**<http://es.slideshare.net/tsimafeiavilin/ui-automationwhitecodedui-common-problems-and-tricks>**

**FOCUS**

AutomationElement Ref1 = AutomationElement.RootElement.FindFirst(TreeScope.Descendants,

new PropertyCondition(AutomationElement.AutomationIdProperty, elementID));

Ref1.SetFocus();

**HOW TO READ TEXT FROM LABEL, TEXTBOX, COMBO**

public static string GetText(string elementID)

{

AutomationElement Ref1 = AutomationElement.RootElement.FindFirst(TreeScope.Descendants,

new PropertyCondition(AutomationElement.AutomationIdProperty, elementID));

object patternObj;

if (Ref1.TryGetCurrentPattern(ValuePattern.Pattern, out patternObj))

{

var valuePattern = (ValuePattern)patternObj;

return valuePattern.Current.Value;

}

else if (Ref1.TryGetCurrentPattern(TextPattern.Pattern, out patternObj))

{

var textPattern = (TextPattern)patternObj;

return textPattern.DocumentRange.GetText(-1).TrimEnd('\r'); // often there is an extra '\r'

}

else

{

return Ref1.Current.Name;

}

return "";

}

**HOW TO SET TEXT INTO A TEXTBOX**

private void SetText(string elementID, string texto)

{

AutomationElement Ref1 = AutomationElement.RootElement.FindFirst(TreeScope.Descendants,

new PropertyCondition(AutomationElement.AutomationIdProperty, elementID));

ValuePattern tx1 = Ref1.GetCurrentPattern(ValuePattern.Pattern) as ValuePattern;

tx1.SetValue(texto);

}

**BUTTON CLICK**

public void ClickBtHx(string valor)

{

string elemento = System.Convert.ToInt32(valor, 16).ToString();

AutomationElement Ref1 = App.FindFirst(TreeScope.Descendants, new

PropertyCondition(AutomationElement.AutomationIdProperty, elemento));

InvokePattern Click1 = (InvokePattern)Ref1.GetCurrentPattern(InvokePattern.Pattern);

Click1.Invoke();

}

public void ClickBtDec(string elemento)

{

AutomationElement Ref1 = App.FindFirst(TreeScope.Descendants, new

PropertyCondition(AutomationElement.AutomationIdProperty, elemento));

InvokePattern Click1 = (InvokePattern)Ref1.GetCurrentPattern(InvokePattern.Pattern);

Click1.Invoke();

}

**DROPDOWNLIST**

public static bool dpSetValue(string elementID, string value)

{

AutomationElement Ref1 = AutomationElement.RootElement.FindFirst(TreeScope.Descendants,

new PropertyCondition(AutomationElement.AutomationIdProperty, elementID));

object patternProvider;

if (Ref1.TryGetCurrentPattern(ValuePattern.Pattern, out patternProvider))

{

ValuePattern valuePatternProvider = patternProvider as ValuePattern;

valuePatternProvider.SetValue(value);

return true;

}

return false;

}

// SELECT AN ELEMENT

var combo = FindElementWithId(scope, id);

ExpandCollapsePattern p = combo.GetCurrentPattern(ExpandCollapsePattern.Pattern) as ExpandCollapsePattern;

p.Expand();

var items = FindAllElementsWithClassName(combo, "ListBoxItem");

if (index &gt;= 0 &amp;&amp; index &lt; items.Count)

{

SelectionItemPattern itemp = items[index].GetCurrentPattern(SelectionItemPattern.Pattern) as

SelectionItemPattern;

itemp.Select();

}

p.Collapse();

**RADIO BUTTON**

AutomationElement Ref1 = AutomationElement.RootElement.FindFirst(TreeScope.Descendants,

new PropertyCondition(AutomationElement.AutomationIdProperty, "r3"));

SelectionItemPattern p = Ref1.GetCurrentPattern(SelectionItemPattern.Pattern) as SelectionItemPattern;

// GET STATE

string valor = p.Current.IsSelected.ToString();

// CHANGE STATE

p.Select();

**WITH CHECKBOX, HOW TO READ AND SET**

AutomationElement Ref1 = AutomationElement.RootElement.FindFirst(TreeScope.Descendants,

new PropertyCondition(AutomationElement.AutomationIdProperty, elementID));

TogglePattern p = Ref1.GetCurrentPattern(TogglePattern.Pattern) as TogglePattern;

string estado = p.Current.ToggleState.ToString();

p.Toggle();

string estado2 = p.Current.ToggleState.ToString();

= = = =

Object objPattern;

TogglePattern togPattern;

if (true == Ref1.TryGetCurrentPattern(TogglePattern.Pattern, out objPattern))

{

togPattern = objPattern as TogglePattern;

return togPattern.Current.ToggleState == ToggleState.On;

}

**LIST BOX**

AutomationElement List1 = AutomationElement.RootElement.FindFirst(TreeScope.Descendants,

new PropertyCondition(AutomationElement.AutomationIdProperty, "listBox1"));

AutomationElementCollection lbItems =

List1.FindAll(TreeScope.Children,

new PropertyCondition(AutomationElement.ControlTypeProperty,

ControlType.ListItem));

for (int i =0; i< lbItems.Count; i++)

{

AutomationElement item = lbItems[i];

string strValor = (string)item.GetCurrentPropertyValue(AutomationElement.NameProperty);

if (strValor == "dos" || strValor=="cuatro")

{

SelectionItemPattern p = item.GetCurrentPattern(SelectionItemPattern.Pattern)

as SelectionItemPattern;

// Get if selected or not

bool selected = p.Current.IsSelected;

// select

if (!selected) p.Select();

}

}

**TAB CONTROL**

AutomationElement tabCtl = AutomationElement.RootElement.FindFirst(TreeScope.Descendants,

new PropertyCondition(AutomationElement.AutomationIdProperty, "tab1"));

AutomationElementCollection tabItems =

tabCtl.FindAll(TreeScope.Children,

new PropertyCondition(AutomationElement.ControlTypeProperty,

ControlType.TabItem));

// get label in tab

AutomationElement item = tabItems[1];

string strValor = (string)item.GetCurrentPropertyValue(AutomationElement.NameProperty);

// click tab

SelectionItemPattern tab1 = item.GetCurrentPattern(SelectionItemPattern.Pattern)

as SelectionItemPattern;

tab1.Select();