private void Grid()

{

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

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

var FILA = Ref1.FindAll(TreeScope.Children,

new PropertyCondition(AutomationElement.ControlTypeProperty, ControlType.Custom));

int conteo = FILA.Count;

// TITULOS

var header = FILA[0].FindAll(TreeScope.Descendants,

new PropertyCondition(AutomationElement.ControlTypeProperty, ControlType.Header));

int conteo2 = FILA.Count;

for (int j = 1; j < conteo2-1; j++)

{

ValuePattern titulo = header[j].GetCurrentPattern(ValuePattern.Pattern) as ValuePattern;

string titulo1 = titulo.Current.Value.ToString();

}

// CADA FILA

for (int i = 0; i < conteo-1; i++)

{

var Celda = FILA[i].FindAll(TreeScope.Children,

new PropertyCondition(AutomationElement.ControlTypeProperty, ControlType.Custom));

// SELECT CELL

AutomationElement cc = Celda[2];

((InvokePattern)cc.GetCurrentPattern(InvokePattern.Pattern)).Invoke();

for (var j = 0; j < Celda.Count; j++)

{

ValuePattern CeldaX = Celda[j].GetCurrentPattern(ValuePattern.Pattern) as ValuePattern;

string value = CeldaX.Current.Value;

// PROBAR si se permite modificar

if (value =="222") CeldaX.SetValue("aa");

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT ROW

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

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

AutomationElement row1 = Ref1.FindFirst(TreeScope.Children,

new PropertyCondition(AutomationElement.NameProperty, "Row 3"));

// get row header

AutomationElement row1Header = row1.FindFirst(TreeScope.Children,

new PropertyCondition(AutomationElement.ControlTypeProperty, ControlType.Header));

// invoke it (select the whole line)

((InvokePattern)row1Header.GetCurrentPattern(InvokePattern.Pattern)).Invoke();

OTRA OPCION

AutomationElement row1Header = FILA[3].FindFirst(TreeScope.Children,

new PropertyCondition(AutomationElement.ControlTypeProperty, ControlType.Header));

// invoke it (select the whole line)

((InvokePattern)row1Header.GetCurrentPattern(InvokePattern.Pattern)).Invoke();

AutomationElement cc = Celdas[0];

((InvokePattern)cc.GetCurrentPattern(InvokePattern.Pattern)).Invoke();

((InvokePattern)cc.GetCurrentPattern(InvokePattern.Pattern)).Invoke();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UPDATE CELL

var targetText = Celda[2].FindFirst(TreeScope.Children,

new PropertyCondition(AutomationElement.ClassNameProperty, "TextBlock"));

= = = = = =

int id = System.Diagnostics.Process.GetProcessesByName("Book")[0].Id;

AutomationElement desktop = AutomationElement.RootElement;

AutomationElement bw = desktop.FindFirst(TreeScope.Children, new

PropertyCondition(AutomationElement.ProcessIdProperty, id));

AutomationElement datagrid = bw.FindFirst(TreeScope.Children, new

PropertyCondition(AutomationElement.AutomationIdProperty, "lv"));

AutomationElementCollection lines = datagrid.FindAll(TreeScope.Children, new

PropertyCondition(AutomationElement.ControlTypeProperty, ControlType.DataItem));

AutomationElementCollection items = lines[1].FindAll(TreeScope.Children, new PropertyCondition(AutomationElement.ControlTypeProperty, ControlType.Custom));

GridItemPattern pattern = items[1].GetCurrentPattern(GridItemPattern.Pattern) as GridItemPattern;

TableItemPattern tablePattern = items[1].GetCurrentPattern(TableItemPattern.Pattern) as TableItemPattern;