

國科會自由軟體專案研究計劃測試報告
Testing Specification Document of NSC Open Source
Project

一個 *Java GUI* 測試工具的設計與製作
NSC 92-2218-E-027-017

陳偉凱 副教授

國立台北科技大學資訊工程研究所

Department of Engineering and Applied Science

National Science Council, Taiwan

2004/10/31

一個 Java GUI 測試工具的設計與製作

Graphical user interface Testing Tool (GTT)

(計畫執行期間：2003.12 至 2004.10)

一個 Java GUI 測試工具之系統測試文件

(GTT Testing Documents)

製作者

張哲銘 92598001

核准者

陳偉凱 副教授

邱任維 92598008

蔡東宏 91598020

沈政文 93598001

杜秉穎 93598013

2004/10/31

國立台北科技大學資訊工程所

GTT 發展小組

目 錄

版次變更記錄	3
Section 1. 測試目的與接受準則(Objectives and Acceptance Criteria)	4
1.1 系統範圍(System Scope)	4
1.2 測試目的(Purpose of this Document)	4
1.3 測試接受準則(Test Acceptance Criteria).....	4
Section 2. 測試成員及工作指派 (Personnel and Responsibility)	5
2.1 測試成員(Personnel)	5
2.2 測試工作指派(Responsibility).....	5
Section 3. 測試環境 (Testing Environment)	6
3.1 操作環境(Operational Environment)	6
3.2 硬體需求(Hardware Specification and Configuration).....	6
3.3 軟體需求(Software Specification and Configuration)	6
3.4 測試資料來源(Test Data Sources)	6
3.5 測試工具與設備(Tools and Equipments).....	7
Section 4. 測試案例(Test Cases)	8
Section 5. 測試程序與時程(Test Procedure and Schedule)	28
Section 6. 測試結果與分析(Test Results and Analysis)	28
字彙(Glossary)	30
參考文獻 (References)	32
附錄一. 追溯表(Traceability Matrix)	33

版次變更記錄

版次	變更項目	變更日期
1.0	第一版	2004/10/31

Section 1. 測試目的與接受準則(Objectives and Acceptance Criteria)

1.1 系統範圍(System Scope)

圖形使用者介面測試工具(GUI Testing Tool，簡稱 GTT)，是介於 User 與 Java Swing-based Application(簡稱 AP)之間的測試工具。一般使用者對 GUI 應用程式是直接性的互動，而 GTT 的技術是位於此兩者互動之間，負責對 Java 應用程式之圖形使用者介面進行錄影(截取)、編輯、播放(執行)、測試的功能。

1.2 測試目的(Purpose of this Document)

本測試文件主要針對本系統進行測試，以便達到以下的目的：

- (1) 定義執行方案以便為達成系統的「初步測試」(Beta Testing)與「驗收測試」(User Acceptance Testing)目標作預先的準備。
- (2) 定義可進行驗收的項目(deliverables)與相關的責任區分(responsible)。
- (3) 與相關的負責單位進行溝通，以便決定相依性與可能風險(Dependencies and Risks)。

1.3 測試接受準則(Test Acceptance Criteria)

本測試計劃需要滿足下面的測試接受準則：

- (1) 測試程序需要依照本測試計劃所訂定的程序進行，所有測試結果需要能符合預期測試結果方能接受。
- (2) 以測試案例為單位，當測試未通過時，需要進行該單元的測試，其接受的準則如第一項中所規定的相同。

Section 2. 測試成員及工作指派 (Personnel and Responsibility)

2.1 測試成員(Personnel)

The test team will consist of:

成員名單與縮寫對照表			
縮寫	姓名	縮寫	姓名
CM	張哲銘	CW	沈政文
ZW	邱任維	BI	杜秉穎
DH	蔡東宏		

Affiliation	Name	Abbreviation	Memo
Project Manager	陳偉凱	KC	
Lead Developer	蔡東宏	DH	
Test Lead	張哲銘	CM	
Tester	邱任維	ZW	
Tester	沈政文	CW	
Tester	杜秉穎	BI	

2.2 測試工作指派(Responsibility)

職務名稱	任務內容說明
Project Manager	Responsible for Project schedules and the overall success of the project.
Lead Developer	Serve as a primary contact/liaison between the development department and the project team.
Test Lead	Ensures the overall success of the test cycles. He/she will coordinate weekly meetings and will communicate the testing status to the project team.
Testers	Responsible for performing the actual system testing.

Section 3. 測試環境 (Testing Environment)

3.1 操作環境(Operational Environment)

本測試環境於 Windows 的 java run-time virtual machine 環境下進行，進行測試前需要進行下面環境的設定：

- (1) 不可同時開啟兩個 GTT 系統。
- (2) 受測於 GTT 之待測程式必須使用 Swing 之 GUI 所建構。
- (3) 受測於 GTT 之待測程式必須是可執行之 Java 程式。

3.2 硬體需求(Hardware Specification and Configuration)

項次	名稱	數量	規格	備註
1	相容於 IBM PC 之個人電腦	1	256 MB RAM 200 MB Hard Drive Space	

3.3 軟體需求(Software Specification and Configuration)

項次	名稱	數量	規格	備註
1	Windows 作業系統	1	XP 版	
2	Java Virtual Machine	1	v1.4	
3	IBM Eclipse	1	release v2.1.2	

3.4 測試資料來源(Test Data Sources)

本測試計畫的測試資料來源可分成下面的資料來源：

- (1) Swing GUI 實作成的程式
- (2) Swing GUI 實作成的 Calculator (CalculatorForInstance.class) 及 MyWindows (MyWindows.class)
- (3) GTT 自我測試的 .gtt 檔。
 1. Construct Test-Flow with AUT.gtt
 2. Open Test-Flow and Add Internal-Test.gtt
 3. Open Test-Flow and Add External-Test.gtt
 4. Execute Test-Flow and Add New Test-Flow.gtt

3.5 測試工具與設備(Tools and Equipments)

本測試計劃的執行採用使用者直接操作的測試方式，以及使用 GTT 測試 GTT 本身，因此測試進行中不需要測試工具的輔助，但對於測試的資料整理則需要文書編輯工具 Word XP。

Section 4. 測試案例(Test Cases)

Identification	1001.
Name	開啟待測應用程式之功能測試
Tested target	GTT
Reference	GTT001 GTT008
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 開啟 GTT 系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 CalculatorForInstance.class 4. 按下[確定]
Expected result	完成上述功能後，GTT 系統會開啟指定的待測應用程式 Calculator
Cleanup	按下[取消]

Identification	1002.
Name	錄影功能測試
Tested target	GTT
Reference	GTT002 GTT004 GTT005 GTT006
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1001 的功能(開啟待測應用程式) 3. 按下[錄影] 4. 操作待測應用程式，依序按下[數字鍵 1]、[數字鍵 2]、[數字鍵 3]、[數字鍵 4]、[數字鍵 5]、[數字鍵 6]、[數字鍵 7]、[數字鍵 8]、[數字鍵 9]、[數字鍵 0] 5. 按下[停止]
Expected result	完成上述功能後，GTT 系統會運算出相關的測試流程並以樹狀結構展現出來
Cleanup	按下[關閉 GTT]

Identification	1003.
Name	儲存測試流程之功能測試
Tested target	GTT
Reference	GTT002 GTT006
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1002 的功能(錄影) 3. 按下[儲存測試流程] 4. 輸入檔案名稱[test] 5. 按下[完成]
Expected result	完成上述功能後，GTT 系統會將測試流程儲存至檔案

	test.gtt 中
Cleanup	按下[取消]

Identification	1004.
Name	另存新檔
Tested target	GTT
Reference	GTT017
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1002 的功能(錄影) 3. 完成 Identification 1003 的功能(儲存測試流程) 4. 按下[另存新檔] 5. 輸入檔案名稱[newtest] 6. 按下確定
Expected result	完成上述功能後，GTT 系統會將測試流程儲存至檔案 newtest.gtt 中
Cleanup	按下[取消]

Identification	1005.
Name	播放測試流程之功能測試
Tested target	GTT
Reference	GTT003
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1002 的功能(錄影) 3. 按下[播放測試流程]
Expected result	完成上述功能後，GTT 系統會對待測應用程式播放測試流程，如果流程正確，測試流程樹狀圖的節點會以綠色顯示，若流程中某節點錯誤，則該節點會以紅色顯示
Cleanup	按下[停止]

Identification	1006.
Name	開啟過去所儲存的測試流程
Tested target	GTT
Reference	GTT019
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟檔案] 3. 找到於 Identification 1004 所儲存的檔案 newtest.gtt 4. 按下[開啟]
Expected result	完成上述功能後，GTT 系統會將測試流程載入至系統中
Cleanup	按下[取消]

Identification	1007.
Name	停止選取插入事件
Tested target	GTT
Reference	GTT007
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟新檔] 3. 按下[設定] 4. 在設定對話空中，按下選取[待測程式] 5. 選取待測程式 CalculatorForInstance.class 6. 按下去[確定] 7. 點選測試流程的最上層節點 8. 按下[插入事件] 9. 按下[停止選取插入事件] 10. 與待測程式互動
Expected result	與 GTT 互動時不應該有元件事件選取對話框出現
Cleanup	

Identification	1008.
Name	插入事件節點
Tested target	GTT
Reference	GTT007
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟新檔] 3. 按下[設定] 4. 在設定對話空中，按下選取[待測程式] 5. 選取待測程式“CalculatorForInstance.class” 6. 按下[確定] 7. 點選測試流程的最上層節點 8. 按下[插入事件] 9. 按下待測程式中的[1] 10. 在元件選取對話框的元件清單中，選取 “[0][JButton][1][1][0]”後，按下[>] 11. 按下[+] 12. 在確認對話框中按下[確定] 13. 在一次於確認對話框中按下[確定] 14. 按下元件選取對話框中的[確定] 15. 於確認對話框中選取[within]後按下[確定]
Expected result	完成上述功能後，GTT 應該於測試流程中新增一個事件節點
Cleanup	按下[取消]

Identification	1009.
Name	插入內部測試節點
Tested target	GTT
Reference	GTT004 GTT006 GTT010
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1009 的功能(插入事件節點) 3. 按下[插入內部測試節點] 4. 點選待測程式中的文字框 5. 在元件選取對話框的元件清單中，選取 “[0][JTextField][1.][result][0]”後，按下[>] 6. 按下[+] 7. 在對話框中選取[Text():String]，並於 return value 中填入 1 8. 在 Assert Message 中填入 “It should be 1” 9. 按下[確定] 10. 在元件選取對話框中按下[確定] 11. 在確認對話框中選取[After]後按下[確定]
Expected result	完成上述功能後，GTT 應該於測試流程中新增一個內部測試節點
Cleanup	按下[取消]

Identification	1010.
Name	插入外部測試節點
Tested target	GTT
Reference	GTT020 GTT035
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1010 的功能(插入內部測試節點) 3. 按下[插入外部測試節點] 4. 於對話框中找到為該待測程式所寫的 JUnit Test 程式 “CalculatorForInstanceTest.class” 5. 於 GTT 找到的測試方法清單中選取 “testEqualClick(): void” 6. 按下[確定] 7. 在確認對話框中選取[After]後按下[確定]
Expected result	完成上述功能後，GTT 應該於測試流程中新增一個外部測試節點
Cleanup	按下[取消]

Identification	1011.
Name	新增複合節點
Tested target	GTT

Reference	GTT025
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1011 的功能(插入外部測試節點) 3. 選取最頂層節點 4. 按下[新增複合節點] 5. 輸入名稱“Test button 1” 6. 按下[確定] 7. 在確認對話框中選取[Within]後按下[確定]
Expected result	完成上述功能後，GTT 應該於測試流程中新增一個複合節點
Cleanup	按下[取消]

Identification	1012.
Name	以 folder 整理節點
Tested target	GTT
Reference	GTT025
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1012 的功能(插入複合節點) 3. 選取事件節點【1】【1】【JButton】及測試節點【1.】【result】【JTextField】 4. 按下滑鼠右鍵 5. 按下[剪下] 6. 選取複合節點“Test button 1” 7. 按下滑鼠右鍵 8. 按下[貼上] 9. 在確認對話框中選取[Within]後按下確定
Expected result	完成上述功能後，GTT 應該將選取的節點變成複合節點“Test button 1”的子節點
Cleanup	按下[取消]

Identification	1013.
Name	手動新增節點
Tested target	GTT
Reference	GTT007
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1017 的功能(開啟過去所儲存的測試流程) 3. 點選最頂層節點 4. 按下[手動新增節點] 5. 於對話框中依序填入：java.awt.Frame、Calculator、javax.swing.JButton、1、0、0 及 1 後按下確定

	6. 於對話框中選取[After]後按下確定 7. 於對話框中選取[Press]後按下確定
Expected result	完成上述功能後，GTT 應該於測試流程中新增一個事件節點
Cleanup	於任意對話框中按下[取消]

Identification	1014.
Name	編輯節點
Tested target	GTT
Reference	GTT008
Severity	1
Instructions	1. 登入系統 2. 完成 Identification 1013 的功能(以 folder 整理節點) 3. 展開複合節點 4. 選取測試節點 【1.】【result】【JTextField】 5. 按下[編輯節點] 6. 將 Represent Text 欄位中的“1.”改成“1” 7. 按下[確定]
Expected result	完成上述功能後，該節點應該顯示成 【1】【result】【JTextField】
Cleanup	按下[取消]

Identification	1015.
Name	刪除節點
Tested target	GTT
Reference	GTT009
Severity	1
Instructions	1. 登入系統 2. 完成 Identification 1013 的功能(以 folder 整理節點) 3. 選取測試節點 【CalculatorForInstanceTest】【testEqualClick】 4. 按下[刪除節點] 5. 在確認對話框中按下[確定] 6. 選取複合節點“Test button 1” 7. 按下[刪除節點] 8. 在確認對話框中按下[確定]
Expected result	完成上述功能後，測試流程中不該留有任何節點
Cleanup	

Identification	1016.
Name	上、下移動節點
Tested target	GTT
Reference	GTT011 GTT026
Severity	1

Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1012 的功能(新增複合節點) 3. 選取【1】【1】【JButton】節點，按一下[上移] 4. 選取複合節點“Test Button 1”，按一下[下移] 5. 選取【1.】【result】【JTextField】按一下[上移] 6. 選取【1】【1】【JButton】節點，按一下[下移] 7. 選取複合節點“Test Button 1”，按一下[上移] 8. 選取【1.】【result】【JTextField】按一下[下移]
Expected result	完成上述功能後，所有的節點一上到下的順序應該是【1.】【result】【JTextField】、【1】【1】【JButton】、“Test Button 1”
Cleanup	

Identification	1017.
Name	剪下及貼上節點
Tested target	GTT
Reference	GTT026
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1016 的功能(上、下移動節點) 3. 選取最頂層節點 4. 按下[新增複合節點] 5. 輸入名稱“Test Calculator” 6. 按下[確定] 7. 在確認對話框中選取[Within]後按下[確定] 8. 選取【1】【1】【JButton】按一下[剪下] 9. 選取複合節點“Test Button 1”後按一下[貼上]，並於對話框中選取[Within]後按[確定] 10. 選取複合節點“Test Button 1”後按一下[剪下] 11. 選取複合節點“Test Calculator”後按一下[貼上]，並於對話框中選取[Within]後按[確定]
Expected result	完成上述功能後，所有的節點一上到下的順序應該是“Test Calculator”、“Test Button 1”及【1.】【result】【JTextField】，其中“Test Button 1”屬於“Test Calculator”的子節點
Cleanup	

Identification	1018.
Name	複製及貼上節點
Tested target	GTT
Reference	GTT026
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1017 的功能(剪下及貼上節點)

	<ol style="list-style-type: none"> 3. 選取“Test Button 1”後按一下[複製] 4. 選取“Test Calculator”後按一下[貼上] 5. 選取第二個“Test Button 1”按滑鼠右鍵,按 Set Name 後,於對話框中輸入“Test Button 2” 6. 選取【1.】【result】【JTextFiled】按一下[複製] 7. 選取複合節點“Test Button 1”,按一下[貼上],於對話框中選取[Within]後按[確定] 8. 選取複合節點“Test Button 2”,按一下[貼上],於對話框中選取[Within]後按[確定]
Expected result	完成上述功能後,流程中應該要有三個節點,每個複合節點裡都應該要有一個事件節點及測試節點
Cleanup	

Identification	1019.
Name	複製及貼上 sharing 節點(proxy copy and paste)
Tested target	GTT
Reference	GTT026
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1018 的功能(複製貼上節點) 3. 展開“Test Button 1” 4. 選取【1】【1】【JButton】節點,按一下[複製 proxy] 5. 選取複合節點“Test Button 2”,按一下[貼上 proxy]並於在對話框中選取[Within]後按[確定]
Expected result	完成上述功能後,“Test Button 2”中應該新增一個事件節點,點選該新增的節點,“Test Button 1”中的原始節點以及被點選的節點都會顯示成藍色。
Cleanup	

Identification	1020.
Name	解除 Sharing(unproxy)
Tested target	GTT
Reference	GTT025
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1019 的功能(複製及貼上 proxy 節點) 3. 展開“Test Button 2” 4. 選取【1】【1】【JButton】節點,按一下[解除 Sharing]
Expected result	完成上述功能後,此時再次點選該節點,並不會顯示成藍色,而原始節點也不再跟著顯示成藍色
Cleanup	

Identification	1021.
----------------	-------

Name	事件節點轉成測試節點
Tested target	GTT
Reference	GTT025
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1018 的功能(複製貼上節點) 3. 選取外層的【1.】【result】【JTextField】按一下[事件節點與測試節點腳色互換]
Expected result	完成上述功能後，原先的事件節點變成測試節點
Cleanup	

Identification	1022.
Name	測試節點轉成事件節點
Tested target	GTT
Reference	GTT025
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1021 的功能(事件節點轉成測試節點) 3. 選取由 Identification 1021 所轉換的測試節點，按一下[事件節點與測試節點腳色互換]
Expected result	完成上述功能後，原先的測試節點轉變回成事件節點
Cleanup	

Identification	1023.
Name	新增重置節點(Rest application)
Tested target	GTT
Reference	GTT026
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1018 的功能(複製及貼上節點) 3. 展開複合節點“Test Button 1”，選取【1.】【result】【JTextField】後按下[新增重置節點]，並於對話框中選取[After]後按下[確定]
Expected result	完成上述功能後，複合節點“Test Button 1”中，應新增一個重置節點
Cleanup	

Identification	1024.
Name	重置所有節點狀態(Rest test script)
Tested target	GTT
Reference	GTT026
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統

	<ol style="list-style-type: none"> 2. 完成 Identification 1022 的功能(新增重置節點) 3. 按下[播放] 4. 待播放完畢後，按下[重置所有節點狀態]
Expected result	完成上述功能後，步驟3執行完後應該會有很多紅色節點(表示錯誤)，按下[重置所有節點狀態]後，所有表示狀態的顏色都會消失
Cleanup	

Identification	1025.
Name	更新文件資料(Synchronize)
Tested target	GTT
Reference	GTT026
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 完成 Identification 1023 的功能(重置所有節點狀態) 3. 按下[更新文件資料]
Expected result	完成上述功能後，所有測試流程將被收合起來
Cleanup	

Identification	1026.
Name	新增元件事件元素-Abstractbutton
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟新檔] 3. 按下[設定] 4. 在設定對話空中，按下選取[待測程式] 5. 選取待測程式“MyWindow.class” 6. 按下[確定] 7. 點選測試流程的最上層節點 8. 按下[錄製] 9. 於 AUT 中的 TextFieldB 輸入“T” 10. 按下 AUT 中的[Copy B to A] 11. 按下[停止錄製] 12. 於對話框中選取[Within]後按確定 13. 選取【Copy B to A】【b2a】【JButton】 14. 於右側事件測試或節點編輯區中點選 【PUSH_NO_BLOCK】【100】【L】【Clicks:1】後按下刪除 15. 按下[新增元件事件元素] 16. 於對話框中選取[PUSH_NO_BLOCK]後按[確定] 17. 按下[新增測試節點] 18. 於 AUT 中點選[TextFieldA]

	19. 於對話框中選取“[1][JTextField][T][test a][0]”後按下[>] 20. 按下[+] 21. 在對話框中選取[Text():String]，並於 return value 中填入 T 後按下[確定]
Expected result	完成上述功能後，執行測試流程應該是完全正確的
Cleanup	

Identification	1027.
Name	新增元件事件元素-AbstractButton
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	7. 登入系統 8. 按下[開啟待測應用程式] 9. 由檔案列表中選擇待測應用程式 MyWindow.class 10. 執行錄影功能，操作 AUT，按下「Copy A to B」按鈕，之後停止錄影。 11. 對剛錄影下來的事件節點，新增一個事件元素，Event Type 選擇 press，Sleep time 為 1000。 12. 對剛錄影下來的事件節點，新增第二個事件元素，Event Type 選擇 release，Sleep time 為 100。 13. 執行「播放」功能。
Expected result	GTT 會對 AUT 上的「Copy A to B」按鈕執行一次 click，之後再執行按住按鈕 1 秒之後才放開。
Cleanup	按下[關閉 GTT]

Identification	1028.
Name	新增元件事件元素-JComboBox
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，操作 AUT，選擇 tab2，在 JComboBox 上選擇 Items-5，之後停止錄影。 5. 對剛錄影下來的事件節點，刪掉舊的事件元素，接著再新增一個事件元素，Event Type 選擇 CLICK_ON_ITEM，選擇 index，Items 輸入 3。 6. 執行「播放」功能。
Expected result	GTT 會點選 JComboBox 中的 Items-3。
Cleanup	按下[關閉 GTT]

Identification	1029.
----------------	-------

Name	新增元件事件元素-JList
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，操作 AUT，在 JList 上選擇 Items-5，之後停止錄影。 5. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 CLICK_ON_ITEM，選擇 index，Items 輸入 1。 6. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 CLICK_ON_ITEM，選擇 index，Items 輸入 3。 7. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 CLICK_ON_ITEM，選擇 index，Items 輸入 7。 8. 執行「播放」功能。
Expected result	GTT 會依序點選 JList 中的 Items-5、Items-1、Items-3 及 Items-7。
Cleanup	按下[關閉 GTT]

Identification	1030.
Name	新增元件事件元素-JSlider
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，操作 AUT，將 tab2 中的 JSpinner 上的 JSlider 由原本的 375 移到 1000 的位置，之後停止錄影。 5. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SCROLL_TO_MINIMUM。 6. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SCROLL_TO_MAXIMUM。 7. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SCROLL_TO_PERCENT，PERCENT 選擇 50%。 8. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SCROLL_TO_VALUE，VALUE 選擇 750。 9. 執行「播放」功能。

Expected result	GTT 會依序移動 JSlider 由 375 到 1000、0、1000、500、最後到 750。
Cleanup	按下[關閉 GTT]

Identification	1031.
Name	新增元件事件元素-JScrollBar
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，操作 AUT，將 JList 的 JScrollPane 往下拉至 10%，之後結束錄影。 5. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SCROLL_TO_MINIMUM。 6. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SCROLL_TO_MAXIMUM。 7. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SCROLL_TO_PERCENT，PERCENT 選擇 50%。 8. 執行「播放」功能。
Expected result	GTT 會依序移動 JList 上的 JScrollPane 由 0%到 10%、100%、0%，最後停在 50%。
Cleanup	按下[關閉 GTT]

Identification	1032.
Name	新增元件事件元素-JTabbedPane
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，點選 JTabbedPane 的 tab2，之後結束錄影。 5. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SELECT_PAGE、PAGE TITLE 輸入 tab1。 6. 對剛錄影下來的事件，新增一個事件元素，Event Type 選擇 SELECT_PAGE_BY_INDEX、PAGE INDEX 輸入 1(即 PAGE TITLE tab2)。 7. 執行「播放」功能。
Expected result	GTT 會依序點 JTabbedPane 的 tab2、tab1、tab2。
Cleanup	按下[關閉 GTT]

Identification	1033.
Name	新增元件事件元素-JTableHeader
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，將 tab2 中的 JTable 的 col1 跟 col2 交換位置，之後結束錄影。 5. 對剛錄影下來的 JTableHeader 事件，新增一個事件元素，Event Type 選擇 MOVE_COLUMN，column 輸入 1，MOVE TO 輸入 2。 6. 對剛錄影下來的 JTableHeader 事件，新增一個事件元素，Event Type 選擇 MOVE_COLUMN，column 輸入 1，MOVE TO 輸入 0。 7. 執行「播放」功能。
Expected result	播放之後，表格上的 header 依序會是 col3、col2 及 col1。
Cleanup	按下[關閉 GTT]

Identification	1034.
Name	新增元件事件元素-JTextComponent
Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，在 TextField B 輸入 “123”，之後結束錄影。 5. 對剛錄影下來的 text b 事件，新增一個事件元素，Event Type 選擇 CLEAR_TEXT。 6. 對剛錄影下來的 text b 事件，新增一個事件元素，Event Type 選擇 ENTER_TEXT，TEXT 輸入 “abc” 7. 對剛錄影下來的 text b 事件，新增一個事件元素，Event Type 選擇 TYPE_TEXT，POSITION 輸入 3, TEXT 輸入 “xyz” 8. 執行「播放」功能。
Expected result	播放之後 TextField B 會顯示 “abcxyz”。
Cleanup	按下[關閉 GTT]

Identification	1035.
Name	新增元件事件元素-JTree

Tested target	GTT
Reference	GTT001 GTT013 GTT021
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，雙擊 JTree 的 color 節點以展開其子樹，之後結束錄影。 5. 對剛錄影下來 tree 1 事件，新增一個事件元素，Event Type 選擇 CLICK_ON_PATH，PATH 輸入 2。 6. 對剛錄影下來 tree 1 事件，新增一個事件元素，Event Type 選擇 CLICK_ON_PATH，PATH 輸入 3。 7. 對剛錄影下來 tree 1 事件，新增一個事件元素，Event Type 選擇 CLICK_ON_PATH，PATH 輸入 4。 8. 對剛錄影下來 tree 1 事件，新增一個事件元素，Event Type 選擇 CLICK_ON_PATH，PATH 輸入 1，CLICK COUNT 輸入 2。 9. 對剛錄影下來 tree 1 事件，新增一個事件元素，Event Type 選擇 CLICK_ON_PATH，PATH 輸入 0，CLICK COUNT 輸入 2。 10. 執行「播放」功能。
Expected result	播放之後 GTT 會展開 colors 子樹，依序點選 blue、viloet 及 red 節點，之後將 colors 子樹收疊，最後將 JTree 收疊。
Cleanup	按下[關閉 GTT]

Identification	1036.
Name	刪除事件元素
Tested target	GTT
Reference	GTT016
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 完成 Identification 1034 (新增元件事件元素-JTree)的動作。 2. 刪除「【CLOCK_ON_PATH】【100】【PATH:2】【COUNT:1】」 3. 刪除「【CLOCK_ON_PATH】【100】【PATH:3】【COUNT:1】」 4. 刪除「【CLOCK_ON_PATH】【100】【PATH:4】【COUNT:1】」 5. 執行「播放」功能。
Expected result	播放之後 GTT 會展開 colors 子樹，但不再點選 blue、viloet 及 red 節點，之後將 colors 子樹收疊，最後將 JTree 收疊。
Cleanup	按下[關閉 GTT]

Identification	1036
Name	編輯元件事件元素
Tested target	GTT
Reference	GTT013 GTT015
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 完成 Identification 1034 (新增元件事件元素-JTree)的動作。 2. 編輯「【CLOCK_ON_PATH】【100】【PATH:4】【COUNT:1】」，將 PATH 改成 5 (即指 yellow 節點)。 3. 編輯「【CLOCK_ON_PATH】【100】【PATH:0】【COUNT:2】」，將 COUNT 成 1。 4. 執行「播放」功能。
Expected result	播放之後 GTT 會展開 colors 子樹，但不再點選 blue、violet 及 yellow。節點，之後將 colors 子樹收疊，最後並不會收疊 JTree，滑鼠停在 JTree 根節點上。
Cleanup	按下[關閉 GTT]

Identification	1037.
Name	上下移動事件元素
Tested target	GTT
Reference	GTT013 GTT015
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 完成 Identification 1034 (新增元件事件元素-JTree)的動作。 2. 將「【CLOCK_ON_PATH】【100】【PATH:2】【COUNT:1】」事件元素移到「【CLOCK_ON_PATH】【100】【PATH:4】【COUNT:1】」下。 3. 執行「播放」功能。
Expected result	播放之後 GTT 會展開 colors 子樹，依序點選 violet、red 及 blue 節點，之後將 colors 子樹收疊，最後將 JTree 收疊。
Cleanup	按下[關閉 GTT]

Identification	1038.
Name	新增間歇時間事件(sleep time)
Tested target	GTT
Reference	GTT013 GTT015
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 完成 Identification 1034 (新增元件事件元素-JTree)的動作。 2. 在「【CLOCK_ON_PATH】【100】【PATH:2】【COUNT:1】」事件元素後新增一個間歇時間事件元

	<p>素，sleep time 輸入 1000。</p> <p>3. 在「【CLOCK_ON_PATH】【100】【PATH:0】【COUNT:2】」事件元素之前新增一個間歇時間事件元素，sleep time 輸入 1000。</p> <p>4. 執行「播放」功能。</p>
Expected result	播放之後 GTT 會展開 colors 子樹，依序點選 blue，停留 1000ms，才繼續點選 violet、red 及 blue 節點，之後將 colors 子樹收疊，再停留 1000ms 之後，才將 JTree 收疊。
Cleanup	按下[關閉 GTT]

Identification	1039.
Name	新增測試元素
Tested target	GTT
Reference	GTT014
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，操作 AUT，在 TextField B 輸入“abcdef”，之後結束錄影。 5. 在剛錄影下來的事件節點 text b 之後新增一個測試節點。 6. 點選測試節點，新增一個測試元素，選擇 Start with: getText(), Return value 輸入 “abcdef”, Assert Message 輸入 “It’s must be abcdef” 7. 執行「播放」功能。
Expected result	播放成功，測試節點能呈現綠色狀態(assert success)。
Cleanup	按下[關閉 GTT]

Identification	1040.
Name	新增測試元素
Tested target	GTT
Reference	GTT014
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 登入系統 2. 按下[開啟待測應用程式] 3. 由檔案列表中選擇待測應用程式 MyWindow.class 4. 執行錄影功能，操作 AUT，在 TextField B 輸入“abcdef”，之後結束錄影。 5. 在剛錄影下來的事件節點 text b 之後新增一個測試節點。 6. 點選測試節點，新增一個測試元素，選擇 Start with: getText(), Return value 輸入”123456”，Assert Message

	輸入 “It’s must be 123456” 7. 執行「播放」功能。
Expected result	測試節點呈現紅色錯誤狀態(assert failure)。
Cleanup	按下[關閉 GTT]

Identification	1041.
Name	編輯測試元素
Tested target	GTT
Reference	GTT012 GTT015
Severity	1
Instructions	1. 完成 Identification 1040。 2. 點選測試節點，編輯測試元素，將 return value 從 “123456”改成 “abcdef” 3. 執行「播放」功能。
Expected result	播放正常，測試節點呈現綠色無誤狀態(assert success)。
Cleanup	按下[關閉 GTT]

Identification	1042.
Name	刪除測試元素
Tested target	GTT
Reference	GTT011 GTT015
Severity	1
Instructions	1. 完成 Identification 1040。 2. 點選測試節點，將測試元素刪除。 3. 執行「播放」功能。
Expected result	播放正常，測試節點呈現綠色無誤狀態。
Cleanup	按下[關閉 GTT]

Identification	1043.
Name	待測程式位移
Tested target	GTT
Reference	GTT023
Severity	1
Instructions	1. 完成 Identification 1002。 2. 使用滑鼠移動待測程式(計算機)至任意位置。 3. 執行「播放」功能。
Expected result	播放流程不受待測程式位置影響，仍然能正確地依測試流程點按在待測程式上正確的按鈕。
Cleanup	按下[關閉 GTT]

Identification	1044.
Name	自我測試-錄影方式製作測試流程
Tested target	GTT
Reference	GTT042

Severity	1
Instructions	1.
Expected result	GTT 執行過程： 1. 開啟新文件，開啟待測程式。 2. 設定錄影層級為高階。 3. 開始錄影，對待測程式進行實際的操作。 4. 結束錄影，對錄影下來的事件流程做適當的編輯與調整。 5. 儲存編輯完成之待測流程。
Cleanup	按下[關閉 GTT]

Identification	1045.
Name	自我測試-開啟測試流程加入測試方法
Tested target	GTT
Reference	GTT042
Severity	1
Instructions	1. 使用 GTT 開啟 Open Test-Flow and Add Internal-Test.gtt 2. 執行「播放」功能。
Expected result	GTT 執行過程： 1. 開啟已經存在的測試流程。 2. 在選定的事件節點之後加入一個測試方法，設定預期結果與錯誤回傳訊息。 3. 在其它的地方加入測試方法。 4. 結束編輯，儲存編輯完成之事件流程。
Cleanup	按下[關閉 GTT]

Identification	1046.
Name	自我測試-開啟測試流程加入外部測試
Tested target	GTT
Reference	GTT042
Severity	1
Instructions	3. 使用 GTT 開啟 Open Test-Flow and Add External-Test.gtt 4. 執行「播放」功能。
Expected result	GTT 執行過程： 1. 開啟已經存在的測試流程。 2. 在選定的事件節點之後加入一個外部測試，選取外部測試類別與欲執行之單元測試。 3. 在其地方加入外部測試，指定適當的 Test case 及 Test suite。 4. 結束編輯，儲存編輯完成之事件流程。
Cleanup	按下[關閉 GTT]

Identification	1047.
Name	自我測試-執行測試流程並加入新的測試
Tested target	GTT
Reference	GTT042
Severity	1
Instructions	<ol style="list-style-type: none"> 1. 使用 GTT 開啟 Execute Test-Flow and Add New Test-Flow.gtt。 2. 執行「播放」功能。
Expected result	<p>GTT 執行過程：</p> <ol style="list-style-type: none"> 1. 開啟已經存在的測試流程，指定待測程式。 2. 執行測試流程，加入新的測試流程。 3. 再執行測流程，完成之後儲存測試流程。
Cleanup	按下[關閉 GTT]

Section 5. 測試程序與時程(Test Procedure and Schedule)

Deliverable	Responsibility	Completion Date
Develop Test cases	Testers	93/09/15
Test Case Review	Test Lead, Dev. Lead, Testers	93/09/20
Requirements Validation Matrix	Test Lead	93/09/25
Execute manual	Testers & Test Lead	93/09/30
Test Case	Everyone testing the product	93/10/15
Complete Defect Reports	Everyone testing the product	93/10/20
Document and communicate test status/coverage	Test Lead	93/10/20
Document and communicate Beta test status/coverage	Test Lead	93/10/25
Final Test Summary Report	Test Lead	93/10/31

Section 6. 測試結果與分析(Test Results and Analysis)

Test Case #	Results (PASS/FAIL)	Comment
1001.	PASS	
1002.	PASS	
1003.	PASS	
1004.	PASS	
1005.	PASS	
1006.	PASS	
1007.	PASS	
1008.	PASS	
1009.	PASS	
1010.	PASS	
1011.	PASS	
1012.	PASS	
1013.	PASS	
1014.	PASS	

1015.	PASS	
1016.	PASS	
1017.	PASS	
1018.	PASS	
1019.	PASS	
1020.	PASS	
1021.	PASS	
1022.	PASS	
1023.	PASS	
1024.	PASS	
1025.	PASS	
1026.	PASS	
1027.	PASS	
1028.	PASS	
1029.	PASS	
1030.	PASS	
1031.	PASS	
1032.	PASS	
1033.	PASS	
1034.	PASS	
1035.	PASS	
1036.	PASS	
1037.	PASS	
1038.	PASS	
1039.	PASS	
1040.	PASS	
1041.	PASS	
1042.	PASS	
1043.	PASS	
1044.	PASS	
1045.	PASS	
1046.	PASS	
1047.	PASS	
RATE	100%	

字彙(Glossary)

Test Case	<p>Test Case is a commonly used term for a specific test. This is usually the smallest unit of testing. A Test Case will consist of information such as requirements testing, test steps, verification steps, prerequisites, outputs, test environment, etc.</p> <p>A set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.</p>
Test Environment	<p>The hardware and software environment in which tests will be run, and any other software with which the software under test interacts when under test including stubs and test drivers.</p>
Testing	<p>The process of exercising software to verify that it satisfies specified requirements and to detect errors.</p> <p>The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs), and to evaluate the features of the software item (Ref. IEEE Std 829).</p> <p>The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component.</p>
Test Procedure	<p>A document providing detailed instructions for the execution of one or more test cases.</p>
Traceability Matrix	<p>A document showing the relationship between Test Requirements and Test Cases.</p>
Validation	<p>The process of evaluating software at the end of the software development process to ensure compliance with software requirements. The techniques for validation is testing, inspection and reviewing.</p>
Verification	<p>The process of determining whether or not the products of a given phase of the software development cycle meet the implementation steps and can be traced to the incoming objectives established during the previous phase. The techniques for verification are testing, inspection and reviewing.</p>
Equivalence Class	<p>A portion of a component's input or output domains for which the component's behaviour is assumed to be the same from the</p>

component's specification.

Acceptance Testing Testing conducted to enable a user/customer to determine whether to accept a software product. Normally performed to validate the software meets a set of agreed acceptance criteria.

參考文獻 (References)

1. Asbury, Weiner, "Programming with JFC," Wiley Computer Publishing, pp.1-33, 1998
2. Chen, Jessica; Subramaniam, Suganthan, "A GUI Environment to Manipulate FSMs for Testing GUI-based Application in Java," Proc. of the 34th Hawaii International Conference on System Sciences, 2001
3. Chow, Tsun, "Testing Software Design Modeled by Finite-State Machines," IEEE Trans. on Software Engineering, Vol SE-4, #3, pp 178-187, May 1978
4. Geary, David M. , "Graphic JAVA Mastering the AWT 2rd Edition," Sun Microsystems, pp.3-11, pp.93-146, 1997
5. Geary, David M. , "Graphic JAVA Mastering the JFC 3rd Edition," Sun Microsystems, pp.3-121, 1999
6. Gerrard, Paul, "Testing GUI Applications," EuroSTAR '97, pp 24-28 November 1997, Edinburgh UK.
7. imbus GmbH, "Tools for Test Automation," imbus GmbH home page, <http://www.imbus.de>
8. Jemmy Module, <http://jimmy.netbeans.org/>
9. JFC Unit, <http://sourceforge.net/project/jfcunit/>
10. JUnit, <http://www.junit.org/>
11. Kepple, Laurence, "The Black Art of GUI Testing automated testing in an event-driven environment," Dr. Dobb's Journal, pp 40-46, Feb 1994
12. Linz, Tilo; Daigl, Matthias, "How to Automate Testing of Graphical User Interface," <http://www.imbus.de>
13. Linz, Tilo; Daigl, Matthias; "GUI Testing Made Painless," <http://www.imbus.de>
14. Manipulate EventQueue for semimodal dialogs, <http://www.javaworld.com/javaworld/javatips/jw-javatip89.html>
15. Memon, A.M.; Pollack, M.E.; Soffa, M.L, "Hierarchical GUI test case generation using automated planning," Software Engineering, IEEE Transactions on , Volume: 27 Issue: 2 , pp 144-155, Feb. 2001
16. Memon, A.M.; Pollack, M.E.; Soffa, M.L, "Using a Goal-Driven Approach to Generate Test Cases for GUIs," Proc. 21st Int'l Conf. Software Engineering, pp 257-266, May. 1999
17. Mercury-Interactive's WinRunner, <http://www-svca.mercuryinteractive.com/products/winrunner/>, Mercury Interactive Corporation, Sunnyvale, CA
18. Newmarch, J. , "Testing Java Swing-based applications," In the 31st International Conference on Technology on Object-Oriented Language and Systems, Nanjing, China, September 1999
19. Niemeyer; Peck, "Exploring Java," O'REILLY, pp.297-348, 1996
20. Ostrand, Thomas; Anodidi, Aaron; Foster, Herber; Goradia, Tarak, "A Visual Test Development Environment for GUI Systems," ISSSTA 98 Clearwater Beach Florida USA
21. Segue's SilkTest, <http://www.segure.com/index.htm>
22. Shehady, Ricard; Siewiorek, Daniel, "A Method to Automate User Interface Testing Using Variable Finite State Machines," Proc. of the 27th Int. Symposium on Fault Tolerant Computing, Seattle, WA, pp80-88, June, 1997
23. SQA Robot, <http://www.rational.com/products/robot>, Rational Software Corp., Cupertino, CA
24. The Life Cycle of a Java Event in JDK 1.1, <http://mindprod.com/event11.html>
25. Walrath; Campione, "The JFC Swing Tutorial," Sun Microsystems, pp.5-56, 1998
26. Walworth, Alan, "Java GUI Testing automated testing is as problematic as it is essential," Dr. Dobb's Journal, pp 30-34, Sept. 1992
27. White, Lee; Almezen, Husain, "Generating Test Cases for GUI Responsibilities Using Complete Interaction Sequence," Software Reliability Engineering, 2000. ISSRE 2000. Proceedings. 11th International Symposium on, 2000.
28. Satadip Dutta, "Abbot – A Friendly JUnit Extension for GUI Testing," Java Developers Journal, April 2003. (<http://abbot.sourceforge.net/>).

附錄一. 追溯表(Traceability Matrix)

Req. No.	Test Spec.	Test Case #	Verification
GTT001 GTT018	GTT Test Plan	1001.	Verified
GTT002 GTT004 GTT005 GTT006	GTT Test Plan	1002.	Verified
GTT002 GTT006	GTT Test Plan	1003.	Verified
GTT017	GTT Test Plan	1004.	Verified
GTT003	GTT Test Plan	1005.	Verified
GTT019	GTT Test Plan	1006.	Verified
GTT007	GTT Test Plan	1007.	Verified
GTT007	GTT Test Plan	1008.	Verified
GTT010 GTT004 GTT006	GTT Test Plan	1009.	Verified
GTT020 GTT035	GTT Test Plan	1010.	Verified
GTT025	GTT Test Plan	1011.	Verified
GTT025	GTT Test Plan	1012.	Verified
GTT007	GTT Test Plan	1013.	Verified
GTT008	GTT Test Plan	1014.	Verified
GTT009	GTT Test Plan	1015.	Verified
GTT011 GTT026	GTT Test Plan	1016.	Verified
GTT026	GTT Test Plan	1017.	Verified
GTT026	GTT Test Plan	1018.	Verified
GTT026	GTT Test Plan	1019.	Verified
GTT026	GTT Test Plan	1020.	Verified
GTT025	GTT Test Plan	1021.	Verified
GTT025	GTT Test Plan	1022.	Verified
GTT026	GTT Test Plan	1023.	Verified
GTT026	GTT Test Plan	1024.	Verified
GTT026	GTT Test Plan	1025.	Verified
GTT001 GTT013 GTT021	GTT Test Plan	1026.	Verified
GTT001 GTT013	GTT Test Plan	1027.	Verified

GTT021			
GTT001 GTT013 GTT021	GTT Test Plan	1028.	Verified
GTT001 GTT013 GTT021	GTT Test Plan	1029.	Verified
GTT001 GTT013 GTT021	GTT Test Plan	1030.	Verified
GTT001 GTT013 GTT021	GTT Test Plan	1031.	Verified
GTT001 GTT013 GTT021	GTT Test Plan	1032.	Verified
GTT001 GTT013 GTT021	GTT Test Plan	1033.	Verified
GTT001 GTT013 GTT021	GTT Test Plan	1034.	Verified
GTT001 GTT013 GTT021	GTT Test Plan	1035.	Verified
GTT016	GTT Test Plan	1036.	Verified
GTT013 GTT015	GTT Test Plan	1037.	Verified
GTT013 GTT015	GTT Test Plan	1038.	Verified
GTT014	GTT Test Plan	1039.	Verified
GTT014	GTT Test Plan	1040.	Verified
GTT012 GTT015	GTT Test Plan	1041.	Verified
GTT011 GTT015	GTT Test Plan	1042.	Verified
GTT023	GTT Test Plan	1043.	Verified
GTT042	GTT Test Plan	1044.	Verified
GTT042	GTT Test Plan	1045.	Verified
GTT042	GTT Test Plan	1046.	Verified
GTT042	GTT Test Plan	1047.	Verified