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

EPTF CLL Base, User Guide

Contents

[1 Introduction 2](#_Toc405384300)

[1.1 Revision history 2](#_Toc405384301)

[1.2 About this Document 2](#_Toc405384302)

[1.2.1 How to Read this Document 2](#_Toc405384303)

[1.2.2 References 2](#_Toc405384304)

[1.2.3 Abbreviations 3](#_Toc405384305)

[1.2.4 Terminology 3](#_Toc405384306)

[1.3 System Requirements 3](#_Toc405384307)

[2 Base 3](#_Toc405384308)

[2.1 Overview 3](#_Toc405384309)

[2.2 Description of files in this feature 4](#_Toc405384310)

[2.3 Description of required files from other features 4](#_Toc405384311)

[2.4 Installation 4](#_Toc405384312)

[2.5 Configuration 5](#_Toc405384313)

[2.6 Design rules 5](#_Toc405384314)

[3 Error messages 6](#_Toc405384315)

[4 Warning messages 6](#_Toc405384316)

[5 Examples 6](#_Toc405384317)

[5.1 Demo Module 6](#_Toc405384318)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2007-12-03 | PA1 | First draft version | EGBOTAT |
| 2007-12-05 | PA2 | Final version after review | EGBOTAT |
| 2008-11-21 | PB1 | New revision | EGBOTAT |
| 2010-12-01 | PC1 | New module parameters | EJNOSVN |
| 2011-09-16 | PD1 | Component Info | ETHJGI |
| 2014-11-24 | PE1 | Added comment in Overview chapter | ESZILSZ |
| 2014-11-24 | PE2 | References section updated | ESZILSZ |
| 2014-12-02 | E | Updated for release | ESZILSZ |

## About this Document

### How to Read this Document

This is the User Guide for the Base feature 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 Base 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](http://www.etsi.org/deliver/etsi_es/201800_201899/20187301/03.02.01_60/es_20187301v030201p.pdf)
2. 1/198 17-CRL 113 200/4 Uen  
   User Guide for the TITAN TTCN-3 Test Executor
3. 109 21-CNL 113 512-20 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. 4/155 16-CNL 113 512  
   EPTF CLL Base, Function Description
7. EPTF CLL for TTCN-3 toolset with TITAN, [Reference Guide](http://ttcn.ericsson.se/TCC_Releases/Libraries/EPTF_Core_Library_CNL113512/doc/apidoc/html/index.html)

### 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

## System Requirements

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

# Base

## Overview

The EPTF Base component is a fundamental component that handles common tasks that should be implemented in all EPTF components. All EPTF components should extend this component (i.e. explicitly) or a component that extends this component (i.e. implicitly).

The Base component

* stores the name of the component
* manages the clean up so that the component could terminate gracefully. This solution is similar to the C++ “destructor” concept.
* provides relative and absolute time querying functions
* provides functions to convert the component type to/from integer
* makes negative testing possible with error and assert functions
* provides process information on every PTC running on components extending the EPTF Base component
* enables/disables DTE handling functionality in the whole CLL

To be able to use EPTF Base, the user component should extend the EPTF\_Base\_CT component and call its initializing function. When the component terminates it should call one of the cleanup/stop/stopAll functions of the Base component.

## Description of files in this feature

The EPTF CLL Base API includes the following files:

* Base
  + EPTF\_CLL\_Base\_Definitions.ttcn: This TTCN-3 module contains common type definitions used by the Base Component.
  + EPTF\_CLL\_Base\_Functions.ttcn: This TTCN-3 module contains the implementation of Base functions.
  + EPTF\_CLL\_Base\_ExternalFunctions.cc: This TTCN-3 module contains external functions for EPTF Base.

## Description of required files from other features

The EPTF Base feature depends on the following features of EPTF:

* Common

## Installation

Since EPTF\_CLL\_Base 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\_Base:

* Copy the files listed in section [‎2.2] and [‎2.3] to the directory of the test suite or create symbolic links to them.
* Import the Base demo or write your own application using EPTF Base.
* 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 Base 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.

The EPTF Base feature defines 2 module parameters:

* tsp\_EPTF\_Base\_serialStopAllComponents : if true: components are stopped one after another, otherwise: in parallel (faster, but might lead to errors). Default: true
* tsp\_EPTF\_Base\_debug : if true, the debug style messages are logged into the logfiles, otherwise not.  
  Default: false

## Design rules

To be able to use the EPTF Base functionality in a proper way, the following design rules have to be followed in any component that uses EPTF Base features:

1. in your init function call all the init functions of those components that your component explicitly extends, e.g. the f\_EPTF\_Base\_init\_CT in case your component extends the EPTF\_Base\_CT component
2. after that, initialize your component variables
3. activate your default altsteps
4. register your cleanup function by f\_EPTF\_Base\_registerCleanup at the end of your init function
5. call f\_EPTF\_Base\_cleanup\_CT before your component terminates, or call f\_EPTF\_Base\_stop if the execution should be stopped (e.g. because of an error the execution is forced to stop immediately)
6. do not call any cleanup function anywhere, especially inside your cleanup function. Only the cleanup function of the Base component should be called
7. make sure that any event that comes after the cleanup function was called, is discarded (or does not cause problems) by e.g. deactivating your default altsteps in your cleanup function
8. make sure, that if any of your functions is called if your feature is not initialized, it does not cause an error, similarly port operations (send function) does not cause error if the port is not connected
9. make sure, that your init and cleanup functions can be called several times (with doing nothing during the additional calls)

# 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.

**ERROR: f\_EPTF\_Base\_init\_CT is not called before calling f\_EPTF\_Base\_RegisterCleanup for <cleanup-function>**

The Base feature was not initialized before trying to register a cleanup function.

**ERROR: Unexpected message received from <component>: <message>.**

The component sent an unexpected message.

# 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.

**WARNING: Unexpected message received during termination process from <component>: <message> Message dropped.**

The component sent an unexpected message during cleanup. The message was ignored.

**WARNING: Cleanup is disabled in stopRemote message. CLEANUP IS NOT PERFORMED.**

The stopRemote message received requested a stop without cleanup.

# Examples

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

* EPTF\_Base\_test.ttcn

## Demo Module

The demo module (EPTF\_Base\_test.ttcn) illustrates a typical usage of the EPTF Base feature.