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

EPTF CLL Central Scheduling, Function Description

Contents

[1 Introduction 2](#_Toc301275947)

[1.1 Revision history 2](#_Toc301275948)

[1.2 How to Read this Document 2](#_Toc301275949)

[1.3 References 2](#_Toc301275950)

[1.4 Scope 2](#_Toc301275951)

[1.5 Recommended way of reading 3](#_Toc301275952)

[1.6 Typographical conventions 3](#_Toc301275953)

[1.7 Abbreviations 3](#_Toc301275954)

[1.8 Terminology 3](#_Toc301275955)

[2 General Description 3](#_Toc301275956)

[3 Functional Interface 4](#_Toc301275957)

[3.1 Naming Conventions 4](#_Toc301275958)

[3.2 General component type hiearchy 5](#_Toc301275959)

[3.3 Public Functions 6](#_Toc301275960)

[3.3.1 Admin Behavior 6](#_Toc301275961)

[3.3.2 Initialization 6](#_Toc301275962)

[3.3.3 Declaring Traffic Cases 7](#_Toc301275963)

[3.3.4 Getting the Component Reference of the Selected LGen 7](#_Toc301275964)

[3.3.5 Handling Execution Result 7](#_Toc301275965)

[3.3.6 Handling Load Generator Status 8](#_Toc301275966)

[3.3.7 Load Generator Index Lookup 8](#_Toc301275967)

[3.3.8 Cleanup 8](#_Toc301275968)

[3.4 Summary Table of all public functions for EPTF Central Scheduling 8](#_Toc301275969)

[3.5 DataSource Client functionality 9](#_Toc301275970)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2008-02-11 | PA1 | First draft version | EGBOTAT |
| 2008-03-17 | PA2 | Added CBE\_Compat description  Added component hierarchy description | EGBOTAT, EGBOZIE |
| 2009-04-06 | PB1 | Removed CBE Compatibility module | EGBOTAT |
| 2011-08-16 | PC1 | Added DataSourceClient functionality | EMIHMIK |

## How to Read this Document

This is the Function Description for the Central Scheduling 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

## Scope

This document is to specify the content and functionality of the Central Scheduling 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

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

# General Description

This document specifies the Central Scheduling feature of the TitanSim CLL.

The EPTF Central Scheduling feature makes it possible to write load test cases using similar methodology than in function test. Load testing means executing several traffic cases in parallel. Such traffic cases might or might not be parameterized. If coordination is necessary then a traffic case parameter database is needed. For efficiency reasons such database must be located within the Central Scheduling Application admin component.

Traffic Cases are handled by Load Generator PTCs, each running one traffic case at a time. The Load Generator is written entirely by the user (for an example, see the demo directory).

To be able to use EPTF Central Scheduling, the user written Central Scheduling Application Admin component should extend the EPTF\_CS\_ApplAdminBase\_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\_CS\_ eg. f\_EPTF\_CS\_Admin\_behavior.

## General component type hiearchy



Figure 1 Component hierarchy of Central Scheduling.

Figure 1 depicts the component extend hierarchy of the Central Scheduling. Users usually extends EPTF\_CS\_ApplAdminBase\_CT with custom database handling resulting in the customizec component type (say) Custom\_Admin\_CT. The similar customization exists for the component type of the traffic generator PTC-s, called “LGen”-s. NOTE: only the directly extended CLL component types are shown here!

## Public Functions

### Admin Behavior

The user extended Central Scheduling Application Admin component must run thefunction f\_EPTF\_CS\_Admin\_behavior(pl\_selfName, pl\_adminIdx, EPTF\_ExecCtrl\_CT pl\_ExecCtrlRef, pl\_EPTF\_CS\_AdminInit, pl\_loadBalancingAvailable) function. The user-specific initialization function must be supplied as an input argument to the standard library behavior. Due to the special rules governing “runs on self” function argument handling, the user might have to wrap

Hence, the user might have to call the standard behavior that “runs on EPTF\_CS\_ApplAdminBase\_CT” with from inside a user-specific behavior function that has a runs-on

Note, that the type of the formal argument of the user-specific initialization function has a function type with “runs on self” declaration , therefore

the be started with a function that calls the following behavior function with the user defined initialization callback function reference:

f\_EPTF\_CS\_Admin\_behavior(pl\_selfName, pl\_adminIdx, pl\_ExecCtrlRef, pl\_EPTF\_CS\_AdminInit, pl\_loadBalancingAvailable)

### Initialization

The behavior function Central Scheduling Application Admin is initialized by calling f\_EPTF\_CS\_Admin\_init\_CT.

This is done automatically by the behavior function. The user defined init function supplied as an argument to the standard behavior is also called in the correct initialization order.

This user defined init function must set the following callback function references defined in component EPTF\_CS\_ApplAdminBase\_CT:

* v\_EPTF\_CS\_addExecBurstElem – mandatory function for adding a traffic case trigger to some buffer
* v\_EPTF\_CS\_sendExecBurst – mandatory function for sending all buffered traffic case triggers to the selected load generator component. The component reference of the selected load generator can be retrieved by calling the function f\_EPTF\_CS\_Admin\_getCompRefOfSelectedLGen.
* v\_EPTF\_CS\_LGenCreate – mandatory function for creating a load generator, connecting it to the Central Scheduling control port (defined in the user extended Admin component) and starting its behavior function.

This user defined init function should set the following callback function references defined in component EPTF\_CS\_ApplAdminBase\_CT:

* v\_EPTF\_CS\_AdminCleanup – optional user defined Admin cleanup function

The user init function must activate its event handler altstep as default. This event handler altstep must handle execution burst result messages, load generator status messages and may optionally handle other user defined messages that can be sent by the load generator to the Admin.

Load generators should call the function f\_EPTF\_CS\_LGenBase\_init\_CT as a first step in their behavior/init function.

### Declaring Traffic Cases

Traffic cases can be declared by calling the function

f\_EPTF\_CS\_Admin\_declareTrafficCase(pl\_tcName, pl\_tcSelector)

This function assigns the integer pl\_tcSelector to the traffic case pl\_tcName.

### Getting the Component Reference of the Selected LGen

The function

f\_EPTF\_CS\_Admin\_getCompRefOfSelectedLGen()

can be used to get the component reference of the load generator selected for the traffic cases. Call this function from the user defined execution burst sender function when sending the execution burst.

### Handling Execution Result

If the message handler of the extended Admin component receives a message containing execution results of a previously sent execution burst from a load generator, it should iterate through this result message and call the function

f\_EPTF\_CS\_handleExecResult(pl\_execResult, pl\_indexInBurst, pl\_LGenIdx)

for each execution result in the message.

After finishing iteration through the result message, the function

f\_EPTF\_CS\_LGenBurstFinished(pl\_LGenIdx)

must be called to make the load generator available for new traffic cases.

### Handling Load Generator Status

If a load generator status message is received, the function

f\_EPTF\_CS\_handleLGenStatusMsg(pl\_enabled, pl\_compRef)

must be called.

### Load Generator Index Lookup

The function

f\_EPTF\_CS\_lookupBySenderAddress(pl\_compRef)

can be used to lookup the load generator index by component reference (sender address in receive operation). It should be used when handling execution result messages.

### Cleanup

Cleanup is done automatically by the behavior function by calling the function

f\_EPTF\_CS\_Admin\_cleanup\_CT()

This function also calls the user defined function reference v\_EPTF\_CS\_AdminCleanup if it is not null.

## Summary Table of all public functions for EPTF Central Scheduling

Table 1. Summary of Central Scheduling functions

|  |  |
| --- | --- |
| Function name | Description |
| f\_EPTF\_CS\_Admin\_behavior | Central Scheduling Application Admin behavior function |
| f\_EPTF\_CS\_Admin\_init\_CT | initialization function |
| f\_EPTF\_CS\_LGenBase\_init\_CT | initialization function for load generators |
| f\_EPTF\_CS\_Admin\_declareTrafficCase | declare a traffic case name-selector assignment |
| f\_EPTF\_CS\_Admin\_getCompRefOfSelectedLGen | Getting the component reference of the selected LGen |
| f\_EPTF\_CS\_handleExecResult | handle a single traffic case execution result |
| f\_EPTF\_CS\_LGenBurstFinished | load generator finished executing traffic cases |
| f\_EPTF\_CS\_handleLGenStatusMsg | handle load generator status (enabled/disabled) message |
| f\_EPTF\_CS\_lookupBySenderAddress | lookup load generator index by component reference |
| f\_EPTF\_CS\_Admin\_cleanup\_CT | cleanup function |

## DataSource Client functionality

The Central Scheduling feature provides DataSourceClient functionality. This means that a CentralScheduling component has several Variables, that are available to use with the help of a DataSource server, for example the UIHandler. For the list of the existing iterators and external data elements see the natural documentation of CentralScheduling.