

A tutorial of GTT

Author: Tung-Hung Tsai, Cheng-Wen Shen

1. A simple AUT

We use a simple example to describe how to use GTT. An application program often has a Toolbar, which has a lot of buttons. We have designed a simple example (AUT) that can be adjusted to enable or disable some buttons. We use this example to explain how to use GTT step by step. First, we introduce the windows of the AUT below.

- Figure 1 shows the main window of the AUT (App main window). It contains a Menubar and a Toolbar. There are three choices on the Menubar: File, Option and Help. There are five buttons on the Toolbar: ButtonA, ButtonB, ButtonC, ButtonD and ButtonE.
- Figure 2 shows the steps in opening the “Toolbar Property” dialog (option→Preference→Toolbar).
- Figure 3 shows the “Toolbar Property” dialog. It contains several radio buttons that are used to enable or disable Toolbar buttons. It also contains a SetDefault Button, an Apply Button and a Cancel Button.
- Figure 4 shows the “Option” confirmation dialog. It is shown when SetDefault Button is clicked. Its purpose is to enable the user to confirm again whether to set radio buttons as their default values.



Figure 1 App main window

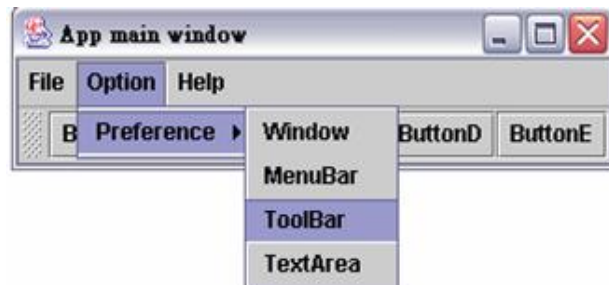


Figure 2 Steps to open ToolbarProperty Dialog



Figure 3 ToolbarProperty DialogBox

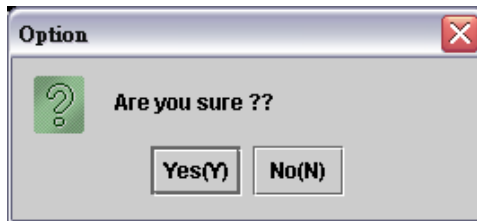



Figure 4 Option Confirm DialogBox

[Back](#)

2 Capture/Replay

1. Open the AUT

After GTT starts, click  to open the java class file of the AUT. The GTT Test Sequence editing area will present a root node called StateModify (See Figure 5).

2. Rename the root node

We can rename StateModify node as MyTest node by selecting the StateModify node, right clicking the button of the mouse, and selecting the Set Name property (See Figure 6).

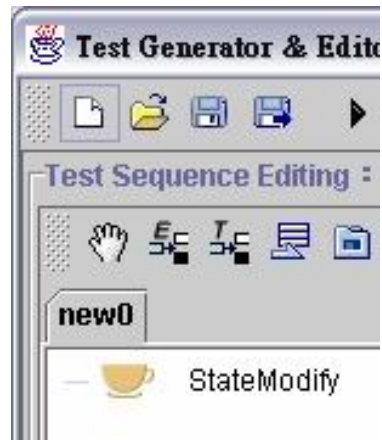


Figure 5 Window of GTT started

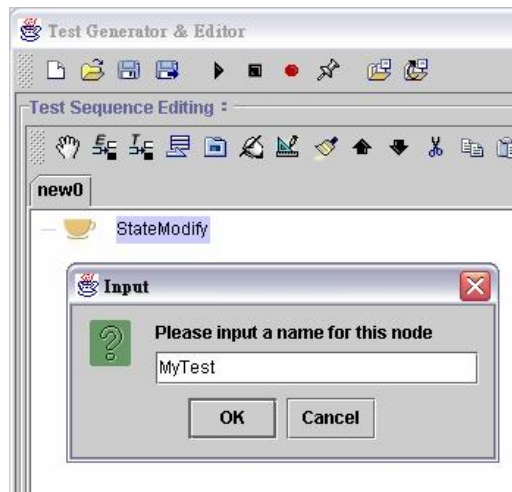






Figure 6 Rename the node


3. **Begin capture**

Every action (event) on the AUT will be captured after clicking . We select MyTest node and click  to begin capture. We use the mouse and keyboard to interact with the AUT. We produce interactions according to the procedure of Table 1.

4. **Stop capture and save scripts**




We can click  to stop recording. A dialog will shown to let us save test flow before, within or after the Mytest node. Choose within to put the captured events into the MyTest node. The captured events area is shown in Figure 7.It can be saved as script file by clicking  button.

5. **Create a Folder node**

Folder nodes can be used to collect continuous test flows. We select the Option node, click  to add a new folder node, name it as "Option→Preference→Toolbar" (See Figure 8). Now we can use the

new folder node to collect Option, Preference and Toolbar nodes.

6. **Cut and paste the selected nodes**

Now we press ctrl key to select Option, Preference and Toolbar, and then click . Then select the Option→Preference→Toolbar node, and click the node's  icon. Then click  to paste the previously cut nodes into the Option→Preference→Toolbar folder. It will become a node that collects Option, Preference and Toolbar event nodes (See Figure 9).




7. **Repeat 5, 6 steps**

We repeat 5 and 6 step to process the next Option, Preference and Toolbar nodes (See Figure 10).

8. **Collect Disable B, Disable D and Disable E nodes**

We can collect these three nodes into a folder. The steps are the same as 5, 6 and 7 steps (See Figure 11)

9. **Replay the test flow**

We have already finished the design of test flow. We can click  to replay the whole test flow (See Figure 12). We can also click  to reset the execution results or click  to restore the AUT to its original state.

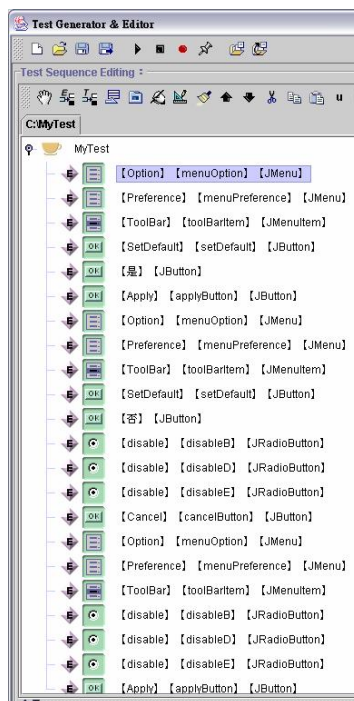
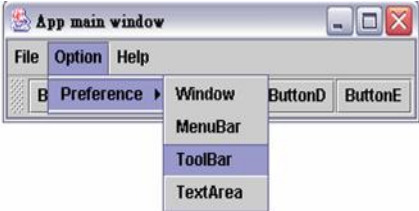

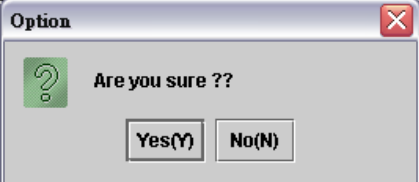

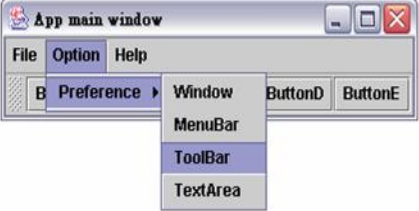

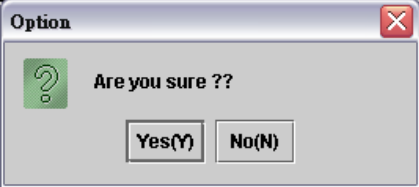

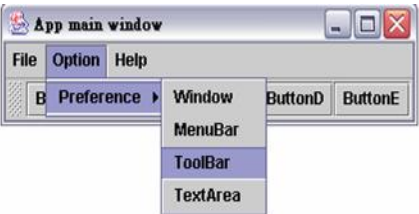



Figure 7 Test flow after capture

Table 1 Interacts on AUT

steps #	interactions	Picture of the AUT
1.	<ul style="list-style-type: none"> ● Option ● Preference ● Toolbar 	
2.	<ul style="list-style-type: none"> ● SetDefault 	
3.	<ul style="list-style-type: none"> ● Yes 	
4.	<ul style="list-style-type: none"> ● Apply 	
5.	<ul style="list-style-type: none"> ● Option ● Preference ● Toolbar 	

6.	<ul style="list-style-type: none"> ● SetDefault 	 <p>The dialog box titled 'ToolBar property' has a 'Selection:' section with a 'SetDefault' button. Below are five rows, each with a button label and two radio buttons: 'enable' (selected) and 'disable'. The buttons are ButtonA, ButtonB, ButtonC, ButtonD, and ButtonE. At the bottom are 'Apply' and 'Cancel' buttons.</p>
7.	<ul style="list-style-type: none"> ● No 	 <p>The dialog box titled 'Option' contains a question mark icon and the text 'Are you sure ??'. At the bottom are 'Yes(Y)' and 'No(N)' buttons.</p>
8.	<ul style="list-style-type: none"> ● Disable B ● Disable D ● Disable E ● cancel 	 <p>The dialog box titled 'ToolBar property' has a 'Selection:' section with a 'SetDefault' button. Below are five rows, each with a button label and two radio buttons: 'enable' and 'disable'. The buttons are ButtonA, ButtonB, ButtonC, ButtonD, and ButtonE. At the bottom are 'Apply' and 'Cancel' buttons.</p>
9.	<ul style="list-style-type: none"> ● Option ● Preference ● Toolbar 	 <p>The screenshot shows the 'App main window' with a menu bar containing 'File', 'Option', and 'Help'. The 'Option' menu is open, showing a sub-menu with 'Preference', 'Window', 'MenuBar', 'ToolBar', and 'TextArea'. On the right side of the window, there are buttons for 'ButtonD' and 'ButtonE'.</p>
10.	<ul style="list-style-type: none"> ● Disable B ● Disable D ● Disable E ● Apply 	 <p>The dialog box titled 'ToolBar property' has a 'Selection:' section with a 'SetDefault' button. Below are five rows, each with a button label and two radio buttons: 'enable' and 'disable'. The buttons are ButtonA, ButtonB, ButtonC, ButtonD, and ButtonE. At the bottom are 'Apply' and 'Cancel' buttons.</p>

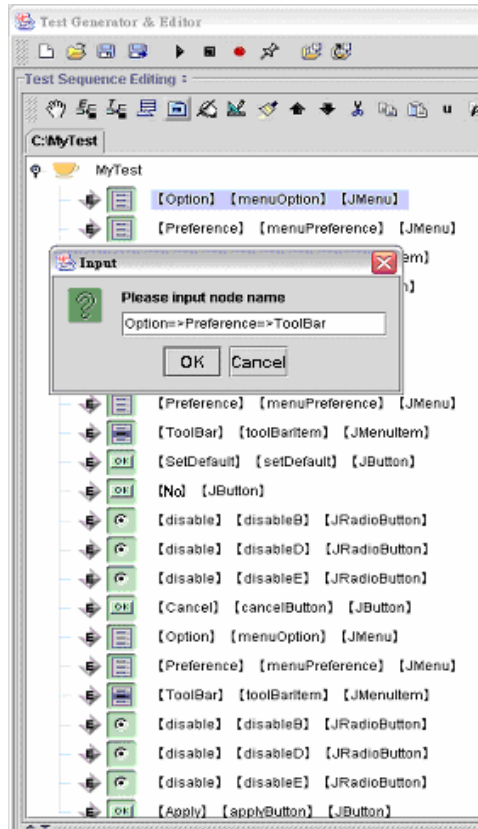


Figure 8 Add a new folder node

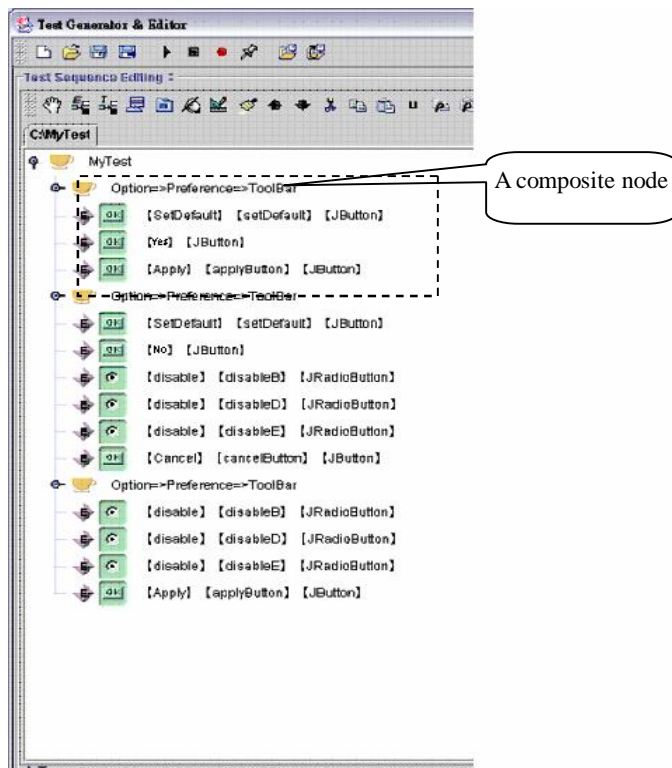


Figure 9 Node of composite event nodes

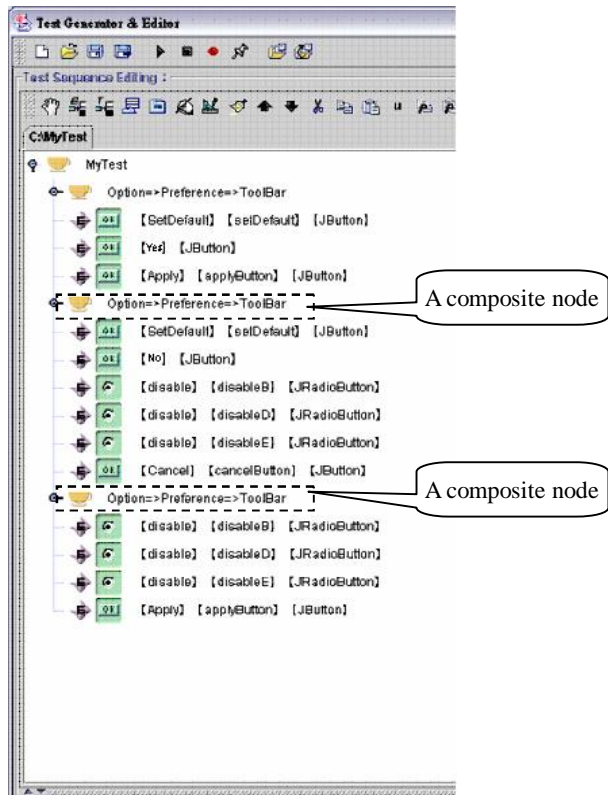


Figure 10 Node of composite event nodes.

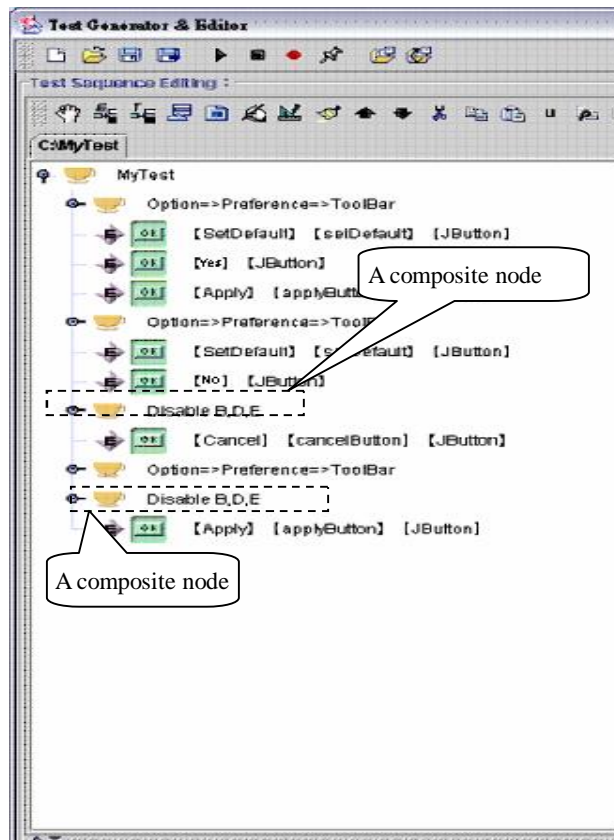


Figure 11 Collect Disable B, Disable D and Disable E nodes.

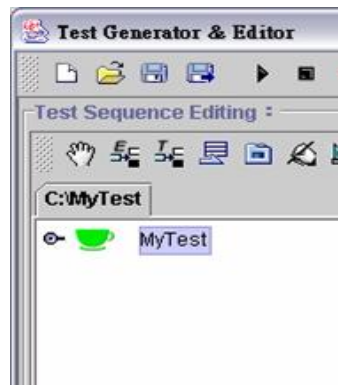


Figure 12 Correct result of reply

[Back](#)

3 Test node and test method

1. Insert test node

We add a new folder (📁) before the Option→Preference→Toolbar node, and rename the folder as T1. Then click the “insert test node” (📁) button, and GTT will display the five button components in the “Test Generator Select Components for Insert Components Checks Dialog” (See Figure 13). Now we can use the dialog to select components that are to be tested with test methods to each component. We use (➤) to add the five components from AUT to the “Input what you need” area. And use (⊕) to open the “Test Generator Modify Check” dialog to add test methods to each component on the “Input what you need” area. Now let us add a test method: isVisible method. First, we select the isVisible method from second Dropdown List prefixed with “is”. And select true for the return value. At last, we add an Assert Message “Button A at this time must be visible.”


2. Add test method in the test flow

After we click ok button on the dialog. GTT will put the test nodes (with test methods) into the folder (See Figure 14).


3. Add other test nodes by copying and pasting

After selecting the T1 node. We can click (📄) to copy it. And then we use (📄) to paste the copy into the position to be tested. But the state of T3 is not the same as T1, the Button B, Button D and Button E are invisible. However, we can use the “The Events or Tests” area of the main window to modify or add the test method of test node and change the node name if necessary.

4. Add test node for “Toolbar Property” dialog

We click  to add a new test node (named as T2). Then open the “Toolbar property” dialog (Option→Preference→Toolbar) of AUT, and select components to be tested. We need to verify whether these radio buttons at this moment are selected or not. Therefore, we use the `isSelected()` test method, then add testing for five components of `enable radioButton` and `disable radioButton` (See Figure 15).

5. Replay the final result

We have already finished the design of test flow. We can click  to replay the test flow. The GTT will show green light on each test node when it is replayed correctly (See Figure 16).

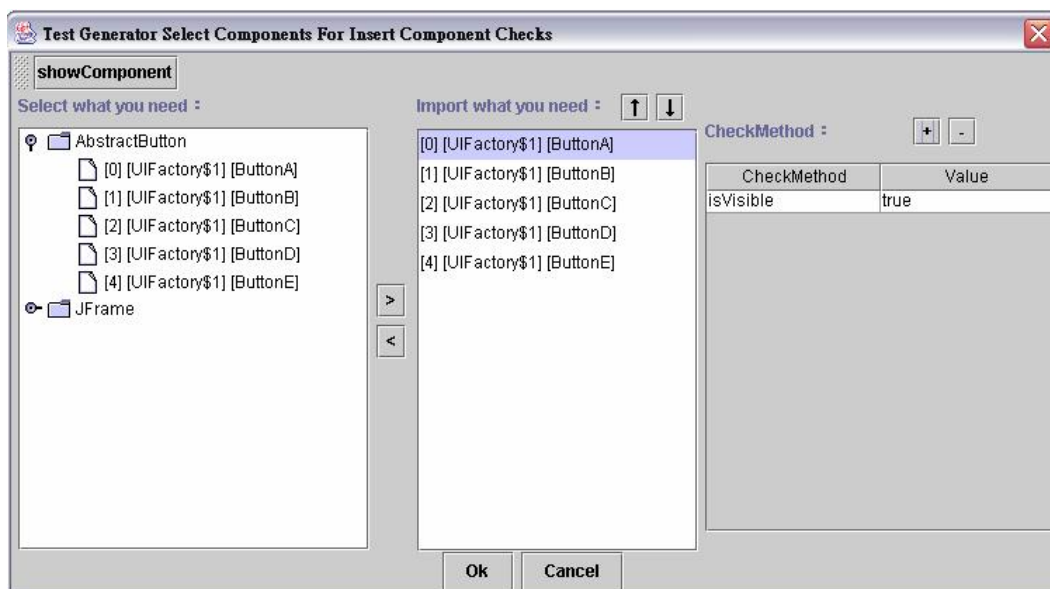


Figure 13 Add test method for component

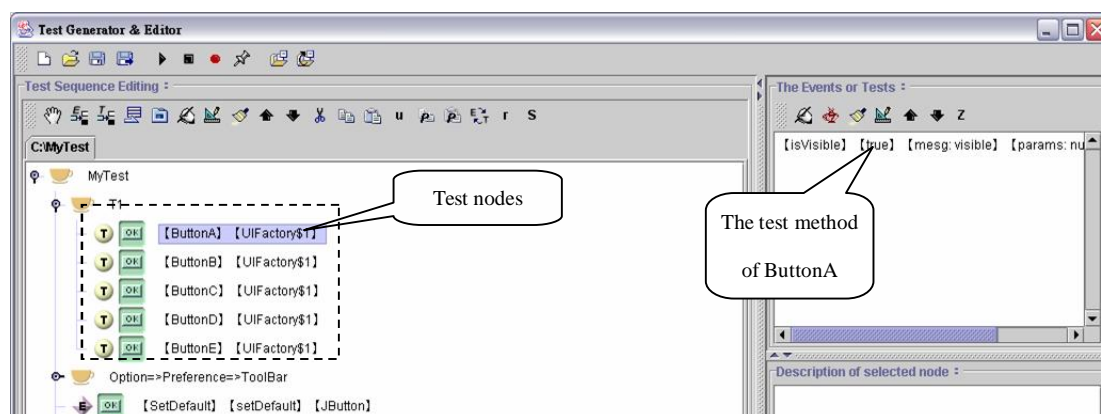


Figure 14 Add the node with test method

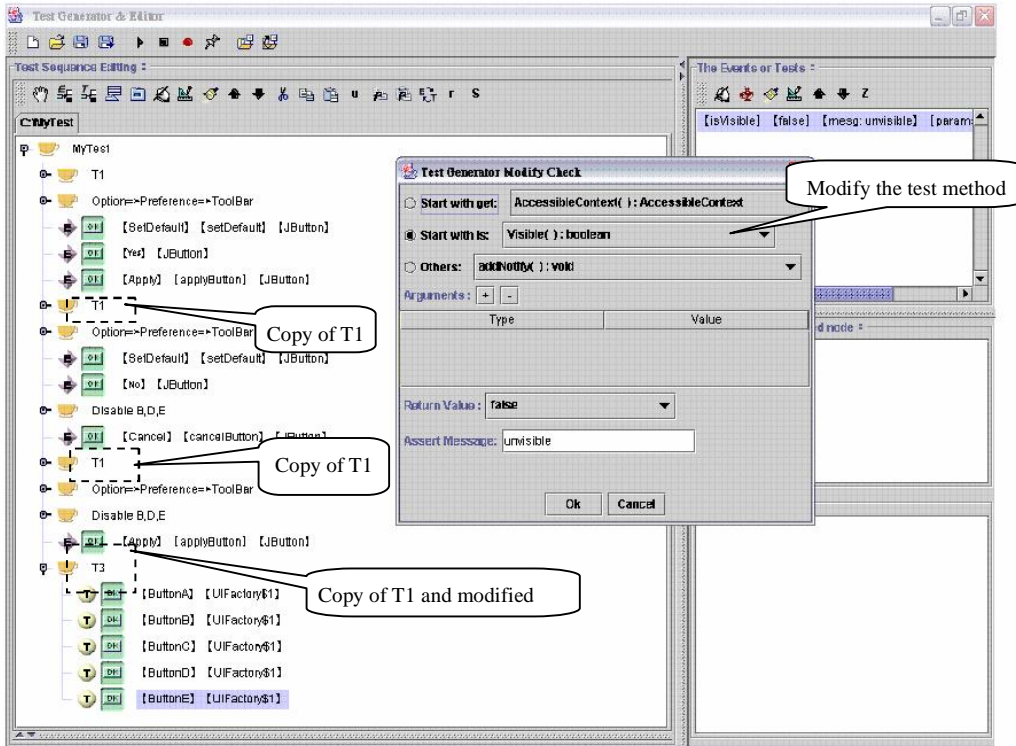


Figure 15 Test nodes

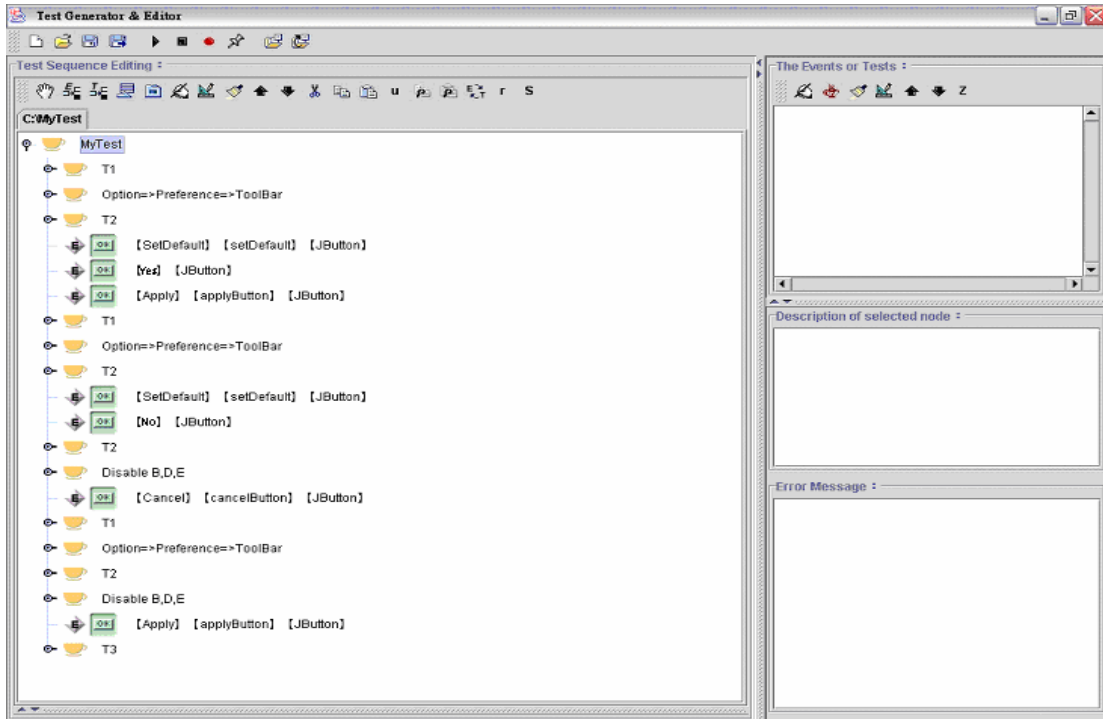


Figure 16 The last result of test flow

[Back](#)