Opening and manipulating Internet Explorer windows programmatically

devblogs.microsoft.com/oldnewthing/20131021-00

October 21, 2013

Raymond Chen

Today's Little Program takes the JavaScript application from a few years ago and converts it to C#. This was inspired by a customer who started with the question, "How can I close all Internet Explorer windows programmatically?"

This was a strange request. After all, the user may be rather upset that their Amazon shopping spree was suddenly terminated mid-stream.

Upon closer questioning, the customer explained that they had a business process which, among other things, opened a bunch of Internet Explorer windows, and when the operation was over, they wanted to close the windows.

Okay, so they didn't really want to close *all* Internet Explorer windows. They just wanted to close the ones that they had opened.

Which is easy to do. In fact, it's a line-by-line translation of the JavaScript version!

```
class Program {
 public static void Main() {
   var ie = new SHDocVw.InternetExplorer();
   ie.Visible = true;
   ie.Navigate("http://www.microsoft.com/");
   System.Threading.Thread.Sleep(5000);
   ie.Quit();
 }
}
```

If you want to open a dozen browser windows and then close them later, then go ahead and open a dozen browser windows, and then close them when you're done.

```
class Program {
 public static void Main() {
  var windows = new System.Collections.Generic.
                   List<SHDocVw.InternetExplorer>();
  for (int i = 0; i < 12; i++) {
   var ie = new SHDocVw.InternetExplorer();
   windows.Add(ie);
   ie.Visible = true;
   ie.Navigate("http://www.microsoft.com/");
  }
  // blah blah wait for user to finish business process
  // (We'll just sleep for a few seconds.)
  System.Threading.Thread.Sleep(10000);
  // Close all the windows
  foreach (var ie in windows) {
   ie.Quit();
  }
 }
}
```

Actually, let's make it a bit more interesting. Suppose the intranet site sends you to the page `http://contoso/finished.aspx` when you finish submitting your Widget Exemption Request. We want to close the Internet Explorer window when that happens.

```
class Program {
 static void ProcessWidgetExemptionRequest() {
  var exit = new System.Threading.EventWaitHandle(
             false, System.Threading.EventResetMode.ManualReset);
  var ie = new SHDocVw.InternetExplorer();
  ie.Visible = true;
  ie.Navigate("http://contoso/start.aspx");
  ie.DocumentComplete += (object frame, ref object url) => {
   if (url as string == "http://contoso/finished.aspx") {
    exit.Set();
   }
  };
  // Wait for the user to finish submitting their Widget
  // Exemption Request.
  exit.WaitOne();
  // Clean up the IE window
  ie.Quit();
 }
 public static void Main() {
 {
  ProcessWidgetExemptionRequest();
 }
}
```

The `ProcessWidgetExemptionRequest` method creates an event that we will use to tell us when the user has finished the operation. We then create an Internet Explorer window, navigate it to the Exemption Request Web site, and hook up an anonymous function that

listens for document completion events. Then we wait.

When we get a completion event that says that the user has reached the Finished page, we set our private event, which causes the wait to complete, at which point we clean up the Internet Explorer window and return to our caller.

You can try out this program yourself, but you probably want to change the URLs.

Raymond Chen

Follow