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

EPTF CLL Buffer, Function Description

Contents

[1 Introduction 2](#_Toc271711346)

[1.1 Revision history 2](#_Toc271711347)

[1.2 How to Read this Document 2](#_Toc271711348)

[1.3 References 2](#_Toc271711349)

[1.4 Scope 2](#_Toc271711350)

[1.5 Recommended way of reading 3](#_Toc271711351)

[1.6 Typographical conventions 3](#_Toc271711352)

[1.7 Abbreviations 3](#_Toc271711353)

[1.8 Terminology 3](#_Toc271711354)

[2 General Description 3](#_Toc271711355)

[3 Functional Interface 4](#_Toc271711356)

[3.1 Naming Conventions 4](#_Toc271711357)

[3.2 Public Functions 4](#_Toc271711358)

[3.2.1 Initialization 4](#_Toc271711359)

[3.2.2 Creating a new buffer 4](#_Toc271711360)

[3.2.3 Clearing the buffer 4](#_Toc271711361)

[3.2.4 Rewinding the buffer 5](#_Toc271711362)

[3.2.5 Get current position in the buffer 5](#_Toc271711363)

[3.2.6 Set position within buffer 5](#_Toc271711364)

[3.2.7 Retrieving length of the buffer 5](#_Toc271711365)

[3.2.8 Retrieving data from the buffer 5](#_Toc271711366)

[3.2.9 Get readable length of the buffer 5](#_Toc271711367)

[3.2.10 Get readable data out of buffer 5](#_Toc271711368)

[3.2.11 Put an octetstring into buffer 6](#_Toc271711369)

[3.2.12 Increase buffer length 6](#_Toc271711370)

[3.2.13 Erase the beginning part of the buffer 6](#_Toc271711371)

[3.2.14 Erase the end part of the buffer 6](#_Toc271711372)

[3.2.15 Check whether buffer contains a complete TLV 6](#_Toc271711373)

[3.2.16 Get Readable Data from buffer from Offset 6](#_Toc271711374)

[3.2.17 Get Data from buffer from Offset 7](#_Toc271711375)

[3.3 Summary Table of all public functions for EPTF Base 7](#_Toc271711376)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2010-06-22 | PA1 | First draft version | EJNOSVN |
| 2010-08-12 | PA1 | Modification after review | EJNOSVN |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## How to Read this Document

This is the Function Description for the Buffer 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. 6/198 17-CRL 113 200 Uen  
   Programmer’s Guide – API Technical Reference for TITAN TTCN-3 Test Executor
7. 4/155 16-CNL 113 512 Uen  
   EPTF CLL Base Function Description

## Scope

This document is to specify the content and functionality of the Buffer 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]

MTC Main Test Component

TLV Tag, Length, Value

## 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 Buffer feature of the TitanSim CLL.

The EPTF Buffer component provides TTCN3 function access to the native buffer of Titan (‎[6] chapter 3.1.2 – TTCN Buffer).

This component

* stores an octetstring and allows of querying, adding to, extending, completing it.
* supports the graceful termination feature of CLL Base in TitanSim ‎[7] .

To be able to use EPTF Buffer, the user component should extend the EPTF\_Buffer\_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\_Buffer\_.

## Public Functions

### Initialization

Before using the EPTF Buffer component the

*f\_EPTF\_Buffer\_init\_CT( pl\_selfName )*

function should be called. This initializes the EPTF Buffer feature. The EPTF self name of the component shall be passed as parameter (pl\_selfName).

### Creating a new buffer

To create a new buffer the function

*integer f\_EPTF\_Buffer\_new()*

has to be called. It returns an identifier to the buffer that has been created.

### Clearing the buffer

If the content of the buffer has to be erased, the function

*f\_EPTF\_Buffer\_clear( pl\_bufferId )*

has to be called. It clears the content, all the position pointers of the buffer identified by the pl\_bufferId.

### Rewinding the buffer

The position of the buffer is set to the beginning of the buffer by the function

*f\_EPTF\_Buffer\_rewind( pl\_bufferId ).*

### Get current position in the buffer

The function

*integer f\_EPTF\_Buffer\_get\_pos( pl\_bufferId )*

gives back the current position of the buffer identified by the pl\_bufferId.

### Set position within buffer

To set the current position in the buffer, the function

*f\_EPTF\_Buffer\_set\_pos( pl\_bufferId, pl\_pos )*

has to be called.

### Retrieving length of the buffer

The function

*integer f\_EPTF\_Buffer\_get\_len( pl\_bufferId )*

returns the length of the buffer.

### Retrieving data from the buffer

The data stored in the buffer can be retrieved by the

*f\_EPTF\_Buffer\_get\_data( pl\_bufferId, out pl\_data )*

function.

### Get readable length of the buffer

To get the max readable length of data in the buffer the function

*integer f\_EPTF\_Buffer\_get\_read\_len( pl\_bufferId )*

has to be called.

### Get readable data out of buffer

The function

*f\_EPTF\_Buffer\_get\_read\_data( pl\_bufferId, out pl\_data )*

gives back the readable part of data out of the buffer.

### Put an octetstring into buffer

To store an octetstring in the buffer, the function

*f\_EPTF\_Buffer\_put\_os( pl\_bufferId, pl\_data)*

has to be called.

### Increase buffer length

The function

*f\_EPTF\_Buffer\_increase\_length( pl\_bufferId, pl\_count )*

increases the length of the buffer by pl\_count bytes.

### Erase the beginning part of the buffer

The function

*f\_EPTF\_Buffer\_cut( pl\_bufferId )*

erases the buffer from the beginning to the current position.

### Erase the end part of the buffer

The function

*f\_EPTF\_Buffer\_cut\_end( pl\_bufferId )*

erases the buffer from the current position to the end of the buffer.

### Check whether buffer contains a complete TLV

*boolean f\_EPTF\_Buffer\_contains\_complete\_TLV( pl\_bufferId )*

This function returns true if the buffer contains a complete tag, length, value.

### Get Readable Data from buffer from Offset

*f\_EPTF\_Buffer\_getReadDataFromOffset(pl\_bufferId, pl\_pos, out pl\_data)*

This function returns the data from pos to the end of read buffer. The read position is not changed

### Get Data from buffer from Offset

*f\_EPTF\_Buffer\_getDataFromOffset(pl\_bufferId, pl\_pos, pl\_count, out pl\_data)*

This function returns pl\_count bytes of data from position pl\_pos to at most the end of read buffer. The read position is not changed

## Summary Table of all public functions for EPTF Base

Table 1. Summary of Base functions

|  |  |
| --- | --- |
| Function name | Description |
| f\_EPTF\_Buffer\_init\_CT | Initializes the Buffer Component |
| f\_EPTF\_Buffer\_new | Creates a new buffer |
| f\_EPTF\_Buffer\_clear | Clears the buffer |
| f\_EPTF\_Buffer\_rewind | Rewinds the buffer |
| f\_EPTF\_Buffer\_get\_pos | Gets the position of the buffer |
| f\_EPTF\_Buffer\_set\_pos | Sets the position of the buffer |
| f\_EPTF\_Buffer\_get\_len | Gets the length of the buffer |
| f\_EPTF\_Buffer\_get\_data | Gets the data of the buffer |
| f\_EPTF\_Buffer\_get\_read\_len | Gets the length of readable data in the buffer |
| f\_EPTF\_Buffer\_get\_read\_data | Gets the readable data in the buffer |
| f\_EPTF\_Buffer\_put\_os | Puts an octetstring into the buffer |
| f\_EPTF\_Buffer\_increase\_length | Increase the length of the buffer |
| f\_EPTF\_Buffer\_cut | Erases the buffer from the beginning to the current position |
| f\_EPTF\_Buffer\_cut\_end | Erases the buffer from the current position to the end |
| f\_EPTF\_Buffer\_contains\_complete\_TLV | Returns whether the buffer contains the complete TLV |
| f\_EPTF\_Buffer\_getReadDataFromOffset | Returns the data from a given position to the end of the read buffer. |
| f\_EPTF\_Buffer\_getDataFromOffset | Returns given bytes of data from a given position to at most the end of read buffer. |