ModusToolbox™ usage:
How to export to IAR Embedded Workbench

V1.0.0 2023-05

Scope of work

› ModusToolbox™ software includes a variety of ways to use applications with 3rd party tools.

› This document helps application developers understand how to export a ModusToolbox™ application to various supported IDEs in addition to the provided Eclipse IDE.

› The content of this document is divided into the following sections:

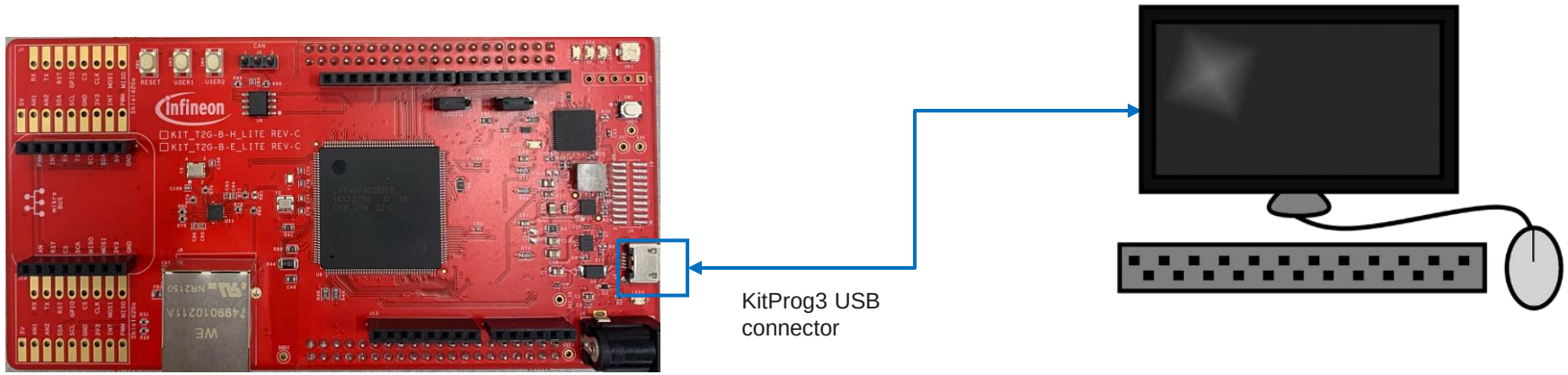
- Software environment
- Hardware connection
- How to export an application to IAR Embedded Workbench (Single-core)
- How to export an application to IAR Embedded Workbench (Multi-core)

Software environment

- **The software environment has the following features:**
  - ModusToolbox™ 3.0 software and application
  - Python 3.8
  - IAR Embedded Workbench version 9.30.1 or later
  - TAVEO™ T2G KIT_T2G-B-H_LITE with KitProg3 FW

Hardware connection

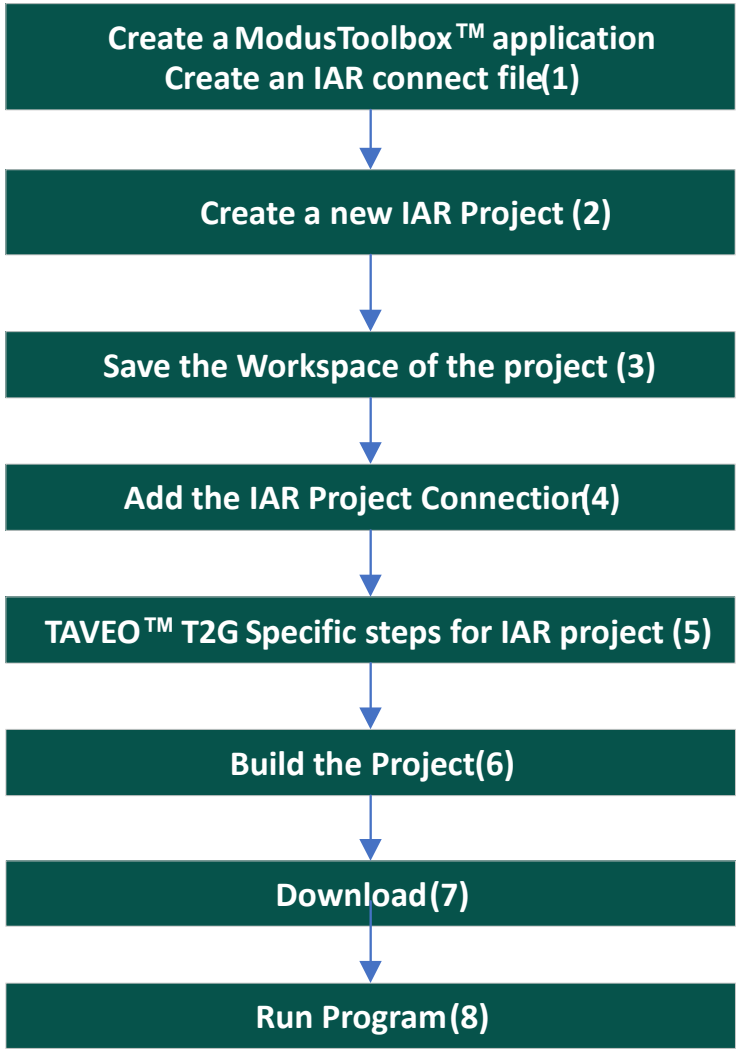
› **Connect the PC and kit board using an USB cable.**



How to export an application to IAR Embedded Workbench (Single-core)

How to export an application to IAR Embedded Workbench (Single-core)

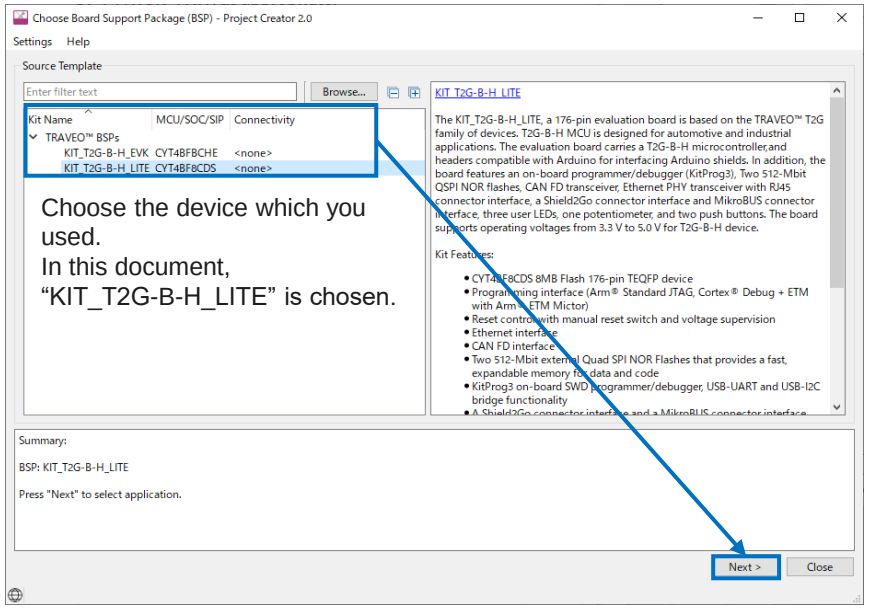
The following flow chart will show you a simple setup process for single-core debugging with IAR.

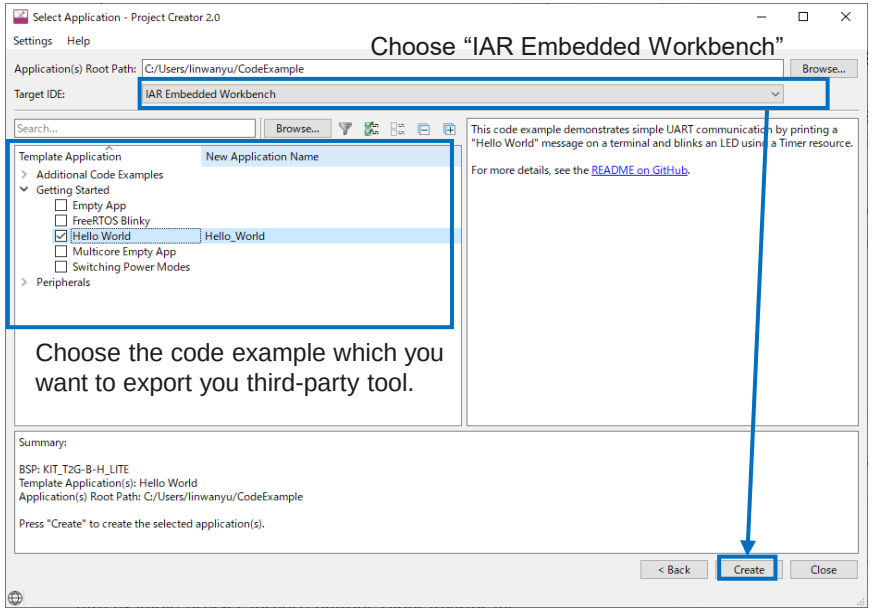


How to export an application to IAR Embedded Workbench (Single-core)

This section explains how to export and set up single-core debugging by IAR with CE “Hello World”.

1. **Create a ModusToolbox™ software application and an IAR connection file**
   - There are two ways to generate an IAR connection file.
     A) If you use the Project Creator tool, choose “*IAR Embedded Workbench”* from the ”*Target IDE”* pull-down menu. Refer to the blue box in the following figure. You can find “project-creator 2.0.0” in the Windows start menu.



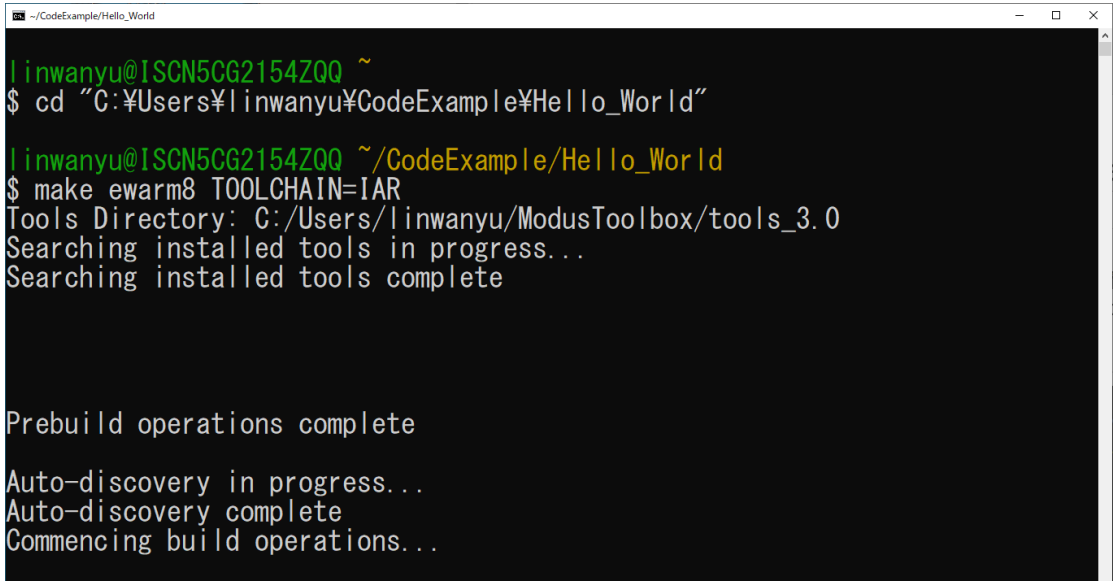


After successful creation, the IAR connection file (*Hello_World.ipcf*) appears in the application directory “\<application(s) Root Path>\<Application Name>”.

How to export an application to IAR Embedded Workbench (Single Core)

B) If you use the command line, open an appropriate shell program (see CLI Set-up Instructions), navigate to the application directory[1], and run the following command:

make ewarm8 TOOLCHAIN=IAR



B)-1. Open the shell program
B)-2. Change the directory, and the code example you want to export to a third-party tool.
B)-3. Enter the command “make ewarm8 TOOLCHAIN=IAR”

After successful creation, the IAR connection file *(mtb-example-hal-hello-world.ipcf)* appears in the application directory *“\<application(s) Root Path>\<Application Name>”*

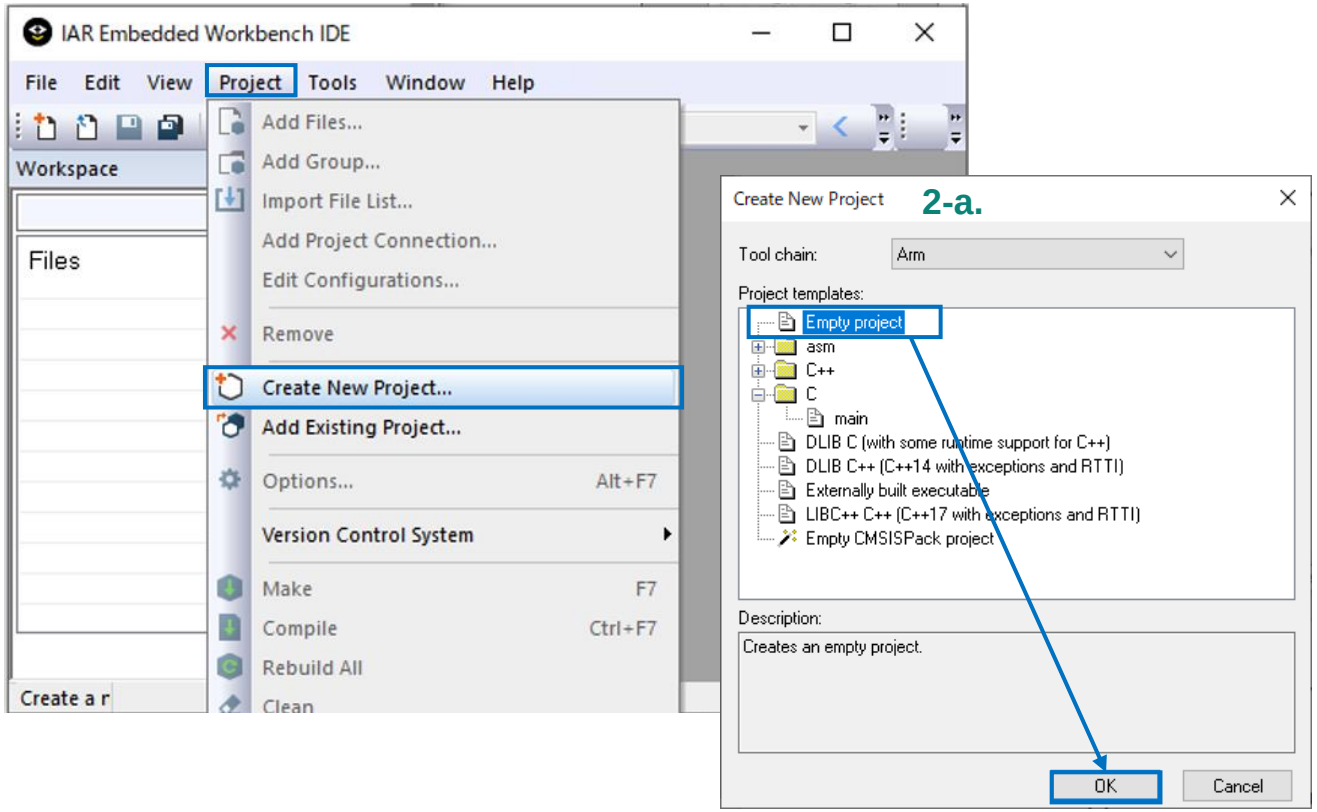
[1] This means the application was already created before, thus it can navigate to the application directory.

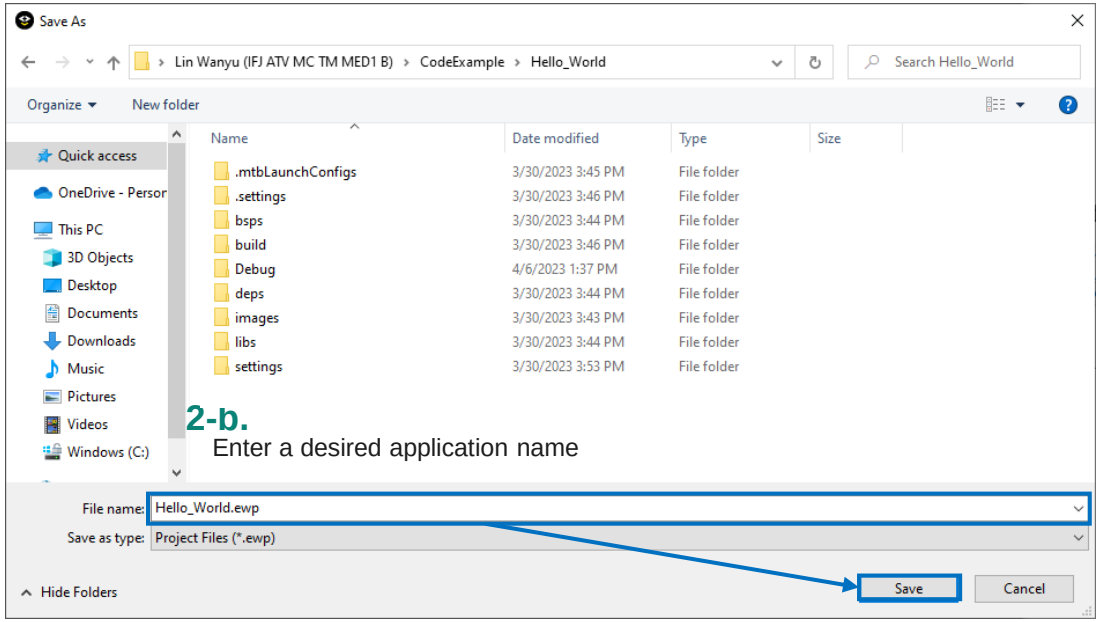
How to export an application to IAR Embedded Workbench (Single Core)

2. Create a new IAR project

a. On the main menu, select "***Project*** > ***Create New Project*** > ***Empty project***" and click "***OK***".

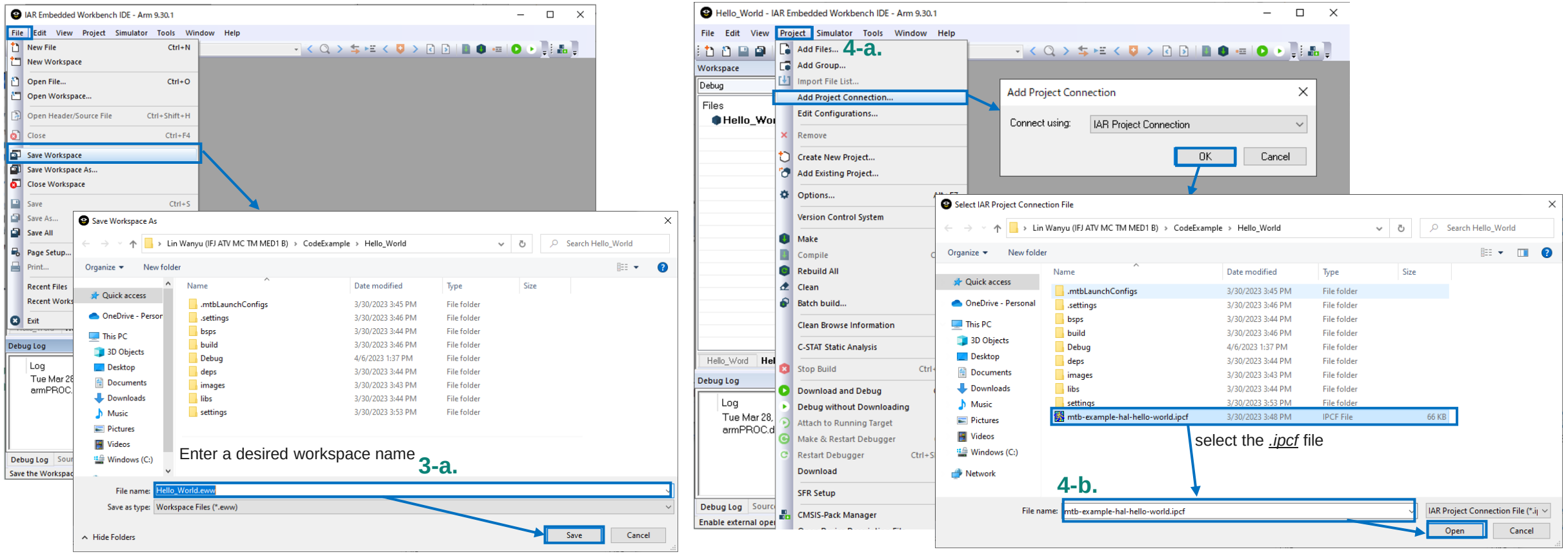
b. Browse to the application directory, enter a desired application name, and click "**Save**".





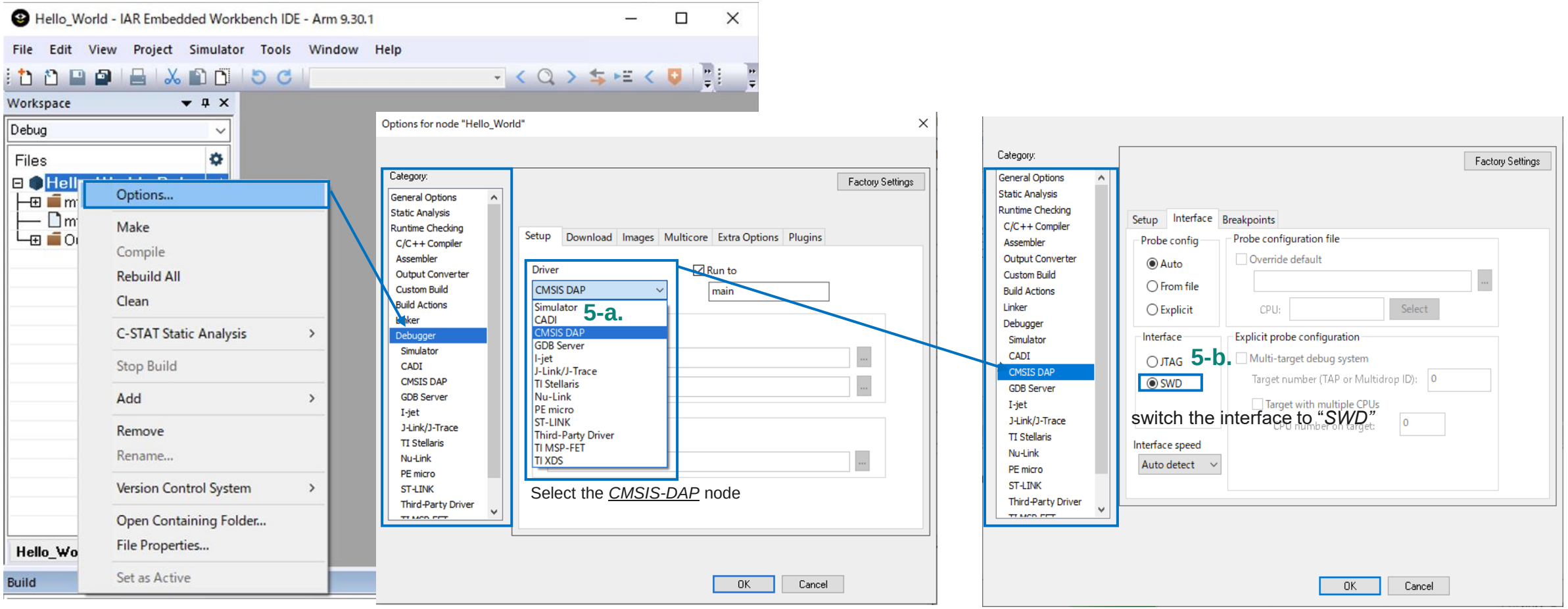
How to export an application to IAR Embedded Workbench (Single Core)

3. **Save the workspace of the project**
   a. Select *"**File** > **Save Workspace**". Enter a desired workspace name.*
4. **Add the IAR Project Connection**
   a. Select "***Project*** > ***Add Project Connection***" and click "***OK***".
   b. On the "*Select IAR Project Connection File*" dialog, select the "*.Ipcf* file" and click "***Open***".



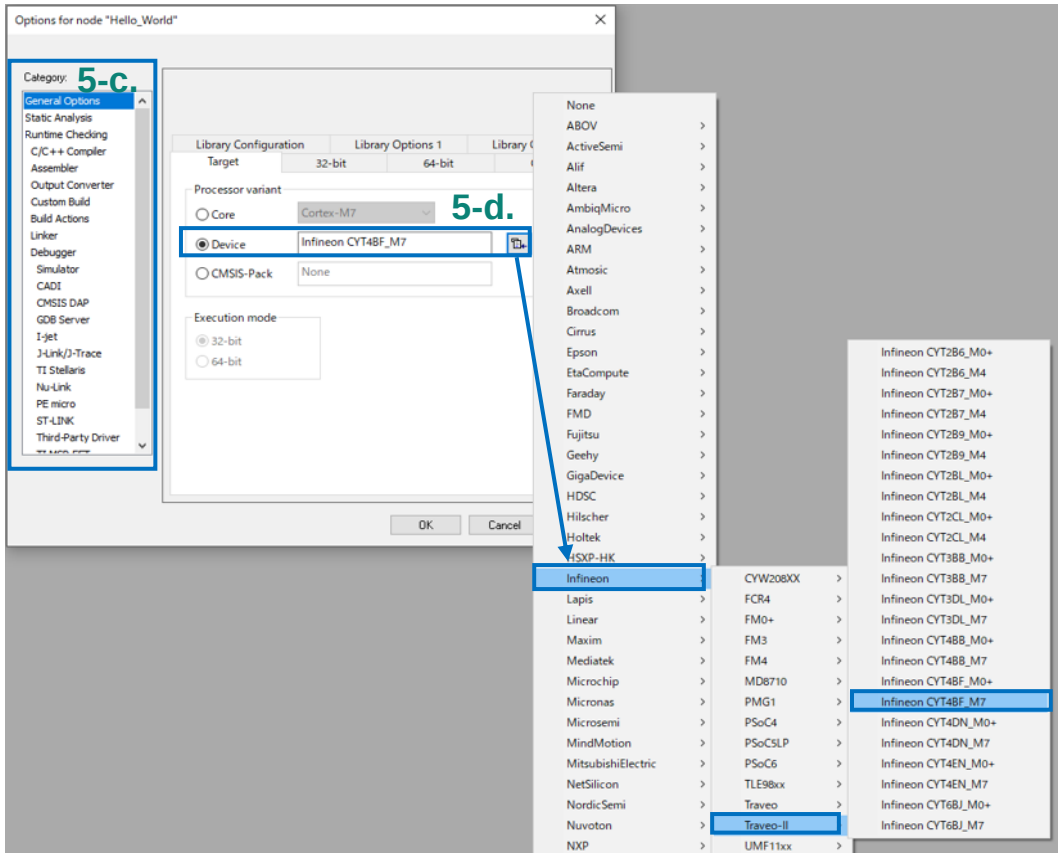
How to export an application to IAR Embedded Workbench (Single Core)

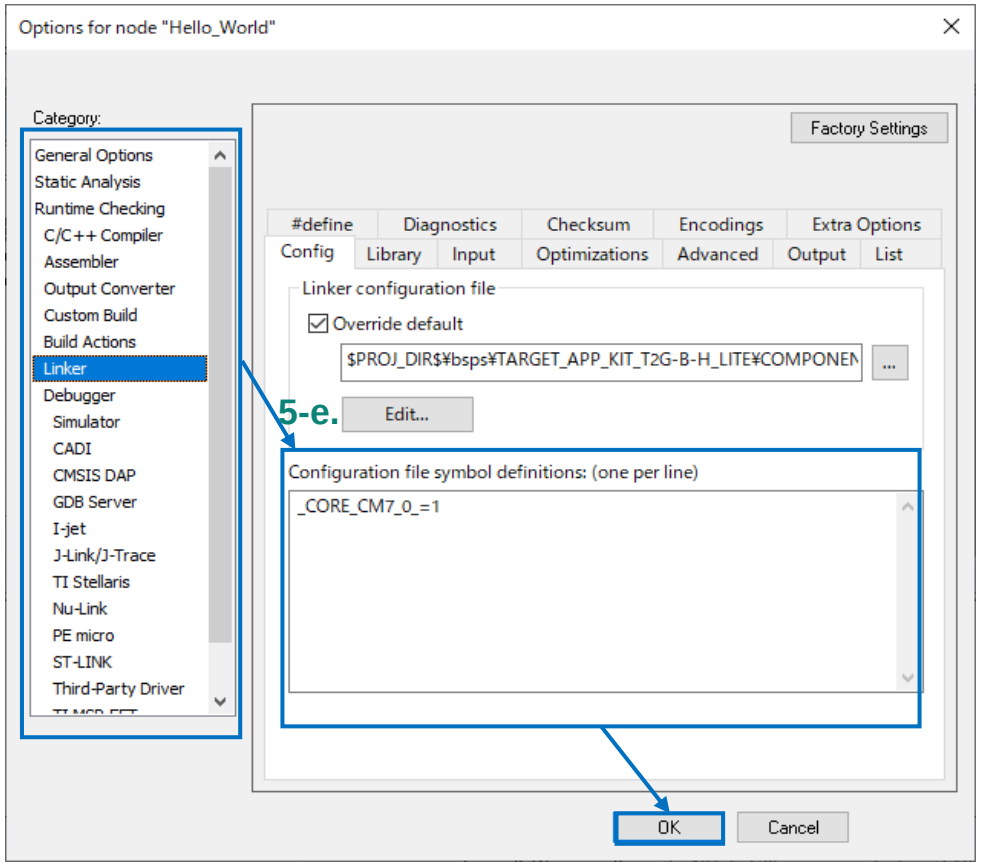
5. **TAVEO™ T2G-specific steps for the IAR project.**
   a. Go to "***Project*** > ***Options*** > ***Debugger***" and select "***CMSIS-DAP***" in the Drive list.
   b. Select the "*CMSIS-DAP*" node, and switch the interface from "*JTAG*" to "*SWD*".



How to export an application to IAR Embedded Workbench (Single Core) (contd.)

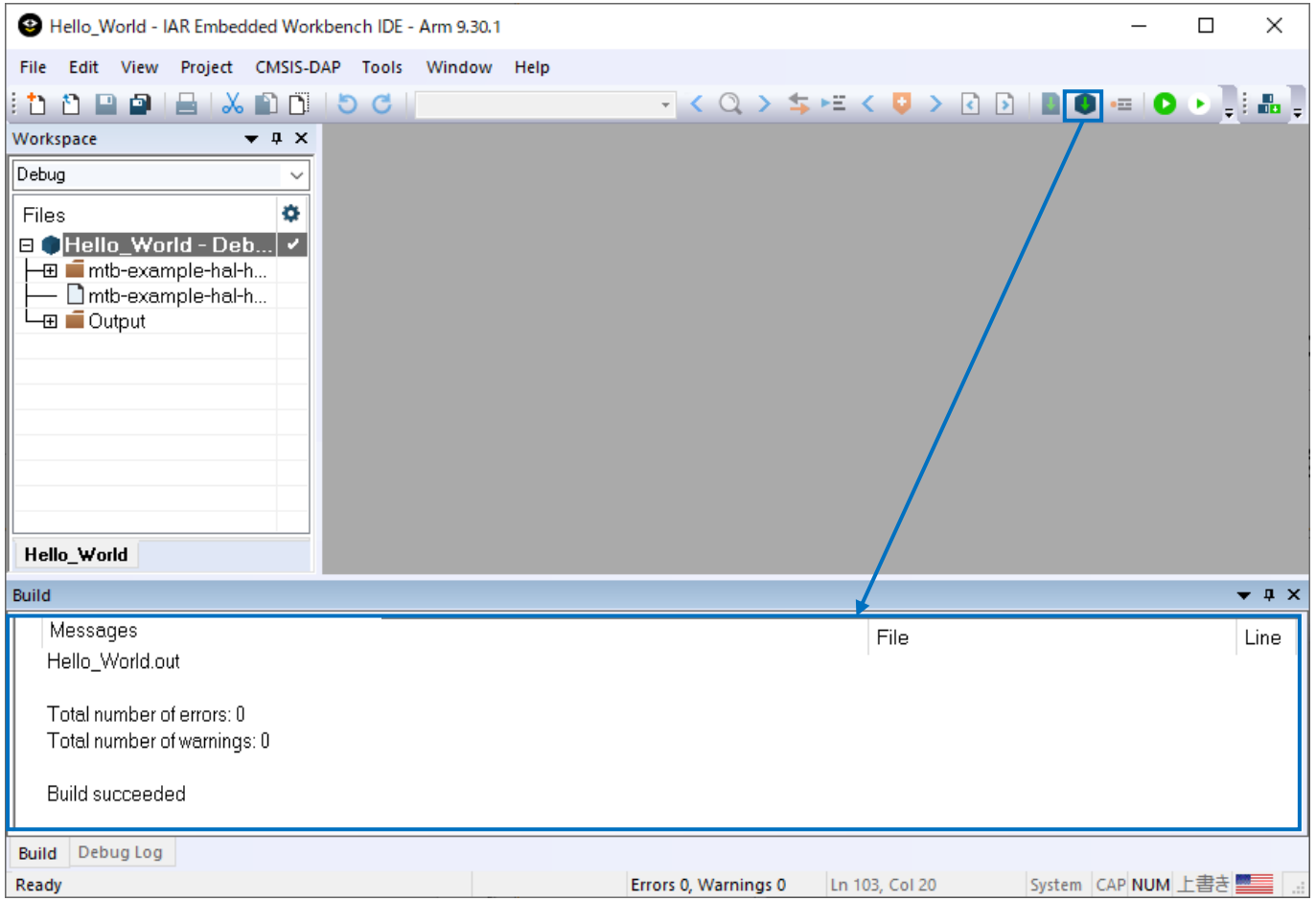
c. Go to "***General Options***".
d. Select the following device: "*Infineon > Traveo-II > Infineon CYT4BF_M7". Note that i*n some versions, "*Traveo-II"* shows as *"TRAVEO™ T2G".*
e. Go to "*Linker", a*dd "*_CORE_CM7_0_=1"* in the Configuration file symbol definitions field, and click the "***OK***".





How to export an application to IAR Embedded Workbench (Single Core)

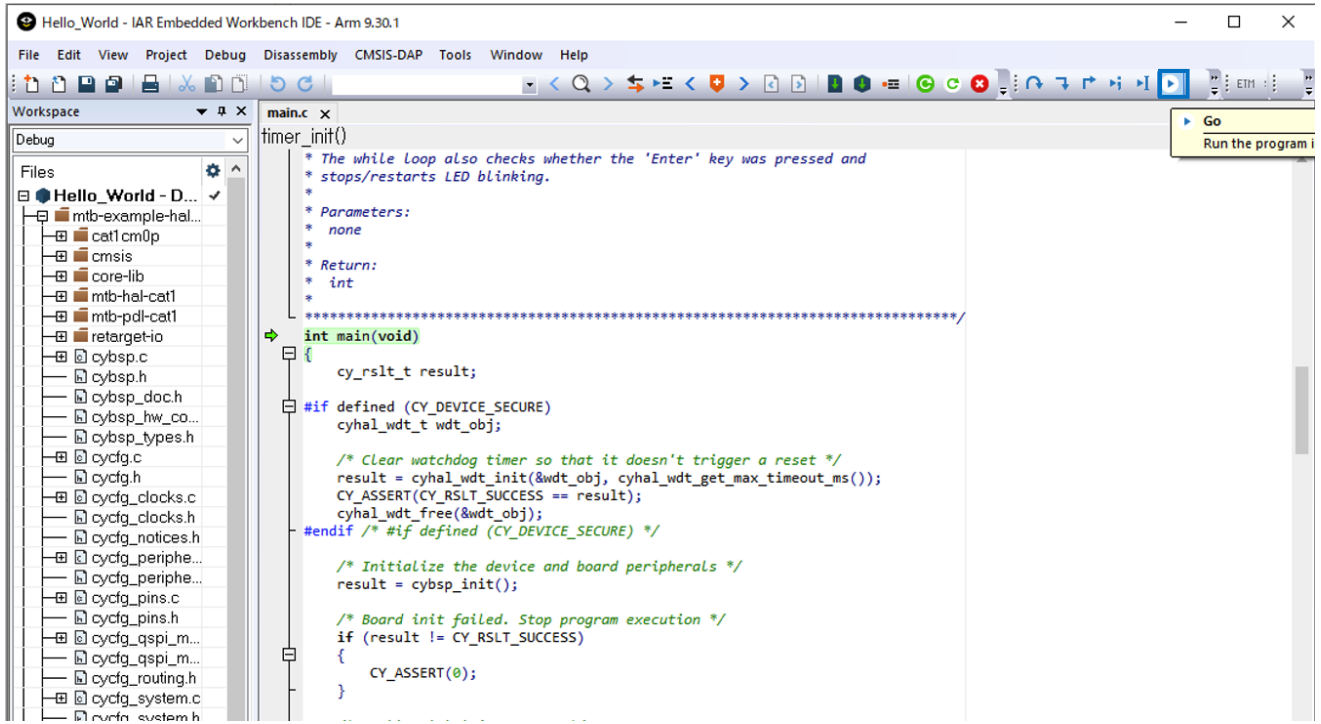
6. **On the main menu, click the "Make" button, and ensure there is no build error.**



How to export an application to IAR Embedded Workbench (Single Core)

7. **Connect to the host PC. Then, click "*Download and Debug*  " to download the application.**

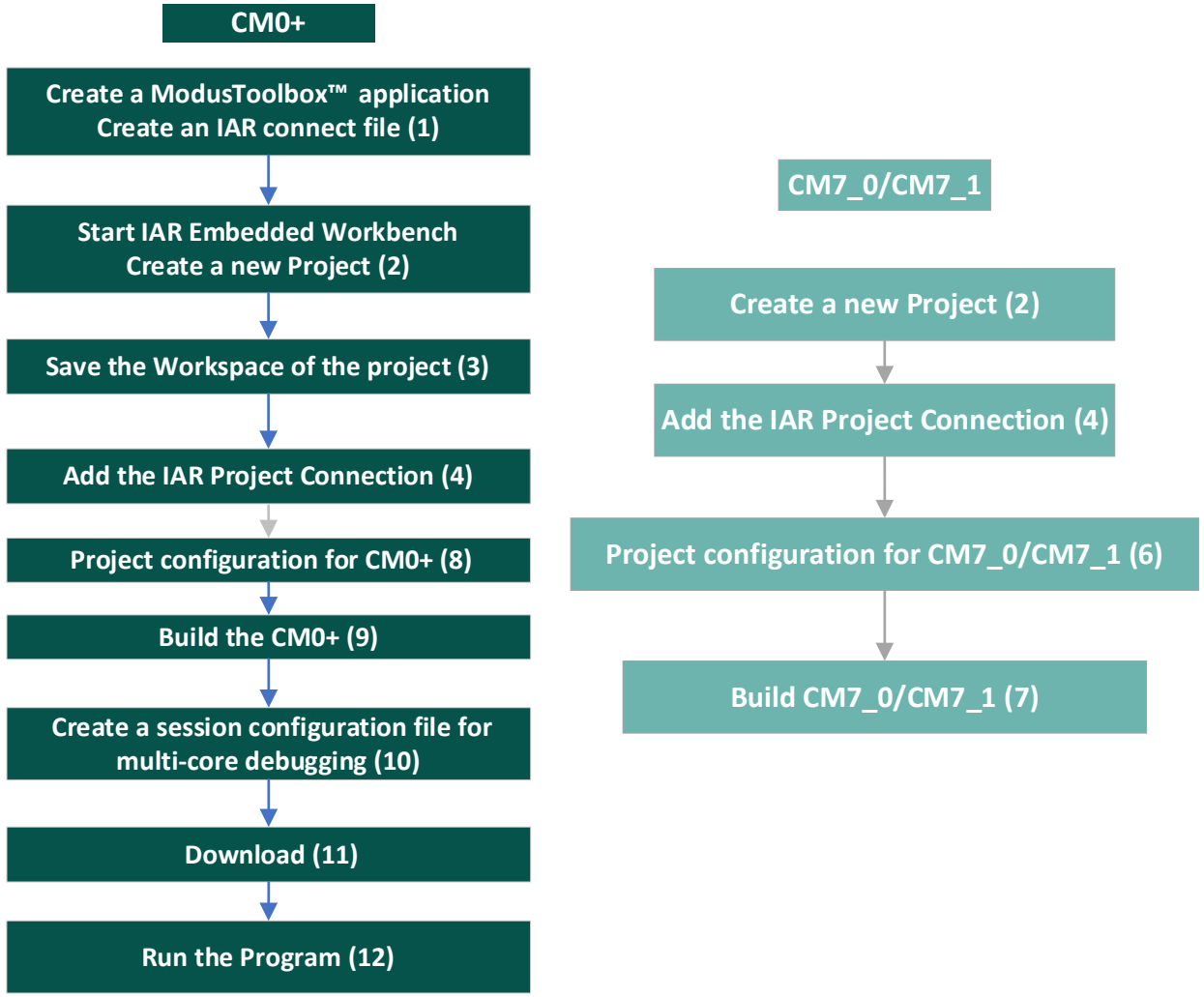
8. **Click "*Go* ' to run the program.**



How to export an application to IAR Embedded Workbench (Multi-core)

How to export an application to IAR Embedded Workbench (Multi-core)

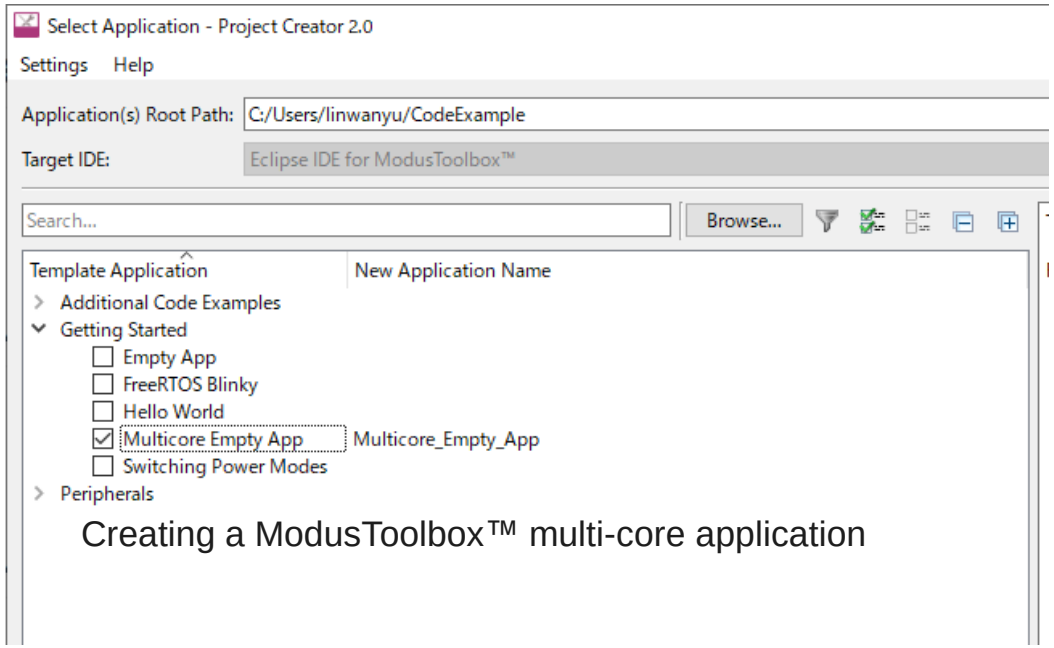
The following flow chart will show you a simple setup process for multi-core debugging with IAR.



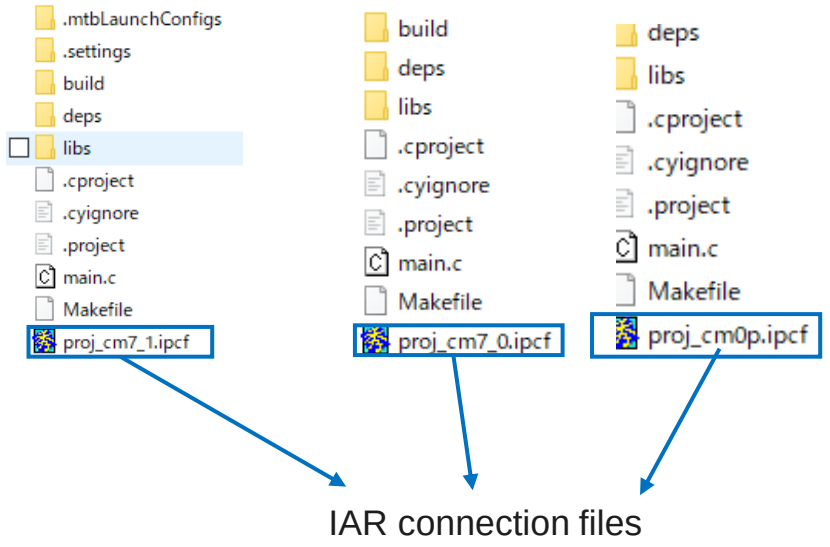
How to export an application to IAR Embedded Workbench (Multi-core)

The following steps will describe how to execute multi-core by IAR with CE “Multicore Empty App”.

1. **Creating[2] a multi-core application and IAR connection files: Each core needs an IAR connection file.**



Creating a ModusToolbox™ multi-core application



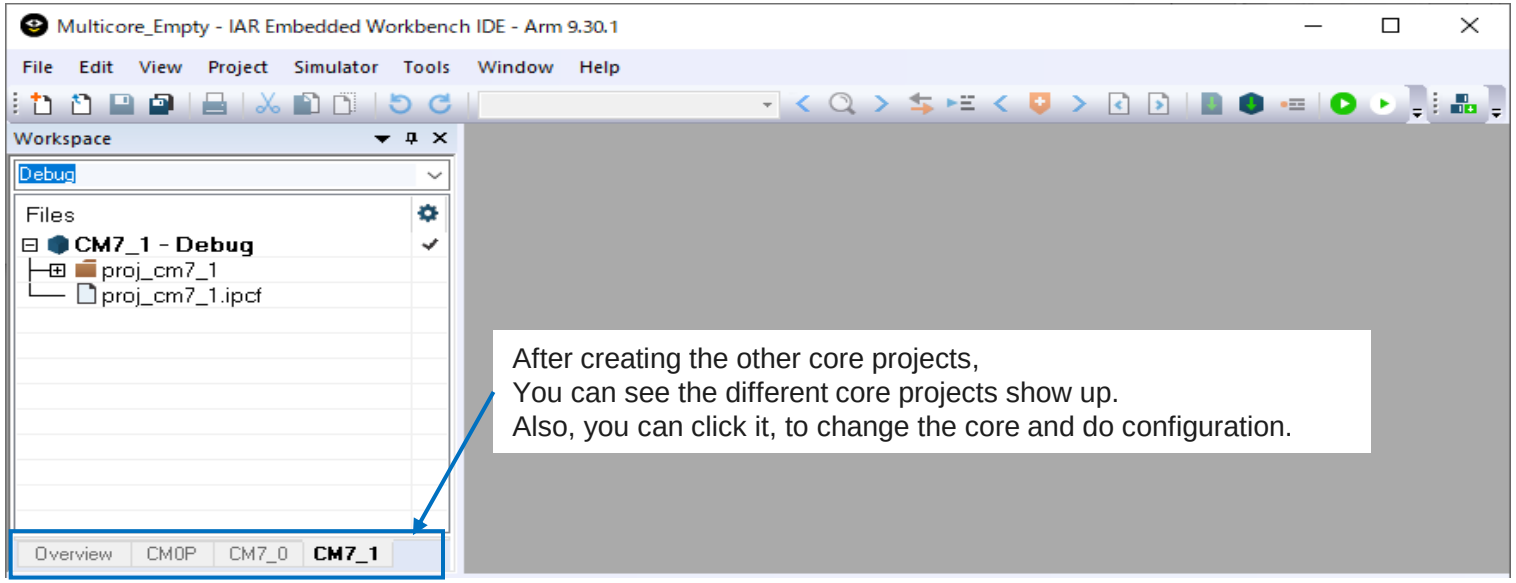
IAR connection files

[2] Please refer to the “How to export an application to IAR Embedded Workbench (Single-core)” It shows how to create a new application, and how to generate an IAR connection file. The name of generated ipcf file is different when using the Project Creator tool or Command Tool.

How to export an application to IAR Embedded Workbench

After IAR connection files are created in each core folder, do the following steps:

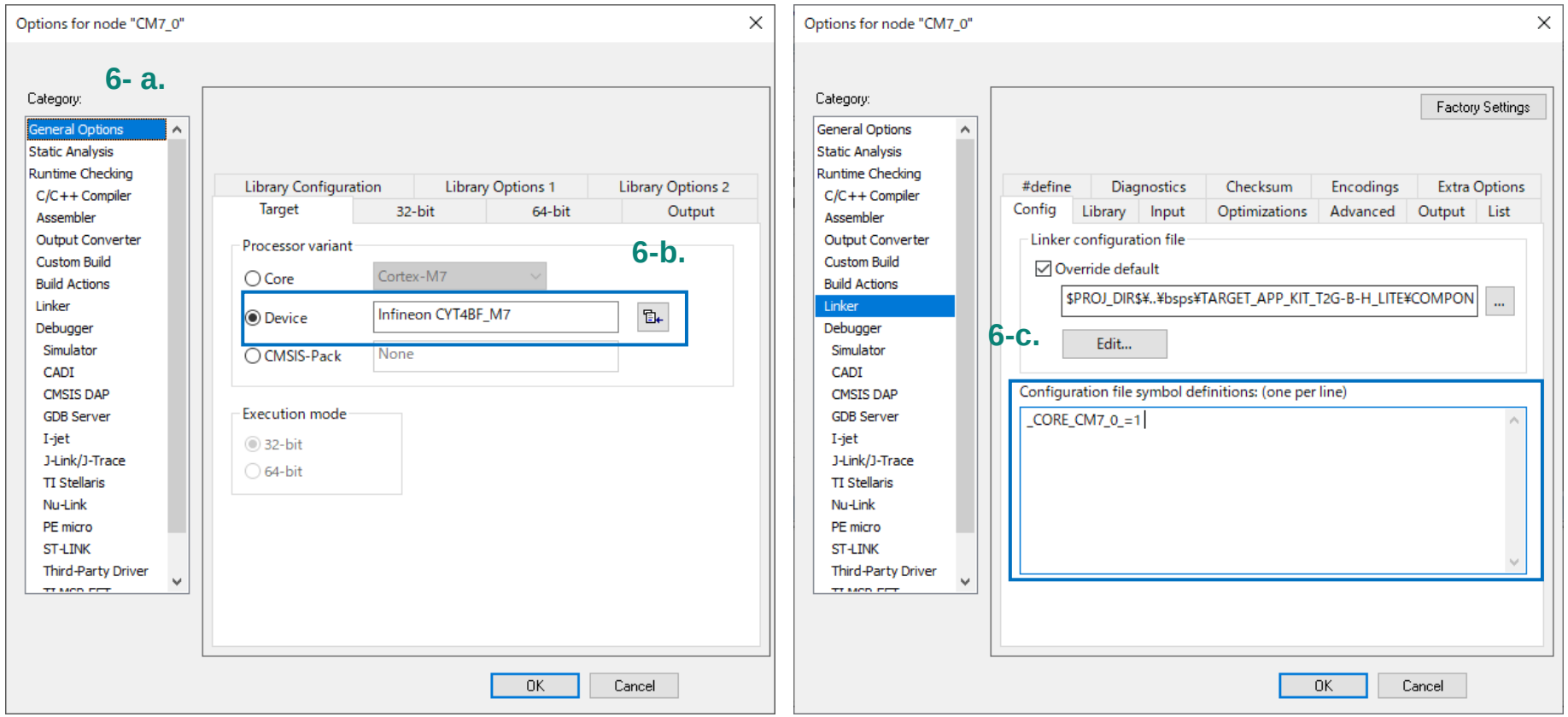
2. **Launch IAR and create[3] a new IAR project**
   a. On the main menu, select "***Project*** > ***Create New Project*** > ***Empty project***" and click "***OK***".
   b. Browse to the application directory, enter a desired application name and click "***Save***".
3. **Save the workspace of the project**
   a. Select "***File*** > ***Save Workspace***". Enter a desired workspace name.
4. **Add the IAR Project Connection**
   a. Select "***Project*** > ***Add Project Connection***" and click "***OK***".
   b. On the "***Select IAR Project Connection File***" dialog, select the "*.ipcf* file" and click "***Open***".
5. **Repeat steps 2 and 4 for other core projects.**



[3] Please refer to the "How to export an application to IAR Embedded Workbench (Single-core)" It shows how to create a new project and how to save; configure the project.

How to export an application to IAR Embedded Workbench

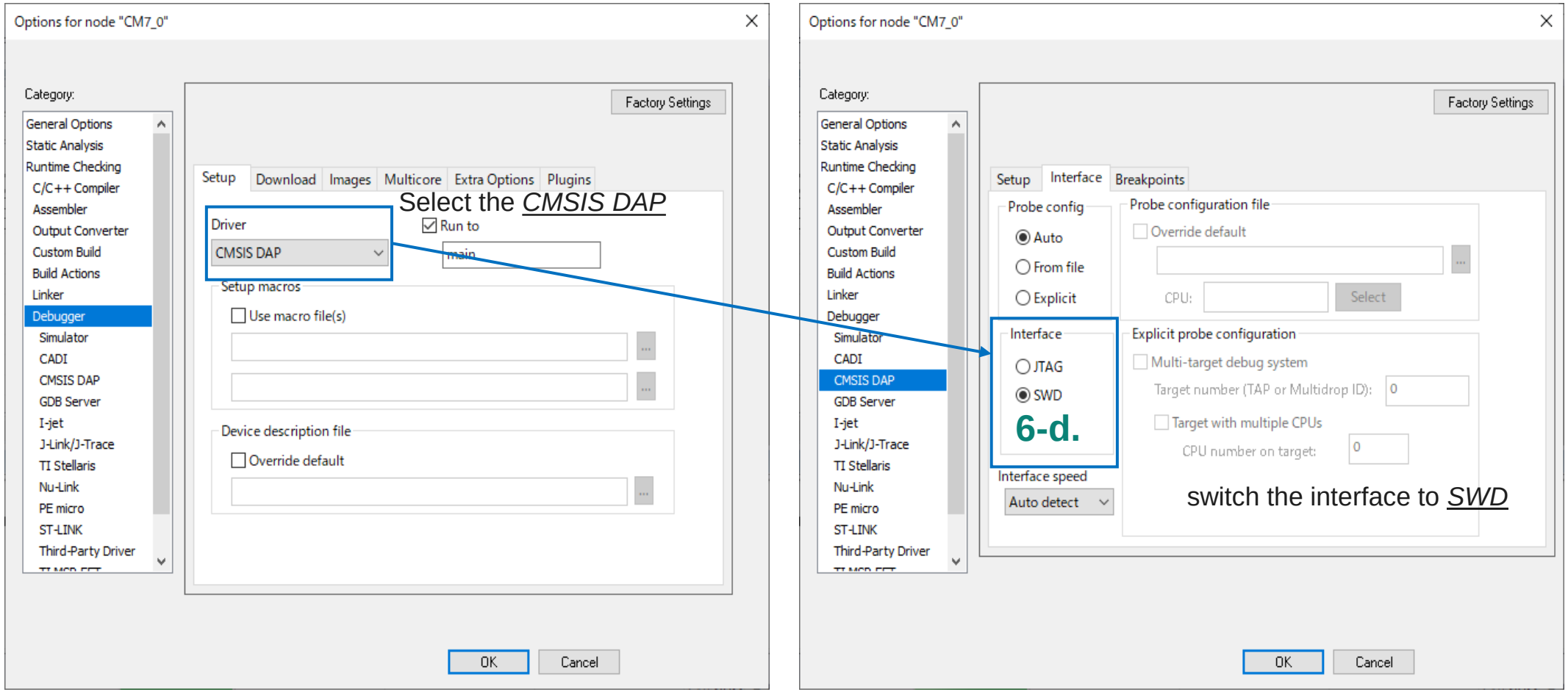
6. **Project configuration for CM7_0/CM7_1.**
   a. Select the CM7_0 core project and go to "***Project***[4] > ***Options*** > ***General Options***"
   b. Select device: "***Infineon*** > ***Traveo-II*** > ***Infineon CYT4BF_M7***". Note that in some versions, "*Traveo-II*" shows as *"TRAVEO™ T2G".*
   c. Add "*_CORE_CM7_0_=1 "*in the "*Configuration file symbol definitions*".



[4] Please refer to "How to export an application to IAR Embedded Workbench (Single-core)" It shows how to open the project options.

How to export an application to IAR Embedded Workbench (contd.)

d. On the dialog. Select the "*Debugger*" category in the Setup tab, and then select the driver as "*CMSIS DAP*".

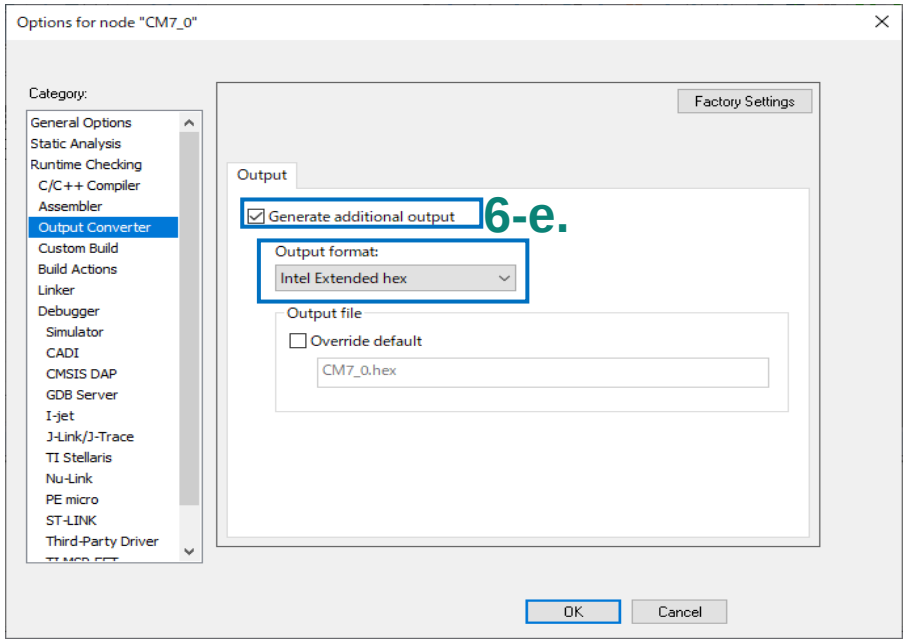


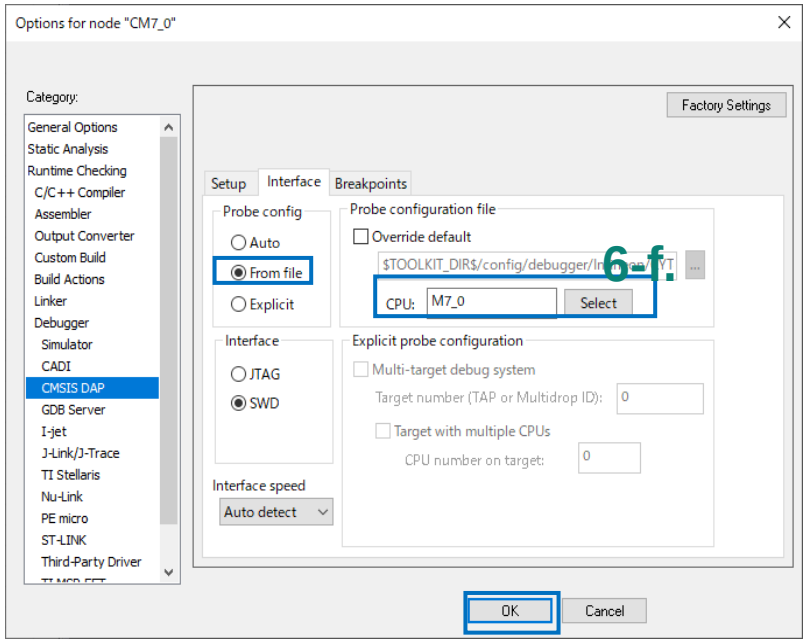
How to export an application to IAR Embedded Workbench (contd.)

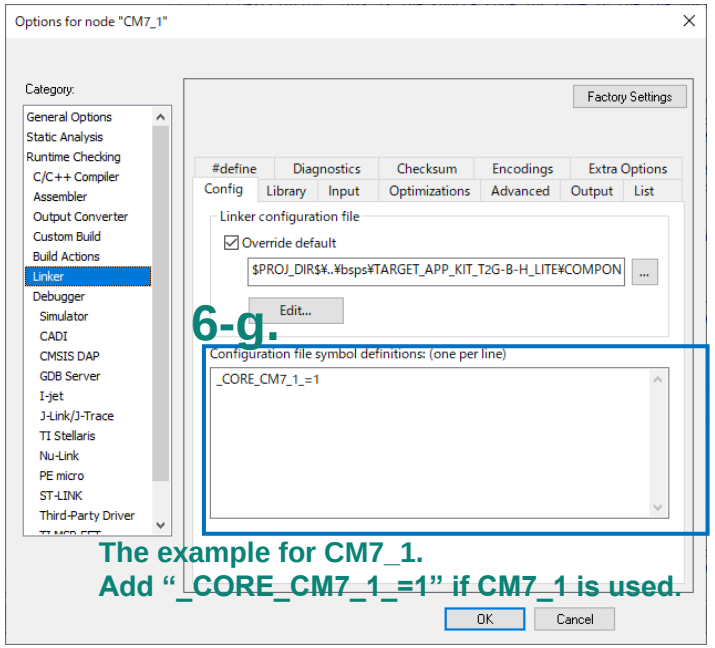
e. Enable hex file generation. On the dialog, select the "*Output Converter*" category. Check the box *"Generate additional output"* and set *"Output format"* as "*Intel Extended hex*".

f. Select the probe in the "***Debugger*** > ***CMSIS_DAP***" category and switch to the "*interface*" tab. Select the "*From file*" radio button, click "*Select*" next to the "*CPU*" label, and choose "*M7_0*". Then click "***OK***".

g. Repeat these steps for CM7_1 if it is used, but for "step d", change to "*_CORE_CM7_1_= 1".*





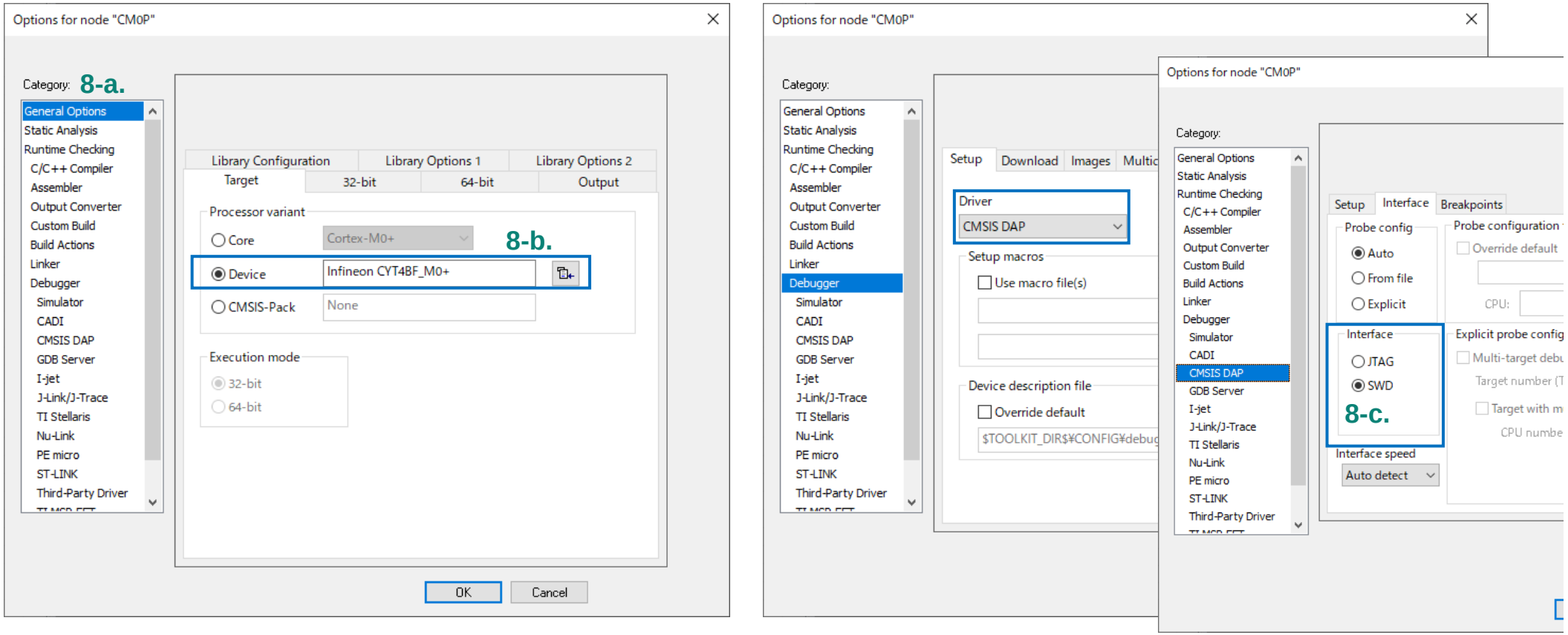


7. Click the "Make[5]  button for CM7_0/CM7_1, and make sure there is no build error.

[5] Please refer to the "How to export an application to IAR Embedded Workbench (Single-core)" It shows how to build the project.

How to export an application to IAR Embedded Workbench

8. **Project configuration for CM0+.**
   a. Select the project and go to "***Project*** > ***Options*** > ***General Options***".
   b. Select device: "***Infineon*** > ***Traveo-II*** > ***Infineon CYT4BF_M0+***"
   c. On the dialog box, select the" *Debugger*" category, and then select the applicable "*Driver*" as "*CMSIS-DAP*". Switch the interface from "*JTAG*" to "*SWD*".



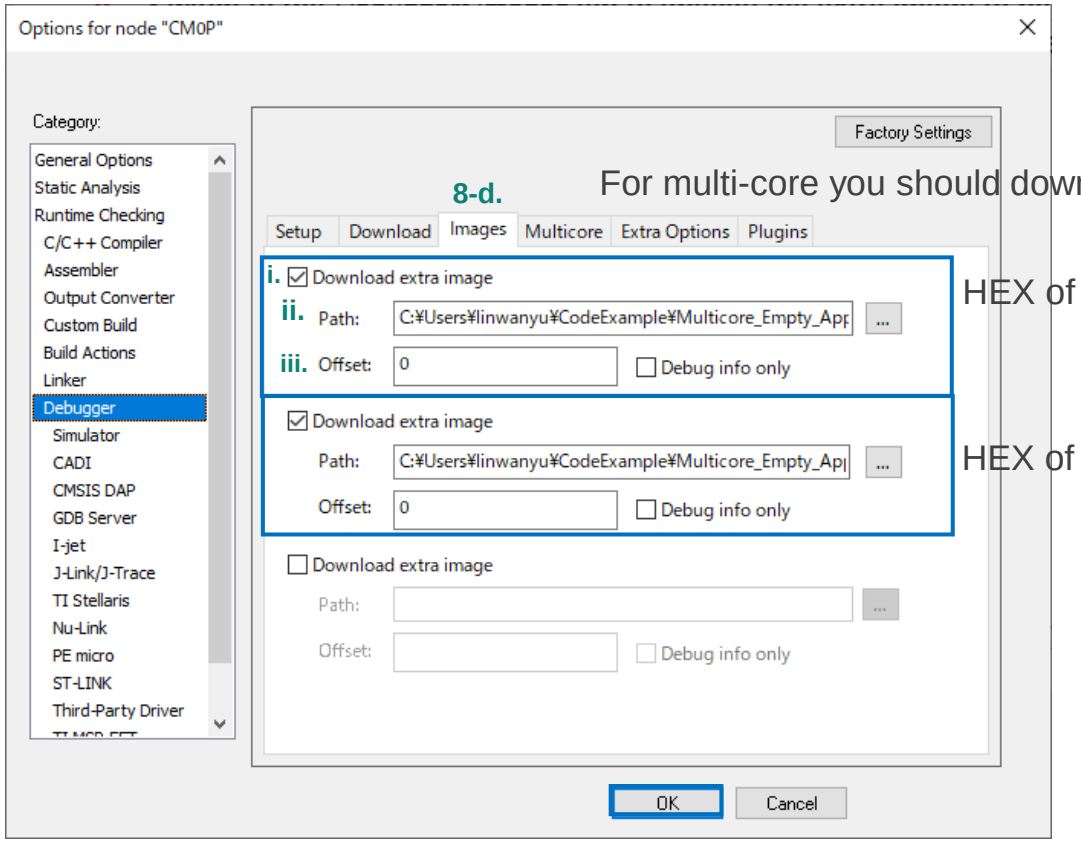
How to export an application to IAR Embedded Workbench(contd.)

d. Switch to the "***Debugger*** > ***Images***" tab to specify the extra image to be downloaded prior to debugging in order to download images of all projects in one process.

i. Select the "*Download extra image*" check box

ii. Provide a "*Path*" to the CM7_0/CM7_1 *HEX*[6] image

iii. Enter 0 for "*Offset*". Then click "***OK***".



For multi-core you should download two extra images.

HEX of CM7_0

HEX of CM7_1

[6] HEX file which is automatically generated by the build function. The file is located in the application directory "\proj_cm7_1\Debug\Exe", each core has a HEX file after executed build.

How to export an application to IAR Embedded Workbench(contd.)

e. Add a prebuild command to build all projects prior to programming/debugging.

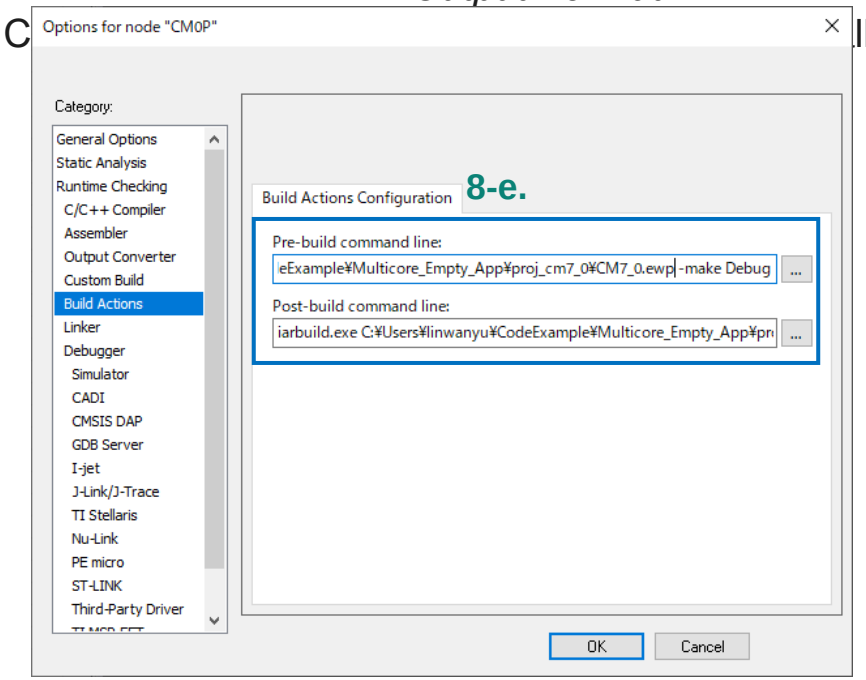
i. In the "*build Actions*" category set, "*Pre-build command line*" to:

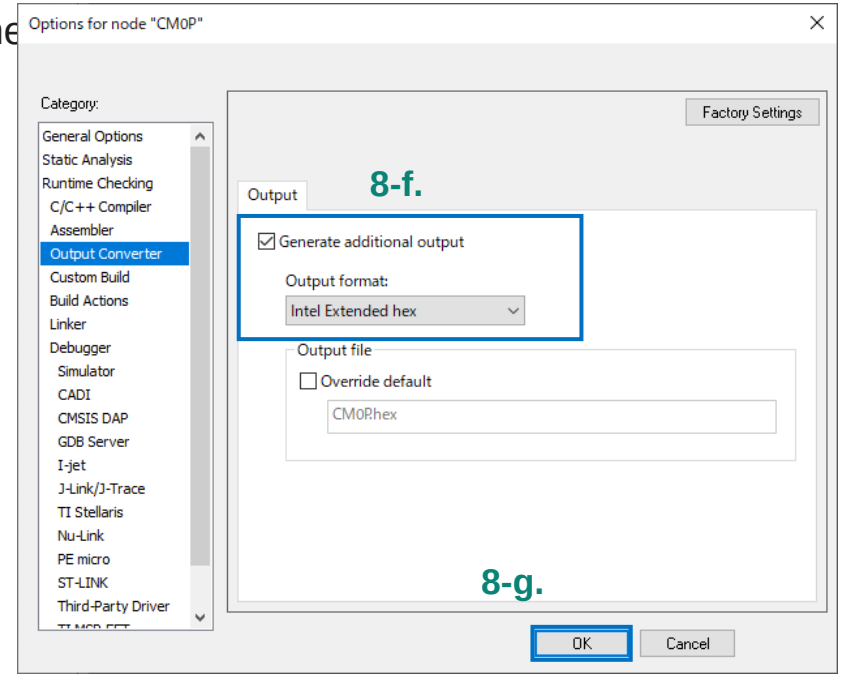
iarbuild.exe <cm4/cm7_proj_loc> ewp -make Debug

(Example: iarbuild.exe C:\Users\...\...\Multicore_Empty_App\proj_cm7_0\CM7_0.ewp -make Debug)
(Example: iarbuild.exe C:\Users\...\...\Multicore_Empty_App\proj_cm7_1\CM7_1.ewp -make Debug)

f. Enable hex file generation. In the "***Runtime Checking*** > ***Output Converter***" category, select the "*Generate additional output*" check box and ensure "*Output Format*" is set to "*Intel Extended hex*".

g. C ll the





9. Click the "Make " button for CM0+, and make sure there is no build error.

How to export an application to IAR Embedded Workbench

10. Create a session configuration file and configure multi-core debugging for the CM0+ project.

a. Create an XML file containing a project list that should be launched in a multi-core debug session.

b. Go to "***Project*** > ***Options*** > ***Debugger***", switch to the "*Multicore*" tab.

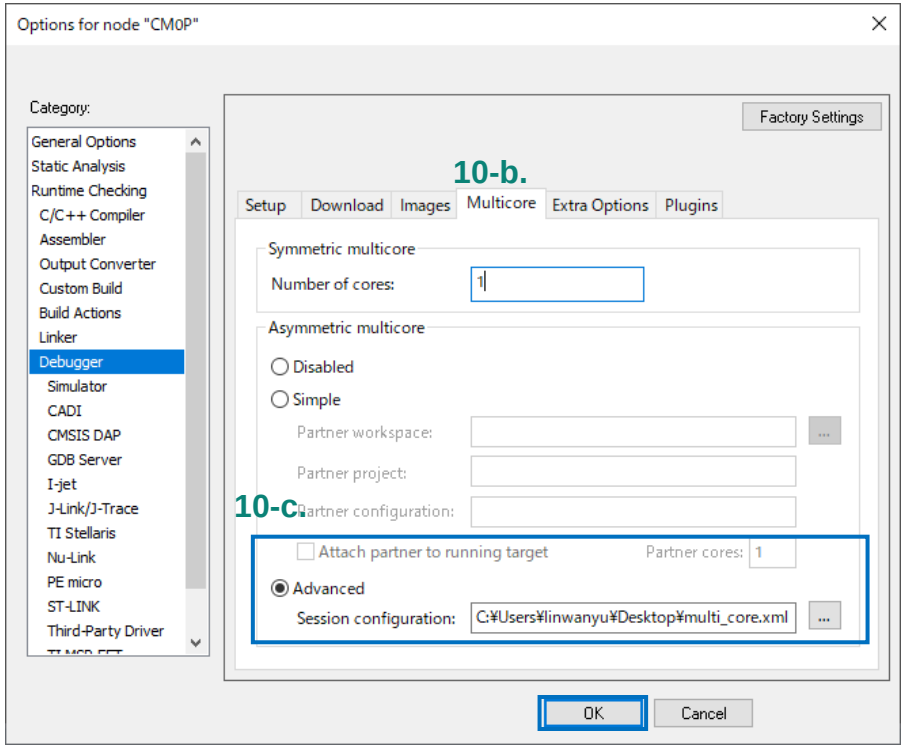
c. Select the "*Advanced*" radio button and specify a path to the session configuration file in the "*Session configuration*" field. Then, click "***OK***".

This is an XML file, please copy it in your PC and update the yellow lines (line 4/line 13/line 21).
Enter your address which is the .eww file address of the application.
Example: *C:\...\...\...\Multicore_Empty_App_IAR\Multicore_Empty_App.eww*

The following shows an example of a triple-core device.
For a dual-core device, remove the third partner node.
Please copy it and replace the address of the .eww file.

10-a.

```
<?xml version="1.0" encoding="utf-8"?>
<sessionSetup>
<partner>
<name>cm0</name>
<workspace>C:\...\...\...\Multicore_Empty_App_IAR\Multicore_Empty_App.eww</workspace>
<project>cm0</project>
<config>Debug</config>
<numberOfCores>1</numberOfCores>
<attachToRunningTarget>false</attachToRunningTarget>
</partner>
<partner>
<name>cm7_0</name>
<workspace>C:\...\...\...\Multicore_Empty_App_IAR\Multicore_Empty_App.eww</workspace>
<project>cm7_0</project>
<config>Debug</config>
<numberOfCores>1</numberOfCores>
<attachToRunningTarget>true</attachToRunningTarget>
</partner>
<partner>
<name>cm7_1</name>
<workspace>C:\...\...\...\Multicore_Empty_App_IAR\Multicore_Empty_App.eww</workspace>
<project>cm7_1</project>
<config>Debug</config>
<numberOfCores>1</numberOfCores>
<attachToRunningTarget>true</attachToRunningTarget>
</partner>
</sessionSetup>
```

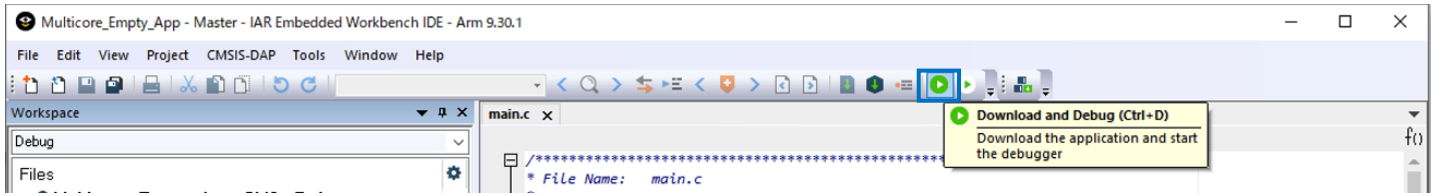


Specify the path in the *Session configuration* field.
(Example: *C:Usersmtw-multi-coreMulticore_Appmulti- core_workspace.xml*)

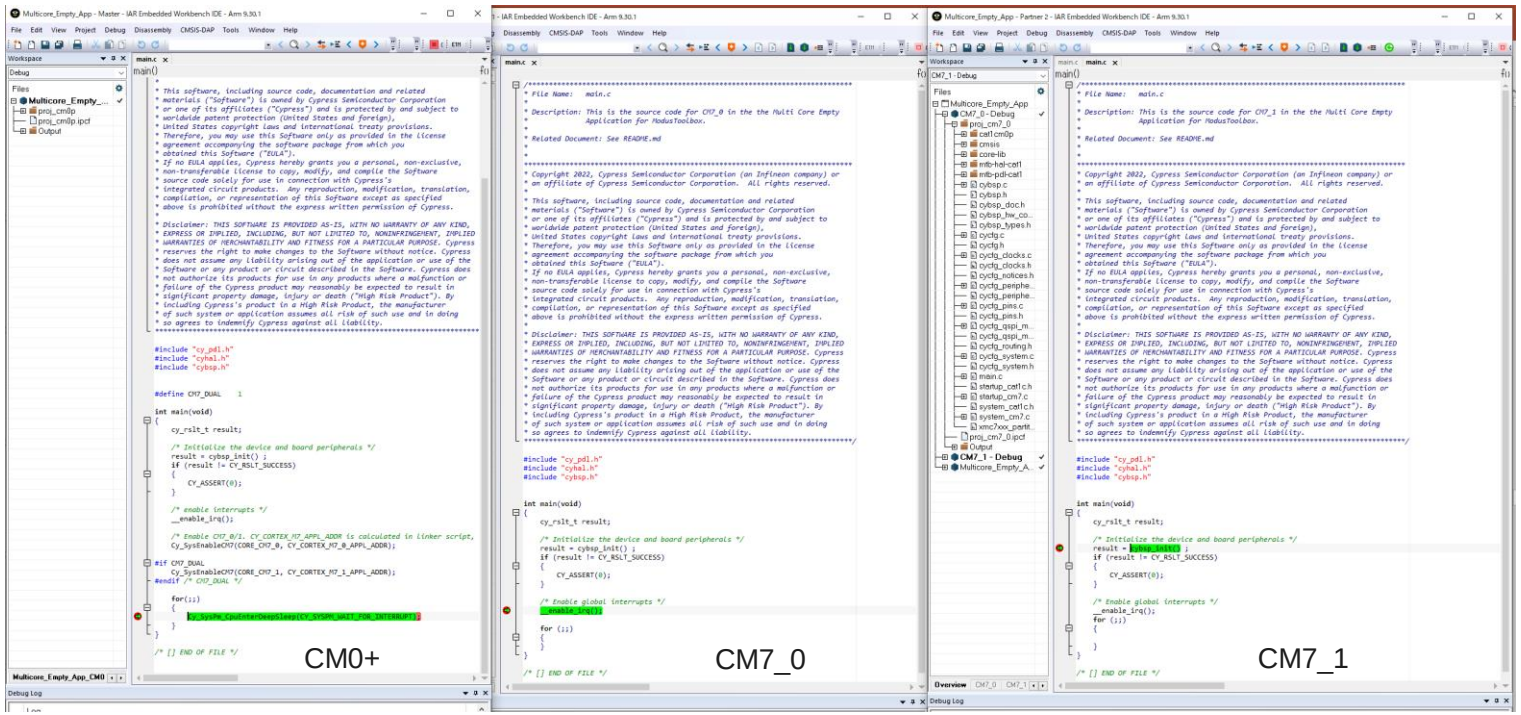
How to export an application to IAR Embedded Workbench

11. **Connect the kit to the host PC. Select CM0+ and build it again; then click "*Download and Debug*" . IAR will open three windows: CM0+, CM7_0, and CM7_1 (the window depends on how many cores you want to debug).**

Switch it to CM0+ Core project



12. **Click "*Go* " to run the program.**



References

User guides

- Eclipse IDE for ModusToolbox™ user guide
- ModusToolbox™ Device Configurator user guide
- ModusToolbox™ tools package user guide

Revision history

Revision	ECN	Submission Date	Description of Change
**	7909676	2023/05/29	Initial release

Important notice and warnings

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-05
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Go to:
www.infineon.com/support

Document reference
002-37557 Rev. **

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.


Infineon