

作者:周英科 email:li_li_li87222@163.com.qq:543246541

Red5

入门教程



简介:

在 flex,java 盛行天下时 red5 把二者连为一体成为更强的强者,为即时通信,流媒体开发,网络在线游戏提供了一个简单易行的方案。越来越多的公司把这个方案当成了首先,熟悉相关技术的人才的缺少也就成了发展的障碍,而相关资料的奇缺成了技术人员成长的障碍。

本书注重实用,着眼入门,用大量实际的例子力求带领您进入 flex,red5 结合开发的天堂。没一个例子都是精心挑选,每一个例子都代表着一个知识点,每一个例子都将成为您以后项目的一部分。当然这是入门级别的书籍,不可能把所有的相关知识都讲到,java,flex,spring,mina 等等等等都 red5 相关,如果要对每一个点讲解那将是一个系列,而不是一本书。为了突出重点,本书只讲解把这些联系起来的纽扣 red5,并讲解如何利用 red5 开发项目。

内容包括 red5 的安装,red5 开发中各种概念,各种问题的解决方案,开发工具的使用等等,但有详有略,如工具的使用只讲解了书本中需要用到的相关知识。最后书本还有数个完整的例子,也是实际开发中常用的例子,如聊天程序,播放视频程序。

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

目录:

- 一. Red5 介绍及安装:
 - Red5 介绍
 - 安装 red5
- 二. 编写运行第一个程序
 - 编写服务器端
 - 编写客户端
 - 运行自己编写的程序
- 三. 远程调用
 - 客户端调用服务器端方法
 - Red5 调用客户端方法
 - 遍历所有连接到服务器端的用户
 - 好友上线通知
- 四. sharedObject 共享对象
 - sharedObject 介绍
 - 游戏中移动同步的原理和简单实现¹
- 五. 服务器执行计划
- 六. 录制播放视频
 - 播放服务器端视频
 - 录制视频
 - 设置视频保存位置
- 七. 利用 scope, room 管理客户端
- 八. 视频播放程序
- 九. 简单聊天程序
- 附录: 开发工具的使用

作者: 周英科 qq: 543246541

email: li_li_li87222@163.com

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

一. Red5 介绍及安装

1. Red5 介绍

Red5是一个采用Java 开发开源免费的Flash 流媒体服务器,功能和 Adobe 的 fms (Flash Media Server) 相同,可以上官方网站 <http://www.osflash.org/red5>免费下载并使用。它支持把音频 (MP3) 和视频 (FLV) 转换成播放流,在下载的同时播放流; 录制客户端播放流 (只支持 FLV), 把摄像头, 麦克风等传入的视频音频录制保存到服务器; 共享对象; 现场直播流发布; 远程调用。Red5使用 RSTP 作为流媒体传输协议。

Red5的功能使其在许多领域得到应用。如视频播放网站, 远程教育, 视频会议程序, 聊天程序, 网络游戏等等。

免费开源使软件更加容易扩展, 下载后你可以对源代码进行修改; 更加经济, 比起 fms 高昂的费用 red5 能为一般的应用节约大笔费用; 服务器端的 java 语言比起 fms 服务器端的 actionscript2 语言更加面向对象更加成熟。这使得 red5 还没到版本 1 就已经红遍世界。

2. red5 安装

第一步 上 java.sun.com 下载 jdk6 并安装。如果不是安装在默认路径请设置 classpath 和 path。设置 classpath 和 path 的方法请参考其他 java 的资料。

第二步 上 <http://www.osflash.org/red5> 下载 Red5 v0.7.0 Final 的 window 安装版。点击下载到 exe 文件即开始安装, 安装过程中选择要把 red5 安装到的目录。安装结束后打开浏览器输入 <http://localhost:5080/> 如果能打开显示“Red5 Test Page”表示安装成功。安装成功后可以运行查看 red5 自带的例子。

第三步 red5 的启动和关闭。安装版的 red5 默认情况下会在计算机启动时启动, 如果想关闭自启动可以右击我的电脑->管理->服务和程序, 找到 red5 的那项把启动方式的自动启动改为手动或其他。更改了启动方式后可以找到 red5 的安装目录下的 wrapper 目录下的 Red5.bat 文件, 双击即可启动, 这样可以从窗口看到客户端连接的情况。关闭窗口即可关闭 red5, 也可以双击 StopRed5-NT.bat 关闭服务程序。

二. 编写运行第一个程序

第一个程序的运行是学习的莫大回报, 也是继续下面学习的关键, 只有顺利的运行了第一个程序才会更加信心百倍的去学习更深一步的知识。

Red5+flex 或 red5+flash 的项目一般都包括服务器和客户端两个部分, 服务器用 java, 客户端用 flex 或 flash。编写并编译完服务器端代码后需要把服务器的应用部署到 red5 的 webapps 目录下这和传统的 java 应用没区别, 客户端可以直接点击 flash 文件运行。

下面开始第一个程序的编写, 它分成服务器和客户端两部分, 以后的程序也都安这种方式分。

1. 编写服务器端

打开 eclipse 新建 java 工程或 javaee 工程 chapter2 (如果不熟悉 eclipse 请参考附件 eclipse 的使用)。把 red5 安装目录下的 red5.jar 以及 lib 目录下的 jar 都添加到工程的类库中。

新建包 first, 然后新建类, 类名称为 Application 并继承 ApplicationAdapter。类的代码如下:

```
package first;
import org.red5.server.adapter.ApplicationAdapter;
import org.red5.server.api.IConnection;
public class Application extends ApplicationAdapter {
    public boolean appConnect(IConnection conn, Object[] args) {
```

```
System.out.println(" 连接");
return true;
}
public String change(String str){
    System.out.println(" 客户端调用服务器");
    return str.toUpperCase();// 传入的字符串转换成大写
}
}
```

2.编写客户端

打开 flexbuilder3 新建 flex 工程(具体 flexbuilder 的使用请参考附录),名字取名为 chapter2。

打开项目中 chapter2.mxml。chapter2.mxml 的代码如下:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
fontSize="12" backgroundGradientAlphas="[1.0, 1.0]"
backgroundGradientColors="#FDF9F9, #FDF9F9" width="442" height="344">
    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
            private var rtmpURL:String="rtmp://localhost/chapter2";
            private var conn:NetConnection=new NetConnection();
            private var isConnectSuccess:Boolean=false;
            private var responder:Responder=new Responder(resultFun);
            private function resultFun(object:String):void{
                trace(object);
                result.text=object.toString();
            }
            private function clickConnect(e:MouseEvent):void{
                conn.addEventListener(NetStatusEvent.NET_STATUS,netStatus);
                conn.connect(rtmpURL);
            }
            private function click(e:MouseEvent):void{
                invoke();
            }
            private function invoke():void{
                if(isConnectSuccess){
                    conn.call("change",responder,str.text);//change 是服务器端方法名称
                }else{
                    Alert.show("还没连接到服务器");
                }
            }
            private function netStatus(e:NetStatusEvent):void{
                trace(e.info.code);
                if(e.info.code=="NetConnection.Connect.Success"){
                    isConnectSuccess=true;
                }
            }
        ]]>
    </mx:Script>
</mx:Application>
```

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

```

    }
  }
]]>
</mx:Script>
<mx:Button x="224" y="175" label="调用服务器方法" click="this.click(event)"/>
<mx:TextInput x="129" y="145" id="str" width="212"/>
<mx:Label x="129" y="119" text="显示从服务器端返回的字符" id="result" width="160"
fontSize="12"/>
<mx:Button x="129" y="175" label="连接服务器" click="this.clickConnect(event)"/>
</mx:Application>

```

3. 部署运行程序

进入 red5 的安装目录下的 webapps 下, 复制 test 文件夹, 把复制的文件夹改名为 chapter2, 删除 chapter2\WEB-INF 目录下的文件夹。

把服务器项目下的 classes 目录下的文件夹 first 拷贝到 chapter2\WEB-INF 下。

把 red5-web.properties 中的 /test 替换为 /chapter2。

把 red5-web.xml 中的 `<bean id="test.service" class="org.red5.server.webapp.test.TestService" singleton="true" lazy-init="default" autowire="default" dependency-check="default" /><bean id="echo.service" class="org.red5.samples.services.EchoService" singleton="true" lazy-init="default" autowire="default" dependency-check="default" />` 删除。

把 org.red5.server.webapp.test.Application 替换为 first.Application。

把 web.xml 中的 `<param-value>test</param-value>` 替换为 `<param-value>/chapter2</param-value>`

把 `<taglib uri="/tds/c" taglib-location="/WEB-INF/tds/c.tld" />` 删除

注意: webapps 下的文件夹名 chapter2, red5-web.properties 中的 chapter2, web.xml 中的 chapter2 要保持相同, 如果一个改了, 其他的也改成相同的, 所有以后的工程都这样。

重新启动 red5

在 flexbuilder3 中选择运行菜单中的运行项运行 chapter2。在文本框里输入字符串, 点击连接按钮, 然后点击调用方法按钮。上面如果显示出输入框里的字符串的大写表示运行正常。

服务器端类有两个方法, appConnect 类从 ApplicationAdapter 继承来, 当 flash 连接时触发, 可以返回 true 或 false。True 表示接受客户端的请求, false 表示拒绝客户端的请求。如果拒绝那么以后 flash 和服务器的交互服务器都不会理会, 所以当想拒绝某些 ip 的请求时只需有在用户连接时检查它的 ip 是否在拒绝 ip 之内, 如果在就返回 false, 那样就可以拒之门外了。change 是自己定义的, 客户可以直接调用的一个远程方法。作用是把客户传入的参数变成大写后返回, 此处目的在于运行一个 red5 和 flex 交互的过程, 并没有检查传入的参数是否有大写或都已经是大写。

客户端的代码稍多些, 位于 `<mx:Script><![CDATA[]]></mx:Script>` 之外的是图形界面的代码, 包括两个按钮, 一个输入框, 一个显示文本, 这就不详细讲解了, 注意给按钮添加事件监听器。过程是这样的, 用户点击连接, 按钮的监听器 clickConnect 响应, conn.connect(rturl) 连接到服务器, 在服务器和客户端建立一条信息交流的通道。当这条通道打通时 netStatus 执行, 把是否是打通的开关 isConnectedSuccess 设置为通的。当用户再点击调用按钮时, 通过这条通道调用服务器端的方法 conn.call("change", responder, str.text)。传给服务器字符串 str.text, 当服务器端方法执行完后把返回值给 responder 定义时的方法 resultFun, 这样一个客户端与服务器交互的过程完成。

三. 远程调用

通过 red5 服务器端和客户端可以直接通过方法调用来交互, 这是非常方便的 (想想你在一

个类里面方法调用发方便性吧), 服务器调可以用客户端的方法, 客户端也可以调用服务器端的方法。其实上面的例子就是通过方法调用实现的。

1.客户端调用服务器端方法

通过 red5 可以从客户端直接调用服务器上 java 编写的方法。调用时指明被调用的方法名称, 如果有返回值则需要指明返回值处理的 responder, 如果被调用的方法有参数则需要传入参数。可以引用第一章的例子, `conn.call("change", responder, str.text);` 表示客户端调用服务器端的名称为 change 的方法, 查看 java 中 change 的定义 `public String change(String str){//函数体}` 发现参数是 java 中 String 类型的; 调用时传入的 str.text 是 actionscript 中 String 类型; 假如 change 有多个入口参数, 则可以把参数列在 str.text 后面, 并用"," 隔开。返回值由 responder 处理, change 返回值是 String 类型的, 要求 responder 定义时的函数 resultFun 的入口参数是 String 类型。

调用过程中需要注意的是参数类型的对应。具体可以看下列表

Java	actionsript
null	null
int/float	Number
double	Double
boolean	Boolean
String	String
List	Array
Map	Object
Bean	Object

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

表说明假如 java 传给 flash 的参数是自定义 Bean 类型时, 客户端接收到的是 Object 类型

当被调用的方法没有返回值时把 responder 设置为 null 。具体怎么实现就不再赘述, 请看第二章的例子。

2.服务器调用客户端方法

不带返回的调用

新建 eclipse 工程 chapter3, 并建立包 first, 在包里新建一个类 Application 并继承 ApplicationAdapter。Application 到代码如下:

```
package first;
import org.red5.server.adapter.ApplicationAdapter;
import org.red5.server.api.IConnection;
import org.red5.server.api.service.IServiceCapableConnection;
public class Application extends ApplicationAdapter {
    @Override
    public boolean appConnect(IConnection arg0, Object[] arg1) {
        // TODO Auto-generated method stub
        callClient(arg0);
        return true;
    }
    private void callClient(IConnection conn){
        if (conn instanceof IServiceCapableConnection) {
            IServiceCapableConnection sc = (IServiceCapableConnection) conn;
            sc.invoke("clientMethod", new Object[]{"One", 1});
        }
    }
}
```

```
    }  
  }  
}
```

复制 webapps 下的 chapter2 文件夹，并改名为 chapter3，把工程下 class 目录下的 first 拷贝覆盖以前的 first，把 red5-web.properties 中的 /chapter2，替换为 /chapter3。把 web.xml 中的 <param-value>/chapter2</param-value> 替换为 <param-value>/chapter3</param-value> 重启 red5。

新建 chapter3 客户端项目，chapter.mxml 代码如下：

```
<?xml version="1.0" encoding="utf-8"?>  
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"  
fontSize="12" backgroundGradientAlphas="[1.0, 1.0]"  
backgroundGradientColors="#FDF9F9, #FDF9F9" width="442" height="344">  
  <mx:Script>  
    <![CDATA[  
      import mx.controls.Alert;  
      private var rtmpURL:String="rtmp://localhost/jiaocheng";  
      private var conn:NetConnection=new NetConnection();  
      private var client1:Object=new Object();  
      private function clickConnect(e:MouseEvent):void{  
        client1.clientMethod=this.clientMethod;  
        conn.client=this.client1;  
        conn.addEventListener(NetStatusEvent.NET_STATUS,netStatus);  
        conn.connect(rtmpURL);  
      }  
      private function netStatus(e:NetStatusEvent):void{  
      }  
      private function clientMethod(str:String,num:Number):void{  
        Alert.show("接收"+str+(num+1));  
      }  
    ]]>  
  </mx:Script>  
  <mx:Button x="129" y="175" label="连接服务器" click="this.clickConnect(event)"/>  
</mx:Application>
```

作者：周英科 qq: 543246541
email: li_li_li87222@163.com

点击调试程序运行客户端，然后点击连接按钮可以发现输出“接收 One2”表示服务器调用客户端到方法成功。

服务器端还是只有俩个方法，当客户端连接时服务器调用 appConnect，在 appConnect 里调用 callClient。在 callClient 中服务器调用客户端的方法 clientMethod，并传给客户端两个参数，一个字符类型，一个数字类型。只有 IServiceCapableConnection 类型的连接才可以调用客户端方法，所以要先对连接类型进行强制转换。

再看客户端，有三个方法，因为不需要处理连接状态的变化事件所以用个空方法以便简单突出重点。用户点击按钮连接，调用 clickConnect 方法连接服务器。还有一个方法给服务器调用的，这个方法附着在 conn.client 上。因为 ActionScript3 中 NetConnection 是封闭的，无法直接把方法附着在其上，但他的一个属性 client 是 Object 类型的，是动态的。

服务器不但可以调用客户端的方法，还可以把客户端方法处理的值返回给服务器。

理服务器调用客户端方法有返回值就必须实现 IPendingServiceCallback 接口，并调用方式改为 `sc.invoke("clientMethod", new Object[]{"One", 1},this);` this 是任意一个实现 IPendingServiceCallback 接口的类，在接口的 `resultReceived` 方法中处理返回值。现在把上面的类改成下面样子

```
package first;
import org.red5.server.adapter.ApplicationAdapter;
import org.red5.server.api.IConnection;
import org.red5.server.api.service.IPendingServiceCall;
import org.red5.server.api.service.IPendingServiceCallback;
import org.red5.server.api.service.IServiceCapableConnection;
public class Application extends ApplicationAdapter implements IPendingServiceCallback{
    @Override
    public boolean appConnect(IConnection arg0, Object[] arg1) {
        callClient(arg0);
        return true;
    }
    private void callClient(IConnection conn){
        if (conn instanceof IServiceCapableConnection) {
            IServiceCapableConnection sc = (IServiceCapableConnection) conn;
            sc.invoke("clientMethod", new Object[]{"One", 1},this);
        }
    }
    public void resultReceived(IPendingServiceCall arg0) {
        System.out.println("来自客户端到返回: "+arg0.getResult());
    }
}
```

把客户端的

```
private function clientMethod(str:String,num:Number):void{
    Alert.show("接收"+str+(num+1));
}
```

改成

```
private function clientMethod(str:String,num:Number):String{
    Alert.show("接收"+str+(num+1));
    Return "客户端返回来的字符串";
}
```

测试可以从 red5 的框里看到输出 “客户端返回来的字符串”

3.遍历所有连接到服务器端的用户

遍历所有链接到服务器的客户端是常常的事，如好友上线，则要及时把这个消息通知给在线的人员。

Red5 通过 `IConnection.getscope()`得到所在 `scope`，通过 `scope` 可以得到连接到这个 `scope` 的所有客户的连接。在得到连接后就可以通过连接调用客户端方法。如调用每个客户端的 `clientMethod` 方法一遍可以把服务器改成下面的样子

```
package chapter2;
import java.util.Iterator;
```

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com


```
import org.red5.server.adapter.ApplicationAdapter;
import org.red5.server.api.IConnection;
import org.red5.server.api.IScope;
import org.red5.server.api.service.IPendingServiceCall;
import org.red5.server.api.service.IPendingServiceCallback;
import org.red5.server.api.service.IServiceCapableConnection;
public class Application extends ApplicationAdapter implements IPendingServiceCallback{
    @Override
    public boolean appConnect(IConnection arg0, Object[] arg1) {
        this.callEvery(arg0.getScope());
        return true;
    }
    private void callEvery(IScope scope){
        Iterator<IConnection> it=scope.getConnections();
        while(it.hasNext()){
            this.callClient(it.next());
        }
    }
    private void callClient(IConnection conn){
        if (conn instanceof IServiceCapableConnection) {
            IServiceCapableConnection sc = (IServiceCapableConnection) conn;
            sc.invoke("clientMethod", new Object[]{conn.getSessionId(), 1},this);
        }
    }
    public void resultReceived(IPendingServiceCall arg0) {
        System.out.println("来自客户端到返回: "+arg0.getResult());
    }
}
```

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

把服务器端的 class 覆盖以前的文件，运行客户端可以看到有一个新的客户连接时每个在线的客户端会弹出新上线用户的用户 id

四. shareObject共享对象

共享对象

游戏中同步的原理

同步的简单实现

共享对象分为本地和远程两种，本地的可以看 adobe 官方文档，很详细，这里就讲解远程的。远程共享对象是供指定客户端群共有的数据。就如你家的钱包是你和你爱人共有的一样（某些特殊除外），可随便花，一个人少了，代表另一个也少了，但是也有限制，不是每个人都可以花（我就不能花你家钱包的钱只有指定有权限的人才可以。要花钱先得获得钱包，然后还要打开钱包。共享对象为数据共享提供了很大的方便，因为数据改变时会在每个客户端同步而不用人去干预太多。

下面以一个移动同步的例子讲解，效果是移动方块，所有其他浏览器中的的方块都同时改变位置。新建服务器项目 chapter4，并新建类，代码如下

```
package first;
import org.red5.server.adapter.ApplicationAdapter;
```

```
import org.red5.server.api.IConnection;

public class Application extends ApplicationAdapter {
    @Override
    public boolean appConnect(IConnection arg0, Object[] arg1) {
        return true;
    }
}
```

修改把 red5-web.xml, web.xml, red5-web.properties 中的 chapter3 替换为 chapter4 重启 red5.
新建 flex 应用 代码如下

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
    <mx:Canvas x="279" y="99" width="105" height="73" id="rect"
background-color="# DA4040"
        mouseDown="this.mouseDown(event)" mouseUp="this.mouseUp(event)"
mouseMove="this.mouseMove(event)">
    </mx:Canvas>
    <mx:Script>
    <![CDATA[
        import mx.controls.Alert;
        private var rtmpURL:String="rtmp://localhost/jiaocheng";
        private var conn:NetConnection=new NetConnection();
        private var isConnectSuccess:Boolean=false;
        private var so:SharedObject;
        private var isDown:Boolean=false;
        private function clickConnect(e:MouseEvent):void{
            if(!isConnectSuccess){
                conn.addEventListener(NetStatusEvent.NET_STATUS,netStatus);
                conn.connect(rtmpURL);
            }
        }
        private function mouseDown(e:MouseEvent):void{
            rect.startDrag();
            isDown=true;
        }
        private function mouseUp(e:MouseEvent):void{
            rect.stopDrag();
            isDown=false;
        }
        private function mouseMove(e:MouseEvent):void{
            if(isDown){
                so.setProperty("x",rect.x);
                so.setProperty("y",rect.y);
            }
        }
    ]]>
    </mx:Script>
</Application>
</mx:Application>
```

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

```

    }
    private function netStatus(e:NetStatusEvent):void{
        trace(e.info.code);
        if(e.info.code=="NetConnection.Connect.Success"){
            isConnectSuccess=true;
        }
        so=SharedObject.getRemote("point",conn.uri,true);
        so.connect(conn);
        so.addEventListener(SyncEvent.SYNC,syncro);
    }
    private function syncro(e:SyncEvent):void{
        rect.x=so.data.x;
        rect.y=so.data.y;
    }
    ]]>
</mx:Script>
    <mx:Button x="10" y="10" label="Button" click="clickConnect(event)"/>
</mx:Application>

```

运行俩个 flash 客户端，在一个里面拖动框，看另一个框是否同步改变位置，如果是则正常。

shareObject 应用时客户端要先用 SharedObject.getRemote("point",conn.uri,true)获取服务器端的一个共享对象；然后用 so.connect(conn)连接对象来同步；最后通过 so.addEventListener(SyncEvent.SYNC,syncro)可以响应共享对象的变化。服务器只响应客户端的连接，没有任何其他的代码。如果要在服务器端处理共享对象则通过服务器端的 ISharedObject 接口实现。为了演示服务器端对共享对象的控制把服务器端的代码改成如下

```

package chapter4;
import org.red5.server.adapter.ApplicationAdapter;
import org.red5.server.api.IConnection;
import org.red5.server.api.IScope;
import org.red5.server.api.so.ISharedObject;
public class Application extends ApplicationAdapter {
    @Override
    public boolean appConnect(IConnection arg0, Object[] arg1) {
        return true;
    }
    @Override
    public boolean appStart(IScope arg0) {
        System.out.println("appStart 程序启动");
        this.createSharedObject(arg0, "point", true);
        ISharedObject so=this.getSharedObject(arg0, "point");
        if(so!=null)
            so.addSharedObjectListener(new ShareObjectListener());
        else
            System.out.println("point 是 null");
        return true;
    }
}

```

作者: 周英科 qq: 543246541

email: li_li_li87222@163.com

```
    }  
}  
并添加类  
package chapter4;  
import java.util.List;  
import java.util.Map;  
import org.red5.server.api.IAttributeStore;  
import org.red5.server.api.so.ISharedObjectBase;  
import org.red5.server.api.so.ISharedObjectListener;  
public class ShareObjectListener implements ISharedObjectListener {  
    public void onSharedObjectClear(ISharedObjectBase arg0) {  
        System.out.println("onSharedObjectClear");  
    }  
    public void onSharedObjectConnect(ISharedObjectBase arg0) {  
        System.out.println("onSharedObjectConnect");  
    }  
    public void onSharedObjectDelete(ISharedObjectBase arg0, String arg1) {  
        System.out.println("onSharedObjectDelete");  
    }  
    public void onSharedObjectDisconnect(ISharedObjectBase arg0) {  
        System.out.println("onSharedObjectDisconnect");  
    }  
    public void onSharedObjectSend(ISharedObjectBase arg0, String arg1,  
        List arg2) {  
        System.out.println("onSharedObjectSend");  
    }  
    public void onSharedObjectUpdate(ISharedObjectBase arg0,  
        IAttributeStore arg1) {  
        System.out.println("onSharedObjectUpdate");  
    }  
    public void onSharedObjectUpdate(ISharedObjectBase arg0,  
        Map<String, Object> arg1) {  
        System.out.println(" onSharedObjectUpdate");  
    }  
    public void onSharedObjectUpdate(ISharedObjectBase arg0, String key,  
        Object value) {  
        System.out.println("更新共享对象的值    "+key+": "+value);  
    }  
}
```

把编译好的文件放到 red5 的应用下，重启 red5。再允许客户端则每次移动都可以看到移动到的位置

五. 服务器自动执行计划

可以做服务器端添加脚本来定时执行一些语句，如调用客户端的某个函数或者清空某些共享对象。

新建服务器包 chapter5，新建类 Application，代码如下：

```
package chapter5;
```

```
import java.util.Iterator;
import org.red5.server.adapter.ApplicationAdapter;
import org.red5.server.api.IConnection;
import org.red5.server.api.IScope;
import org.red5.server.api.scheduling.IScheduledJob;
import org.red5.server.api.scheduling.ISchedulingService;
import org.red5.server.api.service.IServiceCapableConnection;
public class Application extends ApplicationAdapter implements IScheduledJob{
    String jobID;
    @Override
    public boolean appConnect(IConnection arg0, Object[] arg1) {
        return true;
    }
    @Override
    public boolean appStart(IScope arg0) {
        jobID=this.addScheduledJob(60000, this);
        return true;
    }
    private void callEvery(IScope scope){
        Iterator<IConnection> it=scope.getConnections();
        while(it.hasNext()){
            this.callClient(it.next());
        }
    }
    private void callClient(IConnection conn){
        if (conn instanceof IServiceCapableConnection) {
            IServiceCapableConnection sc = (IServiceCapableConnection) conn;
            sc.invoke("clientMethod", new Object[]{conn.getSessionId(), 1});
        }
    }
    public void execute(ISchedulingService arg0)
        throws CloneNotSupportedException {
        System.out.println("运行计划");
        this.callEvery(this.getScope());
    }
}
```

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

把编译后的文件放到 red5下。重启 red5。

新建 flex 应用，代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
    <mx:Script>
    <![CDATA[
        import mx.controls.Alert;
        private var rtmpURL:String="rtmp://localhost/jiaocheng";
```

```
private var conn:NetConnection=new NetConnection();
private var isConnectSuccess:Boolean=false;
private var so:SharedObject;
private var client1:Object=new Object();
private function clickConnect(e:MouseEvent):void{
    client1.clientMethod=this.clientMethod;
    conn.client=this.client1;
    if(!isConnectSuccess){
        conn.addEventListener(NetStatusEvent.NET_STATUS,netStatus);
        conn.connect(rtmpURL);
    }
}
private function netStatus(e:NetStatusEvent):void{
    trace(e.info.code);
    if(e.info.code=="NetConnection.Connect.Success"){
        isConnectSuccess=true;
    }
}
private function clientMethod(str:String,num:Number):void{
    Alert.show("您还未注册");
}
]]>
</mx:Script>
<mx:Button x="110" y="125" label="Button" click="this.clickConnect(event)"/>
```

</mx:Application>

运行客户端，点击连接，则每隔一段时间会弹出窗口提示没有注册。

六. 录制播放视频

1. 播放服务器端视频

新建服务器包 chapter6,并新建类 Application，代码如下：

```
package chapter6;
import org.red5.server.adapter.ApplicationAdapter;
import org.red5.server.api.IConnection;
public class Application extends ApplicationAdapter {
    public boolean appConnect(IConnection arg0, Object[] arg1) {
        return true;
    }
}
```

把编译后的文件放到 red5 下，重启 red5。

编写客户端，新建 flex 应用，代码如下

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
<mx:UIComponent id="ui">
</mx:UIComponent>
<mx:Script>
```



```
<![CDATA[
    import mx.controls.Alert;
    private var rtmpURL:String="rtmp://localhost/jiaocheng";
    private var conn:NetConnection=new NetConnection();
    private var isConnectSuccess:Boolean=false;
    private var netStream:NetStream;
    private var video:Video=new Video();

    private function clickConnect(e:MouseEvent):void{
        if(!isConnectSuccess){
            conn.addEventListener(NetStatusEvent.NET_STATUS,netStatus);
            conn.connect(rtmpURL);
        }
    }
    private function netStatus(e:NetStatusEvent):void{
        trace(e.info.code);
        if(e.info.code=="NetConnection.Connect.Success"){
            isConnectSuccess=true;
            netStream=new NetStream(conn)
            netStream.client=new StreamClient();
            video.attachNetStream(netStream);
            netStream.play("a.flv");
            ui.addChild(video);
        }
    }
}]>
</mx:Script>
    <mx:Button x="378" y="369" label="Button" click="this.clickConnect(event)"/>
</mx:Application>
```

另外新建类 StreamClient 代码如下

```
package
{
    public class StreamClient
    {
        public function StreamClient()
        {
        }
        public function onMetaData(info:Object):void{
            for(var n:* in info){
                trace(n+": "+info[n]);
            }
        }
        public function onPlayStatus(info:Object):void{
            for(var n:* in info){
```

作者: 周英科 qq: 543246541

email: li_li_li87222@163.com

```
        trace(n+": "+info[n]);  
    }  
}  
}
```

把一个 a.flv 的视频文件放到\webapps\jiaocheng\streams 下，运行客户端既可以看到播放视频

2. 录制视频

新建客户端 flex 应用,代码如下:

```
<?xml version="1.0" encoding="utf-8"?>  
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">  
<mx:UIComponent id="ui">  
</mx:UIComponent>  
    <mx:Script>  
    <![CDATA[  
        import mx.controls.Alert;  
        private var rtmpURL:String="rtmp://localhost/jiaocheng";  
        private var conn:NetConnection=new NetConnection();  
        private var isConnectSuccess:Boolean=false;  
        private var netStream:NetStream;  
        private var video:Video=new Video();  
  
        private function clickConnect(e:MouseEvent):void{  
            if(!isConnectSuccess){  
                conn.addEventListener(NetStatusEvent.NET_STATUS,netStatus);  
                conn.connect(rtmpURL);  
            }  
        }  
        private function clickVideo(e:MouseEvent):void{  
            video.attachCamera(Camera.getCamera());  
            ui.addChild(video);  
        }  
        private function netStatus(e:NetStatusEvent):void{  
            trace(e.info.code);  
            if(e.info.code=="NetConnection.Connect.Success"){  
                isConnectSuccess=true;  
                netStream=new NetStream(conn)  
                netStream.client=new StreamClient();  
                netStream.attachAudio(Microphone.getMicrophone());  
                netStream.attachCamera(Camera.getCamera());  
                netStream.publish("b.flv","record");  
            }  
        }  
    ]]>
```

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com


```
</mx:Script>
<mx:Button x="378" y="369" label="录制" click="this.clickConnect(event)"
fontSize="16"/>
<mx:Button x="310" y="369" label="视频" fontSize="16"
click="this.clickVideo(event)"/>
</mx:Application>
```

运行客户端，点击视频可以看到自己的视频，点击录制可以把视频保存到服务器上。如果没有视频只有麦克风则点击视频看不到效果，但还是可以录制视频，只是是声音而已。录制的视频保存在\webapps\jiaocheng\streams下。

3.设置视频保存位置

录制的视频往往比较多，与red5放同一个目录有点不太合适，那么该怎么把录制的视频，播放的视频目录放到其他的文件夹或分区呢？Red5要求实现 `IStreamFilenameGenerator` 接口，并在接口里面设置位置。

新建服务器类，代码如下：

```
package chapter6;
import org.red5.server.api.IScope;
import org.red5.server.api.stream.IStreamFilenameGenerator;
public class PathBean implements IStreamFilenameGenerator {
    public String recordPath = "streams/";
    public String playbackPath ="streams/";
    public String generateFilename(IScope scope, String name, GenerationType type) {
        return this.generateFilename(scope, name,null, type);
    }
    public String generateFilename(IScope scope, String name,
        String extension, GenerationType type) {
        String filename;
        if (type == GenerationType.RECORD)
            filename = recordPath + name;
        else
            filename = playbackPath + name;
        if (extension != null)
            filename += extension;
        return filename;
    }
    public String getRecordPath() {
        return recordPath;
    }
    public void setRecordPath(String recordPath) {
        this.recordPath = recordPath;
    }
    public String getPlaybackPath() {
        return playbackPath;
    }
    public void setPlaybackPath(String playbackPath) {
```

作者：周英科 qq: 543246541
email: li_li_li87222@163.com

```
        this.playbackPath = playbackPath;
    }
    public boolean resolvesToAbsolutePath() {
        // TODO Auto-generated method stub
        return true;
    }
}
```

在 red5-web.xml 中添加下面 bean

```
<bean id="streamFilenameGenerator"
      class="chapter6.PathBean">
    <property name="recordPath" value="E:\streams\" />
    <property name="playbackPath" value="E:\streams\" />
</bean>
```

其中 E:\streams\ 可以改成你计算机中的任意文件夹。把编译后的文件放到 red5 下，运行 red5 重启，运行客户端应用 Record。点击录制，即可在指定的文件目录下看到录制的视频。同样也可以播放指定目录下的视频文件。

七. Scope 与 room 讲解

Red5 有范围之分，就如 jsp 中的范围概念。Red5 中的范围分成 application, room, place, lobby。但是他们都是一样的东西 scope。scope 嵌套，最上的称为 Application，也是默认连接到应用时的范围，连接到 Application 的 url 是 rtmp://localhost/appname/。Application 里面的子范围称为 room 连接的 url 是 rtmp://localhost/appname/roomnum/。当然要连接到子范围必然会先连接到父范围。在每个范围中保存着自己的连接的信息，sharedobject 等。下面以一个例子说明 scope 的应用。这个例子很简单，客户端连接到各种范围，服务器在客户端连接时把自己范围内的客户端 id 打印出来。

服务器端代码：

```
package chapter7;
import java.util.Set;
import org.red5.server.adapter.ApplicationAdapter;
import org.red5.server.api.IClient;
import org.red5.server.api.IConnection;
import org.red5.server.api.IScope;
public class Application extends ApplicationAdapter {
    @Override
    public boolean appStart(IScope arg0) {
        System.out.println(" 启动 appStart");
        return true;
    }
    @Override
    public boolean appConnect(IConnection arg2, Object[] arg1) {
        IScope arg0=arg2.getScope();
        System.out.println(" 连接到 "+arg0.getName()+"ID 列表: ");
        Set<IClient> i=arg0.getClients();
        for(IClient c:i){
            System.out.println(c.getId());
        }
    }
}
```

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

```
        return true;
    }
    @Override
    public boolean roomStart(IScope arg0) {
        System.out.println("启动 roomStart");
        return true;
    }
    @Override
    public boolean roomConnect(IConnection arg2, Object[] arg1) {
        IScope arg0=arg2.getScope();
        System.out.println(" 连接到  "+arg0.getName()+"ID 列表:  ");
        Set<IClient> i=arg0.getClients();
        for(IClient c:i){
            System.out.println(c.getId());
        }
        return true;
    }
}
```

作者: 周英科 qq: 543246541

email: li_li_li87222@163.com

客户端代码:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
fontSize="16">
    <mx:Script>
    <![CDATA[
        import mx.controls.Alert;
        private var rtmpURL:String="rtmp://localhost/jiaocheng/";
        private var isConnectSuccess:Boolean=false;

        private function clickConnect(e:MouseEvent):void{
            Alert.show("链接到: "+e.target.label);
            rtmpURL=e.target.label;
            var conn:NetConnection=new NetConnection();
            conn.addEventListener(NetStatusEvent.NET_STATUS,netStatus);
            conn.connect(rtmpURL);
        }
        private function netStatus(e:NetStatusEvent):void{
            Alert.show(e.info.code);
            if(e.info.code=="NetConnection.Connect.Success"){
                isConnectSuccess=true;
            }
        }
    ]]>
</mx:Script>
```

```
<mx:Button x="25" y="10" label="rtmp://localhost/jiaocheng/"  
click="this.clickConnect(event)"/>  
<mx:Button x="25" y="44" label="rtmp://localhost/jiaocheng/room/"  
click="this.clickConnect(event)"/>  
<mx:Button x="25" y="84" label="rtmp://localhost/jiaocheng/room1/"  
click="this.clickConnect(event)"/>  
<mx:Button x="25" y="120" label="rtmp://localhost/jiaocheng/room/place"  
click="this.clickConnect(event)"/>  
</mx:Application>
```

打开三个客户端，点击各个按钮看服务器端的输出。

八.视频播放

本章将利用以前的知识做一个在线音乐视频播放的程序。当连接到服务器后会把服务器端的视频列表显示出来，当用户点击后播放。尽管功能简陋，但稍加改进就可以成为产品。

程序分服务器端和客户端。

服务器端在第七章的基础上改进，新建类 `StreamService`。代码如下

```
package chapter7;
```

```
import java.io.File;  
import java.io.IOException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Locale;  
import java.util.Map;  
  
import org.apache.commons.logging.Log;  
import org.apache.commons.logging.LogFactory;  
import org.red5.server.api.IScope;  
import org.red5.server.api.Red5;  
import org.springframework.core.io.Resource;
```

```
public class StreamService {  
  
    /**  
     * Getter for property 'listOfAvailableFLVs'.  
     *  
     * @return Value for property 'listOfAvailableFLVs'.  
     */  
    public Map getListOfAvailableFLVs() {  
        IScope scope = Red5.getConnectionLocal().getScope();  
        Map<String, Map> filesMap = new HashMap<String, Map>();  
        Map<String, Object> fileInfo;  
        try {  
            Resource[] flvs = scope.getResources("streams/*.flv");
```

作者：周英科 qq: 543246541
email: li_li_li87222@163.com

```
if (flvs != null) {
    for (Resource flv : flvs) {
        File file = flv.getFile();
        Date lastModifiedDate = new Date(file.lastModified());
        String lastModified = formatDate(lastModifiedDate);
        String flvName = flv.getFile().getName();
        String flvBytes = Long.toString(file.length());
        fileInfo = new HashMap<String, Object>();
        fileInfo.put("name", flvName);
        fileInfo.put("lastModified", lastModified);
        fileInfo.put("size", flvBytes);
        filesMap.put(flvName, fileInfo);
    }
}

Resource[] mp3s = scope.getResources("streams/*.mp3");
if (mp3s != null) {
    for (Resource mp3 : mp3s) {
        File file = mp3.getFile();
        Date lastModifiedDate = new Date(file.lastModified());
        String lastModified = formatDate(lastModifiedDate);
        String flvName = mp3.getFile().getName();
        String flvBytes = Long.toString(file.length());
        fileInfo = new HashMap<String, Object>();
        fileInfo.put("name", flvName);
        fileInfo.put("lastModified", lastModified);
        fileInfo.put("size", flvBytes);
        filesMap.put(flvName, fileInfo);
    }
}
} catch (IOException e) {
    e.printStackTrace();
}
return filesMap;
}

private String formatDate(Date date) {
    SimpleDateFormat formatter;
    String pattern = "dd/MM/yy H:mm:ss";
    Locale locale = new Locale("en", "US");
    formatter = new SimpleDateFormat(pattern, locale);
    return formatter.format(date);
}
}
```

作者: 周英科 qq: 543246541
email: li_li_li87222@163.com

把编译后的文件拷贝到 red5下；修改 red5-web.xml，添加 bean 配置

```
<bean id="flv.service"  
      class="chapter7.StreamService"  
      singleton="true" />
```

删除 id 是 streamFileNameGenerator 的 bean 配置。在 streams 下放几个 mp3 和 flv 格式的文件。
重启 red5。

新建 Flex Application 代码如下：

```
<?xml version="1.0" encoding="utf-8"?>  
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"  
creationComplete="init()">  
  <mx:Script>  
    <![CDATA[  
  
      private var rtmpURL:String="rtmp://localhost/jiaocheng";  
      private var nc:NetConnection=new NetConnection();  
      private var playerVideo:Video=new Video();  
      private var responder:Responder = new Responder(getMediaList);  
  
      [Bindable]  
      public var videoList:ArrayCollection;  
      private var stream:NetStream;  
      private function init():void{  
        playerVideo.x=200;  
        connect();  
      }  
      private function connect():void{  
        nc.addEventListener(NetStatusEvent.NET_STATUS,netStatus);  
        nc.connect(rtmpURL);  
      }  
      private function clickItem(e:ListEvent):void{  
        container.addChild(playerVideo);  
        stream.play(videoList.getItemAt(e.rowIndex).label);  
      }  
  
      public function getVideos():void{  
        nc.call("flv.getListOfAvailableFLVs", responder);  
      }  
      public function getMediaList(list:Object):void{  
        var mediaList:Array = new Array();  
        for(var items:String in list){  
          mediaList.push({label:items, size:list[items].size,
```

作者：周英科 qq: 543246541
email: li_li_li87222@163.com

```
dateModified:list[items].lastModified});  
    }  
    videoList = new ArrayCollection(mediaList);  
  }  
  private function netStatus(e:NetStatusEvent):void{  
    trace(e.info.code);  
    if(e.info.code=="NetConnection.Connect.Success"){  
      getVideos();  
      stream=new NetStream(nc);  
      stream.client=new StreamClient();  
      playerVideo.attachNetStream(stream);  
    }  
  }  
}  
]]>  
</mx:Script>  
<mx:List itemClick="clickItem(event)" x="10" y="21" height="161"  
dataProvider="{videoList}"></mx:List>  
<mx:UIComponent id="container">  
</mx:UIComponent>  
</mx:Application>
```

运行客户端，可以看到一 **streams** 下的文件的名字。点击既可以播放。

附录：

Eclipse 的使用

新建工程：从菜单栏中选择文件菜单（file），然后选择新建（new）。从新建的类型里面选择 java 项目（java project）设置项目名称和保存的路径后点击确定然后点击下一步下一步就行了。

添加库：点击新建的项目，选择菜单栏的工程（project），选择 properties 项，左边选择 java build path，右边选择 libereais 面板，点击 add external jar,找到需要添加的 jar，点击确定。

Flexbuilder 的使用

新建工程：file-->new-->new Flex project-->填入工程名字-->finish

运行：ctrl+f11

调试：f11。调试可以看到 trace(); 语句的输出。

作者：周英科 qq: 543246541
email: li_li_li87222@163.com