|  |  |
| --- | --- |
| **(Confidential)** | |
| Scope of disclosure | [TSCC, IoTT, TSDV] for SKERLET-IoTT only |
| Period of confidentiality | 07 years after Issued Date |
| Head of Information Owner | Head of Engineering Dept |
| Handling restriction | N/A |

**Software Detailed Design**

**Trial development of Skelios Linux**

**Toshiba Software Development (Vietnam) Co., Ltd.**

|  |
| --- |
| Document ID: TSDV-SKERLET-IoTT-PackageTests-Automate-SAD |
| Total: 9 Page No. 1 |

Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rev. No.  (X.YY) | Date  (YY-MM-DD) | Section No. Changed | Change Description | Author | Reviewed by | Approved by |
| 0.01 | 2016-08-15 | All | Initialize document | TienL |  |  |
| 0.02 | 2016-08-16 | All | Collect ideas after meeting | TienL | TrungDT  DungTTV  HieuNV  TuyenHV  VinhNC |  |
| 0.03 | 2016-08-31 | 5. Design structure | Update example structure | TuyenHV | TienL |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Content

[Revision History 2](#_Toc460398990)

[1. Introduction 4](#_Toc460398991)

[1.1. Goal 4](#_Toc460398992)

[1.2. Note 4](#_Toc460398993)

[1.3. Issues 4](#_Toc460398994)

[1.4. Ideas 4](#_Toc460398995)

[2. References 5](#_Toc460398996)

[3. Definitions and Acronyms 6](#_Toc460398997)

[4. Define rules to write test cases 7](#_Toc460398998)

[5. Design structure 8](#_Toc460398999)

# Introduction

## Goal

– Run all test cases from one script #chạy tất cả các testcase từ 1 script  
– Automatically generate summary of the test results #tự sinh ra file kết quả

## Note

– All test cases must be implemented with minimal dependencies #các testcase phải được thực thi với số lượng dependencies tối thiểu.  
– All test cases should not depend on specific environment #các testcase không nên phụ thuộc vào một môi trường đặc biệt nào:  
 • Distribution, hardware, etc. #Phân phối, phần cứng

#Nên thực hiện testcase nào?

Ta không thể test được tất cả các funtions của gói.

#Làm thế nào để tái sử dụng test case?

Test case trong source tree, ptest...

#Làm thế nào để thi hành scripts cho test case đang chạy?

Cấu hình file, kicker script, phân tích scripts v.v.

#Làm thế nào để phân tích kết quả test case?

Trả về các giá trị, hiển thị thông báo, trạng thái các files v.v.

#Test case nào nên thi hành?

Một test case có thể là một phần của chương trình lớn.

#Kỹ thuật + điều kiện test = test case

#Sổ tay test cũng được tạo thành từ các đặc điểm và ví dụ

## Issues

– Which test cases should we implement?  
 • We cannot test all functions of a package  
– How to reuse the existing test cases?  
 • Test cases in source tree, ptest, etc.  
– How to implement scripts for running test cases  
 • Configuration files, kicker script, analyzer script, etc.  
– How to analyze the output (results) of test cases  
 • Return values, displayed messages, states of files, etc.

## Ideas

– A test case may also be a part of specifications of program  
– Specification + test condition = test case  
– Manuals are also created from specifications and examples  
– Basic policy for creating test cases  
 • Create manual for developers TOSHIBA/TSDV

# References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Document/Standards Name/Title** | **Source** | **Version No. /Release or Publication date** | **Brief Description/Section Reference** |
| 1 | TSDV-items\_20160725.pdf | IoTT | 2016-07-25 | TSDV 2016A items |
| 2 | |  | | --- | | TSDV-automation\_20160802.pdf | | IoTT | 2016-08-02 | Meeting minutes between customer (Hayashi-san) with SKERLET team |

# Definitions and Acronyms

|  |  |  |
| --- | --- | --- |
| **No** | **Acronyms** | **Definition** |
| 1 | TSDV | Toshiba Software Development (Vietnam) |
| 2 | IoTT | New name of SWC |

# 

# Define rules to write test cases

* All test cases must be implemented with minimal dependencies.
* All test cases should not depend on specific environment.
* All test cases must be reverted configurations after getting output.

# Design structure

Source code is included the files and folder as following:

* **runTest.sh** file
* Used to execute testing.
* Have some functions as setup, create test list, runtest, test\_processing, cleanup
* **packageTestList.txt** file
* List test cases which are implemented for testing when run **runTest.sh** file
* Include many line, each line have the following format

PACKAGE\_NAME: testcase1 testcase2 testcase3 …

IF **PACKAGE\_NAME: \*** => test all testcases of package

IF **PACKAGE\_NAME: (NULL)** => no test this package

Example:

busybox: ls cd rm

bzip2:

coreutils: \*

* **output** folder
* Contents result.txt file and output logs when run testing
* Structure: output/package/<testcase.log>
* Example: output/package/ls.log
* **testcases** folder
* Contents script, data for all test cases.
* Structure: testcases/package\_name/<testcase.sh>
* Data store in testcases/package\_name/data/testcase.d/<some-config-files>
* Example

testcases/busybox/ls.sh

testcases/busybox/data/ls.d/config.txt

A example structure:

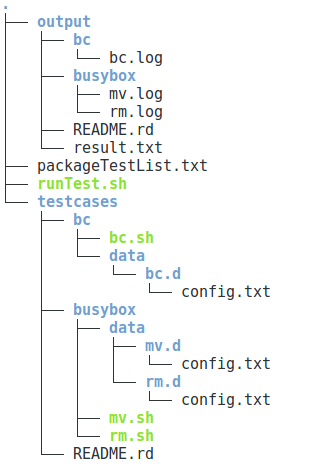


Figure : Example Structure

Format of a test script should be implemented as below:

*#!/bin/bash*

*#=============================================================*

*# FILE : example.sh*

*# DESCRIPTION: Write description for test script*

*# write description for test script*

*#=============================================================*

*# write content of test scipt*

Note:

Some variable environment.

export PWD=`pwd`

export OUTPUT\_DIR=${PWD}/output

export TESTCASES\_DIR=${PWD}/testcases

export CONFIG\_DIR=${PWD}/common\_config

export TMPBASE="/tmp"

export TMP=`mktemp -d ${TMPBASE}/test-XXXXXXXXXX`

export RESULT\_FILE=${OUTPUT\_DIR}/result.txt

test\_case\_name=$(basename $i .sh)

tmp\_dir=$(dirname $i)

package\_name=$(basename $tmp\_dir)

data\_dir=${TESTCASES\_DIR}/${package\_name}/data/${test\_case\_name}.d

mkdir -p ${OUTPUT\_DIR}/${package\_name}

log\_file=${OUTPUT\_DIR}/${package\_name}/${test\_case\_name}.log

test\_passed\_text="Testing for ${test\_case\_name} testcase: PASSED"

test\_failed\_text="Testing for ${test\_case\_name} testcase: FAILED"

package\_name: The name of package

test\_case\_name: The name of test case

See a run\_test.sh script for more detail.

Coding convention should be followed by <https://google.github.io/styleguide/shell.xml>

Indentation: Indent 2 spaces. No tabs.

Put ; do and ; then on the same line as the while, for or if.

All error messages should go to log file.

The result of test have to write to ${OUTPUT\_DIR}/result.txt.

If test script have changed default configuration of the system, please add this changing to the end of script with format.

# Configuration changed from default######################################

# write something here

#####################################################################