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

EPTF CLL Random N Array, Function Description

Contents

[1 Introduction 2](#_Toc235515250)

[1.1 Revision history 2](#_Toc235515251)

[1.2 How to Read this Document 2](#_Toc235515252)

[1.3 References 2](#_Toc235515253)

[1.4 Scope 2](#_Toc235515254)

[1.5 Recommended way of reading 2](#_Toc235515255)

[1.6 Typographical conventions 3](#_Toc235515256)

[1.7 Terminology 3](#_Toc235515257)

[2 General Description 3](#_Toc235515258)

[3 Functional Interface 5](#_Toc235515259)

[3.1 Naming Conventions 5](#_Toc235515260)

[3.2 Public Functions 5](#_Toc235515261)

[3.2.1 Initialization 5](#_Toc235515262)

[3.2.2 Getting or Creating Slots 5](#_Toc235515263)

[3.2.3 Moving a Slot 6](#_Toc235515264)

[3.2.4 Checking the State of an Element 6](#_Toc235515265)

[3.2.5 Getting the Location of an Element 6](#_Toc235515266)

[3.2.6 Getting the Length of a List 6](#_Toc235515267)

[3.3 Summary Table of all public functions for EPTF Random N Array 7](#_Toc235515268)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2009-02-20 | PA1 | First draft version | EBENMOL |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## How to Read this Document

This is the Function Description for the Random N Array 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 Random N Array 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 ‎0 and ‎1.7, 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]

FBQ Free Busy Queue

RNA Random N Array

RFBA Random Free Busy Array

## 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 Random N Array feature of the TitanSim CLL.

The EPTF Random N Array feature makes it possible to

* Manage lists over ‘record of’ data structures easily using index-arithmetics
  + Provide efficient iterators over the elements linked into the same list
  + Provide a number of lists containing elements, each list representing a different state
* Choosing random elements
  + Provide a method from choosing a random element from a list.
* Using an interface similar to FBQ architecture
  + Provide a free and busy list similar to FBQ, called Random Free Busy Array (RFBA)
  + Provide functions with similar interface as defined in FBQ for managing RFBA

The aim of the EPTF Random N Array feature is to provide dynamic memory allocation for the TTCN-3 language in an efficient way.

RNA contains N IntegerList. An IntegerList can represent a state (free, busy, invalid etc.). Each IntegerList stores a specified number of elements. There is an IdxMap, that contains Integer pairs (IntegerLists) for each elements where the first integer of the pairs points to the List where the element is stored and the second stores the index of the element inside that IntegerList.

The advantage of this structure is that an element can be selected randomly from the lists.

RFBA is a specific RNA. It supports two IntegerLists or two states a Free and a Busy state, that can contain elements. The advantage of RFBA is that it integrates well into the current code, since the FBQ has a similar interface. RNA on the other hand is more generic with its N states. Figure 1 shows a simple RFBA.



Figure RandomFreeBusyArray

# 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\_RNA.

## Public Functions

### Initialization

Before using an EPTF Random N Array the

*f\_EPTF\_RNA\_initRNA(pl\_size,* *pl\_numOfLists, pl\_queue)*

function should be called with the queue as the parameter. This initializes the RNA containing *pl\_numOfLists* number of lists or states. List number 0 will contain *pl\_size* elements.

The RFBA can be initialized with the function

*f\_EPTF\_RNA\_initRFBA(pl\_size,* *pl\_queue)*

It will have only two lists or states can be called Free and Busy.

### Getting or Creating Slots

The function *f\_EPTF\_RNA\_getRndOrCreateSlot(pl\_listNum, pl\_queue)* can be used to get an element from a random slot from a list specified by *pl\_listNum*. The function resizes the list if there are no slots available there.

Function *f\_EPTF\_RNA\_createSlots (pl\_number, pl\_listNum, pl\_queue)* can be used to create several new slot items at once in the list specified by *pl\_listNum*.

Function *f\_EPTF\_RNA\_getRndSlot(pl\_listNum, pl\_queue)* can be used to get an element from a random slot from a list specified by *pl\_listNum*. This function cannot resize the list, and returns with a negative index if there are no free slots.

The function *f\_EPTF\_RNA\_getRndOrCreateFreeSlot( pl\_queue)* can be used to get an element from a random slot from the free list in an RFBA. The function resizes the list if there are no slots available there.

Function *f\_EPTF\_RNA\_createFreeSlots (pl\_number, pl\_queue)* can be used to create several new slot items at once in the free list in an RFBA.

Function *f\_EPTF\_RNA\_getRndFreeSlot(pl\_queue)* can be used to get an element from a random slot from the free list in an RFBA. This function cannot resize the list, and returns with a negative index if there are no free slots.

### Moving a Slot

The following functions can be used to move a slot:

*f\_EPTF\_RNA\_moveSlot(pl\_elementIdx, pl\_listTo, pl\_queue)*

can be used to move an element to a list specified by the parameter *pl\_listTo*.

*f\_EPTF\_FBQ\_moveFromFreeToBusySlot(pl\_elementIdx, pl\_queue)* can be used to move an element from the free list into the busy list in an RFBA.

*f\_EPTF\_FBQ\_moveFromBusyToFreeSlot(pl\_elementIdx, pl\_queue)* can be used to move an element from the busy list into the free list in an RFBA.

### Checking the State of an Element

The following functions can be used to check if an element is in a given state (or list) in an RFBA:

*f\_EPTF\_RNA\_itemIsBusy(pl\_elementIdx, pl\_queue)* returns true if the element is in the busy list,

*f\_EPTF\_RNA\_itemIsFree(pl\_elementIdx, pl\_queue)* returns true if the element is in the free list.

### Getting the Location of an Element

The function

*f\_EPTF\_RNA\_getList(pl\_elementIdx, pl\_queue)*

can be used to get the index of the list where *pl\_elementIdx* located.

### Getting the Length of a List

The function *f\_EPTF\_RNA\_getLengthOfList(pl\_listNum, pl\_queue)* can be used To get a length of the list specified by *pl\_listNum.*

The functions *f\_EPTF\_RNA\_getLengthOfBusyList(pl\_queue)* and *f\_EPTF\_RNA\_getLengthOfFreeList(pl\_queue)* can be used to get the length of the busy and the free list respectively in an RFBA.

## Summary Table of all public functions for EPTF Random N Array

Table 1. Summary of Random N Array functions

|  |  |
| --- | --- |
| Function name | Description |
| f\_EPTF\_RNA\_initRNA | Initializes a Random N Array |
| f\_EPTF\_RNA\_getRndOrCreateSlot | Returns an index to an existing or a newly created element |
| f\_EPTF\_RNA\_createSlots | Creates slots in a list |
| f\_EPTF\_RNA\_getRndSlot | Returns an element from a random slot |
| f\_EPTF\_RNA\_moveSlot | Moves a slot to an other list |
| f\_EPTF\_RNA\_getLengthOfList | Returns the length of a specified list |
| f\_EPTF\_RNA\_getList | Gets the location of an element |
| f\_EPTF\_RNA\_initRFBA | Initializes a Random Free Busy Array |
| f\_EPTF\_RNA\_getRndOrCreateFreeSlot | Returns an index to an existing or a newly created element in the free list |
| f\_EPTF\_RNA\_createFreeSlots | Creates slots in the free list |
| f\_EPTF\_RNA\_getRndFreeSlot | Returns an element from a random slot in the free list |
| f\_EPTF\_RNA\_moveFromFreeToBusySlot | Moves a slot from the free list to the busy list |
| f\_EPTF\_RNA\_moveFromBusyToFreeSlot | Moves a slot from the busy list to the free list |
| f\_EPTF\_RNA\_getLengthOfFreeList | Returns the length of the free list |
| f\_EPTF\_RNA\_getLengthOfBusyList | Returns the length of the busy list |
| f\_EPTF\_RNA\_itemIsBusy | Returns true if the item is in the busy list |
| f\_EPTF\_RNA\_itemIsFree | Returns true if the item is in the free list |