

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Xml;

using System.Xml.XPath;

namespace XMLTelephoneBook

{

class Program

{

static void Main(string[] args)

{

var sw = new StreamWriter("TelephoneBook.xml", false);

new Thread(new ParameterizedThreadStart(StartWriter)).Start(sw);

Console.ReadKey();

}

static public void ShowXMLDocument()

{

var xr = new XmlDocument();

xr.Load("TelephoneBook.xml");

var xnode=xr.DocumentElement;

Console.WriteLine("Name {0,-5} {1,-5}; Value {2,-5}",xnode.NodeType, xnode.Name, xnode.Value);

Recursive(xnode);

}

static public void Recursive(XmlNode element)

{

foreach (XmlNode node in element.ChildNodes)

{

Console.WriteLine("Name {0,-5} {1,-5}; Value {2,-5}", node.NodeType, node.Name, node.Value);

if(node.Attributes!=null)

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

{

Console.ForegroundColor=ConsoleColor.Red;

Console.WriteLine("Name {0,-5} {1,-5}; Value {2,-5}",node.Attributes[i].NodeType, node.Attributes[i].Name, node.Attributes[i].Value);

}

Console.ForegroundColor = ConsoleColor.Gray;

if (node.HasChildNodes)

Recursive(node);

}

}

static public void StartWriter(object obj)

{

var fs = obj as TextWriter;

var xrt = new XmlTextWriter(fs)

{

Formatting = Formatting.Indented,

Indentation = 1,

IndentChar = ' ',

};

xrt.WriteStartDocument();

xrt.WriteStartElement("MyContacts");

var rand = new Random();

for (int i = 0; i < 10; i++)

{

xrt.WriteStartElement("Contact");

for (int i1 = 0; i1 < 3; i1++)

{

xrt.WriteStartAttribute("TelephoneNumber"+i1);

xrt.WriteString(095 + " " + rand.Next(1000000));

xrt.WriteEndAttribute();

}

string name = null;

var n = rand.Next(10);

for (int i1 = 0; i1 < n; i1++)

{

var ch = rand.Next(97, 122);

name += **Char**.ConvertFromUtf32(ch);

}

xrt.WriteString(name);

xrt.WriteEndElement();

}

xrt.Close();

ShowXMLDocument();

Console.WriteLine("End ");

}

}

}



static public void ShowTelephone()

{

var xtr = new XmlTextReader("TelephoneBook.xml");

var doc = new XmlDocument();

doc.Load(xtr);

var navigator = doc.CreateNavigator();

var iterator = navigator.Select("MyContacts/Contact");

while (iterator.MoveNext())

{

Console.WriteLine("1 " + iterator.Current.Name);

}

Console.WriteLine("End");

xtr = new XmlTextReader("TelephoneBook.xml");

try

{

while (xtr.Read())

{

if (xtr.NodeType == XmlNodeType.Element)

{

if (xtr.HasAttributes)

{

while (xtr.MoveToNextAttribute())

{

Console.Write(xtr.GetAttribute(xtr.Name) + " ");

}

Console.WriteLine();

}

}

}

}

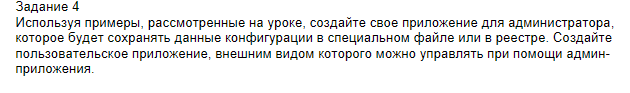
catch (Exception e)

{

throw e;

}

}



using System;

using System.Collections;

using System.Collections.Generic;

using System.Collections.Specialized;

using System.Configuration;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Xml;

using System.Xml.XPath;

using Microsoft.Win32;

namespace WPF\_RegisterAPP

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

this.Loaded += MainWindow\_Loaded;

this.Closed += MainWindow\_Closed;

}

void MainWindow\_Closed(object sender, EventArgs e)

{

try

{

var doc = new XmlDocument();

doc.Load(@"C:\Users\Виктор\Documents\Visual Studio 2013\Projects\HomeWork\WPF\_RegisterAPP\App.config");

XmlNodeList node = doc.SelectNodes("configuration/appSettings/add");

((XmlElement)node[0]).SetAttribute("value", Width.ToString());

((XmlElement)node[1]).SetAttribute("value", Height.ToString());

doc.Save(@"C:\Users\Виктор\Documents\Visual Studio 2013\Projects\HomeWork\WPF\_RegisterAPP\App.config");

WriteToRegister();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

void MainWindow\_Loaded(object sender, RoutedEventArgs routedEventArgs)

{

try

{

NameValueCollection list = ConfigurationManager.AppSettings;

//this.Width = Convert.ToDouble(list["Width"]);

//this.Height = Convert.ToDouble(list["Height"]);

RegistryKey reg = Registry.CurrentUser;

RegistryKey openReg = reg.OpenSubKey(@"Sowtware\MyKey");

this.Width = Convert.ToDouble(openReg.GetValue("Width", 100));

this.Height = Convert.ToDouble(openReg.GetValue("Height", 100));

}

catch (Exception e)

{

Debug.Print(e.Message);

}

}

void WriteToRegister()

{

RegistryKey reg = Registry.CurrentUser;

RegistryKey regOpen=reg.OpenSubKey(@"Sowtware\MyKey",true);

if (regOpen == null)

{

RegistryKey regK = reg.CreateSubKey(@"Sowtware\MyKey");

regK.SetValue("Width", Width.ToString());

regK.SetValue("Height", Height.ToString());

}

else

{

regOpen.SetValue("Width", Width.ToString());

regOpen.SetValue("Height", Height.ToString());

}

reg.Close();

}

bool HasElement(string str)

{

return new Regex(@"Width").IsMatch(str);

}

void AddElement()

{

var doc = new XmlDocument();

doc.Load(@"C:\Users\Виктор\Documents\Visual Studio 2013\Projects\HomeWork\WPF\_RegisterAPP\App.config");

XmlNode node = doc.DocumentElement;

string str = doc.InnerXml;

if (HasElement(str))

return;

XmlNode appSet=node.AppendChild(doc.CreateElement("appSettings"));

XmlElement newNode = appSet.AppendChild(doc.CreateElement("add")) as XmlElement;

newNode.SetAttribute("key", "Width");

newNode.SetAttribute("value", Width.ToString());

newNode = appSet.AppendChild(doc.CreateElement("add")) as XmlElement;

newNode.SetAttribute("key", "Height");

newNode.SetAttribute("value", Height.ToString());

node.AppendChild(appSet);

doc.Save(@"C:\Users\Виктор\Documents\Visual Studio 2013\Projects\HomeWork\WPF\_RegisterAPP\App.config");

MessageBox.Show("Elem add");

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

AddElement();

}

}

}