

Practical No. 07

Aim: Create any GUI application e.g. calculator and automate using autoIT v3 toolkit

Practical No. 07

Aim: Create any GUI application e.g. Calculator and automate using autoit V3 tool.

Theory :

AutoIt is constantly evolving as a programming language. It started as an add-on tool to automate basic tasks in GUI's of other programs, and task automation (such as sending a keystroke or clicking a button.) is still at the heart of AutoIt. With the introduction of many new features, such however, AutoIt has become a more powerful tool than ever before.

Just a few of the new and updated features include :

- GUI Automation - Create a custom graphical interface for your application.
- COM (object) functionality fills the gap of with WSH languages such as VBScript / JScript .
- Loops, Functions and expression parsing.
- An enormous number of functions handling and manipulating strings.
- A Perl-compatible regular expression engine using the PCRE Library , with native 16 bit mode and UCS/UTF support .
- A powerful recursive File List to Array function .
- Easily call Win32 and third-party DLL APIs from within your script .

Calc GUI :

```
#include <EditConstants.au3>
#include <GUIConstantsEx.au3>
#include <StaticConstants.au3>
#include <WindowsConstants.au3>

GUICreate("Calculator", 260, 230)
$idBtn1 = GUICtrlCreateButton("1", 54, 138, 36, 29)
Local $idBtn0 = GUICtrlCreateButton("0", 54, 171, 36, 29)
Local $idBtn1 = GUICtrlCreateButton("1", 54, 138, 36, 29)
Local $idBtn2 = GUICtrlCreateButton("2", 93, 138, 36, 29)
Local $idBtn3 = GUICtrlCreateButton("3", 132, 138, 36, 29)
Local $idBtn4 = GUICtrlCreateButton("4", 54, 106, 36, 29)
Local $idBtn5 = GUICtrlCreateButton("5", 93, 106, 36, 29)
Local $idBtn6 = GUICtrlCreateButton("6", 132, 106, 36, 29)
Local $idBtn7 = GUICtrlCreateButton("7", 54, 73, 36, 29)
Local $idBtn8 = GUICtrlCreateButton("8", 93, 73, 36, 29)
Local $idBtn9 = GUICtrlCreateButton("9", 132, 73, 36, 29)
Local $idBtnPeriod = GUICtrlCreateButton(".", 132, 171, 36, 29)

Local $idBtnMClear = GUICtrlCreateButton("MC", 8, 73, 36, 29)
Local $idBtnMRestore = GUICtrlCreateButton("MR", 8, 106, 36, 29)
Local $idBtnMStore = GUICtrlCreateButton("MS", 8, 138, 36, 29)
Local $idBtnMAdd = GUICtrlCreateButton("M+", 8, 171, 36, 29)

Local $idBtnChangeSign = GUICtrlCreateButton("+/-", 93, 171, 36, 29)
Local $idBtnDivision = GUICtrlCreateButton("/", 171, 73, 36, 29)
Local $idBtnMultiplication = GUICtrlCreateButton("*", 171, 106, 36, 29)
```

Local \$idBtnSubtract = GUICreateButton (" - ", 171, 138, 36, 29)
Local \$idBtnAddition = GUICtrlCreateButton (" + ", 171, 171, 36, 29)
Local \$idBtnAnswer = GUICtrlCreateButton (" = ", 210, 171, 36, 29)
Local \$idBtnInverse = GUICtrlCreateButton (" 1/x ", 210, 138, 36, 29)
Local \$idBtnSqrt = GUICtrlCreateButton (" sqrt ", 210, 73, 36, 29)
Local \$idBtnPercentage = GUICtrlCreateButton (" .% ", 210, 106, 36, 29)
Local \$idBtnBackspace = GUICtrlCreateButton (" backspace ", 54, 37, 63, 29)
Local \$idBtnClearE = GUICtrlCreateButton (" CE ", 120, 37, 62, 29)
Local \$idBtnClear = GUICtrlCreateButton (" C ", 185, 37, 62, 29)
Local \$idEditScreen = GUICtrlCreateButton (" 0. ", 8, 2, 239, 23)
Local \$idLbMemory = GUICtrlCreateButton (" ", 12, 39, 27, 26)
GUI SetState ()
Local \$msg
DO
\$msg = GUIGetMsg()
Until \$msg = \$GUI_EVENT_CLOSE

```
Local $idEdtScreen = GUICtrlCreateButton("0.", 8, 2, 239, 23, BitOr($ES_READONLY,  
$ES_RIGHT), $WS_EX_STATICEDGE)  
Local $idLblMemory = GUICtrlCreateButton("", 12, 39, 27, 26, $SS_SUNKEN)
```

Operations:

```
$1 = InputBox("Maths", "Number:", "")
```

\\$operation = InputBox("Mathe", "+,-,*,:","")

S 2 = InputBox ("Maths", "plus number:", "")

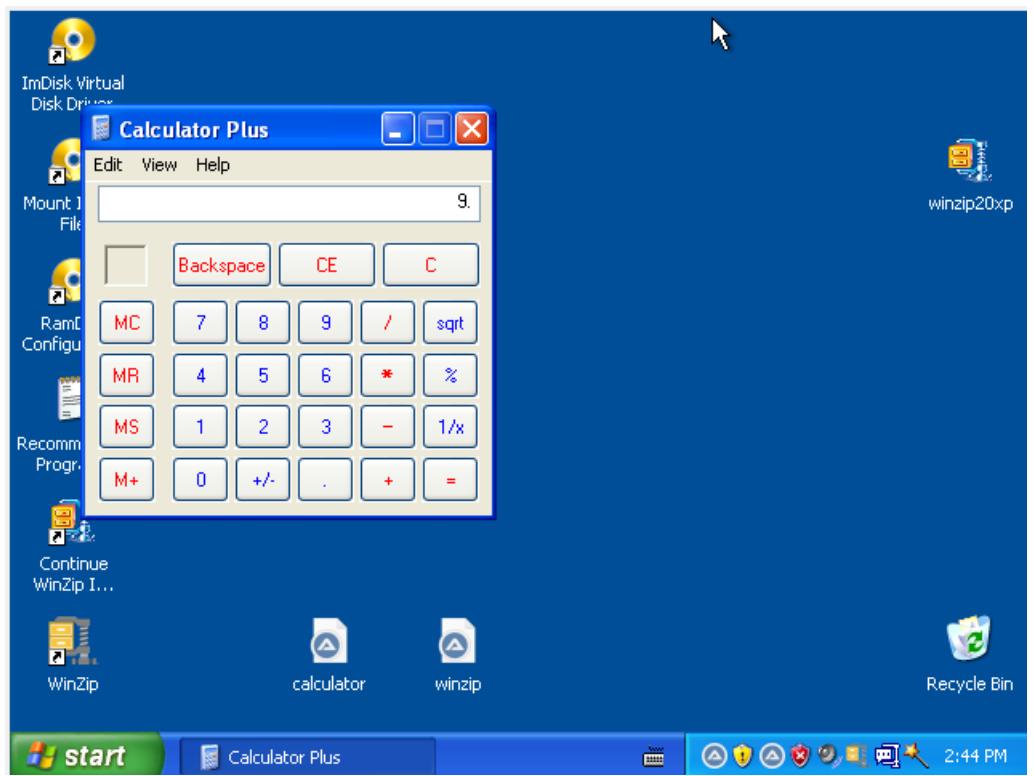
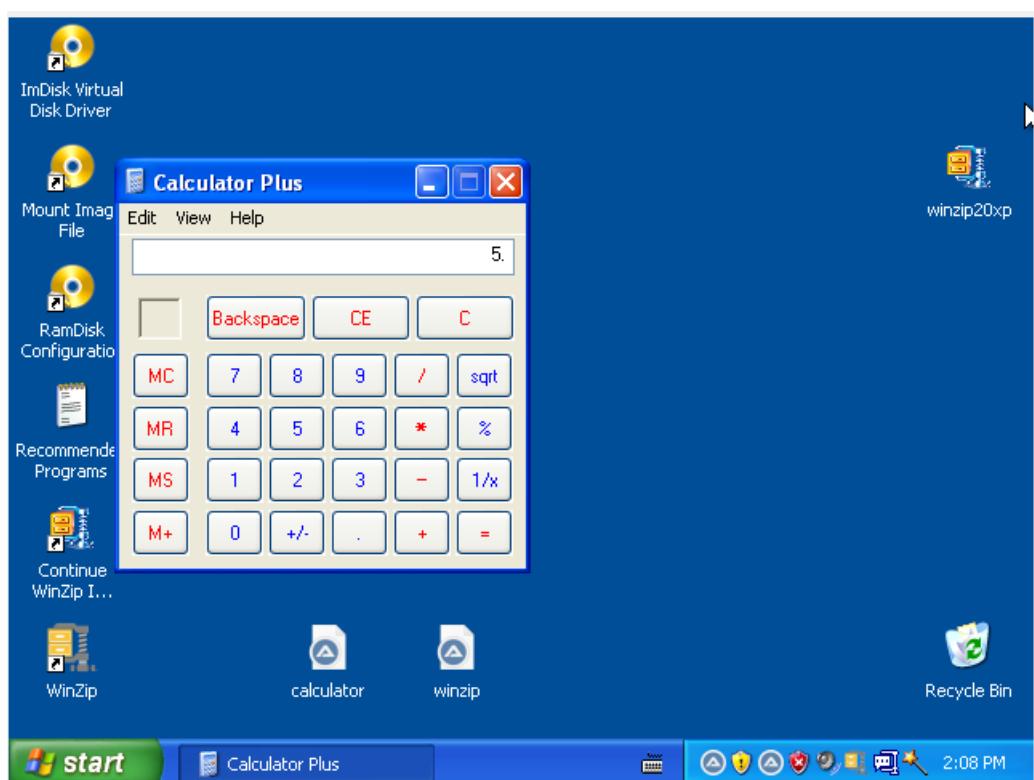
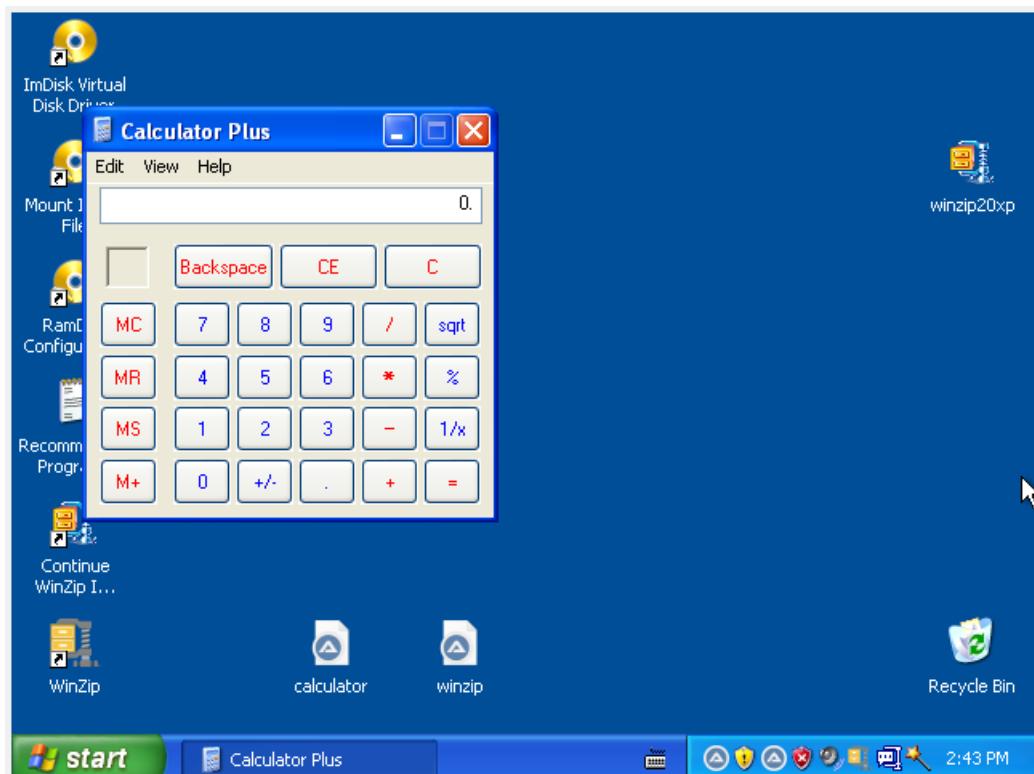
$$\$plus = \$1 + \$2$$

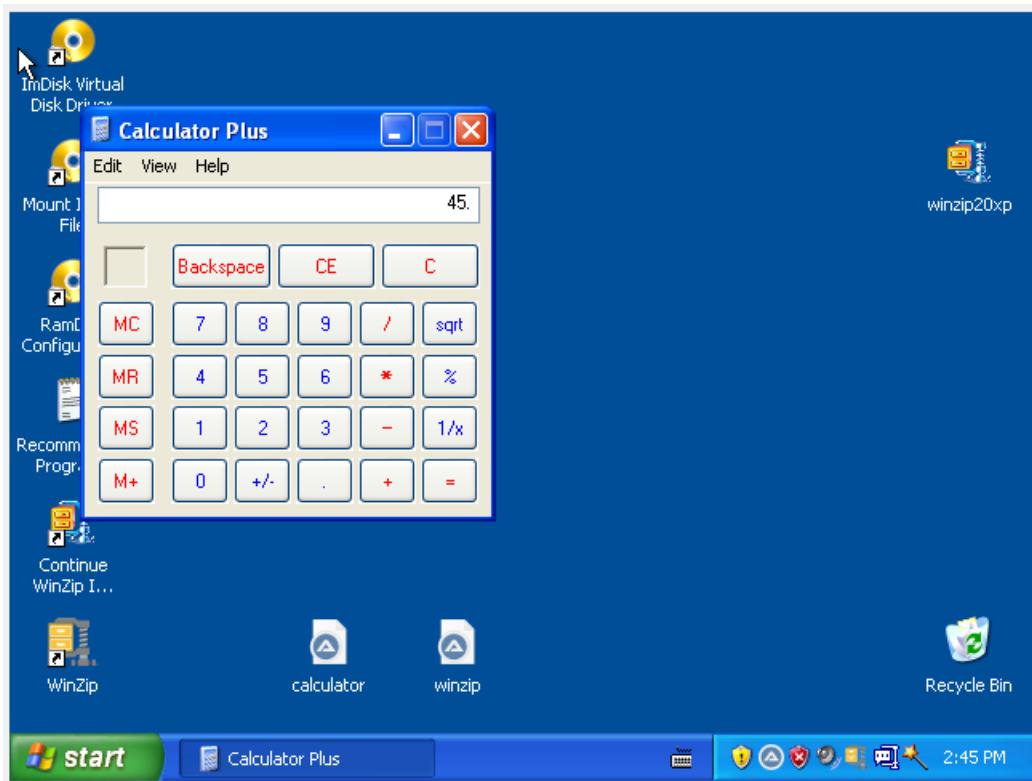
$$\$_{\text{minus}} = \$1 - \$2$$

$$\$times = \$1 * \$2$$

Conclusion:

Thus, we have studied and executed a program in AutoIT v3 to create calculator and automated it using AutoIT v3.





```
IF $operation = "+" Then  
    MsgBox ("", "Maths", "=" & $plus)
```

ENDIF

```
IF $operation = "-" Then  
    MsgBox ("", "Maths", "=" & $minus)
```

ENDIF

```
IF $operation = "*" Then  
    MsgBox ("", "Maths", "=" & $times)
```

ENDIF

Automate :

```
Run ("Calc.exe", "")
```

```
Sleep (2000)
```

```
In WinExists ("Calculator", "") Then
```

```
ControlClick ("Calculator", "", 129)
```

```
Sleep (1000)
```

```
ControlClick ("Calculator", "", 92)
```

```
Sleep (1000)
```

```
ControlClick ("Calculator", "", 133)
```

```
Sleep (1000)
```

```
ControlClick ("Calculator", "", 112)
```

```
Sleep (1000)
```

Conclusion :

Control :

Thus, we have studied and Executed a Program in AutoIt v3 to create a Calculator and automate it using AutoIt v3.

Practical No. 08

Aim : Automate notepad Application using AutoIT.

(Write "Hello" in notepad)

Run("notepad")

(write("Hello"))

Run("notepad")

(write("Hello"))

Send

: Done

(*,"Hello"))

(exit) 0

Run("notepad")

(write("Hello"))

(control)

(left,"Hello"))

(control)

(left,"Hello"))

(control)

(left,"Hello"))

(control)

Run("notepad")

(write("Hello"))

Practical No. 08

Aim: Automate notepad application using AutoIT.

Theory:

First step, we need to know is the name of the Notepad executable. It is a notepad.exe — you can get this information by looking at the properties of the Notepad shortcut in the Start Menu.

To execute

Notepad we use the AutoIT Run function. This function simply launches a given executable and then continues.

Type in the first line of script as:

Run("Notepad.exe")

Run the script - if all goes well then a new instance of Notepad should open. When automating applications AutoIT can check for their window title so that it knows which windows it should work with. With Notepad the window title is obviously Untitled-Notepad.

AutoIT is case sensitive when using window titles so you must get the title exactly right - the best way to do this is to use the AutoIT window AutoIT v3 Window Info. Run the AutoIT v3 Window info from Start Menu / AutoIT v3 / AutoIT Window Info.

With the info Tool open click on the newly opened Notepad window to activate it; the Info Tool will give you information about it. The Information we are interested in is the window title.

Enter the following as the second line in

in the script (use CTRL-V or Edit/Paste to paste our window title from the clipboard.)

WinWaitActive("Untitled - Notepad")

After we are sure the Notepad window is visible we want to type in some text. This is done with the Send Function.

Add this line to our Script.

Send("This is some text.")

Next we want to close Notepad, we can do this with the WinClose function.

WinClose("Untitled - Notepad")

So, we add a line to wait for this dialog to become active (we will also use the window text to make the function more reliable and to distinguish this new window from the original Notepad window):

WinWaitActive("Notepad", "Save")

Next we want to automatically press Alt-N to select the No/Don't save button (Underlined letters in windows usually indicate that you can use the ALT key and that letter as a keyboard shortcut.) In the Send function to send an ALT key we use!

Our complete Script now looks like this:

Run("Notepad.exe")

WinWaitActive("Untitled - Notepad")

Send("This is some test regarding testing")

WinClose("Untitled - Notepad")

WinWaitActive("Notepad", "Save")

Send("!n")

Script to automate Notepad :

```
Run ("Notepad.exe")
WinWaitActive ("Untitled - Notepad")
Send ("This is some text regarding testing")
WinClose ("Untitled - Notepad")
WinWaitActive ("Notepad", "Save")
Send ("!s")
Send ("test.txt")
Send ("!s")
```

Script to copy text from one file to another.

```
Opt ("WinTitleMatchMode", 2)
WinActivate ("old")
WinWaitActive ("old")
Send ("{HOME}{3}")
Send ("+{END}{3}")
Send ("^c")
WinWaitActive ("new")
WinActivate ("new")
Send ("^v")
Send ("{ENTER}{3}")
```

• Diagnostic techniques in Java

• Java Debug Tool

• ("Breakpoint - Breakpoint") and "Breakpoint"

• ("Next instruction", "Step into", "Step over")

• ("Threads - Thread 0") and "Threads"

• ("Call stack") and "Call stack"

• ("Registers") and "Registers"

• ("Registers") and "Registers"

• reading file from external application

• Java Application's File

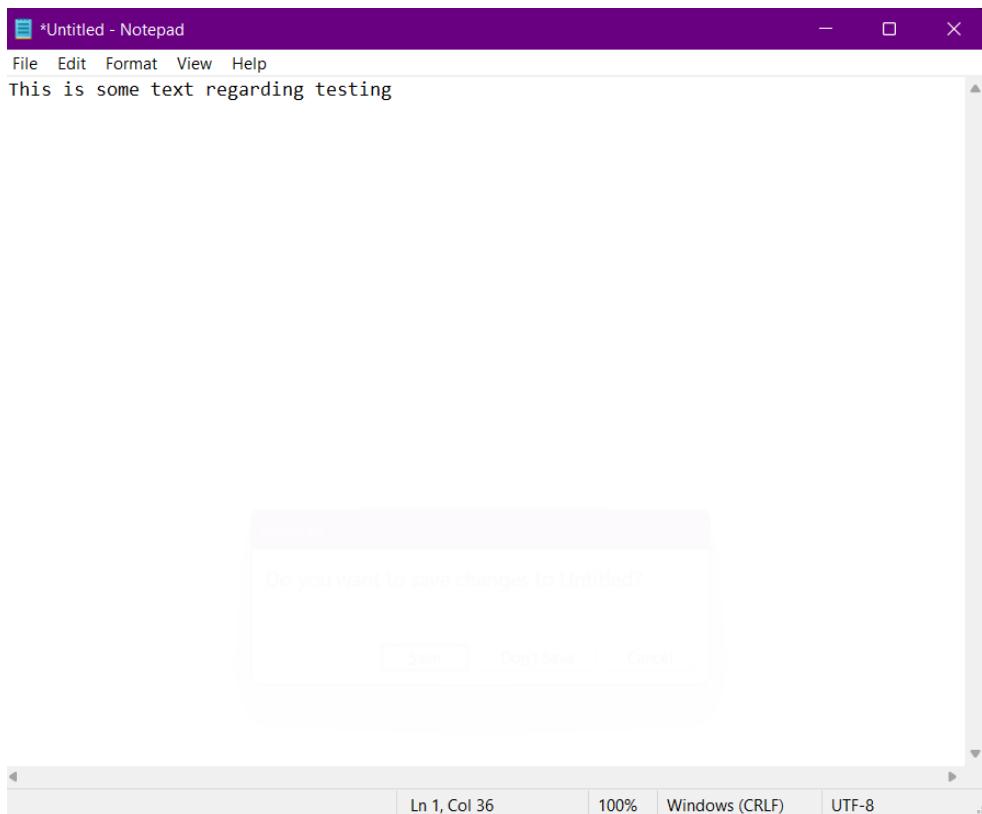
• ("File") and "File"

• ("Open file") and "Open file"

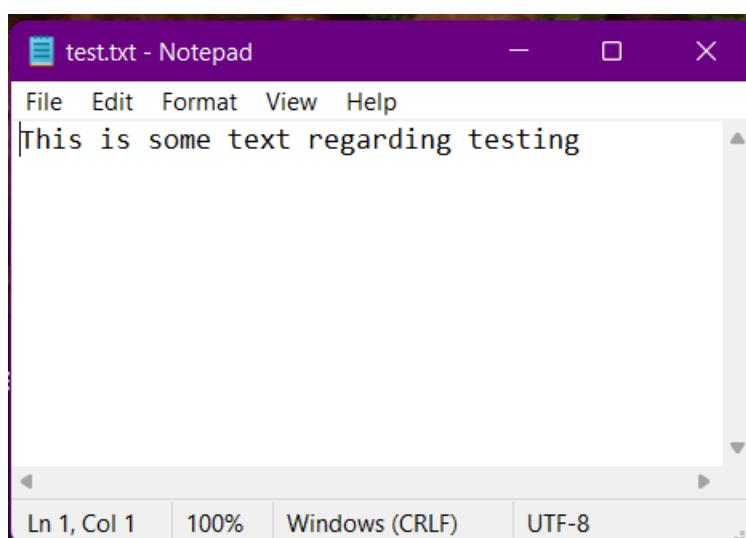
Conclusion:

We learnt technique in this experiment to automate windows notepad Application.

Automating notepad using AutoIt

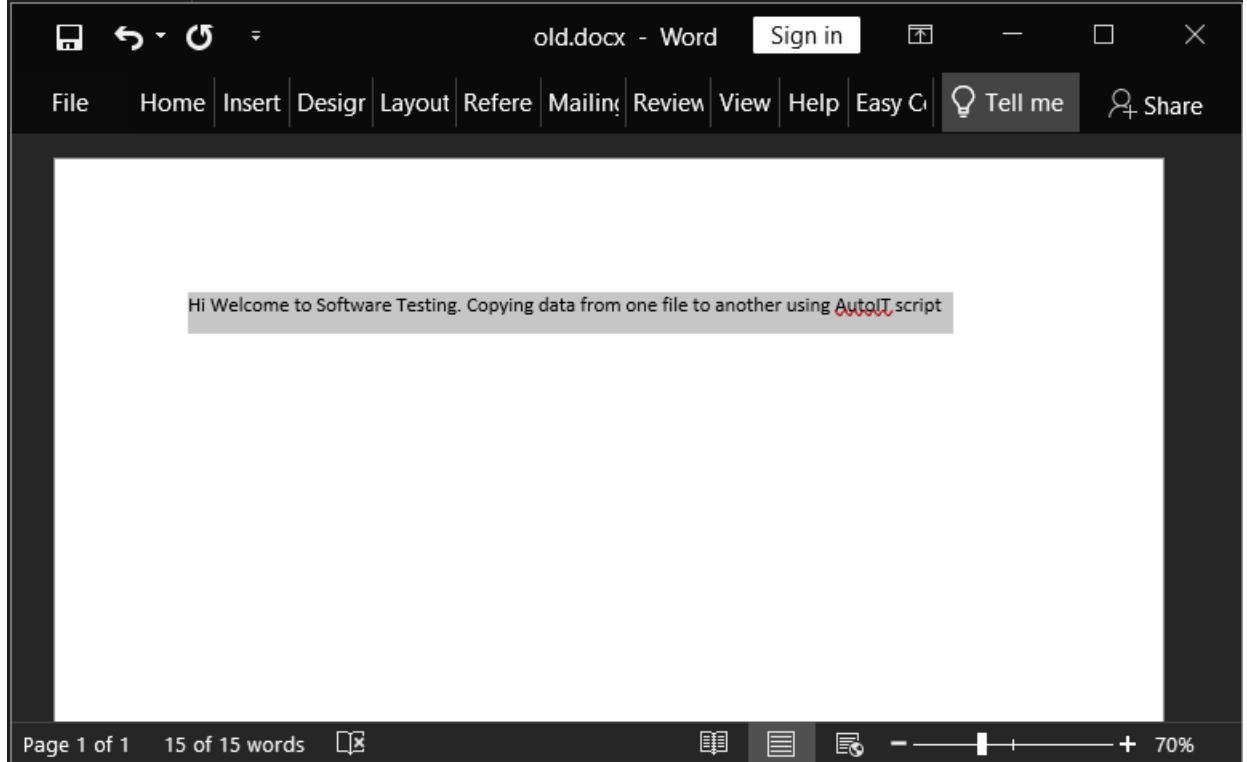


Saved file using automation

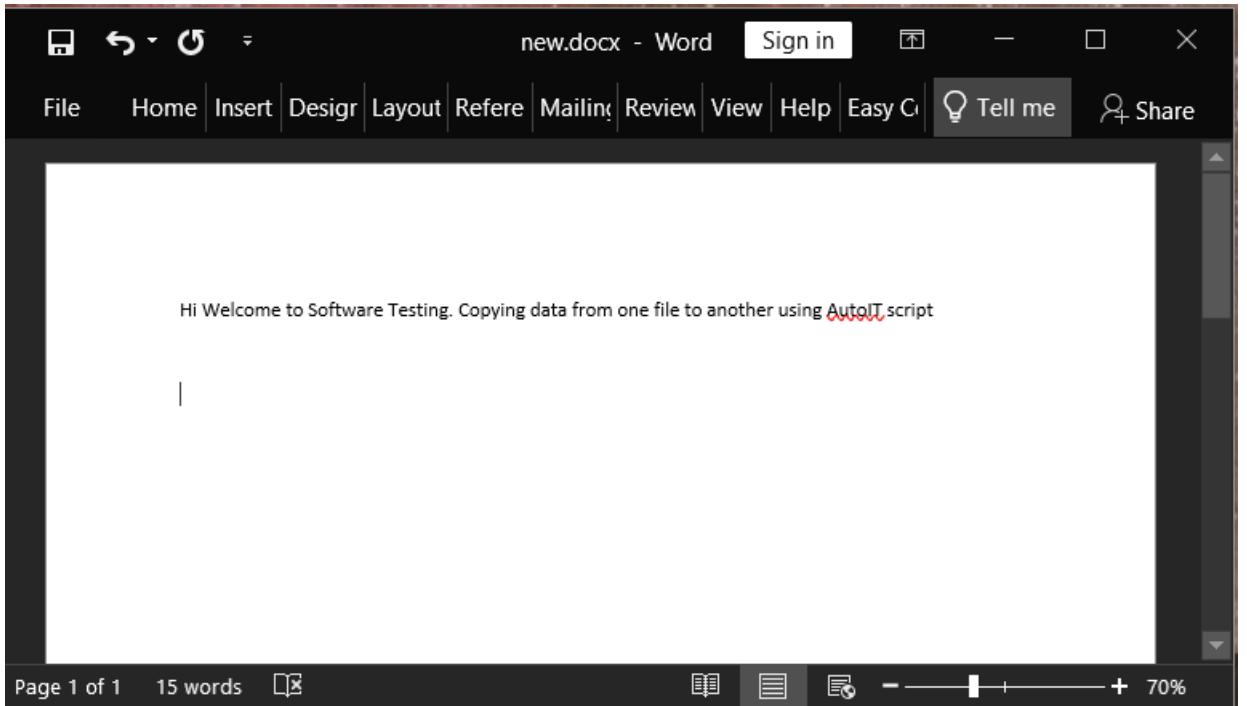


Copying data from one file to another file

Old.docx



New.docx



Conclusion:

We learned technique in the this experiment to automate Windows Notepad Application.

Practical No. 09

Aim : Automate any installation procedure - WinZip.

Experiment No. 05

Aim: Automate any installation procedure - WinZip.

Theory:

1. WinZip installation consists of approximately 10 dialog boxes that you must click buttons (usually Next) to continue. We are going to write a script that simply waits for these dialog boxes to appear and then clicks the appropriate buttons.
2. As usual with these types of installations the window title of each dialog is the same (WinZip.exe setup) so we must use window text to tell the difference between windows.
3. First create a directory that we will use for the WinZip installer and our script file. Copy the WinZip installer to the this directory and create a blank script called winzipinstall.au3.
4. We will now run through the installation manually and write the script as we go. The script lines to automate each dialog will be shown after each picture.
5. The first script line is easy, we want to run winzip26.exe installer. So the first line is:
`Run ("WinZip26.exe")`
6. We need to for this window to popup and become active, and then we need to send the keystroke ALT-S to click the Setup button. The resulting script lines are:

WinWaitActive ("Winzip@90SR-1Setup", "&Setup")

Send ("!s")

7. We ~~enter~~ need to wait for this screen to be active and then click ENTER to accept the install location. The script lines are :

WinWaitActive ("WinzipSetup", "into the following folder")

Send ("\$ENTER")

8. Notice that this window has exactly the same title as of first winzip setup - in fact all the dialogs in the setup have this title! In order to tell the difference between these windows we must also use the window text - on each screen try to pick the most unique text you can. In this case I've chose Winzip Features include. After the window has appeared we will want to press ALT-n:

WinWaitActive ("WinzipSetup", "Winzip Features Include")

Send ("!n")

- Wait for the window to appear and then press ALT-y to accept the agreement :

WinWaitActive ("License Agreement")

Send ("!y")

9. Setup continues in a similar fashion for a few dialogs. The picture of each dialog is shown along with the script lines needed to automate it :

WinWaitActive ("Winzip Setup", "Quick Start (mild)")

Send ("!n")

(Extracted from "autoit3script")

Code:

SetWinAPI("WinZip", "WinZip API");
\$zip = WinZip::Open("C:\Program Files\WinZip\WinZip.exe");

WinZip::Extract("C:\Program Files\WinZip\WinZip.exe", "C:\Temp\WinZip");

("Extracted")

WinZip::Close(\$zip);
\$zip = WinZip::Open("C:\Program Files\WinZip\WinZip.exe");
WinZip::Extract("C:\Program Files\WinZip\WinZip.exe", "C:\Temp\WinZip");
WinZip::Close(\$zip);
\$zip = WinZip::Open("C:\Temp\WinZip\WinZip.exe");
WinZip::Extract("C:\Temp\WinZip\WinZip.exe", "C:\Temp\WinZip");
WinZip::Close(\$zip);

(Extracted from "autoit3script")

Code:

SetWinAPI("WinZip", "WinZip API");
\$zip = WinZip::Open("C:\Program Files\WinZip\WinZip.exe");

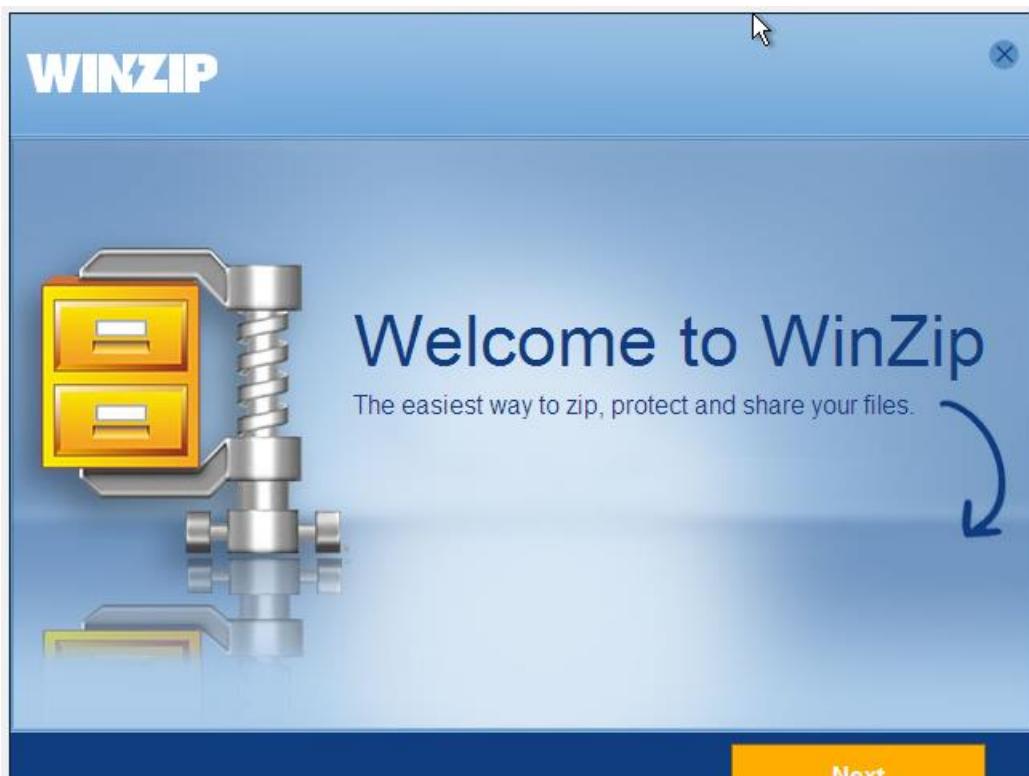
("Extracted from "autoit3script")

Code:

WinZip::Close(\$zip);
\$zip = WinZip::Open("C:\Program Files\WinZip\WinZip.exe");

Conclusion:

Thus, I have studied and executed a program in AutoIt v3 to automate the installation procedure of WinZip.



WINZIP



16.99MB out of 67.05MB @ 1.77MB/s

25%

Installing Status

Please wait while the installer wizard downloads and optimizes the necessary packages.

WINZIP



100%

Installing Status

Please wait while the installer wizard downloads and optimizes the necessary packages.



Next

WINZIP



Thank you for installing WinZip

The easiest way to zip, protect and share your files

© 2004 WinZip International, Inc. All rights reserved.

WinWaitActive("WinZip Setup", "switch between the two interfaces")

Send("!c")

Send("!n")

WinWaitActive("WinZip Setup", "&Extract Setup (recommended)")

Send("!e")

Send("!n")

WinWaitActive("WinZip Setup", " WinZip needs to associate itself with your
archives")

Send('n') Send("!n")

10. This is the final dialog of the setup. Notice that the finish button doesn't have a keyboard shortcut. Luckily, it is the default button on this dialog so we can just press ENTER to select it. IF this were not the case then we would have to TAB between controls or better yet to use the ControlClick function.

WinWaitActive("WinZip Setup", " Thank you for installing this evaluation
version")

Send("%ENTER%")

WinWaitActive("WinZip (Evaluation Version)")

WinClose("WinZip (Evaluation Version)")

Conclusion :

Close → Thus, I have studied and executed a program in AutoIt v3 V3
to automate the installation procedure of WinZip.

Experiment No. 10

Aim: Study and demonstration of Bug tracking tool - Bugzilla.

(Part 1)

(Part 2)

(Part 3)

(Part 4)

(Part 5)

(Part 6)

Aim: Study and demonstration of Bug Tracking Tool - Bugzilla.

Theory:

Bugzilla is an open source tool used for issues and bugs tracking system. It is widely used as a bug-reporting tool for all types of testing functions.

Bugzilla is an open-source tool used to track bugs and issues of a project or a software. It helps the developers and other stakeholders to keep track of outstanding problems with the product.

- It was written by Terry Weissman in TST TCL programming language in 1978.
- Later, bugzilla was written in PERL and it uses the MySQL database.

Software requirements:

- Perl
- Database server
- Web Server.

Bugzilla. Download and Install:

```
git clone --branch release-xx-stable https://github.com/bugzilla/bugzilla
```

5. Log-in

The fields that are present in the default bug screen are:

- Product: Which we selected on the previous page.
- Component: Each product can be split into one or more components based on the use or functionality, etc.

Q) Q) Bugzilla

It is a bug tracking tool developed by Mozilla Foundation.

Want

It is a web based bug tracking tool developed by Mozilla Foundation. It is used to track bugs and issues in software development process. It is a web based application which allows users to log, track and resolve bugs. It is a distributed system which means it can be used by multiple users simultaneously. It is a free and open source software.

It is a web based bug tracking tool developed by Mozilla Foundation.

Bugzilla

It is a web based bug tracking tool developed by Mozilla Foundation.

Want

Want

Want

Want

Want

Want

Want

Want

Conclusion:

Thus, we have studied bug tracking tool - BugZilla.

1) Log-in



2) In the next window

The screenshot shows the Bugzilla Bug Report Form. The form fields are numbered 1 through 11:

- 1 * Product:** Sam's Widget
- 2 Component:** Widget Gears
- 3 Component Description:** Gears for Sam's widgets
- 4 Version:** unspecified
- 5 Severity:** normal
- 6 Hardware:** PC
- 7 OS:** Windows NT
- 8 * Summary:** (empty field)
- 9 Description:** (empty field)
- 10 Attachment:** Add an attachment
- 11 Submit Bug**

A note at the bottom right of the form says: "We've made a guess at your operating system and platform. Check them and make any corrections if necessary."

3) Bug is created

First Last Prev Next This bug is not in your last search results.

Bug 26320 - Gears for sams widget twisted (edit)

| | |
|--|--|
| Status: CONFIRMED (edit) | Reported: 2015-01-07 02:50 PST by James |
| Product: Sam's Widget | Modified: 2015-01-07 03:10 PST (History) |
| Component: Widget Gears | CC List: 1 user including you (edit) |
| Version: unspecified | See Also: (add) |
| Hardware: PC Windows NT | Large text box: 1 |
| Importance: P2 normal | |
| Set Milestone: --- | |
| Assigned To: sam_folk-williams (edit) (take) | |
| QA Contact: (edit) (take) | |
| ② URL: | A multiple-select box: Always Appears Also Always Appears Third Value, Always |
| ③ Whiteboard: | Drop Down List: ... |
| ④ Keywords: | Date Time: <input type="text"/> |
| ⑤ Tags: | Bug ID Field: <input type="text"/> |
| ⑥ Depends on: | Flags: None yet set (set_flags) |
| ⑦ Blocks: | |
| Show dependency tree / graph | |
| Orig. Est.: Current Est.: Hours Worked: Hours Left: %Complete: Gain: Deadline: | 0.0 0.0 0.0 + 0 0.0 0 0.0 2015-01-09 <input style="width: 20px; height: 20px; border: none; background-color: #ccc;" type="button" value="..."/> |
| Summarize time (including time for bugs blocking this bug) | |
| Attachments 8 | |
| Add an attachment (proposed patch, testcase, etc.) | |

4) Deadline in Bugzilla usually gives the time-limit to resolve the bug in given time frame.

| | |
|--|--|
| Orig. Est.: Current Est.: Hours Worked: Hours Left: %Complete: Gain: Deadline: | <input style="width: 20px; height: 20px; border: none; background-color: #ccc;" type="button" value="..."/> January 2015 <input type="button" value="<"/> <input type="button" value=">"/> |
| Summarize time (including time for bugs) | |
| Attachments | |
| Add an attachment (proposed patch, testcase, etc.) | |
| Additional Comments: | |
| <p>Status: CONFIRMED <input style="border: 1px solid blue; padding: 2px; width: 100px; height: 20px; border-radius: 5px;" type="button" value="CONFIRMED"/> IN_PROGRESS RESOLVED</p> <p>James 2015-01-07 02:50:31 PST</p> <p>The widget gears are twisted at the end and not showing correct signal</p> | |
| Save Changes <input style="border: 2px solid red; border-radius: 50%; width: 40px; height: 40px; margin-left: 10px;" type="button" value="Save Changes"/> | |
| Collapse All Comments Expand All Comments | |

You can select bug status over here

select deadline for your bug-report

- Version: Version of the product in which the bug was detected.
- Reporter: Email id of the person logging the bug.
- Severity: Severity of the bug
- Hardware and OS: Machine details from which bug is logged.
- Summary: To provide summary for the bug.
- Description: A complete description of the bug.
- Add an attachment: To provide any supporting file as attachment.

Conclusion:

Thus, we have studied bug tracking tool - Bugzilla.

Experiment No. 11

Aim: Study and demonstrate of test management tool - Test director.

Experiment No. 12

Aim: Study and demonstrate of test management Tool - Test director.

Theory:

Test director offers an organized framework for testing applications before they are deployed. Since test plans evolve with new or modified application requirements, you need a central data repository for organizing and managing testing process. Test director guides you through the requirements specification, test planning, test execution, and defect tracking phases of the testing process. The test director testing process includes four phases:

Step 1: Open the test director options window.

<http://localhost:81/virtualdirectory1//default.html>.

Step 2: The test director window opens.

Step 3: Explore the common test director elements.

1. Analyze your application and determine your testing requirements.

To specify your testing requirements,

Consider the following steps:

Examine the application documentation to

determine your testing scope, test goals, objectives and strategies.

Create a test plan, based on your testing requirements.

2. To create a test plan,

Consider the following steps:

- Examine your application, System environment, and testing resources to determine your testing goals.

Test Director is a test management tool which is integrated with build and deployment tools.

Test Director is a document management tool which is used to manage the entire software development life cycle. It is a web-based application which provides a graphical user interface for managing test cases, test environments, and test results. Test Director also provides reporting and analysis features to help users track progress and identify trends in their test results.

Test Director has several advantages over other test management tools:

- It is highly customizable and can be tailored to specific needs.
- It provides a centralized repository for all test-related data.
- It supports multiple platforms and environments.

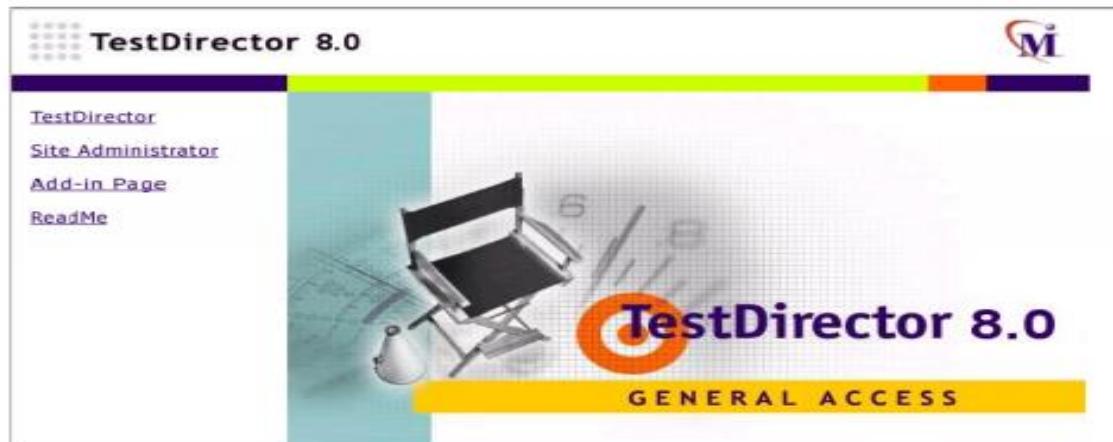
Test Director is widely used in the software industry for managing test cases, test environments, and test results. It is a powerful tool for improving test efficiency and reducing costs. It is also used for managing test automation and regression testing.

Conclusion:

Thus, we have studied test management tool - Test Director.

1. Open the TestDirector Options window.

<http://localhost:81/virtualdirectory1//default.html>



1. The Test Director Login window opens



2. Explore the common TestDirector elements.

| TestDirector toolbar | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---------------------------------|---------------------|-----------|---------------------|---------------|----------|---------------|----------|-------------------------------|--------|---------|----------|------------|----------|--------------------------|--------|---------|----------|------------|----------|-------------------------------|--------|-----------|----------|-------------|----------|---------------------------|--------|-----------|----------|-------------|----------|-----------------------------|--------|-----------|----------|-------------|----------|-----------------------------|--------|-----------|----------|-------------|----------|------------------------|--------|---------|----------|------------|----------|--------------------|--------|---------|----------|------------|----------|---------------------------------|--------|---------|----------|-------------|----------|------------------------------|--------|-----------|----------|-------------|----------|------------------------------|--------|-----------|----------|-------------|
| Menu bar | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module toolbar | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Tools button | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Help button | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Logout button | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>RegID</th> <th>Name</th> <th>Direct Cover Status</th> <th>Author</th> <th>Reviewed</th> <th>Creation Time</th> </tr> </thead> <tbody> <tr> <td>[RQ0000]</td> <td>1 - Mercury Tours Application</td> <td>Failed</td> <td>alex_td</td> <td>Reviewed</td> <td>9:07:29 AM</td> </tr> <tr> <td>[RQ0020]</td> <td>2 - Application Security</td> <td>Failed</td> <td>alex_td</td> <td>Reviewed</td> <td>1:15:09 PM</td> </tr> <tr> <td>[RQ0042]</td> <td>3 - Application Client System</td> <td>Failed</td> <td>shelly_td</td> <td>Reviewed</td> <td>11:29:55 AM</td> </tr> <tr> <td>[RQ0045]</td> <td>4 - Application Usability</td> <td>Failed</td> <td>shelly_td</td> <td>Reviewed</td> <td>11:33:15 AM</td> </tr> <tr> <td>[RQ0046]</td> <td>5 - Application Performance</td> <td>Failed</td> <td>shelly_td</td> <td>Reviewed</td> <td>11:34:29 AM</td> </tr> <tr> <td>[RQ0047]</td> <td>6 - Application Reliability</td> <td>Passed</td> <td>shelly_td</td> <td>Reviewed</td> <td>11:35:05 AM</td> </tr> <tr> <td>[RQ0008]</td> <td>7 - Profile Management</td> <td>Failed</td> <td>alex_td</td> <td>Reviewed</td> <td>9:20:33 AM</td> </tr> <tr> <td>[RQ0007]</td> <td>8 - Booking System</td> <td>Failed</td> <td>alex_td</td> <td>Reviewed</td> <td>9:19:19 AM</td> </tr> <tr> <td>[RQ0011]</td> <td>9 - Flights Reservation Service</td> <td>Failed</td> <td>alex_td</td> <td>Reviewed</td> <td>10:16:39 AM</td> </tr> <tr> <td>[RQ0071]</td> <td>10 - Reservations Management</td> <td>Passed</td> <td>robert_td</td> <td>Reviewed</td> <td>12:58:44 PM</td> </tr> <tr> <td>[RQ0071]</td> <td>10 - Reservations Management</td> <td>Passed</td> <td>robert_td</td> <td>Reviewed</td> <td>12:59:44 PM</td> </tr> </tbody> </table> | | RegID | Name | Direct Cover Status | Author | Reviewed | Creation Time | [RQ0000] | 1 - Mercury Tours Application | Failed | alex_td | Reviewed | 9:07:29 AM | [RQ0020] | 2 - Application Security | Failed | alex_td | Reviewed | 1:15:09 PM | [RQ0042] | 3 - Application Client System | Failed | shelly_td | Reviewed | 11:29:55 AM | [RQ0045] | 4 - Application Usability | Failed | shelly_td | Reviewed | 11:33:15 AM | [RQ0046] | 5 - Application Performance | Failed | shelly_td | Reviewed | 11:34:29 AM | [RQ0047] | 6 - Application Reliability | Passed | shelly_td | Reviewed | 11:35:05 AM | [RQ0008] | 7 - Profile Management | Failed | alex_td | Reviewed | 9:20:33 AM | [RQ0007] | 8 - Booking System | Failed | alex_td | Reviewed | 9:19:19 AM | [RQ0011] | 9 - Flights Reservation Service | Failed | alex_td | Reviewed | 10:16:39 AM | [RQ0071] | 10 - Reservations Management | Passed | robert_td | Reviewed | 12:58:44 PM | [RQ0071] | 10 - Reservations Management | Passed | robert_td | Reviewed | 12:59:44 PM |
| RegID | Name | Direct Cover Status | Author | Reviewed | Creation Time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0000] | 1 - Mercury Tours Application | Failed | alex_td | Reviewed | 9:07:29 AM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0020] | 2 - Application Security | Failed | alex_td | Reviewed | 1:15:09 PM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0042] | 3 - Application Client System | Failed | shelly_td | Reviewed | 11:29:55 AM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0045] | 4 - Application Usability | Failed | shelly_td | Reviewed | 11:33:15 AM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0046] | 5 - Application Performance | Failed | shelly_td | Reviewed | 11:34:29 AM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0047] | 6 - Application Reliability | Passed | shelly_td | Reviewed | 11:35:05 AM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0008] | 7 - Profile Management | Failed | alex_td | Reviewed | 9:20:33 AM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0007] | 8 - Booking System | Failed | alex_td | Reviewed | 9:19:19 AM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0011] | 9 - Flights Reservation Service | Failed | alex_td | Reviewed | 10:16:39 AM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0071] | 10 - Reservations Management | Passed | robert_td | Reviewed | 12:58:44 PM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| [RQ0071] | 10 - Reservations Management | Passed | robert_td | Reviewed | 12:59:44 PM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

- Divide your application into modules or functions to be tested. Build a test plan tree to hierarchically divide your application into testing units, or subjects.
3. Submit defects detected in your application. Quality assurance testers, developers, project managers and end users can add defects during any phase of the testing process.

Conclusion:

Thus, we have studied test management tool - Test director.