|  |  |
| --- | --- |
|  |  |

EPTF CLL StatHandler, User Guide

Contents

[1 Introduction 2](#_Toc298926130)

[1.1 Revision history 2](#_Toc298926131)

[1.2 About this Document 2](#_Toc298926132)

[1.2.1 How to Read this Document 2](#_Toc298926133)

[1.2.2 References 2](#_Toc298926134)

[1.2.3 Abbreviations 3](#_Toc298926135)

[1.2.4 Terminology 3](#_Toc298926136)

[1.3 System Requirements 3](#_Toc298926137)

[2 StatHandler 4](#_Toc298926138)

[2.1 Overview 4](#_Toc298926139)

[2.2 Description of files in this feature 4](#_Toc298926140)

[2.3 Description of required files from other features 5](#_Toc298926141)

[2.4 Installation 5](#_Toc298926142)

[2.5 Configuration 6](#_Toc298926143)

[2.5.1 Module Parameters of the StatHandler Feature 6](#_Toc298926144)

[3 Error messages 6](#_Toc298926145)

[4 Warning messages 7](#_Toc298926146)

[5 Examples 12](#_Toc298926147)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2009-02-16 | PA1 | First draft version | EGBOTAT |
| 2009-02-26 | PA2 | Modifications after review | EGBOTAT |
| 2009-04-22 | PA3 | Added support for registering StatMeasure statistics | EGBOTAT |
| 2009-06-02 | PA4 | PercentileP statistics added | EBENMOL |
| 2011-07-20 | PB1 | DataSource Warning messages | EJNOSVN |

## About this Document

### How to Read this Document

This is the User Guide for the StatHandler of the Ericsson Performance Test Framework (TitanSim), Core Load Library (CLL). TitanSim CLL is developed for the TTCN-3 [1] Toolset with TITAN [2]. This document should be read together with the Function Description of the StatHandler feature [6]. For more information on the TitanSim CLL please consult the Product Revision Information [3], the Users Guide [4] and the Function Specification [5] of the TitanSim.

### References

1. ETSI ES 201 873-1 v3.2.1 (2007-02)  
   The Testing and Test Control Notation version 3. Part 1: Core Language
2. 1/198 17-CRL 113 200 Uen  
   User Guide for the TITAN TTCN-3 Test Executor
3. 109 21-CNL 113 512-2 Uen   
   TitanSim CLL for TTCN-3 toolset with TITAN, Product Revision Information
4. 155 17-CNL 113 512 Uen   
   TitanSim CLL for TTCN-3 toolset with TITAN, Function Specification
5. 198 17-CNL 113 512 Uen  
   TitanSim CLL for TTCN-3 toolset with TITAN, User Guide
6. 27/155 16-CNL 113 512 Uen  
   EPTF CLL StatHandler, Function Description
7. 18/198 17-CNL 113 512 Uen  
   EPTF CLL Variable, User Guide
8. TitanSim CLL for TTCN-3 toolset with TITAN, Reference Guide  
   <http://ttcn.ericsson.se/products/libraries.shtml>

### Abbreviations

CLL Core Load Library

EPTF Ericsson Load Test Framework, formerly TITAN Load Test Framework

GUI Graphics User Interface

TitanSim Ericsson Load Test Framework, formerly TITAN Load Test Framework

TTCN-3 Testing and Test Control Notation version 3 [1]

### Terminology

*TitanSim Core (Load) Library(CLL)* is that part of the TitanSim software that is totally project independent. (I.e., which is not protocol-, or application-dependent). The TitanSim CLL is to be supplied and supported by the TCC organization. Any TitanSim CLL development is to be funded centrally by Ericsson

*Statistics* are temporal statistics of EPTF Variables measured by the StatMeasure feature. The value of a statistics is calculated from a single EPTF Variable, as it changes throughout the time of execution.

*Aggregated Stat* are EPTF Variables calculated from other EPTF Variables (data sources). The word stat or statistics refers aggregated statistics throughout this document.

*Auxiliary Statistics* is an aggregated statistics that is not declared by the user of StatHandler directly, but as a result of declaring another stat. That statistics uses the calculated value of the auxiliary stat in its calculations.

*Auxiliary Variable* An EPTF Variable used by an aggregated statistics to store some data.

*Provider Variable* The Variable that affects the value of a given EPTF Variable.

## System Requirements

In order to use the StatHandler feature the system requirements listed in TitanSim CLL User Guide [5] should be fulfilled.

# StatHandler

## Overview

The EPTF StatHandler feature makes it possible to collect global, aggregated statistics via the existing EPTF Variable (see [7]) interface.

The StatHandler feature consists of master and client components. Users of the StatHandler feature can declare aggregated statistics with a certain statistics method (e.g. Sum), and register local or third party data sources (EPTF Variables).

## Description of files in this feature

The EPTF CLL StatHandler API includes the following files:

* EPTF\_CLL\_StatHandlerClient\_Definitions.ttcn: This TTCN-3 module contains common type definitions that should be used in all StatHandlerClient Components.
* EPTF\_CLL\_StatHandlerClient\_Functions.ttcn: This TTCN-3 module contains the implementation of the StatHandlerClient functions.
* EPTF\_CLL\_StatHandlerLogging\_Definitions.ttcn: This TTCN-3 module contains common type definitions used for logging in the StatHandler feature.
* EPTF\_CLL\_StatHandlerLogging\_Functions.ttcn: This TTCN-3 module contains the implementation of the common logging functions used in the StatHandler feature.
* EPTF\_CLL\_StatHandlerUI\_Definitions.ttcn: This TTCN-3 module contains common type definitions used by the UI handling part of the StatHandler feature.
* EPTF\_CLL\_StatHandlerUI\_Functions.ttcn: This TTCN-3 module contains the implementation of the UI handling part of the StatHandler feature.
* EPTF\_CLL\_StatHandler\_Definitions.ttcn: This TTCN-3 module contains common type definitions that should be used in all StatHandler master components.
* EPTF\_CLL\_StatHandler\_Functions.ttcn: This TTCN-3 module contains the implementation of the StatHandler functions.
* EPTF\_CLL\_StatHandler\_DSFunctions.ttcn: This TTCN-3 module contains the implementation of the StatHandler DataSource functionality.

## Description of required files from other features

The StatHandler feature is part of the TitanSim EPTF Core Load Library (CLL). It relies on several features of the CLL. To use the StatHandler, the user has to obtain the respective files from the following features:

* Base
* Common
* FreeBusyQueue
* HashMap
* Logging
* Semaphore
* StatMeasure
* UIHandler
* Variable
* DataSource

## Installation

Since EPTF\_CLL\_StatHandler is used as a part of the TTCN-3 test environment this requires TTCN-3 Test Executor to be installed before any operation of these functions. For more details on the installation of TTCN-3 Test Executor see the relevant section of [2].

If not otherwise noted in the respective sections, the following are needed to use EPTF\_CLL\_StatHandler:

* Copy the files listed in section [2.2, 2.3] to the directory of the test suite or create symbolic links to them.
* Import the StatHandler demo or write your own application using StatHandler.
* Create Makefile or modify the existing one. For more details see the relevant section of [2].
* Edit the config file according to your needs, see following section [2.5].

## Configuration

The executable test program behaviour is determined via the run-time configuration file. This is a simple text file, which contains various sections. The usual suffix of configuration files is .cfg. For further information on the configuration file see [2].

This StatHandler feature defines TTCN-3 module parameters as defined in [2] clause 4. Actual values of these parameters – when no default value or a different from the default actual value wished to be used – shall be given in the [MODULE\_PARAMETERS] section of the configuration file.

### Module Parameters of the StatHandler Feature

**tsp\_EPTF\_StatHandler\_nameSeparator**

This charstring type module parameter is defined in module EPTF\_CLL\_StatHandler\_Definitions. It is used to define the separator string used for name creation in the feature. Its default value is **“.”**.

**tsp\_debug\_EPTF\_StatHandler**

This boolean type module parameter is defined in module EPTF\_CLL\_StatHandler\_Functions. It can be used for debugging purposes. Its default value is **false**.

**tsp\_debug\_StatHandlerClient\_autoDisconnectTimeout**

This float type module parameter is defined in module EPTF\_CLL\_StatHandlerClient\_Definitions. It defines the inactivity time after which the StatHandlerClient will disconnect from all connected StatHandler master components to free port resources after StatHandler has been set up. Its default value is **5.0** for 5 seconds.  
The disconnect operation is only performed on the StatHandler management port, the collection of statistics will continue and the reset button will be handled via the EPTF Variable interface. The client re-connects to the server if it needs to send a new management message to it. It is recommended to set this value to the same as tsp\_EPTF\_maxRunningTime (which has a default value of 2147483.0) if an automatic disconnection is not preferred.

**tsp\_debug\_EPTF\_StatHandlerClient**

This boolean type module parameter is defined in module EPTF\_CLL\_StatHandlerClient\_Functions. It can be used for debugging purposes. Its default value is **false**.

# Error messages

Please note, that besides the below described error messages, error messages shown in [2] or those of other used features or product may also appear.

**f\_EPTF\_StatHandler\_setMethodForStat: stat method <statMethod> for integer stat <statName> is not supported.**

The method <statMethod> that was set for statistics <statName> is not supported for integer statistics.

**f\_EPTF\_StatHandler\_setMethodForStat: Cannot set “<statMethod>” stat method for stat <statName> := <statResetValue>**

The method <statMethod> cannot be set for statistics <statName> as the type of the reset value <statResetValue> is not supported by statHandler.

**f\_EPTF\_StatHandler\_createAuxData\_Percentile95: Could not create aux stat <auxStatName> for Percentile95.**

The auxiliary statistics with name <auxStatName> could not be declared.

**f\_EPTF\_StatHandler\_createAuxData\_PercentileP: Could not create aux stat <auxStatName> for PercentileP.**

The auxiliary statistics with name <auxStatName> could not be declared.

**as\_EPTF\_StatHandlerClient\_msgHandler: unexpected message received from <sender>**

The <sender> component sent an unexpected message to the StatHandlerClient.

**f\_EPTF\_StatHandlerUI\_initGUI: cannot add element to gui (<returnValue>): <xul>**

The element described by the <xul> data structure could not be added to the GUI. The result of the operation is <returnValue>.

# Warning messages

Please note, that besides the below described warning messages, warning messages shown in [2] or those of other used features or product may also appear.

**f\_EPTF\_StatHandler\_declareStat: redeclaration of <statName> with different properties.**

The statistics with name <statName> was re-declared with a different statistics method or reset value.

**f\_EPTF\_StatHandler\_registerStat: called while reset is in progress**

A data source was attempted to be registered during reset progress.

**f\_EPTF\_StatHandler\_registerStat: provider variable list is empty**

No provider variables were specified for data source registration.

**f\_EPTF\_StatHandler\_registerStat: no such statistics: <statName>**

The statistics with name <statName> does not exist.

**f\_EPTF\_StatHandler\_registerStat\_Single: provider variable list for “<statMethod>” statistics “<statName>” has multiple elements, expected single numerical variable.**

The data source registered should consist of a single element.

**f\_EPTF\_StatHandler\_registerStat\_Single: invalid source type for stat <statName>**

The type of the data source variable differs from the type of the statistics.

**f\_EPTF\_StatHandler\_registerStat\_Mean: provider variable list for “<statMethod>” statistics “<statName>” needs two elements: local mean and number of samples.**

The data source registered should consist of a local mean and a number of samples variable.

**f\_EPTF\_StatHandler\_registerStat\_Mean: invalid source types for stat <statName>**

Provider variable “local mean” should be float, “number of samples” should be integer type.

**f\_EPTF\_StatHandler\_registerStat\_StandardDev: provider variable list for “<statMethod>” statistics “<statName>” needs three elements: local mean, number of samples and S.**

The data source registered should consist of a local mean, a number of samples and a “S” variable.

**f\_EPTF\_StatHandler\_registerStat\_StandardDev: invalid source type for stat <statName>**

Provider variable “local mean” should be float, “number of samples” should be integer and “S” should be float type.

**f\_EPTF\_StatHandler\_registerStat\_GlobalAverage: provider variable list for “<statMethod>” statistics “<statName>” needs two elements: sum and element number.**

The data source registered should consist of a sum and a element number variable.

**f\_EPTF\_StatHandler\_registerStat\_GlobalAverage: invalid source type for stat <statName>**

Provider variable “sum” should be float, “number of samples” should be integer type.

**f\_EPTF\_StatHandler\_registerStat\_Density: provider variable list for “<statMethod>” statistics “<statName>” needs two elements: density and boundaries.**

The data source registered should consist of a density and a boundaries variable.

**f\_EPTF\_StatHandler\_registerStat\_Density: invalid source type for stat <statName>**

Provider variable “density” should be integer list, “boundaries” should be float list type.

**f\_EPTF\_StatHandler\_registerStat\_Density: boundaries <sourceBoundaries> of source differs from boundaries <statBoundaries> stored for the stat <statName>**

The boundaries part of the data source is not the same as the boundaries of the statistics. The latter is assigned the value of the first registered data source.

**f\_EPTF\_StatHandler\_registerStat\_Percentile95: provider variable list for “<statMethod>” statistics “<statName>” needs three elements: density, boundaries and max.**

The data source registered should consist of a density, a boundaries and a max variable.

**f\_EPTF\_StatHandler\_registerStat\_Percentile95: could not register source for aux stat <auxStatName>**

The source could not be registered for the auxiliary statistics.

**f\_EPTF\_StatHandler\_registerStat\_PercentileP: provider variable list for “<statMethod>” statistics “<statName>” needs four elements: density, boundaries, max and pValue.**

The data source registered should consist of a density, a boundaries, a max and a pValue variable.

**f\_EPTF\_StatHandler\_registerStat\_PercentileP: could not register source for aux stat <auxStatName>**

The source could not be registered for the auxiliary statistics.

**f\_EPTF\_StatHandler\_registerStat\_PercentileP: pValue <pValue> of source differs from pValue <pValue>** **stored for the stat**

The pValue of the data source variable differs from the pValue of the statistics.

**f\_EPTF\_StatHandler\_boundariesPostproc: “boundaries” part of source <sourceIndex> changed for stat <statName>**

The boundaries part of the data source was changed by the providing component.

**f\_EPTF\_StatHandler\_boundariesPostproc: source providing Variable with index <index> not found in stat <statName>**

The source which boundaries was changed is not in the source list of statistics <statName>.

**as\_EPTF\_StatHandler\_behavior: unexpected message received from <client>**

Client sent an unexpected message to the StatHandler.

**f\_EPTF\_StatHandler\_setMethodForStat: Stat method “<method>” for integer stat “<statName>” is unsupported or not yet supported.**

Stat method passed to declareStat is undefined.

**f\_EPTF\_StatHandler\_setMethodForStat: Stat method “<method>” for float stat “<statName>” is unsupported or not yet supported.**

Stat method passed to declareStat is undefined.

**f\_EPTF\_StatHandler\_setMethodForStat: Stat method “<method>” for float-list stat “<statName>” is unsupported or not yet supported.**

Stat method passed to declareStat is undefined.

**f\_EPTF\_StatHandlerClient\_registerResetFunction: cannot register null function reference.**

The function reference specified is null.

**f\_EPTF\_StatHandlerClient\_registerResetFunction: re-registration of reset function <function>**

The function reference is already registered.

**f\_EPTF\_StatHandlerClient\_registerResetFunction: cannot deregister null function reference.**

The function reference specified is null.

**as\_EPTF\_StatHandlerClient\_autoDisconnect: restarting auto disconnect timer, pending messages: <n>**

The automatic disconnect timer is restarted because there are <n> pending messages.

f\_EPTF\_StatHandlerUI\_addLocalResetButton: cannot add element to gui: <xul>

The element with data structure <xul> could not be added to the GUI.

**as\_EPTF\_StatHandlerUI\_behavior: statistics not found: <statName>**

The statistics <statName> referred by a bind-to-widget operation was not declared.

**f\_EPTF\_StatHandler\_DSProcessData: Unhandled element:**

This warning message is reported when the DataSource functionality is used and the current data element is unknown for the feature.

**f\_EPTF\_StatHandler\_DSProcessData: Invalid iterator or externalData or parameter:**

This warning message is reported when the DataSource functionality is used and the something with the request was wrong.

**f\_EPTF\_StatHandler\_getParams: Parameters are not correct:**

This warning message is reported when the DataSource functionality is used and the parameters of the current data element are not correct.

**f\_EPTF\_StatHandler\_getParams: Too few parameter is given:**

This warning message is reported when the DataSource functionality is used and the parameters of the current data element are less than required.

**f\_EPTF\_StatHandler\_getParams: Too many parameters are given:**

This warning message is reported when the DataSource functionality is used and the parameters of the current data element are more than required.

# Examples

The “demo” directory of the deliverable contains the following examples:

* EPTF\_StatHandler\_Demo.cfg
* EPTF\_StatHandler\_Demo.prj
* EPTF\_StatHandler\_Demo.ttcn