

上海仰邦科技股份有限公司

# 全彩同步区域 Java 动态库使用 说明

动态库使用说明书

2015-3-30

# Copyright

*All rights reserved. No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by ONBON.*

© 2010-2014Onbon

## 版本历史:

版本号	日期	作者	描述
1.0	2015-3-30	龚丹	全彩同步区域 Java 动态库使用说明; 两个命令: 更新同步区域; 删除同步区域

## 目录

1. 同步区域动态库接口使用须知 .....	4
2. 同步区域动态库接口 .....	4
2.1 更新同步区域接口 .....	4
2.1.1 四级目录 .....	4
3. 格式说明 .....	4
3.1 更新同步区域格式说明 .....	4
3.2 删除同步区域格式说明 .....	5
4. 通讯接口 .....	6
4.1 使用通讯接口须知 .....	6
4.2 使用发送更新同步区域命令接口 .....	6
4.3 使用发送删除同步区域命令接口 .....	6
4.4 更新同步区域返回枚举 .....	7
5. 实时截屏接口 .....	7

## 全彩同步区域 Java 动态库使用说明

### 1. 同步区域动态库接口使用须知

调用动态库接口前，需初始化动态库；调用接口 YQConfigure 类中的 startup 函数。

```
YQConfigure.startup();
```

### 2. 同步区域动态库接口

#### 2.1 更新同步区域接口

根据格式产生更新同步区域物件信息 UpdateMessage。格式说明：请参考[更新同步区域信息格式](#)

##### 2.1.1 四级目录

根据格式产生更新同步区域物件信息 DeleteMessage。格式说明：请参考[删除同步区域信息格式](#)

### 3. 格式说明

#### 3.1 更新同步区域格式说明

更新同步区域格式。

属性	默认值	备注
DeviceType		设备类型 0x0058 – BX-YQ0-75 0x0158 – BX-YQ1-75 0x0258 – BX-YQ1 0x0358 – BX-YQ2 0x0458 – BX-YQ3
MsgSeq	0x0000	消息序列，用于区分各条消息 其有效范围为0x0000～ 0x7FFF 中的偶数，其相对对 应的返回帧 对应的消息序号为此序号 +0x8000
DataLen	N	数据长度
ArealID	0x00	区域序号，0-31
Suffix	bmp	图片格式后缀名，jpg，bmp， png
x		区域横坐标
y		区域纵坐标
w		区域宽度
h		区域高度
Transparency	0	不透明度 0-255
Program	0	0 是全局 1 是和节目 1 绑定 2 是和节目 2 绑定

		N 是和节目 N 绑定 0xffff 作为独立节目播放
Time	5	如果 program 为 0xffff, time 作为同步区播放时间, 单位为秒
picData		完整图片文件内容

### 3.2 删除同步区域格式说明

更新同步区域格式。

属性	默认值	备注
DeviceType		设备类型 0x0058 – BX-YQ0-75 0x0158 – BX-YQ1-75 0x0258 – BX-YQ1 0x0358 – BX-YQ2 0x0458 – BX-YQ3
MsgSeq	0x0000	消息序列, 用于区分各条消息 其有效范围为0x0000~ 0x7FFF 中的偶数, 其相对对应的返回帧 对应的消息序号为此序号 +0x8000
AreaID	0x00	区域序号, 0-31

## 4. 通讯接口

### 4.1 使用通讯接口须知

使用通讯接口时，先初始化 protocol(即初始化 BxcsController 类；传入 aliasName)；然后创建 Socket 连接；传入 IP 地址和端口。

```
BxcsController controller = new BxcsController("Test");
controller.connect("192.168.1.100", 5005);
```

### 4.2 使用发送更新同步区域命令接口

使用更新同步区域命令通讯接口，初始化 BxcsWrite 类，传入 Socket；然后创建更新同步区域命令信息；更新同步区域命令接口函数返回值为：SendResult；是一个枚举类型。请参考：[通讯返回枚举](#)

```
BxcsWrite write = new BxcsWrite (controller.getSocket()); //初始化更新同步区域命令通讯接口类

ArrayList<UpdateMessage>datas = new ArrayList<UpdateMessage>();
//创建更新同步区域命令信息
UpdateMessage um = new UpdateMessage();
//控制卡类型转换
byte[] type = YQHelper.controllerTypeUtils(1);
m.setDeviceType(type);
m.setAreaId(0);
m.setX(0);
m.setY(0);
m.setProgram(0);
m.setSuffix("bmp");
m.setW(64);
m.setH(16);
m.setBlockData(DisplayHelper.convert("C:\\temp\\12.bmp")); //或者使用截屏发送图片数据

datas.add(data);

SendResult result = write.write(datas);
```

### 4.3 使用发送删除同步区域命令接口

使用删除同步区域命令通讯接口。

```
//创建删除同步区域命令信息类
DeleteMessage m = new DeleteMessage();
//控制卡类型转换
byte[] type = YQHelper.controllerTypeUtils(1);
m.setDeviceType(type);
m.setAreaId(0);
//回复数据
BxcsCallback<AckMessage> callback = new BxcsCallback<AckMessage>() {

@Override
public void handle(AckMessage t) {
    System.out.println("handle");
}
}
```

```
@Override
public void nak(NakMessage nak) {
    System.out.println("nak");
}
@Override
public void timeout() {
    System.out.println("timeout");
}
};
//删除函数接口
controller.delete(m, callback);
```

#### 4.4 更新同步区域返回枚举

```
public enum SendResult {
    ERR_NOREPLY,
    ACK,
    NACK,
    ERR_SEND
}
```

## 5. 实时截屏接口

使用实时截屏接口，初始化ScreenStreaming类；然后启动截屏；在CaptureHandler中获取截屏数据，并发给屏幕

```
ScreenStreaming ss = new ScreenStreaming();
ss.setCaptureHandler(new ScreenEventHandler() {
    @Override
    public void captured(byte[] data) {
        //TODO...
        //将截屏数据发送到屏幕；将截屏数据赋给 UpdateMessagesetBlockData 属性
    }
});
//开始截屏
ss.start(X(截屏 X 坐标), Y(截屏 Y 坐标), W(截屏宽度), H(截屏高度));
//停止截屏
ss.stop();
```