Writing automation to wait for a window to be created (and dismiss it)

devblogs.microsoft.com/oldnewthing/20140217-00

February 17, 2014

Raymond Chen

Today's Little Program uses UI Automation to cancel the Run dialog whenever it appears. Why? Well, it's not really useful in and of itself, but it at least provides an example of using UI Automation to wait for an event to occur and then respond to it.

```
using System.Windows.Automation;
class Program
{
 [System.STAThread]
 public static void Main(string[] args)
 {
  Automation.AddAutomationEventHandler(
   WindowPattern.WindowOpenedEvent,
   AutomationElement.RootElement,
   TreeScope.Children,
   (sender, e) => {
    var element = sender as AutomationElement;
    if (element.Current.Name != "Run") return;
    var cancelButton = element.FindFirst(TreeScope.Children,
     new PropertyCondition(AutomationElement.AutomationIdProperty, "2"));
    if (cancelButton != null) {
     var invokePattern = cancelButton.GetCurrentPattern(InvokePattern.Pattern)
                         as InvokePattern;
     invokePattern.Invoke();
     System.Console.WriteLine("Run dialog canceled!");
    }
   });
  System.Console.ReadLine();
  Automation.RemoveAllEventHandlers();
 }
}
```

Okay, let's see what's going on here.

The program registers a delegate with UI automation which is called for any `WindowOpened` event that is an immediate child (`TreeScope.Children`) of the root (`Automation-Element.RootElement`). This will catch changes to top-level unowned windows, but not bother firing for changes that occur inside top-level windows or for owned windows.

Inside our handler, we check if the window's title is *Run*. If not, then we ignore the event. (This will get faked out by any other window that calls itself *Run*.)

Once we think we have a *Run* dialog, we look for the Cancel button, which we have determined by using UI Spy to have the automation ID `"2"`. (That this is the numeric value of `IDCANCEL` is hardly a coincidence.)

If we find the Cancel button, we obtain its Invoke pattern so we can Invoke it, which for buttons means pressing it.

Take this program out for a spin. Run the program and then hit `Win`+`R` to open the Run dialog. Oops, the program cancels it!

Ha-ha!

Raymond Chen

Follow