Assignment 2 Week 1

PART-B

1. Create C# program demonstrates Windows Service- Service should read data from a text file - use file watcher and store it in an output XML file. Prepare your data schema in source (text) and destination files (XML)

Solution:

The solution requires creation of a Windows service. My WS has a class myservice and two methods:

* Public void filwatch()
* Private void OnChanged(object source, FileSystemEventArgs e)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Linq;

using System.ServiceProcess;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Security.Permissions;

using System.Xml.Linq;

namespace Copytext

{

public partial class myservice : ServiceBase

{

//filewatcher created public for common access

FileSystemWatcher watcher = new FileSystemWatcher();

public myservice()

{

InitializeComponent();

}

protected override void OnStart(string[] args)

{

filwatch(); //used to start the service

}

protected override void OnStop()

{

//stop service by stopping watcher

watcher.EnableRaisingEvents = false;

}

//set permissions for working as windows service anonymously

[PermissionSet(SecurityAction.Demand, Name = "FullTrust")]

public void filwatch()

{

watcher.Path = @"D:\";

watcher.EnableRaisingEvents = true;

//set triggering occations

watcher.NotifyFilter = NotifyFilters.LastWrite | NotifyFilters.FileName | NotifyFilters.DirectoryName;

//source

watcher.Filter = "friend.txt";

//pass the method to be handled on notification

watcher.Changed += new FileSystemEventHandler(OnChanged);

watcher.Created += new FileSystemEventHandler(OnChanged);

Console.WriteLine();

}

private void OnChanged(object source, FileSystemEventArgs e)

{

XElement root = new XElement("root");

//fetch text and start wit hxml creation

foreach (String ln in File.ReadAllLines(@"D:\friend.txt"))

{

string[] fields = ln.Split(',');

//split into string to load in as elements

XElement record = new XElement("record");

int pos = 0;

foreach (String sp in fields)

{

pos += 1;

//load new values into the xml on change

XElement field = new XElement(string.Format("field\_{0}", pos.ToString()));

field.Add(sp);

record.Add(field);

}

//append to root

root.Add(record);

}

root.Save(AppDomain.CurrentDomain.BaseDirectory + @"xdocfriend.xml");

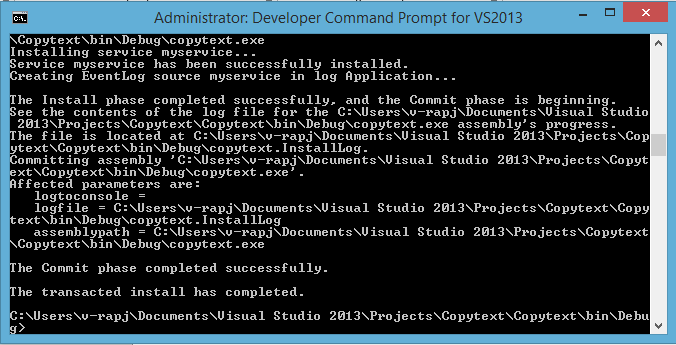
//saved to the solution folder

}

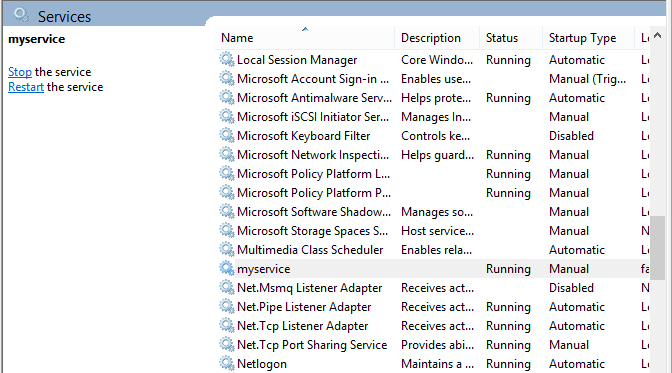
}

}

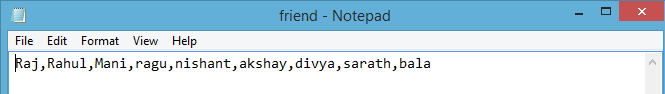
**INSTALL OF SERVICE:**



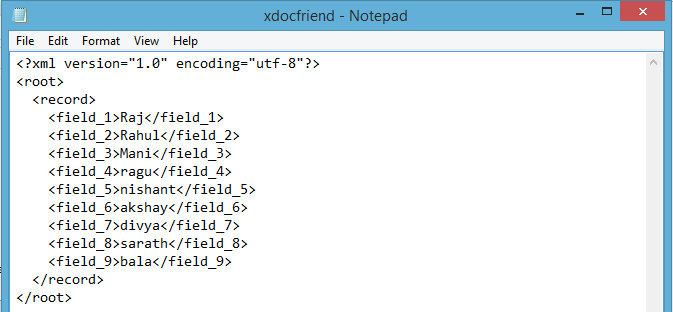
Service running status:



Add elements by comma



The Xml file generated after save is:



2.Create C# program demonstrates Producer / Consumer Problem

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace Week1Assignment2ansreal

{

class producerconsumer

{

static void Main()

{

Object myobj = new object();

Queue<string> queue = new Queue<string>();

Producer p = new Producer(queue);

Comsumer c1 = new Comsumer(queue, myobj, "consumer1");

Comsumer c2 = new Comsumer(queue, myobj, "consumer2");

Thread t1 = new Thread(c1.consume);

Thread t2 = new Thread(c2.consume);

t1.Start();

t2.Start();

Thread tp = new Thread(p.produce);

tp.Start();

Console.ReadLine();

}

}

public class Producer

{

Queue<string> queuestring;

int seq=0;

public Producer(Queue<string> queue)

{

queuestring = queue;

}

public void produce()

{

while (seq++ < 10)

{

string item = "item";

queuestring.Enqueue(item+seq);

Console.WriteLine("Producing {0}", seq);

Thread.Sleep(1000);

}

}

}

public class Comsumer

{

Queue<string> queuestring;

Object consumobj;

string namecon;

public Comsumer(Queue<string> queue, Object myobj, string name)

{

queuestring = queue;

consumobj = myobj;

namecon = name;

}

public void consume()

{

string element;

while (true)

{

lock (consumobj)

{

if (queuestring.Count == 0)

{

continue;

}

element = queuestring.Dequeue();

Console.WriteLine(" {0} is Comsuming {1}", namecon, element);

Thread.Sleep(1000);

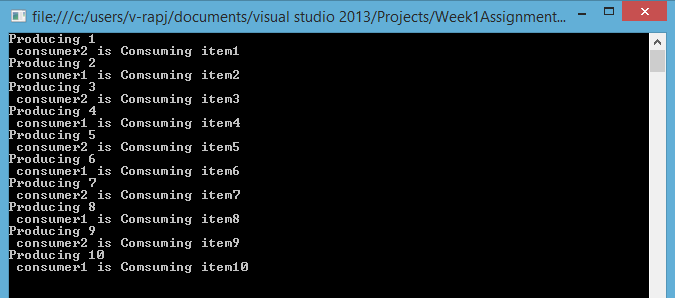
}

}

}

}

}



4. Socket Server using threads

SERVER:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Sockets;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Net;

using System.Threading;

namespace socketServer

{

class Program

{

static IPAddress ipAddress = Dns.GetHostEntry("localhost").AddressList[0];

TcpListener tcpListener = new TcpListener(ipAddress, 8080);

static void Main(string[] args)

{

Program obj = new Program();

obj.starthread();

}

void starthread()

{

tcpListener.Start();

Console.WriteLine("This is a Server program");

Thread newThread = new Thread(new ThreadStart(MyListeners));

newThread.Start();

}

void MyListeners()

{

Socket mySocketforClient = tcpListener.AcceptSocket();

if(mySocketforClient.Connected)

{

Console.WriteLine("Client is now connected to server.");

NetworkStream networkStream = new NetworkStream(mySocketforClient);

StreamWriter streamWriter =new StreamWriter(networkStream);

StreamReader streamReader =new StreamReader(networkStream);

bool run=true;

while(run)

{

string theString = streamReader.ReadLine();

Console.WriteLine("Message recieved by client: {0}",theString);

if (theString == "Quit")

run=false;

}

streamReader.Close();

networkStream.Close();

streamWriter.Close();

}

mySocketforClient.Close();

Console.WriteLine("Press any key to exit from server program");

Console.ReadKey();

}

}

}

CLIENT:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Sockets;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Threading;

namespace ClientofSocketServer

{

class Program

{

StreamReader streamReader;

StreamWriter streamWriter;

NetworkStream networkStream;

TcpClient mySocketforServer = new TcpClient("localHost", 8080);

void myclient()

{

networkStream = mySocketforServer.GetStream();

streamReader =new StreamReader(networkStream);

streamWriter =new StreamWriter(networkStream);

Console.WriteLine("This is client program");

Thread newThread = new Thread(new ThreadStart(myClientWork));

newThread.Start();

}

void myClientWork()

{

try

{

Console.WriteLine("type:");

string str = Console.ReadLine();

string output = "";

while (str != "Quit"&&output!="Quit")

{

streamWriter.WriteLine(str);

streamWriter.Flush();

Console.WriteLine("command to server:");

str = Console.ReadLine();

}

if (str == "Quit")

{

streamWriter.WriteLine(str);

streamWriter.Flush();

}

}

catch

{

Console.WriteLine("Exception reading from Server");

}

finally

{

networkStream.Close();

Console.WriteLine("Enter any key to quit");

Console.ReadKey();

}

}

static void Main(string[] args)

{

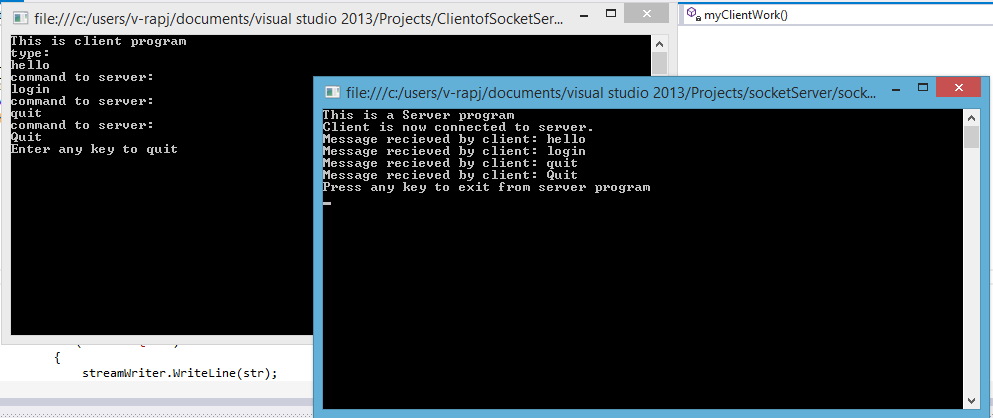
Program obj = new Program();

obj.myclient();

}

}

}



5.

5. You need to react when a file system change is detected in a specific path.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace Week1Assignment2ansreal

{

class Class16

{

public static void Main()

{

FileSystemWatcher mywatch = new FileSystemWatcher();

mywatch.Path = @"D:\";

mywatch.Filter = "mycollect.txt";

mywatch.NotifyFilter = NotifyFilters.LastAccess | NotifyFilters.LastWrite | NotifyFilters.FileName | NotifyFilters.DirectoryName;

mywatch.Changed += new FileSystemEventHandler(OnChanged);

mywatch.Created += new FileSystemEventHandler(OnChanged);

mywatch.Deleted += new FileSystemEventHandler(OnChanged);

mywatch.Renamed += new RenamedEventHandler(OnRenamed);

mywatch.EnableRaisingEvents = true;

Console.WriteLine("Press \'q\' to quit the sample.");

while (Console.Read() != 'q') ;

Console.ReadKey();

}

private static void OnChanged(object source, FileSystemEventArgs e)

{

Console.WriteLine("File: " + e.FullPath + " " + e.ChangeType);

}

private static void OnRenamed(object source, RenamedEventArgs e)

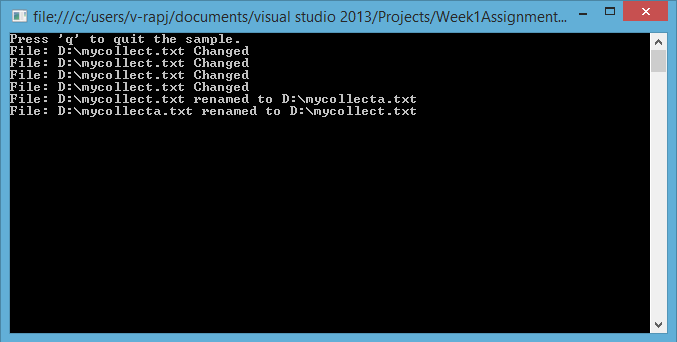
{

Console.WriteLine("File: {0} renamed to {1}", e.OldFullPath, e.FullPath);

}

}

}



12. Create XML Schema for one or more C# Classes. And generate classes from schema. (xsd tool)

Required: an xml file in this case it is xdocfriend.xml

On DCP for VS2013 go to the directory of the xml file

* Xsd xdocfriend.xml
* Xsd xdocfriend.xml /classes

The first creates an xml file and the latter creates a .cs file

