



TESTFRAME ENGINE

VERSION 2008.06

INSTALLATION GUIDE

TestFrame Engine Installation Guide

Copyright © 1999–2008 Logica Nederland B.V.

All rights reserved. No part of this publication may be reproduced, or stored in a retrieval system, or transmitted, in any form, or by any other means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the copyright holder.

TestFrame is a trademark of Logica.

This document also contains registered trademarks and trademarks which are owned by their respective companies or organizations. Logica disclaims any responsibility for specifying which marks are owned by which companies or organizations.

While every precaution has been taken in the preparation of this document, Logica assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

If you have any comments or suggestions regarding this document, please send them via e-mail to ccctesting@logica.com

For more information on Logica and TestFrame visit:

<http://www.logica.com>

<http://www.testframe.com>

For more information on the TestFrame Engine write to:

ccctesting@logica.com

Central Competence Centre Testing

Telephone: 020-5033000

Fax: 020-5033011



TABLE OF CONTENT

WELCOME TO THE TESTFRAME ENGINE	5
1 INSTALLING THE TESTFRAME ENGINE	6
2 ROOT DIRECTORY STRUCTURE.....	10
2.1 THE BIN DIRECTORY	11
2.2 THE CLUSTER DIRECTORY	12
2.3 THE DOC DIRECTORY	13
2.4 THE INCLUDE DIRECTORY	14
2.5 THE KEEP DIRECTORY	15
2.6 THE LIB DIRECTORY	16
2.7 THE LOG DIRECTORY	17
2.8 THE REPORT DIRECTORY	18
2.9 THE SCRIPTS DIRECTORY	19
2.10 THE SNAP DIRECTORY	20
2.11 THE TEMPLATE DIRECTORY.....	21
2.12 THE LICENSE FILE	21
2.13 THE INI FILE	22
3 RUNNING THE INSTALLATION EXAMPLE	23
3.1 RUNNING THE EXAMPLE USING C++	23
3.2 RUNNING THE EXAMPLE USING JAVA	24
3.3 RUNNING THE EXAMPLE USING QARUN	25
3.4 RUNNING THE EXAMPLE USING RATIONAL ROBOT	26
3.5 RUNNING THE EXAMPLE USING VISUAL BASIC.....	27
3.6 RUNNING THE EXAMPLE USING WINRUNNER	28
3.7 RESULTS OF THE INSTALLATION EXAMPLE	29

WELCOME TO THE TESTFRAME ENGINE

Welcome to the TestFrame Engine, Logica's tool for processing automated software tests designed using the TestFrame methodology. The Engine is the heart of TestFrame's automated test environments, linking together test specifications, test software, and test results.

Using this guide

This guide describes how to install the TestFrame Engine.

The first chapter describes the steps to take during installation.

The second chapter gives an overview of the directory structure created during installation, and descriptions of the files placed herein.

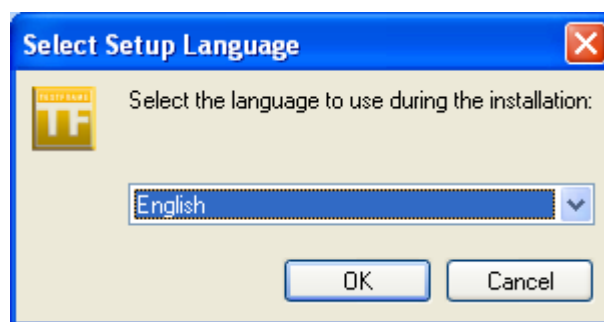
The third chapter describes how to run the installation example in each of the supported environments.

1 INSTALLING THE TESTFRAME ENGINE

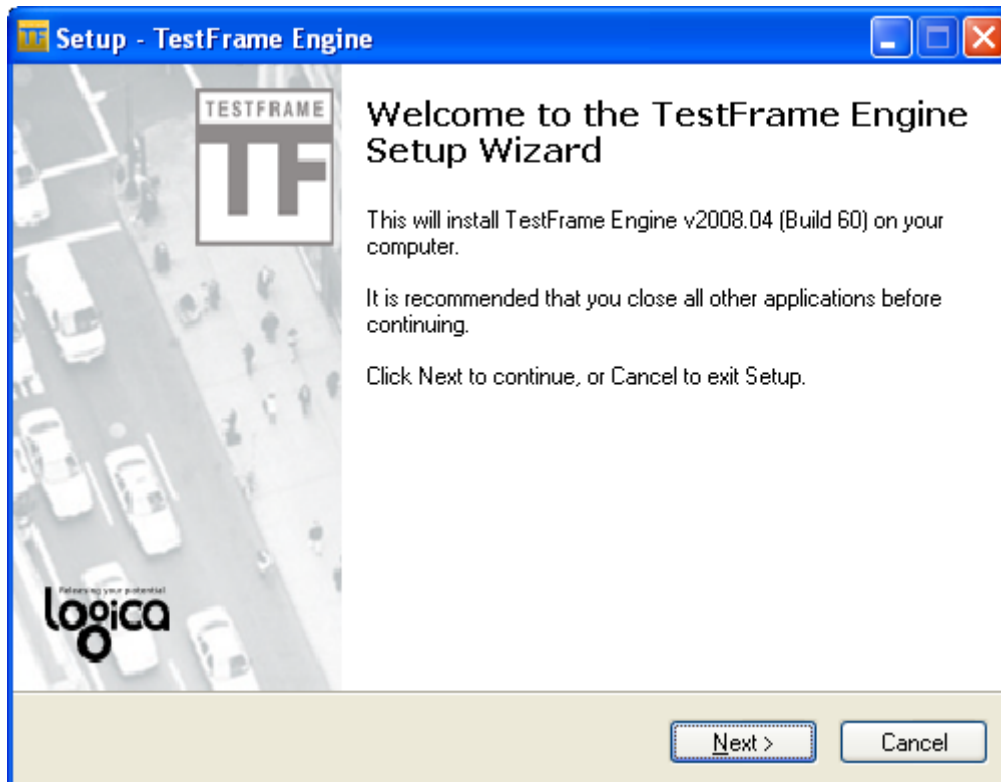
The TestFrame Engine must be installed by running the program TFEngineSetup.exe from the distributed installation files. Before starting this executable, please note the following.

1. **For installation on Windows 2000, XP and Vista systems, administrator rights are needed.**
2. **Uninstall from your machine any previously installed version of the Contest or TestFrame engine before starting the setup executable.**

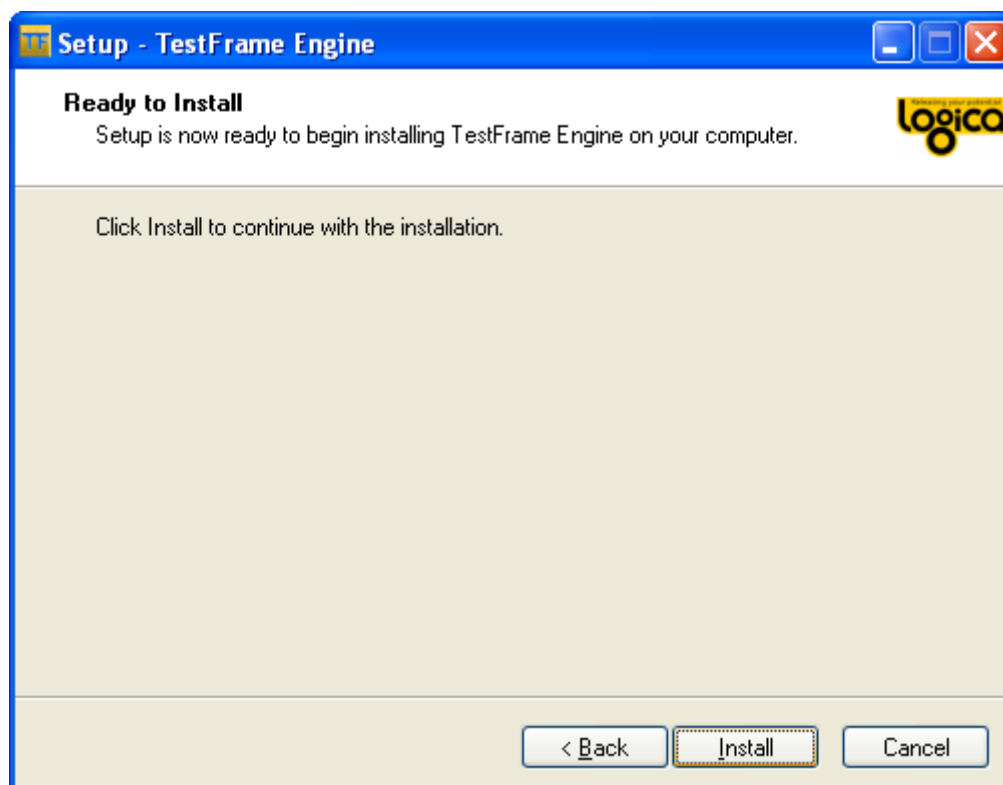
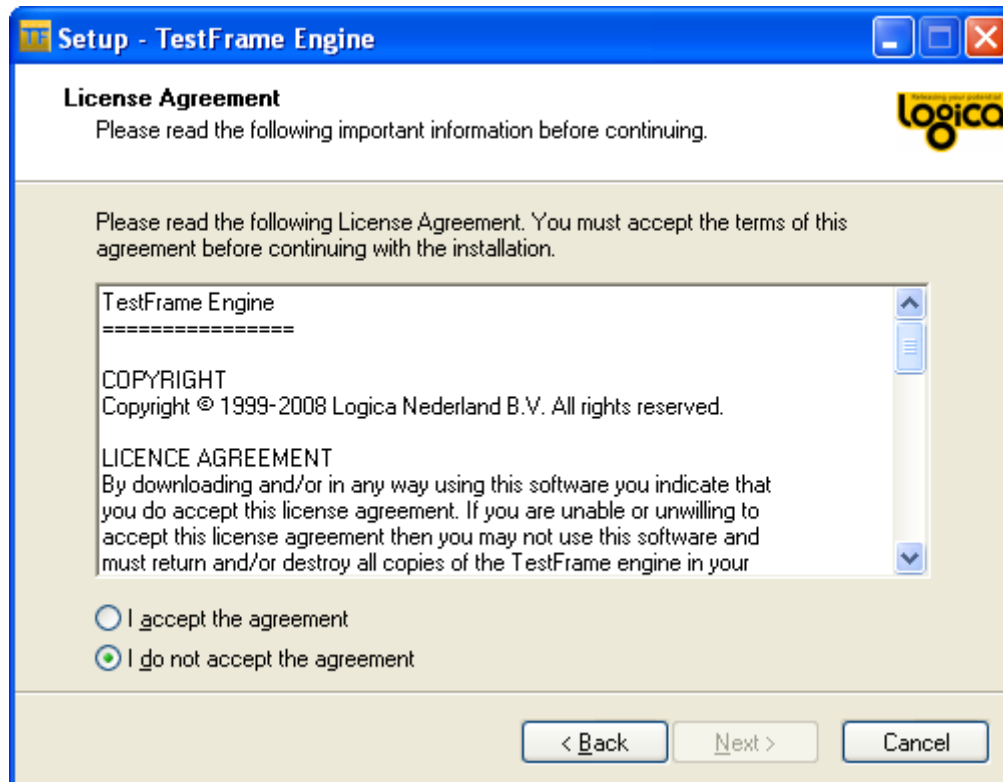
When run, TestFrame Engine v2008.04.exe starts the installation procedure:



Select the language of your choice, and then click OK. The language is used during the installation procedure.



Before you start the installation you have to accept the license agreement (this is also found after installation as EULA.rtf in the root directory where TestFrame Engine is installed):



Now you are ready to install the TestFrame Engine. TestFrame is installed in the directory **C:\TestFrame**. The newly created c:\testframe\bin directory is added to the environment PATH variable.

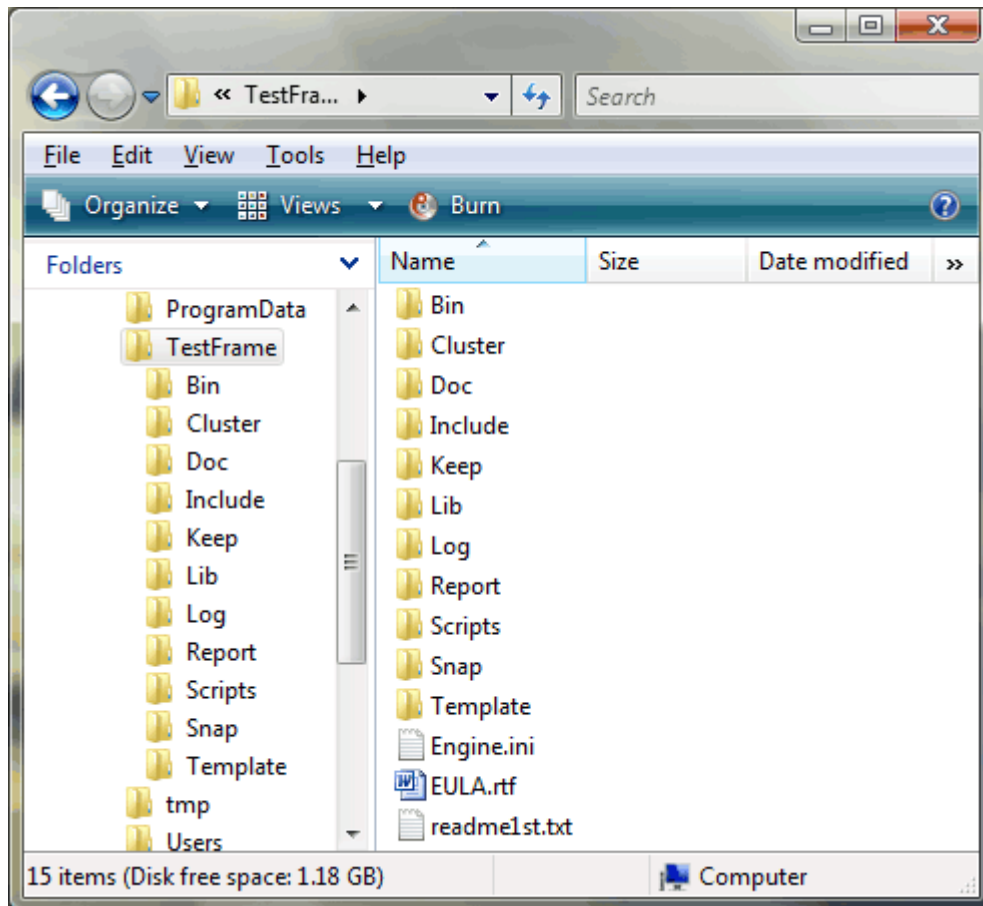
Click Finish, the TestFrame Engine is now installed. When the installation has finished you can view the readme file which contains the latest information on this release of the TestFrame Engine.

=====
Note that if you place the TestFrame Engine or any of the files described in this guide on your system without running this installation procedure, their proper functionalities are not guaranteed.
=====

2 ROOT DIRECTORY STRUCTURE

The root directory of TestFrame Engine is found at the location specified during the installation procedure. In this chapter the TestFrame directory structure will be explained. In the illustrations the root directory is **C:\TestFrame**, the actual directory can be different.

This directory structure is as follows:



In subsequent paragraphs each subdirectory is explained in more detail.

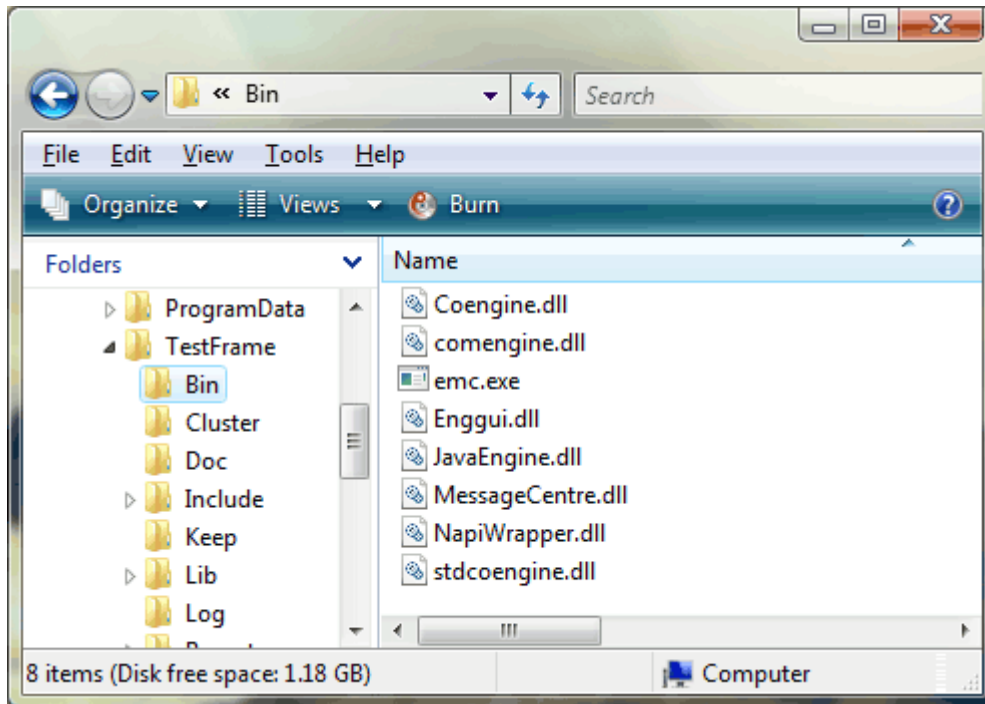
The root directory contains several subdirectories and three files:

- **engine.ini**. This file contains the settings for the TestFrame Engine.
- **EULA.txt**. The license agreement file.
- **readme1st.txt** which contains installation instructions and the latest information on this release of TestFrame Engine (changes and bug fixes).

In order to work with TestFrame Engine you will need a valid license. The file "**license.key**", has to be installed separately. See for more information paragraph 2.12.

2.1 The Bin directory

The **Bin** directory contains the *dll*'s and *executables* needed by the Engine. The content of the **Bin** directory should look like this:

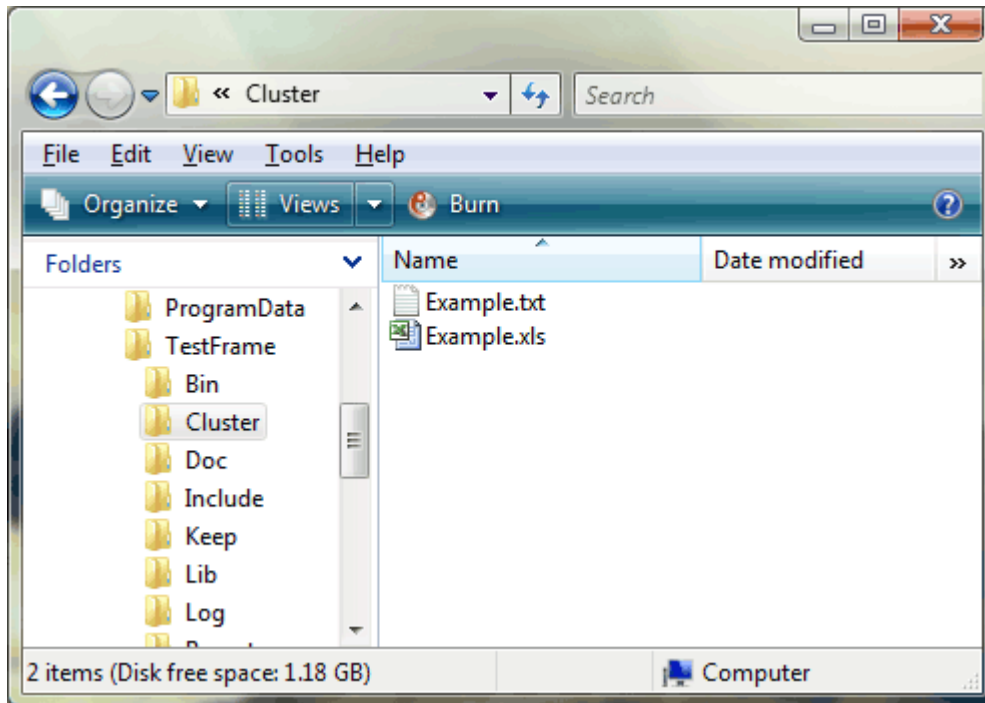


and contains the following files:

- **Coengine.dll**, which contains all the Engine functionality.
- **Comengine.dll**, COM wrapper (used for rational robot).
- **emc.exe**, the Engine Message Centre.
- **Enggui.dll**, TestFrame Engine's graphical user interface.
- **JavaEngine.dll**, Java wrapper to access the TestFrame Engine.
- **MessageCentre.dll**, contains the functionality of the Engine Message Centre.
- **NapiWrapper.dll**, provides access to NAPI components.
- **stdcoengine.dll**, provides Visual Basic program access to the TestFrame Engine.

2.2 The Cluster directory

The **Cluster** directory contains *Cluster files*. The content of the **Cluster** directory should look like this:

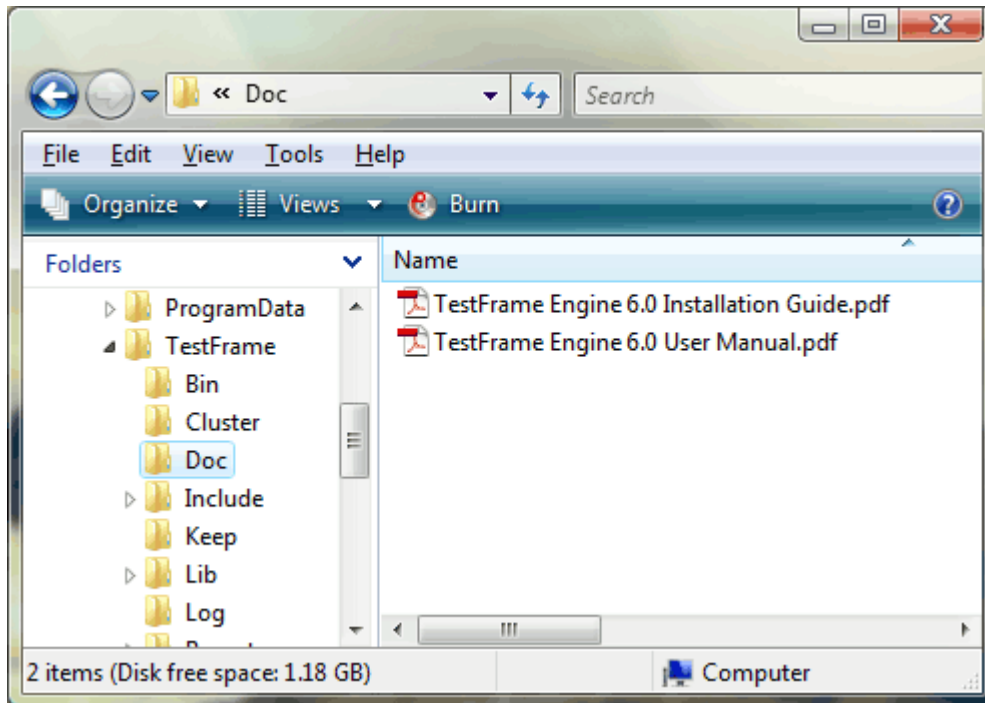


and contains the following files:

- **Example.txt**, the tab delimited cluster file needed for the examples.
- **Example.xls**, a Microsoft® Excel workbook version of this cluster.

2.3 The Doc directory

The **Doc** directory contains *Documentation files*. The content of the **Doc** directory should look like this:

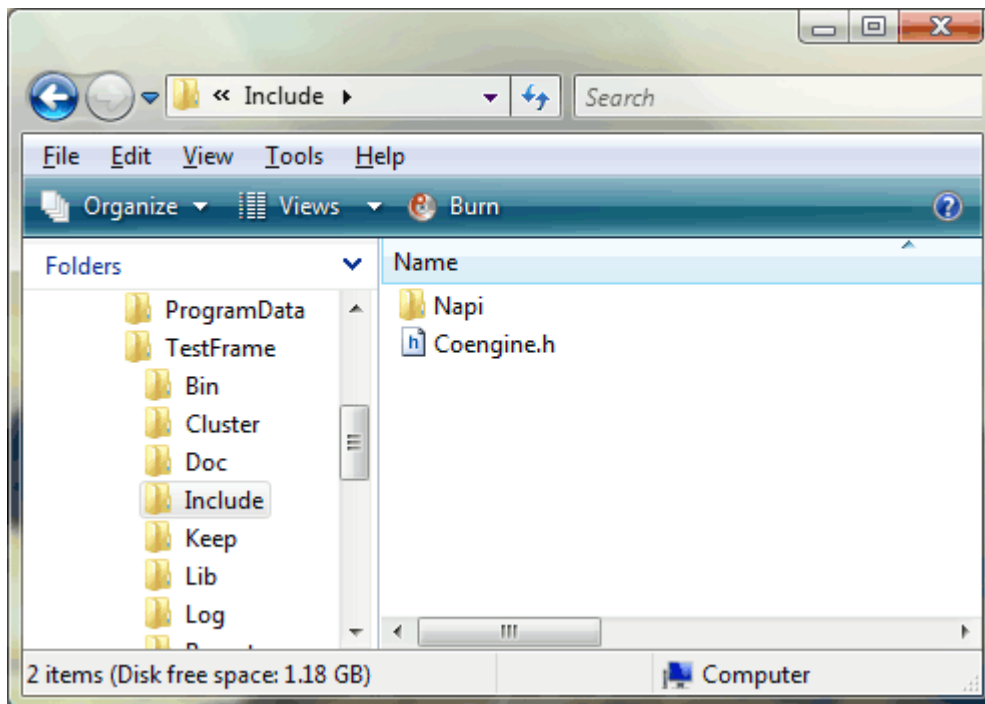


and contains the following files:

- **TestFrame Engine InstallationGuide.pdf**, a copy of this document.
- **TestFrame Engine UserManual.pdf**, the user's guide for TestFrame Engine.

2.4 The Include directory

The **Include** directory is needed for the C++ scripts. Any necessary *include files* are placed here. The content of the **Include** directory should look like this:

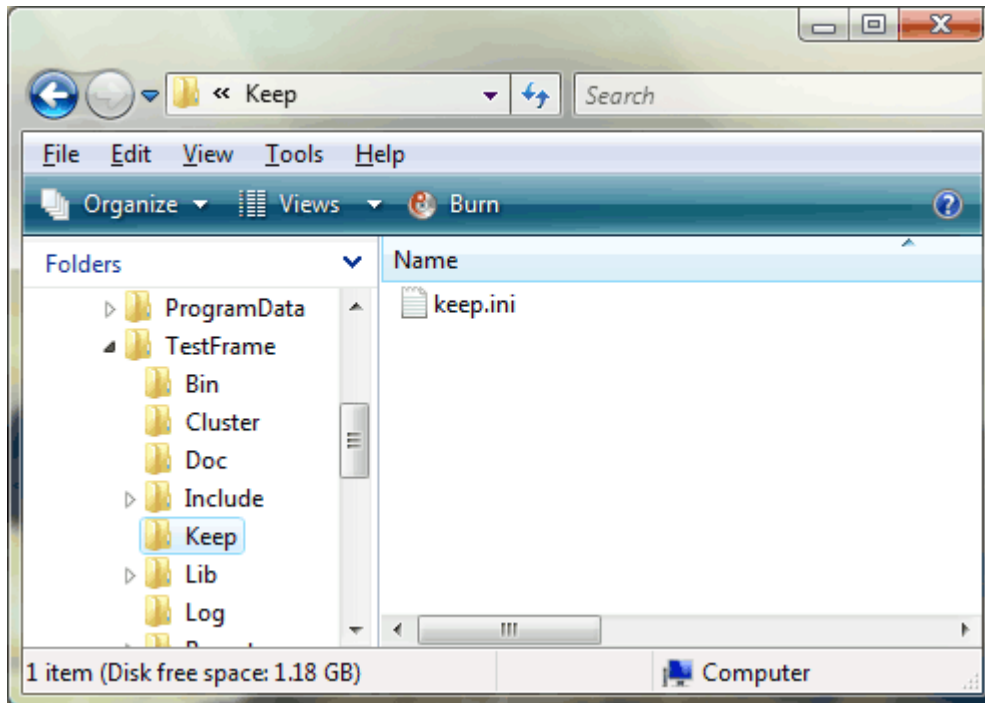


and contains the following files:

- **Coengine.h**, the header file of the Coengine.cpp source file.
- **Napi.h** and **NapiWrapper.h**, in the subdirectory **Napi**.

2.5 The Keep directory

The **Keep** directory contains *Keep files*. The content of the **Keep** directory should look like this:

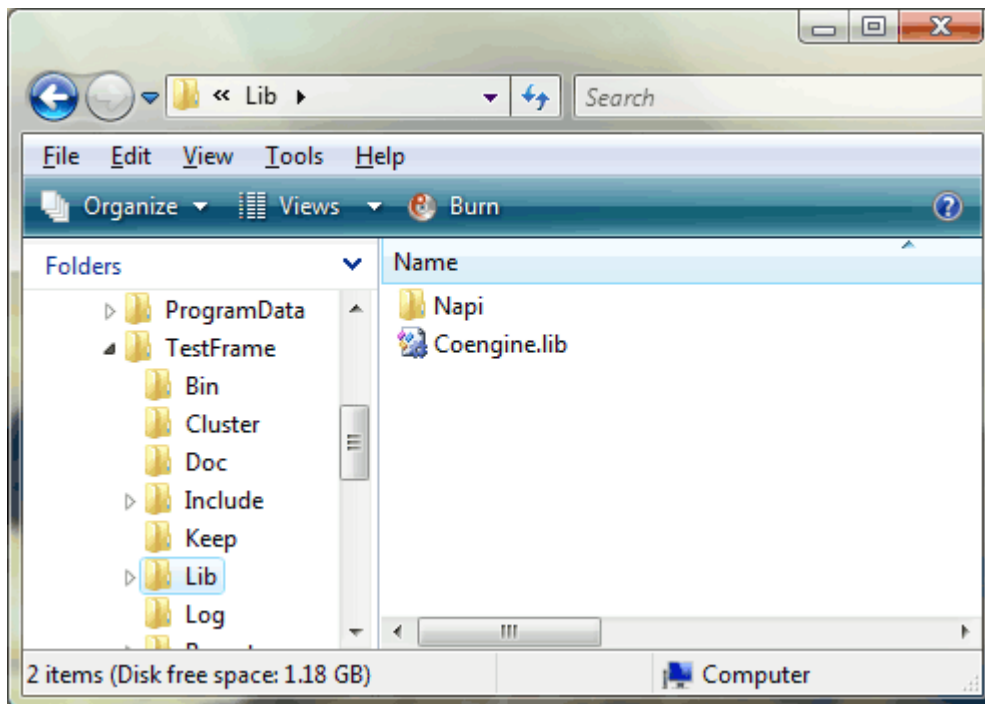


and contains the following files:

- **Keep.ini**, an empty *keep file*.

2.6 The Lib directory

The **Lib** directory is needed for the C++ scripts. The **Lib** directory contains *Library files*. The content of the **Lib** directory should look like this:

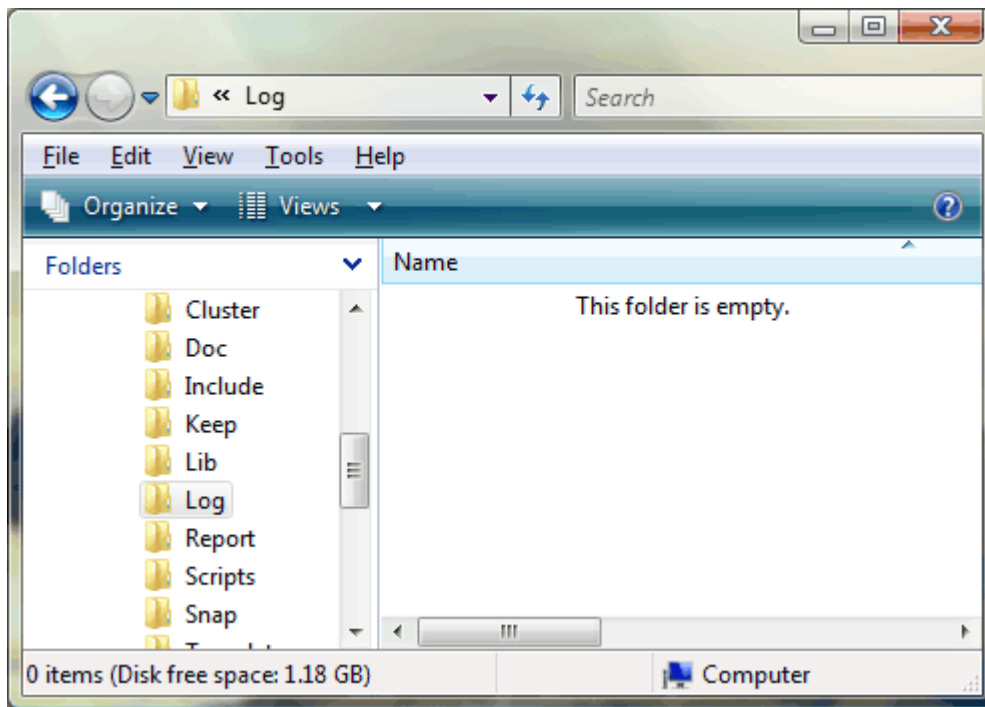


and contains the following files:

- **Coengine.lib**, the coengine *library file* (*Release build only*).
- **Napiwrapper.lib** and **Navigation.lib** (*Release build only*), in the subdirectory **Napi**.

2.7 The Log Directory

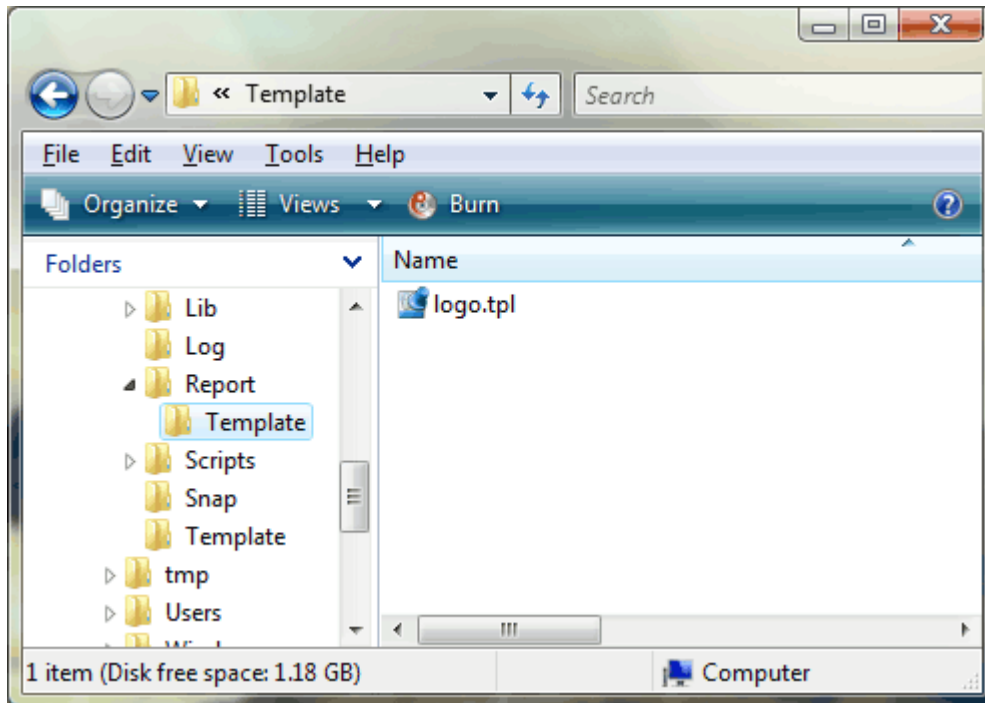
The **Log** directory contains TestFrame Engine *log files*. It is recommended to clean this directory by removing no longer needed files. The content of the **Log** directory should look like this:



Initially this directory is empty.

2.8 The Report Directory

The **Report** directory contains *test reports*. The content of the **Report** directory should look like this:

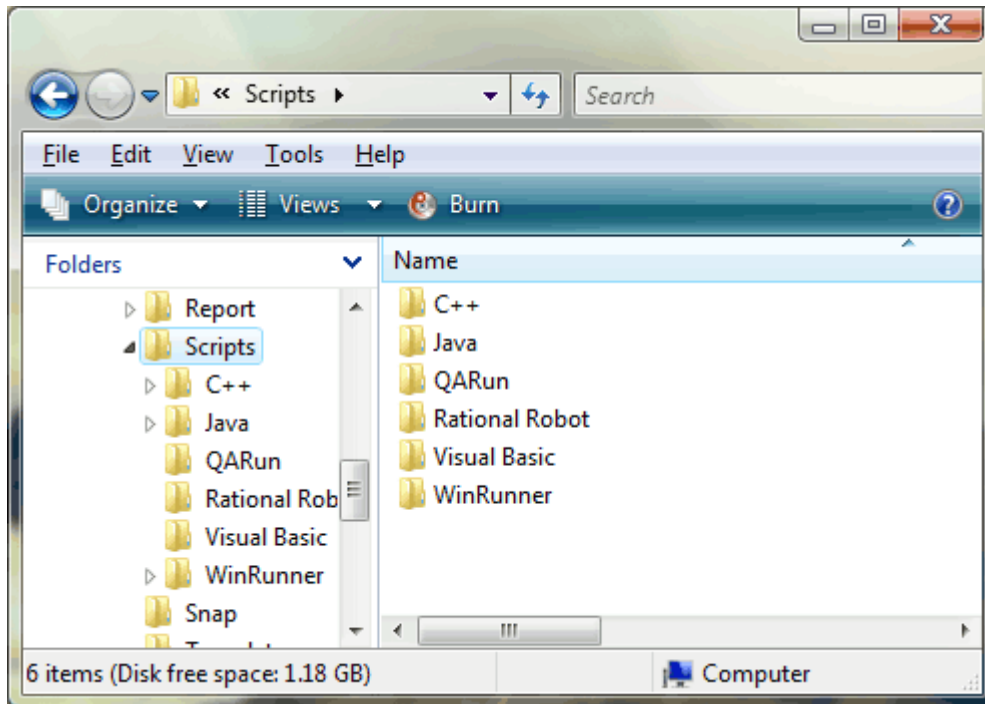


and contains the following files:

- **logo.tpl**, a subdirectory **Template** with the Logica logo (in RTF code) which is used in the *test reports*. It is possible to replace this logo with another.

2.9 The Scripts directory

The **Scripts** directory contains *script files* for different test environments/tools. The content of the **Scripts** directory should look like this:



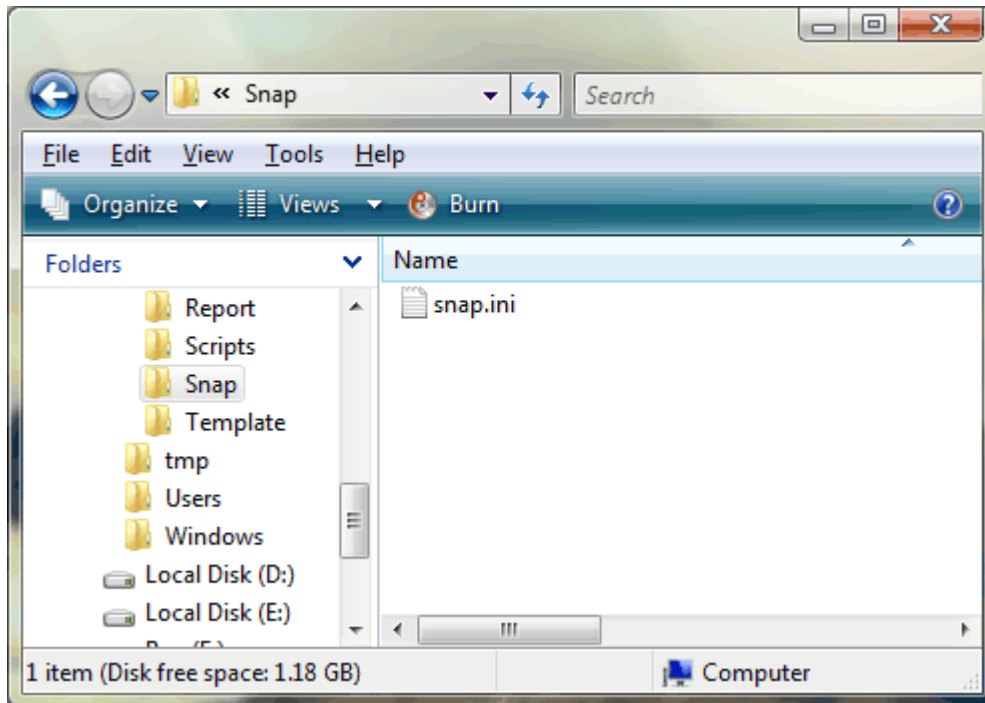
and contains the following subdirectories:

- **C++**
- **Java**
- **Rational Robot**
- **QARun**
- **Quick Test Pro**
- **Visual Basic**
- **WinRunner**

See chapter 3 for a more detailed explanation of each script subdirectory.

2.10 The Snap directory

The **Snap** directory contains *snap files*. The content of the **Snap** directory should look like this:

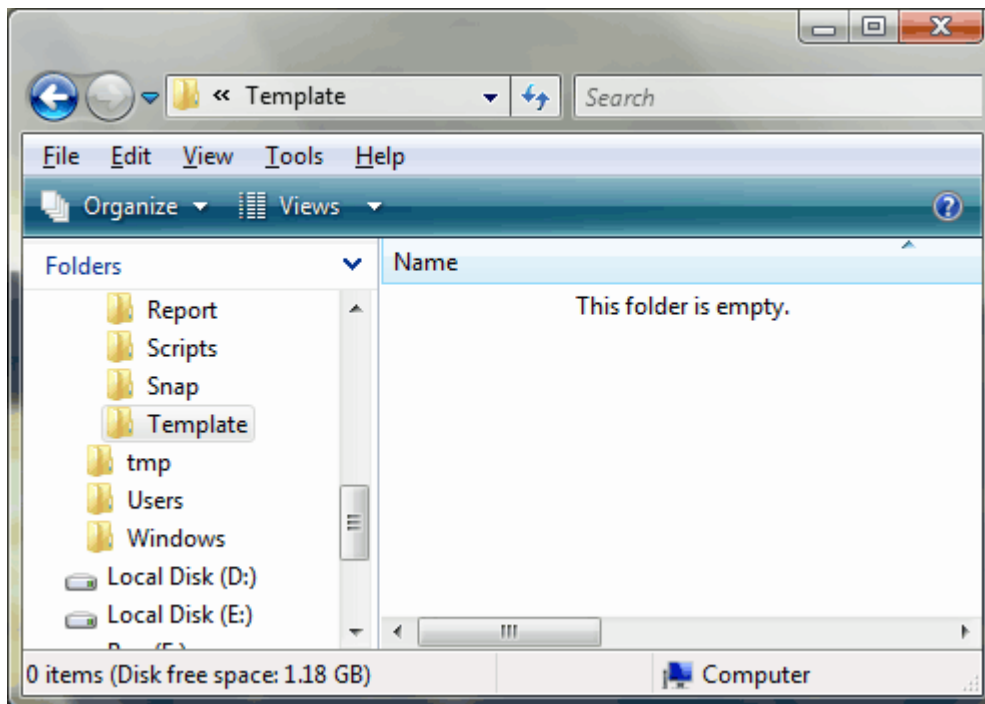


and contains the following files:

- **snap.ini**, an empty *snap file*.

2.11 The Template Directory

The **Template** directory contains TestFrame Engine *cluster templates*. The content of the **Template** directory should look like this:



Initially this directory is empty.

2.12 The license file

To be able to use the TestFrame Engine a valid license is needed. This license file **license.key** must be placed in the directory specified in the section **[LICENSE]** of the **engine.ini** file. The license file is not part of the installation of the TestFrame Engine but can be obtained by sending a request to ccctesting@logica.com.

If the TestFrame Engine can not find a file named **license.key** that contains a valid encoded license key it will not run, and an error message will be written to a file named **error.log** which is placed in the working directory.

Note that when using a test tool, this directory probably is the test tool's directory.

(**Tip:** use Find Files in the Start Menu to find **error.log**.)

2.13 The ini file

The file **engine.ini** in the root directory contains the settings for the TestFrame Engine. Initially the root directory **C:\TestFrame** is used in this ini file. If you installed TestFrame Engine in another directory you should adjust **engine.ini** accordingly.

See the TestFrame Engine User Manual for a detailed description of the settings in **engine.ini**. The content of **engine.ini** should look like this:

```
[CLUSTER]
FileName=Example.txt
Path=C:\TestFrame\Cluster\

[GUI]
Detail=Yes
LaunchGUI=Yes
OnTop=No
Port=4000

[KEEP]
FileName=keep.ini
Path=C:\TestFrame\Keep\

[LICENSE]
Company=Logica
Name=TestFrame Engine
Path=C:\TestFrame\
SerialNumber=550-000001-0-01

[LOG]
Path=C:\TestFrame\Log\

[MASTERSLAVE]
Mode=Standalone
Port=4100

[REPORT]
Arguments=Yes
FileName=ExampleReport.rtf
Overwrite=Yes
Path=C:\TestFrame\Report\
TemplatePath=C:\TestFrame\Report\Template\

[SNAP]
FileName=snap.ini
Path=C:\TestFrame\Snap\

[SYSTEM]
AnythingPrefix=&Anything
ContText=&Cont
TimePrefix=&Time()
DatePrefix=&Date()
EmptyPrefix=&Empty
EngineBuild=1
EngineVersion=2008.04
ExpressionPrefix=&
KeepPrefix=&Keep()
Language=GB
NotEmptyPrefix=&NotEmpty
Precision=6g
RedundantEscapeChars=
SnapPrefix=&Snap()
SpacesPrefix=&Spaces()
WaitForGUI=Yes

[TEMPLATE]
Path=C:\TestFrame\Template\
```

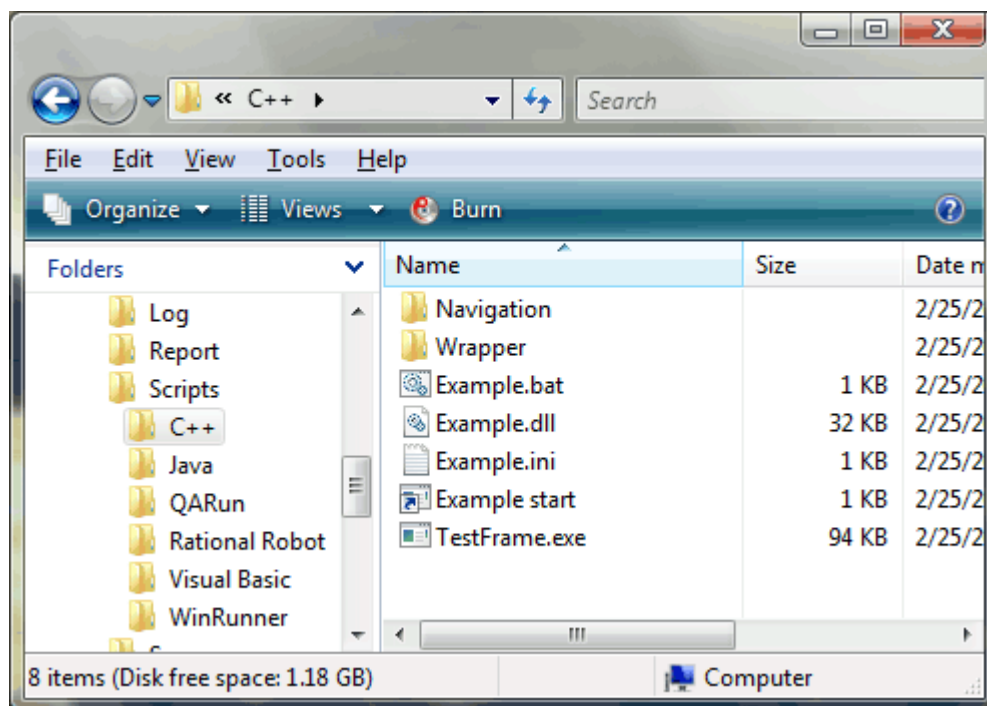
3 RUNNING THE INSTALLATION EXAMPLE

A running example of TestFrame Engine is supplied with the installation. The example can run in all six before mentioned environments. The example cluster **Example.txt** is placed in the **Cluster** directory – see also paragraph 2.2.

The following paragraphs describe how to run this cluster in each of the supported environments; the last paragraph discusses the example's results.

3.1 Running the example using C++

The **Scripts\C++** directory contains the files needed for the C++ example. The content of the **Scripts\C++** directory should look like this:



To run this example, you simply start **example.bat**. This batch file runs the application **TestFrame.exe** with the settings supplied in **example.ini**.

You can create your own **TestFrame.exe** from the files in the **Wrapper** subdirectory. A Microsoft Visual C++ *project file* (**example wrapper.dsp** for Visual C++ 6.0 and **example wrapper.vcproj** for Visual C++ .NET) is supplied for this purpose.

You also need to compile a *dll*, the *navigation library*, containing the implementation of the action words used in the example. The files for this project are located in the **Navigation** subdirectory. You will find there also the necessary *project files* (**example.dll.dsp** and **example.dll.vcproj**).

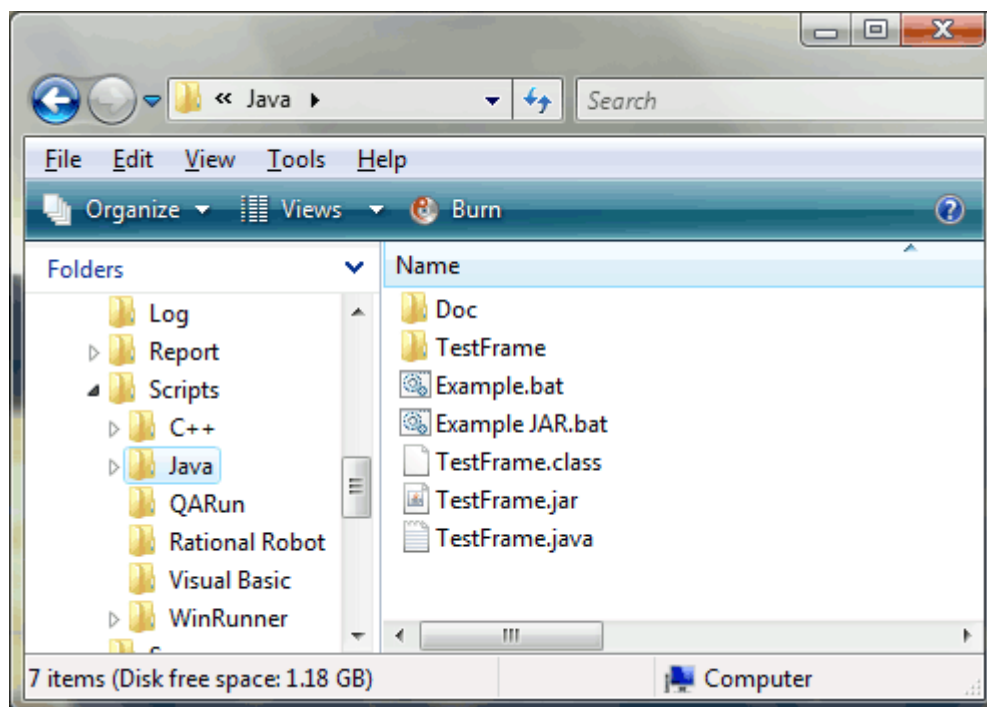
Both the executable (**TestFrame.exe**) and the *navigation library* (**Example.dll**) should be copied to the **Scripts\C++** directory. If you want to keep the ones provided with the installation you should make a copy of them.

The C++ example needs the path to an *ini file* as a command line parameter. This file tells the C++ wrapper which engine configuration file, which cluster, and which report file to use during the test run. It also tells the wrapper which navigation libraries should be loaded. Note this file is another file besides **engine.ini**.

This ini file—**Example.ini**—has only one section—**[TestFrame]**—with the four following entries: **Cluster**, **Report**, **IniFile** and **Library**. Each entry should contain a path. The **Library** entry can contain more than one path, separated by semicolons. This makes it possible to use more than one navigation library in a single test run.

3.2 Running the example using Java

The **Scripts\Java** directory contains the files needed for the Java example. The content of the **Scripts\C++** directory should look like this:



To run this example, you simply start **example.bat**. This batch file runs the Java application wrapper.

You can create your own Java wrapper. Therefore **TestFrame.java** needs to be compiled to a class file. For that you need the Java Development Kit (JDK) 1.5 (or higher) (for more information see www.javasoft.com). Open a Command Window, go to the **Scripts\Java** directory and type:

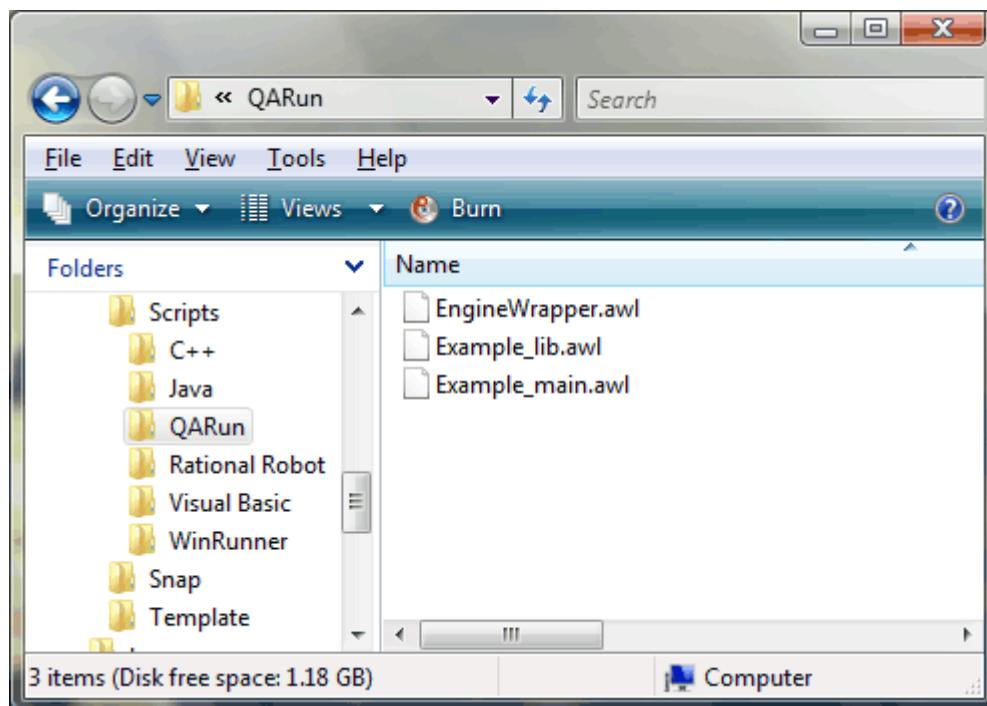
```
C:\TestFrame\Scripts\Java> javac TestFrame.java
```

This will produce the file **TestFrame.class**. This file is also added to this directory during installation. To run the example you need to provide three parameters: the name of the ini file, the cluster name, and the report name respectively. To run the example enter the following line on the command line:

```
C:\TestFrame\Scripts\Java> java TestFrame C:\TestFrame\engine.ini  
C:\TestFrame\Cluster\Example.txt c:\TestFrame\Report\ExampleReport.rtf
```

3.3 Running the example using QARun

The **Scripts\QARun** directory contains the scripts which were developed in QARun for Win32 Version 4.7.0 (Build 301). The content of the **Scripts\QARun** directory should look like this:

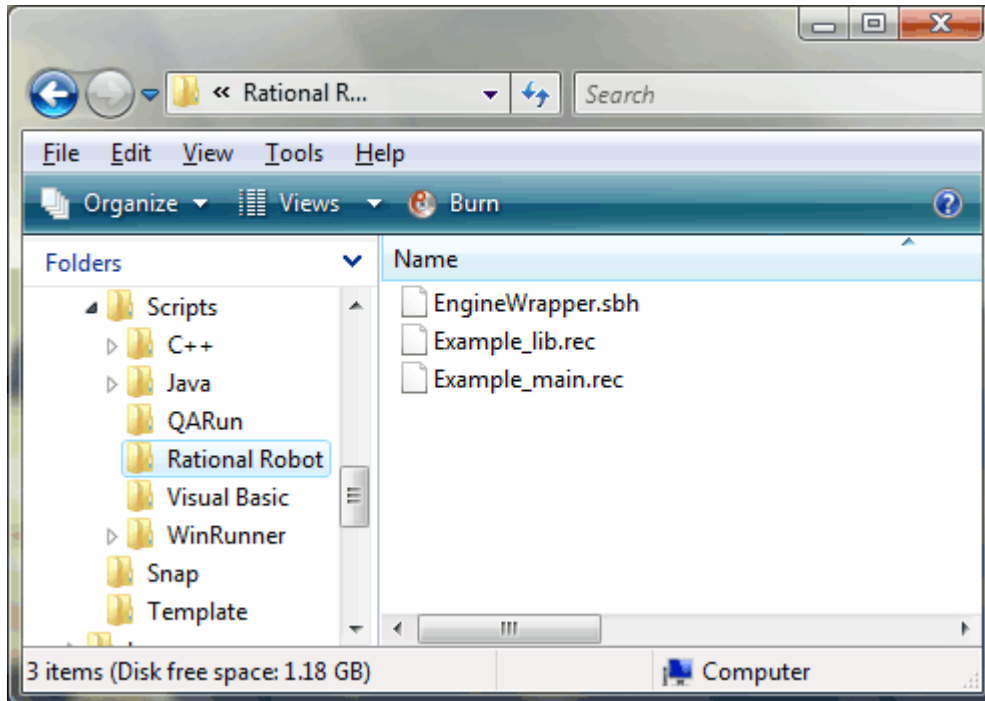


QARun uses a database in which the test scripts are stored. Since QARun databases are dependent on the version used, no database is installed, but only the *awl files*.

To run this example, start QARun, open a database and import the three *awl files* from the directory **Scripts\QARun**. Open the script **Example_main** and run it—by using *Run Script*.

3.4 Running the example using Rational Robot

The **Scripts\Rational Robot** directory contains the scripts which were developed in Rational Robot Version 7.5.0.1081. The content of the **Scripts\Rational Robot** directory should look like this:

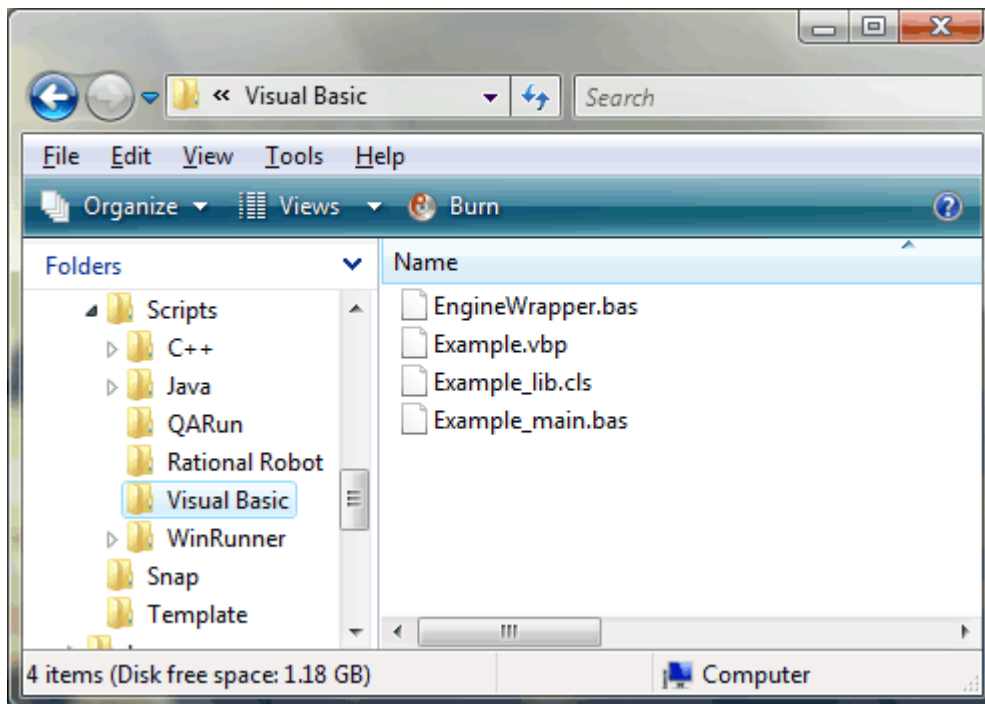


Rational Robot uses a database in which the test scripts are stored. With this example, however, no database is installed, but only the *rec files* and the *sbh file*.

To run this example, start Rational Robot, login on a repository, and place in the project as GUI scripts the two *rec files*, and as Project Header File the *sbh file* from the directory Rational Robot. Select the option *Playback Script* and choose the script **Example_main**.

3.5 Running the example using Visual Basic

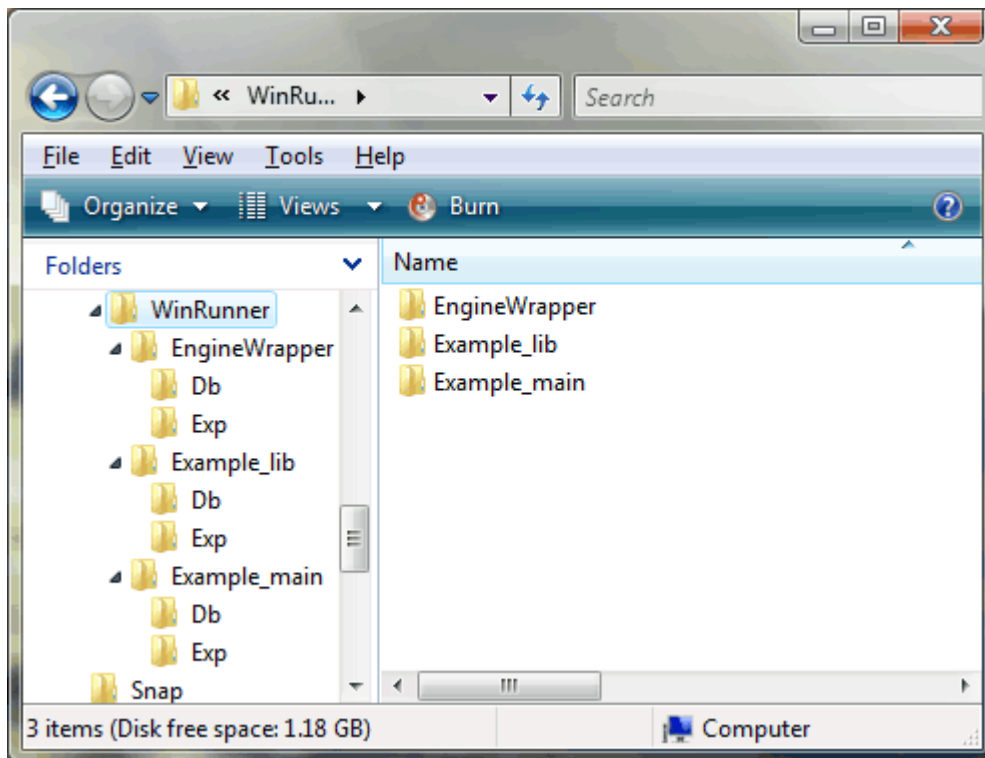
The **Scripts\Visual Basic** directory contains the files needed for the Visual Basic example. The content of the **Scripts\Visual Basic** directory should look like this:



To run the Visual Basic example, make an executable named **Example.exe**. A Microsoft Visual Basic 6.0 *project file* (**Example.vbp**) is supplied for this purpose. The executable (**Example.exe**) should be copied to the **Scripts\Visual Basic** directory.

3.6 Running the example using WinRunner

The **Scripts\WinRunner** directory contains the scripts which were developed in WinRunner version 9.2. The content of the **Scripts\WinRunner** directory should look like this:



To run this example, start WinRunner, open the script **Example_main** and run it—preferably using the option *“Run Minimized From top”*.

3.7 Results of the installation example

In the cluster **Example.txt** some functionality of the TestFrame Engine is demonstrated.

After the example test has been run, a *test report* which contains the results of the test is stored in the **Report** directory as **ExampleReport.rtf**:

