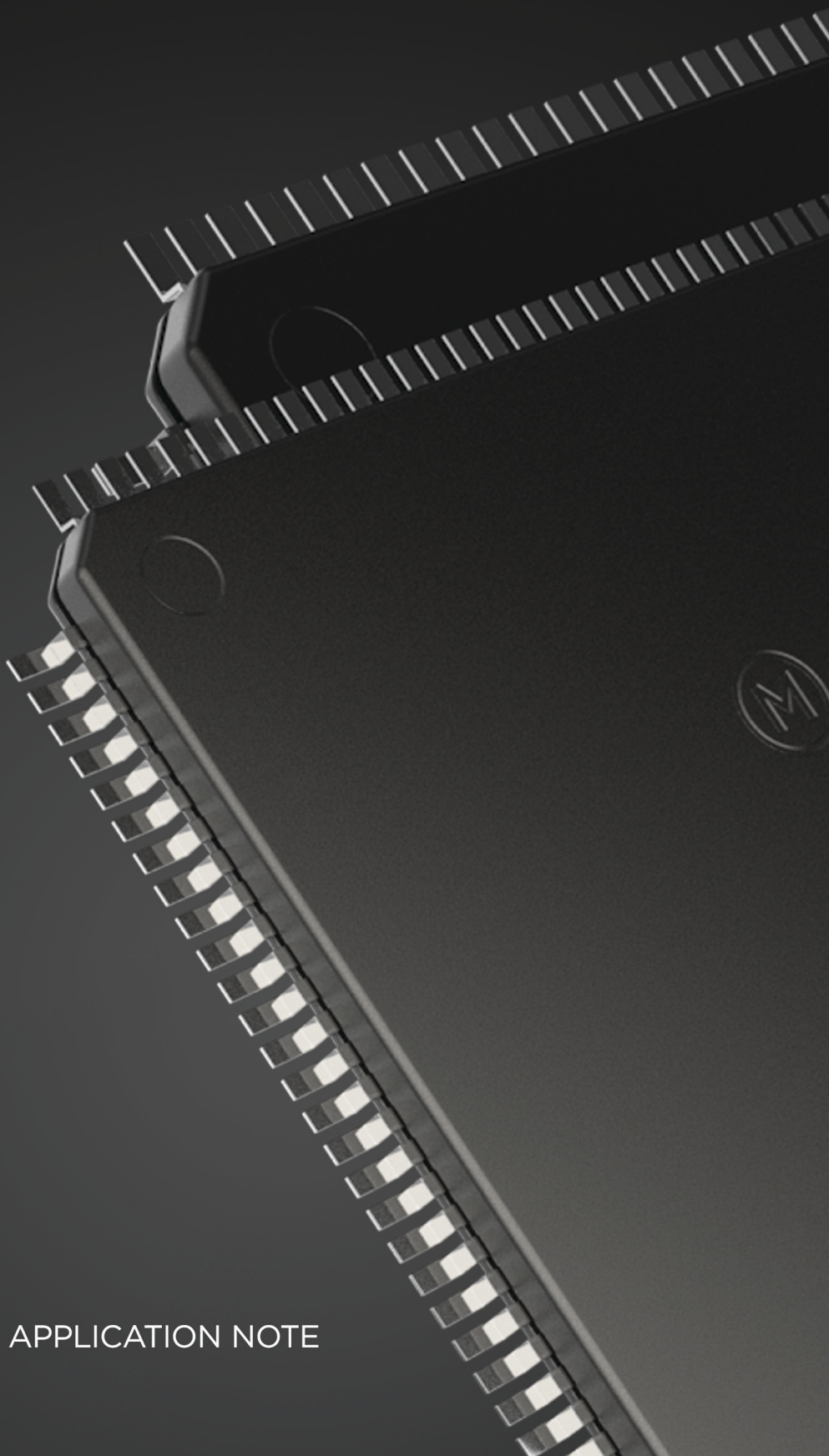
TASKING®

HOW TO MIGRATE FROM BIFACES TO TASKING IDE



APPLICATION NOTE

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

INTRODUCTION

The Build and Integration Framework for Automotive Controller Embedded Software (BIFACES) from Infineon, is an internally developed build framework for Infineon automotive micro-controllers' software development. This application note describes how you can migrate a project created with Infineon's BIFACES to the TASKING VX-toolset for TriCore Eclipse IDE.

The benefits of using the TASKING VX-toolset for TriCore Eclipse IDE instead of BIFACES is that you have full control over the tool options through the graphical Eclipse user interface. The integrated Pin Mapper eases pin status configuration. You also have direct access to the integrated debugger. Your complete development integrated environment all in one place.

In this application note we assume that you are familiar with BIFACES and already have it installed by following the BIFACES User Manual. We also assume you have installed the Base Projects you want to migrate from.

The following BIFACES downloads from `https://myicp.infineon.com` are assumed in this application note, but of course the same guidelines can be followed for newer versions:

Infineon download	Description
BIFACES_V1_0_2_Win32.zip BIFACES_V1_0_2_Win64.zip	BIFACES V1.0.2 for Windows 32-bit BIFACES V1.0.2 for Windows 64-bit
BaseProjects_AURIX1G_V1_0_1_11_0.zip	BIFACES Base Projects for AURIX devices based on iLLD V1.0.1.11.0. Supported devices: • TC21A • TC22A • TC23A • TC26B • TC27C • TC27D • TC29B
BaseProjects_AURIX2G_V1_0_1_11_0.zip	BIFACES Base Projects for AURIX 2G devices based on iLLD V1.0.1.11.0. Supported devices: • TC33A • TC33AED • TC35xA • TC36A • TC37xA • TC38A • TC39B

These versions work together with the TASKING VX-toolset for TriCore v6.3r1.

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

PREPARING THE BIFACES BASE PROJECTS

1. Install BIFACES (either the 32-bit or 64-bit version) and the BFACES Base Projects according to the BIFACES User Manual if you have not done so already.
2. Start BIFACES by double-clicking on `StartBifases.bat` in the `BifacesWin32` or `BifacesWin64` directory, depending on which version you installed.

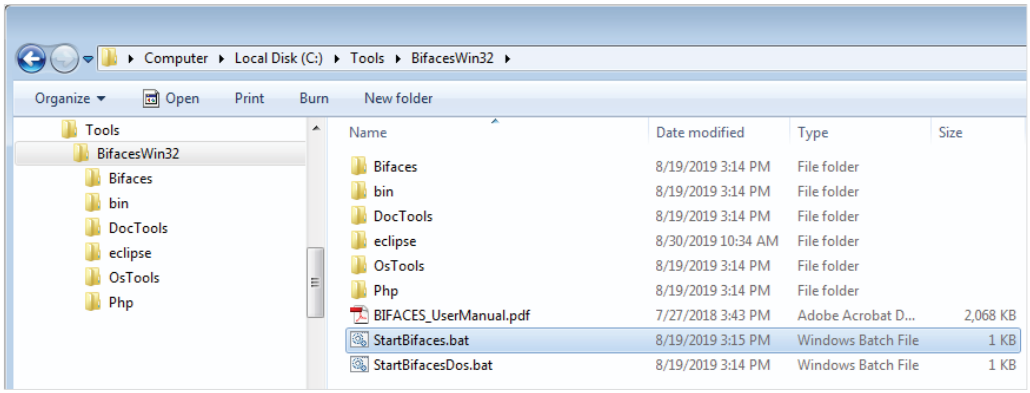


Figure 1: Start BIFACES

3. Select the workspace where you installed the Base Projects and click **Launch**:

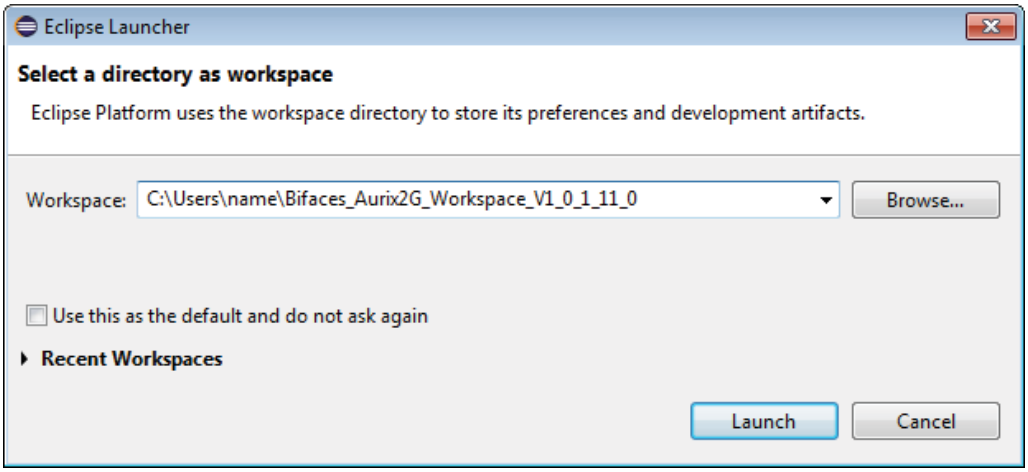


Figure 2: Select workspace

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

4. Click **F5** or select **File -> Refresh** to refresh the content of the Project Explorer view.
5. Edit file `Config_Tasking.mk` in `BaseFramework_TC39B\1_ToolEnv\0_Build\1_Config\Config_Tricore_Tasking` and change the variable `B_TASKING_TRICORE_PATH` to the path where your TASKING VX-toolset for TriCore v6.3r1 is located. For example:

   ```
   B_TASKING_TRICORE_PATH= C:\TriCore_v6.3r1\ctc
   ```

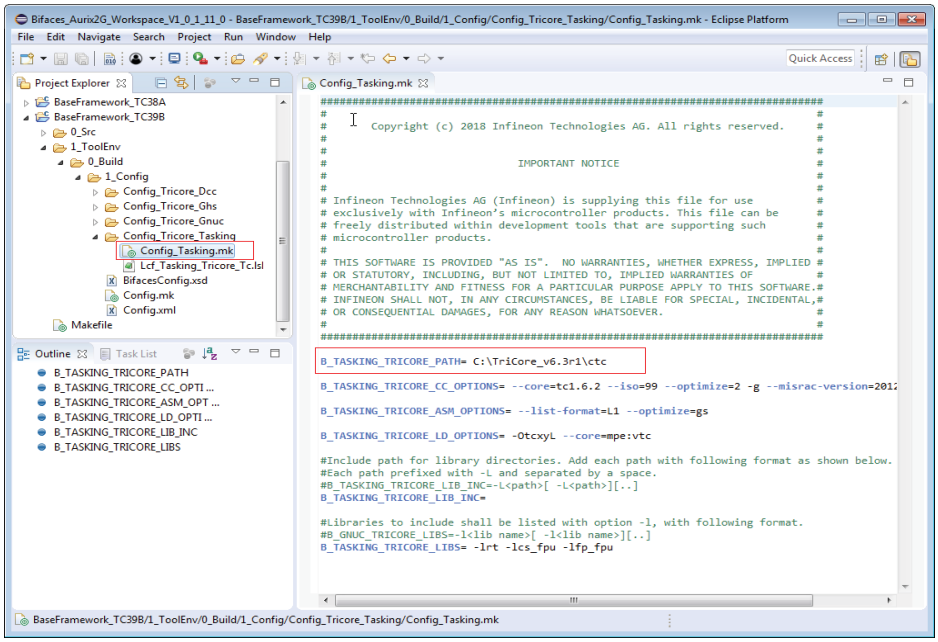


Figure 3: Change path to TASKING VX-toolset for TriCore

6. Edit `Config.xml` in `BaseFramework_TC39B\1_ToolEnv\0_Build\1_Config` and make the following changes:
   a. Set `primaryToolchain` under `architecture` to `Tasking`.
   b. Set `enable` under the Tasking `toolchain` to `true` (this should already be the case).
   c. Set `enable` under all of the other `toolchain` entries to `false`.

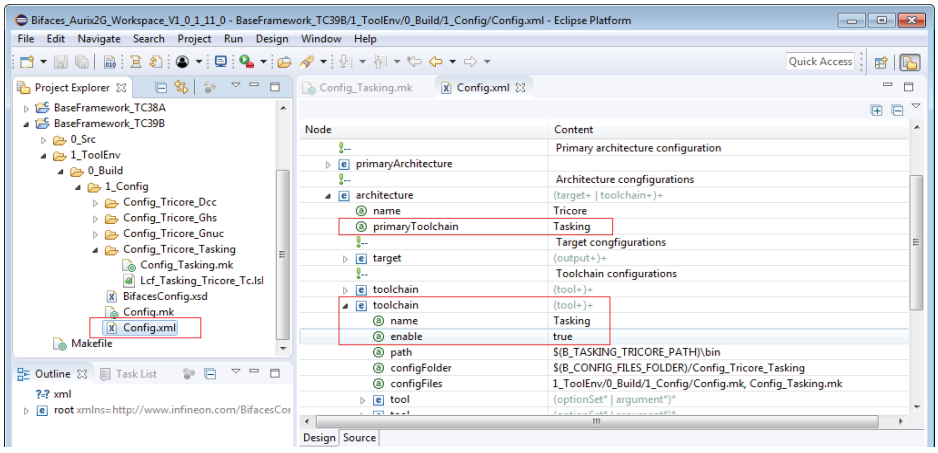


Figure 4: Edit Config.xml

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

7. Add a Build target.

   a. Double-click on `Makefile`.

   b. In the Outline view right-click on **all** and select **Add build Target**.

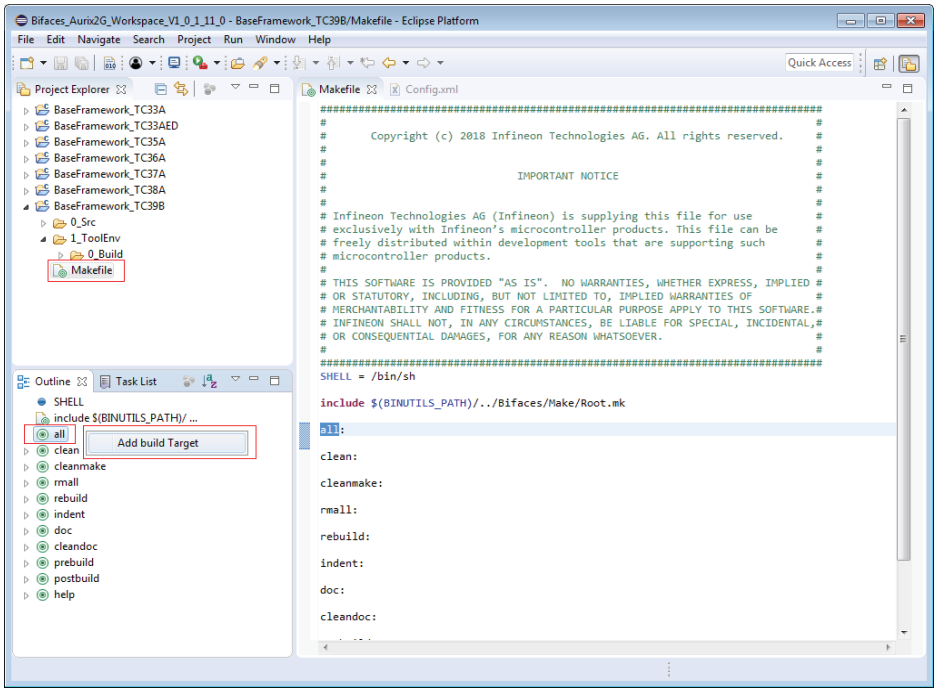


Figure 5: Add a build target

8. Double-click on **all** under the just created **Build Targets**.

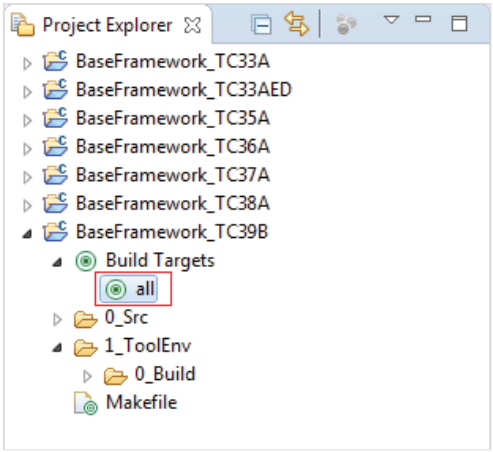


Figure 6: Build the project in BIFACES

The project will be built and the directory `9_Make` will be created which contains the file `Tricore_IncludePathList.opt` that is needed in the TriCore Eclipse IDE.

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

CREATE AN EMPTY PROJECT USING TRICORE ECLIPSE IDE

1. Start the TriCore Eclipse IDE.
2. From the **File** menu, select **New » TASKING TriCore C/C++ Project**.

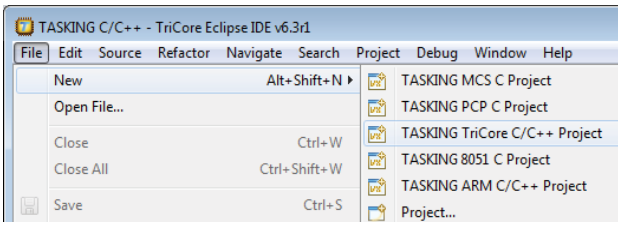


Figure 7: New TASKING TriCore C/C++ Project

3. Specify a project name (for example BaseFramework_TC39B), select **TASKING TriCore Application » Empty Project** and click **Next**.

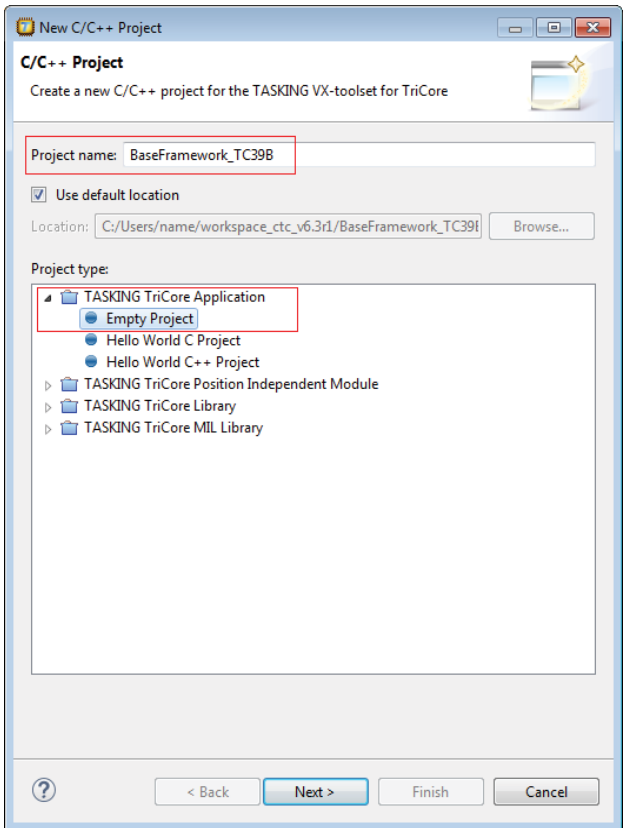


Figure 8: Project name / Empty Project

4. Select a processor. In this application note we select **TC39xB** from the AURIX 2G Family.
5. Disable the actions **Add startup file(s) to the project** and **Add linker script file to the project** and click **Finish**.

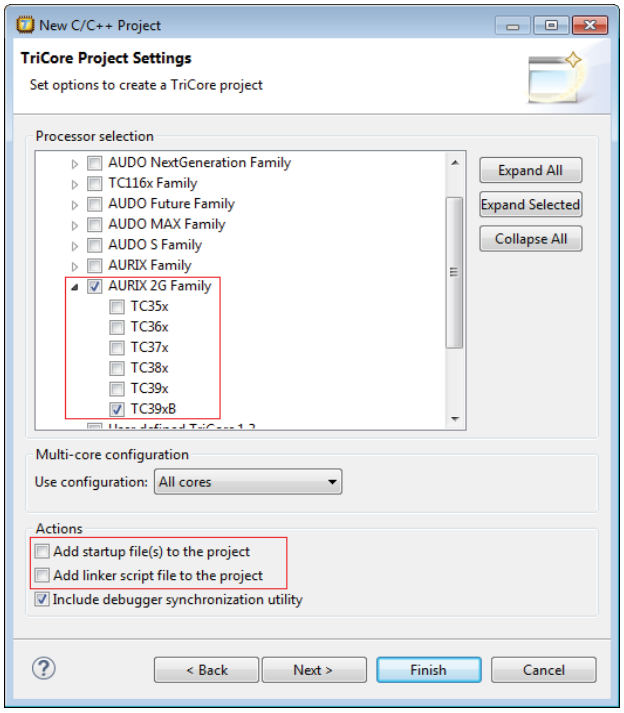


Figure 9: Project settings

Your C/C++ Projects view should now look like this:

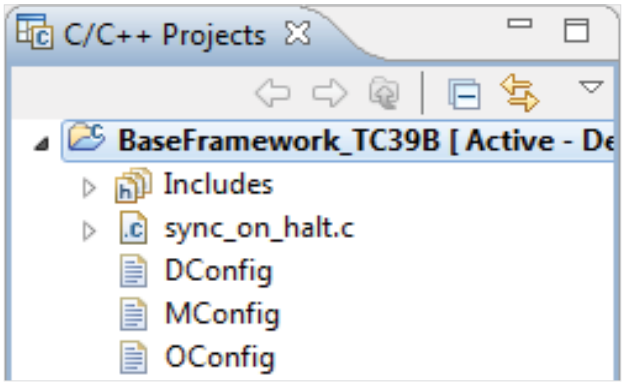


Figure 10: C/C++ Projects view

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

COPY DATA FROM BIFACES TO THE TASKING ECLIPSE IDE PROJECT

1. Copy (Ctrl+C) `0_Src` from the BaseFramework project (`BaseFramework_TC39B\0_Src`) from BIFACES and Paste (Ctrl+V) it in the BaseFramework project in the TriCore Eclipse IDE.

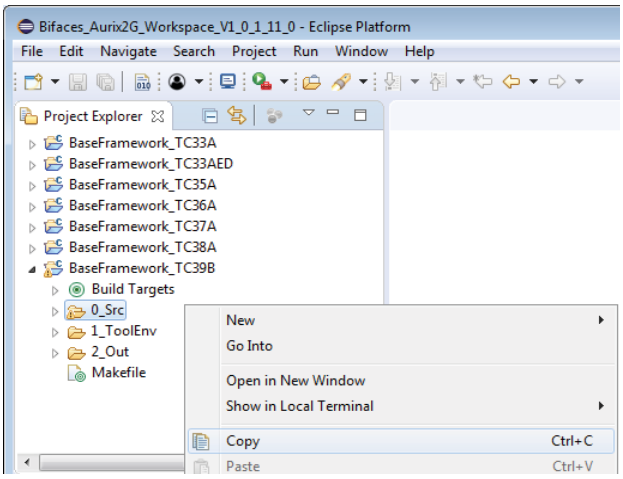


Figure 11: Copy 0_Src

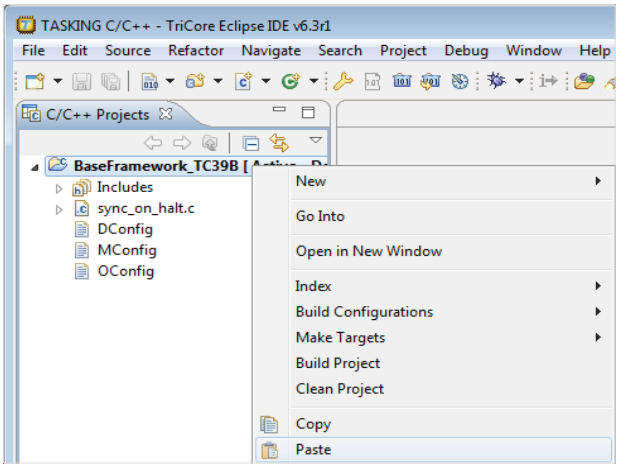


Figure 12: Paste 0_Src

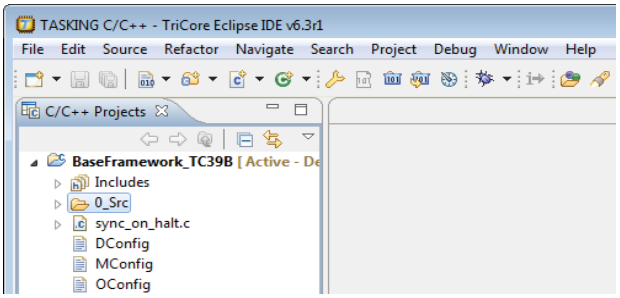


Figure 13: Result

2. Copy (Ctrl+C) `Lcf_Tasking_Tricore_Tc.lsl` from the BaseFramework project directory (`BaseFramework_TC39B\1_ToolEnv\0_Build\1_Config\Config_Tricore_Tasking`) and Paste (Ctrl+V) it in the BaseFramework project in the TriCore Eclipse IDE.

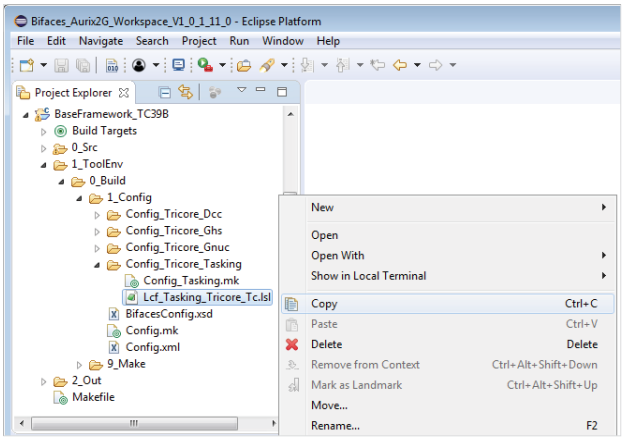


Figure 14: Copy LSL file

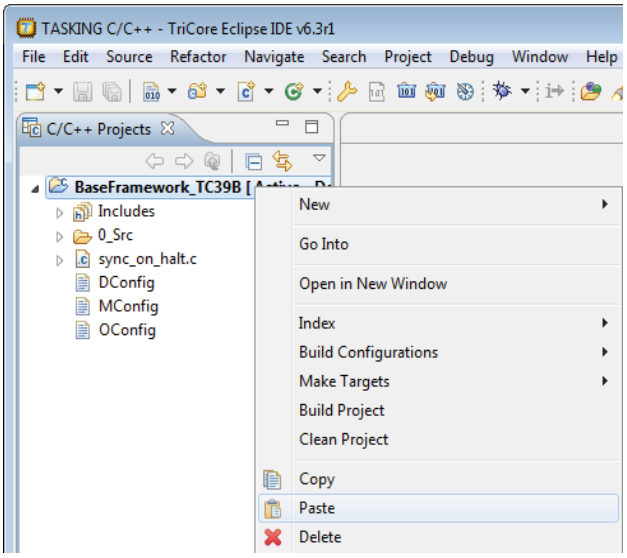


Figure 15: Paste LSL file

3. Copy (Ctrl+C) `Tricore_IncludePathList.opt` from the BaseFramework project directory (`BaseFramework_TC39B\1_ToolEnv\0_Build\9_Make`) and Paste (Ctrl+V) it in the BaseFramework project in the TriCore Eclipse IDE.

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

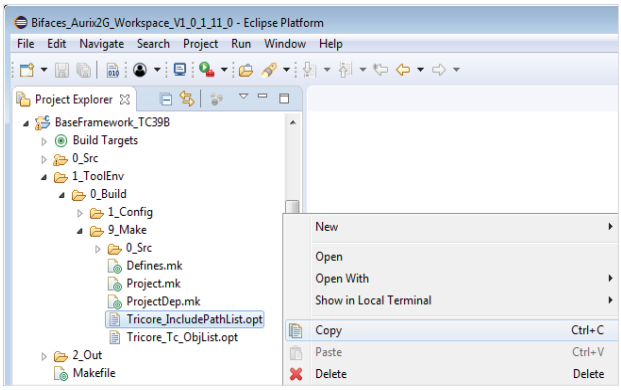


Figure 16: Copy include path list

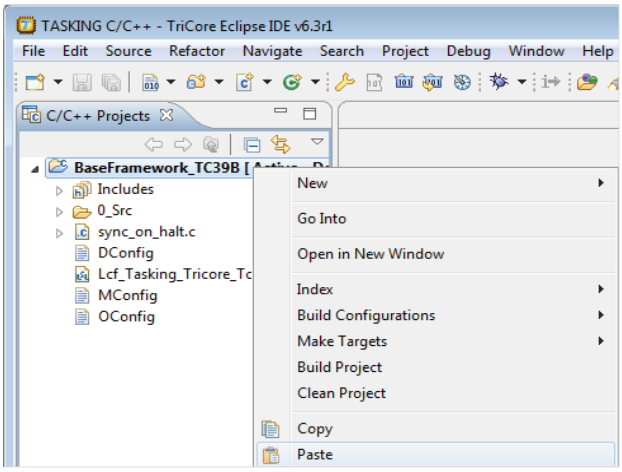


Figure 17: Paste include path list

ADAPT THE TASKING ECLIPSE IDE PROJECT

1. Open `Tricore_IncludePathList.opt` and change all `0_Src/` paths to `../0_Src/` and Save(Ctrl+S) the file.

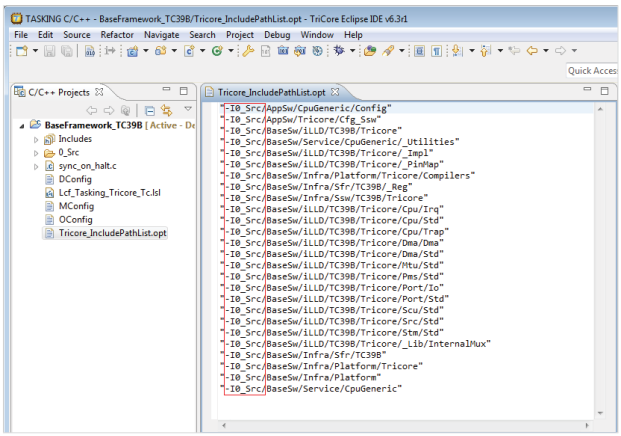


Figure 18: Open the options file

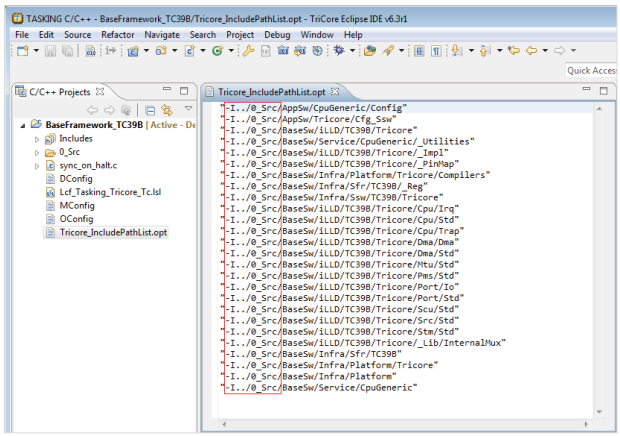


Figure 19: Add ../ to include path

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

2. Disable **Automatic inclusion of '.sfr'** file from the Properties dialog (**Project » Properties for** *project_name*, **C/C++ Build » Settings» Tool Settings » C/C++ Compiler » Preprocessing**).

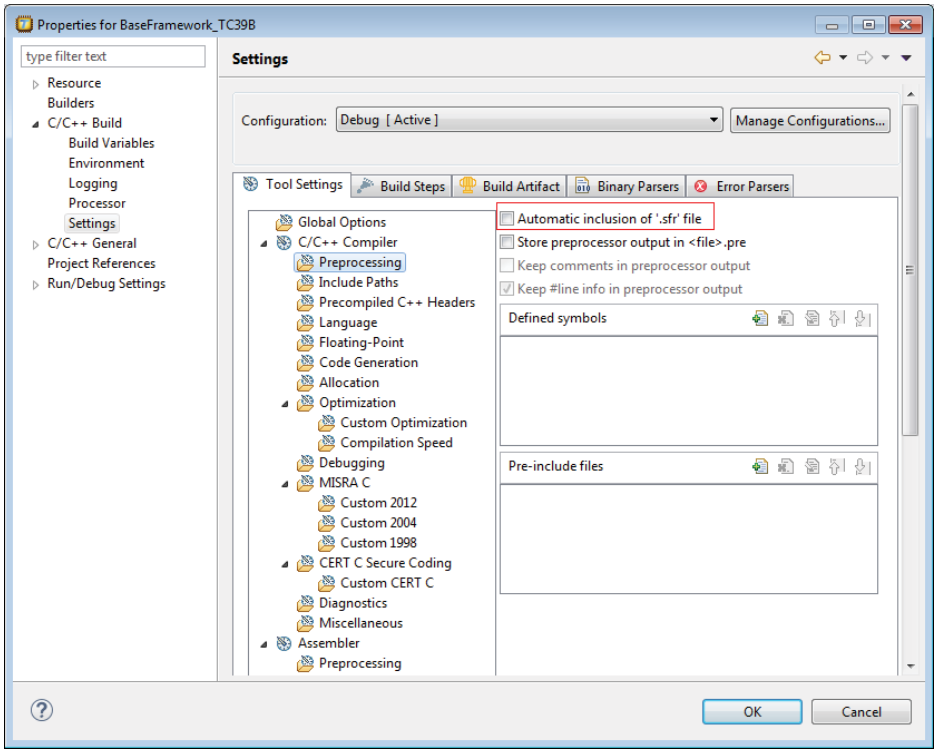


Figure 20: No automatic inclusion of SFR file

3. Add -f "*path_to_opt_file*\`Tricore_IncludePathList.opt`" to the **Additional optionsfield** (**Project » Properties for** *project_name*, **C/C++ Build » Settings» Tool Settings » C/C++ Compiler » Miscellaneous**), where *path_to_opt_file* is the absolute path to the option file.

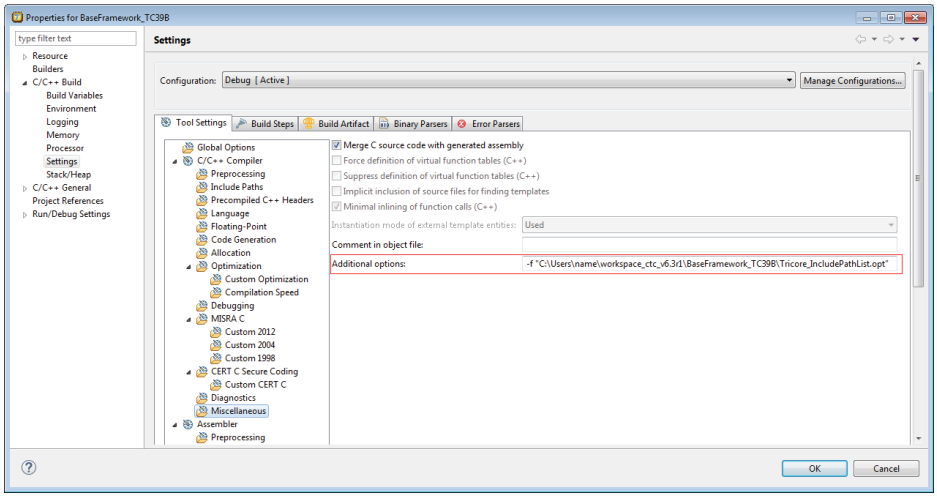


Figure 21: Add options file to the additional options

HOW TO MIGRATE FROM BIFACES TO TASKING IDE

4. Select `Lcf_Tasking_Tricore_Tc.lsl` as Linker Script File (**Project » Properties for** *project_name*, **C/C++ Build » Settings» Tool Settings » Linker » Script File**). You can use the **Browse** button to look for the file.

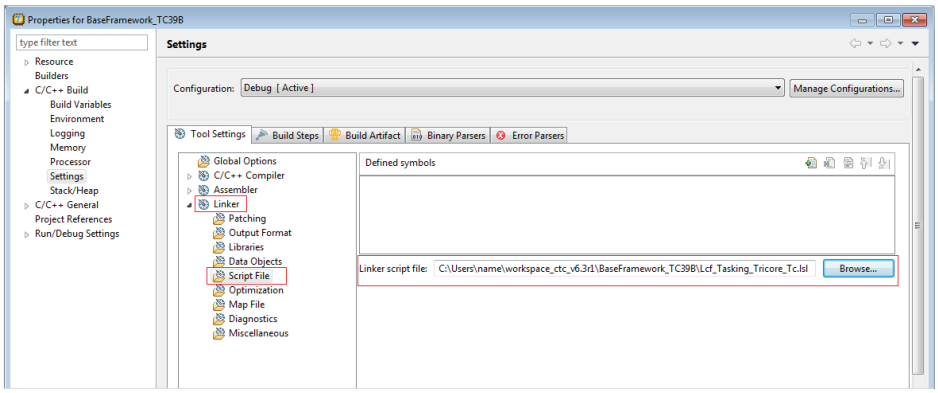


Figure 22: Specify LSL file

BUILD THE PROJECT

1. From the **Project** menu, select **Build** <*project-name*> (or select **Rebuild** <*project-name*> to do a rebuild).
2. Check if any errors occurred. If an error occurred with include paths, please check and adapt `Tricore_IncludePathList.opt` accordingly.

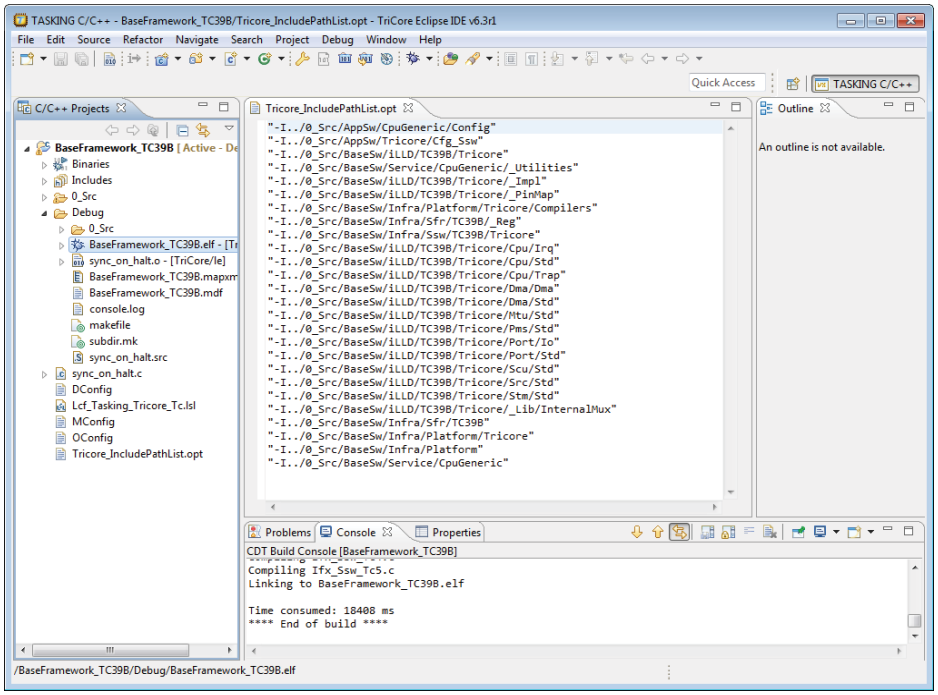


Figure 23: Build result

The project is now ready.

This concludes the Application Note on how to migrate from BIFACE to the TASKING TriCore Eclipse IDE.