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

EPTF CLL Load Regulator, Function Description

Contents

[1 Introduction 2](#_Toc298764458)

[1.1 Revision history 2](#_Toc298764459)

[1.2 How to Read this Document 2](#_Toc298764460)

[1.3 References 2](#_Toc298764461)

[1.4 Scope 2](#_Toc298764462)

[1.5 Recommended way of reading 3](#_Toc298764463)

[1.6 Typographical conventions 3](#_Toc298764464)

[1.7 Abbreviations 3](#_Toc298764465)

[1.8 Terminology 3](#_Toc298764466)

[2 General Description 3](#_Toc298764467)

[3 Functional Interface 4](#_Toc298764468)

[3.1 Naming Conventions 4](#_Toc298764469)

[3.2 Public Functions 4](#_Toc298764470)

[3.2.1 Initialization 4](#_Toc298764471)

[3.2.2 Cleanup 5](#_Toc298764472)

[3.2.3 Calculating the Next CPS 5](#_Toc298764473)

[3.2.4 Behavior Altstep 5](#_Toc298764474)

[3.2.5 Exponential Smoothing Function 5](#_Toc298764475)

[3.3 Summary Table of all public functions for EPTF Load Regulator 6](#_Toc298764476)

[3.4 UI handling with external data elements 6](#_Toc298764477)

[3.5 Load Regulator UI 6](#_Toc298764478)

[3.5.1 Initialization of LoadRegulatorUI 6](#_Toc298764479)

[3.6 Summary Table of all public functions for EPTF LoadRegulatorUI 7](#_Toc298764480)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2007-11-30 | PA1 | First draft version | EGBOTAT |
| 2007-12-06 | PA2 | Final version after review | EGBOTAT |
| 2008-02-15 | PA3 | Minor changes for bandwidth limited traffic control | EGBOTAT |
| 2009-08-12 | PB1 | New calcCPS method added | ETHJGI |
| 2010-04-06 | PC1 | Init method parameter update | EJNOSVN |
| 2011-07-18 | PD1 | External data elements | ELSZSKU |

## How to Read this Document

This is the Function Description for the Load Regulator of the Ericsson Performance Test Framework (TitanSim), Core Load Library (CLL). TitanSim CLL is developed for the TTCN-3 [1] Toolset with TITAN [2]. For more information on the TitanSim CLL please consult the Product Revision Information [3].

## 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. TitanSim CLL for TTCN-3 toolset with TITAN, Reference Guide  
   <http://ttcn.ericsson.se/products/libraries.shtml>
6. 2/155 16-CNL 113 512 Uen  
   EPTF CLL UIHandler, Function Description

## Scope

This document is to specify the content and functionality of the Load Regulator feature of the TitanSim CLL.

## Recommended way of reading

The readers are supposed to get familiar with the concept and functionalities of TitanSim CLL [4]. They should get familiar with the list of acronyms and the glossary in Section 1.7 and 1.8, respectively.

## Typographical conventions

Important concepts are denoted by *italic* font wherever they are first used in the given context.

## Abbreviations

CLL Core Load Library

EPTF Ericsson Load Test Framework, formerly TITAN Load Test Framework

TitanSim Ericsson Load Test Framework, formerly TITAN Load Test Framework

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

CPS Calls Per Second

SUT System Under Test

## 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

# General Description

This document specifies the Load Regulator feature of the TitanSim CLL.

The EPTF Load Regulator feature makes it possible to regulate the CPS of a load generator based on the load of the SUT.

To be able to use EPTF Load Regulator, the user component should extend the EPTF\_LoadRegulator\_CT component.

# Functional Interface

Apart from this description a cross-linked reference guide for the TitanSim CLL Functions can be reached for on-line reading [5].

## Naming Conventions

All functions have the prefix f\_EPTF\_LoadRegulator eg. f\_EPTF\_LoadRegulator\_behavior().

## Public Functions

### Initialization

Before using the EPTF Load Regulator functions the

f\_EPTF\_LoadRegulator\_init\_CT(pl\_getSutLoad,  
pl\_calcNextCps,  
pl\_postCalcCps,  
pl\_EPTF\_loadRegulator\_measWinSize,  
pl\_EPTF\_loadRegulator\_updateTimeout,  
pl\_EPTF\_loadRegulator\_loadVarianceThreshold,  
pl\_EPTF\_loadregulator\_cpsDelta,  
pl\_EPTF\_loadRegulator\_errorTolerance,  
pl\_EPTF\_loadRegulator\_smoothingFactor   
pl\_dataSource\_compRef )

function must be called. This initializes the EPTF Load Regulator feature.

A function reference has to be passed to the function via the parameter pl\_getSutLoad.

Optionally, a function reference can be passed via the pl\_calcNextCps. This function should calculate the next CPS based on the load. By default, this parameter is ‘null’, and in this case the function f\_EPTF\_LoadRegulator\_calculateNextCps (see 3.2.3) will be used.

A user defined function that is called after updating the CPS can be specified via the optional parameter pl\_postCalcCps. This function can be used e.g. to distribute the calculated CPS to a weighted CPS list.

The following parameters are optional, by default they initialize component variables to module parameters. They can be overridden here. The parameter *pl\_EPTF\_loadRegulator\_measWinSize* sets the measurement window size used in the *calculateNextCps* functions, the *pl\_EPTF\_loadRegulator\_updateTimeout* sets the frequency of regulating, the *pl\_EPTF\_loadRegulator\_loadVarianceThreshold* set the load variance threshold, the *pl\_EPTF\_loadregulator\_cpsDelta* is the maximum value that the CPS value will be changed with, the *pl\_EPTF\_loadRegulator\_errorTolerance* sets the precision of the target load measurement, the *pl\_EPTF\_loadRegulator\_smoothingFactor* sets the amount of smoothing of the measured load values to avoid spikes (0.0: extreme smoothing, 1,0: no smoothing).

### Cleanup

Before stopping the Load Regulator component, the function

f*\_EPTF\_LoadRegulator\_cleanup\_CT()*

should be called.

### Calculating the Next CPS

*f\_EPTF\_LoadRegulator\_calculateNextCps(loadToReach, oldCps)*

This is the default CPS calculation function that is used if the function reference passed to the initialization function is null. It calculates the next CPS from the previous CPS and the last two loads, according to the specified load to reach. The load and load to reach can be CPU usage, or bandwidth in case of bandwidth limited traffic control.

*f\_EPTF\_LoadRegulator\_calculateNextCps\_limitMax(loadToReach, oldCps)*

A variation of the function *f\_EPTF\_LoadRegulator\_calculateNextCps*. It only regulates the CPS if the current load reaches the target load setting. In this case the *f\_EPTF\_LoadRegulator\_calculateNextCps* function is used to decrease the CPS to keep the load stable at the specified maximal load to reach setting. Below that boundary the original CPS is used, no regulation takes place. The original CPS is restored when the CPS that keeps the load at the target load setting would go above the original CPS level. When the regulator is registered into the ExecCtrl by the function *f\_EPTF\_LoadRegulator\_registerToExecCtrl* this is indicated by a blue status LED with text *Auto-off*. The CPS level can only be changed when no regulation takes place via the "EPTF\_ExecCtrl.Regulator.totalValue."&v\_selfName variable or via the RegulatedValue parameter on the ExecCtrl GUI.

### Behavior Altstep

*as\_EPTF\_LoadRegulator\_behavior()*

This altstep is the behavior for the EPTF Load Regulator. It is activated as default by the initialization function.

### Exponential Smoothing Function

*f\_EPTF\_exponentialSmoothingFunction(pl\_measuredLoad, pl\_prevLoad, pl\_smoothingFactor)*

This function is used to smooth the measured load.

## Summary Table of all public functions for EPTF Load Regulator

Table 1. Summary of Load Regulator functions

|  |  |
| --- | --- |
| Function name | Description |
| f\_EPTF\_LoadRegulator\_init\_CT | initializes the Load Regulator component |
| f\_EPTF\_LoadRegulator\_cleanup\_CT | cleanup function for Load Regulator component |
| f\_EPTF\_LoadRegulator\_calculateNextCps | default CPS calculation function |
| as\_EPTF\_LoadRegulator\_behavior | behavior altstep |
| f\_EPTF\_exponentialSmoothingFunction | load smoothing function |

## UI handling with external data elements

According to the customizable GUI concept LoadRegulator feature provides DataSource external data elements in order to create GUI using the XML GUI description. See also [6].

Therefore the LoadRegulator feature provides the following external data elements:

* Enabled
* CpsToReach
* LoadToReach
* CurrentLoad
* LoadIsStable

In order to use these external data elements the pl\_dataSource\_compRef parameter of the init function must be set to the appropriate DataSource server component reference.

## Load Regulator UI

The aim of the EPTF LoadRegulatorUI feature was to make a GUI for the LoadRegulator variables defined in EPTF LoadRegulator feature with the minimal user interaction. However, in the future the suggested way of GUI handling is the customizable GUI using DataSource external data elements.

To be able to use EPTF LoadRegulatorUI the user should extend one LoadRegulatorUI component.

### Initialization of LoadRegulatorUI

For using the EPTF LoadRegulatorUI function the

f\_EPTF\_LoadRegulatorUI\_init\_CT ( pl\_selfName, pl\_getSutLoad, pl\_calcNextCps, pl\_parentid, pl\_UIHandler, pl\_loggingUi\_compRef )

function should be called. This function initializes and starts the main LoadRegulatorUI component, which creates the GUI for LoadRegulatorUI. Input parameters are a name, a getSUTLoad function pointer, a calculateNextCps function pointer, the parent widgetID, where to put LoadRegulator, the UIHandler component of the application and LoggingUI component reference.

## Summary Table of all public functions for EPTF LoadRegulatorUI

Table 1. Summary of LoadRegulatorUI functions

|  |  |
| --- | --- |
| Function name | Description |
| f\_EPTF\_LoadRegulatorUI\_init\_CT | Function to initialize LoadRegulatorUI component |