DAVE™ Software Development Kit overview

The DAVE™ SDK project provides the tool used to develop DAVE™ 4 software components or applications called "DAVE™ APPs". It provides facilities to create user interface configurations, edit template files, create a signal designer view and create documentation for the APPs. In this help contents are described the procedures/steps to develop DAVE™ 4 applications.

The audience for this help contents are:

- Experienced professionals with DAVE™ 4
- Professionals who want to develop DAVE™ 4 applications using the available application libraries

Prerequisites:

- Working knowledge of DAVE™ 4 Code Engine (DAVE™ CE) and its associated features
- Usage of XMC microcontrollers and knowledge of the hardware capabilities (available in the reference manual)
- Basic programming skills
- Basic object oriented programming skills (OOPS)
- Basic JAVA/Groovy programming skills

> Note: You do not have OOPS/JAVA/Groovy programming skills?
> → Follow the procedure as recommended in this basic tutorial.

Preparing the workbench

This tutorial will get you started using the DAVE™ Software Development Kit (SDK) in the Workbench.

First, you have to verify the workbench is properly configured for APP development. It is assumed that:

- Your Workbench has its default settings. To reset the current perspective to its original layout, from menu bar select **Window > Reset Perspective... > Yes**
- You are familiar with the "DAVE™ APP" concept and its applications

Creating a simple APP

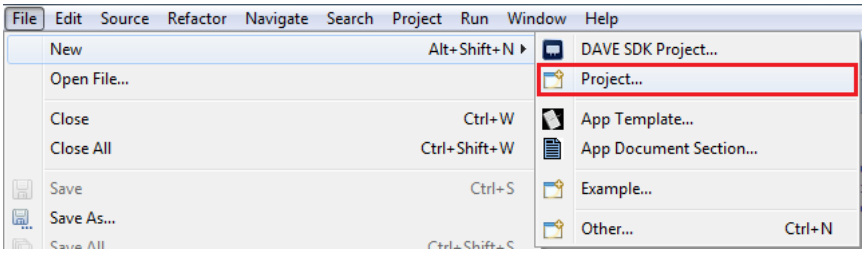
In this tutorial, you will use DAVE™ SDK to create a simple "Hello World" application (APP). This tutorial describes the process of creating a new APP project where the build is automatically managed by DAVE™ SDK, and running the APP in DAVE™ 4.

To create a simple "Hello World" APP using DAVE™ SDK, perform the following steps:

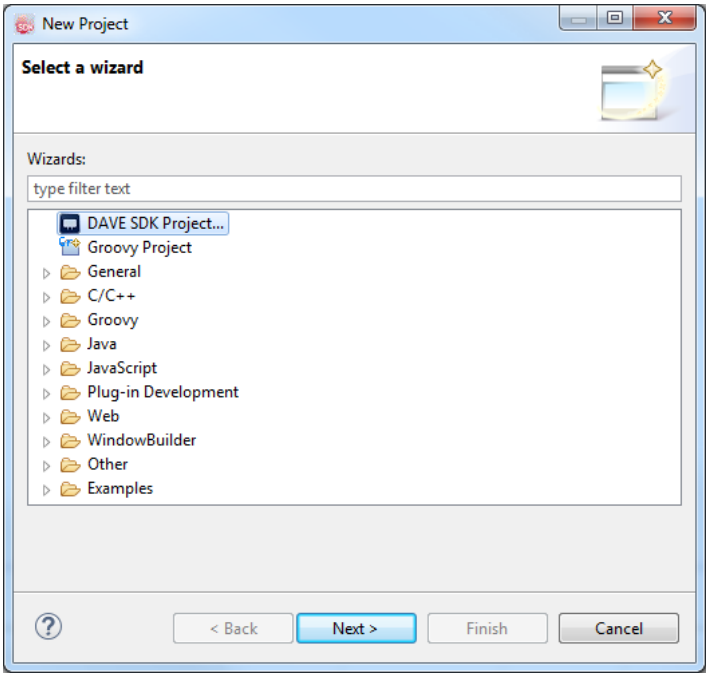
1. Create a new APP project
2. UI Design
3. Write a template for code generation
4. Update the APP manifest
5. Create an Update Site
6. Import the APP into DAVE™
7. Template Debugging
8. Explore by yourself!

Step 1: Create a new APP project

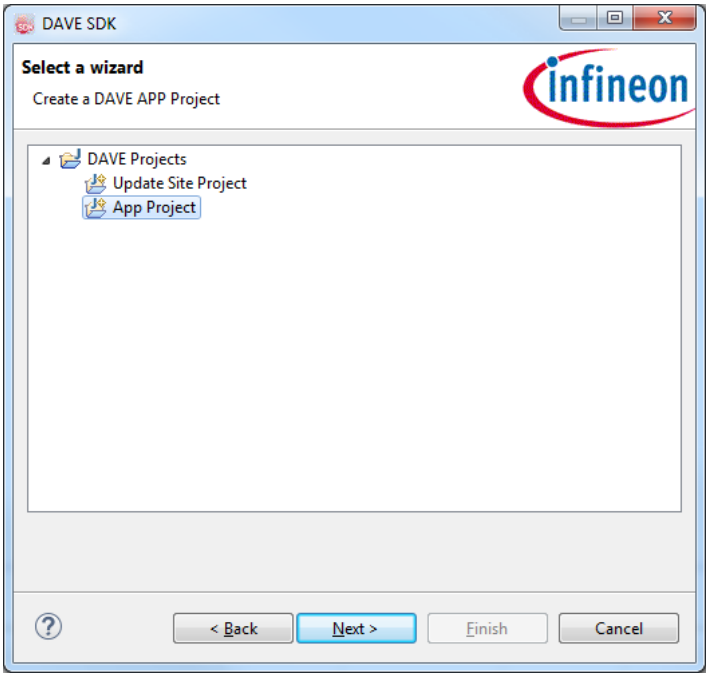
1. Select **File > New > Project...**



2. Select the type of project to create. For this tutorial, select **DAVE SDK Project...** and click **Next**.

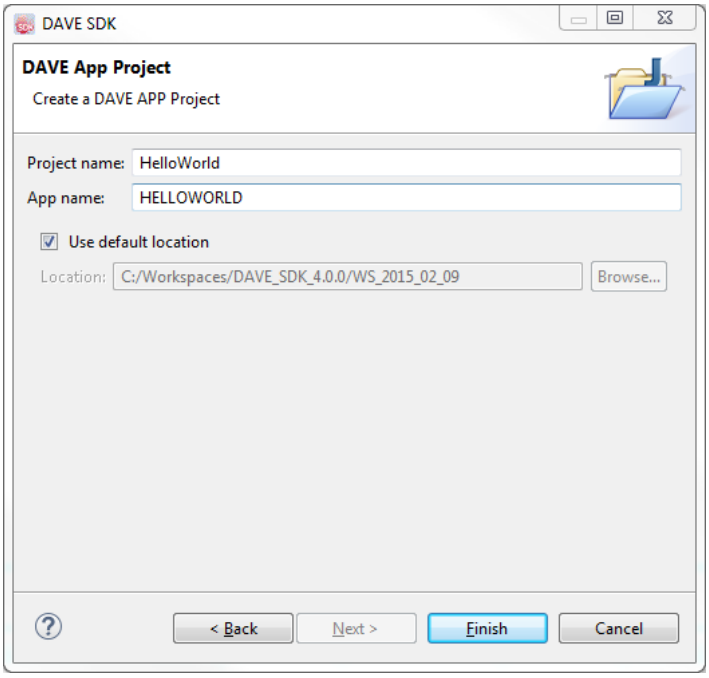


3. The **DAVE SDK Project** wizard opens.



Expand **DAVE Projects** list and select **APP Project**. After that click **Next**.

4. The **DAVE APP Project** view appears.

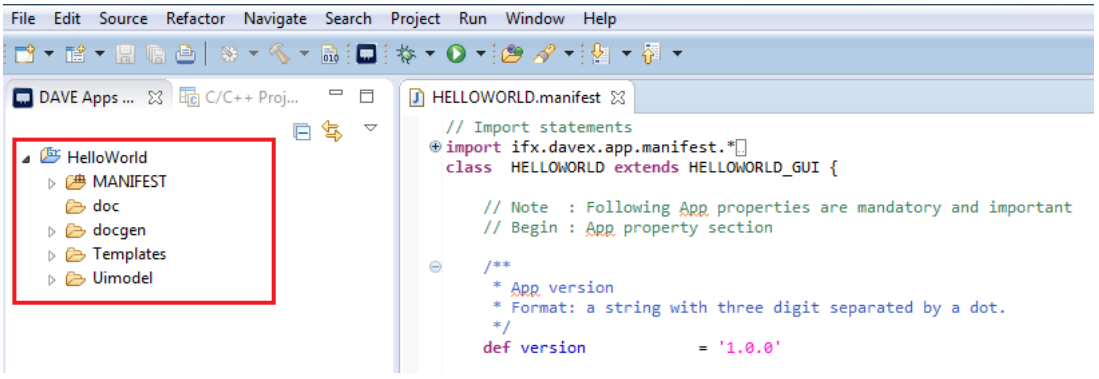


a. In the **Project name** field, type a name for the project, such as **HelloWorld**.
b. In the **APP name** field, type a name for the APP, such as **HELLOWORLD**.

   APP name must be always written in UPPERCASE.

c. Click **Finish**.

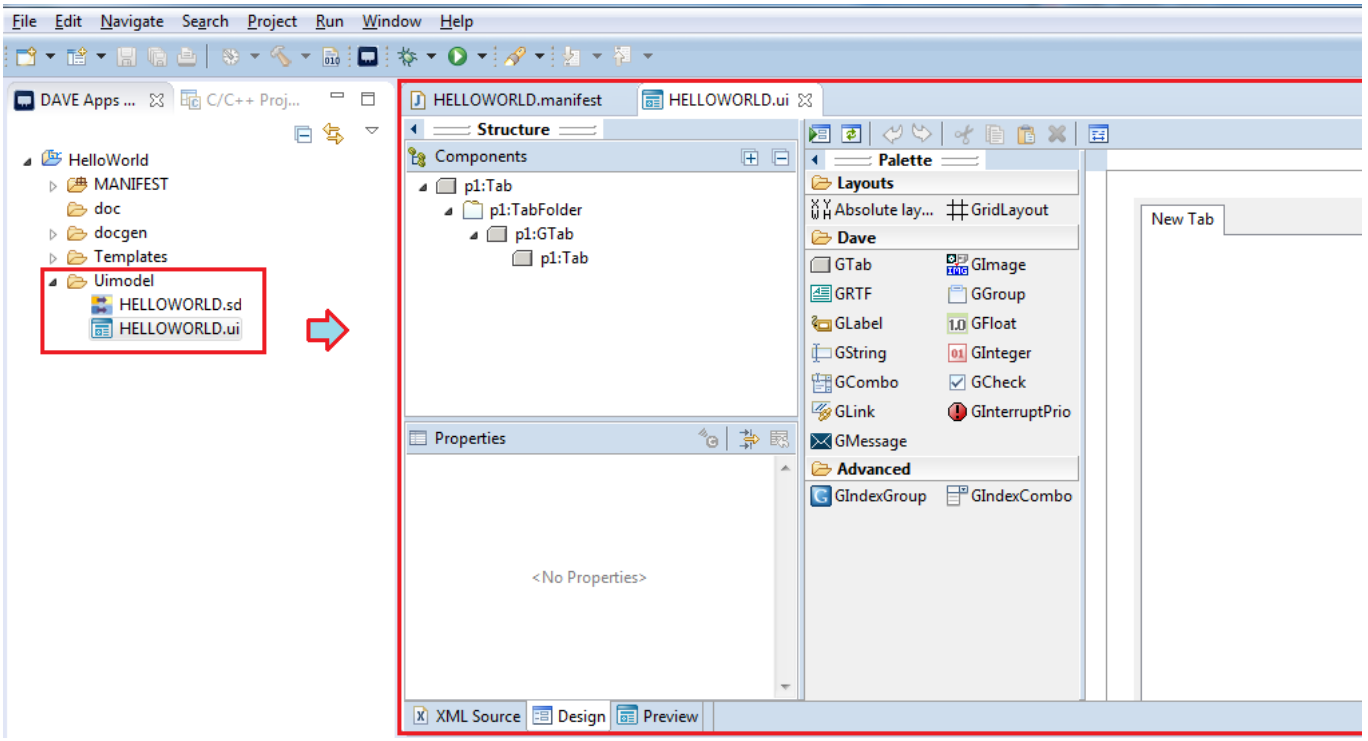
5. In the left-up part of the screen you could see the **APP project folder (HelloWorld)**, expand it and you will see some additional folders.



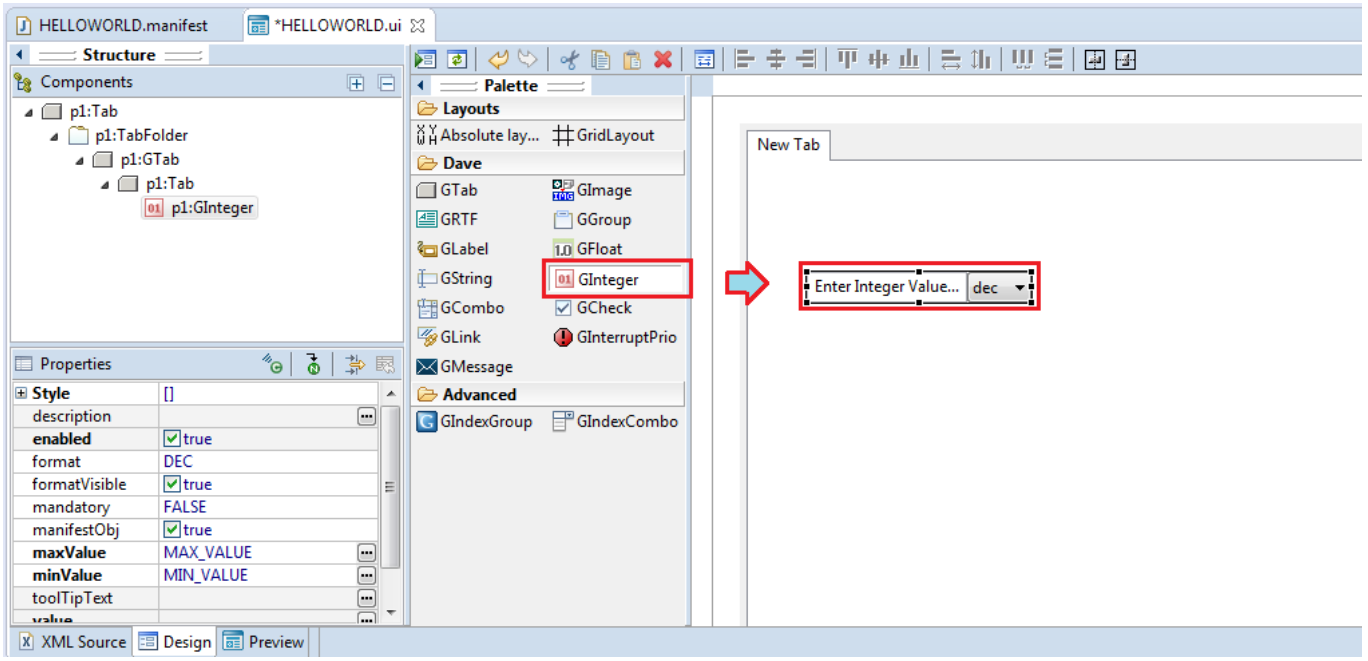
a. **MANIFEST** folder
   - Groovy code used for User Interface logic
b. **doc** folder
   - Location of generated APP documentation
c. **docgen** folder
   - Source files required for APP documentation generation
d. **Templates** folder
   - Code generation templates and files to be copied during code generation
e. **Uimodel** folder
   - User Interface configuration files

Step 2: UI Design

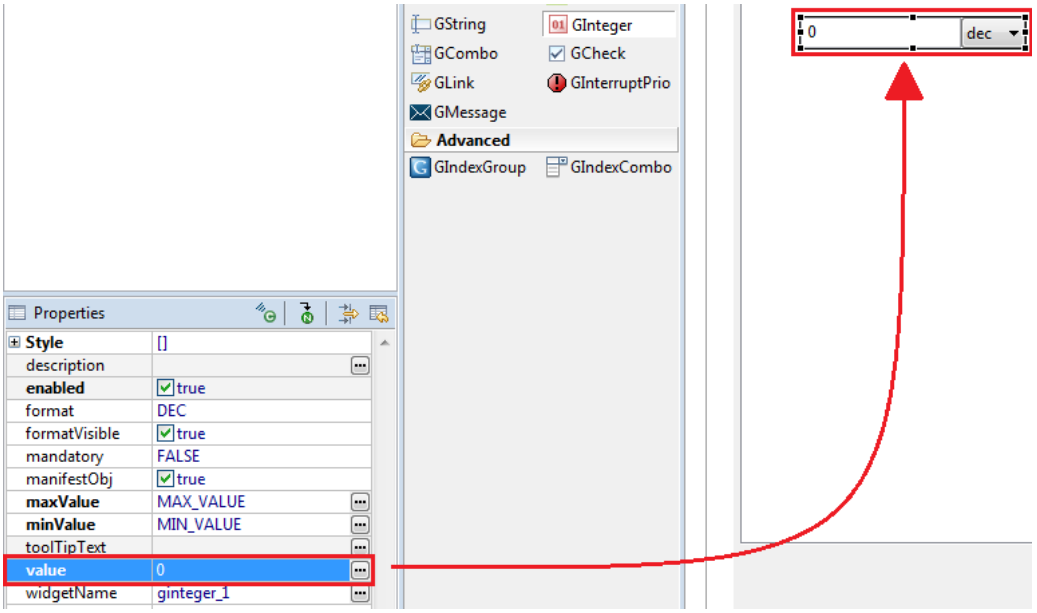
1. At the **APP project explorer (HelloWorld)**, expand the **Uimodel** folder and double click **HELLOWORLD.ui**.
   A new tab called **HELLOWORLD.ui** should appear at the right.



2. Drop a **GInteger** widget into the **APP UI**.



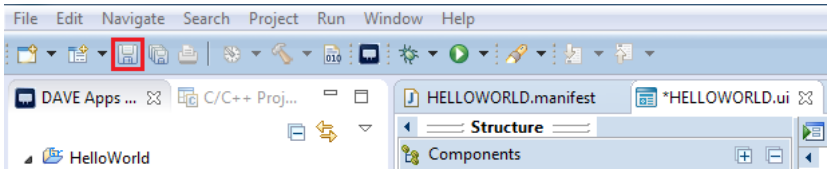
3. Select a **default value** for the **widget**, such as "**0**".



4. Now it is time to **save** the UI and **generate** the UI interface.

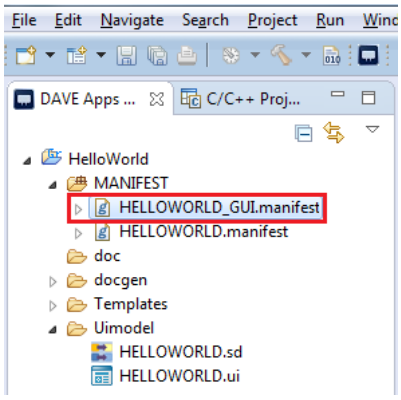
   a. You can save the UI and generate the UI interface in two ways:

      - Pressing the **diskette icon** under the menu bar



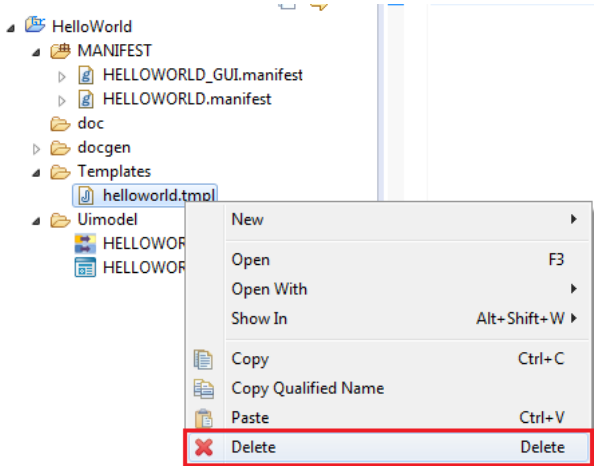
      - Pressing the button combination **CTRL+S**

   b. By saving, DAVE™ SDK automatically generates the **HELLOWORD_GUI.manifest** file.
   This file is the **interface** between UI and the user defined manifest **HELLOWORLD.manifest**.

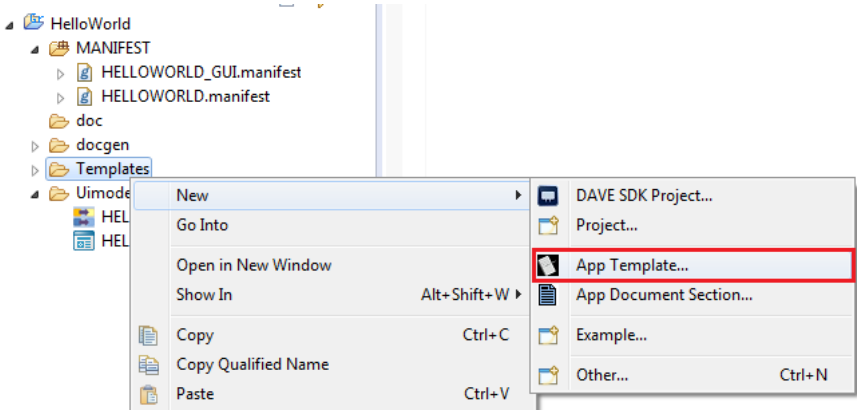


Step 3: Write a template for code generation

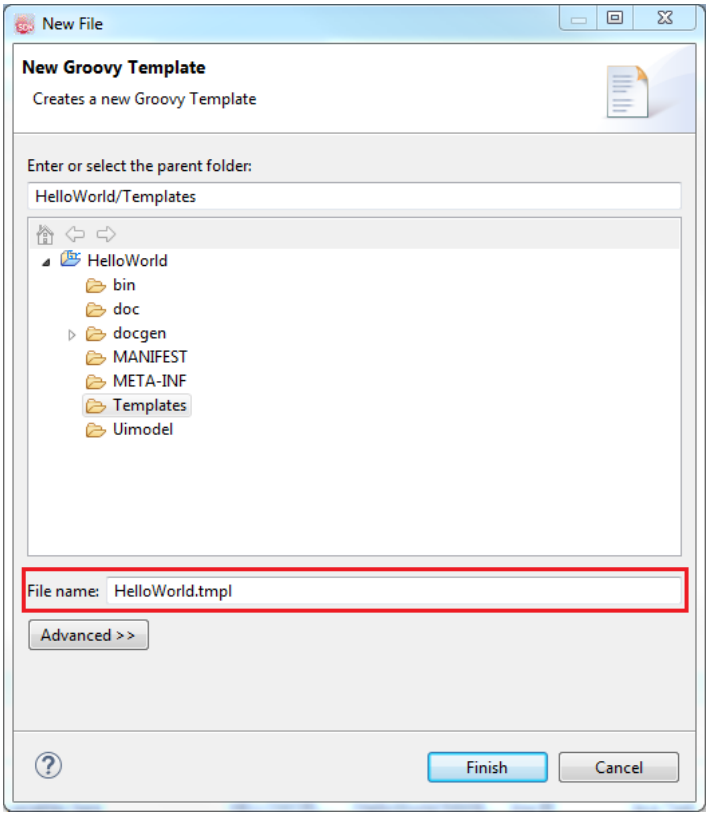
1. By default, DAVE™ SDK will add a file here called like your APP name but in lowercase, in this case **helloworld.tmpl**.
   We are going to remove it and then we are going to create a new one so you can learn how to do it.
   Just expand the **Templates** folder and **right click** on **helloworld.tmpl** > **Delete** > **OK**.



2. We are going to add a new template to **Templates** folder, so **right click** in **Templates** folder > **New** > **APP Template...**

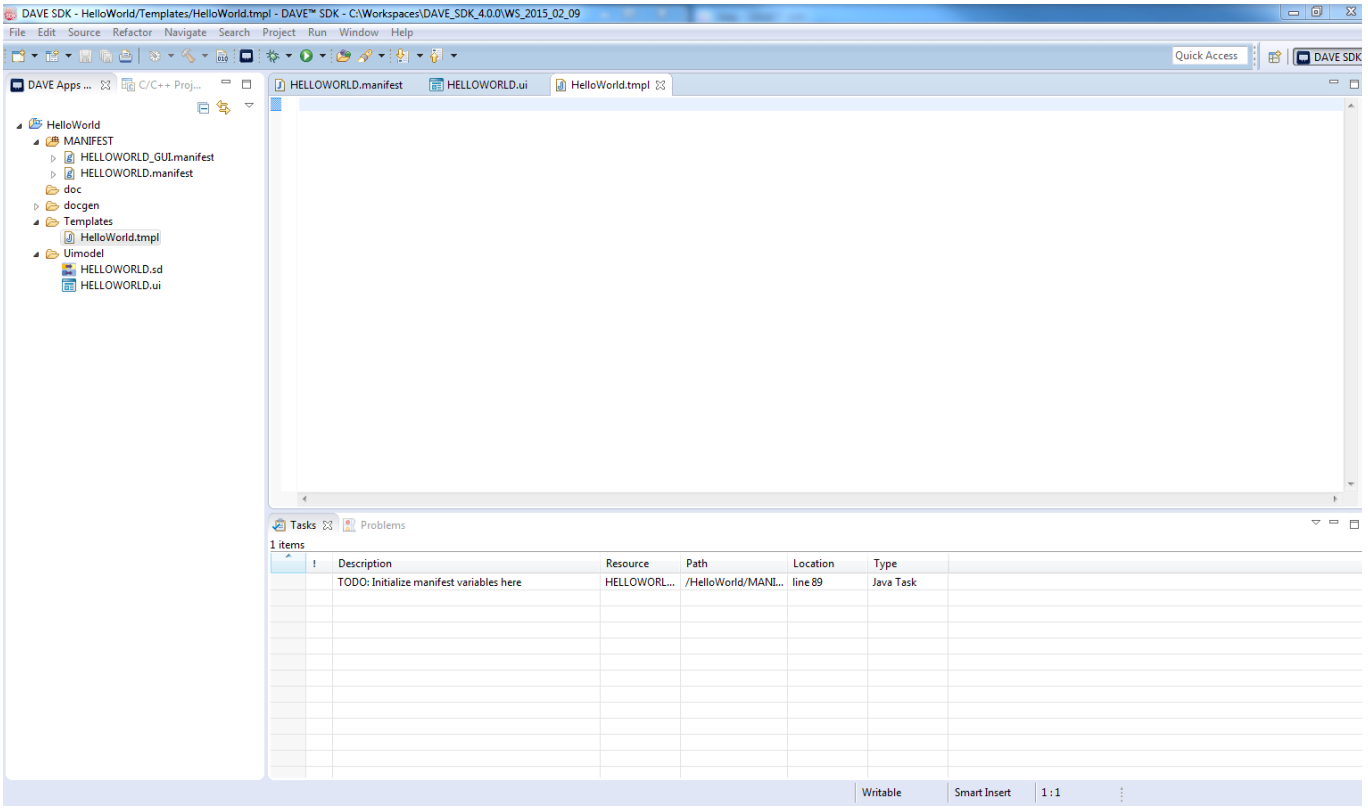


3. This should open the **New Groovy Template** wizard.



In the **File name** field, change the default name for other, such as **HelloWorld.tmpl** and click **Finish**.

4. A new tab called **HelloWorld.tmpl** will open automatically.



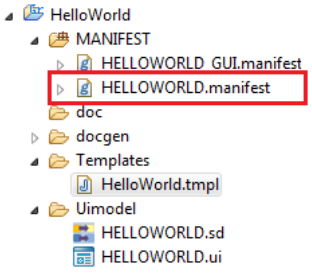
5. Copy the following code into **HelloWorld.tmpl** and **Save** the template.

```
for (HELLOWORLD app in appInstancesList) {
  // Create a macro for each instance of the APP
  out.print("""
      #define ${app.getInstanceLabel()}_MYINTEGER ${app.ginteger_1.value}
  """)
}
```

> GUI values can be read directly from the UI bypassing the APP manifest.

Step 4: Update the APP manifest

1. Open **HELLOWORLD.manifest** file



2. Define the **devices that are compatible with this APP** by modifying the **softwareIDs** APP property as in the code below:

```
def softwareIDs = [
    "XMC4.5.00.F144.ALL":"1.0.0",
    "XMC4.4.00.F100.*":"1.0.0",
    "XMC1.3.01.Q024.*":"1.0.0"
]
```

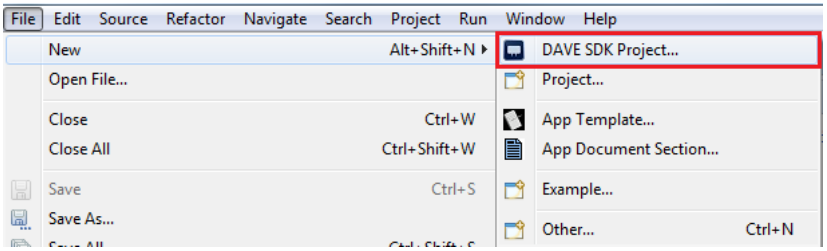
3. At the end of the manifest file, look for the method called **generateFiles()**. This method is executed on code generation event, and you can call some special methods inside **generateFiles()** to execute code generation based on the **templates**.

   Replace the content of the method **generateFiles()** with the one below and **Save**:

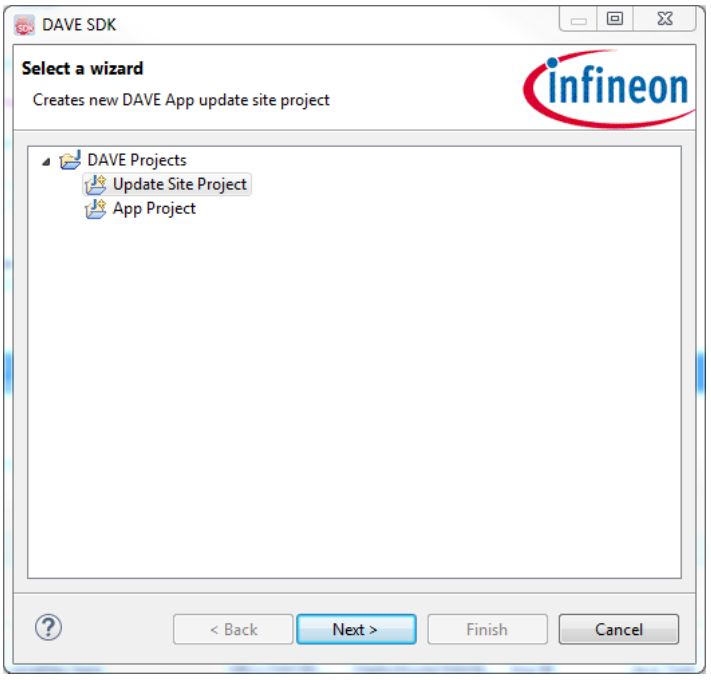
```
def generateFiles(){
    generate("HelloWorld.tmpl", "HELLOWORLD.h")
}
```

Step 5: Create an Update Site

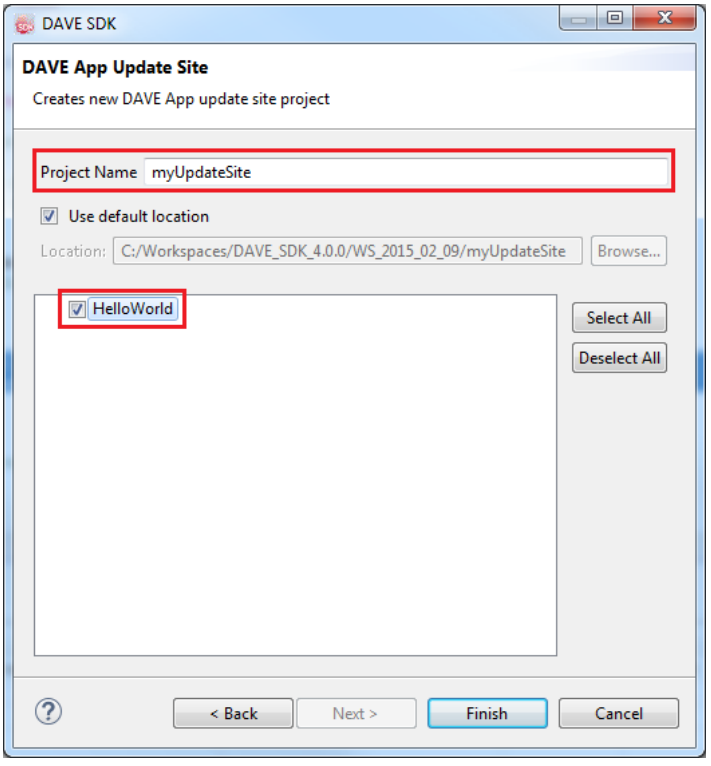
1. **Update Sites** are used to distribute a set of APPs and install them in DAVE™.
   Add a new update site project by **File > New > DAVE SDK Project...**



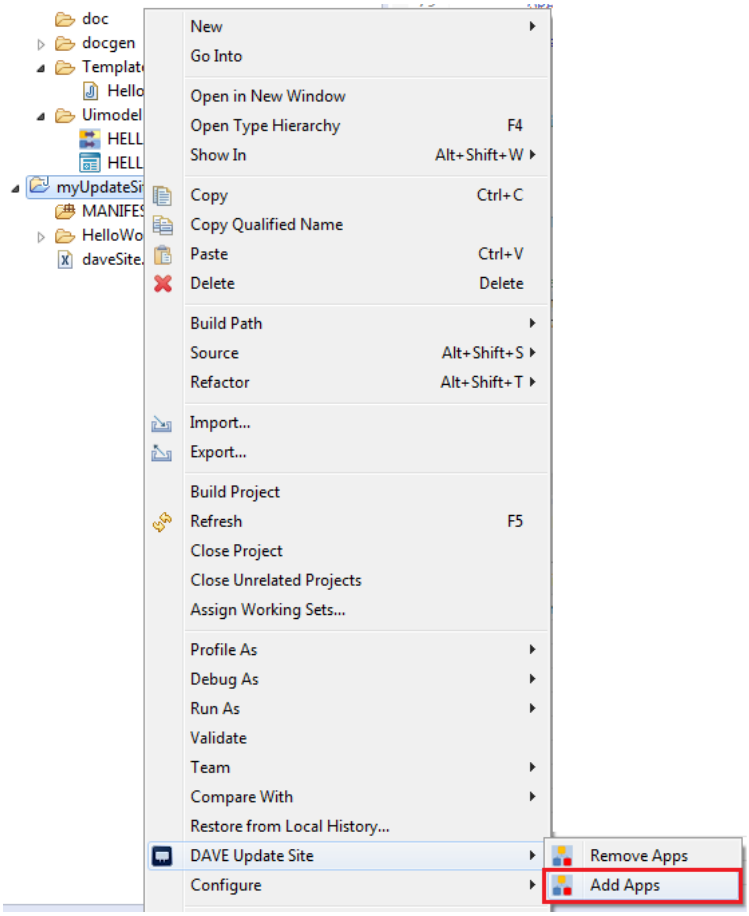
2. Select **Update Site Project** and click **Next**.



3. In the **Project Name** field introduce a name for the project, such as **myUpdateSite**.
   You **must select at least one APP** from the list below, in this case we only have one possible APP, so select **HelloWorld** APP.
   Click **Finish**.

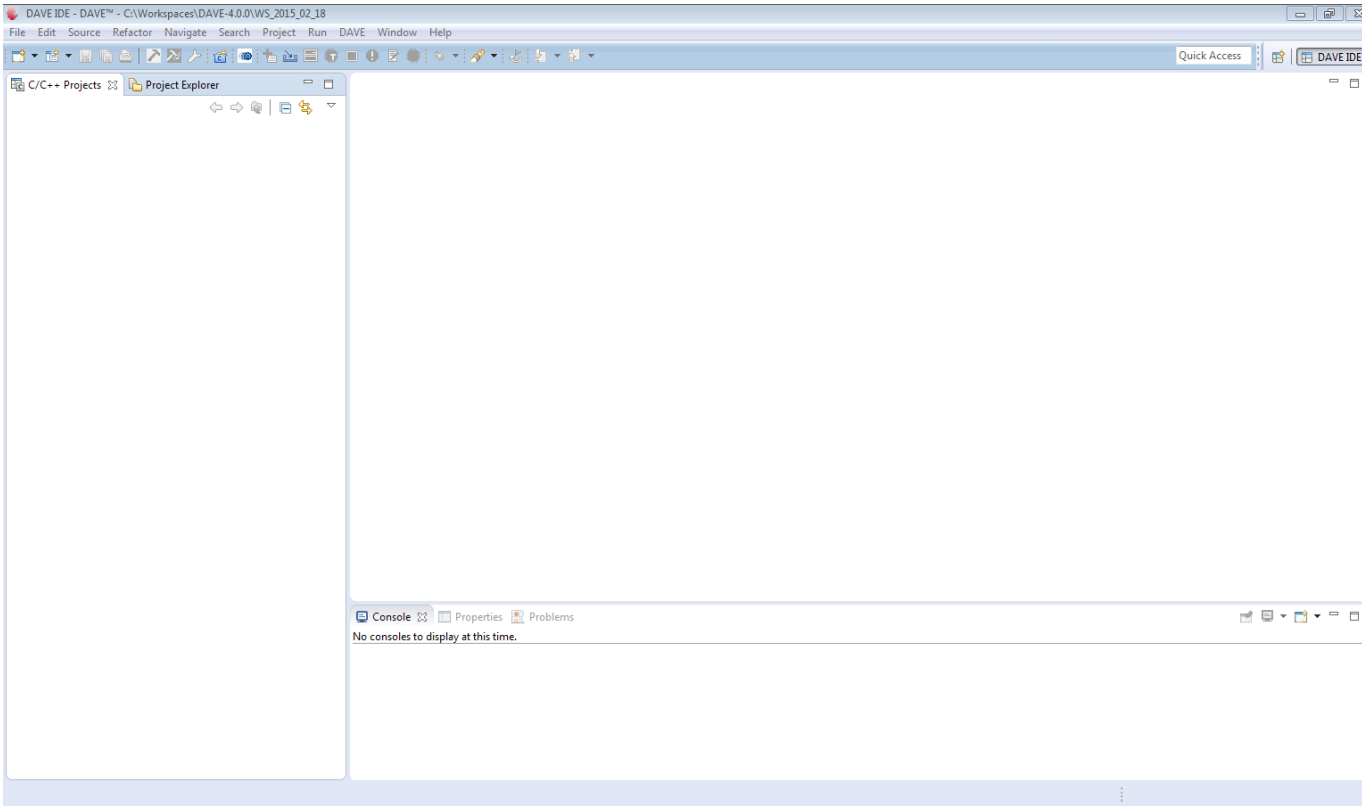


4. If you have more APPs available in the workspace and you want to add them to the Update Site, you can do it later by **right click** in **myUpdateSite** folder **> DAVE Update Site > Add APPs**.

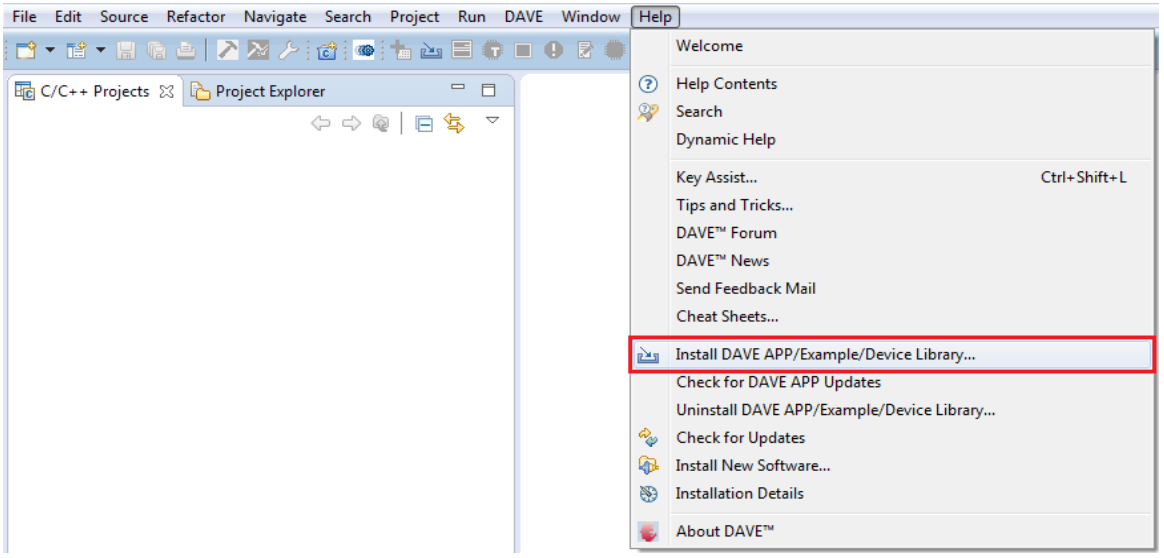


Step 6: Import the APP into DAVE™

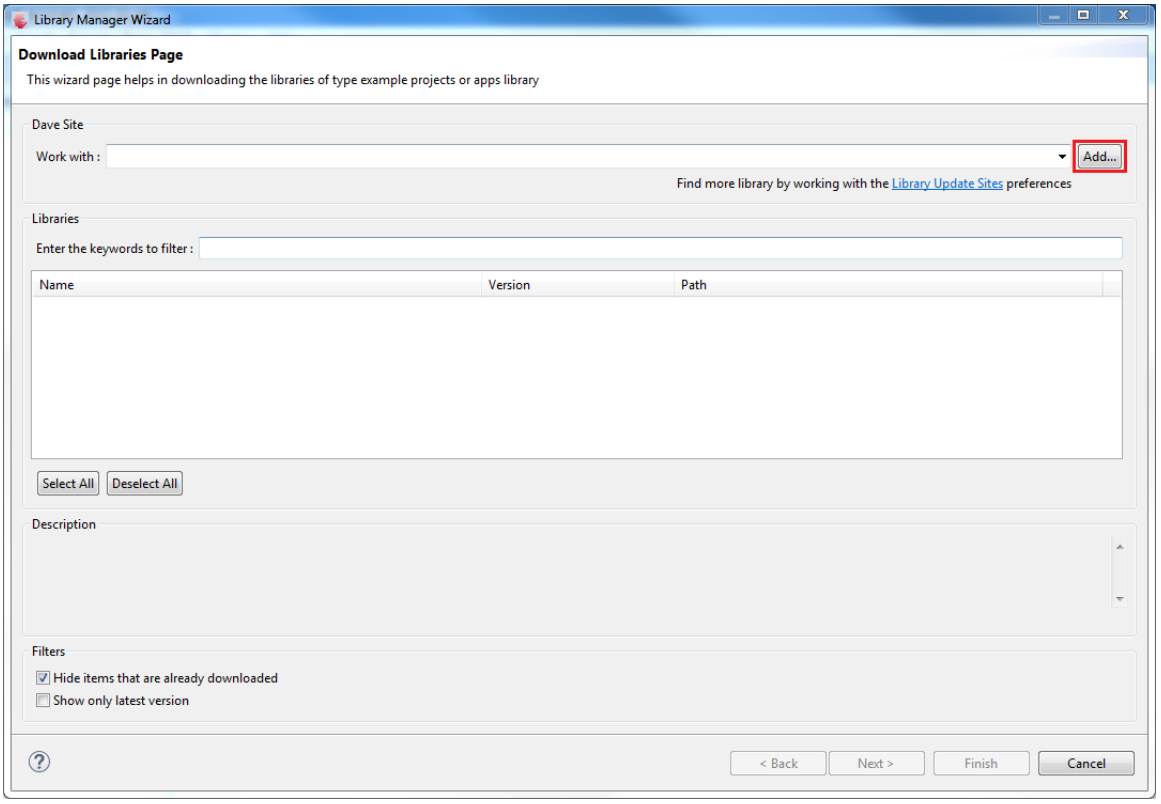
1. Open **DAVE™ 4** and select a workspace.



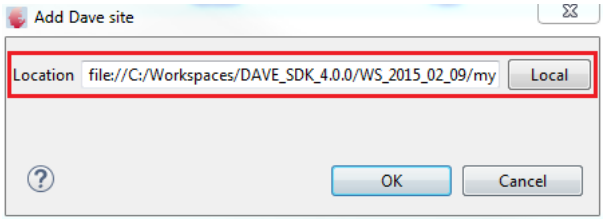
2. **Importing APPs** into **DAVE™** workspace is possible **using the Update Site** folder stored in **DAVE™ SDK** workspace.
   Go to **Help > Install DAVE APP/Example/Device Library...**



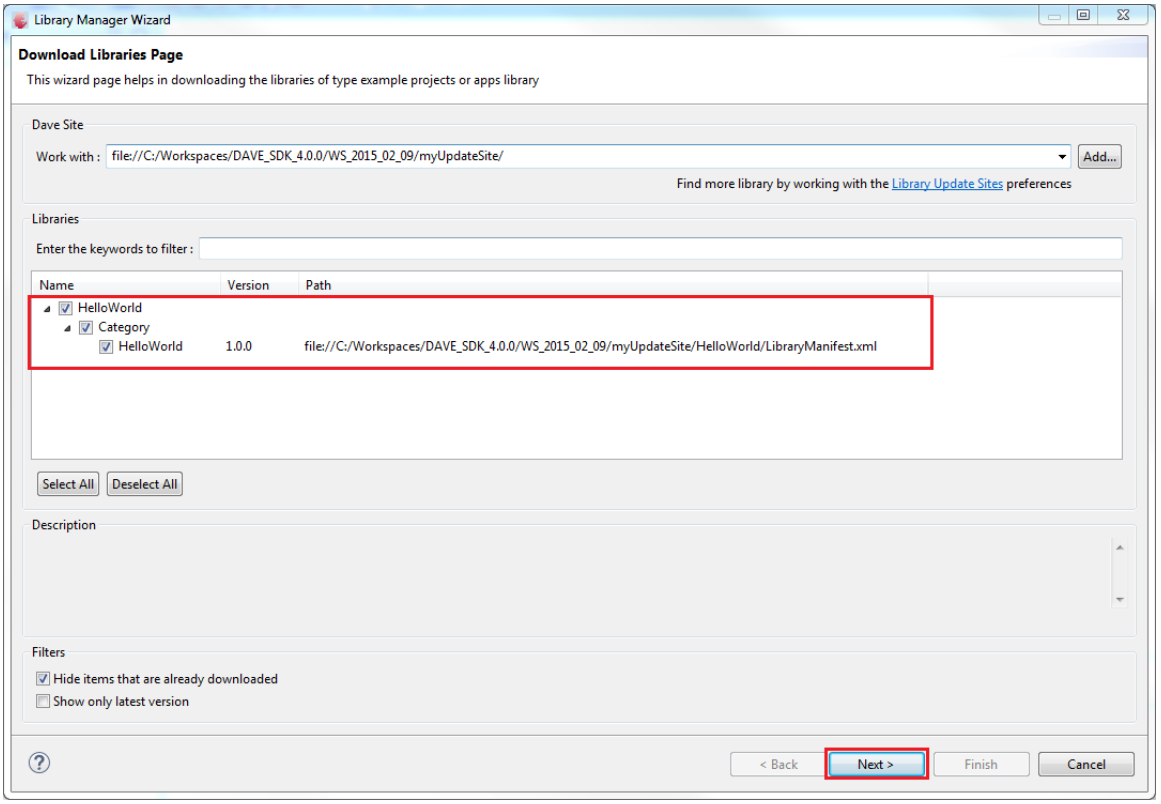
3. This will show up the **Library Manager Wizard** window, and we need to add the **Device Library**, so click **Add...** button.



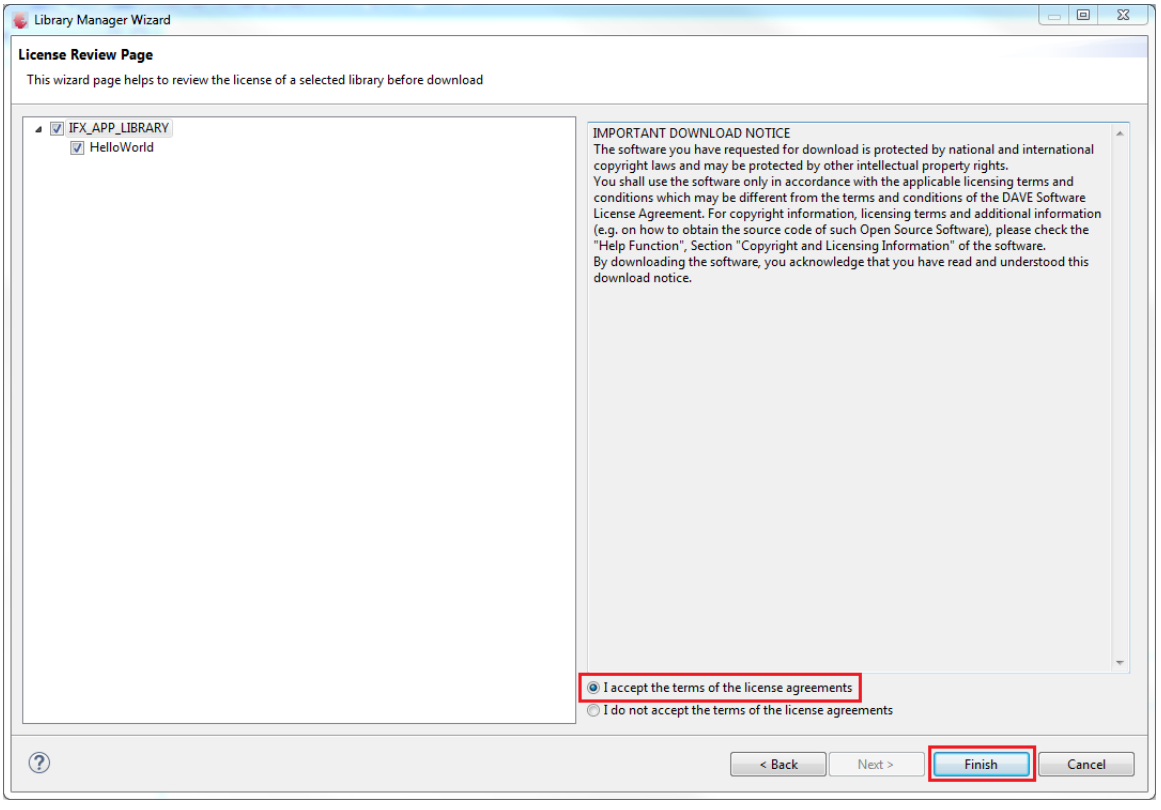
4. Click **Local** and select your **DAVE™ SDK Update Site folder**, in this case **myUpdateSite**. After that click **OK**.



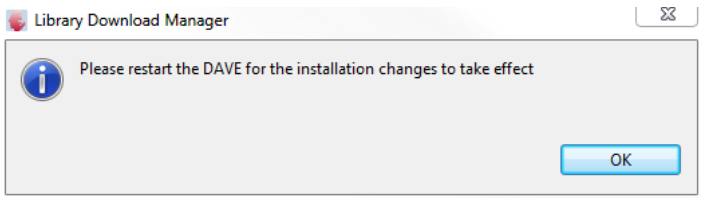
5. Select the **HelloWorld checkbox** and click **Next**.



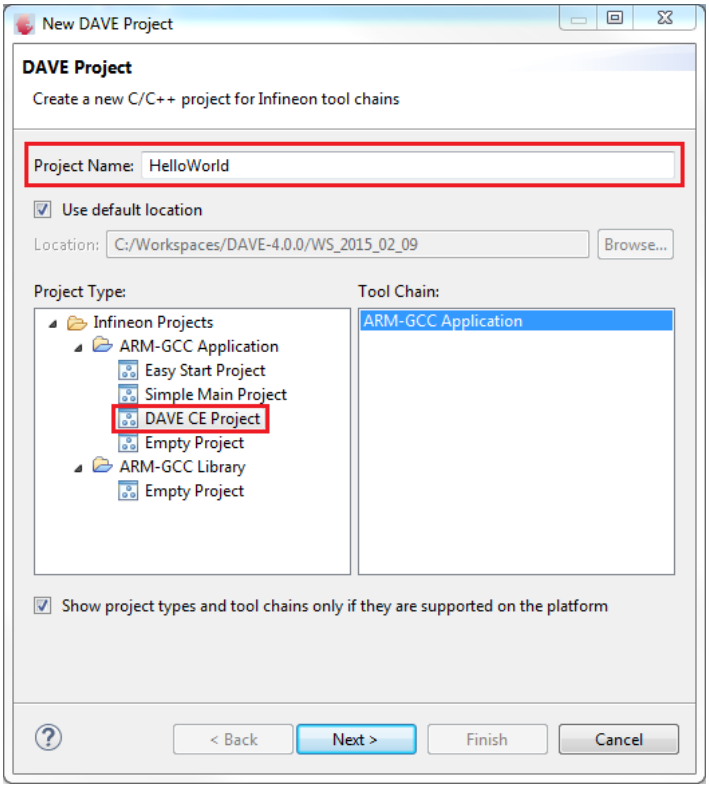
6. **Accept the terms of the license agreements** and click **Finish**.



7. Restart DAVE™ for the installation changes take effect clicking **OK** button.



8. Create a **DAVE™ CE project** going to **File > New > DAVE Project...** and select **Infineon Projects > ARM-GCC Application > DAVE CE Project**.
In the **Project Name** field enter a name for the project, such as **HelloWorld**.
Click **Next.**

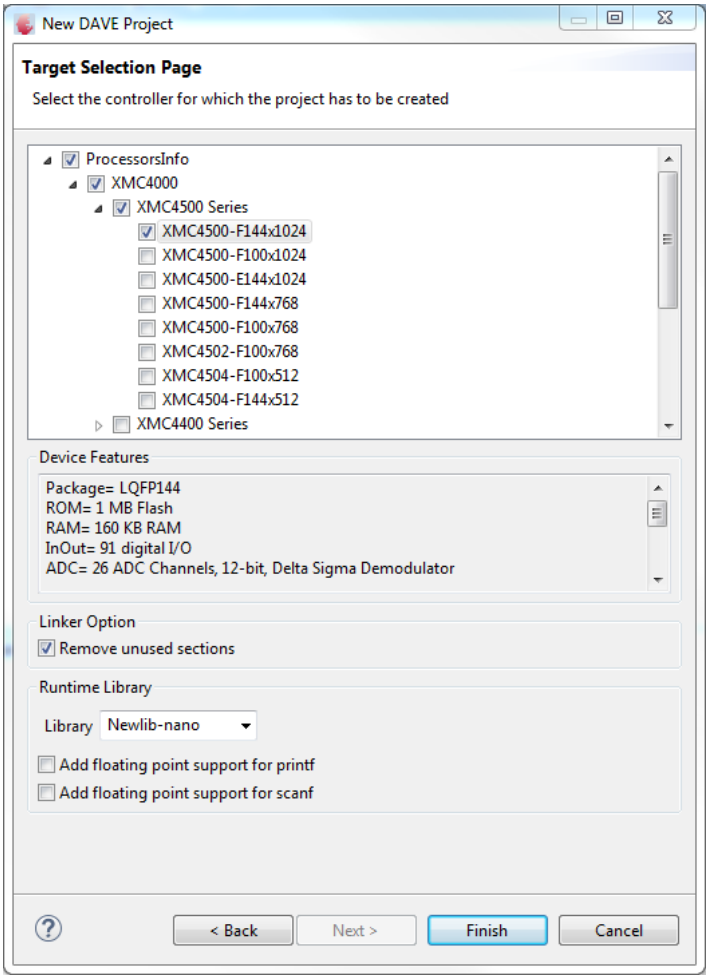


It is supposed you have installed de proper device description libraries in DAVE™ 4, otherwise you can not continue and you must install this libraries before, for more information please refer to DAVE™ 4 help contents.

9. **Select a Target**, such as **XMC4500-F144x1024** and click **Finish**.

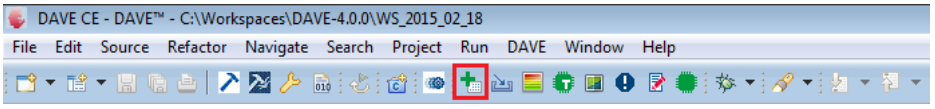
   Remember you have to select a device which you have declared in the **softwareIDs** variable in your APP manifest.

```
def softwareIDs = [
    "XMC4.5.00.F144.ALL":"1.0.0",
    "XMC4.4.00.F100.*":"1.0.0",
    "XMC1.3.01.Q024.*":"1.0.0"
]
```

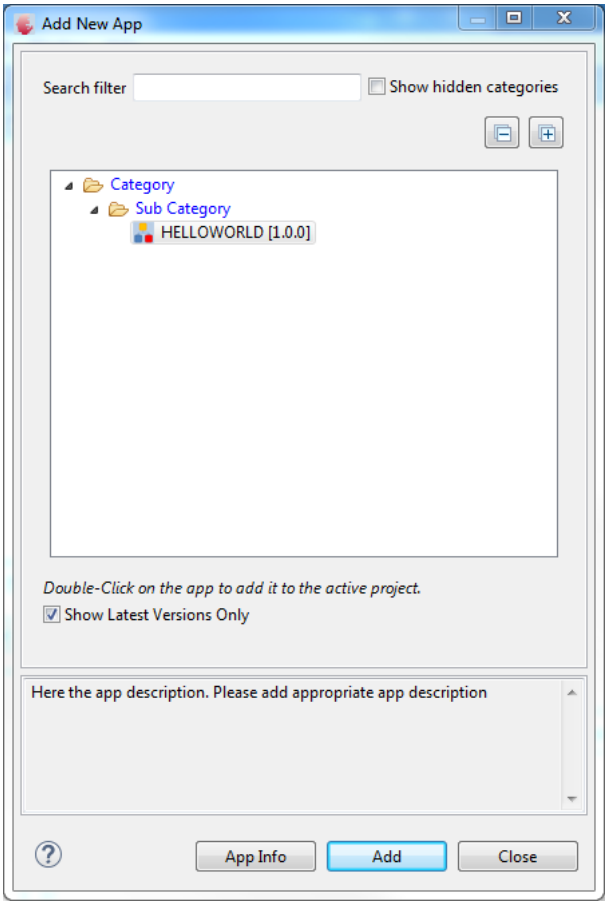


Step 7: Template Debugging

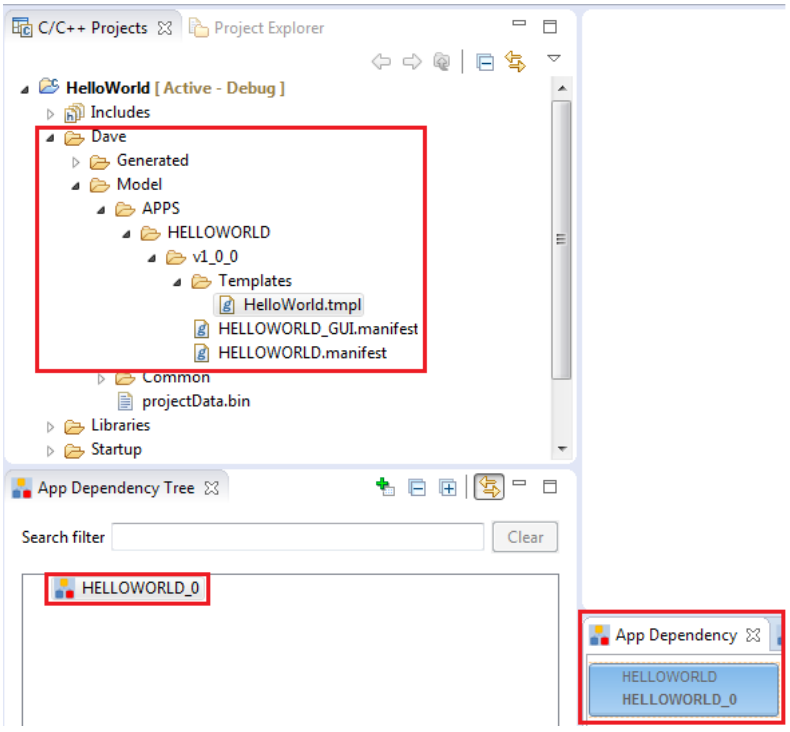
1. Now we are able to debug the template we have created before. First we need to add our **HELLOWORLD** APP, so click **Add New APP** button



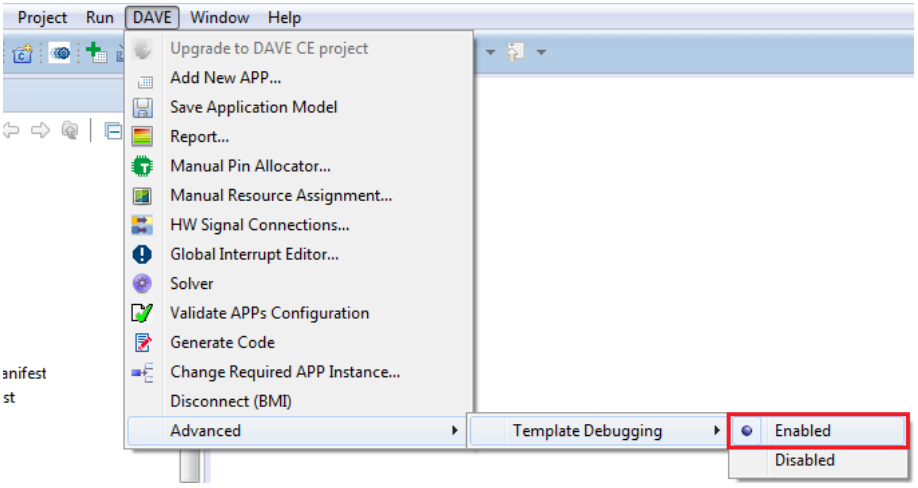
2. Select **HELLOWORLD APP** and click **Add**, then click **Close**.



3. You can observe how our HELLOWORLD APP have been added successfully, and you can expand **Dave > Model > APPS > HELLOWORLD > v1_0_0 > Templates** to find our **HelloWorld.tmpl** template, the one which we are going to debug.



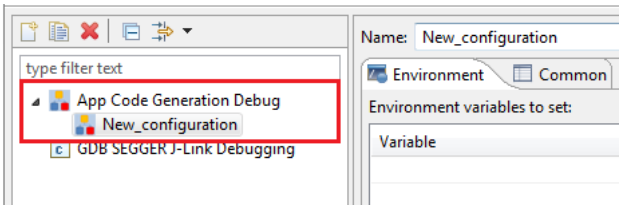
4. **Before debugging**, we need to **enable the Template Debug**.
Go to **DAVE > Advanced > Template Debugging** and click **Enabled**.



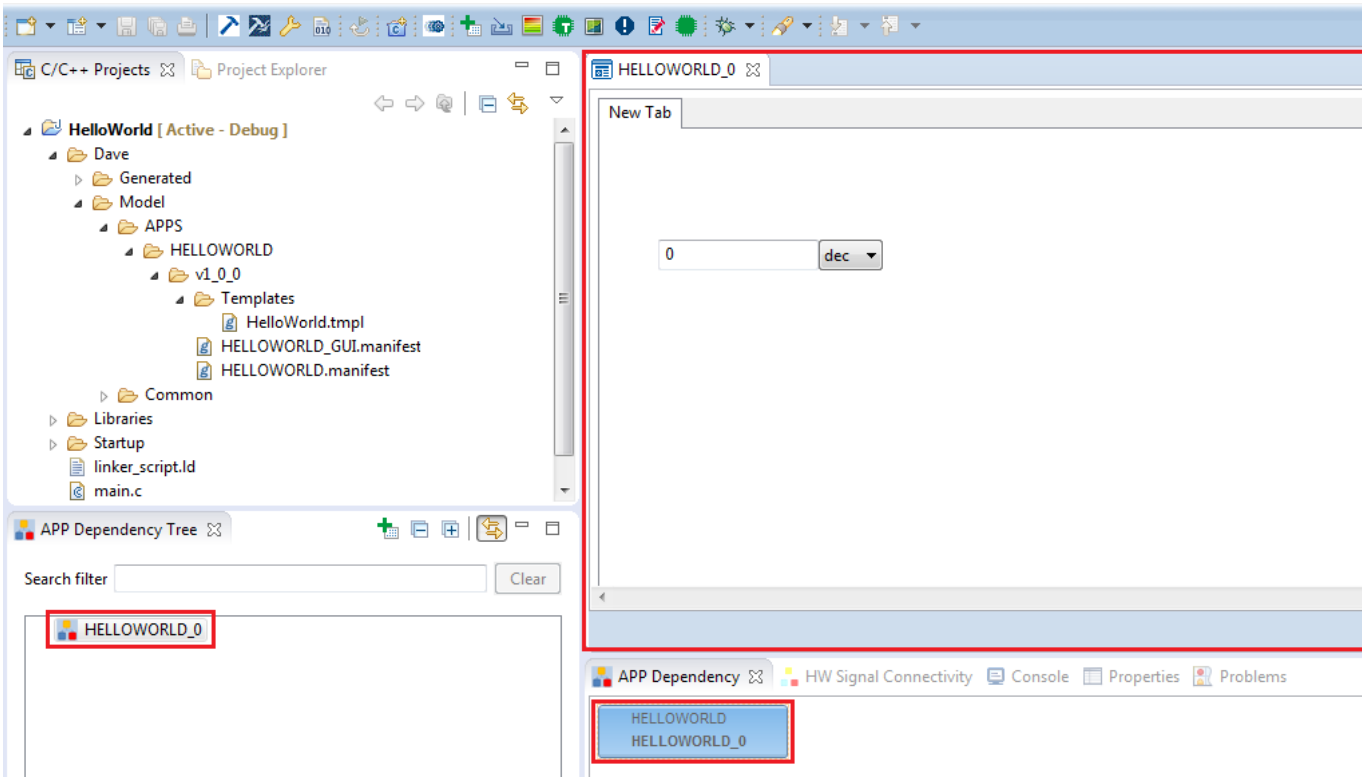
In DAVE™ it is possible to easily debug templates execution. Enabling template debugging will make visible the template debug profile.
As a side effect, some Java/Groovy context menu items are activated, user shall ignore them.

5. We are going to create a **template debug configuration**.
   Right click on **Debug Arrow**, click **Debug Configurations...**

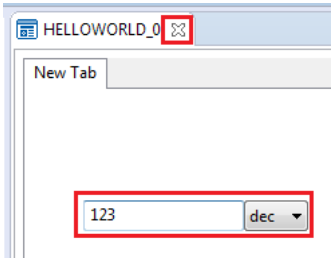
6. **Double click APP Code Generation Debug**, then close the window clicking **Close** button.



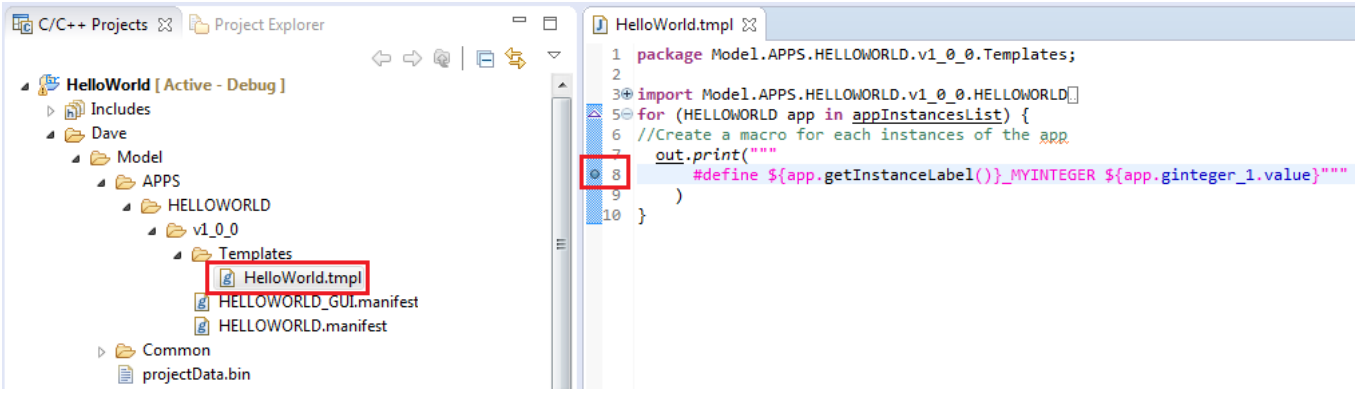
7. Now double click **HELLOWORLD_0** APP in the **APP Dependency Tree**, this will show you the APP UI we had programmed before in DAVE™ SDK.



8. Enter a value such as **123** and close the tab.

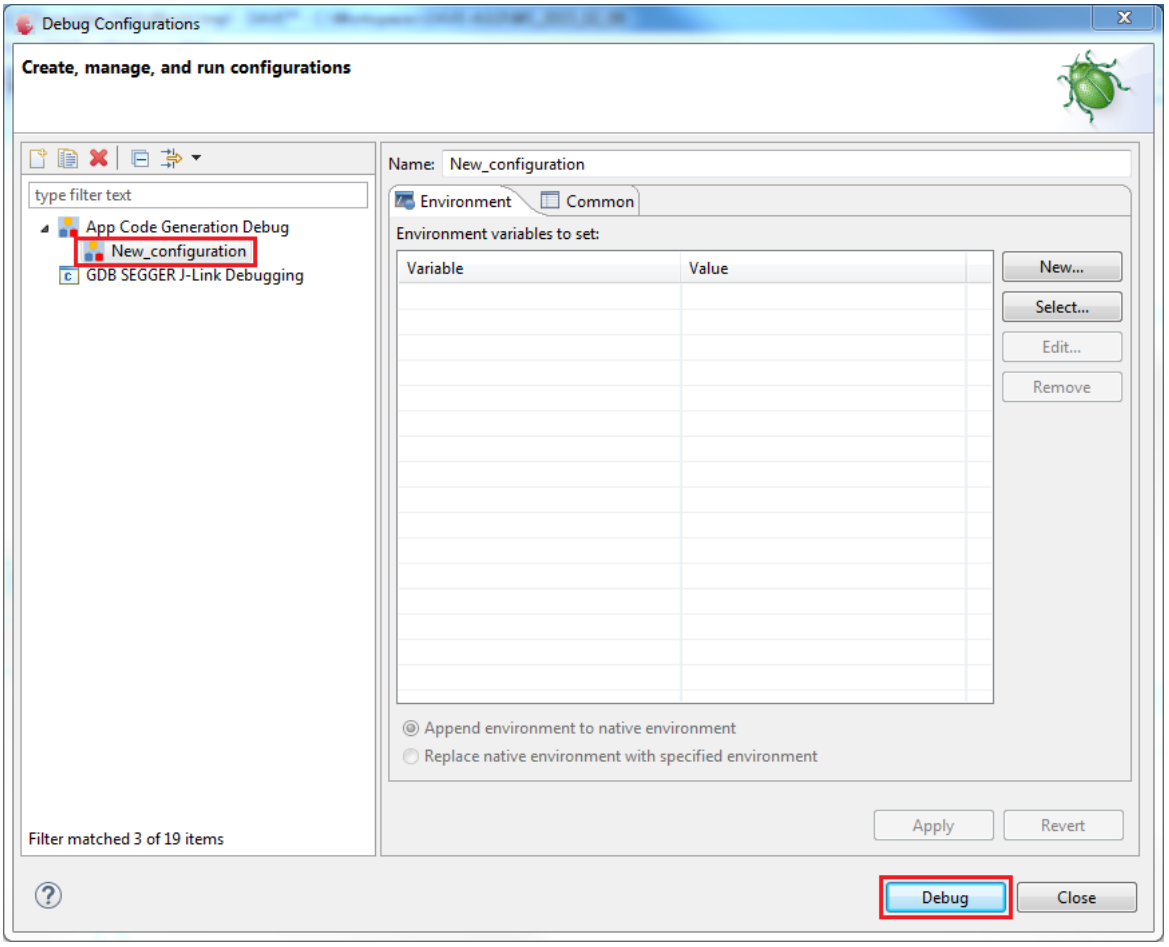


9. Now open **HelloWorld.tmpl** by **clicking it twice** and **add a breakpoint at line 8**.

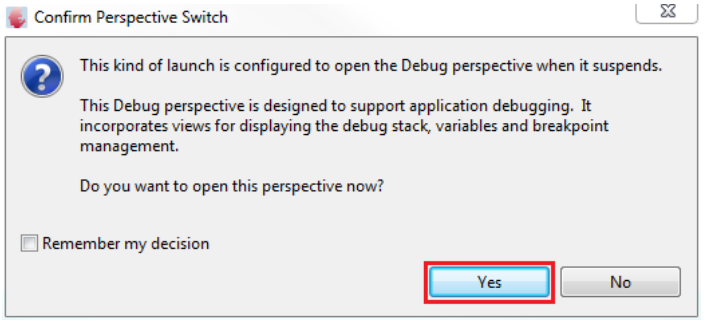


10. Time to debug! **Click on Debug HelloWorld** button.

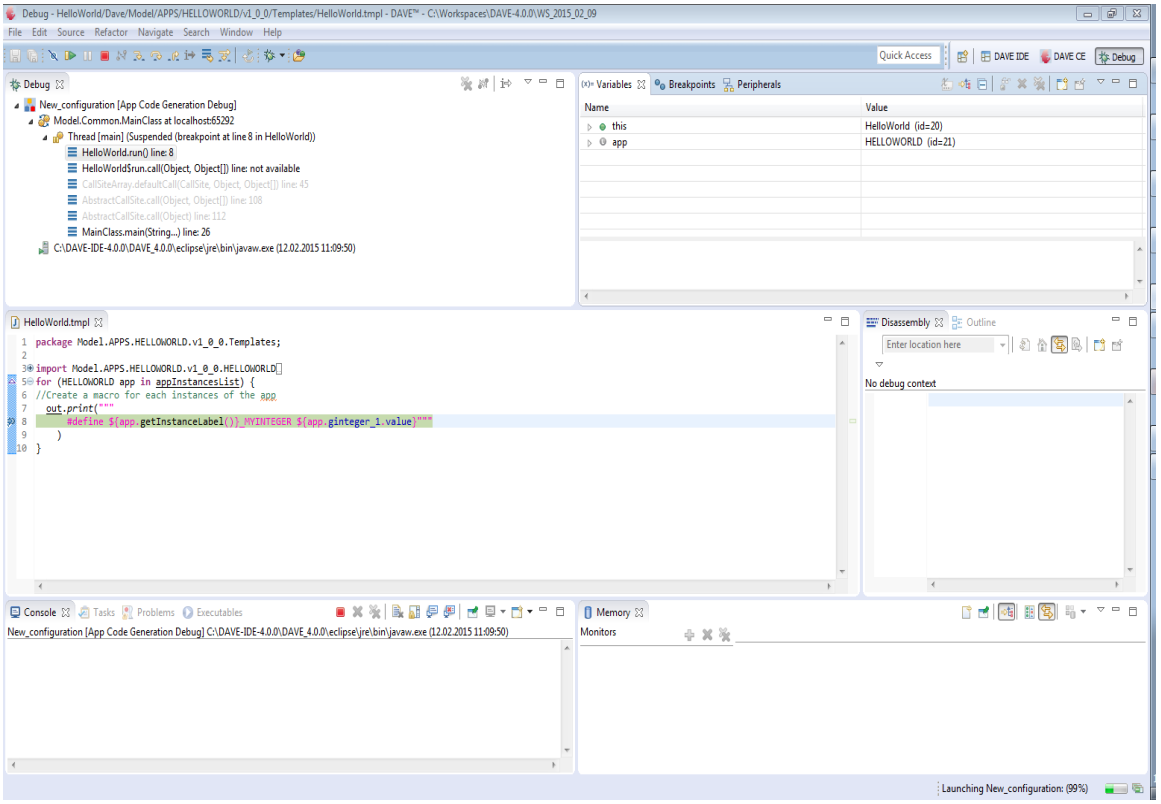
11. The **Debug Configurations window** will appear, select the **New_configuration** we have created before and click on **Debug** button.



12. A new window ask you about a **perspective switch**, click **Yes**.

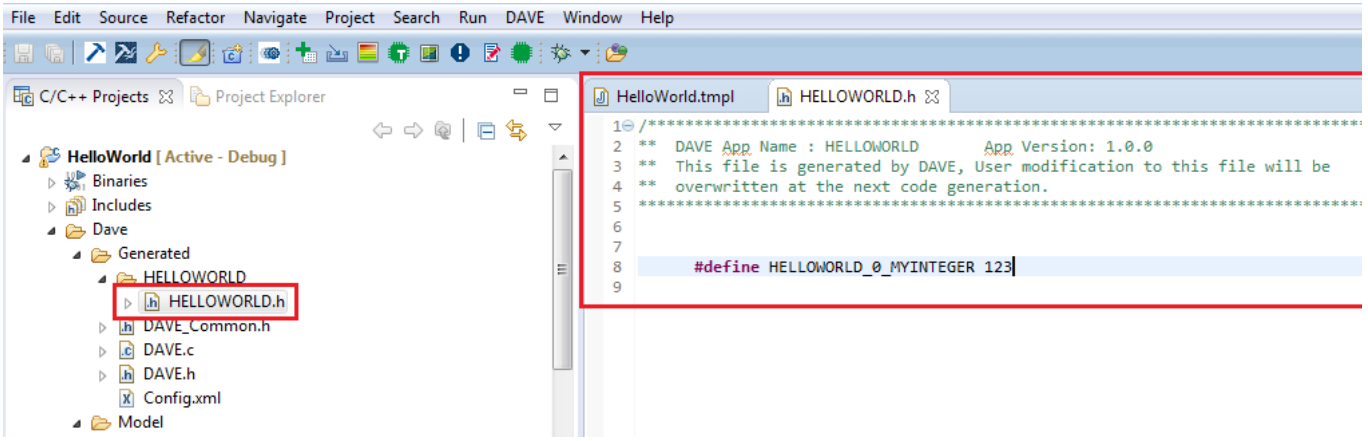


13. Congratulations, you are now able to debug your template!



Running this debug profile has the same effect like pressing the Generate Code button, but DAVE™ will show the debug perspective on first breakpoint.

14. Now you can see the generated code based on the template.
Go to **Dave > Generated > HELLOWORLD** and open **HELLOWORLD.h**



Step 8: Explore by yourself!

1. You have learned the basic steps through an **APP development process**, keep exploring a little bit more **HELLOWORLD** APP!
2. Try to add another instance of the same APP.
3. Play with the GInteger values.
4. Try to generate the code with and without debugging.
5. Look at the generated code results.

6. Enjoy!