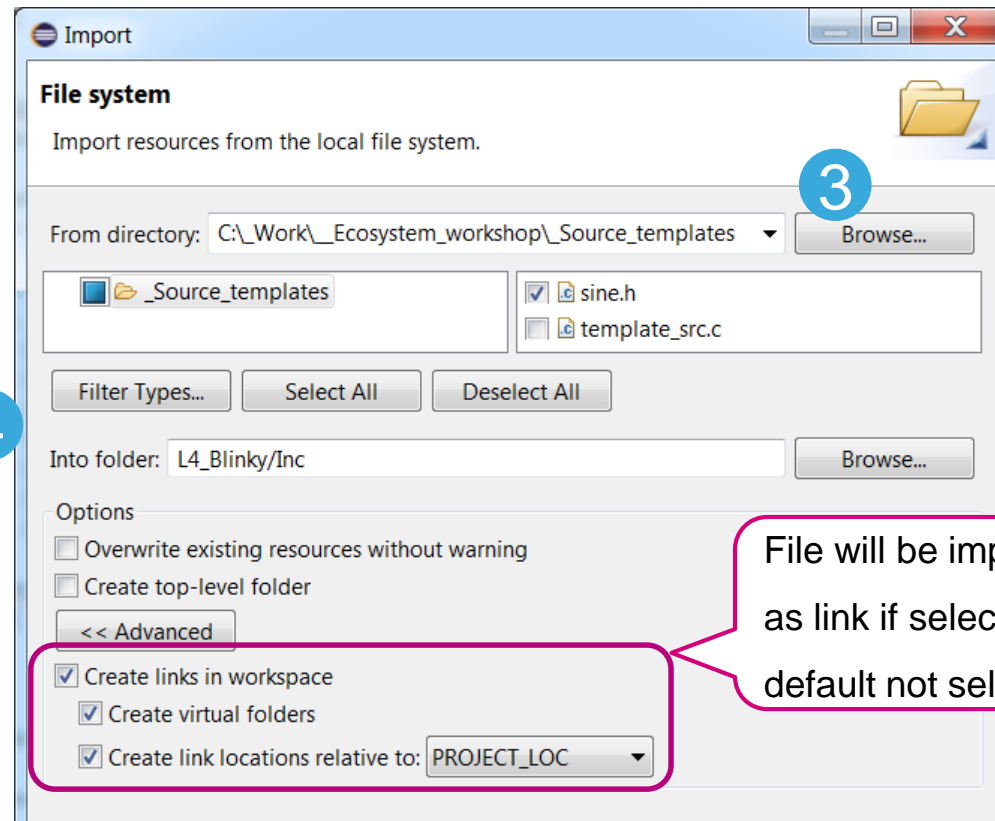
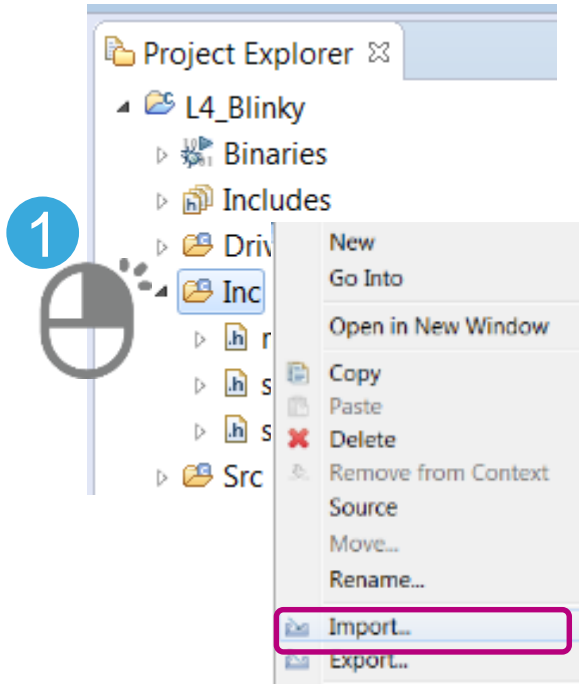
- Build automatically
- Refresh using native hooks or polling
- Refresh on access
- Save automatically before build
- Always close unrelated projects without prompt

Launching

Save required dirty editors before launching

- Always
- Never
- Prompt

1. Right-click the files folder within the project to select **IMPORT**
2. Select General->File System
3. Browse for files to be imported (`sine.h` in below example)
4. Select import options (click on Advanced button) and click **Finish** button
5. File will pop-up in selected folder

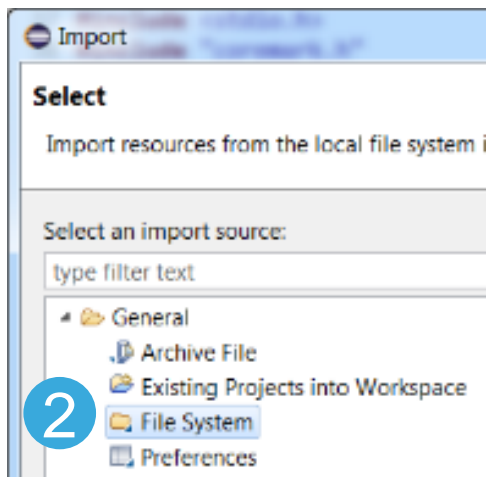
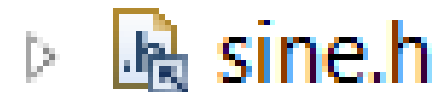


File copied into the folder



5

File linked to the folder



File will be imported as link if selected (by default not selected)



Importing projects into workspace

import from archive file

1. Select *File->Import* from SW4STM32 main menu
2. Select “Existing Projects...” from Import->General window

3 Select “Select archive file” and browse proper .zip /.tar file from Import Project window

4 Select project from the Projects lists and press Finish

2

Import

Select

Create new projects from an archive file or directory.

Select an import source:

type filter text

- General
- Archive File
- Existing Projects into Workspace
- File System
- Preferences
- C/C++
- Git
- Install
- Oomph

Import Projects

Select a directory to import into

Select archive file: C:_ST_Site\Trainings&Seminars\2016\201608 - Toc

Projects:

- L4_Blinky (L4_Blinky)
- L4_DAC_ADC (L4_DAC_ADC)
- L4_DAC_ADC_STMStudio (L4_DAC_ADC_STMStudio)
- L4_DAC_ADC_USART (L4_DAC_ADC_USART)
- L4_DAC_ADC_USART_SWO (L4_DAC_ADC_USART_SWO)
- LL_L4_DAC_ADC_Init (LL_L4_DAC_ADC_Init)
- LL_L4_DAC_ADC_Unitary (LL_L4_DAC_ADC_Unitary)

Options

- Search for
- Copy project
- Hide projects that already exist in the workspace

Working sets

Add project to working sets

Next >

5 Projects will be added into current workspace.

C/C++ - Eclipse

File Edit Source Refactor Navigate Search Project

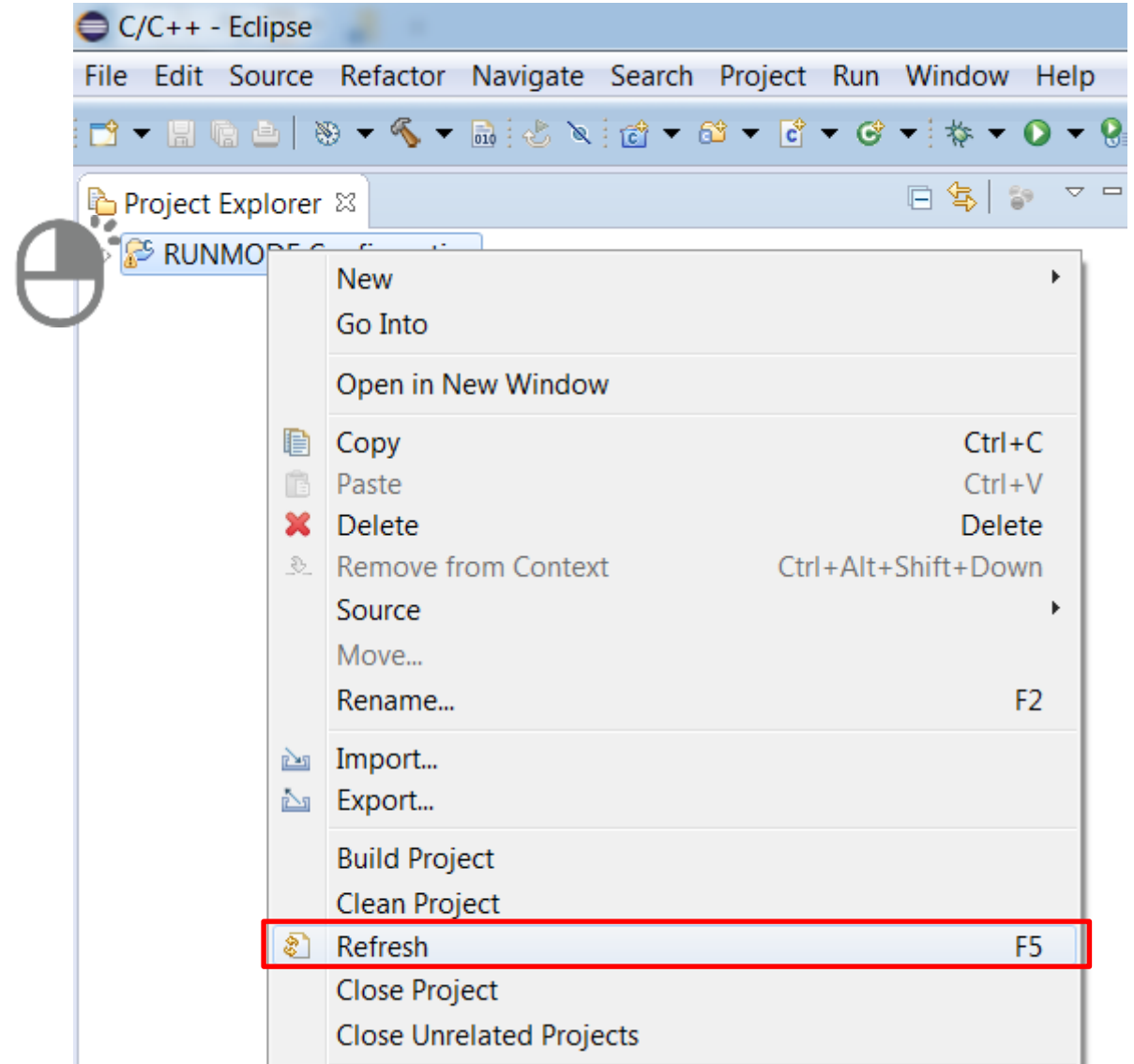
Project Explorer

- L4_Blinky
- L4_DAC_ADC
- L4_DAC_ADC_STMStudio
- L4_DAC_ADC_USART
- L4_DAC_ADC_USART_SWO
- LL_L4_DAC_ADC_Init
- LL_L4_DAC_ADC_Unitary



Project refresh after configuration change

- After performing any change to optimization/debug level, we strongly recommend to refresh project configuration (F5) and to build it again





Enable float support in printf

- By default “**nano**” version of the **stdio** (standard input-output library) is selected.
- This helps to keep code size low but does not allow use of float numbers in printf.
- In order to change it, we have to disable “nano” library option within project settings.

C/C++ Build -> Settings -> MCU GCC Linker -> Miscellaneous

Properties for L0_test

Settings

MCU Settings

- MCU GCC Compiler
 - Dialect
 - Preprocessor
 - Symbols
 - Includes
 - Optimization
 - Debugging
 - Warnings
 - Miscellaneous
- MCU GCC Linker
 - General
 - Libraries
 - Miscellaneous
 - Shared Library Settings

Linker flags: -specs=nosys.specs -specs=nano.specs

Other options (-Xlinker [option])

Remove this option

- Task configuration is done in **Task Tags** field within **Window**→**Preferences**→**C/C++**
- This is possible to **add new** and **modify existing** keywords that will be recognized after saving source file as Tasks marking (even being inside comment field - example (1) below).
- Double-clicking the particular task moves to the place of the code where the task is placed (2)

The image shows a composite screenshot of the Eclipse IDE illustrating the 'Come Back Later' task workflow. It is divided into three main sections:

- Left Panel (Preferences):** The 'Task Tags' preference page is shown. A table lists existing tags:

Tag	Priority
COME_BACK_LATER	High
FIXME	Normal
TODO (default)	Normal
XXX	Normal

 A 'New...' button is highlighted with a red box. A green arrow points from this button to the 'New Task Tag' dialog.
- Center Panel (New Task Tag Dialog):** A dialog box titled 'New Task Tag' is shown. The 'Tag' field contains 'COME_BACK_LATER' and the 'Priority' dropdown is set to 'High'. The 'OK' button is highlighted with a red box. A blue circle with the number '1' is placed next to the dialog.
- Right Panel (Code Editor):** The main editor shows a C file with the following code snippet:


```

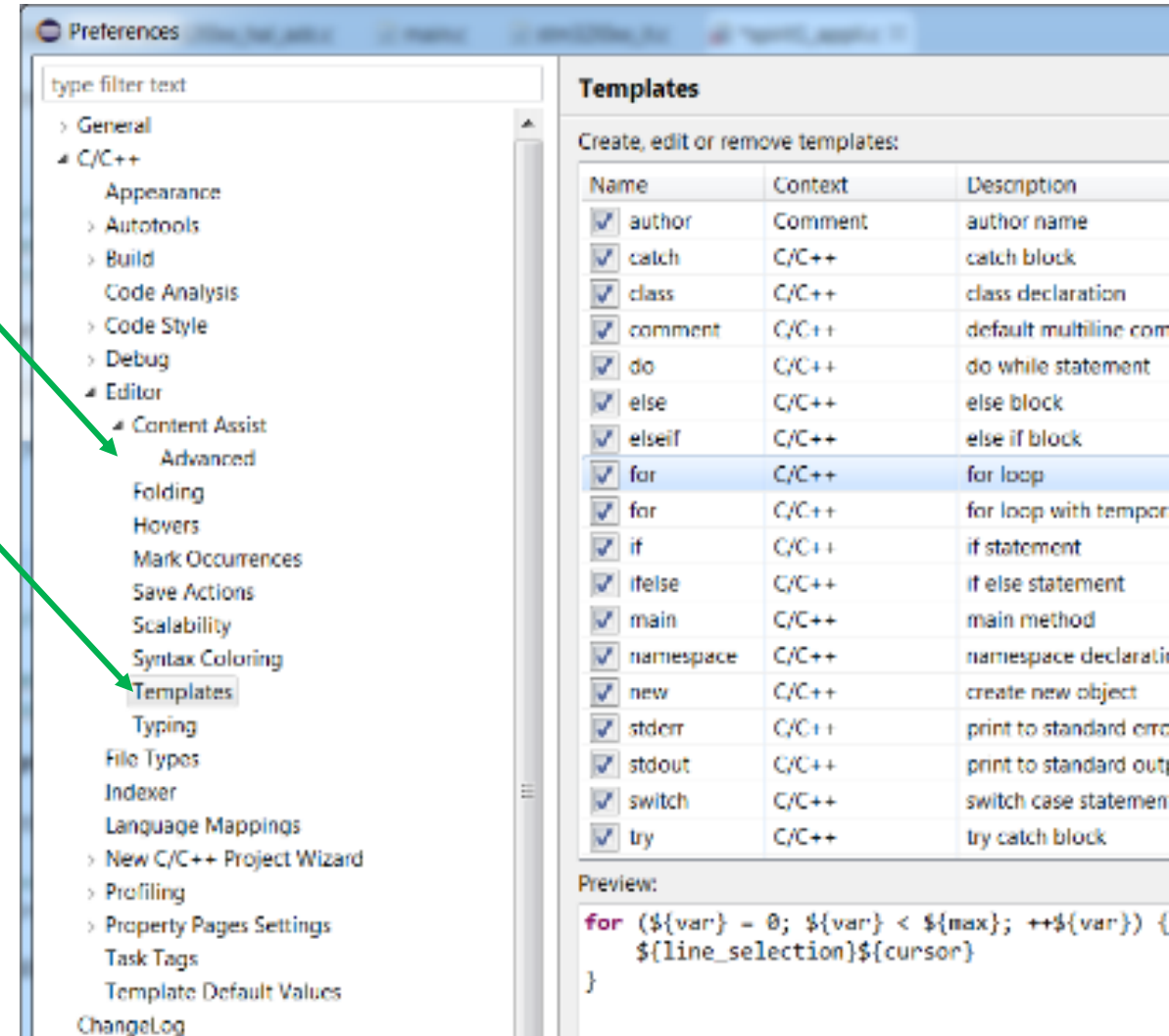
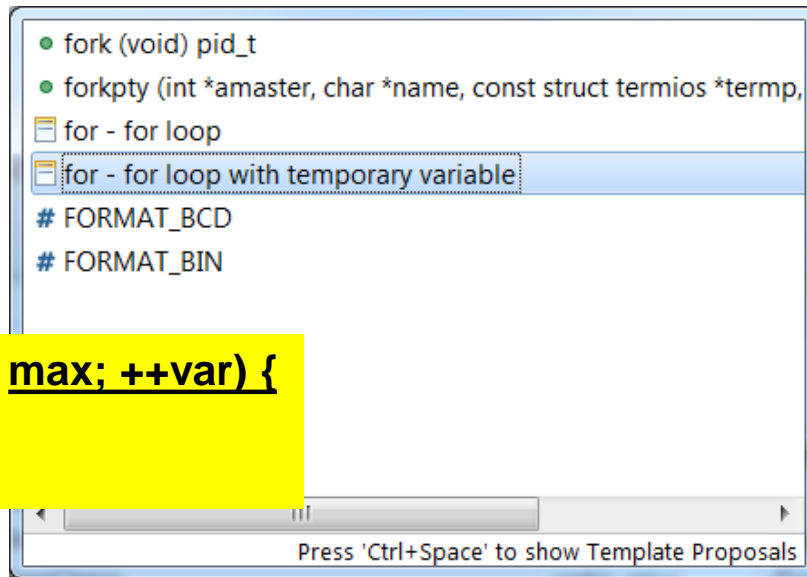
71 /* USER CODE END 0 */
72
73 int main(void)
74 {
75     /* USER CODE BEGIN 1 */
76     /* COME_BACK_LATER here I need to put some initialization */
77     /* USER CODE END 1 */
78
79     /* MCU Configuration -----
80
81     /* Reset of all peripherals, Initializes the Flash interface and the SysTick timer.
82     HAL_Init();
83
84
      
```

 The comment line 76 is highlighted with a red box. A green arrow points from this line to the 'Tasks' view at the bottom.
- Bottom Panel (Tasks View):** The 'Tasks' view at the bottom of the IDE shows a table with 2 items:

!	Description	Resource	Path
!	COME_BACK_LATER here I need to put some initialization	main.c	/L0_DAC_ADC/Src
	XXX Register index	stm321053r8t...	/L4_DAC_ADC/Drive

 A red box highlights the 'Tasks' view, and a green arrow points from the comment in the code editor to the first item in the table. A blue circle with the number '2' is placed next to the arrow.

- Basic configuration is done within:
 - Window→Preferences→C/C++→Editor→Content Assist
- Templates configuration is done within
 - Window→Preferences→C/C++→Editor→Templates
- Insert by Ctrl+Space during typing, i.e.
 - for <Ctrl+Space>
 then select required template to be used





Managing output files

useful modifications in post-build steps

18

- Open properties window for the project (Project→Properties)
- Select C/C++Build→Settings→Build Steps
- Modify Command field using below information (this is possible to add new commands after a **&&** string)

- To generate a binary (.bin) output file use:

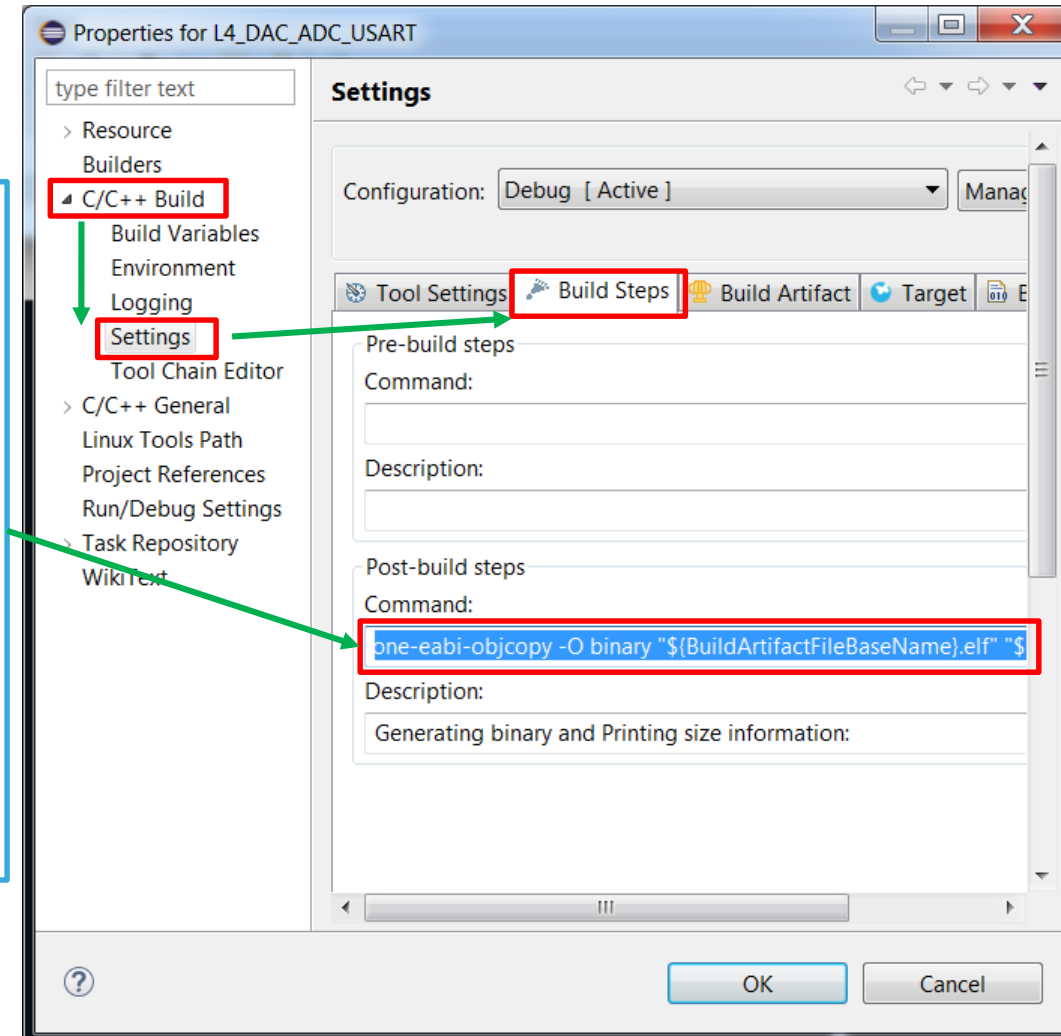
```
arm-none-eabi-objcopy -O binary "out_file.elf" "out_file.bin"
```

- To generate an INTEL hex (.hex) output file use:

```
arm-none-eabi-objcopy -O ihex "out_file.elf" "out_file.hex"
```

- To print information about code size after compilation use:

```
arm-none-eabi-size "out_file"
```



Restoring default configuration

resetting the perspective

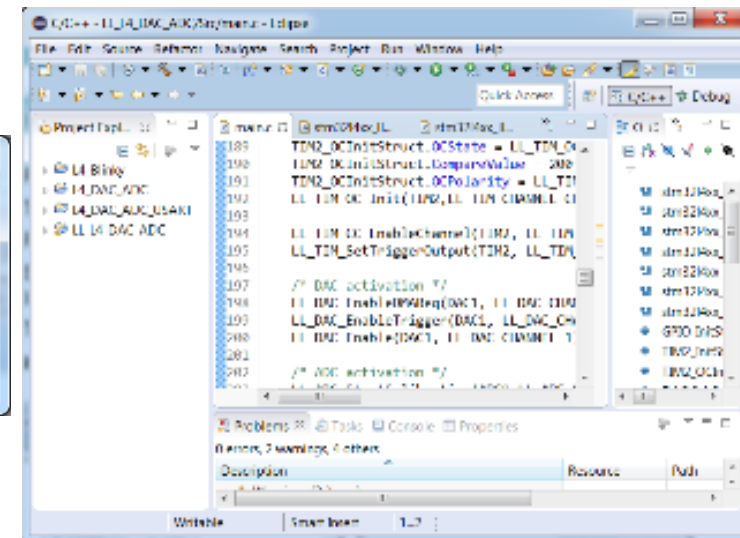
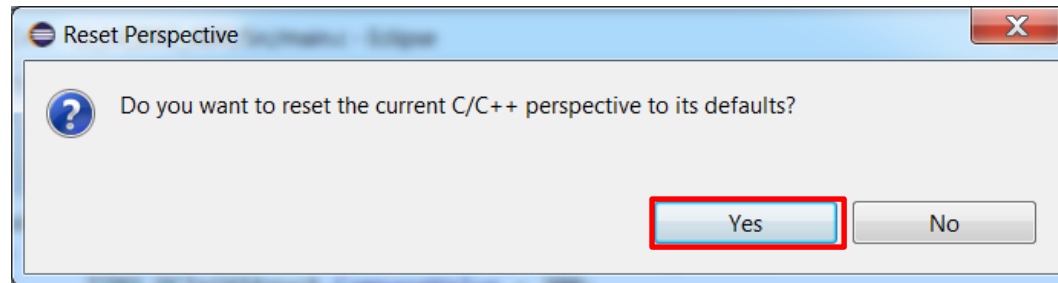
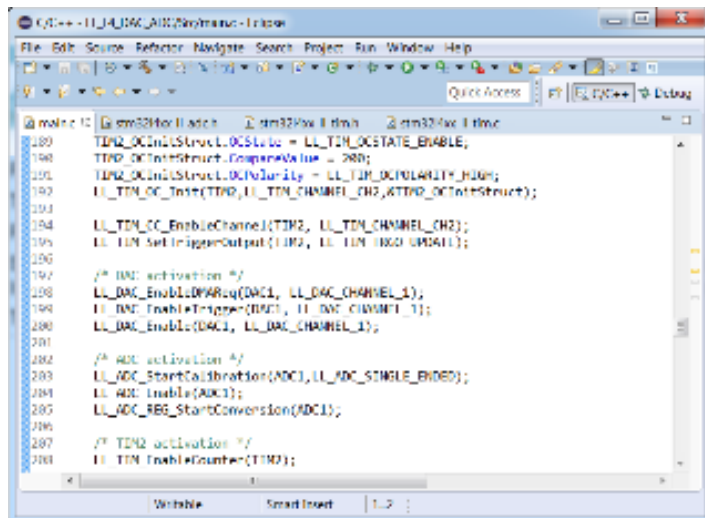
- Perspective is a configuration of the windows and toolbars within current SW4STM32 workspace.
- There are different perspectives for editing, debugging, file sharing etc.
- Perspectives can be modified by the user (Window→Perspective)
- There is a function available to reset the perspective to its default view.
- This option is available in Window→Perspective→Reset Perspective

Example:

Before (no project explorer nor output console windows)

Select Window→Perspective→Reset Perspective

After (project explorer and output console windows are restored)





Issues with entering debug session

There are few typical root causes of the issues when entering debug mode:

- Wrong version of the HW debugger selected: i.e. STLinkV2-1 instead of STLinkV2 or vice versa
- Already running debug session(s)
- STLink owned by other debug/programming application (i.e. STMStudio, STLink Utility)
- A “ghost” GDB process existing in PC memory and locking an access to the debug port
- Wrong connection type selected. Debugged MCU requires connect under reset option – it can be corrected within .cfg configuration file or by erasing the memory by STLink Utility application

Some PC oriented issues can be verified by connection trial using STLink Utility. Those are typically:

- Bad quality USB cable (especially when we use board with high current consumption)
- An issue with USB port (some USB3.0 ports show problems with correct operation with STLink)



In most of the cases root cause can be tracked by turn-on more verbose debug session (please refer to the next slide)

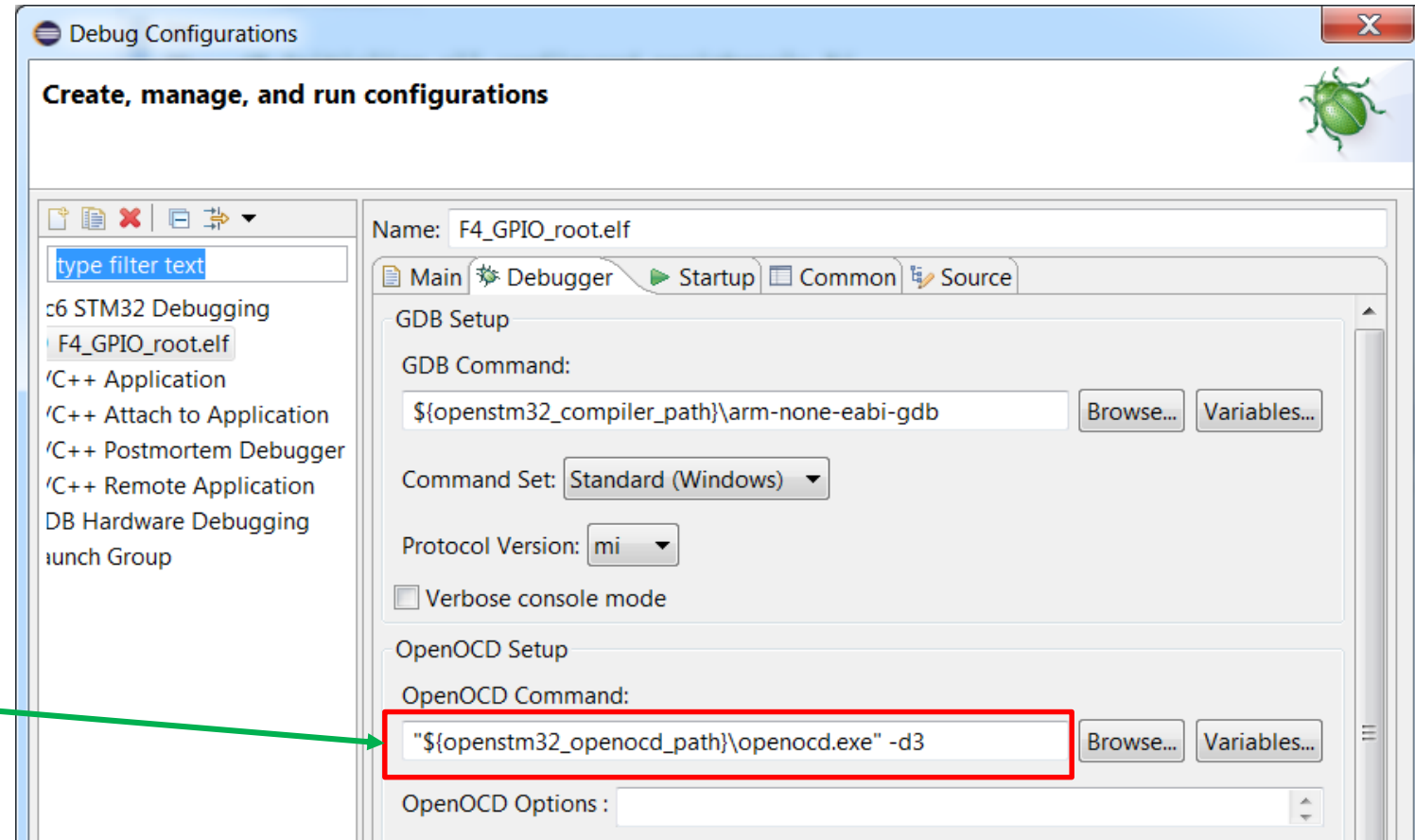


Issues with entering debug session

more verbose debug session

22

- An useful option is increasing debug level to **3** in OpenOCD configuration
- It generates much more messages concerning OpenOCD operations helping us to detect a root cause of the issue with debug session
- To do it, please add **-d3** option in OpenOCD Command within current debug configuration







Issues with entering debug session

detecting still running debug sessions

To detect still running debug sessions, please switch to debug perspective

1. If the  **Debug** button is not available, please click  button (or go to Window->Perspective->Open Perspective) and select “Debug” line form the list
2. Please have a look whether there are running debug sessions in the Debug window
3. To close them, they should be first terminated (**STOP** button) and then removed (**XX** button)
4. After removing all missed debug sessions, try to run debug session once again.

The image shows a sequence of four steps in an IDE:

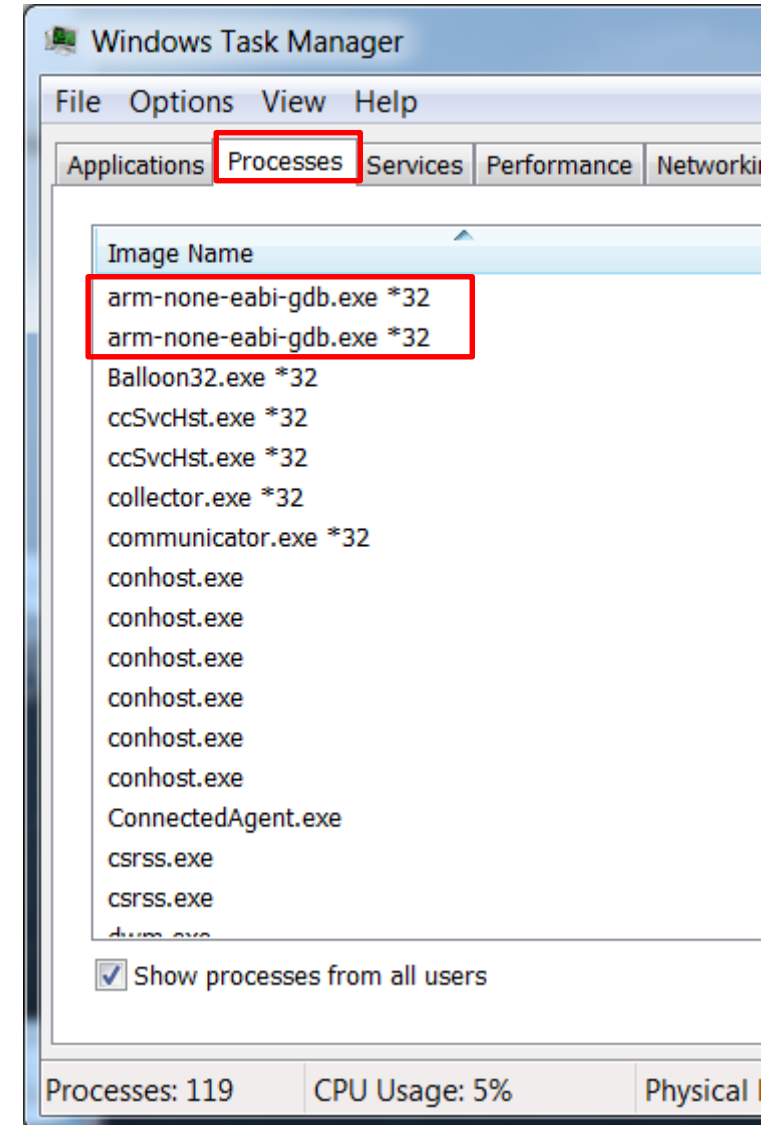
- Step 1:** The 'Open Perspective' dialog box is open, and the 'Debug' option is selected and highlighted with a red box.
- Step 2:** The 'Debug' perspective is active, showing a tree view of debug sessions. A session for 'L4_DAC_ADC.elf' is visible, with a thread 'Thread #1 (Suspended : Breakpoint)' at 'main() at main.c:90 0x8002b86'.
- Step 3:** The 'Terminate (Ctrl+F2)' button is highlighted in the top toolbar.
- Step 4:** The 'Debug' perspective is shown with a list of terminated sessions. A context menu is open over the list, showing options like 'Remove All Terminated Launches' and 'Add new'.



Issues with entering debug session

detecting “ghost” GDB processes in the system

- Display list of active tasks in the PC (Windows OS) by pressing **Ctrl+Alt+Del** and select “Start Task Manager”
- Select “Processes” tab from Windows Task Manager
- Highlight “ghost” **arm-none-eabi-gdb.exe** process
- Press **End Process** button





Issues with entering into debug session

detecting issue with busy STLink

25

- Configure more verbose debug session (**-d3** option)
- Run debug session and observe the messages displayed under **Console** tab
- Identify lines with **Error** attribute (first column in an example below)

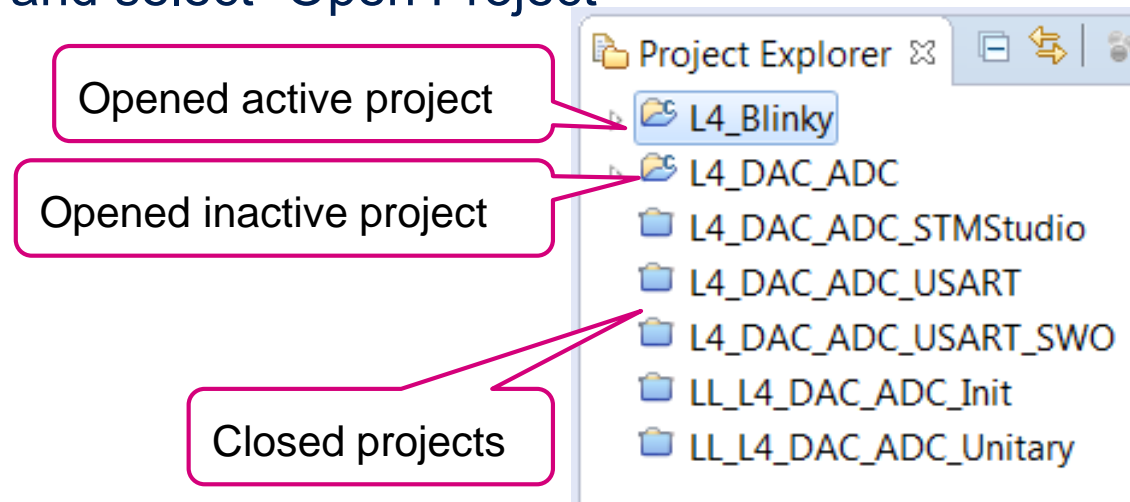
```
Problems Tasks Console Properties
<terminated> L4_DAC_ADC.elf [Ac6 STM32 Debugging] openocd
Debug: 255 89 core.c:1603 adapter_khz_to_speed(): have interface set up
Info : 256 89 core.c:1388 adapter_init(): clock speed 1800 kHz
Debug: 257 89 openocd.c:137 handle_init_command(): Debug Adapter init complete
Debug: 258 89 command.c:145 script_debug(): command - ocd_command ocd_command type ocd_transport init
Debug: 259 90 command.c:145 script_debug(): command - ocd_transport ocd_transport init
Debug: 261 90 transport.c:240 handle_transport_init(): handle_transport_init
Debug: 262 90 hla_transport.c:154 hl_transport_init(): hl_transport_init
Debug: 263 90 hla_transport.c:171 hl_transport_init(): current transport hla_swd
Debug: 264 90 hla_interface.c:44 hl_interface_open(): hl_interface_open
Debug: 265 90 hla_layout.c:42 hl_layout_open(): hl_layout_open
Debug: 266 90 stlink_usb.c:1698 stlink_usb_open(): stlink_usb_open
Debug: 267 91 stlink_usb.c:1716 stlink_usb_open(): transport: 1 vid: 0x0483 pid: 0x374b serial:
Error: 268 517 libusb1_common.c:97 jtag_libusb_open(): libusb_open() failed with LIBUSB_ERROR_ACCESS
Error: 269 517 stlink_usb.c:1729 stlink_usb_open(): open failed
Debug: 270 518 hla_layout.c:49 hl_layout_open(): failed
```



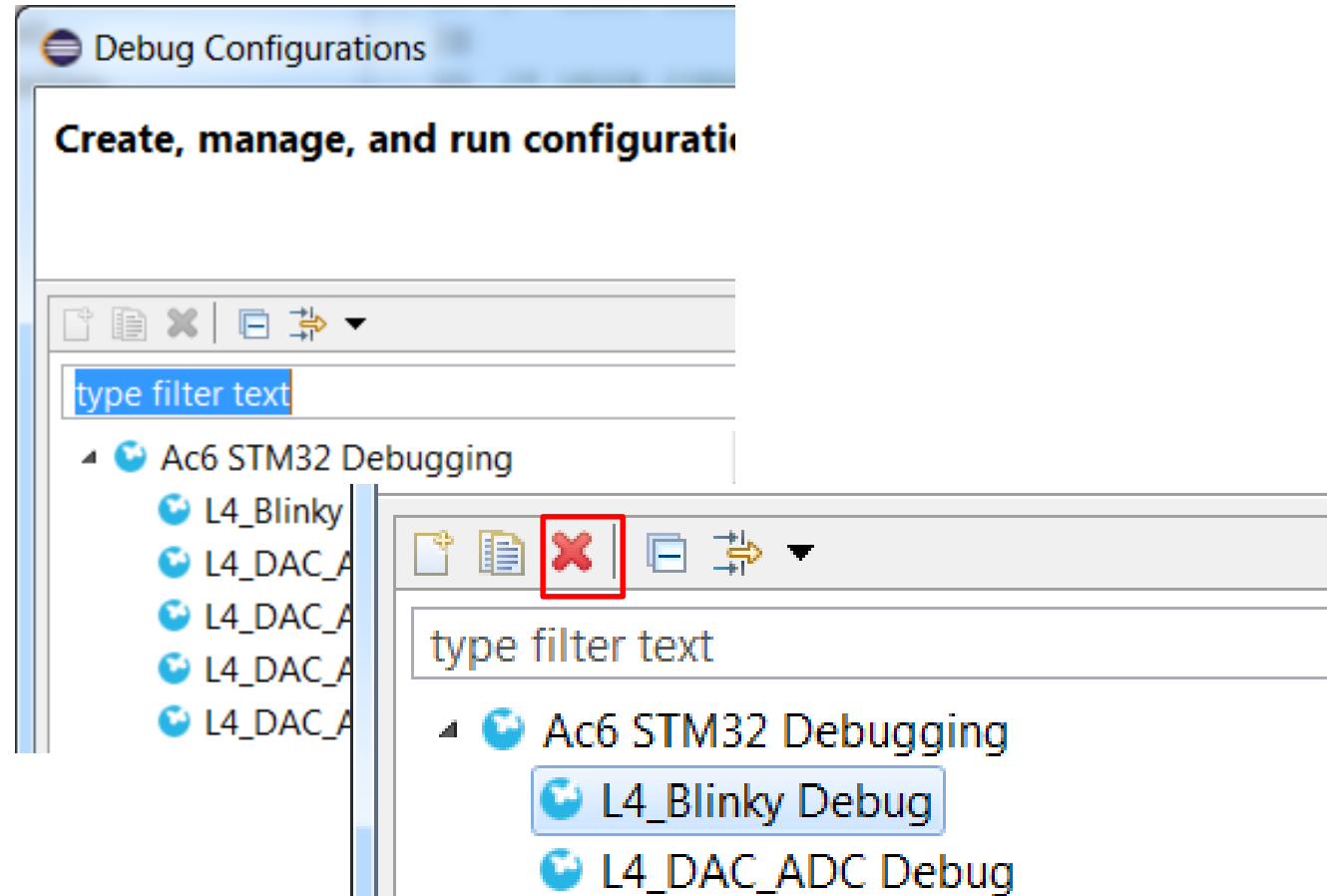
Managing multiproject workspace

closing and opening projects

- In the workspaces which contains more projects there is a risk of work on the sources from inactive project
- For this reason a good practice is to close inactive projects (it is not related to any project/file erasing).
- To close the project, please click right mouse button on project name and select “Close Project”
- To close all opened, but not active project, please click right mouse button on the active project name and select “Close Unrelated Projects”
- To reopen closed project, please double click left mouse button on its name or click right mouse button on project name and select “Open Project”



- In the workspaces which contains more projects there is a risk that we are running not correct debug session (by simple click on “bug” icon)
- To display available debug sessions within the workspace please use **Run->Debug Configurations** on click on small “down” arrow near the “bug” button
- To select a correct debug session, please click on its name
- To delete any debug configuration press red ‘X’ button





Our goals for this session

28

- ✓ Get knowledge about some features of SW4STM32 toolchain:
 - ✓ Selection C dialect and parallel build options
 - ✓ Configuration of the indexer
 - ✓ How to run an external application
 - ✓ How to set autosave before build option
 - ✓ How to use “come back later” messages within the code
 - ✓ How to use and configure autocompleting option
 - ✓ Managing different output files (.hex, .bin files generation)
 - ✓ How to run debug session and handling most common issues related to this operation

Enjoy!

 /STM32

 @ST_World

 st.com/e2e

www.st.com/mcu