# WORKING ON TAFJ

Amendment History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Status | Description |
| 0.1 | 07-Jul-16 | Ilamaran N | Initial Draft |  |
| 0.2 | 07-Jul-16 | Muthamil M | Reviewed |  |

Table of Contents

[1.TAFJ: 1](#_Toc455738475)

[2. Difference between TAFC and TAFJ: 1](#_Toc455738476)

[3. Files required for using TAFJ in Eclipse: 2](#_Toc455738477)

[4. Starting with TAFJ: 2](#_Toc455738478)

[5. Creating a sample project: 10](#_Toc455738479)

[6. To compile a routine: 11](#_Toc455738480)

[7. Debug a routine in Eclipse: 12](#_Toc455738481)

[8. Taking JAR and moving JAR: 14](#_Toc455738482)

[9. Creation of Template using Eclipse: 19](#_Toc455738483)

[10. Similarly for Template.Fields: 23](#_Toc455738485)

[11. Using TAFJ in Classic: 24](#_Toc455738486)

## 

## 1. TAFJ

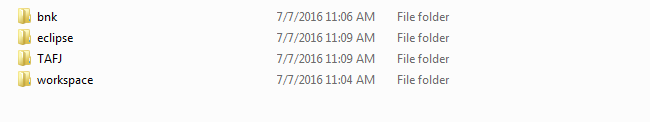
TAFJ (Temenos Application Framework Java) is a BASIC runtime and compiler, written in 100% java. It allows compiling and running BASIC programs. TAFJ has been written specifically for T24 needs, and all tests and proof of concepts have been done using T24. TAFJ also manage the connectivity on JDBC compliant databases like jBASE J4, Oracle, MS SQL.

## 2. Difference between TAFC and TAFJ

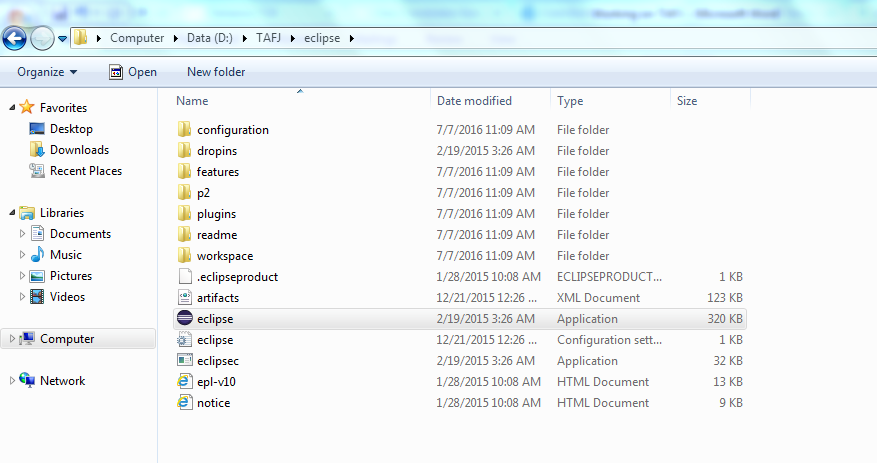
**T**emenos **A**pplication **F**ramework **C**-version is so called since R09 is nothing else but "classic" jBase with JDC interface

**T**emenos **A**pplication **F**ramework **J**ava is a complete new development and runtime platform replacement using java

## 3. Files required for using TAFJ in Eclipse

****

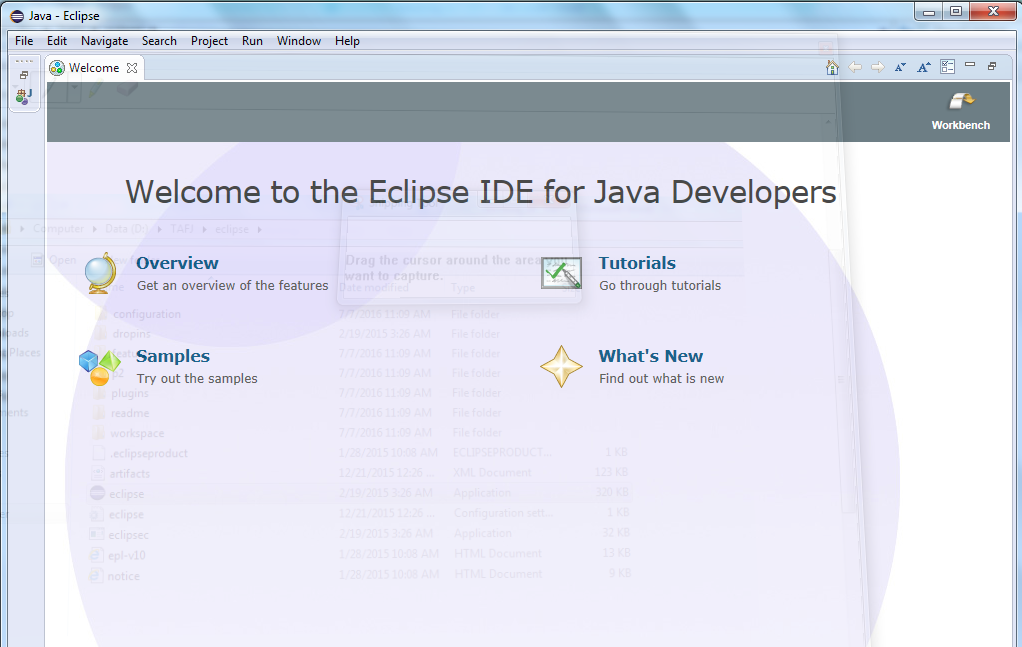
The above folders have to be added to the desired path where we are going to run the eclipse.



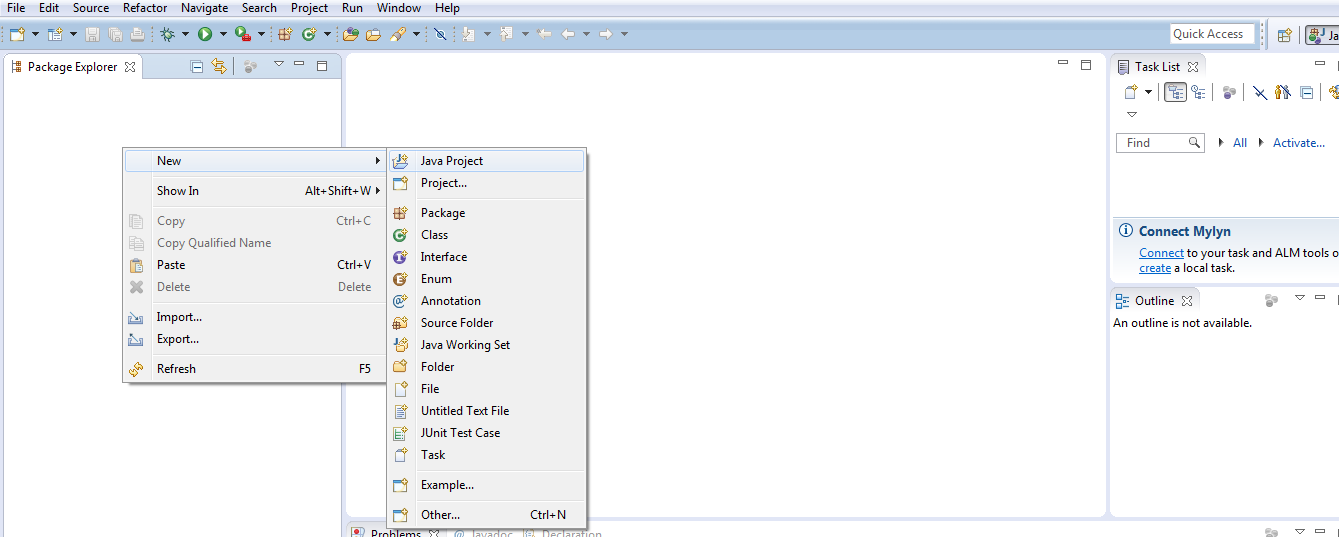
After moving the files to the selected folder, the eclipse icon present inside the Eclipse folder has to be selected as shown above.

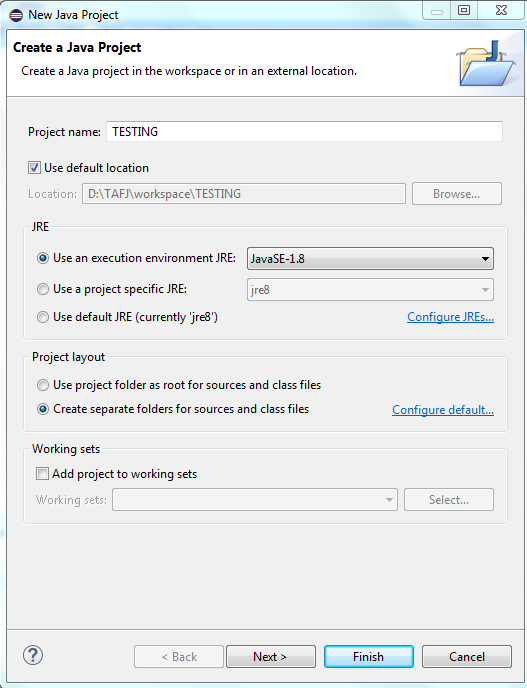
## 4. Starting with TAFJ

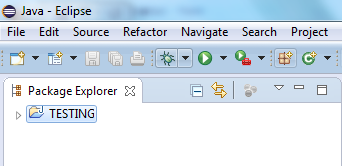
After selecting the given icon, the eclipse window will get opened as shown below.

****

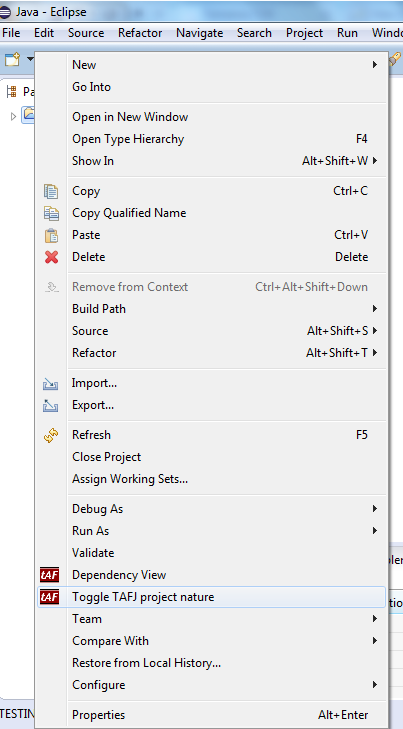
Select File -> New -> Java Project which is present at the tool bar on the top

****

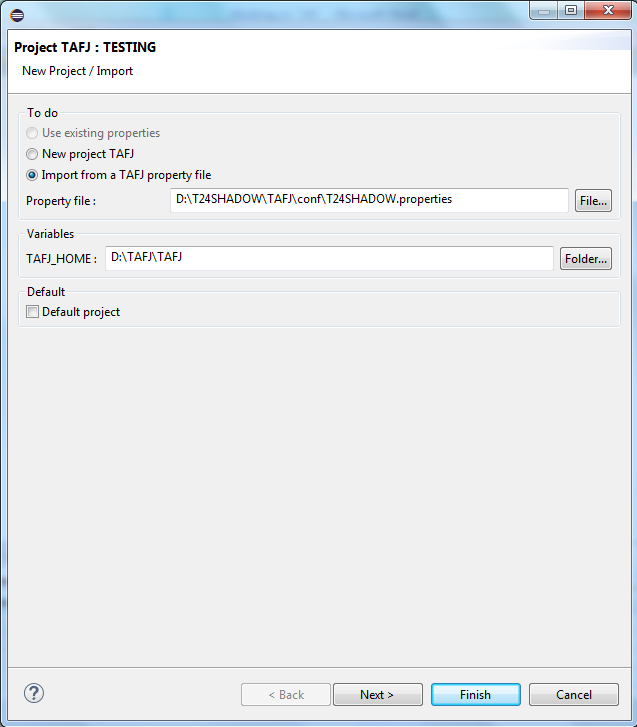
****

****

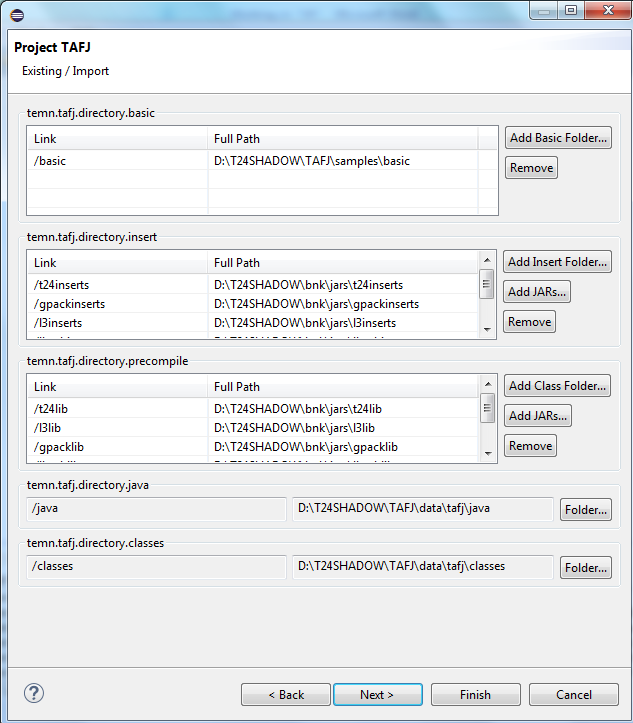
Right click on the project name and select “Toggle TAFJ Project Nature” as shown below

****

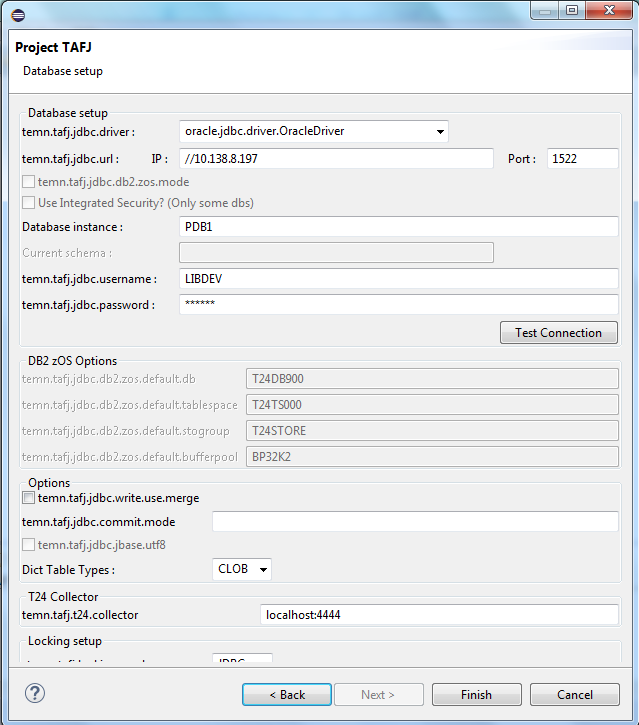
In TAFJ\_HOME give the required path of TAFJ present and in property file give the location of T24.properties as shown below and click the **Next** button.

****

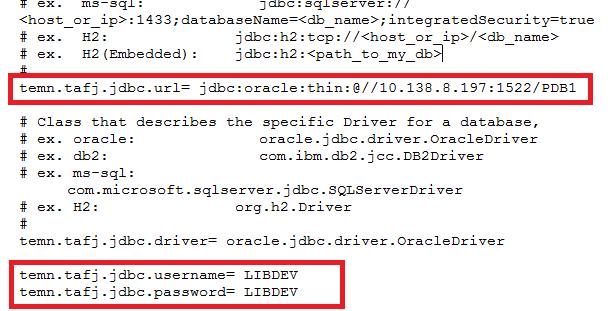
The class files necessary for running the eclipse will be loaded as shown below once the location of .property file is given. Then click **Next** button.

****

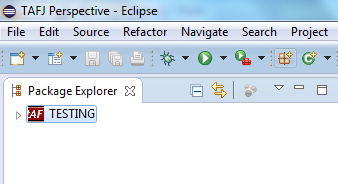
The next step is the database connectivity. In IP address give the required value followed by the Port number by checking the t24.properties file. In username and password give the required details which will be used to connect the Classic mode. Finally click the **Finish** button.

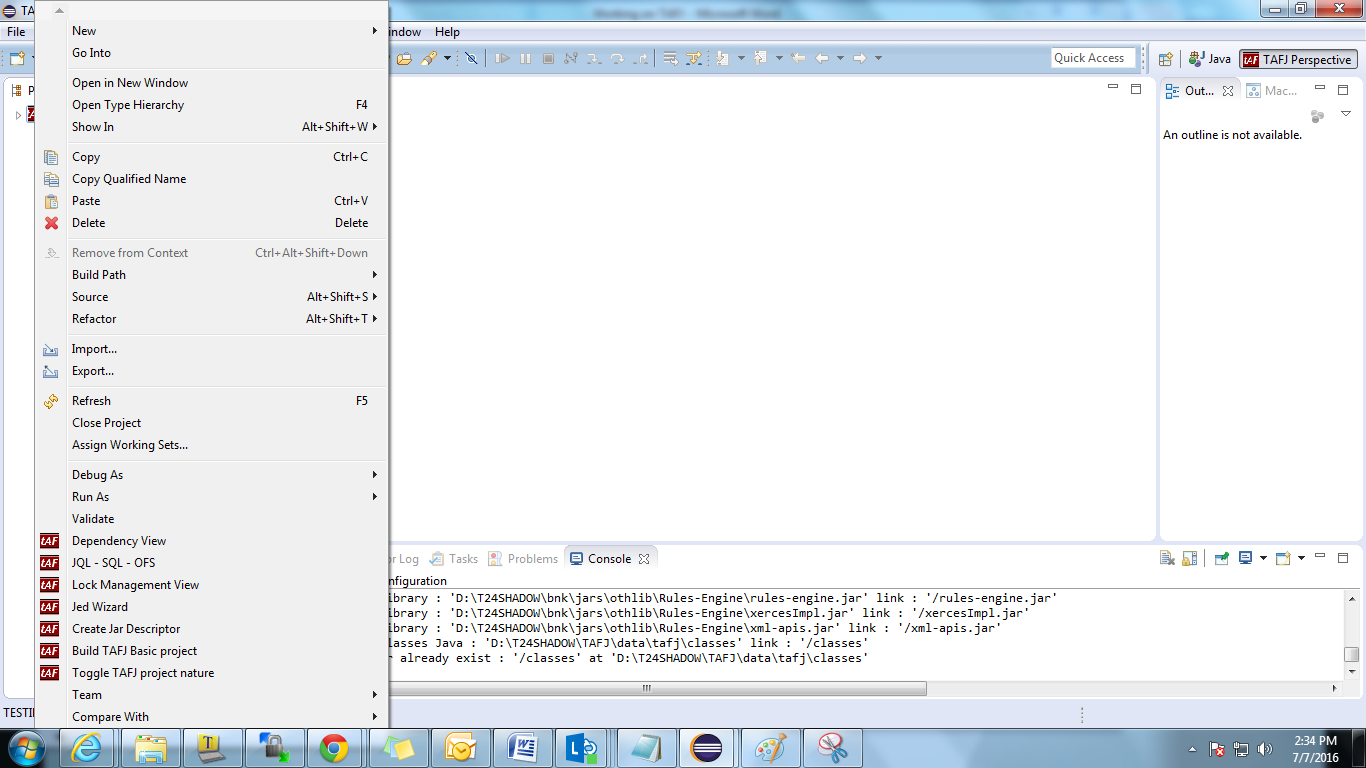


These changes are to be made in **tafj.properties** which will be available in the path D:\T24SHADOW\TAFJ\conf



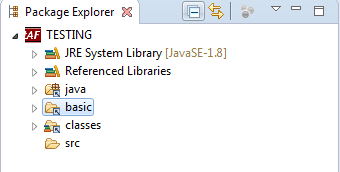
After clicking the Finish button, check for the icon of project name and it should be changed as shown below which indicates that the process is finished without errors. Also right click the project name and check whether all the functionalities are available as shown below.



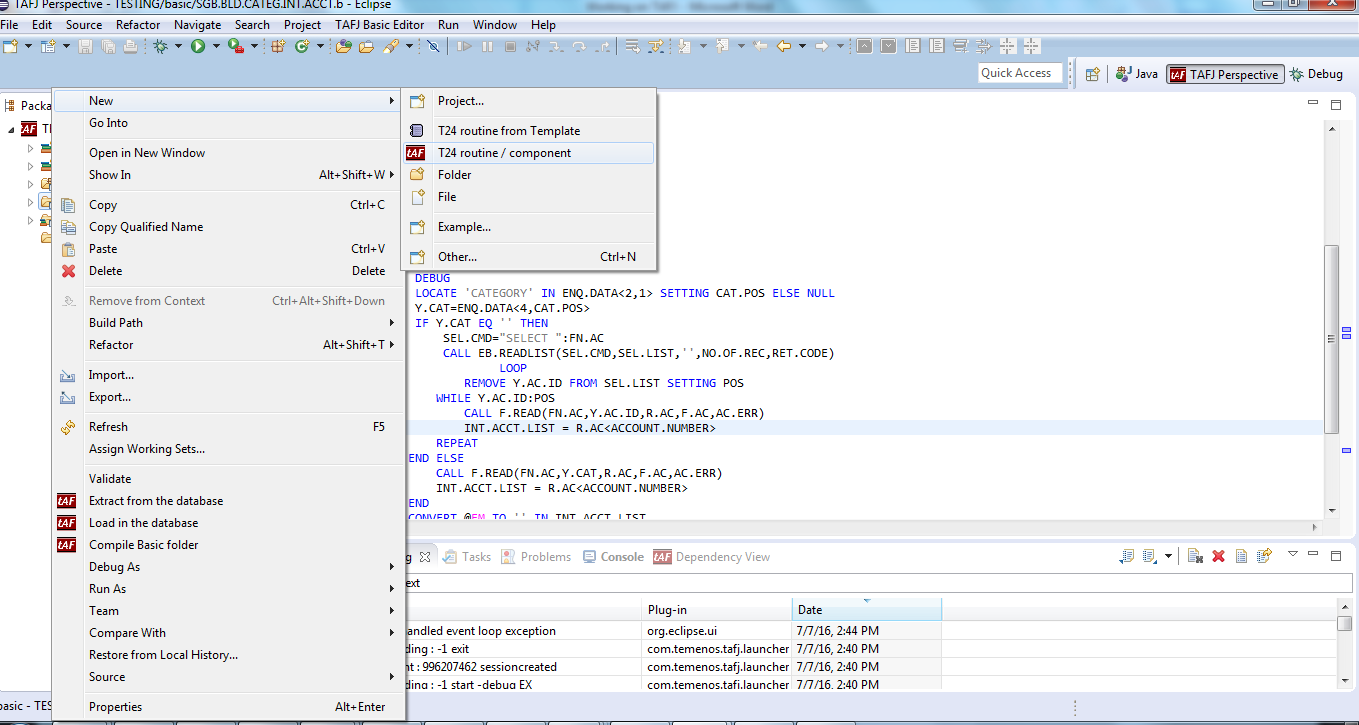


## 5. Creating a sample project

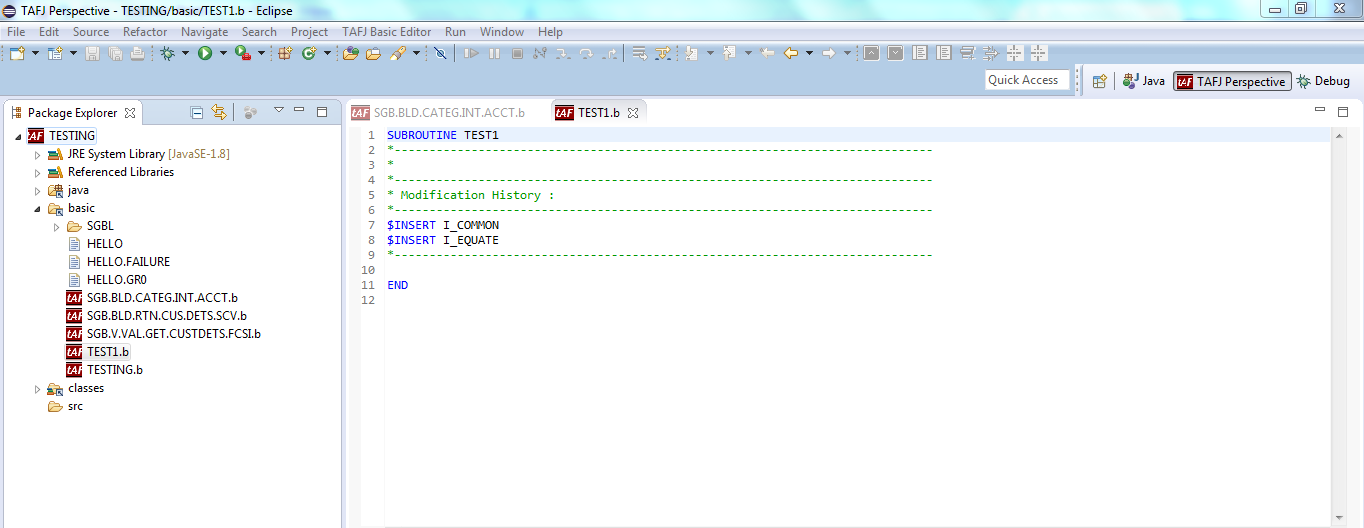
We can create both routine and template here. First select the **basic** button available under the project name Testing as shown below.

****

Right click on the Basic button and select the required information needed where both creation of Template and Routine will be available as shown below

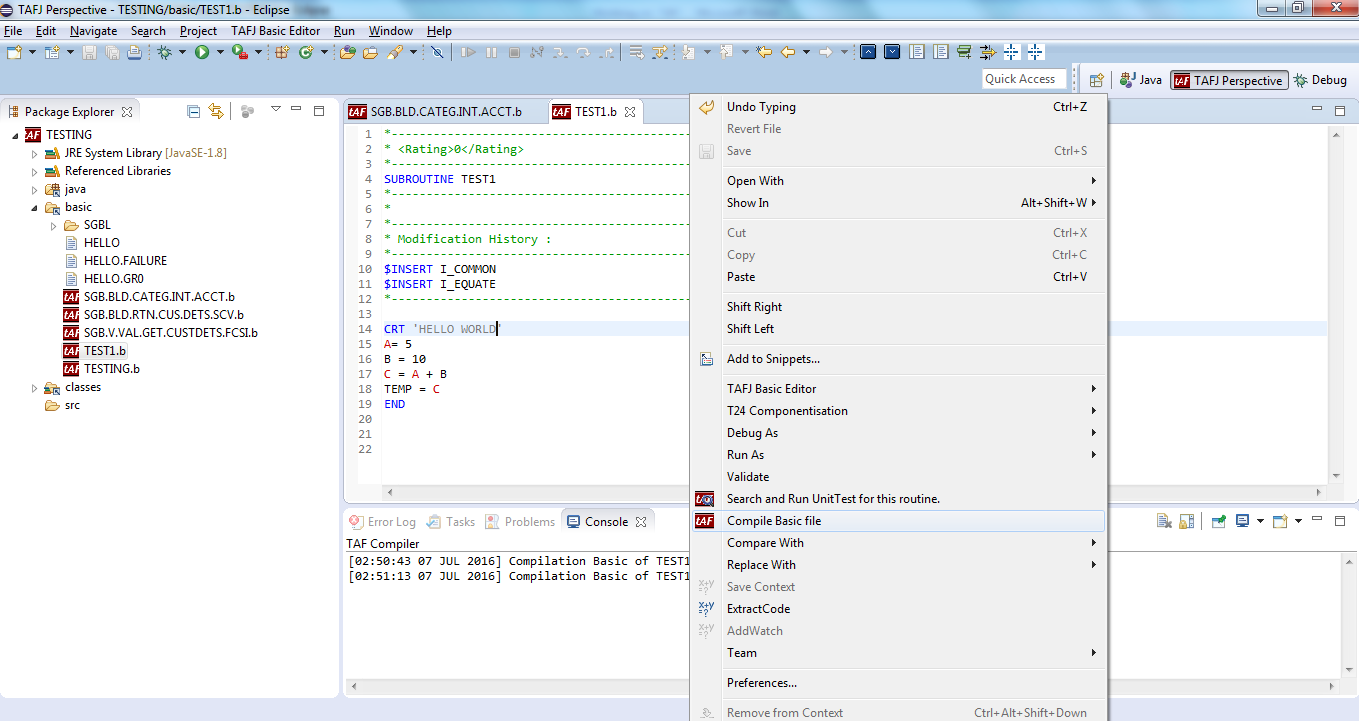
****

Once the **T24 routine/Component** is selected, the template that contains the required insert file will be shown where we can modify the needed routine

****

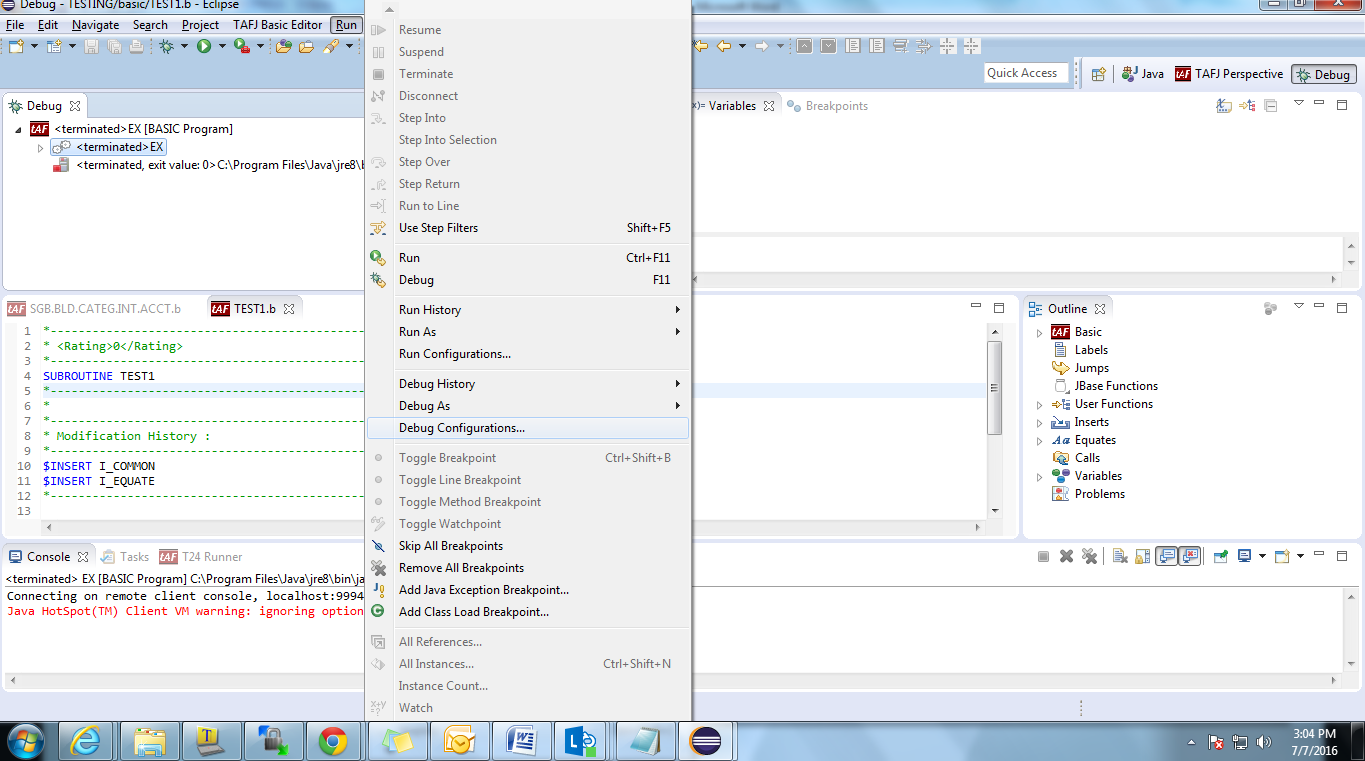
## 6. To compile a routine

To compile a routine we can simply press CTRL + S or right click on the workspace where an option as **“Compile Basic File"** is listed.

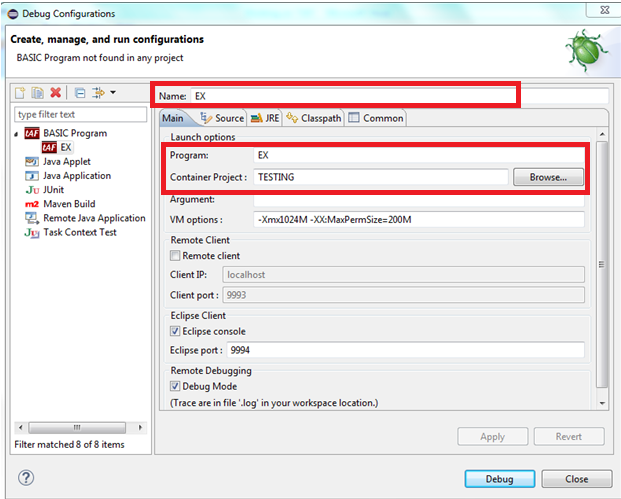
****

## 7. Debug a routine in Eclipse

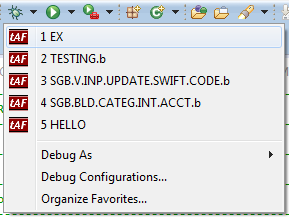
We can also use the option of Debug in eclipse itself.



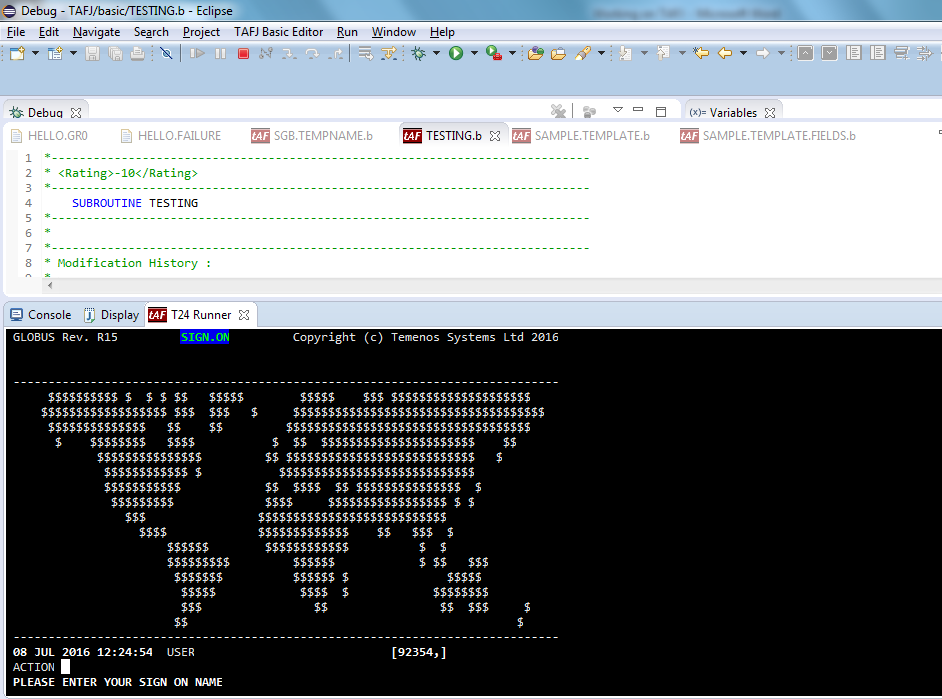
In the debug configuration, give the following details where in name it can be any name and in Program give ‘EX’ which acts as the connectivity between classic and eclipse. In Container Project give the name of the Container where the routine is present. Check the button in **Eclipse console** which helps to display the putty on the bottom of the eclipse screen. After giving all the details click the Apply button.

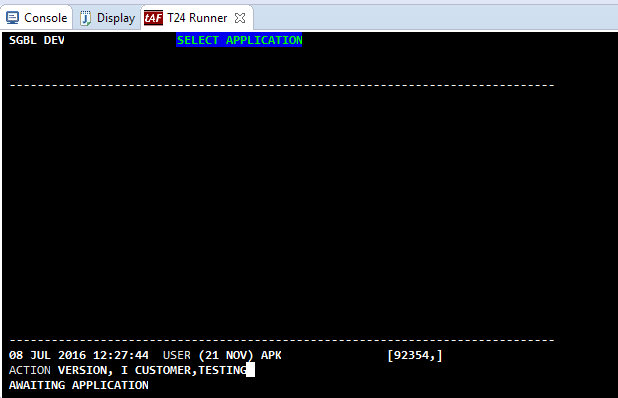


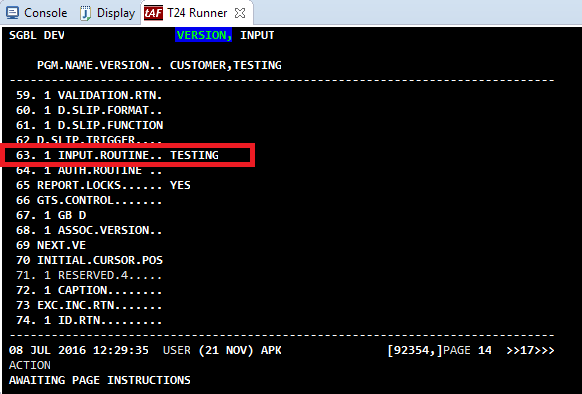
After giving the required details, give **Apply** and press the **Close** button. Then move on to the Tool bar and select the **Insect button** which means debug and select **EX** so that our Debug starts.



The Classic window will be shown in eclipse as shown below and thus enter into the desired application and start the Debug as we do in TAFC. Attach the routine done in Eclipse to any of version so that debug works perfectly. We have to mark Breakpoint in routines before starting the debug. It is similar to typing debug in the routine.





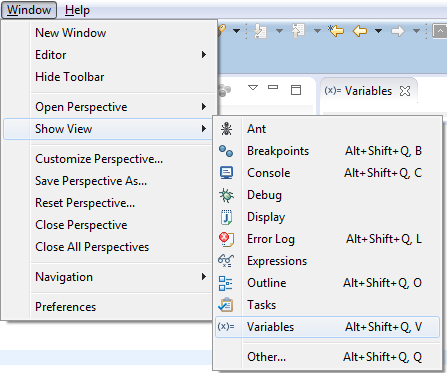




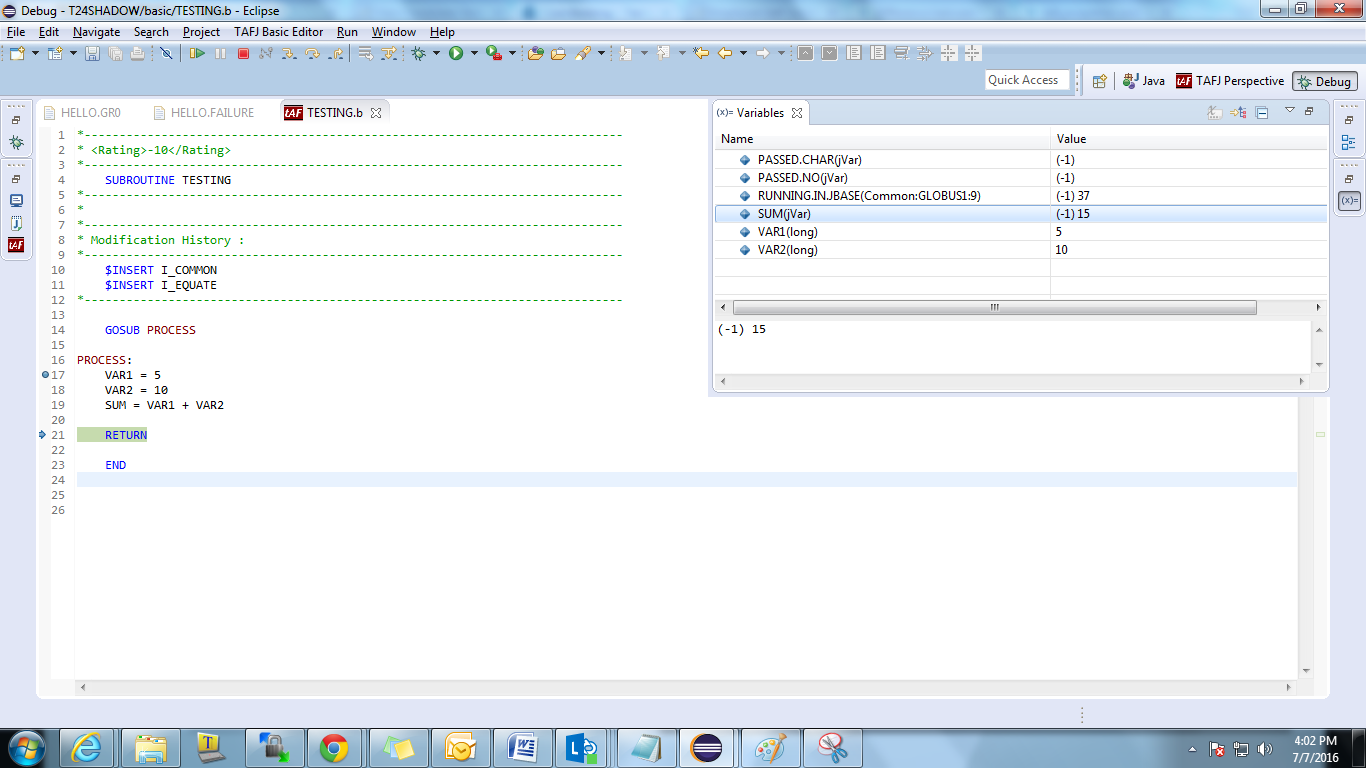




Eclipse deals with an option called **Variables** that helps to check whether our routine works correctly and passing through all the arguments.



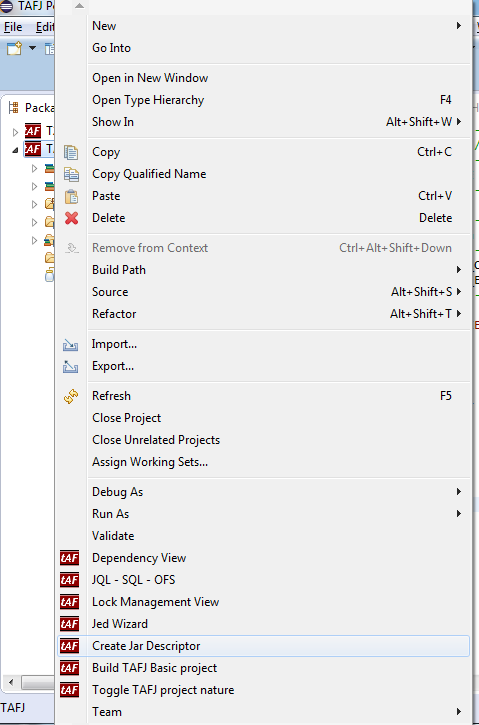
After reaching to the debug, we can navigate to the next line using **F6** andthe value will be available in the Variables tab as shown below



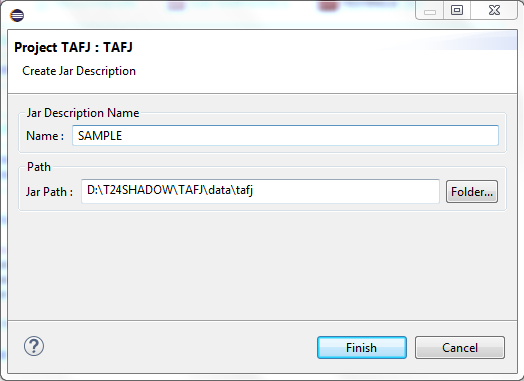
## 8. Taking JAR and moving JAR

JAR(Java Archive) is the package of class files which has to be extracted from the workspace after obtaining the desired output. This JAR file again has to be moved to a desired path so that the routine done by the developer work perfectly.

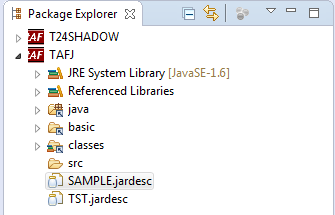
First right click on the Container project where our routine exists.



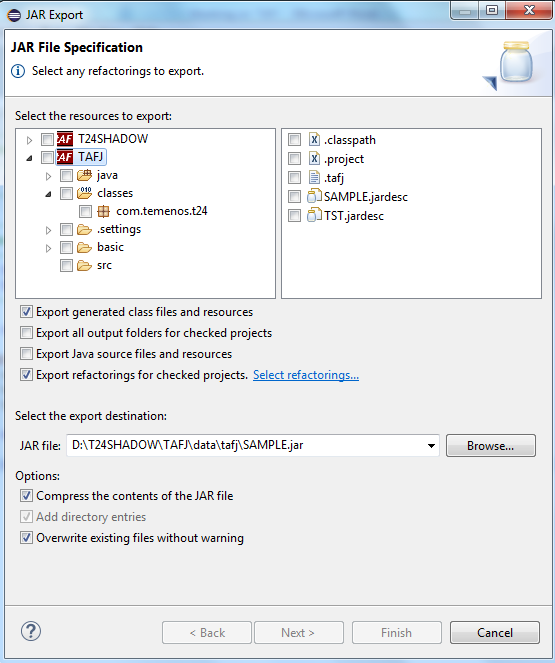
Enter the name of the JAR in **Name** column and in **Jar Path type** the path where the JAR has to be moved and click **Finish** button.



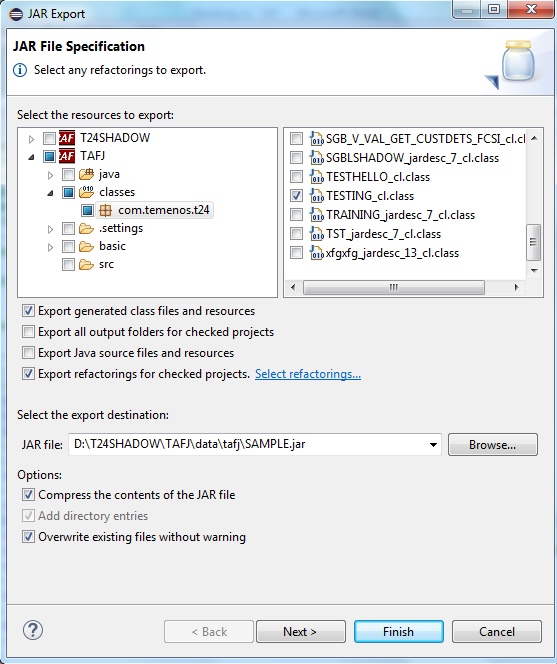
Now we can see the name of JAR created under our Container with name Sample.



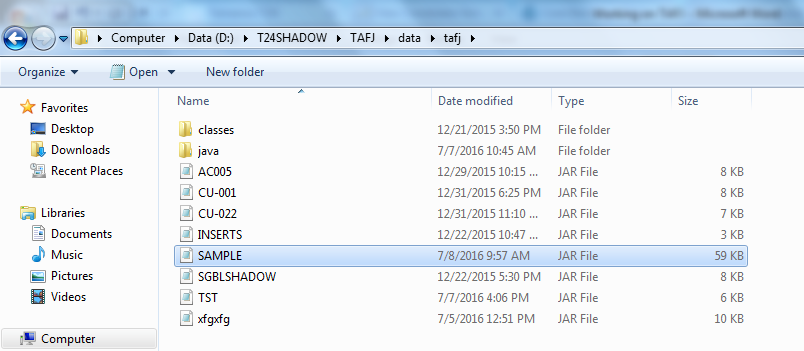
Double click the Jar file created by us and we will get a screen as shown below



Select the jar file required for us as shown below and click Finish button.

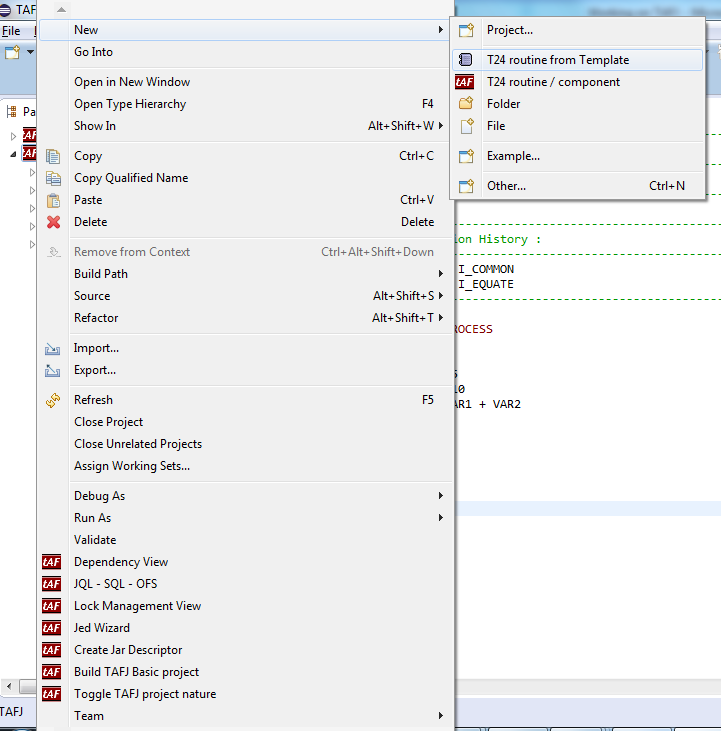


Check for the desired folder and our Jar file will be located. Move the file to the desired server path using WINSCP.

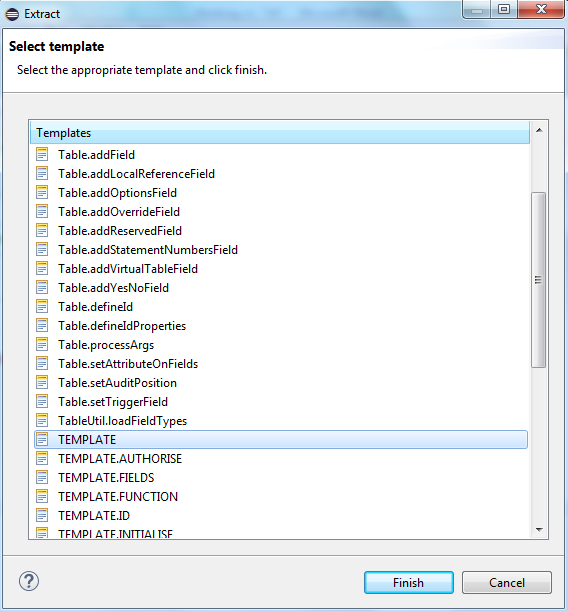


## 9. Creation of Template using Eclipse

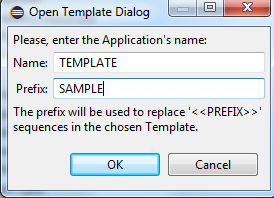
Creation of Template and Template.Fields looks very easier while creating in Eclipse.



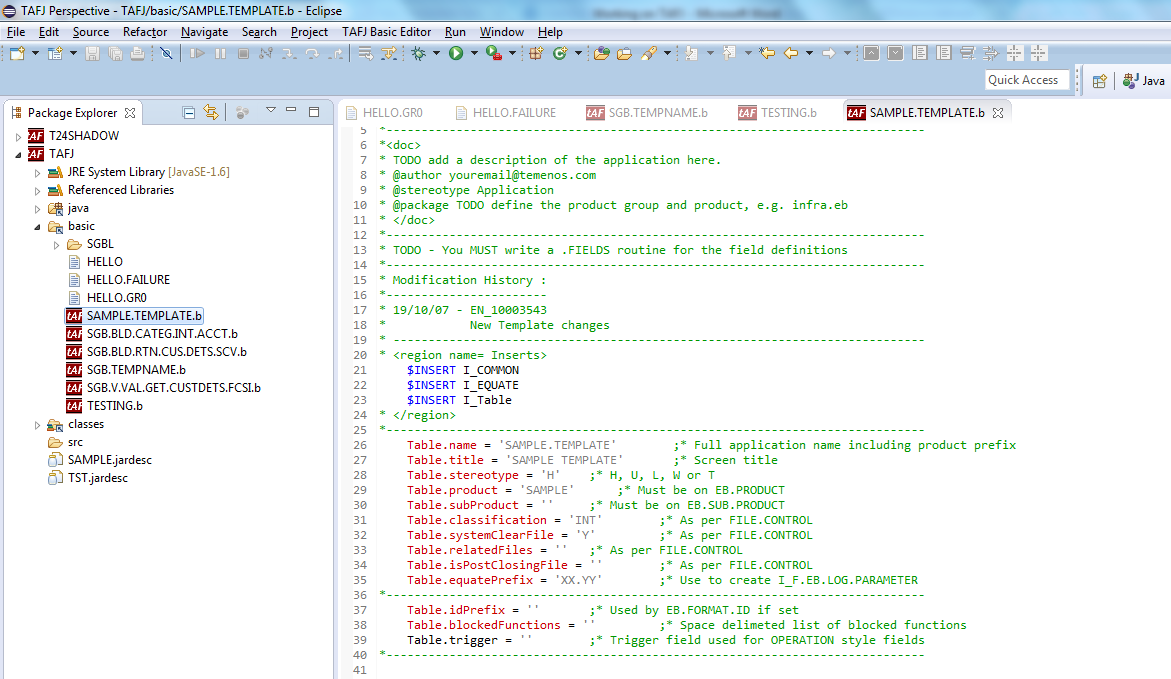
A list of Template type will be listed after selecting **T24 routine from template** where we will be selecting the required type of template. Since we need a sample template, we are selecting the type as **Template** and click **Finish** button.



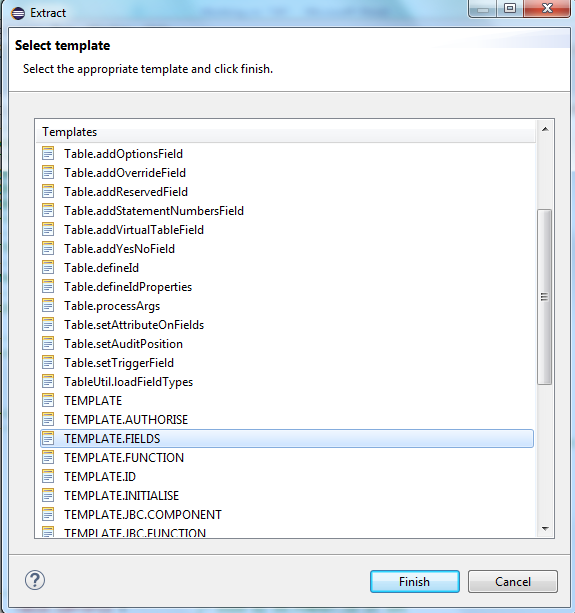
After giving Finish button, a pop up window prompts to enter the Name and Prefix. Enter the name and prefix and click OK button so that Prefix followed by Template will be our Template name.

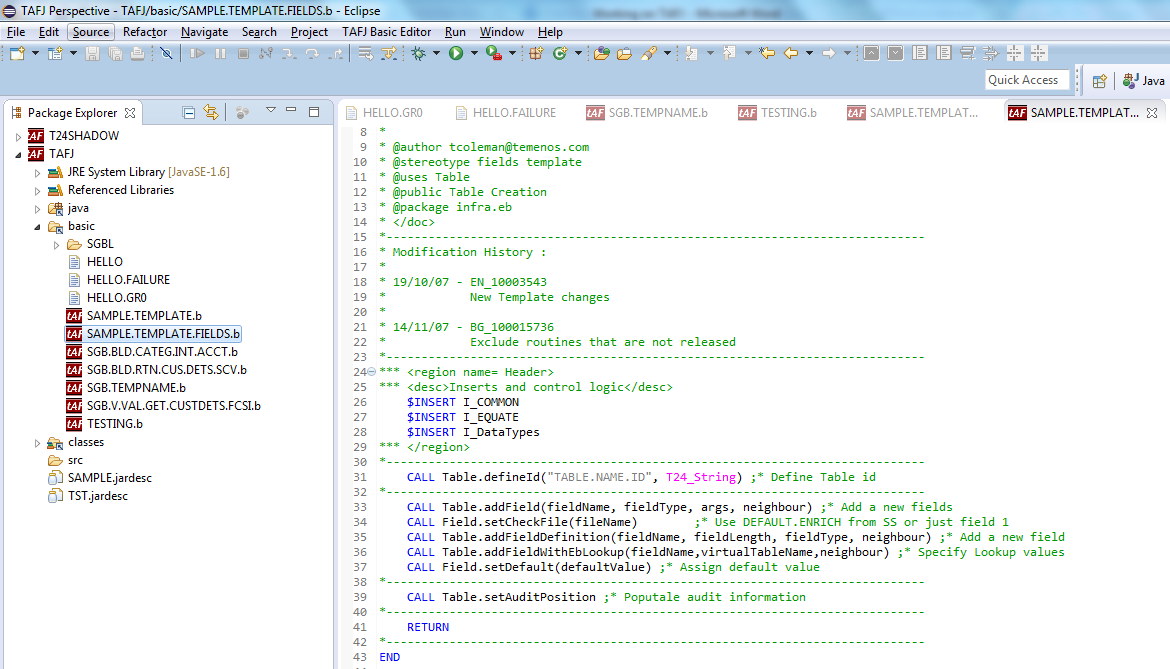


Template will be created as shown below



## 10. Similarly for Template.Fields

****

****

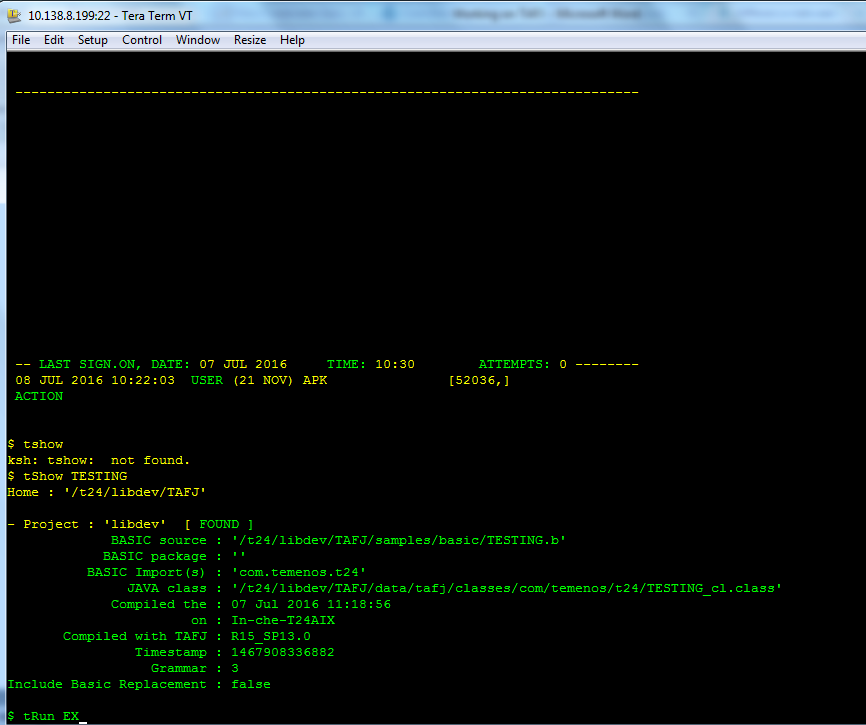
## 11. Using TAFJ in Classic

Usage of classic window for TAFJ is similar to TAFC except some of the basic keywords. Some of them are

TAFJ tRun – We will be using tRun EX to log in into Classic

TAFJ tShow – To check the availability of file and path

TAFJ tCompile – To compile a routine or template



tCompile /data/r16mbarc/TAFJ/samples/basic/RTN1.b

tShow RTN1