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

EPTF CLL Semaphore, Function Description

Contents

[1 Introduction 2](#_Toc280093388)

[1.1 Revision history 2](#_Toc280093389)

[1.2 How to Read this Document 2](#_Toc280093390)

[1.3 References 2](#_Toc280093391)

[1.4 Scope 2](#_Toc280093392)

[1.5 Recommended way of reading 2](#_Toc280093393)

[1.6 Typographical conventions 3](#_Toc280093394)

[1.7 Abbreviations 3](#_Toc280093395)

[1.8 Terminology 3](#_Toc280093396)

[2 General Description 3](#_Toc280093397)

[3 Functional Interface 4](#_Toc280093398)

[3.1 Naming Conventions 4](#_Toc280093399)

[3.2 Public Functions 4](#_Toc280093400)

[3.2.1 Initialization 4](#_Toc280093401)

[3.2.2 Create a Semaphore 4](#_Toc280093402)

[3.2.3 Delete a Semaphore 5](#_Toc280093403)

[3.2.4 Unlock a Semaphore 5](#_Toc280093404)

[3.2.5 Lock a Semaphore 5](#_Toc280093405)

[3.2.6 Increase the lock level of a Semaphore 5](#_Toc280093406)

[3.2.7 Decrease the lock level of a Semaphore 5](#_Toc280093407)

[3.2.8 Wait for Unlocking the Semaphore 6](#_Toc280093408)

[3.3 Summary Table of all public functions for EPTF Semaphore 6](#_Toc280093409)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2007-12-05 | PA1 | First draft version | ETHJGI |
| 2007-12-06 | PA2 | Updated after review | ETHJGI |
| 2010-10-20 | PB1 | Multi-lock levels | ETHJGI |
|  |  |  |  |

## How to Read this Document

This is the Function Description for the Semaphore 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 Semaphore 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. Moreover, whenever a concept is mentioned that has a special meaning as described in the Glossary (Section ‎1.8) of this document, then these occurrences are marked with an initial arrow, e.g., 🡪 TitanSim.

## 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]

## 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

*Semaphore* The TTCN-3 way to solve blocking scenarios and perform synchronization

# General Description

This document specifies the Semaphore feature of the TitanSim CLL.

The EPTF Semaphore feature makes it possible to implement blocking functions without using test port receive statements.

There are some situations when a function shall block its execution until some event arrives. This event can be a message received on a test port or a scheduled timer event etc. The drawback of using a simple test port receive statement for waiting for a certain message is that, if some other message arrives on the same test port that cannot be extracted from the queue at the moment, then the message we are waiting for will never got extracted, because it is after the previous message in the event queue. The EPTF Semaphore feature provides a general solution for this problem by using timer events only.

Also, by using EPTF Semaphores it is possible to synchronize events on the component.

The Semaphores can be created on demand. They are identified by an integer ID. Using this ID the created semaphore can be unlocked later somewhere in the code. The function which blocks the execution until the semaphore is unlocked can be used to wait for the required event. When that event arrives, its handler function unlocks the semaphore and the waiting function will continue its operation.

The states of the semaphore can be:

* Locked (lock levels 1 and above)
* Unlocked (lock level 0)

The function that waits for the semaphore to be unlocked blocks its execution until the semaphore is unlocked. After creation the semaphore is in locked state (with lock level 1).

The lock level of the semaphore can be increased and decreased one-by-one. This can be used for example to block execution until all answers for pending requests arrive.

# 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\_Semaphore\_.

## Public Functions

### Initialization

Before using the EPTF Semaphore functions the Semaphore feature should be initialized by calling the function:

f\_EPTF\_Semaphore\_init\_CT(selfName)

### Create a Semaphore

To create a Semaphore, call the function

f\_EPTF\_Semaphore\_new() return integer

The ID of the Semaphore is returned by this function.

The created semaphore is in locked state (lock level is 1).

### Delete a Semaphore

To delete a Semaphore, call the function

f\_EPTF\_Semaphore\_delete(id)

The Semaphore with the given ID is deleted. After that it cannot be locked.

### Unlock a Semaphore

To unlock a Semaphore call:

f\_EPTF\_Semaphore\_unlock(id)

This function sets the locking state of the semaphore to unlocked (lock level 0). The function that waits for unlock will continue its execution after this function is called.

### Lock a Semaphore

To lock a Semaphore call:

f\_EPTF\_Semaphore\_lock(id)

This function sets the locking state to locked (lock level 1 if it was unlocked, or does not change if it was already locked). The function that waits for unlock will block its execution.

### Increase the lock level of a Semaphore

To increase the level of a Semaphore call:

f\_EPTF\_Semaphore\_increaseLock(id)

This function increases the lock level of the semaphore by 1. The function that waits for unlock will block its execution until the lock level decreases to zero.

### Decrease the lock level of a Semaphore

To decrease the lock level of a Semaphore call:

f\_EPTF\_Semaphore\_decreaseLock(id)

This function decreases the lock level of the semaphore by 1. The function that waits for unlock will continue its execution when lock level is decreased to zero.

### Wait for Unlocking the Semaphore

To block the execution until the Semaphore is unlocked call the function:

f\_EPTF\_ Semaphore\_waitForUnlock(id, maxWaitTime, autoDeleteSemaphore) return boolean

This function blocks the execution until the given Semaphore is unlocked. The maximum time to wait in seconds can also be specified. By default it is set to infinity (value -1.0). If the semaphore is not unlocked, the function will return true when the maximum waiting time passes, or never return if maximum waiting time is set to -1. If the semaphore is unlocked, the function will return false. The third parameter autoDeleteSemaphore can be used to enable/disable the deletion of the semaphore automatically before this function exits. By default the semaphore is deleted automatically.

To be able to unlock the Semaphore when this function is called, the event handler of the unlocking event should be activated as default before calling this function, and that event handler shall unlock the semaphore when the requested event arrives.

Note, that this function automatically deletes the Semaphore if the argument autoDeleteSemaphore is set to true.

## Summary Table of all public functions for EPTF Semaphore

Table 1. Summary of EPTF Semaphore functions

| Function name | Description |
| --- | --- |
| [f\_EPTF\_Semaphore\_init\_CT](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Functions-ttcn.html#f_EPTF_Semaphore_init_CT) | Initializes the [EPTF\_Semaphore\_CT](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Definitions-ttcn.html#EPTF_Semaphore_CT) component |
| [f\_EPTF\_Semaphore\_new](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Functions-ttcn.html#f_EPTF_Semaphore_new) | Creates a new semaphore and returns its index. |
| [f\_EPTF\_Semaphore\_delete](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Functions-ttcn.html#f_EPTF_Semaphore_delete) | Deletes the semaphore. |
| [f\_EPTF\_Semaphore\_lock](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Functions-ttcn.html#f_EPTF_Semaphore_lock) | Locks the semaphore. |
| [f\_EPTF\_Semaphore\_unlock](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Functions-ttcn.html#f_EPTF_Semaphore_unlock) | Unlocks the semaphore. |
| [f\_EPTF\_Semaphore\_increaseLock](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Functions-ttcn.html#f_EPTF_Semaphore_lock) | Increases the lock level by one. |
| [f\_EPTF\_Semaphore\_decreaseLock](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Functions-ttcn.html#f_EPTF_Semaphore_lock) | Decreases the lock level by one. |
| [f\_EPTF\_Semaphore\_waitForUnlock](http://mwlx122.eth.ericsson.se:8080/EPTF_CORE_REFERENCE_GUIDE/R2/files/Semaphore/EPTF_CLL_Semaphore_Functions-ttcn.html#f_EPTF_Semaphore_waitForUnlock) | Blocks the execution until the semaphore is unlocked. |