

Automated GUI testing using PERL

Author:

Girish M C
Intuit Technologies Ltd,
Bangalore, Karnataka, India
Girish_mc@intuit.com

Abstract:

Perl has been used for years in QA for various tasks such as preparing configuration files before the tests run and parsing log files after the test. In the past few years a lot of extensions have been added to Perl. Now, in addition to the above tasks Perl can already be used in all parts of the automatic test environment. It can be used to behave like a regular user, move the mouse and click on buttons on any application running on Microsoft Windows or to become a web client and test any web based application. This is nothing but Automated GUI testing using PERL and perl module Win32::GuiTest

Keywords:

PERL, GUI Test. Win32 GUI

Introduction

This paper describes Automated GUI testing using PERL and perl module Win32::GuiTest. Win32::GuiTest is Perl module for automation of operations on Win32 GUI interface. It provides a set of methods for manipulating window handles, controls and to simulate user input. (Example: FindWindowLike(undef, \$parent_window_title, ""); PushChildButton(\$window[0], \$button[0], 0.25); etc.

Win32::GuiTest module can identifies windows by their text, class, and parent window attributes.

Let me also tell few Words about Windows:

- Graphical user interfaces manage windows. Windows are just reusable objects with which users can interact.
- Almost all GUIs have more than just one window. Windows have common elements that you need to consider before writing a program that interacts with a GUI.
- Each window belongs to a window class (making it possible to search them by class).
- Windows have an organizational hierarchy; every GUI has at least one root window, and every window may have child windows. Windows form a tree. This makes them searchable (by class or not) in depth: start from a root window and search among its siblings.
- Some windows have text attached to them. This is useful to identify windows.
- Windows have an numeric ID that uniquely identifies them.

This means that you can identify windows by any of their text, class, and parent window attributes. You can also pinpoint a window by its ID.

Audience

This paper would be very helpful for QA engineers with some background in Perl. And all other people who maybe interested in knowing what automated GUI testing using Perl is all about. They can be working on a Web site, Web application, or any other product.

Where has this applied?

It was successfully implemented for products running on Windows Vista and Windows XP 64 bit Operating Systems since no commercially “off the shelf tools” worked successfully on Vista OS.

Agenda

- Introduction
- The challenge
- Learnings
- Prerequisites
- How PERL GUI automation works
- Few words about Windows
- Tangible benefits
- What next
- Demo
- Q&A

The challenge

- Why and when to use Perl for Automated GUI Testing?

- Automated GUI testing using PERL on alpha versions of upcoming platforms/operating systems, on which neither of the commercial test tools work
- If you are looking forward to enhancing organizational productivity by migrating applications to the new operating system platform? Example: say for Windows Vienna, a future version of Microsoft Windows
- Are you under pressure because only manual testing can be done on alpha versions of upcoming platforms, because none of the commercial test tools are supported on alpha versions or pre-releases of operating systems ? Example: Mercury Quick Test Professional, Mercury Win Runner, Segue's Silk Test, etc will not work
- Using Open Source is trendy - instead of spending tons of dollars on commercial tools you could get the same value for free.

Learnings

- Intuit and Intel being a product development companies, before moving from current versions of the Windows operating system to Windows Vista™ and Windows XP 64 Bit™, we had to test our Hardware and applications to ensure that they are compatible with the new operating system.
 - Intel - Intel® PROSet/Wireless Software
 - Intuit – QuickBooks SDK
- But the challenge was that neither of the commercially available test automation tools supported the alpha version/pre release of Windows Vista™.
- This problem is being faced by most of the product development companies for migrating from older Operating System to upcoming platforms.
- Perl has been used for years in QA for various tasks such as preparing configuration files before the tests run and parsing log files after the test. In the past few years a lot of extensions have been added to Perl. Now, in addition to the above tasks Perl can already be used in all parts of the automatic test environment. It can be used to behave like a regular user, move the mouse and click on buttons on any application running on Microsoft Windows or to become a web client and test any web based application. This is nothing but Automated GUI testing using PERL and perl module Win32::GuiTest

- We successfully performed test automation on Windows Vista™, Windows XP 64 Bit™
- Perl supports tons of standard and non-standard packages. Ex: Win32::GuiTest, File::Copy, Text::ParseWords & many
- It has lot of built-in functions and we can use C, MSDN functions
- Perl runs on all platforms and is far more portable than other tools like Shell, PHP, Python, Ruby, Tcl Tools

Prerequisite software

1. ActivePerl

It includes PPM (Perl Package manager), for installing Perl extension modules

2. Win32::GuiTest

Perl module for automation of operations on Win32 GUI interface. It provides a set of methods for manipulating window handles, controls and to simulate user input.

3. Win32GuiTest.exe*

Script recording application has been written to use with Win32::GuiTest. This is an optional GUI test Utility



```
## Main/Core
```

```
MouseMoveAbsPix(574, 490);
```

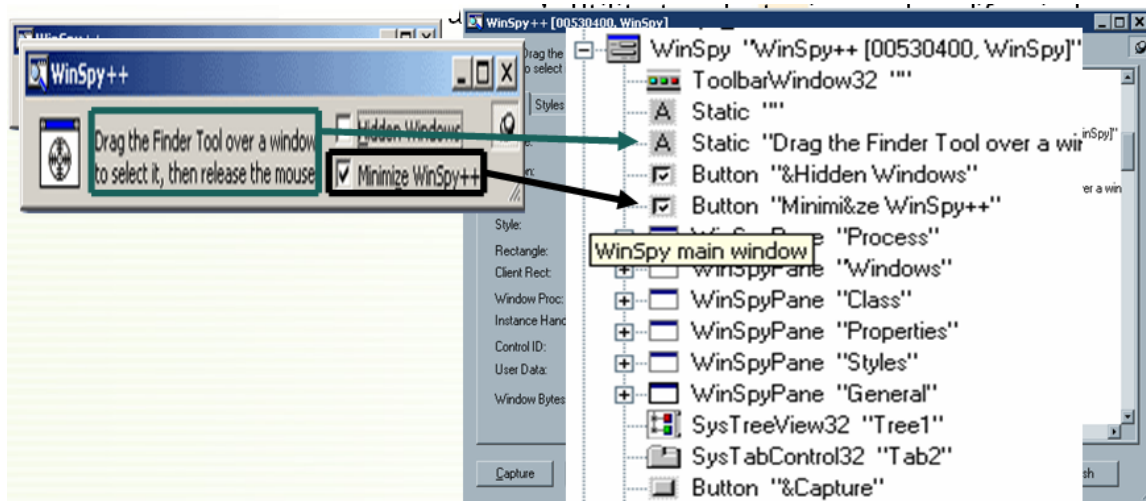
```
SendMouse (' {LEFTCLICK}');
```

```
MouseMoveAbsPix(366, 304);
```

```
SendMouse (' {LEFTCLICK}');
```

4. WinSpy++*

Win32 programmer's Utility to select, view and modify window properties



How PERL GUI automation works

- All windows which are opened on your screen are system objects. Win32 port keeps track of them and can give information about them
- Let's write a PERL script, which will open the Calculator application and does some calculation
- Firstly the Calculator Application is opened

```
system "start calc.exe";
```

- To move the input focus to Calculator. But which window is Calculator? And how to manipulate this window?

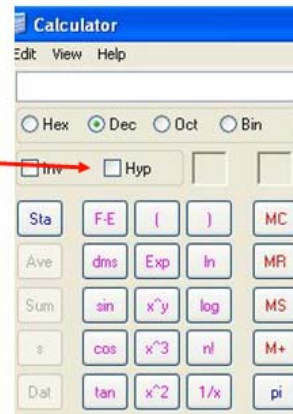
```
my @mainwin = FindwindowLike(undef, "calculator", "");  
SetForegroundwindow($mainwin[0]);
```

- Let's find the window handle of "Hyp" button and then click this button.

```
my @wnds = GetChildwindows($mainwin[0]);  
foreach my $winHandle (@wnds)  
{  
    my $winName = GetwindowText($winHandle);  
    my $winClass = GetClassName($winHandle);  
    print "$winHandle-->$winName-->$winClass\n";  
}  
PushButton(@wnds[5])  
my $buttonName = GetwindowText(@wnds[5]);  
print "$buttonName\n";
```

The result

```
852250 --> --> Edit  
721098 --> Hyp --> Button  
524612 --> Inv --> Button  
393536 --> --> Button  
590108 --> Sta --> Button  
327980 --> Hex --> Button  
327982 --> Dec --> Button  
393534 --> Oct --> Button  
262414 --> Bin --> Button  
852162 --> --> Button  
786730 --> --> Button
```



Few words about Windows Controls

- Graphical user interfaces manage windows. Windows are just reusable objects with which users can interact.
- Almost all GUIs have more than just one window. Windows have common elements that you need to consider before writing a program that interacts with a GUI.
- Each window belongs to a window class (making it possible to search them by class).
- Windows have an organizational hierarchy; every GUI has at least one root window, and every window may have child windows. Windows form a tree. This makes them searchable (by class or not) in depth: start from a root window and search among its siblings.
- Some windows have text attached to them. This is useful to identify windows. Windows have a numeric ID that uniquely identifies them. This means that you can identify windows by any of their text, class, and parent window attributes. You can also pinpoint a window by its ID.

Tangible benefits

- Test automation on alpha versions of upcoming platforms/operating systems, on which neither of the commercial test tools work
- It's a freeware software package
- Perl runs on all platforms and is far more portable than other environments
- Perl scripts can be ported between 32 Bit and 64 Bit operating systems without any modifications at all.
- Modularization (Packages): It's support for Win32 GUI Testing and many other standard packages/plug-ins
- Perl provides the most advanced regular expressions
- Perl allows to use all system calls including those necessary for networking
- Perl is significantly faster than other scripting languages
- Object-oriented techniques
- A large number of built-in checks by the compiler and at runtime

What next

- Think of PERL, when migrating from older Operating System to upcoming platforms.

- Think of PERL for GUI Automaton to test your applications, to ensure that they are compatible with the new operating system.
- Think of PERL for GUI Automaton when neither of the commercially available tools work for you
- Be the first to Automate your applications on **Windows Vienna™** , which is getting ready and will reach us in some time

Useful Links

- **ActivePerl:** <http://www.activestate.com/>
- **Win32::GuiTest:** [Win32-GuiTesthttp://search.cpan.org/dist/Win32-GuiTest/](http://search.cpan.org/dist/Win32-GuiTest/)