



Tutorial for the experimental Java code generator and executor of the TITAN Eclipse





Intro

This tutorial shows how to create and use Java TITAN Eclipse Project.

This kind of project based on languages TTCN-3, ASN.1 and Java.

The tutorial is restricted only for the first steps.

It uses step by step instructions instead of general and full description.

Prerequisites



- Eclipse with installed TITAN EclipsePlug-ins. (See install guide)
- Installed TITAN Core (using C/C++ compiler) (if parallel mode is used)
- Environment is set.
- User is experienced in the TITAN based on C/C++.

Content



- Concept of a TITAN Java project
- Check the version of TITAN Eclipse plug-ins
- Check the TITAN installation path
- Create a HelloJavaTITAN project (create project, create ttcn3 module, create cfg file)
- Build and run a HelloJavaTITAN project in single mode
- Build and run a HelloJavaTITAN project in parallel mode

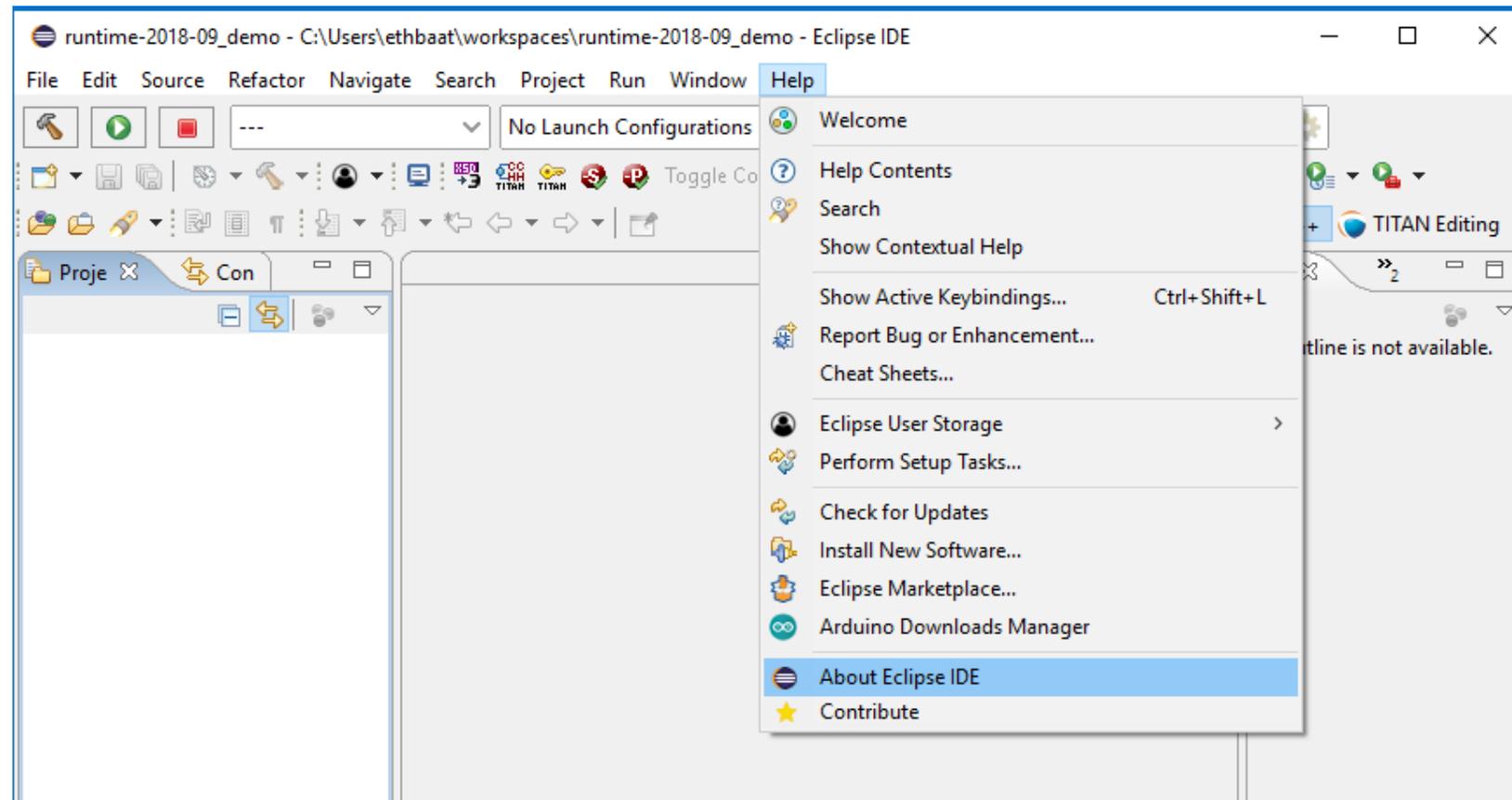
TITAN Java Project



- It is an Eclipse project with complete IDE support to develop and execute TTCN-3 based projects.
- It contains source files in TTCN-3 language (*.ttcn files), in ASN.1 language (*.asn files) and in Java language test (*.java) for implementation of test ports and external functions.
- It is designed to give support the following workflow:
 1. Generates Java files from TTCN-3, ASN.1 and Java files
 2. Compiles .java files to .class files
 3. Executes the compiled project(s) in single or in parallel mode

1. Check the TITAN Eclipse Plug-ins Version

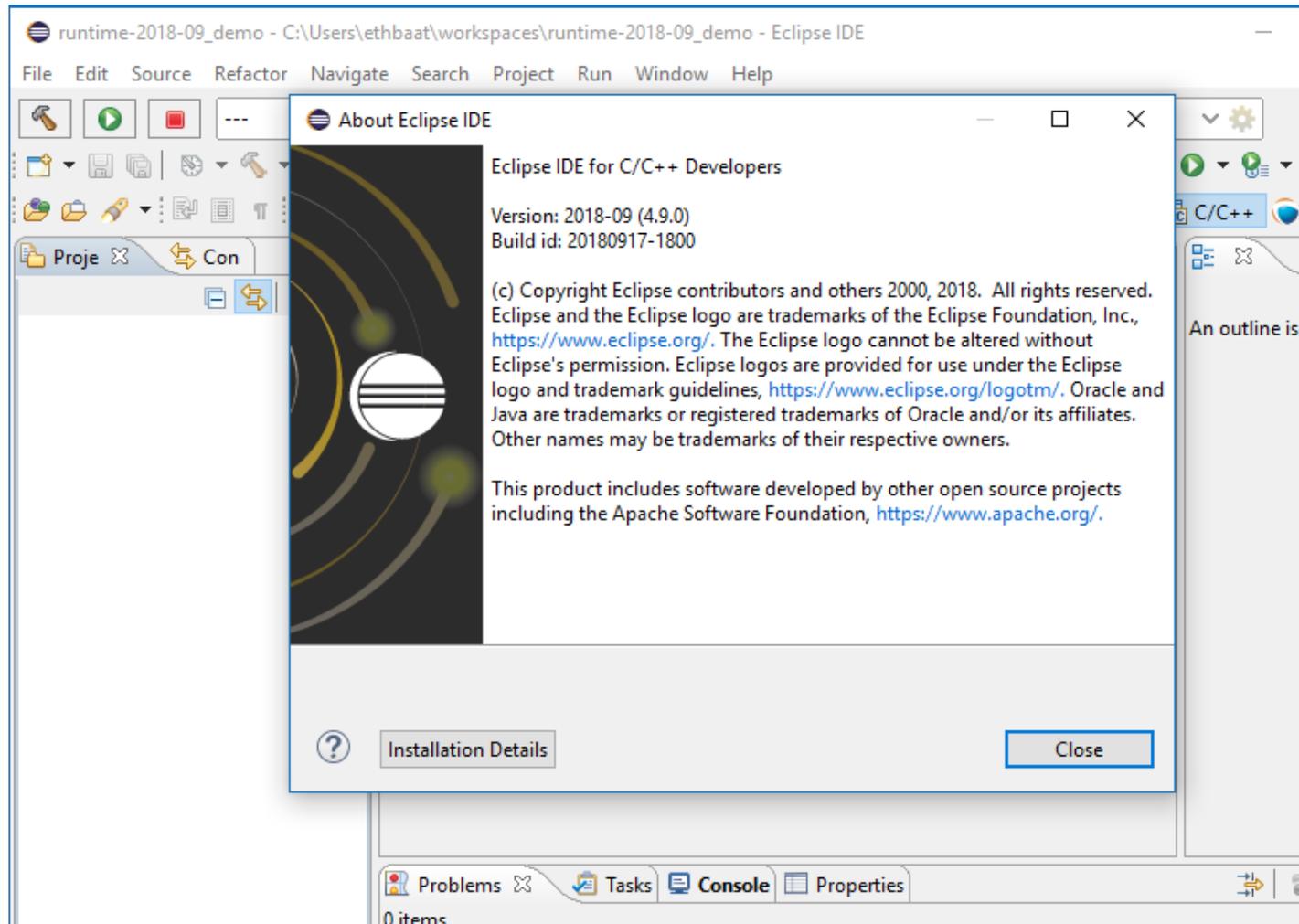
1. Start Eclipse with installed TITAN Eclipse plug-ins.
2. Check the TITAN version. It shall be at least 6.6.0 (CNL113200/6 R6A):
 - Select on menu Help>About Eclipse.



2. Check the TITAN Eclipse Plug-ins Version



— Push button “Installation Details”:



3. Check the TITAN Eclipse Plug-ins Version



— Check if the version is at least 6.6.0:

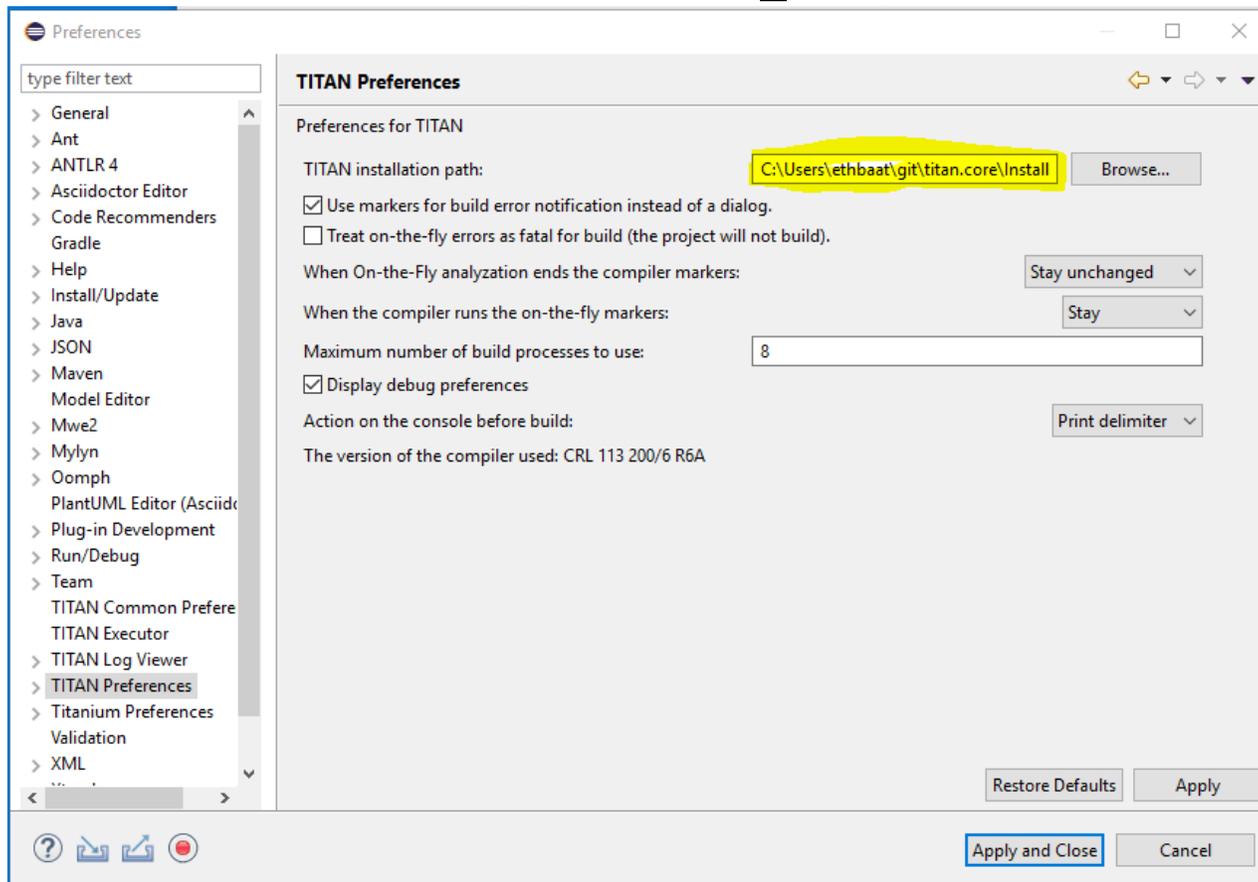
The screenshot shows the 'Eclipse IDE Installation Details' dialog box. The 'Plug-ins' tab is selected, displaying a table of installed software. The table has four columns: Name, Version, Id, and Provider. The 'TITAN' related plug-ins are highlighted in blue.

Name	Version	Id	Provider
> Mylyn Tasks Connector: Bugzilla	3.24.2.v20180905-0003	org.eclipse.mylyn.bugzilla_feature.feature....	Eclipse Mylyn
> Mylyn WikiText	3.0.25.201806201740	org.eclipse.mylyn.wikitext_feature.feature....	Eclipse Mylyn
> Oomph Setup	1.11.0.v20181203-0741	org.eclipse.oomph.setup.feature.group	Eclipse Oomph Project
RPM Tools	7.0.1.201809121533	org.eclipse.linuxtools.rpm.feature.group	Eclipse Linux Tools
> TITAN Log Viewer	6.6.0.CRL_113_200_6_R6A_...	TITAN_Log_Viewer.feature.group	Eclipse Titan Project
TITAN_Designer_Feature	6.6.0.CRL_113_200_6_R6A_...	TITAN_Designer.feature.group	Eclipse Titan Project
TITAN_Executor_Feature	6.6.0.CRL_113_200_6_R6A_...	TITAN_Executor.feature.group	Eclipse Titan Project
Titan_external_dependencies	6.6.0.CRL_113_200_6_R6A_...	Titan_external_dependencies.feature.group	
Titanium_external_dependencies	6.6.0.CRL_113_200_6_R6A_...	Titanium_external_dependencies.feature.g...	
Titanium_Feature	6.6.0.CRL_113_200_6_R6A_...	Titanium.feature.group	Eclipse Titan Project
> TM Terminal	4.5.0.201809101428	org.eclipse.tm.terminal.feature.feature.gro...	Eclipse.org - Target Manager
TM Terminal Serial Connector Extensions	4.5.0.201809101428	org.eclipse.tm.terminal.connector.cdtseria...	Eclipse.org - Target Manager
> Valgrind Tools Integration	7.0.1.201809121533	org.eclipse.linuxtools.valgrind.feature.group	Eclipse Linux Tools

4. Check the TITAN Installation Path



- Select menu `Window>Preferences>TITAN Preferences`. The TITAN installation path shall be the root folder of the TITAN (= `$TTCN3_DIR`):



1. Create a Titan Java Project (Hello World)

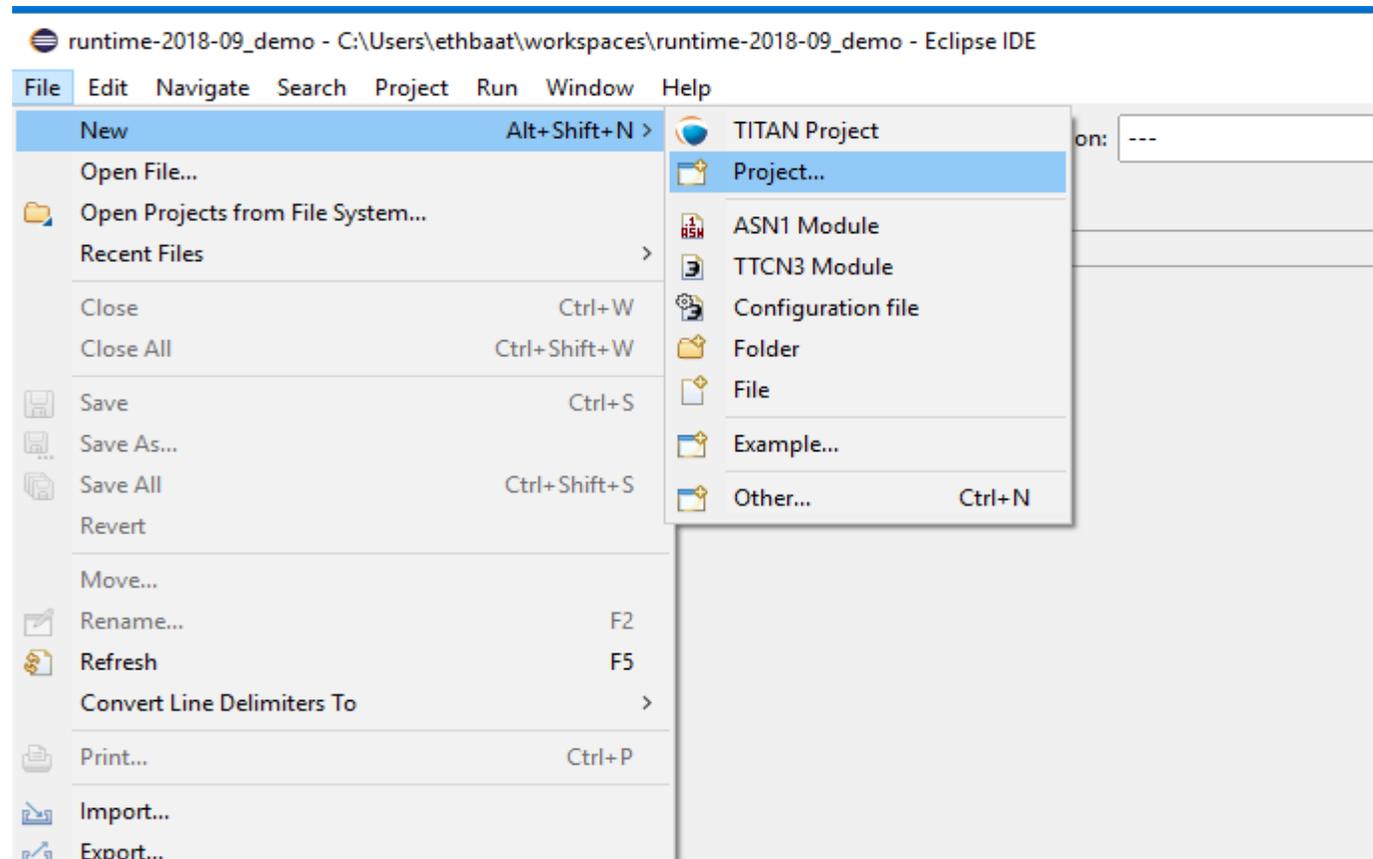


- Select the perspective “TITAN Editing” by selecting menu Window>Perspectives>Open perspective>Others>TITAN Editing
- Create a TITAN Java Project

Create a TITAN Java Project



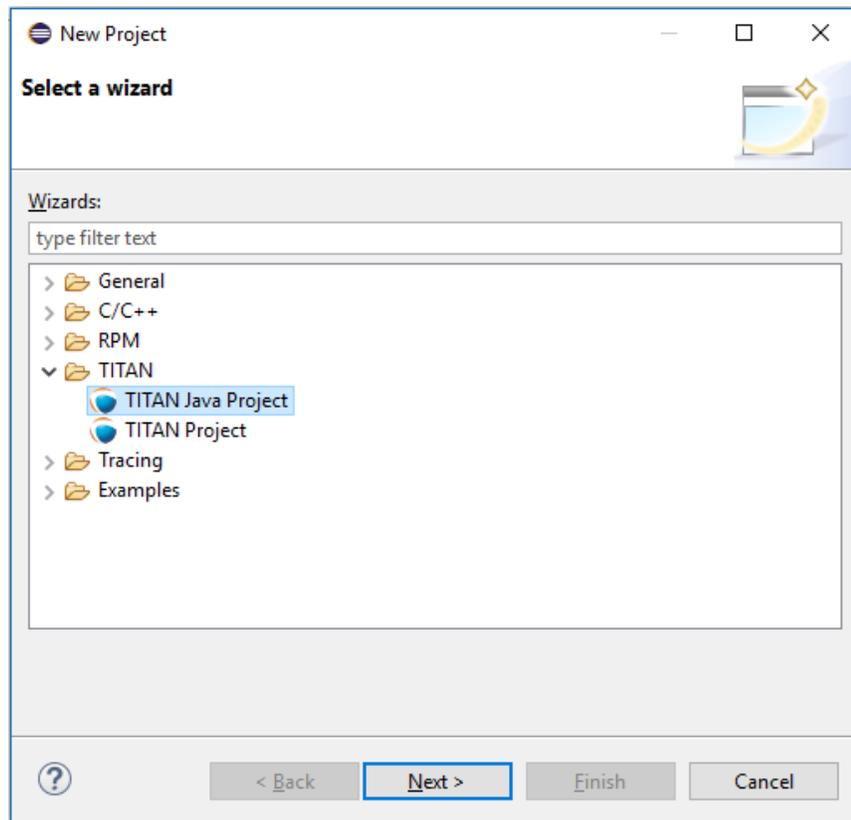
- Select menu File>New>Project... (Note: TITAN Project is not the good choice!)



Create a TITAN Java Project



— Select TITAN>TITAN Java Project in the “New project” dialog, then press Next.



Create a TITAN Java Project



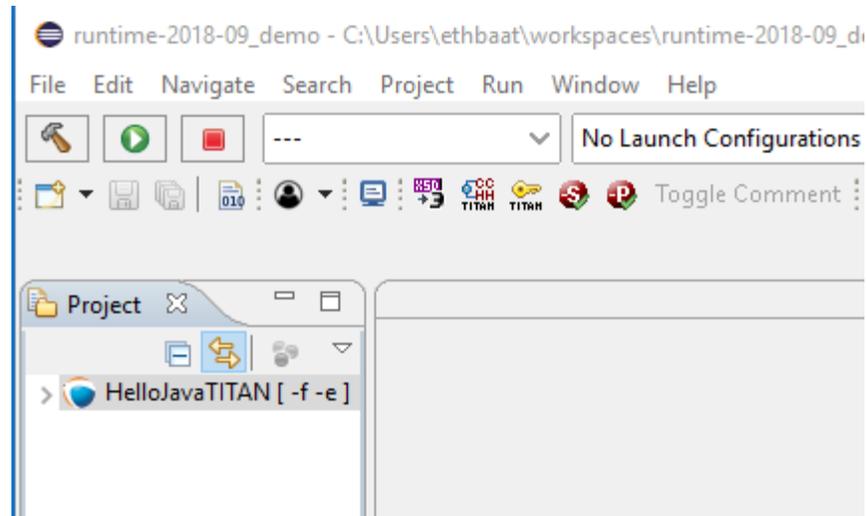
— Fill the field “Project name” in, then press Finish:

The screenshot shows a dialog box titled "New TITAN Java Project". The main heading is "Create a TITAN Java Project" with a sub-heading "Create a new TITAN Java project in the workspace or in an external location". The "Project name:" field contains the text "HelloJavaTITAN". The "Use default location" checkbox is checked. The "Location:" field contains the path "C:\Users\ethbaat\workspaces\runtime-2018-09_demo\Hello." and a "Browse..." button. The "Choose file system:" dropdown is set to "default". At the bottom, there are buttons for "?", "< Back", "Next >", "Finish" (which is highlighted with a blue border), and "Cancel".

Create a TITAN Java Project



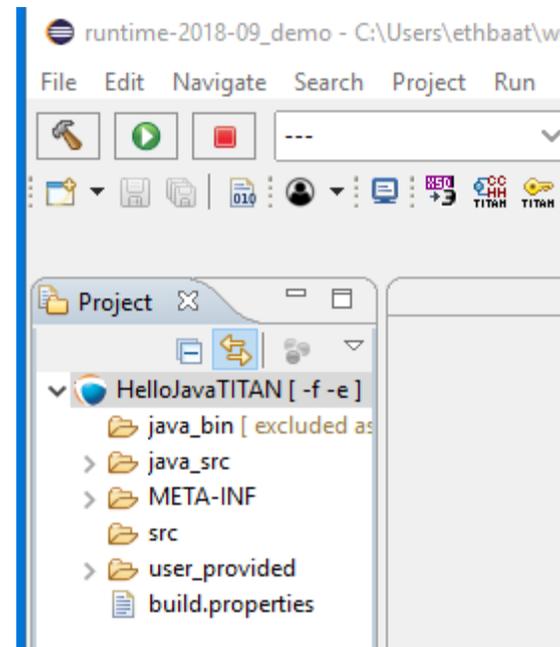
— The created project can be seen. Select the small arrow left to the project name:



Components of the Project



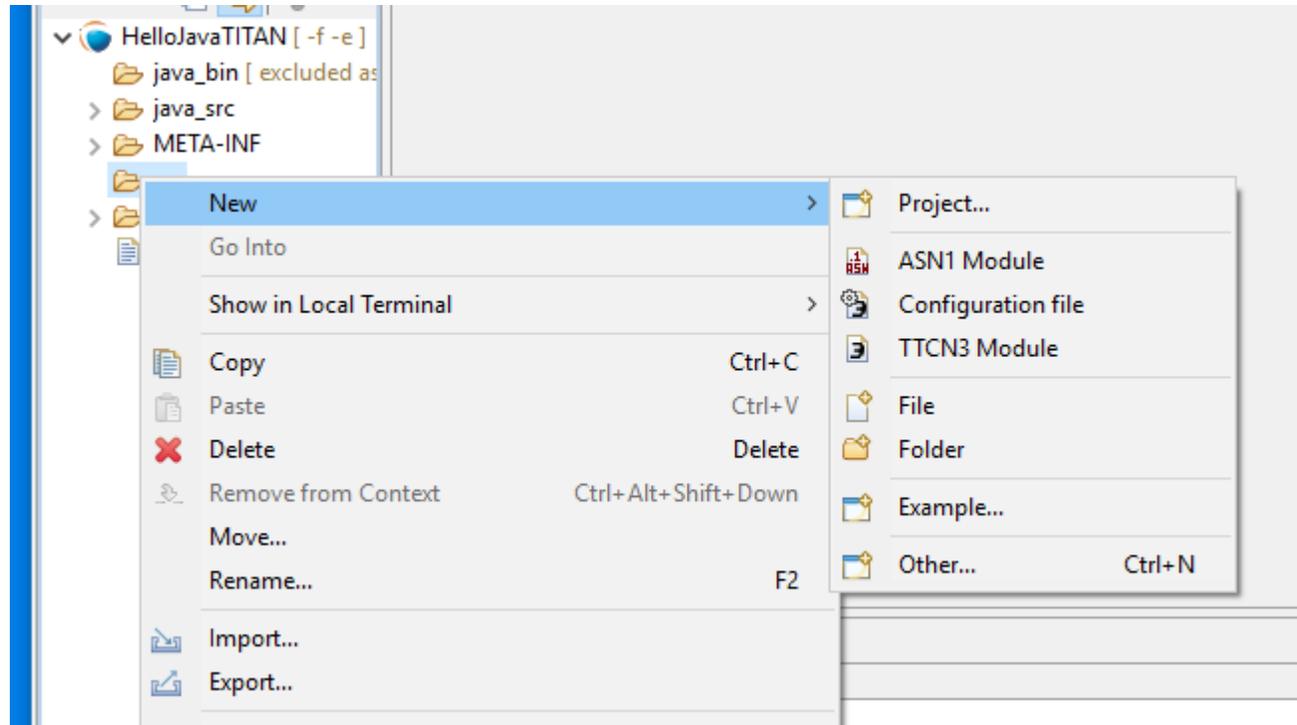
- We can see the content of the project. They role is as follows:
- src: .ttcn and .asn files
- user_provided: .java files, implementation of test ports and ttcn external functions
- java_src: .java files generated from the content of src
- java_bin: storage for compiled .class files
- Others are project administration files



Create a TTCN-3 Module



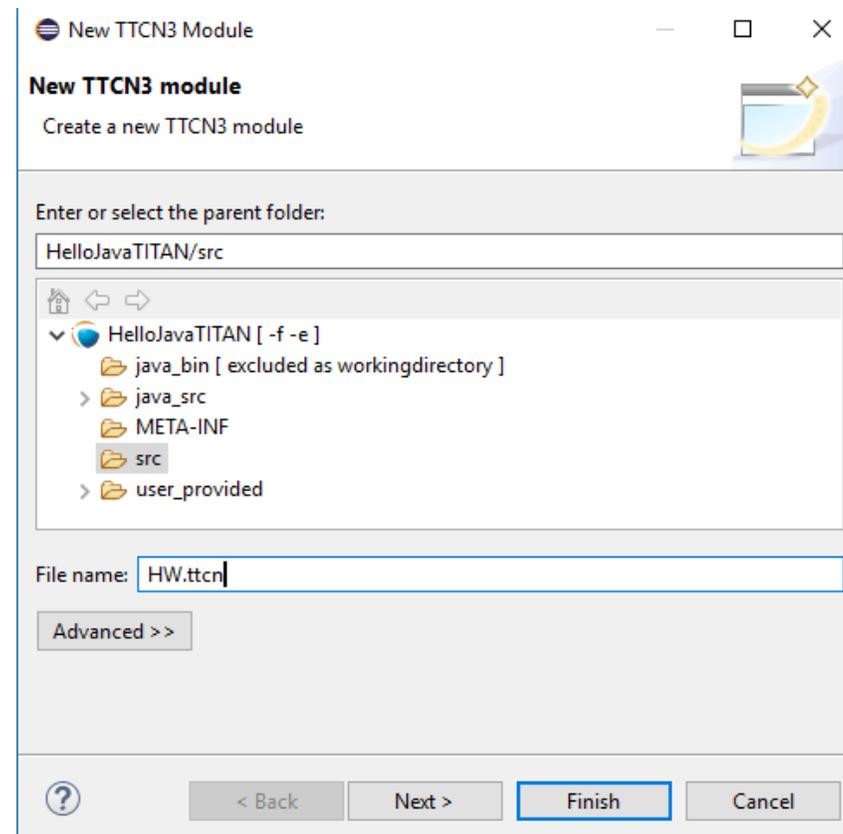
— Select New>TTCN3 Module in popup menu of folder HelloJavaTitan>src:



Create a TTCN-3 Module



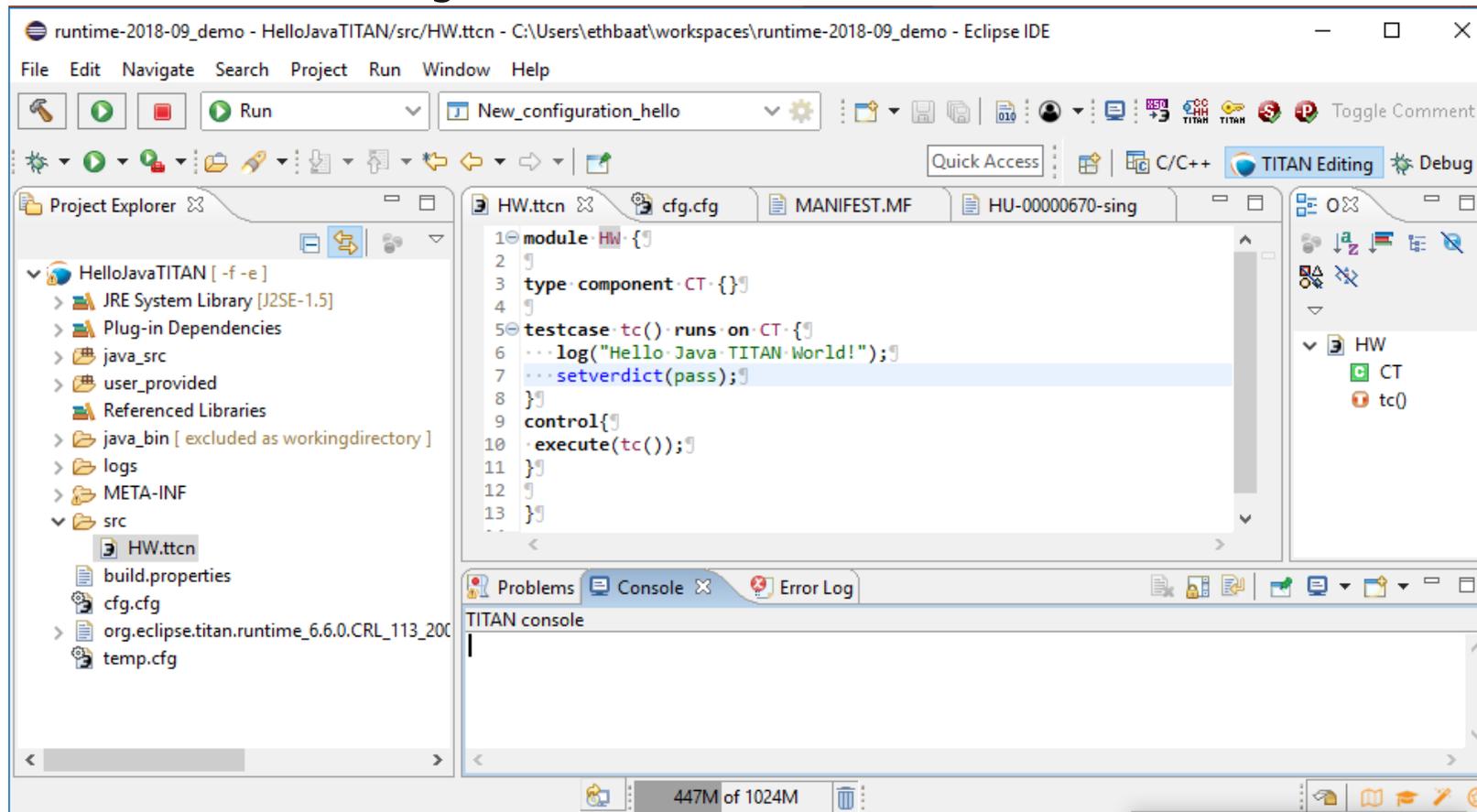
— Write a name in the field “File name” then press the button “Finish”:



Edit the File



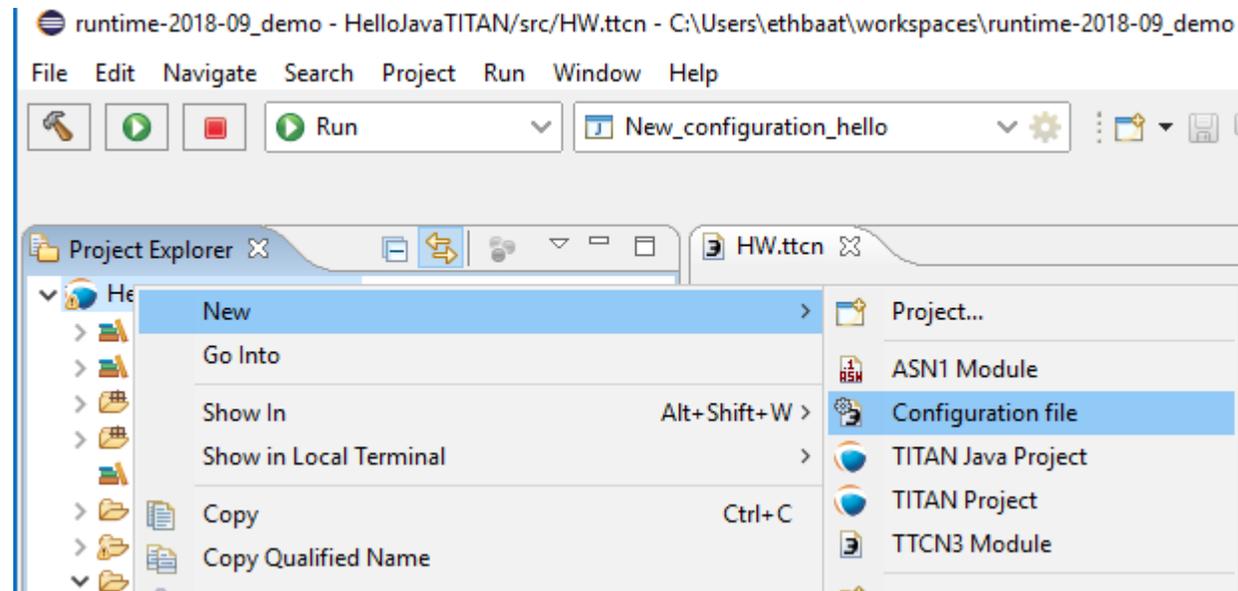
— Edit and save the file according to the screenshot:



Create a Configuration File



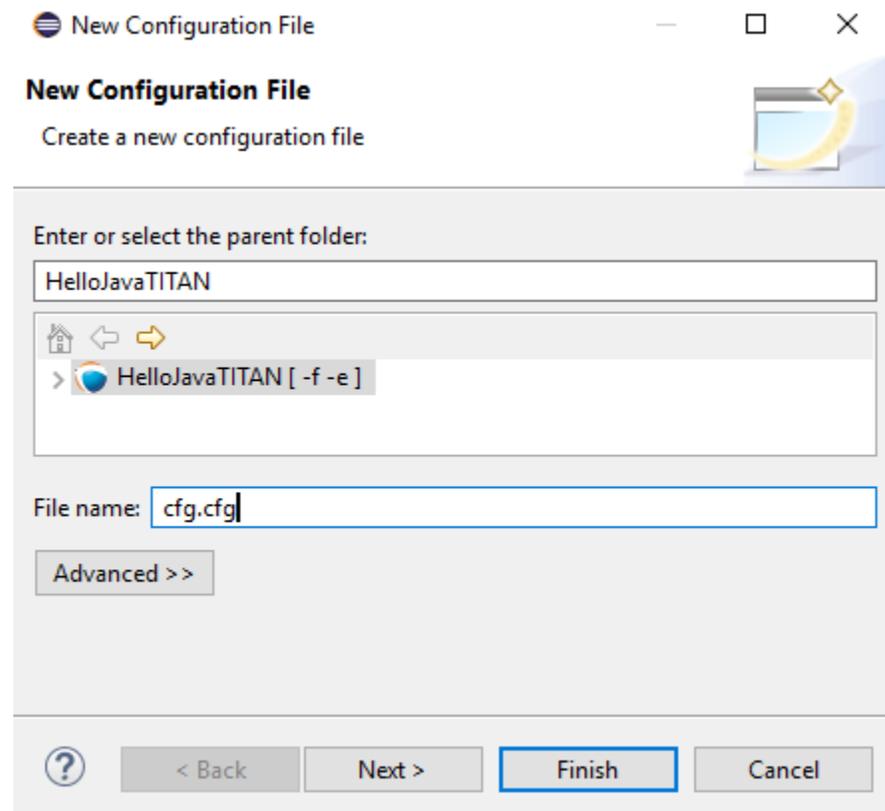
- Select “New>Configuration file” in popup menu of HelloJavaProject, then...



Create the Configuration File



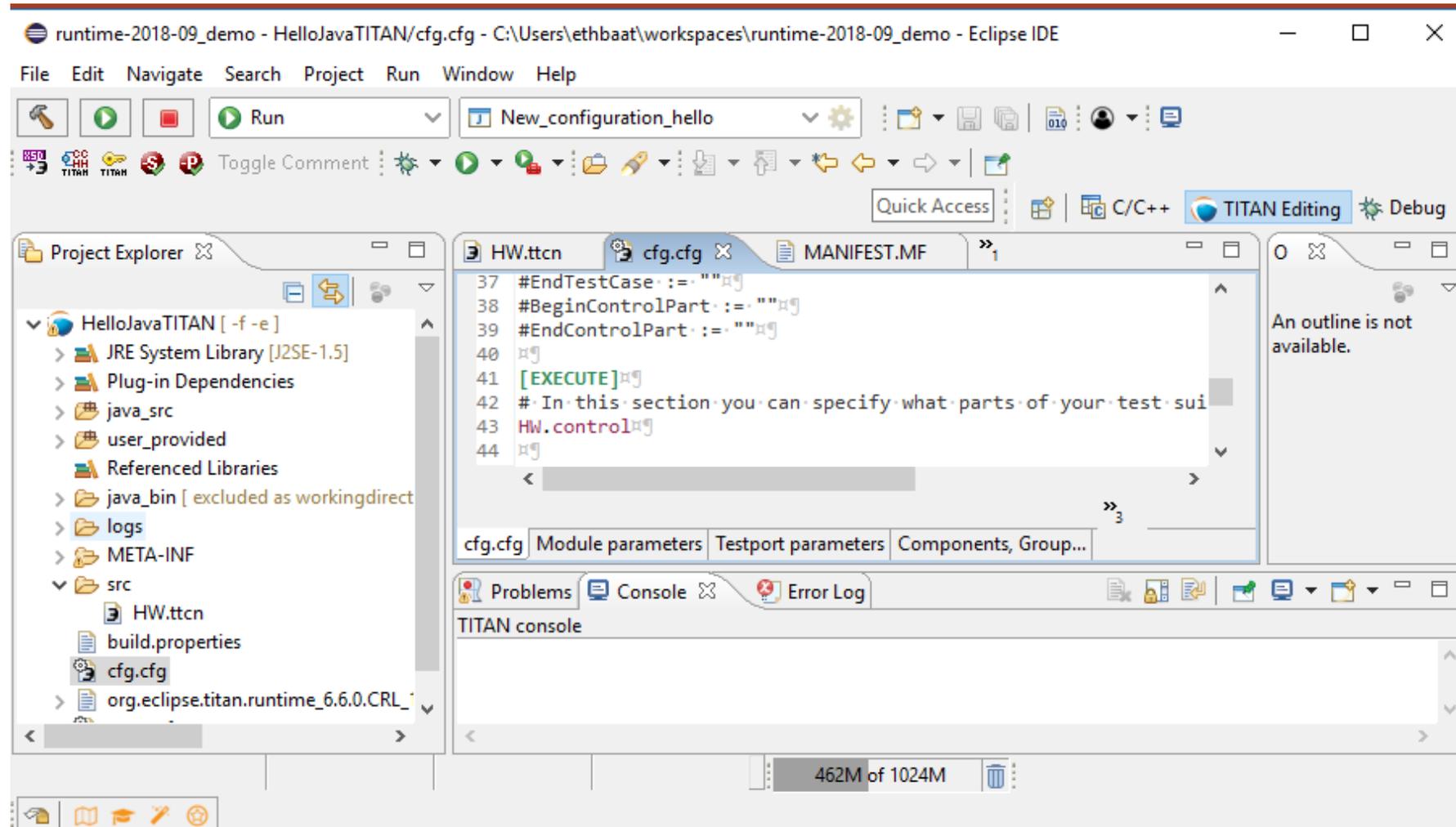
— Write a config file name in. Then push Finish.



Edit the Config File



— A config file editor opens. In the section [EXECUTE] , write the project name to execute its control part:



Edit the Config File

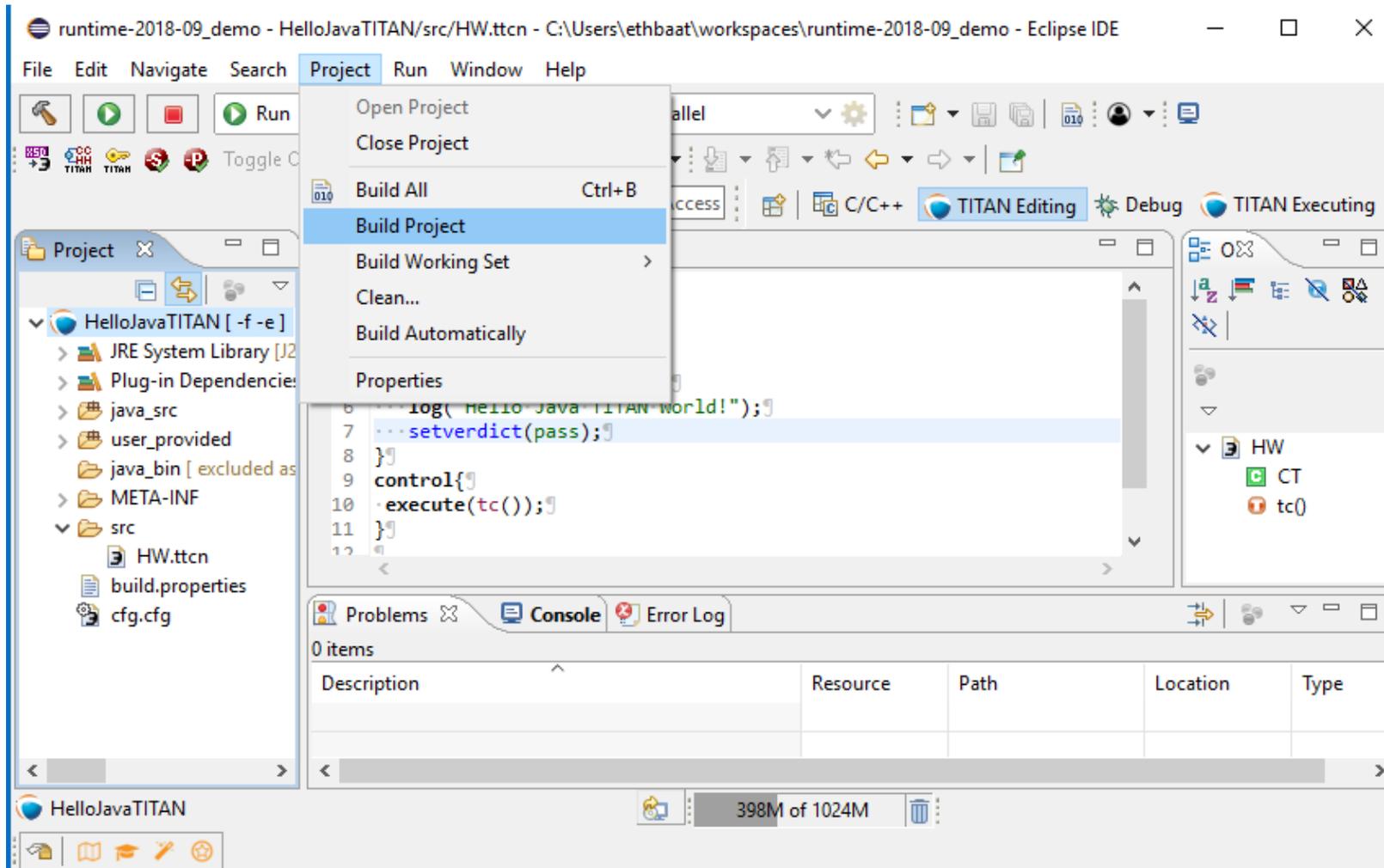


— Modify the section [LOGGING] with adding the USER log to the ConsoleMask:

```
ConsoleMask := USER | ERROR | WARNING | TESTCASE | STATISTICS | PORTEVENT
```

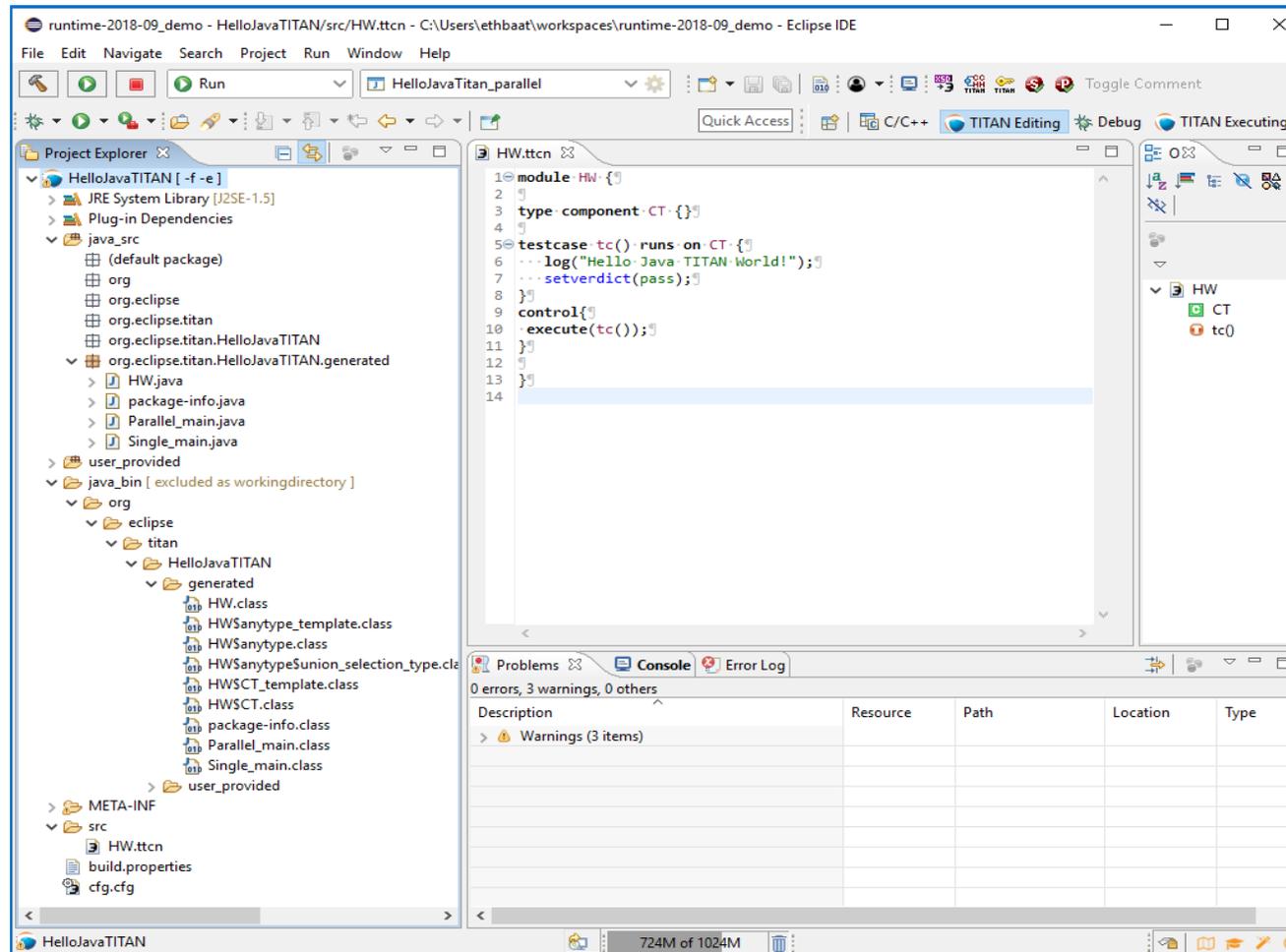
Build the Project

— Select “Build Project” in the popup window of project “HelloJavaTITAN” (or in Project>Build).



The Result of the Build

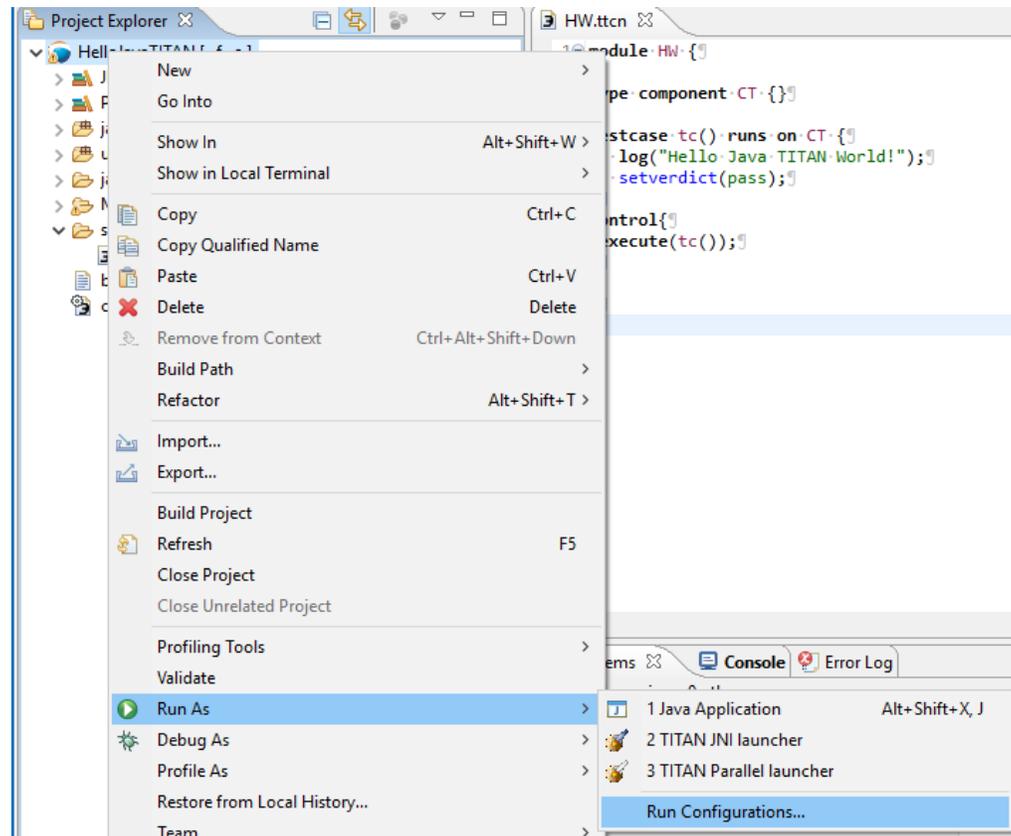
— “.java” files will be generated in the folder “java_src” and then “.class” files in the java_bin folder:



Running the Project in Single Mode



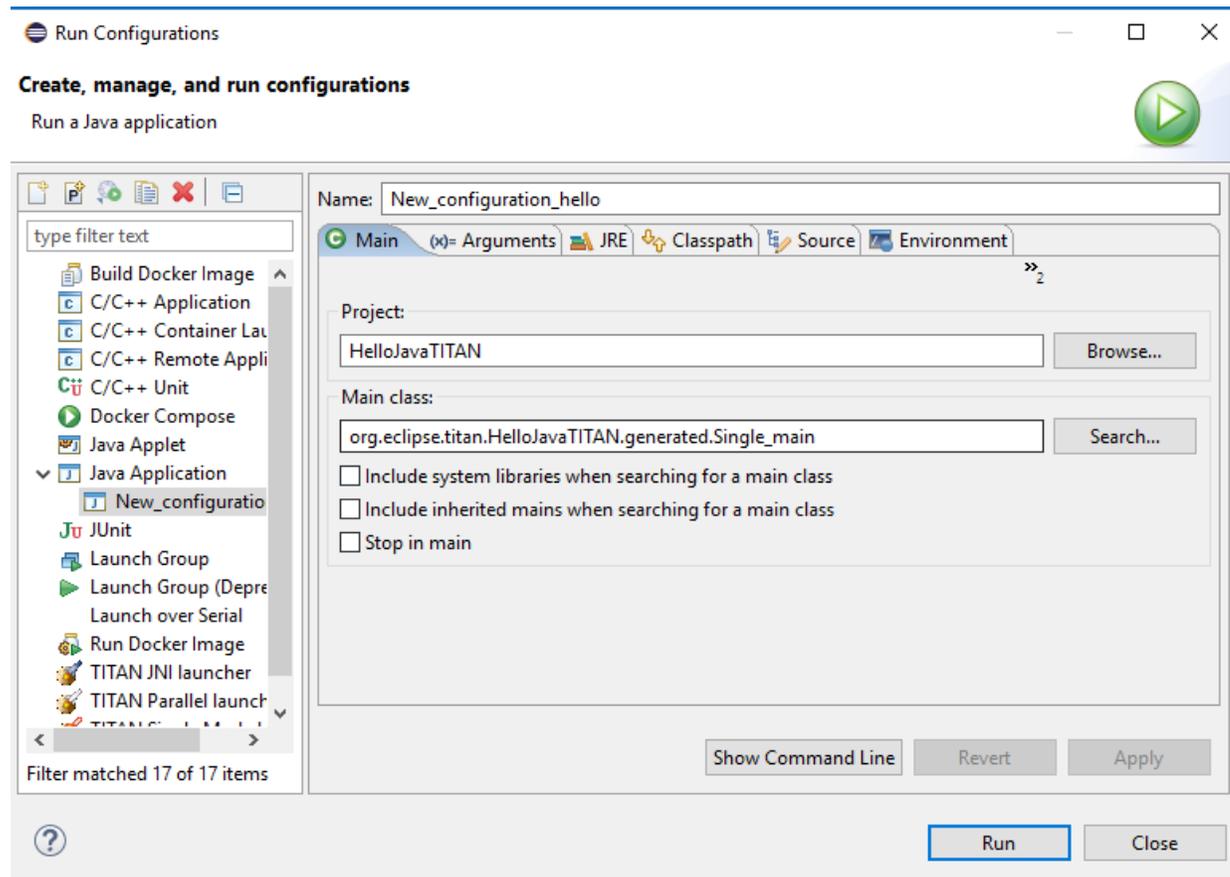
— Select Run As> Run Configurations... in the popup menu of the project, then...



Running the Project in Single Mode



- Double click on “Java Application” (or click on “Java Application” and then the “New Configuration” button. Fill the fields in. The main class shall be selected from list. After this select the tab “Arguments”



Running the Project in Single Mode

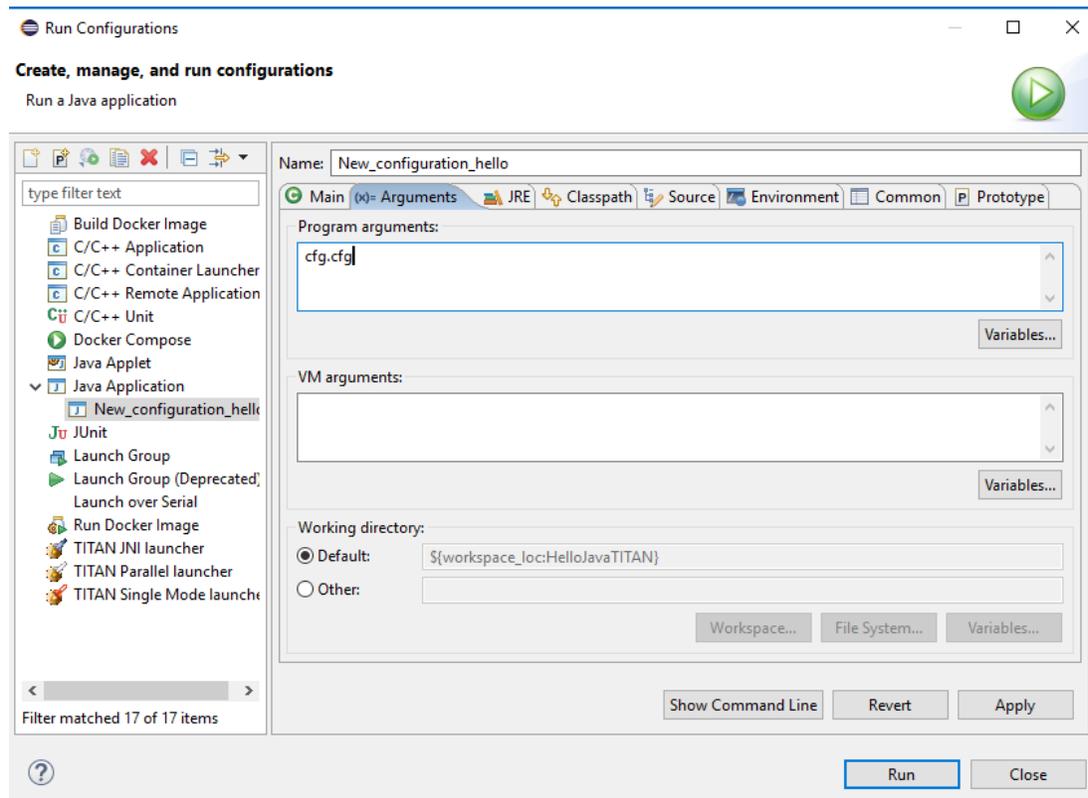


- **Note:** In single mode, the execution relies totally on the Java workflow: the Java code generated by the codegen will be built and executed.

Running the Project in Single Mode



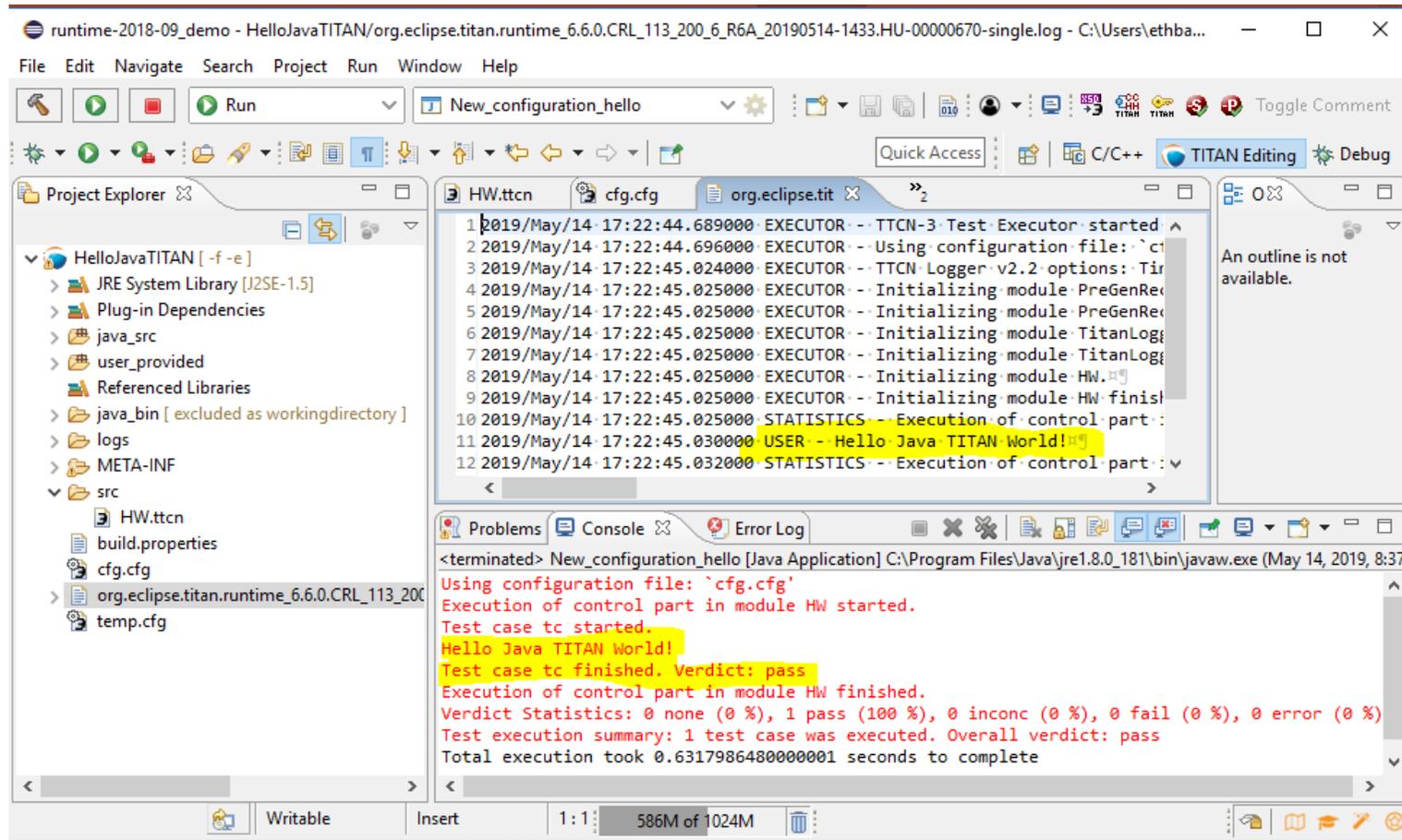
- In the tab “Arguments”, in the field “Program arguments”, write the name of the cfg file (with path relative to the project folder). It is correct only in single mode! Push the button “Run”!



The Result of the Running



— The project will be executed. The run will be logged on the console and in config file(s):
The console and the file log:



Running the Project in Parallel Mode



- **Important note:** as the main controller has not been yet written in Java, code execution in parallel mode relies on the legacy (C++) main controller, that has to be started from command line. This results in a hybrid execution: the legacy main controller and the Java components will work together to get the expected result. In the future, this will change into a pure Java execution, much simpler from users point of view, with no need for commands in a separate console.

Running the Project in Parallel Mode

Set the internal TCP Port in the Config File



- If the project contains more than one component (contains component create statement) then the project can run only in parallel mode. But even our first project can run in parallel mode.
- In parallel mode the Main Test Controller (mctr_cli) will control the Host Controller (HC) and every parallel test component (PTC).
- Before running we suggest to define the internal TCP port number, which the mctr_cli controls the components. It requires to step:
 1. In the cfg file set the internal TCP Port:

```
51
52 [MAIN_CONTROLLER]
53 # The options herein control the behavior of MC.
54
55 TCPPort := 6999
56 KillTimer := 10.0
57 # NumHCs := 0
58 # LocalAddress :=
59
```

Running the Project in Parallel Mode

Start the Main Controller



2. Start the `mctr_cli` in a terminal, in the project folder. It will listen on the port specified in the `cfg` file.

```
~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
ethbaat@HU-00000670 ~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
$ cd ~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
ethbaat@HU-00000670 ~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
$ mctr_cli cfg.cfg

*****
* TTCN-3 Test Executor - Main Controller 2 *
* Version: CRL 113 200/6 R6A *
* Copyright (c) 2000-2019 Ericsson Telecom AB *
* All rights reserved. This program and the accompanying materials *
* are made available under the terms of the Eclipse Public License v2.0 *
* which accompanies this distribution, and is available at *
* https://www.eclipse.org/org/documents/epl-2.0/EPL-2.0.html *
*****

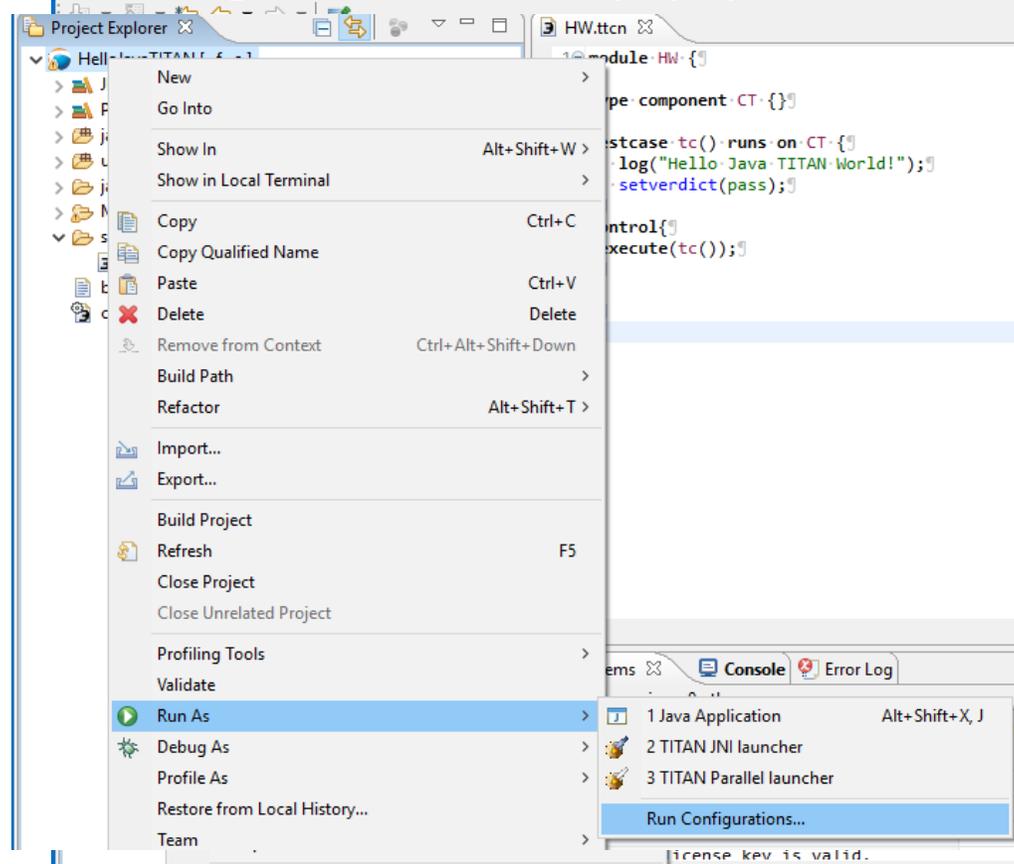
Using configuration file: cfg.cfg
MC@HU-00000670: Listening on TCP port 6999.
MC2>
```

Running the Project in Parallel Mode

Start the Host Controller



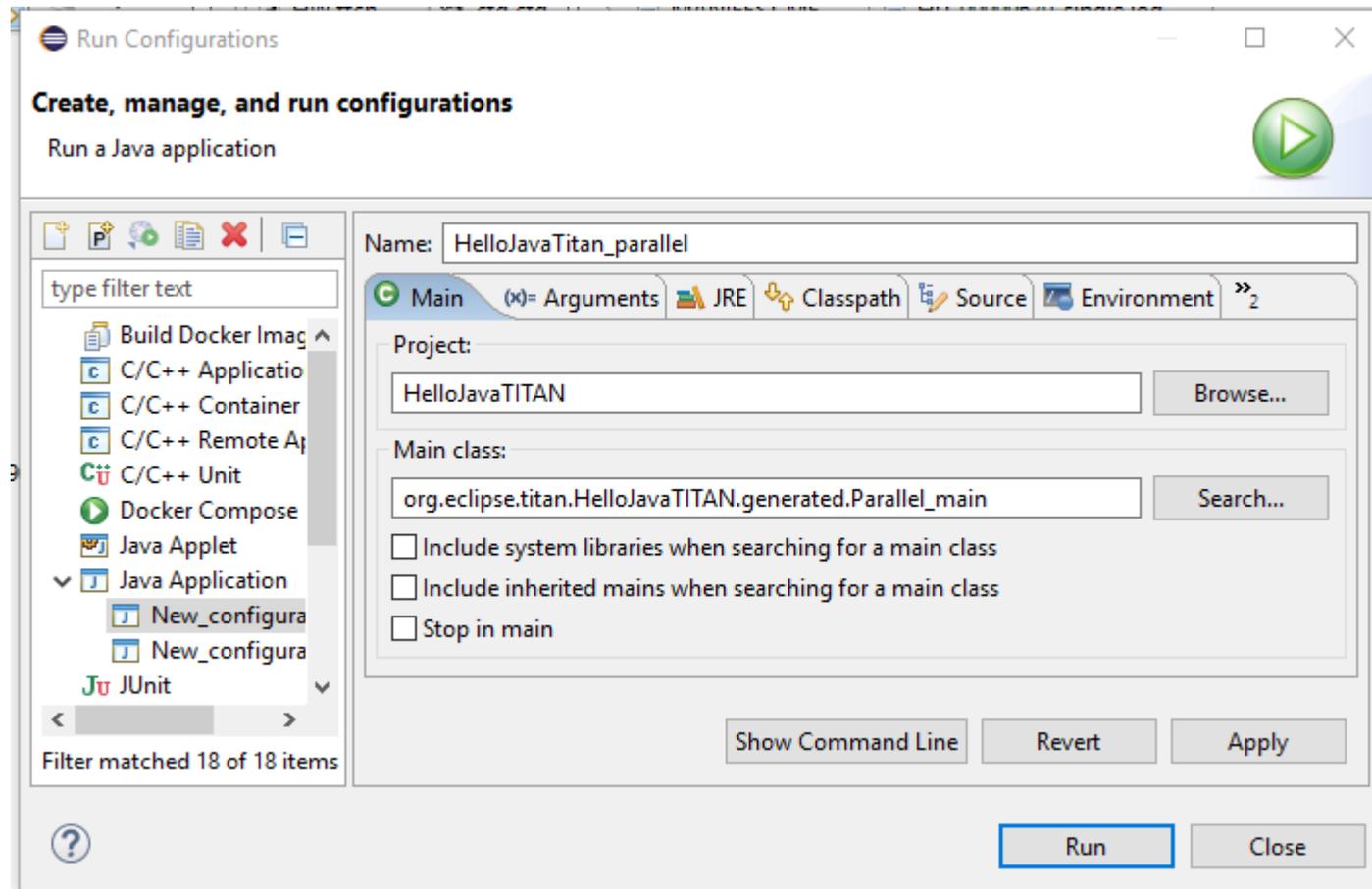
3. Select Run As> Run Configurations... in the popup menu of the project, then...



Running the Project in Parallel Mode

Start the Host Controller

4. Double click on "Java Application" in the dialog "Run Configuration" and fill the form in:

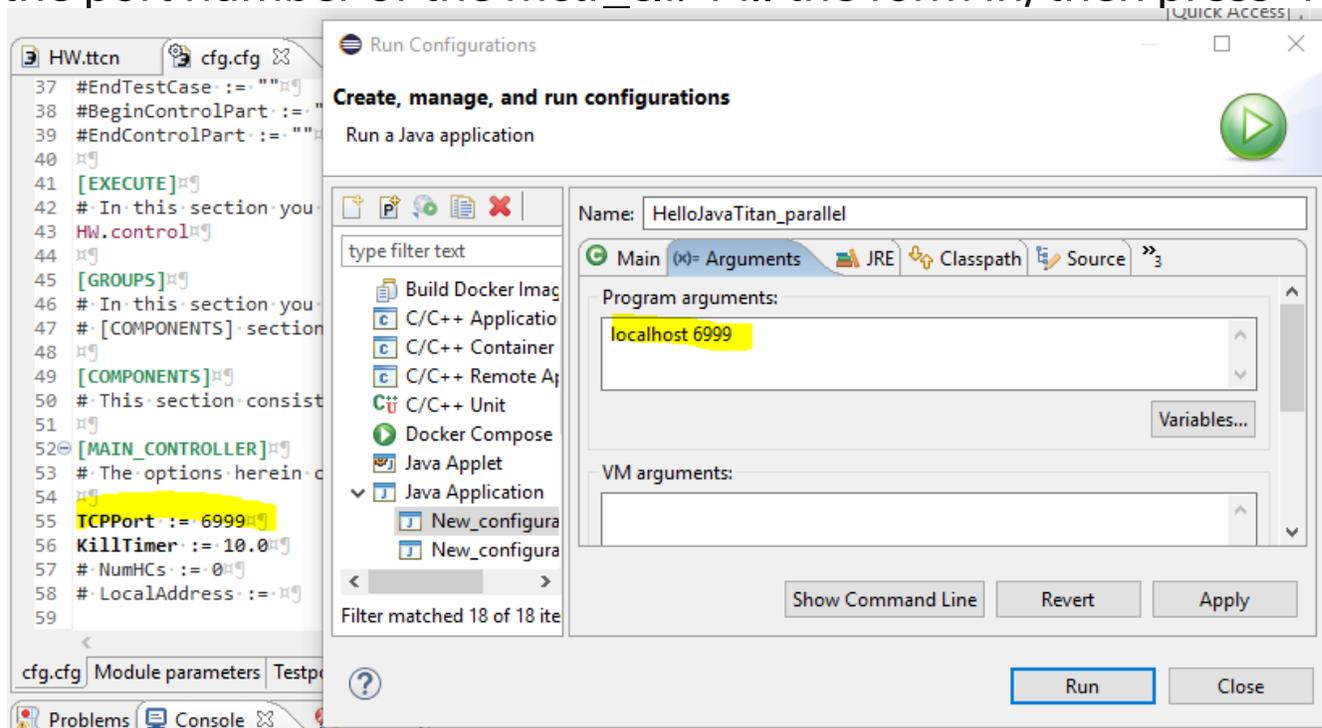


Running the Project in Parallel Mode

Start the Host Controller



4. Select the "Arguments" tab and fill the form in. In parallel mode the arguments are the arguments of the Host Controller (HC). The first argument is the (IP) address of the mctr_cli and the second arg is the port number of the mctr_cli. Fill the form in, then press "Run". The HC starts and connects to ...



Running the Project in Parallel Mode

Start the Host Controller



- The HC starts and connects to the mctr_cli. The mctr_cli reflects that.

```
~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
ethbaat@HU-00000670 ~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
$ cd ~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
ethbaat@HU-00000670 ~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
$ mctr_cli cfg.cfg
*****
* TTCN-3 Test Executor - Main Controller 2 *
* Version: CRL 113 200/6 R6A *
* Copyright (c) 2000-2019 Ericsson Telecom AB *
* All rights reserved. This program and the accompanying materials *
* are made available under the terms of the Eclipse Public License v2.0 *
* which accompanies this distribution, and is available at *
* https://www.eclipse.org/org/documents/epl-2.0/EPL-2.0.html *
*****
Using configuration file: cfg.cfg
MC@HU-00000670: Listening on TCP port 6999.
MC2> MC@HU-00000670: New HC connected from HU-00000670.ericsson.se [127.0.0.1]. HU-00000670: windows 10 10.0 on amd64.
```

Running the Project in Parallel Mode

Create MTC and start MTC. Logs



- In the terminal, type the command “cmtc” (create main test component) and “smtc” (start mtc).
- TITAN will log into the console and into the log file. The console log:

```
~/workspaces/runtime-2018-09_demo/HelloJavaTITAN
* TTCN-3 Test Executor - Main Controller 2
* Version: CRL 113 200/6 R6A
* Copyright (c) 2000-2019 Ericsson Telecom AB
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v2.0
* which accompanies this distribution, and is available at
* https://www.eclipse.org/org/documents/epl-2.0/EPL-2.0.html
*****
Using configuration file: cfg.cfg
MC@HU-00000670: Listening on TCP port 6999.
MC2> MC@HU-00000670: New HC connected from HU-00000670.ericsson.se [127.0.0.1]. HU-00000670: Windows 10 10.0 on amd64.
MC2> cmtc
MC@HU-00000670: Downloading configuration file to all HCs.
HC@HU-00000670: Warning: The address of MC was set to a local IP address. This may cause incorrect behavior if a HC from a
e host also connects to MC.
MC@HU-00000670: Configuration file was processed on all HCs.
MC@HU-00000670: Creating MTC on host HU-00000670.ericsson.se.
MC@HU-00000670: MTC is created.
MC2> smtc
Executing all items of [EXECUTE] section.
MC2> MTC@HU-00000670: Execution of control part in module HW started.
MTC@HU-00000670: Test case tc started.
MTC@HU-00000670: Hello Java TITAN World!
MTC@HU-00000670: Test case tc finished. Verdict: pass
MTC@HU-00000670: Execution of control part in module HW finished.
MC@HU-00000670: Test execution finished.
Execution of [EXECUTE] section finished.
emtc
MC@HU-00000670: Terminating MTC.
MTC@HU-00000670: Verdict Statistics: 0 none (0 %), 1 pass (100 %), 0 inconc (0 %), 0 fail (0 %), 0 error (0 %)
MTC@HU-00000670: Test execution summary: 1 test case was executed. Overall verdict: pass
MC@HU-00000670: MTC terminated.
MC2>
```

File Log



— The file log:

The screenshot shows the Eclipse IDE interface. The main editor displays a log file named 'HU-00000670-mtc.log'. The log contains the following entries:

```
6 2019/May/15 09:28:20.839000 PARALLEL -- Initializing variables, timers and ports of compone ^
7 2019/May/15 09:28:20.845000 PARALLEL -- Component type HW.CT was initialized.
8 2019/May/15 09:28:20.847000 USER - Hello Java TITAN World!
9 2019/May/15 09:28:20.847000 VERDICTOP - setverdict(pass): none -> pass
10 2019/May/15 09:28:20.851000 PARALLEL -- Terminating component type HW.CT.
11 2019/May/15 09:28:20.853000 PARALLEL -- Component type HW.CT was shut down inside testcase.
12 2019/May/15 09:28:20.853000 EXECUTOR - Waiting for PTCs to finish.
13 2019/May/15 09:28:20.855000 VERDICTOP - Setting final verdict of the test case.
14 2019/May/15 09:28:20.858000 VERDICTOP - Local verdict of MTC: pass
15 2019/May/15 09:28:20.859000 VERDICTOP - No PTCs were created.
16 2019/May/15 09:28:20.859000 TESTCASE -- Test case tc finished. Verdict: pass
17 2019/May/15 09:28:20.860000 STATISTICS -- Execution of control part in module HW finished.
18 2019/May/15 09:28:26.226000 STATISTICS -- Verdict Statistics: 0 none (0%), 1 pass (100%),
```

The IDE also shows a 'Problems' window at the bottom, which is currently empty. The status bar at the bottom indicates 'Writable', 'Insert', '13:43', and '513M of 1024M'.



<https://projects.eclipse.org/projects/tools.titan>