

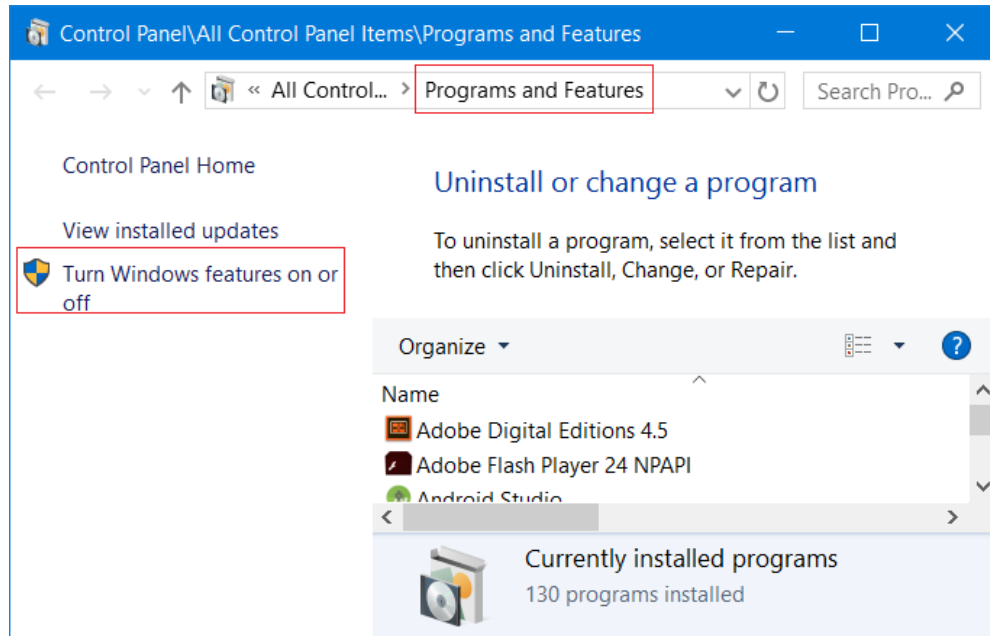
# MESSAGE ORIENTED ARCHITECTURE

## MICROSOFT MESSAGE QUEUE (MSMQ)

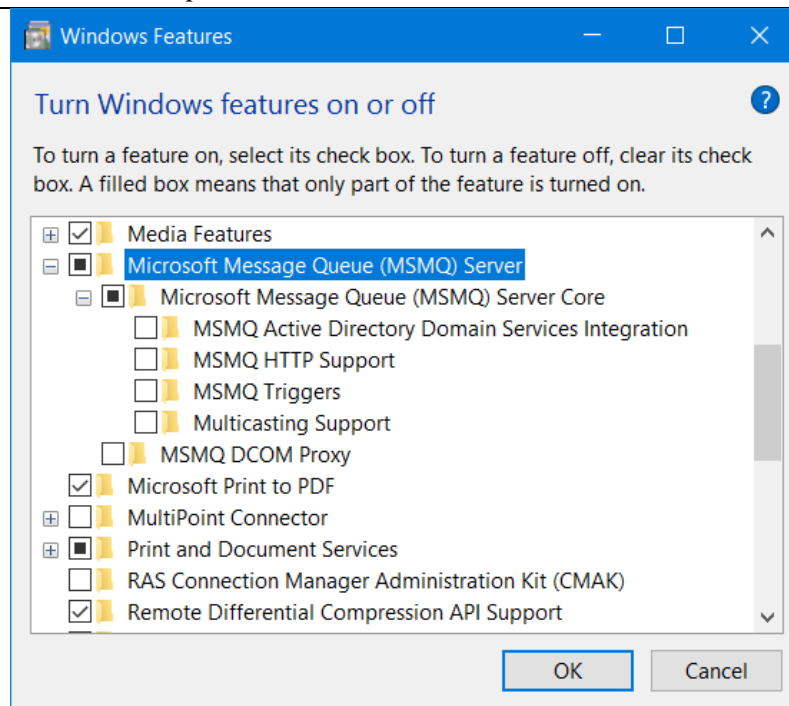
### CẤU HÌNH

Hướng dẫn cấu hình và làm việc với MSMQ trong Windows

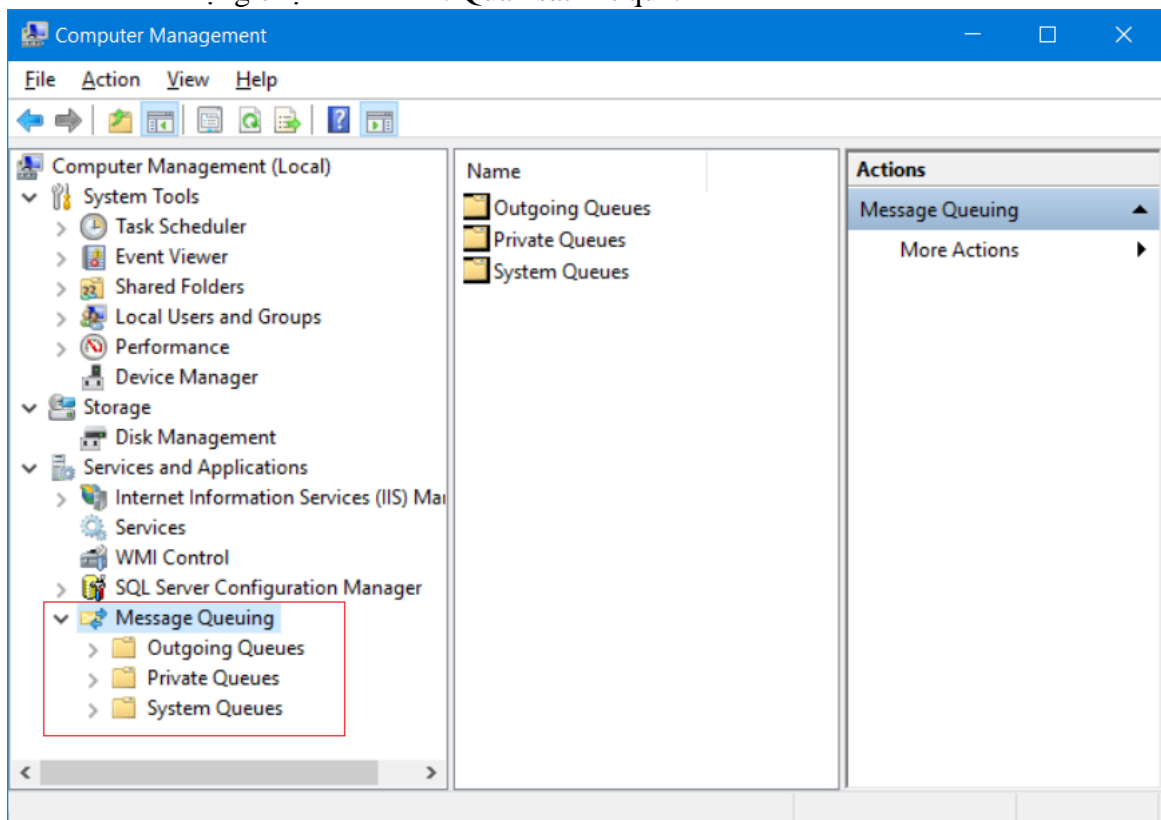
1. Mở Control Panel, chọn “Programs and Features” rồi chọn “Turn Windows features on or off”



Chọn như hình



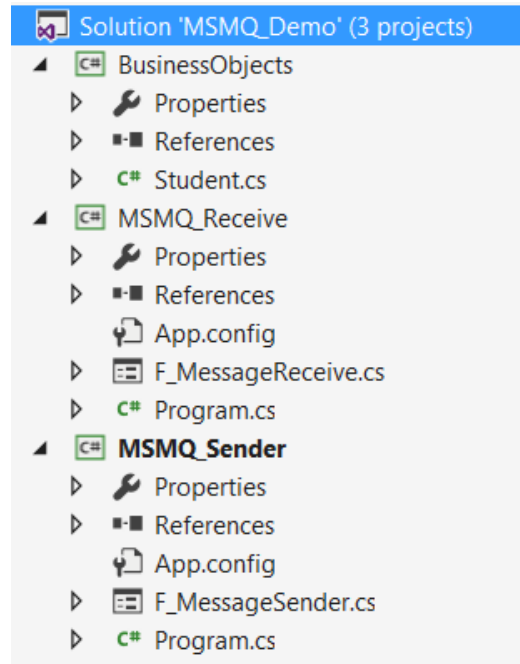
2. Mở Computer Management bằng cách nhấn chuột phải lên My Computer rồi chọn Manage. Sau đó mở rộng chọn như hình. Quan sát kết quả.



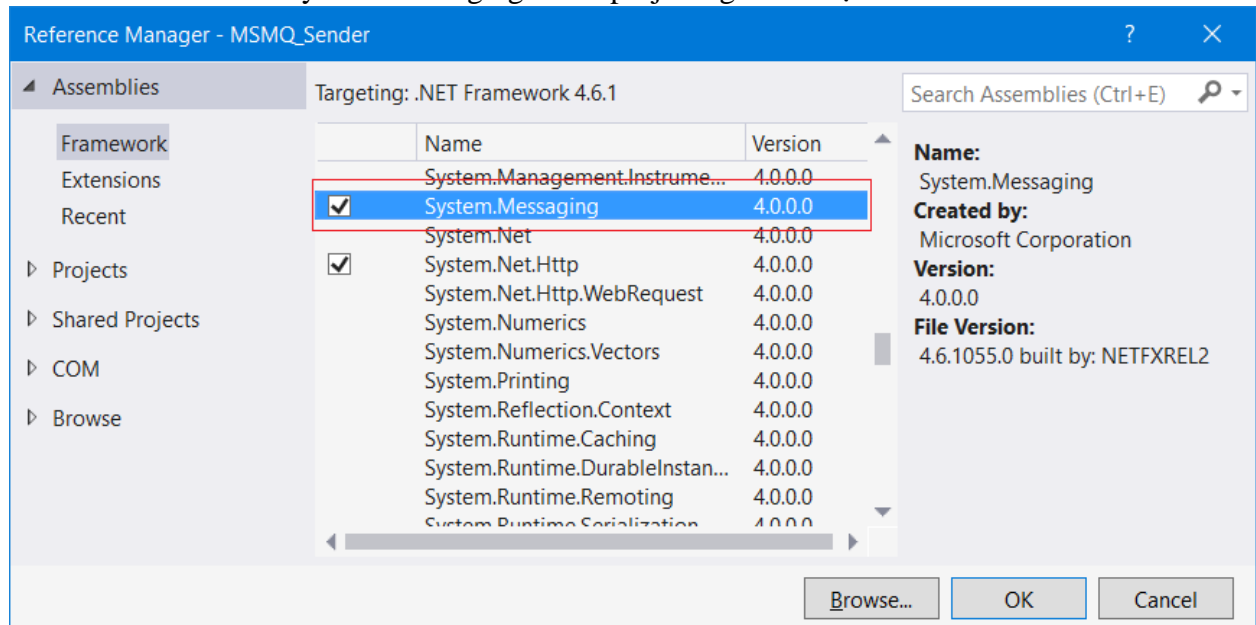
## ỨNG DỤNG MINH HỌA

Trong ví dụ này chúng ta làm một ứng dụng về gửi/nhận message dạng text và dạng object  
Tạo 3 Project C# mới:

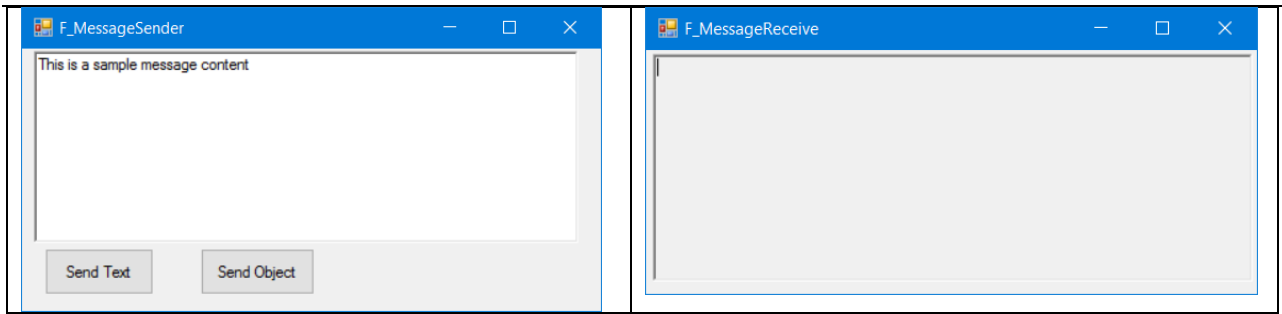
- Một cho việc gửi message
- Một cho việc nhận message.
- Một để chứa business objects



Thêm Reference đến System.Messaging cho 2 projects gửi và nhận như hình



Giao diện:



Code cho đối tượng dùng để gửi (business object)

```
using System;
namespace BusinessObjects
{
    [Serializable] //chú ý đối tượng phải được Serialize
    public class Student
    {
        public long StudentId { get; set; }
        public string FullName { get; set; }
        public DateTime DOB { get; set; }
        public Student():this(0,"no-name",new DateTime())
        {
        }
        public Student(long id,string fname, DateTime dob)
        {
            StudentId = id;FullName = fname;DOB = dob;
        }
        public override string ToString()
        {
            return FullName+"\t"+DOB;
        }
    }
}
```

Code cho form send message

```
using BusinessObjects;
using System;
using System.Messaging;
using System.Windows.Forms;
namespace MSMQ_Sender
{
    public partial class F_MessageSender : Form
    {
        MessageQueue queue = null;
        public F_MessageSender()
        {
            InitializeComponent();
            init();
        }

        private void init()
        {
            string path = @".\private$\phongkehoach";
            //string path = @"hbmnl\private$\phongkehoach";
            if (MessageQueue.Exists(path))
            {
                queue = new MessageQueue(path, QueueAccessMode.Send);
            }
            else
            {
            }
        }
    }
}
```

```

        queue = MessageQueue.Create(path, true);
        queue.Label = "queue cho phong ke hoach";
    }
    private void sendButton_Click(object sender, EventArgs e)
    {
        string message = richTextBox1.Text;
        MessageQueueTransaction transaction = new MessageQueueTransaction();
        transaction.Begin();
        queue.Send(message, transaction);
        transaction.Commit();
    }
    private void SendObjectButton_Click(object sender, EventArgs e)
    {
        Student st = new Student(1001L, "Nguyễn văn Tèo", new DateTime(1999, 10, 15));
        MessageQueueTransaction transaction = new MessageQueueTransaction();
        transaction.Begin();
        queue.Send(st, transaction);
        transaction.Commit();
    }
}
}
}

```

## Code cho form receive message

```

using BusinessObjects;
using System.Messaging;
using System.Windows.Forms;
namespace MSMQ_Receive
{
    public partial class F_MessageReceive : Form
    {
        private MessageQueue queue;
        public F_MessageReceive()
        {
            InitializeComponent();
            init_queue();
        }
        void init_queue()
        {
            string path = @".\private$\phongkehoach";
            queue = new MessageQueue(path);
            queue.BeginReceive();
            queue.ReceiveCompleted += Queue_ReceiveCompleted;
        }

        private void Queue_ReceiveCompleted(object sender, ReceiveCompletedEventArgs e)
        {
            var msg = e.Message;
            int type = msg.BodyType;
            XmlMessageFormatter fmt = new XmlMessageFormatter(
                new System.Type[] { typeof(string), typeof(Student) }
            );
            msg.Formatter = fmt;
            var result = msg.Body;
            var t = result.GetType();
            if (t.Equals(typeof(Student)))
            {
                SetText(t + ":" + result);
            }
            else

```

```

        SetText(""+result);
        queue.BeginReceive();//loop back
    }

    delegate void SetTextCallback(string text);
    private void SetText(string text)
    {
        // InvokeRequired required compares the thread ID of the
        // calling thread to the thread ID of the creating thread.
        // If these threads are different, it returns true.
        if (this.MessagesRichTextBox.InvokeRequired)
        {
            SetTextCallback callback = new SetTextCallback(SetText);
            this.Invoke(callback, new object[] { text });
        }
        else
        {
            this.MessagesRichTextBox.AppendText(text + "\n");
        }
    }
}

```

Thực thi đồng thời hai chương trình rồi nhập nội dung cần gửi sau đó nhấn nút gửi. Quan sát kết quả.

## TRUY XUẤT QUA MẠNG

Cấu hình cho 2 máy có thể truy xuất qua mạng (LAN/internet)

Chú ý đường dẫn

```

string path = @"FormatName:Direct:OS:serverName\private$\queue_name";
hoặc
string path = @"FormatName:Direct:TCP:serverIPAddress\private$\queue_name";

```

Nếu truy xuất server từ xa, ta không thể kiểm tra queue có tồn tại hay không cũng như tạo queue mới, nên code phải modify lại

```

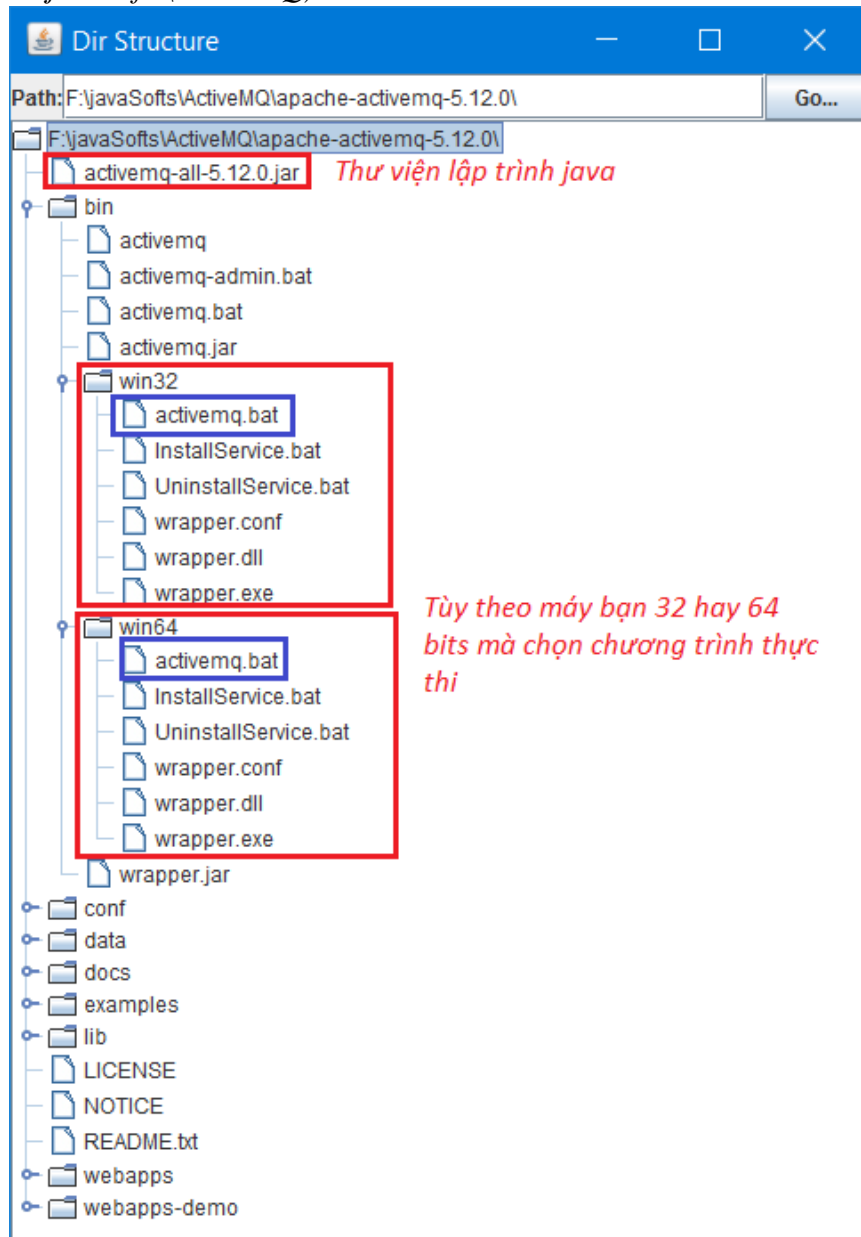
private void init()
{
    string path = @"FormatName:Direct:OS:serverName\private$\queue_name";

    queue = new MessageQueue(path, QueueAccessMode.Send);
}

```

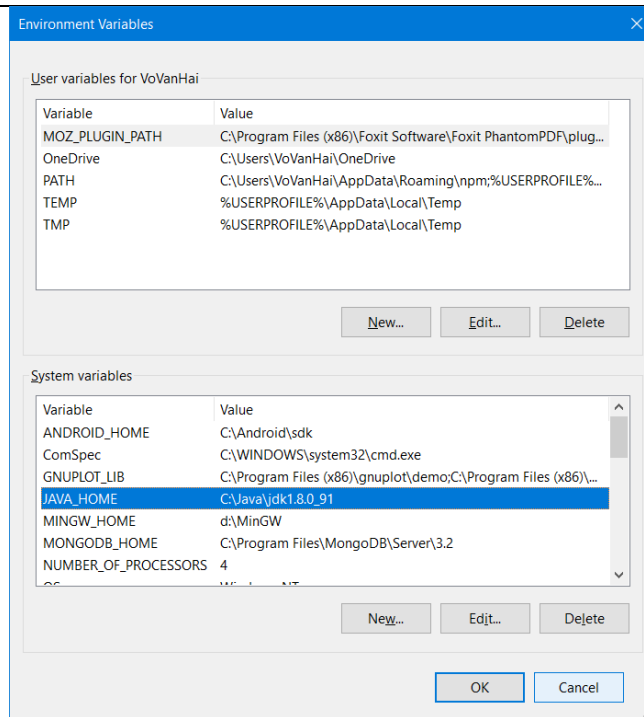
# APACHE ACTIVEMQ

Download tại <http://activemq.apache.org/> sau đó tiến hành giải nén ra một thư mục nào đó (giả sử F:\javaSofts\ActiveMQ\apache-activemq-5.12.0\)



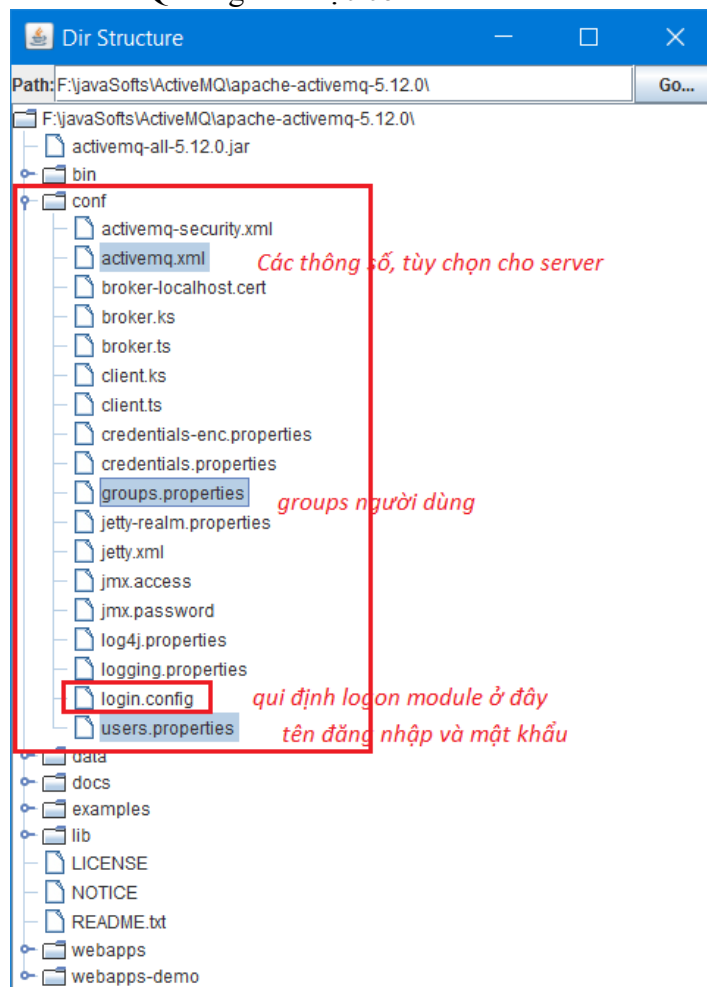
## CẤU HÌNH

1. Cần cấu hình biến môi trường JAVA\_HOME là thư mục nơi cài đặt JDK. Ví dụ như sau



## 2. Cấu hình cho server

Các file cấu hình của ActiveMQ trong thư mục conf





Cấu hình login module (bắt buộc login khi kết nối)

File **login.config**

```
activemq {
    org.apache.activemq.jaas.PropertiesLoginModule required
    org.apache.activemq.jaas.properties.user="users.properties"
    org.apache.activemq.jaas.properties.group="groups.properties"
    reload=true;
};
```

Cấu hình groups, file **groups.properties**

```
#group_name=users_list
admins=admin,teo
users=ty
dhcn=men
guests=guest
```

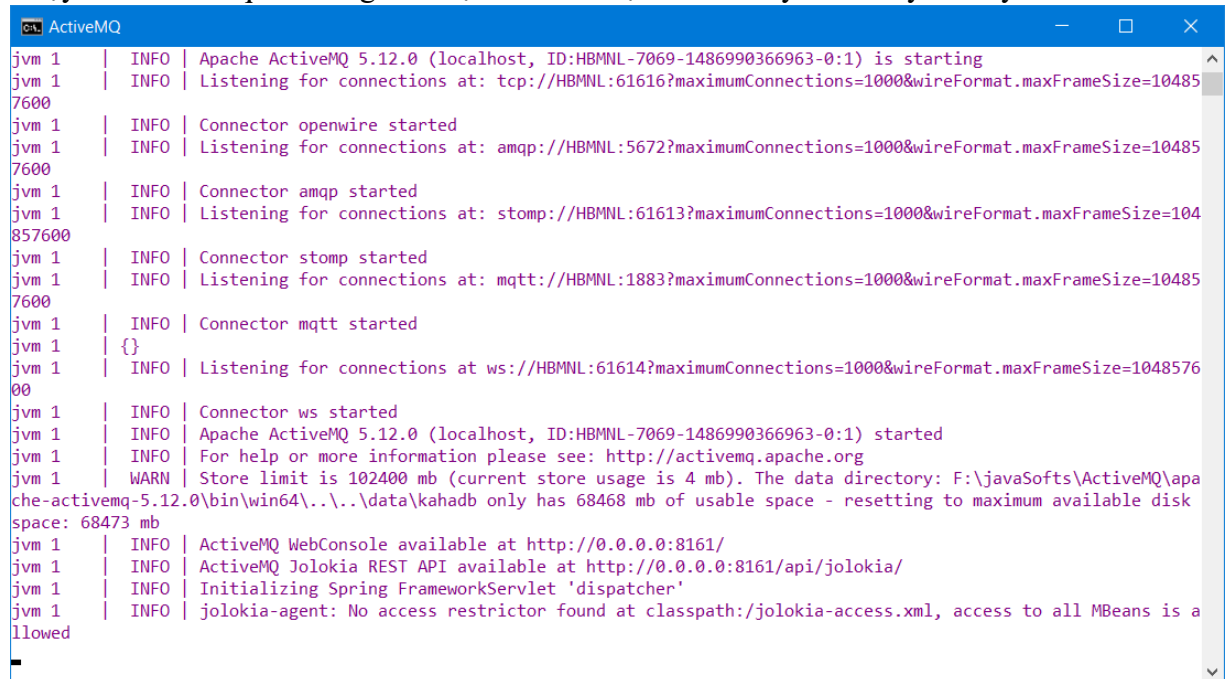
Cấu hình người dùng, file **users.properties**

```
#username=password
admin=admin
teo=123
ty=456
men=123
```

Đọc thêm phần security ở đây: <http://activemq.apache.org/security.html>

### 3. Thực thi

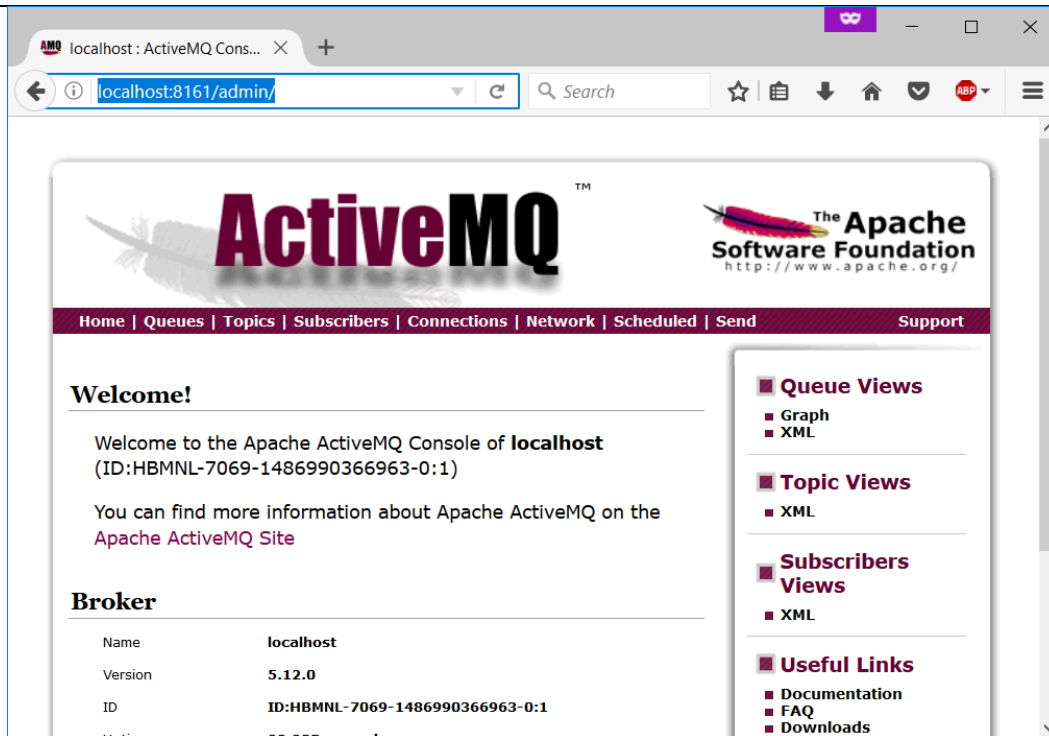
Chạy file activemq.bat trong thư mục Win32 hoặc Win64 tùy theo máy 32 hay 64 bits.



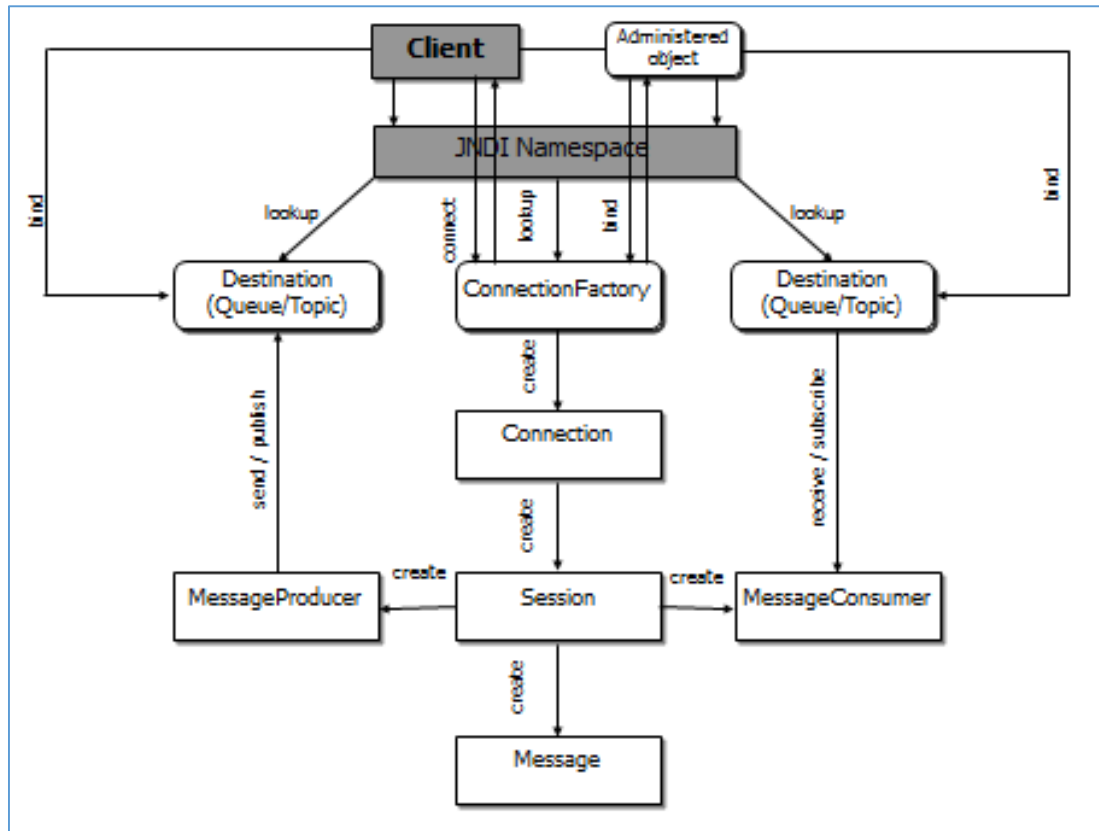
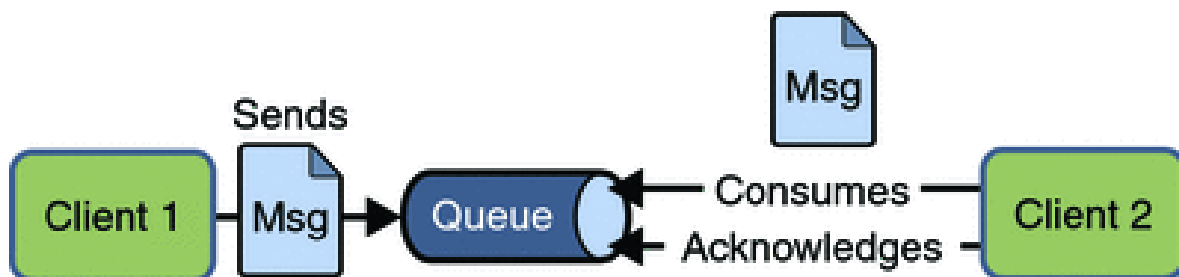
```
ActiveMQ
jvm 1 | INFO | Apache ActiveMQ 5.12.0 (localhost, ID:HBMNL-7069-1486990366963-0:1) is starting
jvm 1 | INFO | Listening for connections at: tcp://HBMNL:6161?maximumConnections=1000&wireFormat.maxFrameSize=104857600
jvm 1 | INFO | Connector openwire started
jvm 1 | INFO | Listening for connections at: amqp://HBMNL:5672?maximumConnections=1000&wireFormat.maxFrameSize=104857600
jvm 1 | INFO | Connector amqp started
jvm 1 | INFO | Listening for connections at: stomp://HBMNL:61613?maximumConnections=1000&wireFormat.maxFrameSize=104857600
jvm 1 | INFO | Connector stomp started
jvm 1 | INFO | Listening for connections at: mqtt://HBMNL:1883?maximumConnections=1000&wireFormat.maxFrameSize=104857600
jvm 1 | INFO | Connector mqtt started
jvm 1 | {}
jvm 1 | INFO | Listening for connections at ws://HBMNL:61614?maximumConnections=1000&wireFormat.maxFrameSize=104857600
jvm 1 | INFO | Connector ws started
jvm 1 | INFO | Apache ActiveMQ 5.12.0 (localhost, ID:HBMNL-7069-1486990366963-0:1) started
jvm 1 | INFO | For help or more information please see: http://activemq.apache.org
jvm 1 | WARN | Store limit is 102400 mb (current store usage is 4 mb). The data directory: F:\javaSofts\ActiveMQ\apache-activemq-5.12.0\bin\win64\..\data\kahadb only has 68468 mb of usable space - resetting to maximum available disk space: 68473 mb
jvm 1 | INFO | ActiveMQ WebConsole available at http://0.0.0.0:8161/
jvm 1 | INFO | ActiveMQ Jolokia REST API available at http://0.0.0.0:8161/api/jolokia/
jvm 1 | INFO | Initializing Spring FrameworkServlet 'dispatcher'
jvm 1 | INFO | jolokia-agent: No access restrictor found at classpath:/jolokia-access.xml, access to all MBeans is allowed
```

(lưu ý: không đóng cửa sổ này)

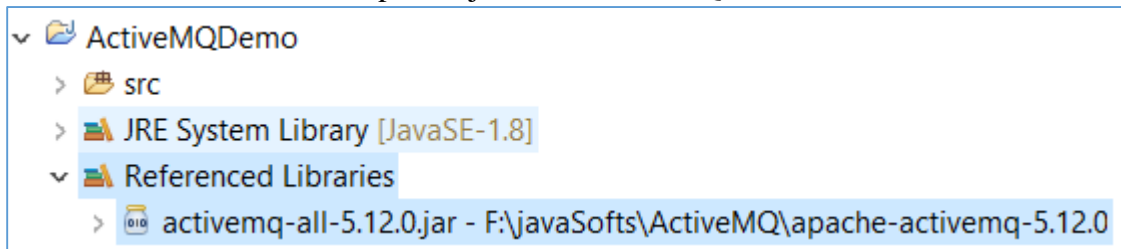
Mở cửa sổ browser, gõ địa chỉ: <http://localhost:8161/admin/>. Nếu có yêu cầu đăng nhập gõ username và password đều là admin



4. Đọc thêm tài liệu cấu hình tại <http://activemq.apache.org/getting-started.html>

**NGUYÊN LÝ LẬP TRÌNH****MÔ HÌNH POINT-TO-POINT****CODE MINH HỌA CHO GỬI VÀ NHẬN BẰNG JAVA**

1. Khởi động eclipse, chọn perspective là Java. Tạo project mới
2. Thêm tham chiếu đến thư viện lập trình java của ActiveMQ



3. Code helper

```

package helper;

import java.io.StringReader;
  
```

```

import java.io.StringWriter;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.Marshaller;
import javax.xml.bind.Unmarshaller;

public class XMLConvert<T> {
    private T type;

    public XMLConvert(T type) {
        this.type = type;
    }

    @SuppressWarnings("all")
    public T xml2Object(String xml) throws Exception{
        T sv=null;
        JAXBContext ctx= JAXBContext.newInstance(type.getClass());
        Unmarshaller ms = ctx.createUnmarshaller();
        sv=(T) ms.unmarshal(new StringReader(xml));
        return sv;
    }

    public String object2XML(T obj) throws Exception{
        JAXBContext ctx= JAXBContext.newInstance(type.getClass());
        Marshaller ms=ctx.createMarshaller();
        StringWriter sw = new StringWriter();
        ms.marshal(obj, sw);
        return sw.toString();
    }
}

```

## Sample object

```

package data;
import java.io.Serializable;
import java.util.Date;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

@XmlRootElement
@XmlType(propOrder={"mssv","hoten","ngaysinh"})
public class Person implements Serializable{
    private long mssv;
    private String hoten;
    private Date ngaysinh;
    public Person(long mssv, String hoten, Date ngaysinh) {
        this.mssv = mssv;
        this.hoten = hoten;
        this.ngaysinh = ngaysinh;
    }
    public Person() {
    }
    public long getMssv() {
        return mssv;
    }
    public void setMssv(long mssv) {
        this.mssv = mssv;
    }
    public String getHoten() {
        return hoten;
    }
}

```

```

    }
    public void setHoten(String hoten) {
        this.hoten = hoten;
    }
    public Date getNgaysinh() {
        return ngaysinh;
    }
    public void setNgaysinh(Date ngaysinh) {
        this.ngaysinh = ngaysinh;
    }
    @Override
    public String toString() {
        return mssv+"\t"+hoten+"\t"+ngaysinh;
    }
}

```

#### 4. Code Sender

```

import java.util.Date;
import java.util.Properties;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageProducer;
import javax.jms.Session;
import javax.naming.Context;
import javax.naming.InitialContext;
import org.apache.log4j.BasicConfigurator;
import data.Person; import helper.XMLConvert;

public class QueueSender {
    public static void main(String[] args) throws Exception{
        //config environment for JMS
        BasicConfigurator.configure();
        //config environment for JNDI
        Properties settings=new Properties();
        settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
            "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
        settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
        //create context
        Context ctx=new InitialContext(settings);
        //lookup JMS connection factory
        ConnectionFactory factory=
            (ConnectionFactory)ctx.lookup("ConnectionFactory");
        //lookup destination. (If not exist-->ActiveMQ create once)
        Destination destination=
            (Destination) ctx.lookup("dynamicQueues/thanthidet");
        //get connection using credential
        Connection con=factory.createConnection("admin","admin");
        //connect to MOM
        con.start();
        //create session
        Session session=con.createSession(
            /*transaction*/false,
            /*ACK*/Session.AUTO_ACKNOWLEDGE

```

```

    );
    //create producer
    MessageProducer producer = session.createProducer(destination);
    //create text message
    Message msg=session.createTextMessage("hello message from ActiveMQ");
    producer.send(msg);

    Person p=new Person(1001, "Thân Thị Đệ", new Date());
    String xml=new XMLConvert<Person>(p).object2XML(p);

    msg=session.createTextMessage(xml);
    producer.send(msg);
    //shutdown connection
    session.close();con.close();
    System.out.println("Finished...");
}
}

```

## 5. Code Receiver

```

package demo01.ex01;

import java.util.Properties;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageConsumer;
import javax.jms.MessageListener;
import javax.jms.ObjectMessage;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.Context;
import javax.naming.InitialContext;
import org.apache.log4j.BasicConfigurator;

public class QueueReceiver {
    public static void main(String[] args) throws Exception{
        //thiết lập môi trường cho JMS
        BasicConfigurator.configure();
        //thiết lập môi trường cho JNDI
        Properties settings=new Properties();
        settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
            "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
        settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
        //tạo context
        Context ctx=new InitialContext(settings);
        //lookup JMS connection factory
        Object obj=ctx.lookup("ConnectionFactory");
        ConnectionFactory factory=(ConnectionFactory)obj;
        //lookup destination
        Destination destination
            =(Destination) ctx.lookup("dynamicQueues/thanthidet");
        //tạo connection
        Connection con=factory.createConnection("admin","admin");
        //nối đến MOM
        con.start();
    }
}

```

```

//tạo session
Session session=con.createSession(
    /*transaction*/false,
    /*ACK*/Session.CLIENT_ACKNOWLEDGE
);

//tạo consumer

MessageConsumer receiver = session.createConsumer(destination);
//blocked-method for receiving message - sync
//receiver.receive();
//Cho receiver lắng nghe trên queue, chừng có message thì notify -
async
System.out.println("Tý was listened on queue...");
receiver.setMessageListener(new MessageListener() {
    @Override
    //có message đến queue, phương thức này được thực thi
    public void onMessage(Message msg) { //msg là message nhận được
        try {
            if(msg instanceof TextMessage){
                TextMessage tm=(TextMessage)msg;
                String txt=tm.getText();
                System.out.println("Nhận được "+txt);
                msg.acknowledge();//gửi tín hiệu ack
            }
            else if(msg instanceof ObjectMessage){
                ObjectMessage om=(ObjectMessage)msg;
                System.out.println(om);
            }
            //others message type....
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
}
}

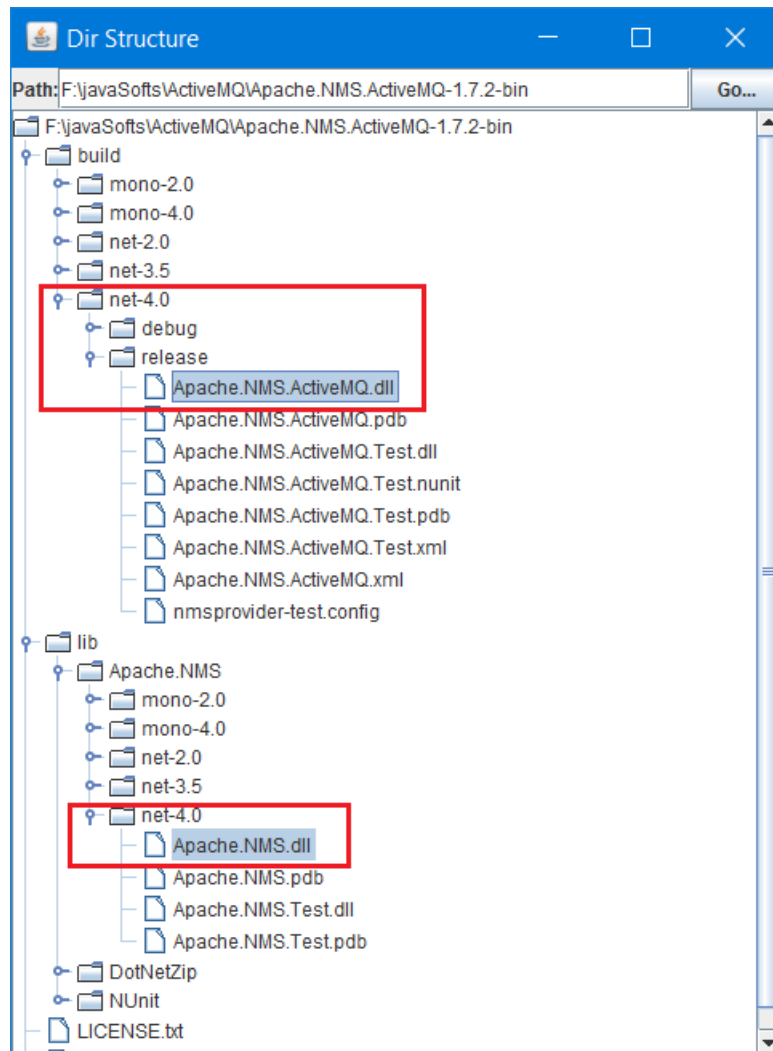
```

**CODE MINH HỌA CHO GỬI VÀ NHẬN BẰNG C#**

1. Download thư viện dùng cho .Net tại

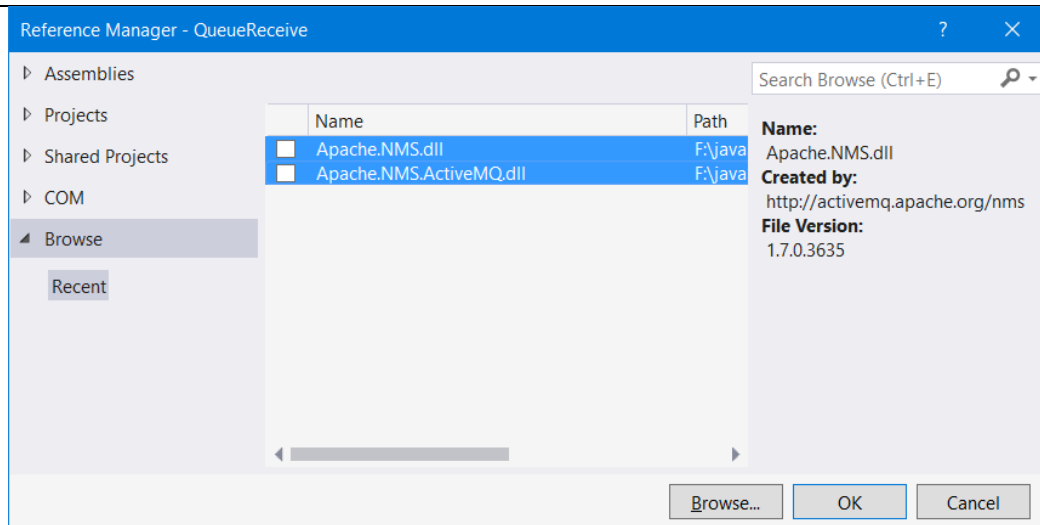
<https://cwiki.apache.org/confluence/display/NMS/ActiveMQ+Downloads>

Giải nén ta có cấu trúc thư mục như hình sau. Chú ý các chỗ đánh dấu là 2 thư viện chính ta cần làm việc.



2. Mở Visual Studio, tạo project mới, thêm tham chiếu như hình





## 3. Code helper

```
using System;

namespace ActiveMQSender
{
    [Serializable]
    public class Person
    {
        public long Mssv { get; set; }
        public string Hoten { get; set; }
        public DateTime Ngaysinh { get; set; }

        public Person() { }

        public Person(long mssv, string hoten, DateTime ngaysinh)
        {
            this.Mssv = mssv; Hoten = hoten; Ngaysinh = ngaysinh;
        }
    }
}

using System.Xml.Serialization;
using System.IO;
namespace ActiveMQSender
{
    public class XMLObjectConverter<T>
    {
        public string object2XML(T p)
        {
            string xml = "";
            XmlSerializer ser = new XmlSerializer(typeof(T));
            using (MemoryStream ms = new MemoryStream())
            {
                ser.Serialize(ms, p);
                ms.Position = 0;
                xml = new StreamReader(ms).ReadToEnd();
            }
            return xml;
        }
    }
}
```

## 4. Code cho sender

```
using System;
using System.IO;
using Apache.NMS;
using Apache.NMS.ActiveMQ;
using Apache.NMS.ActiveMQ.Commands;

namespace ActiveMQSender
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("sending message. Enter to exit.");
            //tạo connection factory
            IConnectionFactory factory = new
            ConnectionFactory("tcp://localhost:61616");
            //tạo connection
            IConnection con = factory.CreateConnection("admin", "admin");
            con.Start();//nối tới MOM
            //tạo session
            ISession session =
            con.CreateSession(AcknowledgementMode.AutoAcknowledge);
            //tạo producer
            ActiveMQQueue destination = new ActiveMQQueue("thanthidet");
            IMessageProducer producer = session.CreateProducer(destination);
            //send message
            //biến đối tượng thành XML document String
            Person p = new Person(1001, "Truong Van COi", new DateTime());
            //string xml = genXML(p).ToLower();
            string xml = new XMLObjectConverter<Person>().object2XML(p);

            Console.WriteLine(xml.ToLower());
            IMessage msg = new ActiveMQTextMessage("Hola mondo");
            producer.Send(msg);

            //shutdown
            session.Close();
            con.Close();
            Console.ReadKey();
        }
    }
}
```

## 5. Code cho Receiver

```
using System;
using Apache.NMS;
using Apache.NMS.ActiveMQ;
using Apache.NMS.ActiveMQ.Commands;

namespace ActiveMQDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("receiving message. Enter to exit.");
            //tạo connection factory
```

```

        IConnectionFactory factory = new
ConnectionFactory("tcp://localhost:61616");
        //tạo connection
        IConnection con = factory.CreateConnection("admin","admin");
        con.Start();//nối tới MOM
        //tạo session
        ISession session =
con.CreateSession(AcknowledgementMode.AutoAcknowledge);
        //tạo consumer
        ActiveMQQueue destination = new ActiveMQQueue("thanthidet");
        IMessageConsumer consumer = session.CreateConsumer(destination);
        //nhận message - lắng nghe
        consumer.Listener += Consumer_Listener;
        Console.ReadKey();
    }

    private static void Consumer_Listener(IMessage message)
    {
        if(message is ActiveMQTextMessage)
        {
            ActiveMQTextMessage msg = message as ActiveMQTextMessage;
            Console.WriteLine("receive:" +msg.Text);
        }
    }
}

```

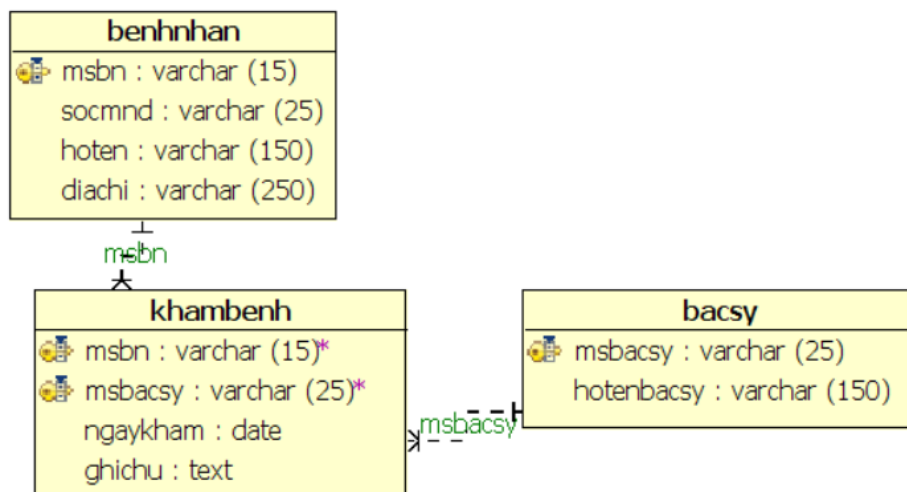
6. Thực thi xem kết quả

## THỰC THI CHÉO

1. Chạy Sender của Java, Receiver của C#
  2. Chạy Sender của C#, Receiver của Java
- Quan sát kết quả

## BÀI TẬP

1. Cho cơ sở dữ liệu quản lý thông tin bệnh nhân được cho như sau



Yêu cầu:

Chương trình được viết cho 2 loại nhân sự của phòng khám đặt tả như sau

**Tại máy tính của Nhân viên lễ tân (viết bằng C#):**

**NHẬN BỆNH**

Thông tin bệnh nhân

Mã số bệnh nhân:  ☐

Số CMND:  ☐

Họ và tên:

Địa chỉ:

Lưu thông tin

Khi bệnh nhân vào khám thì nhân viên lễ tân sẽ nhập thông tin của bệnh nhân nếu đến khám lần đầu hoặc tìm thông tin bệnh nhân nếu tái khám (tìm theo mã số bệnh nhân hoặc theo số CMND). Sau khi tìm thấy hoặc nhập vào, nhấn nút “Lưu thông tin” thì chương trình sẽ notify cho máy tính đặc trong phòng của bác sỹ khám (giả sử có n bác sỹ, mỗi bác sỹ có một máy chạy độc lập).

**Tại máy tính trong phòng bác sỹ (viết bằng Java)**

Khi có 1 bệnh nhân cần khám, chương trình sẽ đưa vào danh sách đợi để bác sỹ gọi vào.

**Bác sỹ khám bệnh**

Danh sách bệnh nhân chờ khám

20130918\_030  
20130918\_031  
20130918\_032  
20130918\_033  
20130918\_034

Gọi khám...

Thông tin bệnh nhân được chọn

Mã số bệnh nhân:

Số CMND:

Họ tên:

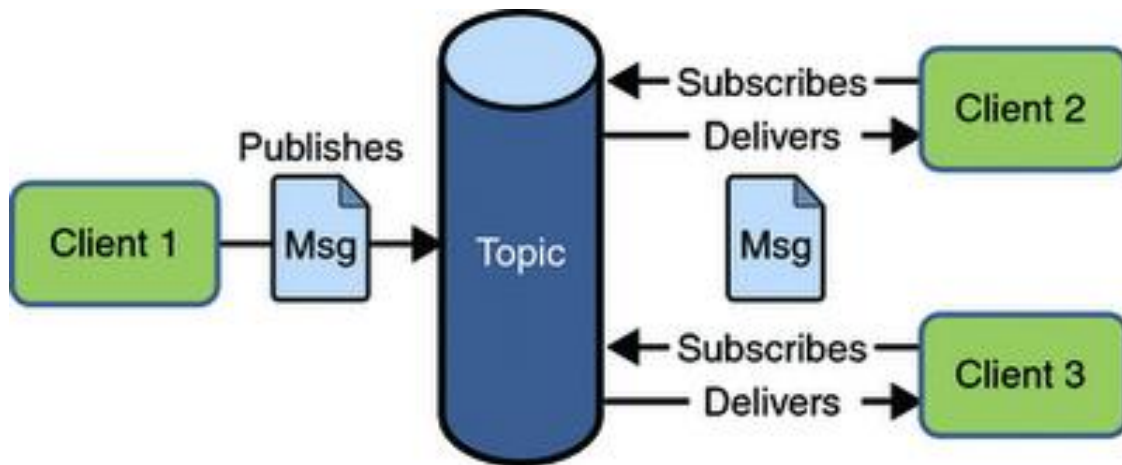
Địa chỉ:

Nội dung khám:

Cập nhật thông tin khám bệnh

- Giả sử ta có một công việc cần phân chia cho N máy tính làm riêng rẽ sau đó ta tổng hợp kết quả lại. Yêu cầu mô phỏng cho hệ thống này: lập tài liệu thiết kế, hiện thực chương trình.

## MÔ HÌNH PUBLISHER – SUBSCRIBER



### CODE MINH HỌA CHO GỬI VÀ NHẬN BẰNG JAVA

#### Publisher

```

import java.util.Properties;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageProducer;
import javax.jms.Session;
import javax.naming.Context;
import javax.naming.InitialContext;
import org.apache.log4j.BasicConfigurator;

public class TopicPublisher {
    public static void main(String[] args) throws Exception{
        //thiết lập môi trường cho JMS logging
        BasicConfigurator.configure();
        //thiết lập môi trường cho JNDI
        Properties settings=new Properties();
        settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
            "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
        settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
        //tạo context
        Context ctx=new InitialContext(settings);
        //lookup JMS connection factory
        Object obj=ctx.lookup("TopicConnectionFactory");
        ConnectionFactory factory=(ConnectionFactory)obj;
        //tạo connection
        Connection con=factory.createConnection("admin","admin");
        //nối đến MOM
        con.start();
        //tạo session
        Session session=con.createSession(
            /*transaction*/false,
            /*ACK*/Session.AUTO_ACKNOWLEDGE
        );
        Destination destination=(Destination)
        ctx.lookup("dynamicTopics/thanthidet");
    }
}

```

```

        //tạo producer
        MessageProducer producer = session.createProducer(destination);
        //Tạo 1 message
        Message msg=session.createTextMessage("xin chào người ẹp");

        //gửi
        producer.send(msg);
        //shutdown connection
        session.close();
        con.close();

        System.out.println("Finished...");
    }
}

```

### Subscriber

```

import java.util.Properties;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageConsumer;
import javax.jms.MessageListener;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.Context;
import javax.naming.InitialContext;
import org.apache.log4j.BasicConfigurator;

public class TopicSubscriber {
    public static void main(String[] args) throws Exception{
        //thiết lập môi trường cho JMS
        BasicConfigurator.configure();
        //thiết lập môi trường cho JNDI
        Properties settings=new Properties();
        settings.setProperty(Context.INITIAL_CONTEXT_FACTORY,
            "org.apache.activemq.jndi.ActiveMQInitialContextFactory");
        settings.setProperty(Context.PROVIDER_URL, "tcp://localhost:61616");
        //tạo context
        Context ctx=new InitialContext(settings);
        //lookup JMS connection factory
        Object obj=ctx.lookup("TopicConnectionFactory");
        ConnectionFactory factory=(ConnectionFactory)obj;
        //tạo connection
        Connection con=factory.createConnection("admin","admin");
        //nối đến MOM
        con.start();
        //tạo session
        Session session=con.createSession(
            /*transaction*/false,
            /*ACK*/Session.CLIENT_ACKNOWLEDGE
        );
        //tạo consumer
        Destination destination=(Destination) ctx.lookup("dynamicTopics/thanthidet");
        MessageConsumer receiver = session.createConsumer(destination);
    }
}

```

```

//receiver.receive();//blocked method
//Cho receiver lắng nghe trên queue, chừng có message thì notify
receiver.setMessageListener(new MessageListener() {
    @Override
    //có message đến queue, phương thức này được thực thi
    public void onMessage(Message msg) { //msg là message nhận được
        try {
            if(msg instanceof TextMessage){
                TextMessage tm=(TextMessage)msg;
                String txt=tm.getText();
                System.out.println("XML= "+txt);
                msg.acknowledge();//gửi tín hiệu ack
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
}
}

```

## CODE MINH HỌA CHO GỬI VÀ NHẬN BẰNG C#

### Publisher

```

using System;
using Apache.NMS;
using Apache.NMS.ActiveMQ;
using Apache.NMS.ActiveMQ.Commands;
namespace TopicSender
{
    class Program
    {
        static void Main(string[] args)
        {
            IConnectionFactory factory = new ConnectionFactory("tcp://localhost:61616");
            IConnection con = factory.CreateConnection("admin", "admin");
            con.Start();//nối tới MOM
            ISession session = con.CreateSession(AcknowledgementMode.AutoAcknowledge);

            ActiveMQTopic destination = new ActiveMQTopic("thanthidet");
            IMessageProducer producer = session.CreateProducer(destination);
            ActiveMQTextMessage tmsg = new ActiveMQTextMessage("hola. mondo");
            producer.Send(tmsg);
            Console.WriteLine("press any key to continue");
            Console.ReadKey();
        }
    }
}

```

### Subscriber

```

using System;
using Apache.NMS;
using Apache.NMS.ActiveMQ;
using Apache.NMS.ActiveMQ.Commands;
namespace TopicReceiver
{
    class Program

```

```

{
    static void Main(string[] args)
    {
        Console.WriteLine("press enter to exit");
        IConnectionFactory factory = new ConnectionFactory("tcp://localhost:61616");
        IConnection con = factory.CreateConnection("admin", "admin");
        con.Start();//nối tới MOM
        ISession session = con.CreateSession(AcknowledgementMode.AutoAcknowledge);

        ActiveMQTopic destination = new ActiveMQTopic("thanthidet");
        IMessageConsumer producer = session.CreateConsumer(destination);

        producer.Listener += Producer_Listener;
        Console.ReadLine();
    }

    private static void Producer_Listener(IMessage message)
    {
        if(message is ActiveMQTextMessage)
        {
            ActiveMQTextMessage tmsg = message as ActiveMQTextMessage;
            String txt = tmsg.Text;
            Console.WriteLine(txt);
        }
    }
}

```

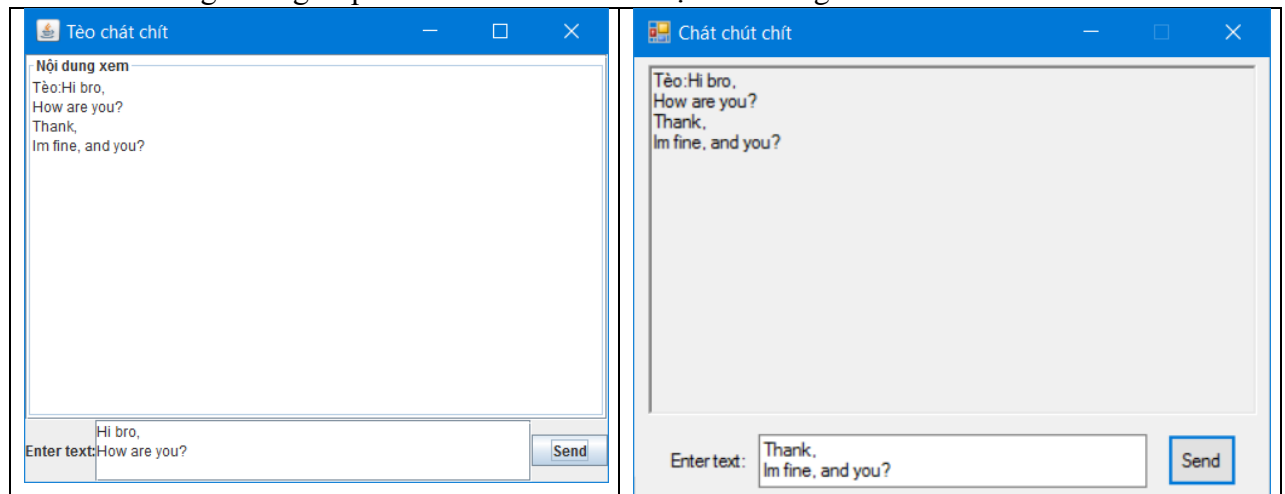
## THỰC THI CHÉO

1. Chạy Sender của Java, Receiver của C#
2. Chạy Sender của C#, Receiver của Java

Quan sát kết quả

## BÀI TẬP

1. Viết chương trình groups chat với chat client được viết bằng Java và C#



2. \*\*\* Viết chương trình bằng java thực hiện công việc như sau:  
Chương trình chia sẻ màn hình giữa một máy server và một tập các máy client.  
a. Máy server có thể trong thấy màn hình của tất cả các máy clients kết nối



- b. Khi máy server chọn chế độ trình diễn thì tất cả các máy client sẽ thấy được màn hình chia sẻ của máy server.

Hãy viết tài liệu thiết kế sau đó cài đặt hệ thống này

## SECURITY

### CẤU HÌNH

#### BÀI TẬP

1. Với bài tập 1 phần mô hình P2P,
  - Tạo 2 users có tên **LanHT** và **HungNV** với mật khẩu tương ứng là 123Cba và 321Abc thuộc vào 2 nhóm có tên **bacsy** và **letan**.
  - Cấu hình security sao cho nhân viên thuộc nhóm letan chỉ được phép gửi và nhân viên thuộc nhóm bacsy chỉ có quyền đọc queue **KhamBenh**.