|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  | **MSG-134** | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  | **ETC FRA user guide** | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  |  |  |
|  |  | Version : | V1.4 |
|  |  | Date : | 10/01/2018 |
|  | Status : | Validé |
|  | Usage : | Livrable |
|  | Author : | FRANCE |
|  | Type : | | MUT: Manuel d'utilisation |

***TABLE OF CONTENTS***

1. Introduction 5

1.1 Purpose 5

1.2 Reference Documents 5

1.3 Terminology 5

1.3.1 Acronyms 5

1.3.2 Definitions 7

2. Prerequisites 8

2.1 Hardware prerequisites 8

2.2 Software prerequisites 8

2.2.1 MÄK RTI versions compatibility 8

2.2.2 Pitch RTI versions compatibility 8

3. Tool structure 9

3.1 Configuration directory 9

3.1.1 SUT configuration directory for CS Verification ETC 9

3.1.2 SUT configuration directory for HLA Declaration Management ETC 10

3.1.3 SUT configuration directory for HLA Object Management ETC 10

3.1.4 SUT configuration directory for HLA Services Verification ETC 11

3.2 Gradle projects 12

3.2.1 Common component ETC\_FRA\_Common 12

3.2.2 Test case CS Verification 12

3.2.3 Test case HLA Declaration Management 12

3.2.4 Test case HLA Object Management 12

3.2.5 Test case HLA Services Verification 12

4. Installation 13

5. Configuration / Setup 14

5.1 RTI configuration 14

5.1.1 MÄK RTI parameters 14

5.1.2 Pitch RTI parameters 14

5.2 IVCT configuration 14

5.2.1 Environment variables 14

5.2.2 CLASSPATH configuration 17

5.2.3 Log configuration 17

5.2.4 Language configuration 18

5.3 ETC configuration 18

5.3.1 CS Verification ETC 18

5.3.2 HLA Declaration Management ETC 19

5.3.2.1 MÄK RTI specific configuration 19

5.3.3 HLA Object Management ETC 19

5.3.3.1 MÄK RTI specific configuration 19

5.3.4 HLA Services Verification ETC 19

6. Usage 20

6.1 General instructions to launch ETC 20

6.2 CS Verification ETC 20

6.2.1 ETC Start 20

6.2.2 In-progress information 20

6.2.3 ETC Stop 21

6.2.4 Results 21

6.3 HLA Declaration Management ETC 23

6.3.1 ETC Start 23

6.3.2 In-progress information 24

6.3.3 ETC Stop 24

6.3.4 Results 24

6.4 HLA Object Management ETC 26

6.4.1 ETC Start 26

6.4.2 In-progress information 27

6.4.3 ETC Stop 27

6.4.4 Results 27

6.5 HLA Services Verification ETC 29

6.5.1 ETC Start 29

6.5.2 In-progress information 30

6.5.3 ETC Stop 30

6.5.4 Results 30

7. Troubleshooting 32

8. Annex 33

8.1 CS Verification ETC 33

8.1.1 Example of TcParam.json SuT configuration file 33

8.1.2 Example of Test Suite configuration file 33

8.1.3 Example of result file 33

8.2 HLA Declaration Management ETC 34

8.2.1 Example of TcParam.json SuT configuration file 34

8.2.2 Example of Test Suite configuration file 34

8.2.3 Example of result file 35

8.3 HLA Object Management ETC 36

8.3.1 Example of TcParam.json SuT configuration file 36

8.3.2 Example of Test Suite configuration file 36

8.3.3 Example of result file 37

8.4 HLA Services Verification ETC 38

8.4.1 Example of TcParam.json configuration file 38

8.4.2 Example of Test Suite configuration file 38

8.4.3 Example of result file 38

8.5 Synthetic presentation of IVCT 42

8.5.1 Composition 42

8.5.2 Installation 42

8.5.3 Compilation 42

8.5.4 Execution 43

8.6 rtiSimple4etcFra test federate 44

8.6.1 Structure 44

8.6.2 Installation 44

8.6.3 Configuration 44

8.6.4 Usage 44

***Figures***

[Figure 1: ETC FRA configuration overview 15](#_Toc503518967)

[Figure 2: Configuration file tree for a specific test case of a federate to be tested (SuT) 16](#_Toc503518968)

[Figure 3: Log window 18](#_Toc503518969)

page

***Tables***

[Table 1: Minimal hardware configuration for ETC FRA 8](#_Toc503518970)

[Table 2: MÄK RTI versions compatibility 8](#_Toc503518971)

[Table 3: Pitch RTI versions compatibility 8](#_Toc503518972)

[Table 4: TcParam.json parameters for CS Verification ETC 9](#_Toc503518973)

[Table 5: TcParam.json parameters for HLA Declaration Management ETC 10](#_Toc503518974)

[Table 6: TcParam.json parameters for HLA Object Management ETC 11](#_Toc503518975)

[Table 7: TcParam.json parameters for HLA Services Verification ETC 12](#_Toc503518976)

[Table 8: Problems and solutions 32](#_Toc503518977)

# Introduction

## Purpose

This document is the user guide of Test Cases (TC) developed by France, as a part of the effort of development to produce a NATO certification capability within NATO group MSG-134 (cf. document [R1]). It aims at describing how to install and use those TC.

The development of the French TC is performed in order to be used within IVCT, the Integration Verification and Certification Tool developed by Germany as a framework for the development of TC.

In MSG-134 terminology, a TC is composed of 2 parts:

* The Abstract Test Case (ATC) which corresponds to the functional specifications of a TC
* The Executable Test Case (ETC) which corresponds to the implementation of a TC (i.e. executable program that runs within IVCT)

That document is focuses on the usage of ETC.

ETCs developed by France are named ETC FRA.

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| **http://www.atome77.com/images/icones/icone-information.jpg** | Information |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | Warning |

## Reference Documents

|  |  |  |
| --- | --- | --- |
| **ID** | **Reference** | **Document name** |
| [R1] | MSG-134 Technical Activity Proposal | Technical Activity Proposal MSG-134 NATO Distributed Simulation Architecture & Design, Compliance Testing and Certification |
| [R2] | AC/323 (SGMS)D/2 | NATO Modeling and Simulation Master Plan version 1.0 |
| [R3] | IEEE 1516.1-2010 | IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) – Federate Interface Specification |
| [R4] | IEEE 1516.2-2010 | IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) – Object Model Template (OMT) Specification |
| [R5] | IEEE 1516-2010 | IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) – Framework and Rules |

## Terminology

### Acronyms

| Acronym | Meaning |
| --- | --- |
| **ATC** | Abstract Test Case |
| **ETC** | Executable Test Case |
| **FCTT** | Federate Compliance Testing Tool |
| **FOM** | Federation Object Model |
| **FRA** | France |
| **HLA** | High Level Architecture |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **IEEE 1516-2010** | Last version of HLA standard published by IEEE in August 2010 |
| **IP** | Internet Protocol |
| **IVCT** | Integration Verification and Certification Tool |
| **JDK** | Java Development Kit |
| **JSON** | JavaScript Object Notation |
| **MSG** | Modelling & Simulation Group (NATO, see also NMSG) |
| **NATO** | North Atlantic Treaty Organization |
| **RTI** | RunTime Infrastructure |
| **SOM** | Simulation Object Model |
| **SuT** | System under Test |
| **TC** | Test Case |
| **XML** | eXtensible Markup Language |

### Definitions

| Name | Definition |
| --- | --- |
| **Abstract Test Case** | A complete and independent specification of the actions required to achieve a specific test purpose (or a specified combination of test purposes), defined at the level of abstraction of a particular abstract test method, starting in a stable testing state and ending in a stable testing state. This specification may involve one or more consecutive or concurrent connections. |
| **Compliance** | The statement that an asset fulfills the required behavior rules of a given standard. |
| **Federate** | IEEE 1516-2010: An application that may be or is currently coupled with other software applications under a federation object model (FOM) Document Data (FDD) and a runtime infrastructure (RTI). |
| **Federation** | IEEE 1516-2010: A named set of federate applications and a common federation object model (FOM) that are used as a whole to achieve some specific objective. |
| **Integration, Verification, and Certification Tool** | Software framework to support integration and verification task for simulation federates and to perform the certification tests for a SuT. |
| **System under Test** | The System which is the target of Compliance Testing. |
| **Test Case** | A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. |

# Prerequisites

## Hardware prerequisites

Reference configuration:

|  |  |  |  |
| --- | --- | --- | --- |
| **Processor** | **Memory** | **Hard drive** | **Operating system** |
| Intel i5 2,5 GHz | 4 GB minimum, 8 GB recommended | About 350 MB | Windows 7 Professional |

Table 1: Minimal hardware configuration for ETC FRA

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The execution of ETC FRA has not been tested under Windows 8 or later. It does not guarantee its execution on these operating systems. |

## Software prerequisites

* JDK 8 (jdk-8u66-windows-x64.exe), environment variable JAVA\_HOME must be defined
* Pitch or MÄK RTI
* IVCT V0.3.1
* Zip tool

### MÄK RTI versions compatibility

|  |  |  |
| --- | --- | --- |
| **Version** | **Patch** | **Compatibility** |
| 4.4.1 | - | KO |
| 4.4.1 | i | OK |
| 4.4.2 | - | KO |
| 4.4.2 | h | OK |

Table 2: MÄK RTI versions compatibility

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | There is still a dysfunction detected with MÄK RTI: HLARTIversion attribute of MOM is not correctly received from RTI, so there is still a need to recompile ETC FRA in order to work with MAK RTI. |

### Pitch RTI versions compatibility

|  |  |  |
| --- | --- | --- |
| **Version** | **Patch** | **Compatibility** |
| 5.0.1.0 | - | OK |
| 5.2.0 | - | OK |
| 5.3.2.1 | - | OK |

Table 3: Pitch RTI versions compatibility

# Tool structure

ETC FRA are composed of several directories:

* one configuration directory: ETC\_FRA\_Config
* five Graddle projects (directories): ETC\_FRA\_Common, TS\_CS\_Verification, TS\_HLA\_Declaration, TS\_HLA\_Object, TS\_HLA\_Services

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | ETC FRA are delivered with a test federate named rtiSimple4etcFra.  The way to use that federate is presented in §8.5. |

## Configuration directory

The most important files of the configuration directory are:

* IVCTconfig.xml file: SuT main configuration file
* IVCTtestsuites.xml file: test suite main configuration file
* IVCTsut directory: directory that contain one subfolder for each ETC

There are about twenty subfolders under IVCTsut directory, with a name starting with a number. Each subfolder is used to test one or more ETC FRA. For that reason, there are one or more subdirectories under each subfolder, which name correspond to one ETC FRA.

### SUT configuration directory for CS Verification ETC

The name of the directory must be [SuTName]\TS\_CS\_Verification\.

That directory must contain TcParam.json configuration file as described in §8.1.1.

The following attributes must be defined in that file:

| **Attribute name** | **Attribute description** | **Attribute value example** |
| --- | --- | --- |
| **sutName** | Name of the SuT | "SuT1" |
| **resultDirectory** | Directory where the resultants files will be generated | "D:\SuT1\TS\_CS\_Verification\" |
| **fomFiles** | List of FOM files. Contains a list of filenames | N/A |
| **fileName** | One FOM filename. Repeated as many times as required | "D:\SuT1\TS\_CS\_Verification\FOM1.xml" |
| **somFiles** | List of SOM files. Contains a list of filenames | N/A |
| **fileName** | One SOM filename. Repeated as many times as required | "D:\SuT1\TS\_CS\_Verification\SOM1.xml" |

Table 4: TcParam.json parameters for CS Verification ETC

Of course, the FOM and SOM files identified in the configuration file must be present at the correct location on the file system.

In the ETC\_FRA\_Config directory, an example of such a configuration file is given in ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\.

### SUT configuration directory for HLA Declaration Management ETC

The name of the directory must be [SuTName]\TS\_HLA\_Declaration\.

That directory must contain a TcParam.json configuration file as described in §8.2.1.

The following attribute must be defined in that file:

| **Attribute name** | **Attribute description** | **Attribute value example** |
| --- | --- | --- |
| **federationName** | Name of the HLA federation | "Federation1" |
| **sutName** | Name of the SuT | "SuT1" |
| **rtiAddress** | RTI IP address | "192.168.0.1" |
| **rtiPort** | RTI port | "1234" |
| **testDuration** | Waiting time of ETC before generation of the results (sec) | "60" |
| **resultDirectory** | Directory where the resultants files will be generated | "D:\SuT1\TS\_HLA\_Declaration\" |
| **fomFiles** | List of FOM files. Contains a list of filenames | N/A |
| **fileName** | One FOM filename. Repeated as many times as required | "D:\SuT1\TS\_HLA\_Declaration\FOM1.xml" |
| **somFiles** | List of SOM files. Contains a list of filenames | N/A |
| **fileName** | One SOM filename. Repeated as many times as required | "D:\SuT1\TS\_HLA\_Declaration\SOM1.xml" |

Table 5: TcParam.json parameters for HLA Declaration Management ETC

Of course, the FOM and SOM files identified in the configuration file must be present at the correct location on the file system.

In the ETC\_FRA\_Config directory, an example of such a configuration file is given in ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Declaration\.

### SUT configuration directory for HLA Object Management ETC

The name of the directory must be [SuTName]\TS\_HLA\_Object\.

That directory must contain a TcParam.json configuration file as described in §8.3.1.

The following attribute must be defined in that file:

| **Attribute name** | **Attribute description** | **Attribute value example** |
| --- | --- | --- |
| **federationName** | Name of the HLA federation | "Federation1" |
| **sutName** | Name of the SuT | "SuT1" |
| **rtiAddress** | RTI IP address | "192.168.0.1" |
| **rtiPort** | RTI port | "1234" |
| **testDuration** | Waiting time of ETC before generation of the results (sec) | "60" |
| **resultDirectory** | Directory where the resultants files will be generated | "D:\SuT1\TS\_HLA\_Object\" |
| **fomFiles** | List of FOM files. Contains a list of filenames | N/A |
| **fileName** | One FOM filename. Repeated as many times as required | "D:\SuT1\TS\_HLA\_Object\FOM1.xml" |
| **somFiles** | List of SOM files. Contains a list of filenames | N/A |
| **fileName** | One SOM filename. Repeated as many times as required | "D:\SuT1\TS\_HLA\_Object\SOM1.xml" |

Table 6: TcParam.json parameters for HLA Object Management ETC

Of course, the FOM and SOM files identified in the configuration file must be present at the correct location on the file system.

In the ETC\_FRA\_Config directory, an example of such a configuration file is given in ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Object\.

### SUT configuration directory for HLA Services Verification ETC

The name of the directory must be [SuTName]\TS\_HLA\_Services\.

That directory must contain a TcParam.json configuration file as described in §8.4.1.

The following attribute must be defined in that file:

| **Attribute name** | **Attribute description** | **Attribute value example** |
| --- | --- | --- |
| **federationName** | Name of the HLA federation | "Federation1" |
| **sutName** | Name of the SuT | "SuT1" |
| **rtiAddress** | RTI IP address | "192.168.0.1" |
| **rtiPort** | RTI port | "1234" |
| **testDuration** | Waiting time of ETC before generation of the results (sec) | "60" |
| **resultDirectory** | Directory where the resultants files will be generated | "D:\SuT1\TS\_HLA\_Services\" |
| **fomFiles** | List of FOM files. Contains a list of filenames | N/A |
| **fileName** | One FOM filename. Repeated as many times as required | "D:\SuT1\TS\_HLA\_Services\FOM1.xml" |
| **somFiles** | List of SOM files. Contains a list of filenames | N/A |
| **fileName** | One SOM filename. Repeated as many times as required | "D:\SuT1\TS\_HLA\_Services\SOM1.xml" |

Table 7: TcParam.json parameters for HLA Services Verification ETC

Of course, the FOM and SOM files identified in the configuration file must be present at the correct location on the file system.

In the ETC\_FRA\_Config directory, an example of such a configuration file is given in ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Services\.

## Gradle projects

### Common component ETC\_FRA\_Common

This component is not a test case, but gathers all the common code that is used by ETC FRA. Its existence avoids code duplication.

This component is implemented in a Graddle project tree named ETC\_FRA\_Common. It contains several files and an ETC\_FRA\_Common directory (which contains the source code).

### Test case CS Verification

This test case is implemented in a Graddle project tree named TS\_CS\_Verification. It contains several files, TestSchedules directory (which contains only one configuration file) and a TS\_CS\_Verification directory (which contains the source code).

The TestSchedules directory must contain a file as described in §8.1.2. The name of the file will be used as a parameter to the sts (startTestSchedule) IVCT command.

### Test case HLA Declaration Management

This test case is implemented in a Graddle project tree named TS\_HLA\_Declaration. It contains several files, TestSchedules directory (which contains only one configuration file) and a TS\_HLA\_Declaration directory (which contains the source code).

The TestSchedules directory must contain a file as described in §8.2.2. The name of the file will be used as a parameter to the sts (startTestSchedule) IVCT command.

### Test case HLA Object Management

This test case is implemented in a Graddle project tree named TS\_HLA\_Object. It contains several files, TestSchedules directory (which contains only one configuration file) and a TS\_HLA\_Object directory (which contains the source code).

The TestSchedules directory must contain a file as described in §8.3.2. The name of the file will be used as a parameter to the sts (startTestSchedule) IVCT command.

### Test case HLA Services Verification

This test case is implemented in a Graddle project tree named TS\_HLA\_Services. It contains several files, TestSchedules directory (which contains only one configuration file) and a TS\_HLA\_Services directory (which contains the source code).

The TestSchedules directory must contain a file as described in §8.4.2. The name of the file will be used as a parameter to the sts (startTestSchedule) IVCT command.

# Installation

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | As a prerequisite to the installation of ETC FRA, the following environment variables must be define:   * IVCT\_CONF which corresponds to IVCT configuration directory * IVCT\_TS\_HOME which corresponds to ETC root directory |

To install ETC FRA, perform the following steps:

* Create a configuration directory corresponding to %IVCT\_CONF%, with a IVCTconfig.xml file and a IVCTsut subdirectory that contain one subfolder for each SuT (as described in §5.2.1). If you are reusing ETC\_FRA\_Config configuration directory, just copy all the files present of that directory to the directory corresponding to %IVCT\_CONF%
* Modify IVCTconfig.xml file in order to be sure that the value of <sutDir> tag corresponds to an existing directory (see §5.2.1)
* Create IVCTtestsuites.xml file in %IVCT\_TS\_HOME%. If you are reusing ETC\_FRA\_Config configuration directory, just move IVCTtestsuites.xml file to the directory described in %IVCT\_TS\_HOME%
* Copy the 5 Gradle projects in %IVCT\_TS\_HOME%
* For ETC\_FRA\_Common Common component:
  + Execute the Gradle compilation from within ETC\_FRA\_Common subdirectory:
    - gradlew install
* Then for each ETC:
  + Execute the Gradle compilation from within ETC directory:
    - gradlew install
  + Extract the generated .zip file in [ETCName]\[ETCName]\build\distributions subdirectory (Example: TS\_CS\_Verification\TS\_CS\_Verification\build\distributions)

The ETC FRA are now installed.

# Configuration / Setup

## RTI configuration

ETC FRA are able to certify a federate that uses one of the two main RTIs available on the market. The first one is delivered by Pitch Technologies and the second one is delivered by VT MÄK (reseller in Europe: Antycip Simulation). The application cannot use both RTI simultaneously. The choice has to be done using IVCT[[1]](#footnote-1). The following chapters explain how to configure the RTI used.

### MÄK RTI parameters

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | If the federate use MÄK RTI, it is necessary to configure the following parameters in the rid.mtl file:   * RTI\_forceFullCompliance 1 * RTI\_processingModel 0 (if the test federate is synchronous) * RTI\_momServiceAvailable 1 (set to 1 by default if RTI\_forceFullCompliance is set to 1) * RTI\_momVerboseLevel 0   For version 4.4.2 from patch h, it is also required to activate the following parameter:   * RTI\_momEvokeCallbackObserverEnabled 1 |

### Pitch RTI parameters

No specific setup is required.

## IVCT configuration

### Environment variables

IVCT Framework configuration is based on the following environment variables:

* IVCT\_CONF must contain the absolute path of the folder where the IVCTconfig.xml file is located, which itself describes the absolute location of the folders where SuT configurations are located
* IVCT\_HOME must contain the absolute path of the folder where the IVCT Framework is located
* IVCT\_TS\_HOME must contain the absolute path of the folder where the IVCTtestsuites.xml file is located, which itself describes the Java packages of the dynamic libraries of the ETCs

Overview:

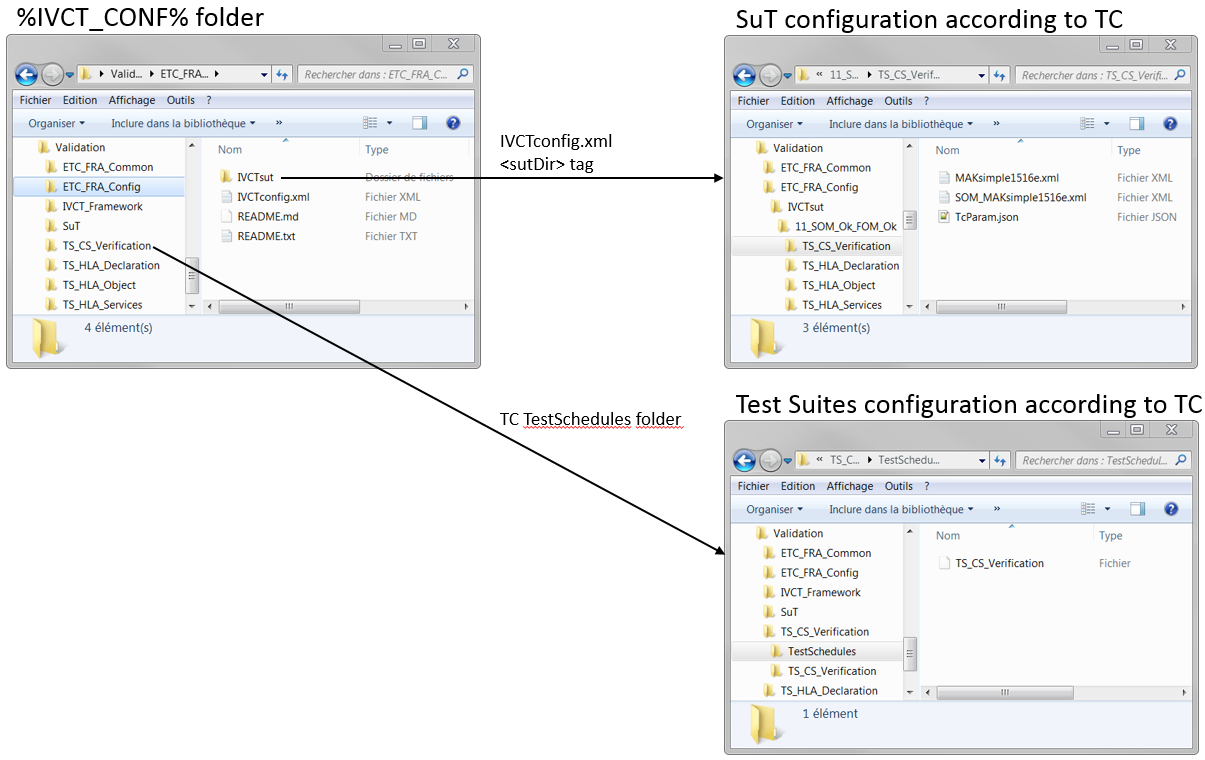


Figure 1: ETC FRA configuration overview

Example of IVCTconfig.xml file content:

**<?xml version="1.0" encoding="UTF-8"?>**

**<ivctConfig>**

**<pathNames>**

**<sutDir>D:\\Users\\HLA\\Documents\\GitHub\\ETC\_FRA\_Config\\IVCTsut</sutDir>**

**</pathNames>**

**</ivctConfig>**

The folder specified in the <sutDir> tag contains one subfolder per federate to be tested (SuT named 11\_SOM\_Ok\_FOM\_Ok in the screenshot below). This subfolder contains itself one subdirectory for each test case that includes a JSON configuration file (TcParam.json) describing the parameters of the test case. If necessary, other SuT-specific configuration files for this test case are added to that subdirectory.

Example of a test case subdirectory content corresponding to a SuT:

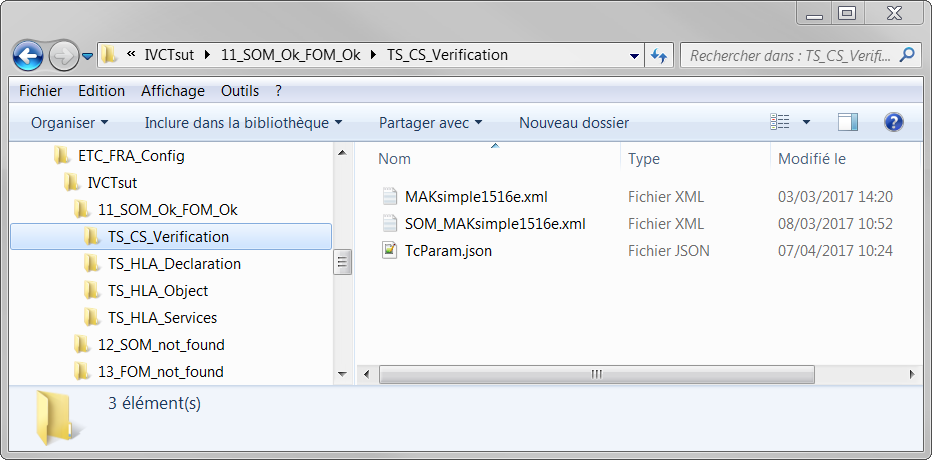


Figure 2: Configuration file tree for a specific test case of a federate to be tested (SuT)

The JSON configuration file has the following format:

**{**

**"sutName" : "TestFederate"**

**"resultDirectory" : "…\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification"**

**"fomFiles" : [**

**{ "fileName" : "…\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\MAKsimple1516e.xml" }**

**]**

**"somFiles" : [**

**{ "fileName" : "…\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\SOM\_MAKsimple1516e.xml" }**

**]**

**}**

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The file paths in the JSON file are "absolute" and, in the previous illustration, the beginning of the paths has been replaced with "..." for more clarity. |

The folder specified by the IVCT\_TS\_HOME variable contains the IVCTtestsuites.xml file which indicates IVCT Framework how to access to ETCs:

**<?xml version="1.0" encoding="UTF-8"?>**

**<ivctConfig>**

**<testSuites>**

**<testSuite>**

**<name>TS\_CS\_Verification</name>**

**<packageName>nato.ivct.etc.fr.tc\_cs\_verification</packageName>**

**<tsRunFolder>TS\_CS\_Verification\\TS\_CS\_Verification\\bin</tsRunFolder>**

**</testSuite>**

**<testSuite>**

**<name>TS\_HLA\_Declaration</name>**

**<packageName>nato.ivct.etc.fr.tc\_hla\_declaration</packageName>**

**<tsRunFolder>TS\_HLA\_Declaration\\TS\_HLA\_Declaration\\bin</tsRunFolder>**

**</testSuite>**

**<testSuite>**

**<name>TS\_HLA\_Object</name>**

**<packageName>nato.ivct.etc.fr.tc\_hla\_object</packageName>**

**<tsRunFolder>TS\_HLA\_Object\\TS\_HLA\_Object\\bin</tsRunFolder>**

**</testSuite>**

**<testSuite>**

**<name>TS\_HLA\_Services</name>**

**<packageName>nato.ivct.etc.fr.tc\_hla\_services</packageName>**

**<tsRunFolder>TS\_HLA\_Services\\TS\_HLA\_Services\\bin</tsRunFolder>**

**</testSuite>**

**</testSuites>**

**</ivctConfig>**

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The <name> tag corresponds to the test case installation subfolder under the folder specified by IVCT\_TS\_HOME environment variable.  The <packageName> and <tsRunFolder> tags refer respectively to the Java package name and the ETC runtime directory (based on the value of the IVCT\_TS\_HOME environment variable). |

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The file path separator used is "\\" to avoid any Escape sequence interpretation. For example, "\b" is interpreted as a backspace character. |

The test case installation subfolder contains a (freely-named) file listing the ETCs to be executed. The name of an ETC is a Java class that implements the AbstractTestCase interface.

### CLASSPATH configuration

Change the value of CLASSPATH variable in IVCT (file %IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat) to refer to an ETC FRA dynamic library.

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | The CLASSPATH variable must be set using the IVCT\_TS\_HOME variable in the user interface launching script named “%IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat ("Cmd Line Tool" component) of the IVCT framework.  For instance, for CS Verification ETC, the following value must be added to CLASSPATH variable:  %IVCT\_TS\_HOME%\TS\_CS\_Verification\TS\_CS\_Verification\build\distributions\TS\_CS\_Verification-X.Y.Z\lib\\* |

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | A specific value for CLASSPATH variable must be set for each ETC, because the name of the ETC is part of the path.  For instance, for HLA Services Verification ETC, the following value must be added to CLASSPATH variable:  %IVCT\_TS\_HOME%\TS\_HLA\_Services\TS\_HLA\_Services\build\distributions\TS\_HLA\_Services-X.Y.Z\lib\\* |

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | The CLASSPATH variable must be correctly set as any inconsistency makes the test case impossible to execute (Java ClassNotFoundException error). |

### Log configuration

In order to have the right kind of log, please:

* Modify logback.xml in %IVCT\_HOME%\LogSink\build\distributions\LogSink-X.Y.Z\lib\LogSink-X.Y.Z.jar to add line:

<logger name="nato.ivct.etc" level="INFO" />

* Modify JMSTestRunner\_logback.xml in %IVCT\_TS\_HOME%\[ETCName]\[ETCName]\build\distributions\[ETCName]-X.Y.Z\lib\TC-X.Y.Z.jar to change line

From

<root level="WARN">

To

<root level="INFO">

With that configuration, the different steps performed by an ETC are logged in the log window as follow:

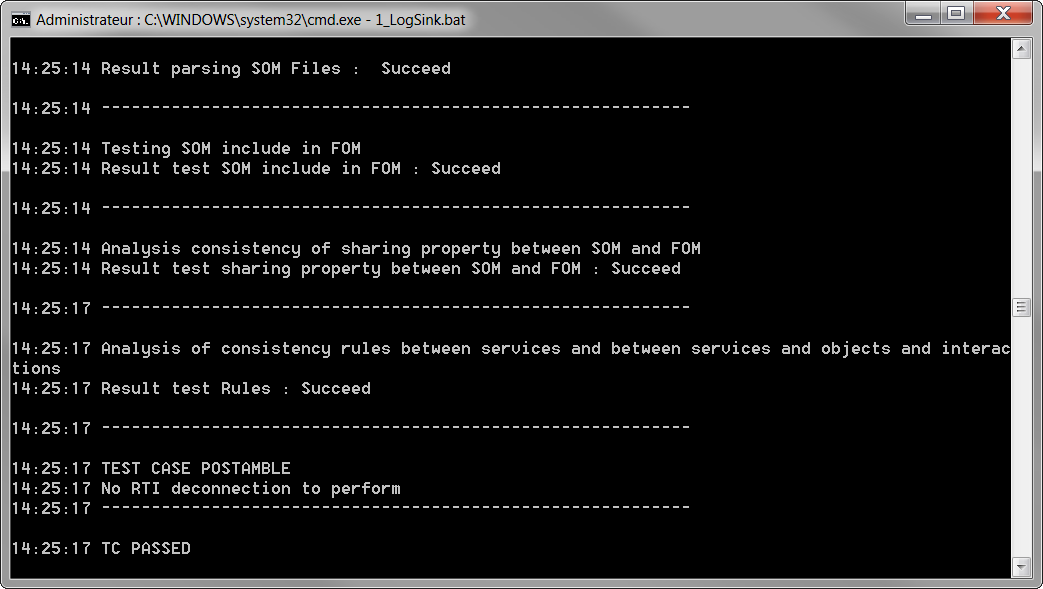


Figure 3: Log window

The same information is logged in the log file generated in %IVCT\_HOME%\LogSink\build\distributions\LogSink-X.Y.Z\logs\.

### Language configuration

Change the value of JAVA\_OPTS in IVCT (file %IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat):

* English: set JAVA\_OPTS=-Duser.country=US -Duser.language=EN
* French: set JAVA\_OPTS=-Duser.country=FR -Duser.language=FR

## ETC configuration

### CS Verification ETC

Modify paths and values inside %IVCT\_CONF%\IVCTsut\[SuTName]\CS\_Verification\TcParam.json to set the correct configuration for that ETC (see §3.1.1).

If required, copy FOM and SOM files in the directories as specified in TcParam.json.

### HLA Declaration Management ETC

Modify paths and values inside %IVCT\_CONF%\IVCTsut\[SuTName]\TS\_HLA\_Declaration\TcParam.json file to set the correct configuration for that ETC (see §3.1.2).

If required, copy FOM and SOM files in the suitable directories as specified in TcParam.json file.

#### MÄK RTI specific configuration

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | Because of a dysfunction detected with MÄK RTI, that ETC requires some source code modification and recompiling in order to work with MÄK RTI.   1. Edit file named:   TS\_HLA\_Declaration\TS\_HLA\_Declaration\src\main\java\nato\ivct\etc\fr\tc\_lib\_hla\_declaration\HLA\_Declaration\_BaseModel.java   1. Assign true to variable RTImak:   private boolean RTImak = **true**;   1. Recompile ETC library:    * + Execute the Gradle compilation from within TS\_HLA\_Declaration\ directory:        - gradlew install      + Extract the generated .zip file in TS\_HLA\_Declaration\TS\_HLA\_Declaration\build\distributions subdirectory |

### HLA Object Management ETC

Modify paths and values inside %IVCT\_CONF%\IVCTsut\[SuTName]\TS\_HLA\_Object\TcParam.json file to set the correct configuration for that ETC (see §3.1.3).

If required, copy FOM and SOM files in the suitable directories as specified in TcParam.json file.

#### MÄK RTI specific configuration

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | Because of a dysfunction detected with MÄK RTI, that ETC requires some source code modification and recompiling in order to work with MÄK RTI.   1. Edit file named:   TS\_HLA\_Object\TS\_HLA\_Object\src\main\java\nato\ivct\etc\fr\tc\_lib\_hla\_object\HLA\_Object\_BaseModel.java   1. Assign true to variable RTImak:   private boolean RTImak = **true**;   1. Recompile ETC library:    * + Execute the Gradle compilation from within TS\_HLA\_Object\ directory:        - gradlew install      + Extract the generated .zip file in TS\_HLA\_Object\TS\_HLA\_Object\build\distributions subdirectory |

### HLA Services Verification ETC

Modify paths and values inside %IVCT\_CONF%\IVCTsut\[SuTName]\TS\_HLA\_Services\TcParam.json file to set the correct configuration for that ETC (see §3.1.4).

If required, copy FOM and SOM files in the suitable directories as specified in TcParam.json file.

# Usage

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | In order for ETC FRA to works properly, **they must be started before launching SuT**. |

## General instructions to launch ETC

An ETC FRA execution is started by launching %IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat script, which then allows to enter test case execution commands (according to its configuration) in the command line tool.

For example:

**ssut 11\_SOM\_Ok\_FOM\_Ok**

**st TS\_CS\_Verification**

**sts TS\_CS\_Verification**

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | Reminder of the main IVCT commands:   * ssut (setSUT): Specifies the name of the SuT (matching a subfolder in the <sutDir> directory of the IVCTconfig.xml file) * st (setTestSuites): Indicates test suite to be executed (matching both a subfolder contained in the directory specified by the ssut command and a name of a <name> tag of the IVCTtestsuites.xml file) * sts (startTestSchedule): Indicates the test schedule to be started (matching a file included in the directory indicated by the st command) |

## CS Verification ETC

### ETC Start

Example of entered commands (bold) in the command line tool:

> **ssut 11\_SOM\_Ok\_FOM\_Ok**

> **st TS\_CS\_Verification**

> **sts TS\_CS\_Verification**

> Start Test Case: TC\_001\_Files\_Check (1 of 1)

The verdict is: TC\_001\_Files\_Check PASSED ok

Verdicts are:

Test schedule finished: TS\_CS\_Verification

### In-progress information

The log window shows following messages during the verification process:

15:23:13 -----------------------------------------------------------

Test purpose

Check federate FOM/SOM files existence

Check federate FOM/SOM files parsing

Check federate FOM/SOM files sharing options

Check federate FOM/SOM files rules conformance

15:23:13 TEST CASE PREAMBLE

15:23:13 No RTI connection to perform

15:23:13 -----------------------------------------------------------

15:23:13 TEST CASE BODY

15:23:13 -----------------------------------------------------------

15:23:13 Federate name :TestFederate

15:23:13 -----------------------------------------------------------

15:23:13 Testing FOM Files

15:23:13 FOM Files :

15:23:13 D:\Users\HLA\Desktop\Validation\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\MAKsimple1516e.xml

15:23:13

15:23:13 Result parsing FOM Files : Succeed

15:23:13 -----------------------------------------------------------

15:23:13 Testing SOM Files

15:23:13 SOM Files :

15:23:13 D:\Users\HLA\Desktop\Validation\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\SOM\_MAKsimple1516e.xml

15:23:13

15:23:13 Result parsing SOM Files : Succeed

15:23:13 -----------------------------------------------------------

15:23:13 Testing SOM include in FOM

15:23:13 Result test SOM include in FOM : Succeed

15:23:13 -----------------------------------------------------------

15:23:13 Analysis consistency of sharing property between SOM and FOM

15:23:13 Result test sharing property between SOM and FOM : Succeed

15:23:17 -----------------------------------------------------------

15:23:17 Analysis of consistency rules between services and between services and objects and interactions

15:23:17 Result test Rules : Succeed

15:23:17 -----------------------------------------------------------

15:23:17 TEST CASE POSTAMBLE

15:23:17 No RTI deconnection to perform

15:23:17 -----------------------------------------------------------

15:23:17 TC PASSED

### ETC Stop

ETC automatically stops after verification.

### Results

A result file is created in the folder designated by the resultDirectory parameter of the configuration file (see §3.1.1).

The result file is named CS\_Verification\_report\_ followed by the execution date and time of the test case.

The result file contains:

* The name of the SuT (from the configuration file)
* The success status of each step of the CS Verification ETC:
  + Existence and XML validity of the FOM files
  + Existence and XML validity of the SOM files
  + SOM inclusion into FOM
  + Validity of the sharing state between SOM and FOM files
  + Rules compliance state

Refer to §8.1.3 for an example of a CS Verification result file.

## HLA Declaration Management ETC

### ETC Start

Example of entered commands (bold) in the UI terminal window:

> **ssut 11\_SOM\_Ok\_FOM\_Ok**

> **st TS\_HLA\_Declaration**

> **sts TS\_HLA\_Declaration**

> Start Test Case: TC\_001\_Publish\_Subscribe\_Check (1 of 1)

The verdict is: TC\_001\_Publish\_Subscribe\_Check PASSED ok

Verdicts are:

Test schedule finished: TS\_HLA\_Declaration

### In-progress information

The log window shows following messages during the verification process:

TC\_001\_Publish\_Subscribe\_Check - 18:32:27 -----------------------------------------------------------

Test purpose

Check federate objects publication conformance

Check federate interactions publication conformance

Observe federate for objects and interactions publications

Compare with SOM/FOM publication declarations

TC\_001\_Publish\_Subscribe\_Check - 18:32:27 TEST CASE PREAMBLE

TC\_001\_Publish\_Subscribe\_Check - 18:32:27 Testing FOM Files

TC\_001\_Publish\_Subscribe\_Check - 18:32:30 Testing SOM Files

TC\_001\_Publish\_Subscribe\_Check - 18:32:30 Testing SOM include in FOM

TC\_001\_Publish\_Subscribe\_Check - 18:32:30 Analysis consistency of sharing property between SOM and FOM

TC\_001\_Publish\_Subscribe\_Check - 18:32:34 Analysis of consistency rules betweenservices and between services and objects and interactions

TC\_001\_Publish\_Subscribe\_Check - 18:32:36 RTI connected successfully

TC\_001\_Publish\_Subscribe\_Check - 18:32:36 -----------------------------------------------------------

TC\_001\_Publish\_Subscribe\_Check - 18:32:36 TEST CASE BODY

TC\_001\_Publish\_Subscribe\_Check - 18:32:36 Wait while federate stimulation

- 18:32:44 following federate TestFederate

- 18:32:45 publishInteractionClass HLAinteractionRoot.WeaponFire

- 18:32:45 subscribeInteractionClass HLAinteractionRoot.WeaponFire

TC\_001\_Publish\_Subscribe\_Check - 18:33:36 Stop waiting federate actions

TC\_001\_Publish\_Subscribe\_Check - 18:33:36 ######################################

#####################

Certification results "TestFederate"

Date : 2017\_08\_23\_18h33m36s

Results for the data and the interactions certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

…/…

TC\_001\_Publish\_Subscribe\_Check - 18:33:36 TEST CASE POSTAMBLE

TC\_001\_Publish\_Subscribe\_Check - 18:33:38 TC PASSED

A very large amount of log information is displayed by IVCT internal processes while waiting for the end of the federate execution. Non-useful lines have been removed from the previous example.

### ETC Stop

ETC automatically stops after specified time period (testDuration attribute, see §3.1.2).

### Results

Two result files are created in the folder designated by the resultDirectory parameter of configuration file (see §3.1.2).

The result files are named:

* HLA\_Declaration\_certified\_data\_ followed by the execution date and time of the test case for the compliant declarations done by the SuT
* HLA\_Declaration\_non\_certified\_data\_ followed by the execution date and time of the test case for the non-compliant declarations done by the SuT

Both result files contain:

* A header with:
  + The name of the SuT (from the configuration file)
  + The execution date and time
* A list of objects declared as published and subscribed by the SuT
* A list of interactions declared as published and subscribed by the SuT

The HLA\_Declaration\_certified\_data\_\* file contains the expected published/subscribed (and really published/subscribed) objects and interactions declarations.

The HLA\_Declaration\_non\_certified\_data\_\* contains the expected published/subscribed (and not really published/subscribed) objects and interactions declarations.

Refer to §8.2.3 for an example of an HLA Declaration Management result file.

## HLA Object Management ETC

### ETC Start

Example of entered commands (bold) in the UI terminal window:

> **ssut 11\_SOM\_Ok\_FOM\_Ok**

> **st TS\_HLA\_Object**

> **sts TS\_HLA\_Object**

> Start Test Case: TC\_001\_Object\_Interaction\_Check (1 of 1)

The verdict is: TC\_001\_Object\_Interaction\_Check PASSED ok

Verdicts are:

Test schedule finished: TS\_HLA\_Object

### In-progress information

The log window shows following messages during the verification process:

TC\_001\_Object\_Interaction\_Check - 18:46:56 -----------------------------------------------------------

Test purpose

Check federate objects/interactions conformance

Observe federate for objects/interactions creation

Compare with SOM/FOM files publications

TC\_001\_Object\_Interaction\_Check - 18:46:56 TEST CASE PREAMBLE

TC\_001\_Object\_Interaction\_Check - 18:46:56 Testing FOM Files

TC\_001\_Object\_Interaction\_Check - 18:46:58 Testing SOM Files

TC\_001\_Object\_Interaction\_Check - 18:46:59 Testing SOM include in FOM

TC\_001\_Object\_Interaction\_Check - 18:46:59 Analysis consistency of sharing property between SOM and FOM

TC\_001\_Object\_Interaction\_Check - 18:47:04 Analysis of consistency rules between services and between services and objects and interactions

TC\_001\_Object\_Interaction\_Check - 18:47:05 RTI connected successfully

TC\_001\_Object\_Interaction\_Check - 18:47:05 -----------------------------------------------------------

TC\_001\_Object\_Interaction\_Check - 18:47:05 TEST CASE BODY

TC\_001\_Object\_Interaction\_Check - 18:47:05 Wait while federate stimulation

- 18:47:14 following federate TestFederate

TC\_001\_Object\_Interaction\_Check - 18:48:05 Stop waiting federate actions

TC\_001\_Object\_Interaction\_Check - 18:48:05 ###########################################################

Certification results "TestFederate"

Date : 2017\_08\_23\_18h48m05s

Results for the data and the interactions certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

…/…

TC\_001\_Object\_Interaction\_Check - 18:48:05 TEST CASE POSTAMBLE

TC\_001\_Object\_Interaction\_Check - 18:48:07 TC PASSED

A very large amount of log information is displayed by IVCT internal processes while waiting for the end of the federate execution. Non-useful lines have been removed from the previous example.

### ETC Stop

ETC automatically stops after specified time period (testDuration attribute, see §3.1.3).

### Results

Two result files are created in the folder designated by the resultDirectory parameter of configuration file (see §3.1.3).

The result files are named:

* HLA\_Object\_certified\_data\_ followed by the execution date and time of the test case for the compliant declarations done by the SuT
* HLA\_Object\_non\_certified\_data\_ followed by the execution date and time of the test case for the non-compliant declarations done by the SuT

Both result files contain:

* A header with:
  + The name of the SuT (from the configuration file)
  + The execution date and time
* A list of objects instances sent and received by the SuT
* A list of interactions instances sent and received by the SuT

The HLA\_Object\_certified\_data\_\* contains the expected sent/received (and really sent/received) objects and interactions instances.

The HLA\_Object\_non\_certified\_data\_\* contains the expected sent/received (and not really sent/received) objects and interactions instances.

Refer to §8.3.3 for an example of an HLA Object Management result file.

## HLA Services Verification ETC

### ETC Start

Example of entered commands (bold) in the UI terminal window:

> **ssut 11\_SOM\_Ok\_FOM\_Ok**

> **st TS\_HLA\_Services**

> **sts TS\_HLA\_Services**

> Start Test Case: TC\_001\_Services\_Check (1 of 1)

The verdict is: TC\_001\_Services\_Check PASSED ok

Verdicts are:

Test schedule finished: TS\_HLA\_Services

### In-progress information

The log window shows following messages during the verification process:

TC\_001\_Services\_Check - 19:39:45 -----------------------------------------------------------

Test purpose

Check federate services conformance

Observe federate for services

Compare with SOM/FOM files services

TC\_001\_Services\_Check - 19:39:45 TEST CASE PREAMBLE

TC\_001\_Services\_Check - 19:39:45 Testing FOM Files

TC\_001\_Services\_Check - 19:39:47 Testing SOM Files

TC\_001\_Services\_Check - 19:39:48 Testing SOM include in FOM

TC\_001\_Services\_Check - 19:39:48 Analysis consistency of sharing property between SOM and FOM

TC\_001\_Services\_Check - 19:39:53 Analysis of consistency rules between servicesand between services and objects and interactions

TC\_001\_Services\_Check - 19:39:54 RTI connected successfully

TC\_001\_Services\_Check - 19:39:54 -----------------------------------------------------------

TC\_001\_Services\_Check - 19:39:54 TEST CASE BODY

TC\_001\_Services\_Check - 19:39:54 Wait while federate stimulation

- 19:40:04 following federate TestFederate

TC\_001\_Services\_Check - 19:40:54 Stop waiting federate actions

TC\_001\_Services\_Check - 19:40:54 ###############################################

############

Certification results "TestFederate"

Date : 2017\_08\_23\_19h40m54s

Results for the services certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

…/…

TC\_001\_Services\_Check - 19:40:54 TEST CASE POSTAMBLE

TC\_001\_Services\_Check - 19:40:56 TC PASSED

A very large amount of log information is displayed by IVCT internal processes while waiting for the end of the federate execution. Non-useful lines have been removed from the previous example.

### ETC Stop

ETC automatically stops after specified time period (testDuration attribute, see §3.1.4)

### Results

Two result files are created in the folder designated by the resultDirectory parameter of configuration file (see §3.1.4).

The result files are named:

* HLA\_Services\_certified\_data\_ followed by the execution date and time of the test case for the compliant services used by the SuT
* HLA\_Services\_non\_certified\_data\_ followed by the execution date and time of the test case for the non-compliant services used by the SuT

Both result files contain:

* A header with:
  + The name of the SuT (from the configuration file)
  + The execution date and time
* A list of services used or not used by the SuT

The HLA\_Services\_certified\_services\_\* contains the expected used (resp. not used) and effectively used (resp. not used) services.

The HLA\_Services\_non\_certified\_services\_\* contains the expected used (resp. not used) and effectively not used (resp. used) services.

Refer to §8.4.3 for an example of an HLA Services Verification result file.

# Troubleshooting

| **Problem** | **Solution** |
| --- | --- |
| Java ClassNotFoundException error when starting ETC | Implementation class for ETC not found.  Check the value of CLASSPATH variable in %IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat script (see §5.2.2). |
| No log from ETC at runtime | Incorrect values in log configuration files.  Check the content of logback.xml in %IVCT\_HOME%\LogSink\build\distributions\LogSink-X.Y.Z\lib\LogSink-X.Y.Z.jar and JMSTestRunner\_logback.xml in %IVCT\_TS\_HOME%\[ETCName]\[ETCName]\build\distributions\[ETCName]-X.Y.Z\lib\TC-X.Y.Z.jar (see §5.2.3) |
| Invalid language when starting ETC | Value of JAVA\_OPTS environment variable incorrectly set.  Check the value of JAVA\_OPTS variable in %IVCT\_HOME%\UI\build\distributions\UI-X.Y.Z\bin\UI.bat script (see §5.2.4) |
| Everything seems to work properly, all federates (including SuT and ETC) are in the same HLA federation, but ETC does not receive any information sent by SuT. | Different versions of the RTI library are used by different federates.  Link all the federates (including ETC) with the same version of the RTI library |

Table 8: Problems and solutions

# Annex

## CS Verification ETC

### Example of TcParam.json SuT configuration file

**{**

**"sutName" : "TestFederate"**

**"resultDirectory" : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification"**

**"fomFiles" : [**

**{ "fileName"    : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\MAKsimple1516e.xml" }**

**]**

**"somFiles" : [**

**{ "fileName"    : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\SOM\_MAKsimple1516e.xml" }**

**]**

**}**

### Example of Test Suite configuration file

TC\_001\_Files\_Check

### Example of result file

-----------------------------------------------------------

Federate name :TestFederate

-----------------------------------------------------------

FOM Files :D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\MAKsimple1516e.xml

Result parsing FOM Files : Succeed

-----------------------------------------------------------

SOM Files :D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_CS\_Verification\SOM\_MAKsimple1516e.xml

Result parsing SOM Files : Succeed

-----------------------------------------------------------

Result test SOM include in FOM : Succeed

-----------------------------------------------------------

Result test sharing property between SOM and FOM : Succeed

-----------------------------------------------------------

Result test Rules : Succeed

## HLA Declaration Management ETC

### Example of TcParam.json SuT configuration file

**{**

**"federationName" : "FEDERATION\_TEST"**

**"sutName" : "TestFederate"**

**"rtiAddress" : "127.0.0.1"**

**"rtiPort" : "8989"**

**"testDuration" : "60"**

**"resultDirectory" : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Declaration"**

**"fomFiles" : [**

**{ "fileName"    : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Declaration\MAKsimple1516e.xml" }**

**]**

**"somFiles" : [**

**{ "fileName"    : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Declaration\SOM\_MAKsimple1516e.xml" }**

**]**

**}**

### Example of Test Suite configuration file

TC\_001\_Publish\_Subscribe\_Check

### Example of result file

###########################################################

Certification results "TestFederate"

Date : 2017\_04\_05\_15h24m21s

Results for the data and the interactions certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

Counter and send status Counter and receive status

Objets

HLAobjectRoot

BaseEntity

AccelerationVector 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

DeadReckoningAlgorithm 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

Orientation 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

WorldLocation 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

VelocityVector 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

Interactions

HLAinteractionRoot

WeaponFire

EventIdentifier 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FireControlSolutionRange 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FireMissionIndex 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FiringLocation 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FiringObjectIdentifier 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

FuseType 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

InitialVelocityVector 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

MunitionObjectIdentifier 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

MunitionType 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

QuantityFired 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

RateOfFire 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

TargetObjectIdentifier 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

WarheadType 2 R : ExpectedSeen (D : ExpectedNotSeen) 2 R : ExpectedSeen (D : ExpectedNotSeen)

## HLA Object Management ETC

### Example of TcParam.json SuT configuration file

**{**

**"federationName" : "FEDERATION\_TEST"**

**"sutName" : "TestFederate"**

**"rtiAddress" : "127.0.0.1"**

**"rtiPort" : "8989"**

**"testDuration" : "60"**

**"resultDirectory" : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Object"**

**"fomFiles" : [**

**{ "fileName"    : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Object\MAKsimple1516e.xml" }**

**]**

**"somFiles" : [**

**{ "fileName"    : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Object\SOM\_MAKsimple1516e.xml" }**

**]**

**}**

### Example of Test Suite configuration file

TC\_001\_Object\_Interaction\_Check

### Example of result file

###########################################################

Certification results "TestFederate"

Date : 2017\_04\_05\_15h30m41s

Results for the data and the interactions certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

Counter and send status Counter and receive status

Objets

HLAobjectRoot

BaseEntity

AccelerationVector 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

DeadReckoningAlgorithm 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

Orientation 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

WorldLocation 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

VelocityVector 30 R : ExpectedSeen (D : ExpectedNotSeen) 26 R : ExpectedSeen (D : ExpectedNotSeen)

Interactions

HLAinteractionRoot

WeaponFire

EventIdentifier 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FireControlSolutionRange 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FireMissionIndex 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FiringLocation 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FiringObjectIdentifier 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

FuseType 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

InitialVelocityVector 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

MunitionObjectIdentifier 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

MunitionType 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

QuantityFired 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

RateOfFire 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

TargetObjectIdentifier 6 R : ExpectedSeen (D : ExpectedNotSeen) 5 R : ExpectedSeen (D : ExpectedNotSeen)

## HLA Services Verification ETC

### Example of TcParam.json configuration file

**{**

**"federationName" : "FEDERATION\_TEST"**

**"sutName" : "TestFederate"**

**"rtiAddress" : "127.0.0.1"**

**"rtiPort" : "8989"**

**"testDuration" : "60"**

**"resultDirectory" : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Services"**

**"fomFiles" : [**

**{ "fileName"    : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Services\MAKsimple1516e.xml" }**

**]**

**"somFiles" : [**

**{ "fileName"    : "D:\Users\HLA\Documents\GitHub\ETC\_FRA\_Config\IVCTsut\11\_SOM\_Ok\_FOM\_Ok\TS\_HLA\_Services\SOM\_MAKsimple1516e.xml" }**

**]**

**}**

### Example of Test Suite configuration file

TC\_001\_Services\_Check

### Example of result file

###########################################################

Certification results "TestFederate"

Date : 2017\_11\_20\_16h28m31s

Results for the services certificated

The columns "State reception" and "State sending" use the following marking :

"R" for "Result" and "D" for "Declaration at start from the SOM"

###########################################################

Counter and services status

Services HLA 1516-2010

Federation management

Connect 1 R : ExpectedSeen (D : ExpectedNotSeen)

Disconnect 1 R : ExpectedSeen (D : ExpectedNotSeen)

Connection Lost 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Create Federation Execution 1 R : ExpectedSeen (D : ExpectedNotSeen)

Destroy Federation Execution 1 R : ExpectedSeen (D : ExpectedNotSeen)

List Federation Executions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Report Federation Executions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Join Federation Execution 1 R : ExpectedSeen (D : ExpectedNotSeen)

Resign Federation Execution 1 R : ExpectedSeen (D : ExpectedNotSeen)

Register Federation Synchronization Point 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Synchronization Point Registration 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Announce Synchronization Point 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Synchronization Point Achieved 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Synchronized 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Federation Save 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Initiate Federate Save 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federate Save Begun 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federate Save Complete 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Saved 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Abort Federation Save 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Federation Save Status 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Save Status Response 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Federation Restore 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Federation Restoration Request 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Restore Begun 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Initiate Federate Restore 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federate Restore Complete 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Restored 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Abort Federation Restore 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Federation Restore Status 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Federation Restore Status Response 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Declaration management

Publish Object Class Attributes 1 R : ExpectedSeen (D : ExpectedNotSeen)

Unpublish Object Class Attributes 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Publish Interaction Class 1 R : ExpectedSeen (D : ExpectedNotSeen)

Unpublish Interaction Class 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Subscribe Object Class Attributes 1 R : ExpectedSeen (D : ExpectedNotSeen)

Unsubscribe Object Class Attributes 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Subscribe Interaction Class 1 R : ExpectedSeen (D : ExpectedNotSeen)

Unsubscribe Interaction Class 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Start Registration For Object Class 1 R : ExpectedSeen (D : ExpectedNotSeen)

Stop Registration For Object Class 1 R : ExpectedSeen (D : ExpectedNotSeen)

Turn Interactions On 1 R : ExpectedSeen (D : ExpectedNotSeen)

Turn Interactions Off 1 R : ExpectedSeen (D : ExpectedNotSeen)

Object management

Reserve Object Instance Name 1 R : ExpectedSeen (D : ExpectedNotSeen)

Object Instance Name Reserved 1 R : ExpectedSeen (D : ExpectedNotSeen)

Release Object Instance Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Reserve Multiple Object Instance Names 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Multiple Object Instance Names Reserved 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Release Multiple Object Instance Names 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Register Object Instance 1 R : ExpectedSeen (D : ExpectedNotSeen)

Discover Object Instance 1 R : ExpectedSeen (D : ExpectedNotSeen)

Update Attribute Values 15 R : ExpectedSeen (D : ExpectedNotSeen)

Reflect Attribute Values 11 R : ExpectedSeen (D : ExpectedNotSeen)

Send Interaction 3 R : ExpectedSeen (D : ExpectedNotSeen)

Receive Interaction 3 R : ExpectedSeen (D : ExpectedNotSeen)

Delete Object Instance 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Remove Object Instance 1 R : ExpectedSeen (D : ExpectedNotSeen)

Local Delete Object Instance 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attributes In Scope 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attributes Out Of Scope 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Attribute Value Update 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Provide Attribute Value Update 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Turn Updates On For Object Instance 1 R : ExpectedSeen (D : ExpectedNotSeen)

Turn Updates Off For Object Instance 1 R : ExpectedSeen (D : ExpectedNotSeen)

Request Attribute Transportation Type Change 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Attribute Transportation Type Change 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Attribute Transportation Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Report Attribute Transportation Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Interaction Transportation Type Change 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Interaction Transportation Type Change 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Interaction Transportation Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Report Interaction Transportation Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Ownership management

Unconditional Attribute Ownership Divestiture 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Negotiated Attribute Ownership Divestiture 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Attribute Ownership Assumption 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Divestiture Confirmation 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Divestiture 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Acquisition Notification 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Acquisition 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Acquisition If Available 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Unavailable 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Attribute Ownership Release 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Release Denied 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Attribute Ownership Divestiture If Wanted 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Cancel Negotiated Attribute Ownership Divestiture 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Cancel Attribute Ownership Acquisition 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Confirm Attribute Ownership Acquisition Cancellation 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Attribute Ownership 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Inform Attribute Ownership 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Is Attribute Owned By Federate 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time management

Enable Time Regulation 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Regulation Enabled 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Time Regulation 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Time Constrained 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Constrained Enabled 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Time Constrained 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Advance Request 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Advance Request Available 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Next Message Request 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Next Message Request Available 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Flush Queue 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Time Advance Grant 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Asynchronous Delivery 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Asynchronous Delivery 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query GALT 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Logical Time 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query LITS 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Modify Lookahead 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Query Lookahead 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Retract 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Retraction 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Change Attribute Order Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Change Interaction Order Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Data distribution management

Create Region 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Commit Region Modifications 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Delete Region 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Register Object Instance With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Associate Regions For Updates 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Unassociate Regions For Updates 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Subscribe Object Class Attributes With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Unsubscribe Object Class Attributes With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Subscribe Interaction Class With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Unsubscribe Interaction Class With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Send Interaction With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Request Attribute Value Update With Regions 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Support Services

Get Automatic Resign Directive 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Set Automatic Resign Directive 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Federate Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Federate Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Object Class Handle 1 R : ExpectedSeen (D : ExpectedNotSeen)

Get Object Class Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Known Object Class Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Object Instance Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Object Instance Name 1 R : ExpectedSeen (D : ExpectedNotSeen)

Get Attribute Handle 5 R : ExpectedSeen (D : ExpectedNotSeen)

Get Attribute Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Update Rate Value 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Update Rate Value For Attribute 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Interaction Class Handle 1 R : ExpectedSeen (D : ExpectedNotSeen)

Get Interaction Class Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Parameter Handle 13 R : ExpectedSeen (D : ExpectedNotSeen)

Get Parameter Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Order Type 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Order Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Transportation Type Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Transportation Type Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Available Dimensions For Class Attribute 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Available Dimensions For Interaction Class 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Dimension Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Dimension Name 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Dimension Upper Bound 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Dimension Handle Set 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Get Range Bounds 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Set Range Bounds 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Normalize Federate Handle 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Normalize Service Group 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Object Class Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Object Class Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Attribute Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Attribute Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Attribute Scope Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Attribute Scope Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Enable Interaction Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Interaction Relevance Advisory Switch 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Evoke Callback 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Evoke Multiple Callbacks 23 R : ExpectedSeen (D : ExpectedNotSeen)

Enable Callbacks 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

Disable Callbacks 0 R : NotExpectedNotSeen (D : NotExpectedNotSeen)

## Synthetic presentation of IVCT

|  |  |
| --- | --- |
| alert, caution, exclamation, exclamation mark, sign, triangle, warning icon | The current annex describes version 0.3.1 of IVCT, which is the last one working using command line, and delivered without any Web interface. |

|  |  |
| --- | --- |
| alert, caution, exclamation, exclamation mark, sign, triangle, warning icon | The current annex describes required actions to use IVCT under Windows environment only. |

### Composition

Technically speaking, IVCT comprises the following components:

* Apache Active MQ, JMS communication backbone between others Java components of IVCT
* LogSink, Java application that corresponds to a log console which centralized and displayed all messages
* UI, Java application sending commands to test cases (ETC) execution engine
* JMSTestRunner, Java application that corresponds to test cases execution engine. That application is not visible from a user perspective, it runs in background

### Installation

|  |  |
| --- | --- |
| alert, caution, exclamation, exclamation mark, sign, triangle, warning icon | An Internet connection is required during installation. There are indeed some elements to download from Internet. |

The first required action is to download and install a version of Apache Active MQ (<http://activemq.apache.org/>). The version used during development was 5.13.0.

Then, it is required to get a local copy of the source code of IVCT hosted under GitHub at the following address: <https://github.com/MSG134/IVCT_Framework>.

To do so, a good way is to have a GitHub account (create it if required) and to use GitHub Desktop tool (<https://desktop.github.com/>) which is a graphic tool to manage a GitHub repository.

Nevertheless, any tool that allows to create a local copy of an IVCT repository hosted under GitHub may be used.

By the time the source code is available, the following environment variables must be defined:

* IVCT\_HOME: Installation directory of IVCT, i.e. local copy of the source code. Example: D:\RMT\git\IVCT\_Framework
* IVCT\_CONF: Directory of ETC configuration files. Example: D:\RMT\git\TS\_CS\_Verification\ETC\_FRA\_Config
* IVCT\_TS\_HOME: Directory where ETC are extracted (TS\_CS\_Verification, TS\_CS\_Verification, TS\_HLA\_Declaration, TS\_HLA\_Object, TS\_HLA\_Services directories). Example: D:\RMT\git

### Compilation

|  |  |
| --- | --- |
| alert, caution, exclamation, exclamation mark, sign, triangle, warning icon | An Internet connection is required during compilation. There are indeed some elements to download from Internet. |

The compilation is performed using Gradle build environment which requires no installation.

To compile IVCT:

* Go to the following directory: %IVCT\_HOME%
* Launch the following command: gradlew install

Compilation of all sub projects is automatically done.

By the time the compilation is over, it is required to extract zip files that correspond to LogSink et UI executables (UI component also includes JMSTestRunner component). To do so:

* Got to %IVCT\_HOME%\LogSink\build\distributions directory and extract LogSink-0.3.1.zip file. That will create directory LogSink-0.3.1
* Got to %IVCT\_HOME%\UI\build\distributions directory and extract UI-0.3.1.zip file. That will create directory UI-0.3.1

Note: To clean all generated elements, just launch the following command: gradlew clean.

### Execution

The first required action is to start Apache Active MQ. To do so, from the installation directory:

* Go to bin directory
* Launch the following command: activemq.bat start

|  |  |
| --- | --- |
| **http://www.atome77.com/images/icones/icone-information.jpg** | Apache Active MQ is then started once and for all, it is not required to worry about it anymore. |

Then, LogSink must be started. To do so:

* Got to the following directory: %IVCT\_HOME%\LogSink\build\distributions\LogSink-0.3.1\bin\
* Launch the following command: LogSink.bat

|  |  |
| --- | --- |
| alert, caution, exclamation, exclamation mark, sign, triangle, warning icon | Each time UI component is stopped, LogSink component is also automatically stopped. It is then required to systematically start again LogSink component before starting again UI component. |

Before starting UI component, the corresponding launch script must be configured to designate ETC that will be used. To do so:

* Got to the following directory: %IVCT\_HOME%\UI\build\distributions\UI-0.3.1\bin\
* Edit UI.bat file
* Modify CLASSPATH to designate the directory that contains ETC libraries (cf. §5.2.2). For instance, to use TS\_HLA\_Declaration ETC with Pitch RTI, here is the value that must be assigned to CLASSPATH:

set CLASSPATH=%APP\_HOME%\lib\UI-0.3.1.jar;C:\Program Files\prti1516e\lib\prti1516e.jar;%IVCT\_TS\_HOME%\TS\_HLA\_Declaration\TS\_HLA\_Declaration\build\distributions\TS\_HLA\_Declaration-1.0.0\lib\\*

By the time the configuration is done, UI component is started as follows:

* Got to the following directory: %IVCT\_HOME%\UI\build\distributions\UI-0.3.1\bin\
* Launch the following command: UI.bat

In the command line window that opens, it is then possible to enter commands to start ETC. Cf. §6 for more details.

In addition to UI component, JMSTestRunner component is started in background (new Java process). That last component executes the ETC.

## rtiSimple4etcFra test federate

### Structure

The test federate is delivered as zip file named rtiSimple4etcFra.zip. It is an Eclipse project.

### Installation

Unzip rtiSimple4etcFra.zip. It will create an rtiSimple4etcFra directory.

In Eclipse, select “File / Import… / General / Existing projects into workspace” and then choose rtiSimple4etcFra directory.

On rtiSimple4etcFra project, select “Build Path… / Configure Build Path…”. In the “Libraries” tab, select “Add External JARs…” and select the libRTI installed by Pitch or MÄK RTI.

### Configuration

In Eclipse, select “Run / Run configurations…”.

Create a new Java Application configuration, select rtiSimple4etcFra as the project and test\_ETC\_FRA.rtiSimple4etcFra as the Main class in the “Main” tab. Enter the name of the federate (for instance: TestFederate) as “Program arguments:” in “Arguments” tab.

### Usage

Just run the configuration previously created.

To change the name of the federate, just change the program argument.

|  |  |
| --- | --- |
| **alert, caution, exclamation, exclamation mark, sign, triangle, warning icon** | You can start the test federate as many times as you wish, but you need to use a different name of federate each time. |

1. Some precisions required for using IVCT with VT MÄK RTI [↑](#footnote-ref-1)