

iSYSTEM winIDEA debugger

Abstract

This document describes tips and hints for using the legacy version of iSYSTEM winIDEA up until version 8 and the newer version of the debugger as target system. Normally, only a winIDEA project file template needs to be adapted to the specific configuration of your hardware.

Table of Contents

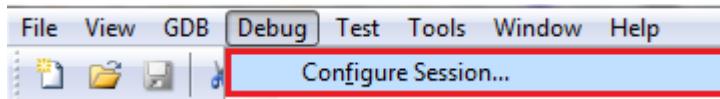
Abstract	1
1 Integration with newer versions of winIDEA	2
2 Integration with the Legacy Version of winIDEA.....	5
3 Prepare the Project File Template	5
3.1 Step 1: Hardware settings	5
3.2 Step 2: Disable the build manager	5
3.3 Step 3: Specify the binary to download	6
4 Specify the Project File Template	8
5 Using a Running winIDEA Instance.....	9
5.1 TEE settings.....	9
5.2 Launch winIDEA manually	9
6 Original Binary Test (OBT) with winIDEA	9
6.1 Debugging the test object.....	10
6.2 Enable logging	11
7 Troubleshooting.....	12
7.1 Endless loop when running the test.....	12
7.2 The test run gets stuck in ts_winidea_sync().....	13

1 Integration with newer versions of winIDEA

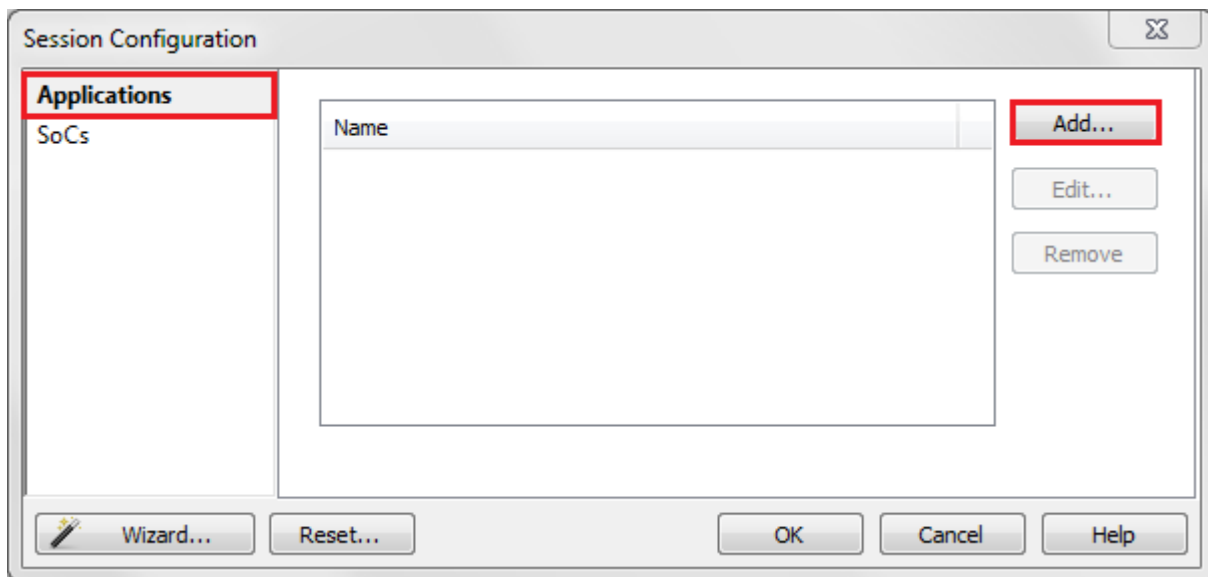
Since version 9, winIDEA comes with new menu and dialog options. If you are using an older version of winIDEA, please refer to chapter 2 of this application note.

It is strongly recommended to make a copy of your winIDEA development project and to adjust the copy so as to fit for the TESSY test run. The following subchapters will guide you through these few adjustments.

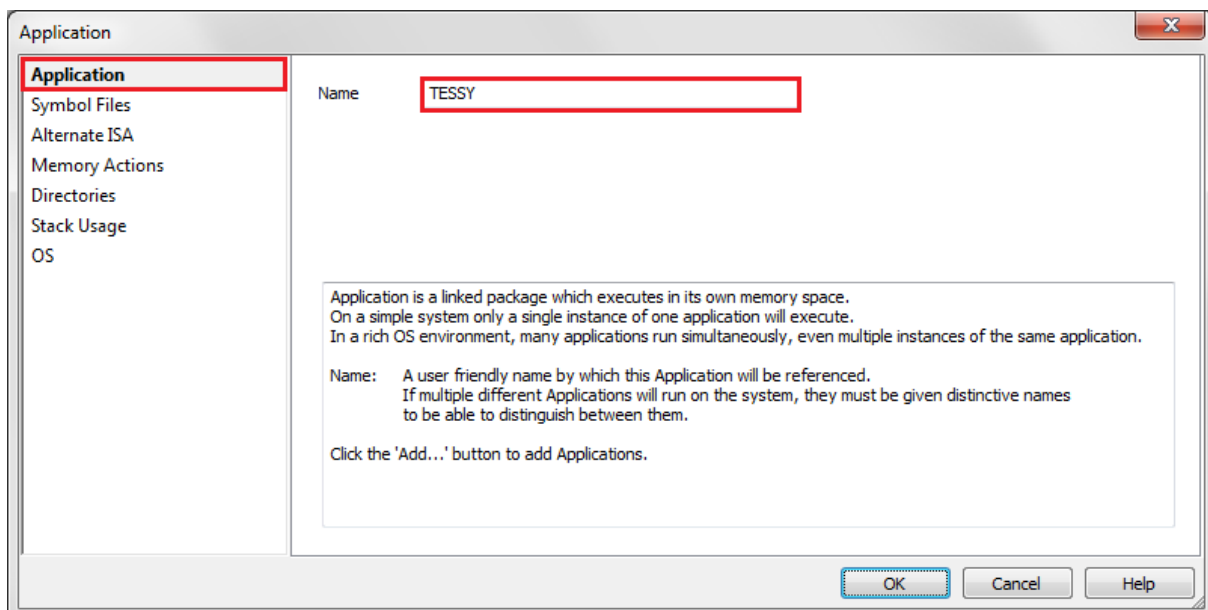
Open the **Session Configuration** dialog.



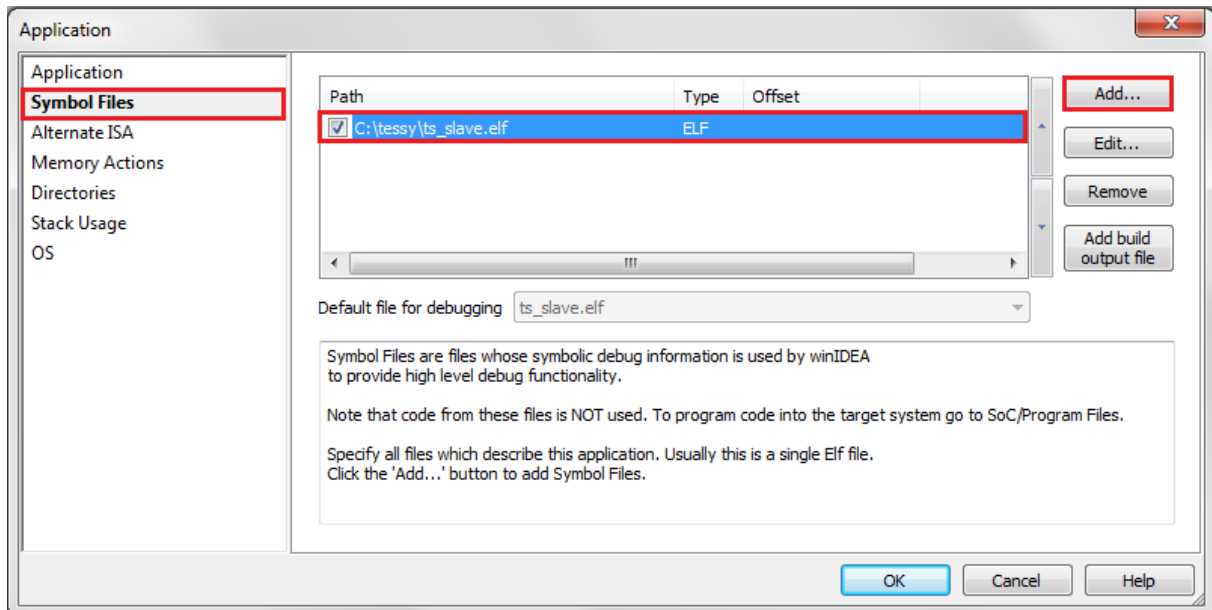
Select **Add...** or **Edit...** to open the **Application** dialog.



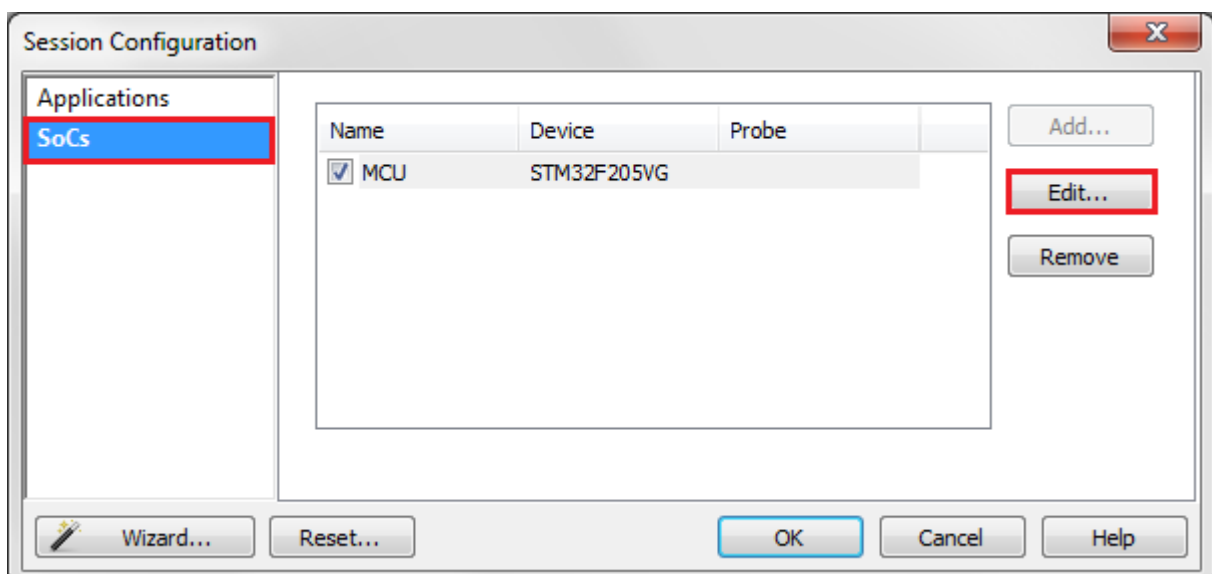
Enter a name, e.g. TESSY.



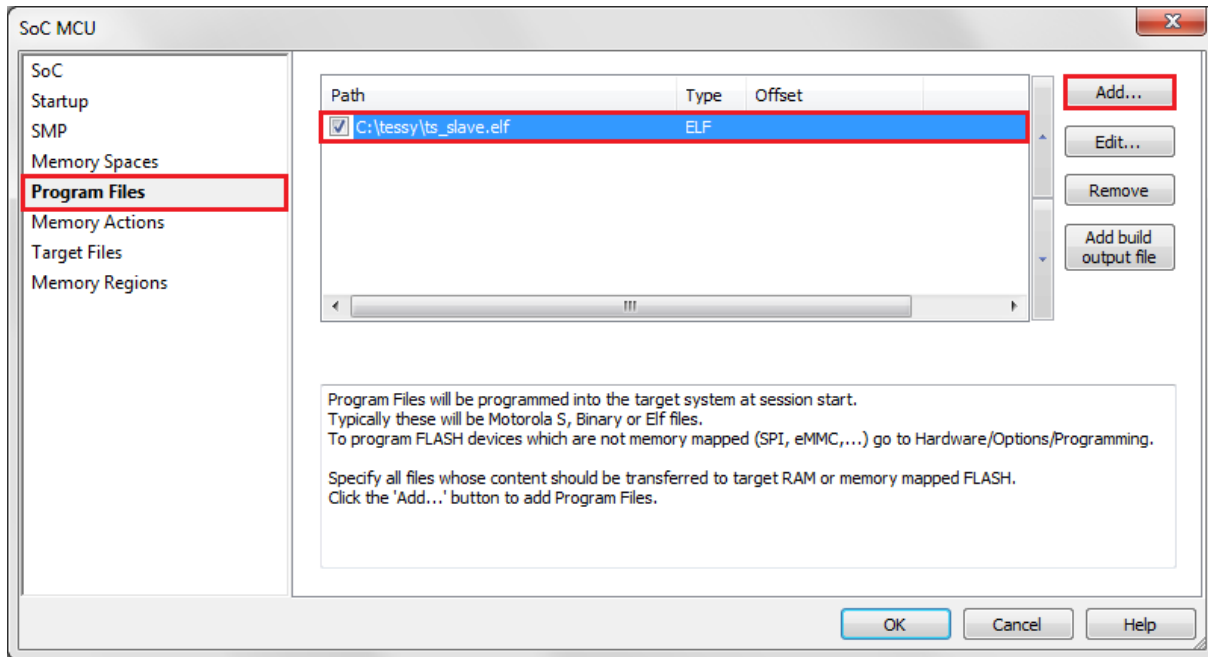
Select **Symbol Files** from the left column of the dialog. Click **Add...** to enter the path to the target binary file which is used as the symbol file as well as the program file during the test run. *The file has to reside within the TESSY test area.* If there is no file by now, let TESSY build the test driver and run the test, so that the target binary file will be copied into the test area automatically, or copy a target binary file from your winIDEA development project and rename it to `ts_slave.elf`. If the file extension is different, keep in mind to adjust the file extension within the TEE attribute **ExeFile Extension** as well.



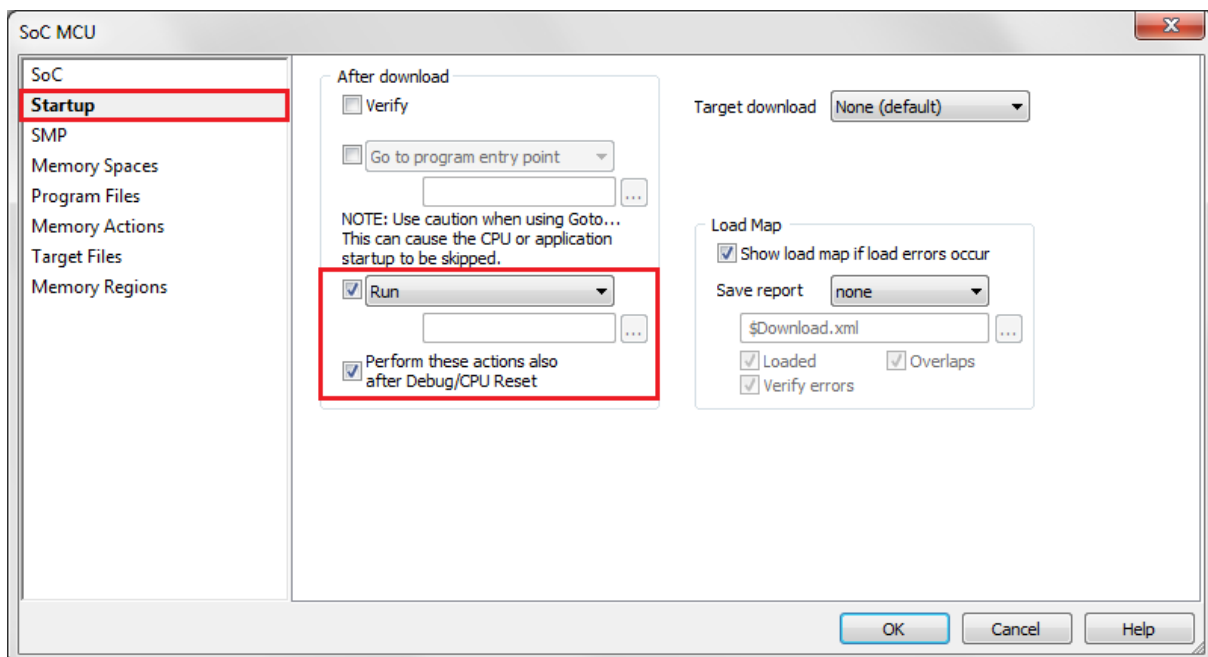
Click **OK** and select **SoCs** from the **Session Configuration** dialog.



Click **Edit...** to open the **SoC MCU** dialog. Select **Program Files** from the left column and click **Add...** to enter the target binary again.



Finally, select **Startup** from the left column and check if the dialog contains a **Run** option. If so, make sure the it is configured as shown below.



You can skip chapter 2 and chapter 3 and jump straight to chapter 4.

2 Integration with the Legacy Version of winIDEA

The integration with the legacy iSYSTEM winIDEA debugger uses a predefined project file in conjunction with a generated winIDEA script file to communicate with TESSY. The templates for the project file and the script file are located in the subdirectory `sys\targets\winidea` within the TESSY installation.

The project file specifies the slave application to be downloaded (By default, the binary will be copied to the TESSY testarea directory).

Please note: If you have selected a different location for the TESSY testarea (other than the default `C:\tessy`) during the installation process, you need to customize the winIDEA project file accordingly to the description below.

TESSY starts the winIDEA debugger for each test execution. Please make sure to close any running instance of winIDEA before starting the test run within TESSY.

3 Prepare the Project File Template

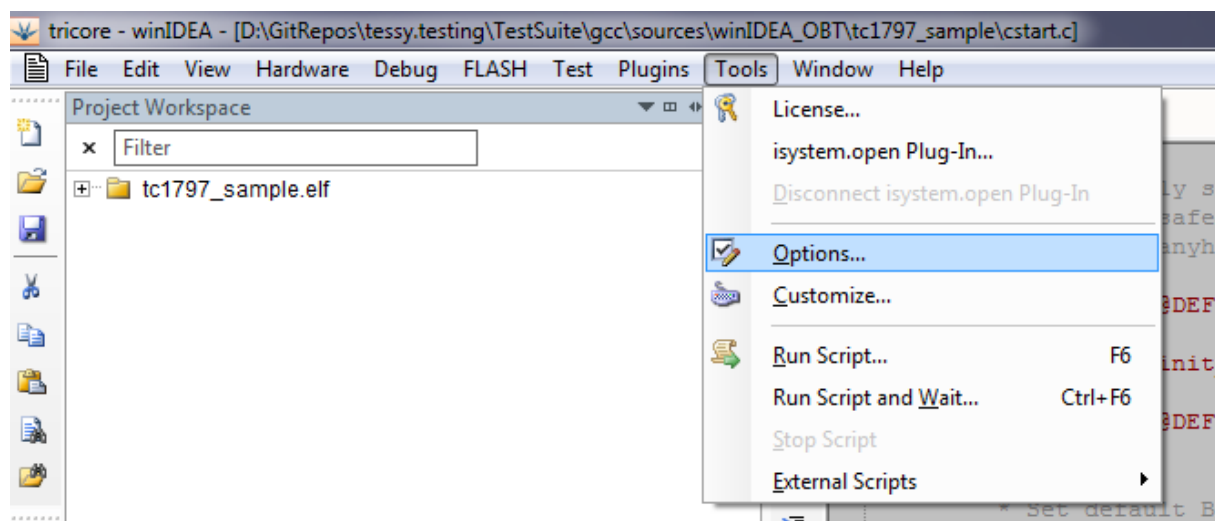
The winIDEA project file holds all required information for connecting to the target hardware and downloading the binary file. You need to specify the project file template to be used within the TESSY Environment Editor (TEE). You may either adapt the given templates or copy your own winIDEA project file and do the changes as described below.

3.1 Step 1: Hardware settings

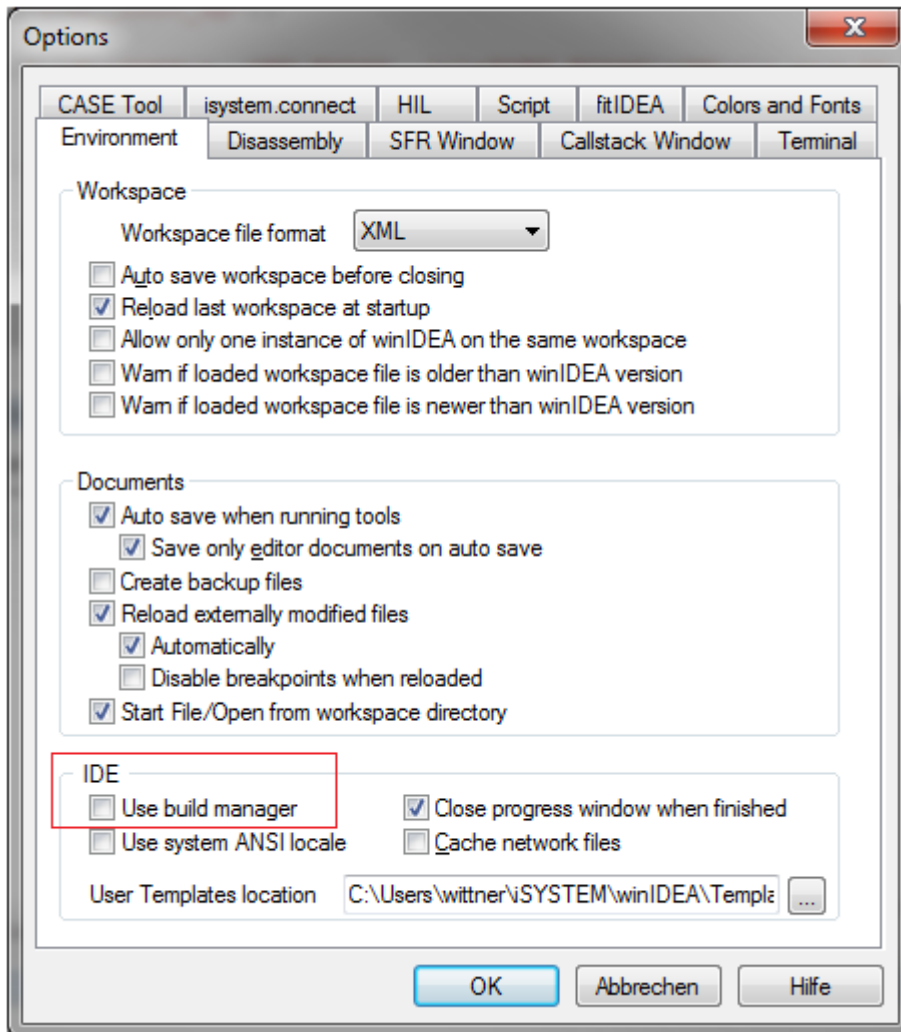
The given project templates in the subdirectory `sys\targets\winidea` will probably not fit to your specific hardware environment. Please review the settings or copy the project file which you are using for your own development and debugging purposes.

3.2 Step 2: Disable the build manager

Because the test binary will be created by TESSY, you need to disable the build manager within winIDEA as shown below: Select the **Tools->Options...** menu entry:

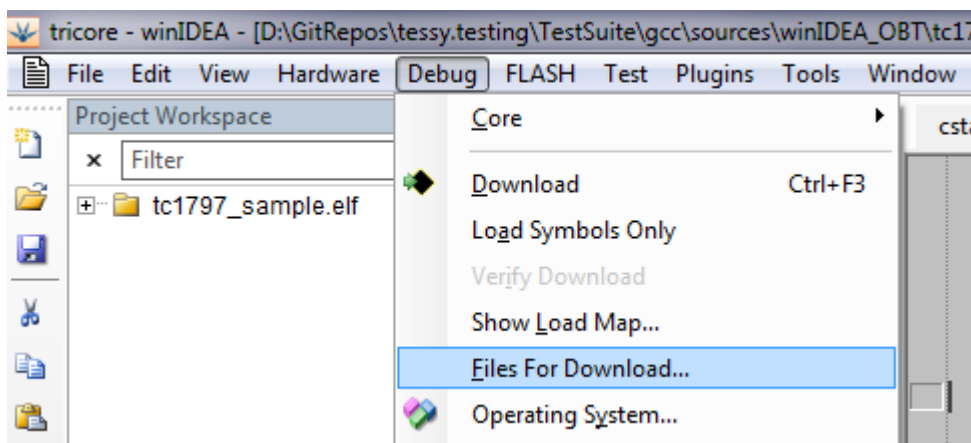


Within the options dialog disable the **Use build manager** option:

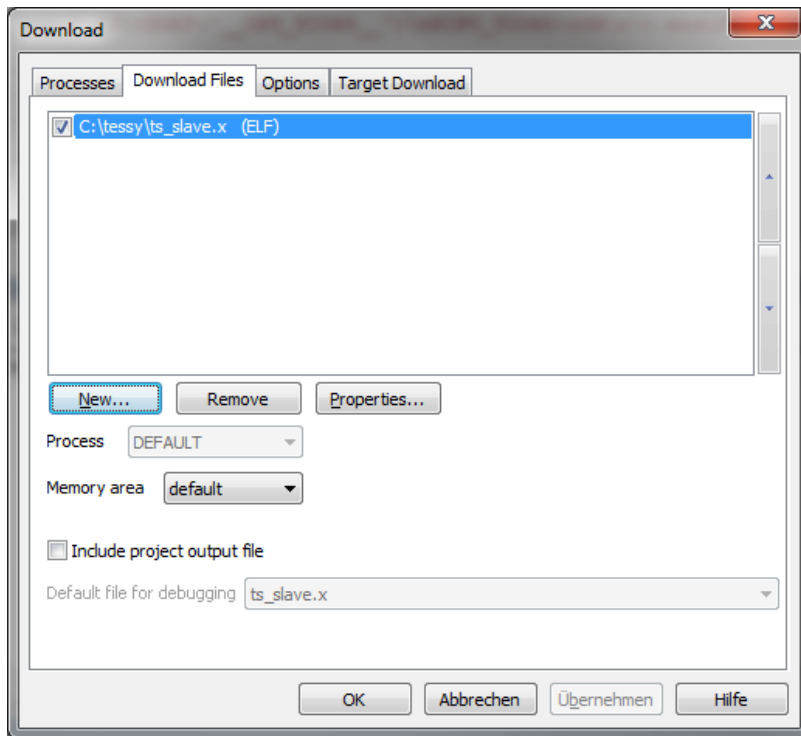


3.3 Step 3: Specify the binary to download

You need to tell winIDEA which binary to load before executing the test. Select **Files for Download ...** from the **Debug** menu.

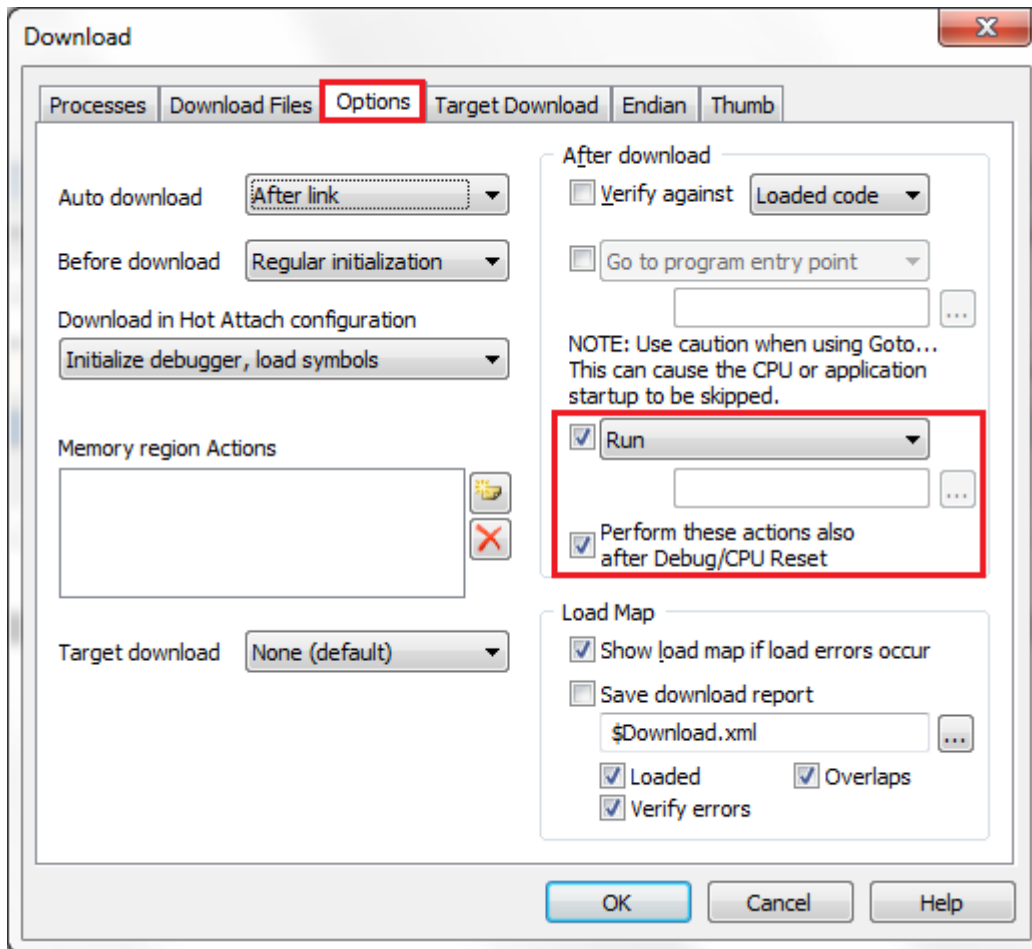


Within the **Download** dialog, check the list of files. TESSY will copy the generated binary from the module's test directory into the TESSY testarea directory (default `c:\tessy`). If you chose another location for the testarea directory during installation, then remove the file and choose the binary file at the appropriate location.



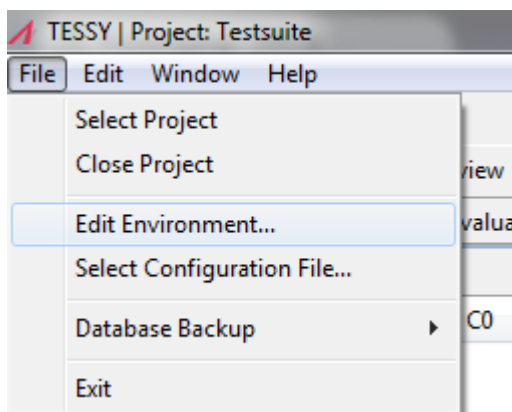
To ensure that the binary file is at the right location, please start the test execution within TESSY to copy the generated test application binary into the testarea directory. For some newer versions of winIDEA the **Run** option has to be enabled as shown below. Also enable option **Perform these actions also after Debug/CPU Reset**.

Please note: If the test run gets stuck after winIDEA was started and the binary was loaded, disable the **Run** option and try again.

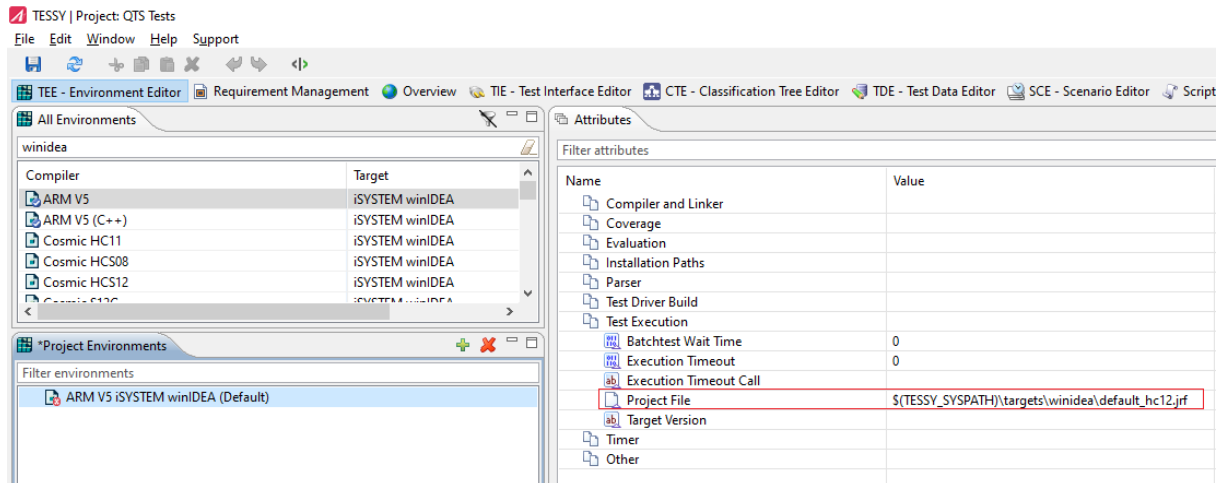


4 Specify the Project File Template

Within TESSY, you need to specify the project file template to be used for test execution within the environment editor (TEE).



Within your configuration for the given project (i.e. within the **File** node of TEE as shown below) select the respective winIDEA project file:



Please note: This project file template will be copied into the testarea directory for each test execution. All changes need to be carried out within the template file to take effect.

5 Using a Running winIDEA Instance

It is possible to run TESSY batch tests using the same instance of a running winIDEA IDE again and again. This chapter describes the prerequisites.

5.1 TEE settings

Please alter the TEE variable **Master Script Template** from

```
Master Script Template | $(TESSY_SYSPATH)\targets\winidea\tslowm_py.tpl
```

to

```
Master Script Template | $(TESSY_SYSPATH)\targets\winidea\tslowm_keep_running_py.tpl
```

On fast machines it might be useful to increase the **Batchtest Wait Time**.

```
Batchtest Wait Time | 2
```

5.2 Launch winIDEA manually

At first execute a normal TESSY test run in order to let TESSY generate the proper winIDEA project file in TESSY's test area. Next, launch winIDEA manually and open the winIDEA project file (C:\tessy\ts_slave.xjrf) from TESSY's test area. The setup is now prepared for TESSY batch tests.

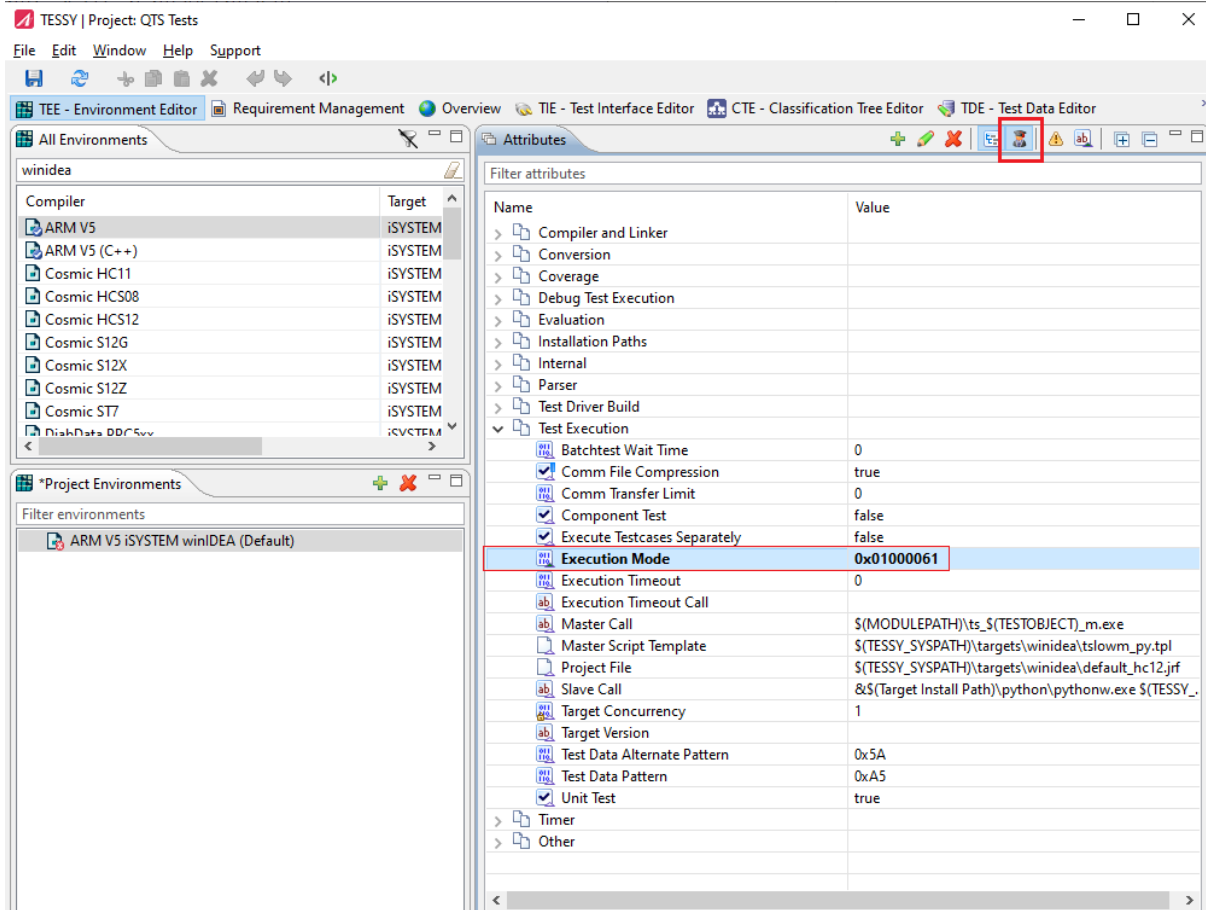
Please note: *If no running instance is found the script launches one on its own but closes the instance after each single test object test run again!*

6 Original Binary Test (OBT) with winIDEA

Since TESSY V3.1 there is a new feature available that provides testing the original binary application using the iConnect interface of winIDEA. The tests defined within TESSY will be written as YAML files which will be executed by winIDEA.

The OBT is supported for all winIDEA target environments that TESSY provides. You need to change the **Execution Mode** attribute within TEE to enable the OBT mode for winIDEA targets. This attribute is normally hidden, so that you will need to switch to “Expert” mode to see this attribute within TEE:

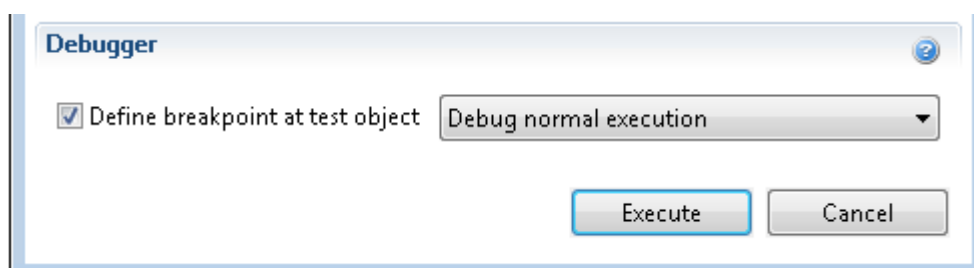
Change the value of the execution mode for the “iSYSTEM winIDEA” target to **0x01000061** as shown below (i.e. set the bit 0x01000000):



Now winIDEA will run all tests with the OBT mode. You need to start winIDEA and download your whole application. TESSY will then connect to winIDEA and execute each test by sending a YAML file to winIDEA.

6.1 Debugging the test object

If you want to find out why a test case does not yield the expected result, you can run the respective test case with the “Define Breakpoint at Test Object” feature of the TESSY test execution dialog checked:

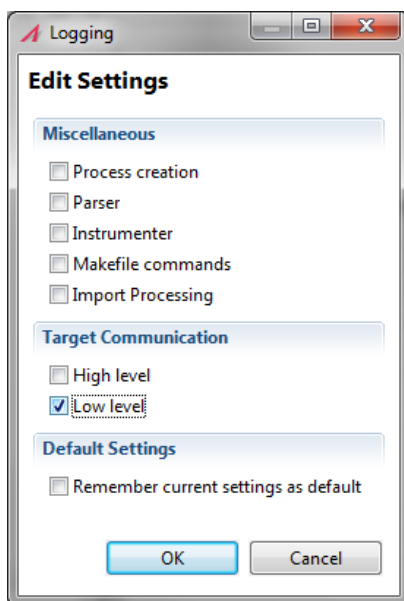


TESSY will then execute the YAML script and return (as if the test was aborted). Within winIDEA, test execution should be stopped at your test object and you can step through your code.

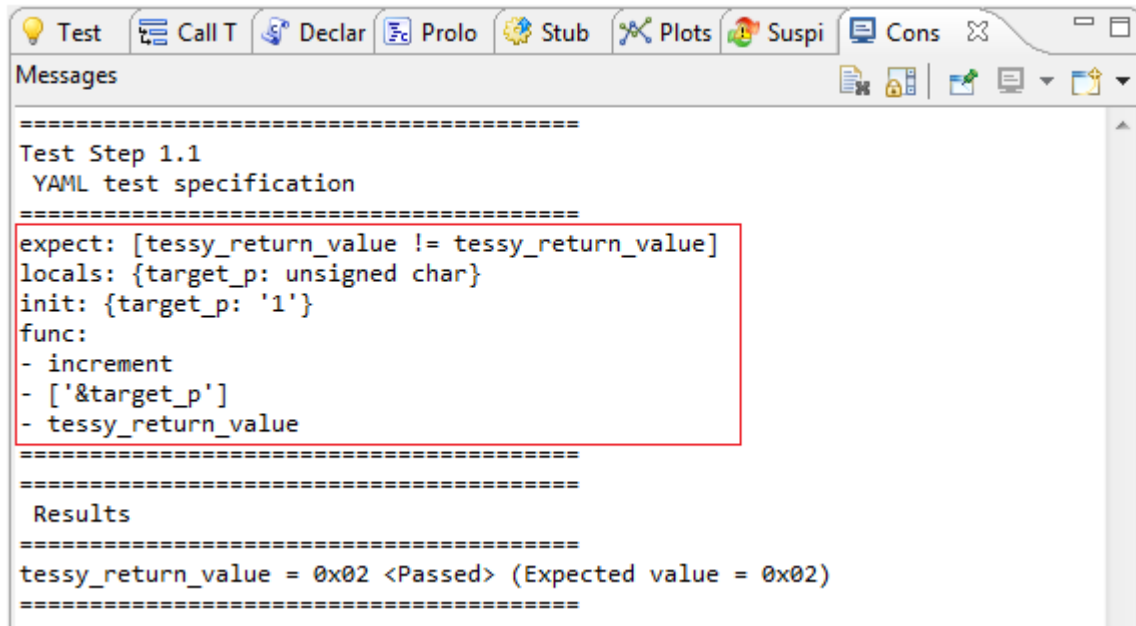
Please note: Only the first selected test case/step within TESSY can be debugged. It is not possible to run to the next test case/step within winIDEA because the YAML script is already finished.

6.2 Enable logging

If you encounter any problems running a test, it may be useful to switch on the **Low Level** logging (Choose **Logging...** from the **Help** menu):



This will print out the generated YAML script to the console view:



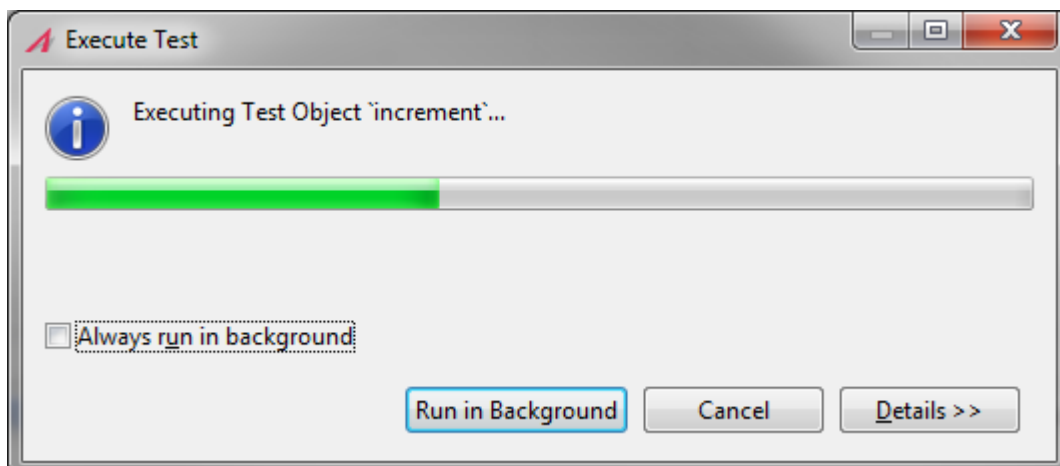
You may copy and paste the YAML script (the red marked part shown above) into testIDEA and run it from there to see more error messages or connect iSYSTEM support.

Please note: The expected results within the TESSY-generated YAML file will always yield failed results when running from within testIDEA. This is for technical reasons: TESSY parses the failed results to collect the actual result values.

7 Troubleshooting

7.1 Endless loop when running the test

If the build of the test driver has been successful and TESSY starts to execute the test, normally the winIDEA window should appear with a copy of the project file you specified as template (as described above). If winIDEA could not be started, TESSY will wait forever and display the execute test dialog as shown below:



In this case press the **Cancel** button and start winIDEA from the program menu manually.

Make sure to **disable all dialogues** that appear **on startup of winIDEA**. For use with TESSY, winIDEA needs to be started without any user interaction on startup.

Then close winIDEA and restart the test from TESSY. The winIDEA window should now be displayed and the test execution within winIDEA should start automatically (i.e. driven through a command line script generated by TESSY).

7.2 The test run gets stuck in `ts_winidea_sync()`

Launch winIDEA using your project file template, open the **Download** dialogue (Choose **Files For Downloads...** from the **Debug** menu), select **Options**, and disable the **Run** option. Try to execute the test run again.