# IeController

HomePage | RecentChanges | Preferences | RubyGarden

IeController is a package by Wiki:ChrisMorris that drives Wiki:InternetExplorer and commits testing against it. It compares favorably to the Perl library http://samie.sf.net

The latest version is at the bottom of this:

http://clabs.org/ruby.htm

Make sure you also install two other packages found there: clutil and clxmlserial. (6/23/03 - Latest cvs no longer requires these -- this was a weird dependency that snuck in.)

It's rough and in progress, but if you're interested, let me know. Go to my name page [ChrisMorris] for contact info. (You can browse the cvs of it here: http://clabs.org/iec)

---

This correspondence inspired a hello-world example for it:

> Where's a "hello world" example?

> http://samie.sf.net , your competition in Perl, does this:

```
my $URL = "http://www.amazon.com";
Navigate($URL);
WaitForBusy();
$IEDocument = GetDocument();
$seconds = WaitForDocumentComplete();
print "Amazon took $seconds seconds to load\n";
SetListBoxItem("url","index=baby");
SetEditBox("field-keywords","rattlesnake");
ClickFormImage("Go");
$seconds = WaitForDocumentComplete();
print "Diaper page took $seconds seconds to load\n";
```

My version would be as follows:

```
require 'cl/iec'


VISIBLE = true
iec = ClIEController.new(VISIBLE)
iec.navigate('http://www.amazon.com')
form = IEDomFormWrapper.new(iec.form)
form.url = 'Baby'
form.invoke('field-keywords').value = 'rattlesnake'
form.submit
iec.wait
```

Comments:

- the navigate method automatically calls the wait method, form.submit does not (but I see now it should).
- IEDomFormWrapper? basically has a method_missing routine which handles a bunch of dirty work. In prepping this, I found I usually have add a 'def form...' method to my test scripts which will automatically take care of creating the wrapper -- that helper function needs to be moved into the iec lib.
- The built-in drop down wrapper allows you to set the drop down based on the display value, not the internal value ('index=baby').
- Because Amazon put a hyphen in their edit box input name, the slick method_missing approach in the form wrapper is thwarted. Normally, you can do:

```
iec.form.field-keywords = 'rattlesnake'
```

- … but the hyphen is parsed as a method call so only the term 'field' is caught by method missing -- so the lib tries to find a form field called 'field' and it fails to do so. invoke is a WIN32OLE method that allows you to force a call through to the WIN32OLE instance. Because we're also not getting to use the built in text box wrapper, we have to also reference the .value property.
- The iec.form call takes an index argument that defaults to 0, but you can set it to reference other forms on a multiple form page.

Here's another example, using Google and helper routines that probably need to find their way into the lib:

```
require 'cl/iec'

def form
  IEDomFormWrapper.new(@iec.form)
end

def submit
  form.submit
  @iec.wait
end

VISIBLE = true
@iec = ClIEController.new(VISIBLE)
@iec.navigate 'http://www.google.com'
form.q = 'Programming Ruby'
submit
```

Neither of these websites seems to have easily testable output. But, ugly as it is, here's a way to get the first returned link and click it using the ClIEDomViewer?. First time requires a dump to find the path to the first link:

```
dv = ClIEDomViewer.new(@iec)
File.open('dom.dump.txt', 'w') do |f| dv.outputDom(f) end
```

Digging through dom.dump.txt, we find the first link here:

```
[snip]
nodeName: -HTML-BODY1-DIV1
nodeName: -HTML-BODY1-DIV1-P1
nodeName: -HTML-BODY1-DIV1-P1-A1
nodeName: -HTML-BODY1-DIV1-P1-A1-B1
nodeName: -HTML-BODY1-DIV1-P1-A1-B1-#text1
nodeValue: Programming
nodeName: -HTML-BODY1-DIV1-P1-A1-#text1
nodeValue:
nodeName: -HTML-BODY1-DIV1-P1-A1-B2
nodeName: -HTML-BODY1-DIV1-P1-A1-B2-#text1
nodeValue: Ruby
nodeName: -HTML-BODY1-DIV1-P1-A1-#text2
nodeValue: : The Pragmatic Programmer's Guide
[snip]
```

so, we now have a path to the first link: "-HTML-BODY1-DIV1-P1-A1"
and we can do:

```
root = dv.htmlRootNode
dv.buildNodeWrapperTree(root)
link = dv.getNodeWrapperFromPath('HTML-BODY1-DIV1-P1-A1', root)
link.node.click
@iec.wait
```

Obviously, this is fragile should Google ever change their layout, but
without consistent ids on the <a> tags, or other somesuch, I don't
know of another way to do it.

Chris

http://clabs.org

---

HomePage | RecentChanges | Preferences | RubyGarden
Edit text of this page | View other revisions
Last edited June 23, 2003 10:04 am (diff)
Search: [                    ]