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

EPTF PTC Deployment User Documentation

Contents

[1 Introduction 2](#_Toc211150437)

[1.1 Revision history 2](#_Toc211150438)

[1.2 About this Document 2](#_Toc211150439)

[1.2.1 How to Read this Document 2](#_Toc211150440)

[1.2.2 Presumed Knowledge 2](#_Toc211150441)

[1.2.3 References 2](#_Toc211150442)

[1.2.4 Abbreviations 2](#_Toc211150443)

[1.3 System Requirements 3](#_Toc211150444)

[1.4 Configuration 3](#_Toc211150445)

[2 Description of the PTC Deploying feature 3](#_Toc211150446)

[2.1 Host related operations 3](#_Toc211150447)

[2.1.1 Adding hosts 3](#_Toc211150448)

[2.1.2 Removing hosts 3](#_Toc211150449)

[2.1.3 Query hosts 4](#_Toc211150450)

# Introduction

## Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Rev | Characteristics | Prepared |
| 2007-11-12 | PA1 | First draft version | EZSOSZA |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## About this Document

### How to Read this Document

This is the User Documentation for the PTC deployment component, the EPTF\_PTCD\_CT of the Ericsson Performance Test Framework (TitanSim), Core Load Library (CLL).

### Presumed Knowledge

To use this functionality, the knowledge of the TTCN-3 language [1] is essential. Additionally, the user should be familiar with the TitanSim CLL.

### References

[1] ETSI ES 201 873-1 v.2.2.1 (02/2003)  
The Testing and Test Control Nota­tion version 3. Part 1: Core Language

[2] 1/1553-CRL 113 200 Uen  
User Documentation for the TITAN TTCN-3 Test Executor

[3] 3/155 16 -CNL 113 512 Uen   
 EPTF PTC Deplyment Functional Description

### 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]

PTC Parallel Test Component

PTCD PTC Deployment

## System Requirements

In order to use the PTC Deployment feature, the following system requirements must be satisfied:

* TITAN TTCN-3 Test Executor installed. For the installation guide see [2].

## 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].

The tsp\_PTCD\_database TSP can be used to give initial value to the PTCD database from the run-time configuration file.

# Description of the PTC Deploying feature

In case you are simulating many users for your test, and have a none-homogenous computer park, what is used for running many PTCs, you may want to specify how many PTCs you want to run on a selected computer.

With the PTCD component you can give a weight to every host, thus the hosts with bigger weights will be chosen more often for deploying a PTC.   
The weights must be positive integers.

To use the functionality in your component, you need to extend EPTF\_PTCD\_CT first. You also need to call f\_EPTF\_PTCD\_init\_CT when you are initializing your component.

## Host related operations

### Adding hosts

With the function f\_EPTF\_PTCD\_AddHosts, the database can be filled up with hosts and their weights. It requires a role, and a list of hosts with weights. If the role is already exists, the host will be appended to the existing hosts.  
Returns no value.

Example:  
f\_EPTF\_PTCD\_AddHosts(“roleA”,{{“hostA”,35},{“192.168.1.2”,11}});

### Removing hosts

The function f\_EPTF\_PTCD\_RemoveHosts, deletes the given hosts from the database. It requires a role, and a list of hosts.  
Returns no value.

Example:  
f\_EPTF\_PTCD\_RemoveHosts(“roleB”,{“192.168.2.1”,”hostC”});

### Query hosts

The function f\_EPTF\_PTCD\_GetHost requires a role and a method indicating integer value what is optional and chooses a host from that role according to the way described in the Functional Description of EPTF PTCD[3].  
The default method is a random over against the deterministic way.  
Note, that, when using the deterministic algorithm, you must query for a host at least sum of the weights to get all the hosts surely chosen, and than you can be sure that the distribution of the hosts are following the weights that was specified.  
Returns the name of the host.

Example:  
var charstring host := f\_EPTF\_PTCD\_GetHost(“roleA”);

var charstring host := f\_EPTF\_PTCD\_GetHost(“roleA”, 1);  
If you would like to generate the host in a deterministic way.