

# NetSDK\_JAVA 主动注册

## 使用说明书













# 前言

## 概述

本文档详细描述了 NetSDK JAVA 版本主动注册的 Demo 演示和开发基本流程。

## 符号约定

在本文档中可能出现下列标识，代表的含义如下。

标识	说明
 <b>危险</b>	表示有高度潜在危险，如果不能避免，会导致人员伤亡或严重伤害。
 <b>警告</b>	表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
 <b>注意</b>	表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 <b>防静电</b>	表示静电敏感的设备。
 <b>当心触电</b>	表示高压危险。
 <b>激光辐射</b>	表示强激光辐射。
 <b>风扇警告</b>	表示危险运动部件，请远离运动风扇叶片。
 <b>当心机械伤人</b>	表示设备部件机械伤人。
 <b>窍门</b>	表示能帮助您解决某个问题或节省您的时间。
 <b>说明</b>	表示是正文的附加信息，是对正文的强调和补充。

## 修订记录

版本号	修订内容	发布日期
V1.0.0	首次发布。	2020.10

# 目录

前言.....	I
第 1 章 概述.....	1
1.1 简介 .....	1
1.2 主动注册使用示意.....	1
第 2 章 主动注册 Demo 操作 .....	2
第 3 章 主动注册接口调用.....	6
3.1 配置设备主动注册信息 .....	6
3.1.1 简介 .....	6
3.1.2 接口总览.....	6
3.1.3 流程说明.....	6
3.1.4 示例代码.....	8
3.2 启动/停止主动注册服务 .....	9
3.2.1 简介 .....	9
3.2.2 接口总览.....	10
3.2.3 流程说明.....	10
3.2.4 示例代码.....	11
附录 1 法律声明 .....	13
附录 2 网络安全建议.....	14

# 第 1 章 概述

## 1.1 简介

NetSDK 主动注册主要用于解决诸如公网服务器无法搜索内网设备等网络限制问题，或是设备端 IP 不固定的情况（例如，设备处于 4G 网络或者 Wi-Fi 网络时 IP 可能不固定）。同时也有利于监听服务器高效配置设备，实现应用场景的快速实现或恢复。

NetSDK 涉及的主动注册功能主要包括三个部分：

- 获取与配置目标设备的主动注册信息。
- 监听服务器的启动和停止。
- 监听服务器快速添加、删除处于监听本服务器状态的设备。

Java 版本的 Demo 提供了上述三个主要功能的基本实现，另外还提供了设备信息的导入导出，并对单个设备提供了拉流、对讲等基本功能的演示。

## 1.2 主动注册使用示意

主动注册功能的示意图如图 1-1 所示，详细说明请参见表 1-1。

图1-1 主动注册功能示意图

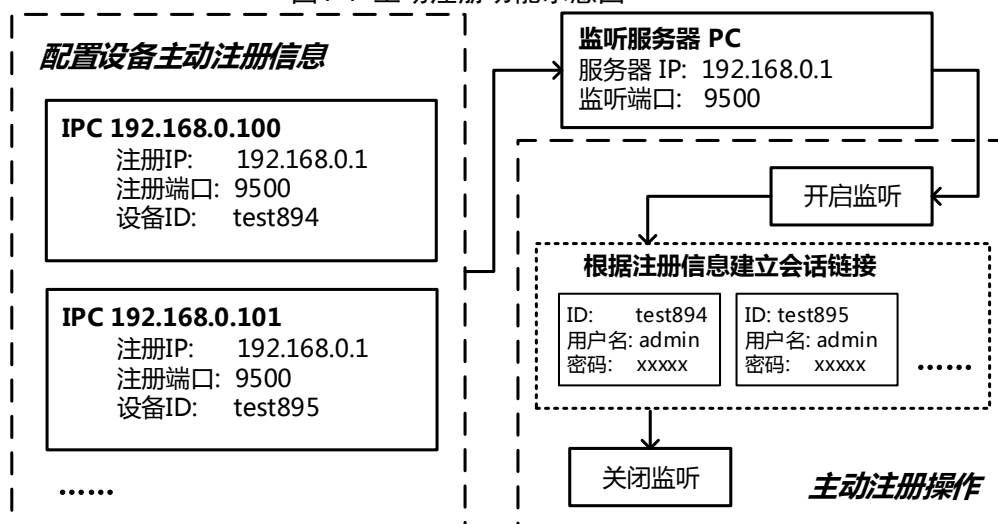


表1-1 主动注册使用说明

序号	功能	说明
1	设备主动注册信息的获取和配置	配置设备主动注册信息有两种方式：设备 Web 端配置和 SDK 配置。SDK 支持固定 IP 端设备主动注册信息的获取和配置，详细介绍请参见“第 2 章主动注册 Demo 操作”。
2	服务器监听的开启和停止	SDK 提供了主动注册监听的开启和停止。

# 第 2 章 主动注册 Demo 操作

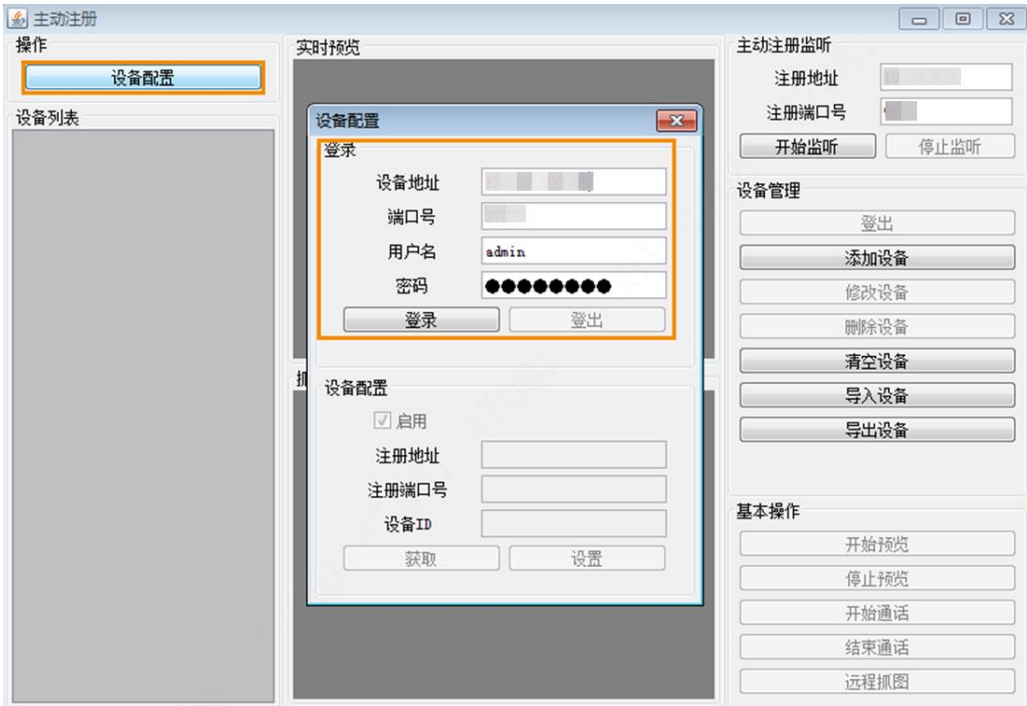
配置设备主动注册到平台。支持通过 Demo 配置或在设备端直接配置主动注册，本文以通过 Demo 配置为例。

## 说明

当由于网络限制，平台访问不到设备时，请采用设备 WEB 端配置主动注册。关于在设备 WEB 端配置主动注册的操作流程，请参见设备的使用说明书。

- 步骤1 启动 Demo。运行 AutoRegister.java 文件，打开主动注册界面。
- 步骤2 登录设备。单击“设备配置”，在弹出的“设备配置”对话框输入设备的 IP 地址、端口号、用户名和密码，单击“登录”。

图2-1 登录设备



- 步骤3 获取和配置主动注册。
  - 1. 单击“获取”，从设备端获取主动注册配置信息。
  - 2. （可选）如果获取到的配置信息不是设备需要注册到的平台，请将信息修改为设备需要注册到的平台的信息。

## 说明

设备 ID 可任意设置，但每台设备的 ID 不能重复。


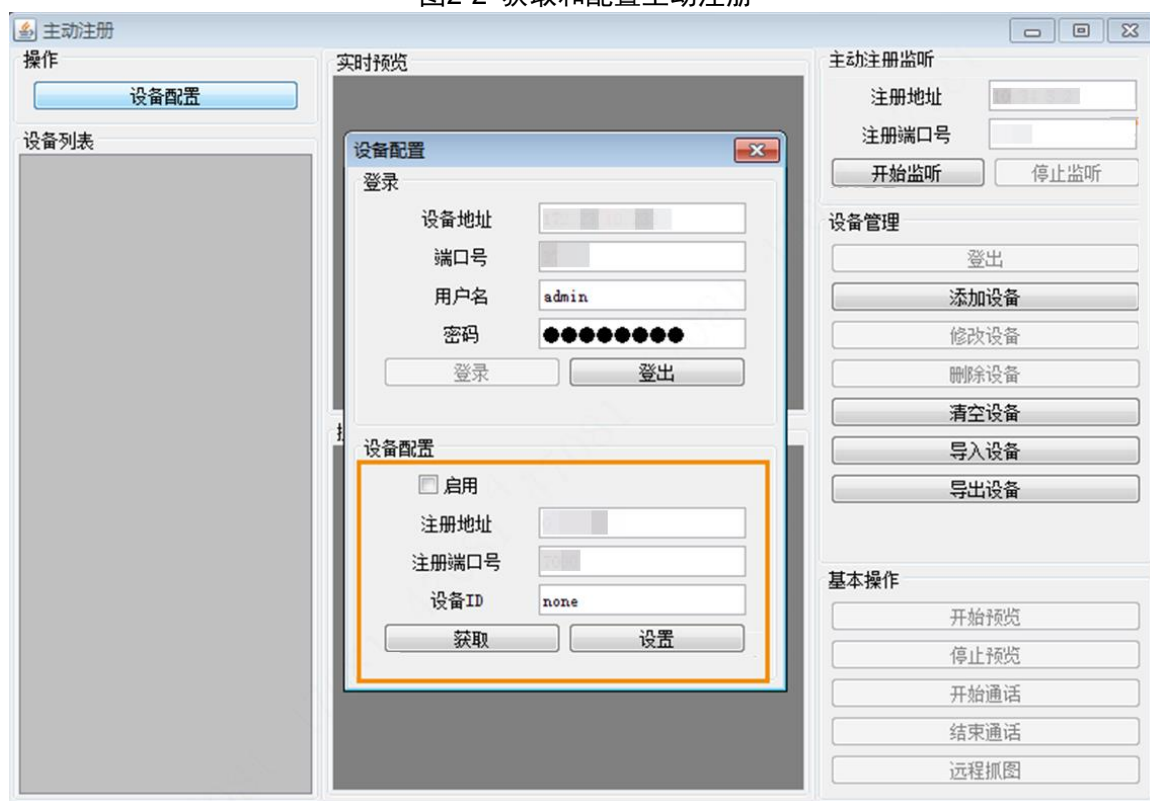
- 3. 选择“启用”，单击“设置”。  
平台下发主动注册信息到设备，主动注册配置成功。
- 4. 单击“登出”，然后单击“设备配置”对话框右上角的 。  
退出主动注册配置界面。

图2-2 获取和配置主动注册



- 步骤4 配置监听地址和端口，单击“开始监听”。
- 平台开启主动注册监听。
- 步骤5 添加设备到设备列表。单击“添加设备”，在弹出的对话框填写设备 ID、用户名和密码，单击“添加”。
- 等待设备主动注册到平台。当设备名称显示为绿色且能查看该设备通道，表示设备成功注册到平台。

图2-3 监听设备并添加设备到设备列表

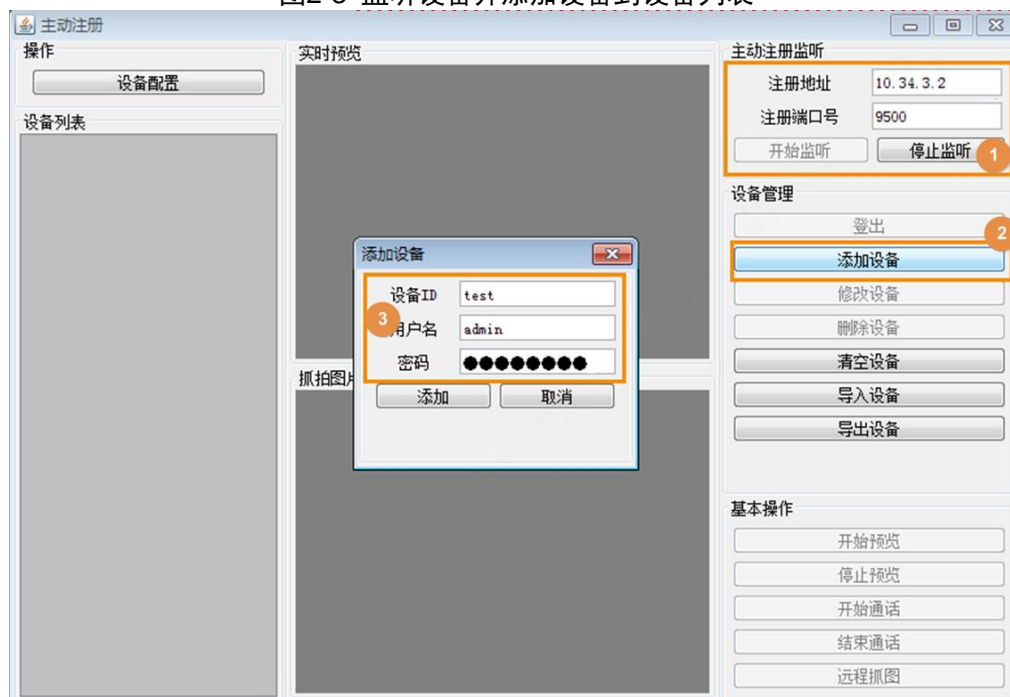


图2-4 成功添加设备到设备列表

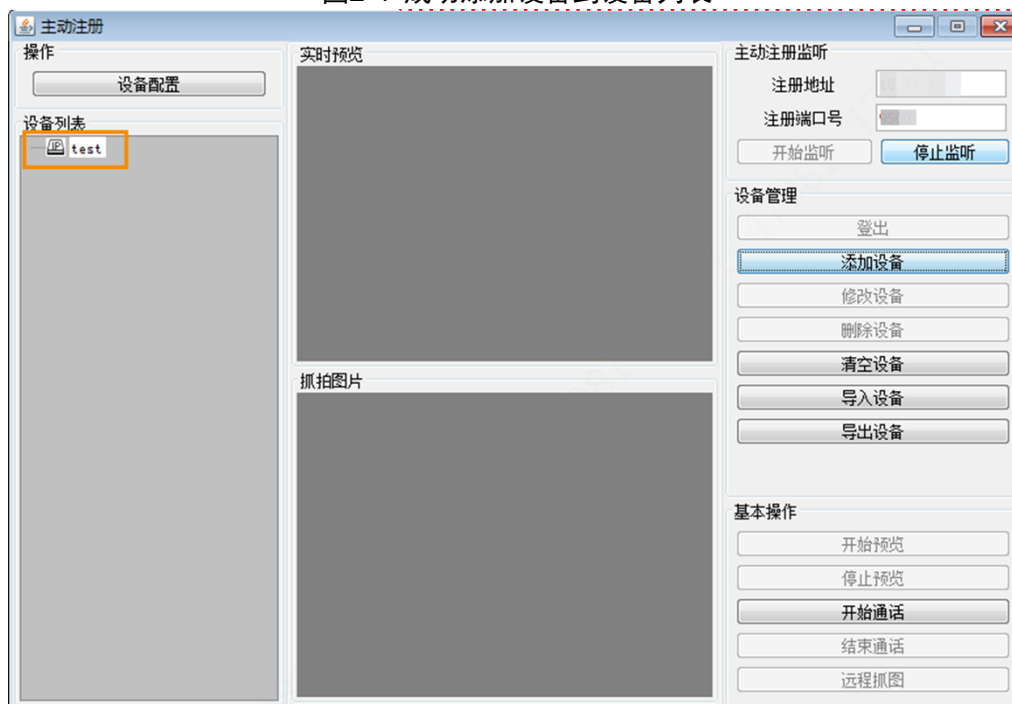
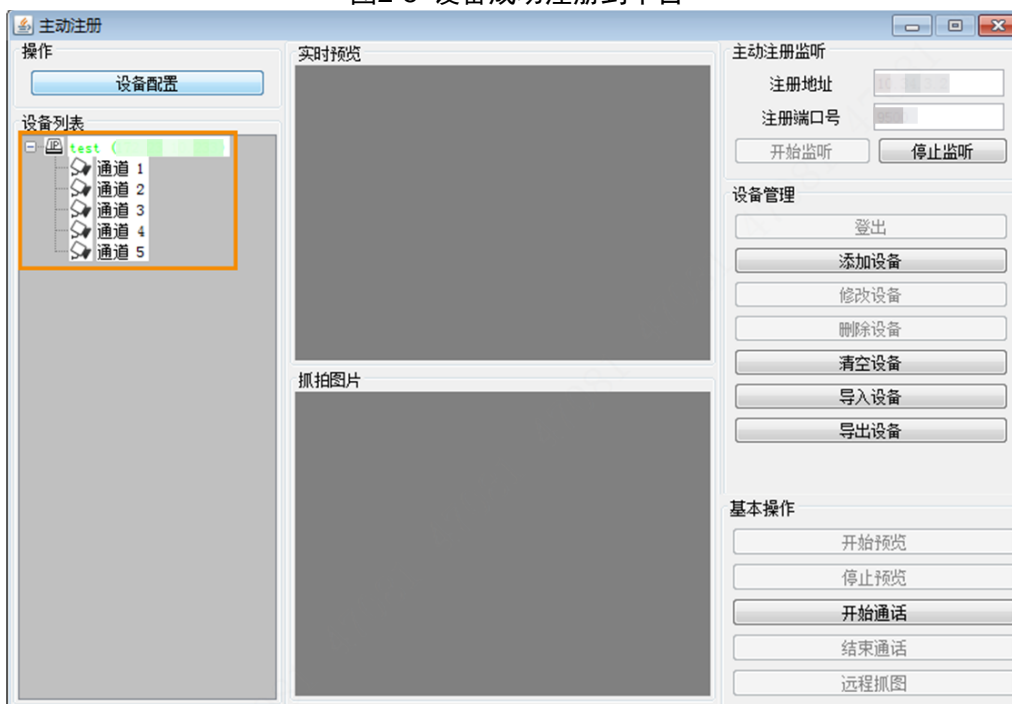
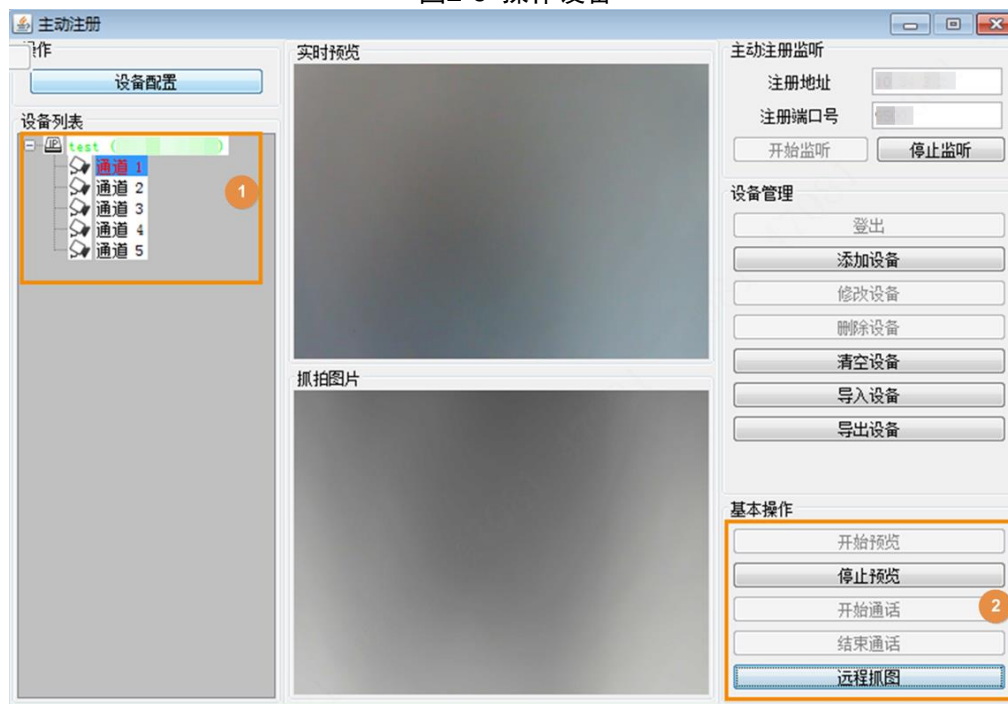


图2-5 设备成功注册到平台



步骤6 操作设备。在设备列表选择设备通道,在右侧“基本操作”区域根据界面提示执行操作,例如实时预览监控画面、抓图、与设备端通话等。

图2-6 操作设备





## 第 3 章 主动注册接口调用

### 3.1 配置设备主动注册信息

#### 3.1.1 简介

主动注册配置，即用户通过调用 SDK 接口，对设备的主动注册信息进行配置，包括配置主动注册使能、设备 ID 和要注册到的平台的 IP 地址等。

#### 3.1.2 接口总览

表3-1 设备主动注册配置接口说明

接口	说明
CLIENT_Init	初始化 SDK
CLIENT_LoginWithHighLevelSecurity	高安全级别登录
CLIENT_GetNewDevConfig	获取配置信息
CLIENT_ParseData	解析配置信息
CLIENT_PacketData	组合要下发的配置信息
CLIENT_SetNewDevConfig	下发配置信息
CLIENT_Logout	登出设备
CLIENT_Cleanup	清理 SDK 资源

#### 3.1.3 流程说明

获取/配置设备主动注册的基本流程如图 3-1 所示。

图3-1 获取/设置设备主动注册基本流程



## 流程说明

- 步骤1 调用 `CLIENT_Init` 初始化 SDK。
- 步骤2 调用 `CLIENT_LoginWithHighLevelSecurity` 接口登录到设备。
- 步骤3 （可选）调用 `CLIENT_GetNewDevConfig` 接口获取设备的主动注册配置信息，并调用 `CLIENT_ParseData` 接口解析配置信息。
- 步骤4 调用 `CLIENT_PacketData` 组装设置好的主动注册配置信息。
- 步骤5 调用 `CLIENT_SetNewDevConfig` 发送主动注册配置信息。
- 步骤6 调用 `CLIENT_Logout` 接口登出设备。
- 步骤7 调用 `CLIENT_Cleanup` 接口清理释放 SDK 资源。

## 注意事项

- `CLIENT_Init` 和 `CLIENT_Cleanup` 接口需成对调用，支持多线程多次成对调用，建议全局均只调用一次，勿反复的初始化/清理资源。
- 虽可直接编写参数下发配置，但建议在原来配置的基础上修改下发配置，从而避免出现缺漏关键字段的情况。
- 用户需要事先确定设备是否支持主动配置。
- SDK 获取/配置注册配置前需要先完成 SDK 的基本操作，包括初始化并登录到目标设备。

### 3.1.4 示例代码

```
// 初始化、清理资源等相关的通用代码请参考《NetSDK_Java 编程指导手册》
//主动注册配置的获取和写入

//获取主动注册配置信息
/**
 * 获取网络协议
 * @param m_hLoginHandle 登录句柄
 * @return
 */
public static CFG_DVRIP_INFO getDVRIPConfig(LLong m_hLoginHandle) {
    CFG_DVRIP_INFO msg = new CFG_DVRIP_INFO();
    if(!ToolKits.GetDevConfig(m_hLoginHandle, -1, NetSDKLib.CFG_CMD_DVRIP, msg)){
        return null;
    }
    return msg;
}

//使用的 ToolKits 的获取配置方法,该方法的简单实现如下,具体实现请参考 ToolKits.java
/*
 * 获取单个配置
 * @param hLoginHandle 登录句柄
 * @param nChn 通道号 -1 表示全通道
 * @param strCmd 配置名称
 * @param cmdObject 配置对应的结构体对象
 * @return 成功返回 true
 */
public static boolean GetDevConfig(LLong hLoginHandle,int nChn,String strCmd,Structure cmdObject){
    //...初始化参数略
    //获取设备配置
    if(netsdkapi.CLIENT_GetNewDevConfig( hLoginHandle, strCmd , nChn, strBuffer,
nBufferLen,error,3000)){
        cmdObject.write();
        //解析获取到的配置信息到 cmdObject
        if(configapi.CLIENT_ParseData(strCmd,strBuffer,cmdObject.getPointer(),cmdObject.size(),null)){
            cmdObject.read();
            //...略
        }
    }
}

//设置主动注册配置信息
/**
 * 网络协议配置
 * @param m_hLoginHanle 登录句柄
 * @param enable 使能
```

```

* @param address 服务器地址
* @param nPort 服务器端口号
* @param deviceId 设备 ID
* @param info 获取到的网络协议配置
* @return
*/
public static boolean setDVRIPConfig(LLong m_hLoginHandle, boolean enable, String address, int
nPort, byte[] deviceId, CFG_DVRIP_INFO info){
    CFG_DVRIP_INFO msg=info;
    //参数配置,略,具体请参考 AutoRegisterModule.java
    //下发配置
    return ToolKits.SetDevConfig(m_LoginHandle,-1,NetSDKLib.CFG_CMD_DVRIP,msg);
}

/*
* 设置单个配置,该代码在 ToolKits.java 内
* @param hLoginHandle 登录句柄
* @param nChn 通道号,-1,表示全通道
* @param strCmd 配置名称
* @param cmdObject 配置对应的结构体
* @return 成功返回 true
*/
public static boolean SetDevConfig(LLong hLoginHandle,int nChn,String strCmd,Structure cmdObject){
    //参数配置, 略
    cmdObject.write();
    //打包配置数据
    if(configapi.CLIENT_PacketData(strCmd,cmdObject.getPointer(),cmdObject.size(),szBuffer,nBuffer
Len)){
        cmdObject.read();
        //下发配置
        if(netsdkapi.CLIENT_SetNewDevConfig(hLoginHandle,strCmd,nChn,szBuffer,nBufferLen,error
,restart,30000)){
            result=true;
        }else{
            result=false;
        }
    }
    return result;
}
}

```

## 3.2 启动/停止主动注册服务

### 3.2.1 简介

SDK 提供了主动注册服务，用于监听感知注册到平台的设备，包括启动监听和停止监听。

### 3.2.2 接口总览

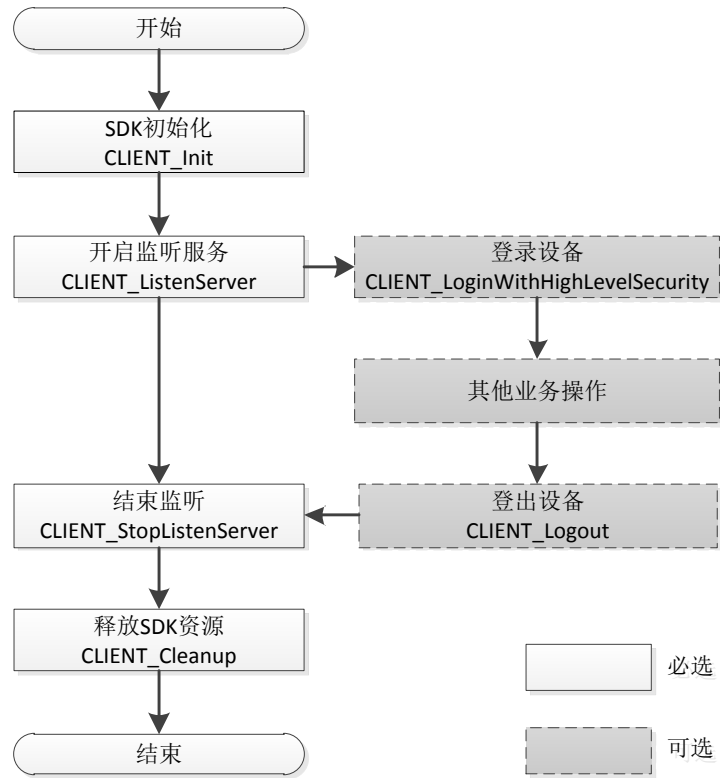
表3-2 启动/停止主动注册服务接口说明

接口	说明
CLIENT_Init	初始化 SDK
CLIENT_ListenServer	开始主动注册监听
CLIENT_StopListenServer	停止主动注册监听
CLIENT_Cleanup	清理 SDK 资源

### 3.2.3 流程说明

平台启动/停止主动注册服务的基本流程如图 3-2 所示。

图3-2 启动/停止监听服务基本流程



#### 流程说明

- 步骤1 调用 CLIENT\_Init 初始化接口，完成 SDK 初始化。
- 步骤2 调用 CLIENT\_ListenServer 接口开始监听。CLIENT\_ListenServer 接口需要传入主动注册监听回调函数 fServiceCallBack，通过该回调函数可以获取到注册到本服务平台的设备的 IP、ID 和端口等信息。监听状态下，回调函数返回设备主动注册传上来的 ID、IP 及端口等信息，用户在回调中获取到信息后可以登录需要的设备并执行其他业务操作。
- 步骤3 调用 CLIENT\_StopListenServer 接口结束监听。
- 步骤4 调用 CLIENT\_Cleanup 接口清理释放 SDK 资源。

#### 注意事项

- CLIENT\_Init 和 CLIENT\_Cleanup 接口需成对调用，支持多线程多次成对调用，但我们建议全局均只调用一次，勿反复初始化/清理资源。
- 请确保设备主动注册配置中的目标 IP 与端口和服务器 IP 与端口完全匹配。
- 服务器只有处于监听状态下时才能向主动注册的设备发起通信，因此所有需要服务器向设备下发命令的操作都必须在侦听状态下完成，不能中途停止监听。
- 如果需要在回调函数中调用 SDK 的接口，请另起一个线程执行，否则容易引起程序崩溃。

### 3.2.4 示例代码

```
// 初始化、清理资源等相关的通用代码请参考《NetSDK_Java 编程指导手册》
// 配置监听回调函数
/**
 * 侦听服务器回调函数
 */
public class ServiceCB implements fServiceCallBack {
    @Override
    public int invoke(LLong lHandle, final String plp, final int wPort,
        int lCommand, Pointer pParam, int dwParamLen, Pointer dwUserData) {
        // 将 pParam 转化为序列号
        byte[] buffer = new byte[dwParamLen];
        pParam.read(0, buffer, 0, dwParamLen);
        String deviceId = "";
        try {
            deviceId = new String(buffer, "GBK").trim();
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        switch(lCommand) {
            // 验证期间设备断线回调
            case EM_LISTEN_TYPE.NET_DVR_DISCONNECT: {
                //略，具体请参考 AutoRegister.javabreak;
            }
            // 设备注册携带序列号
            case EM_LISTEN_TYPE.NET_DVR_SERIAL_RETURN: {
                //界面操作,略
                //另起线程执行登录操作
                new SwingWorker<LLong, String>() {
                    @Override
                    protected LLong doInBackground() {
                        return login(deviceTreeNode);
                    }
                    @Override
                    protected void done() {
                        try {
                            if(get() == null) {return;}
                            //更新界面操作,略
                        }.execute();
                        break;
                    }
                }
                break;
            }
            default:
                break;
        }
    }
}
```

```

    }
    return 0;
}

}

//开始监听并注册侦听服务回调函数
private ServiceCB serviceCallback=new ServiceCB();
/**
 * 开启服务
 * @param address 本地 IP 地址
 * @param port 本地端口, 可以任意
 * @param callback 回调函数
 */
public static boolean startServer(String address, int port, fServiceCallBack callback) {
    mServerHandler = LoginModule.netsdk.CLIENT_ListenServer(address, port, 1000, callback, null);
    if (0 == mServerHandler.longValue()) {
        System.err.println("Failed to start server." + ToolKits.getErrorCodePrint());
    } else {
        System.out.printf("Start server, [Server address %s][Server port %d]\n", address, port);
    }
    return mServerHandler.longValue() != 0;
}

//结束监听

/**
 * 结束服务
 */
public static boolean stopServer() {
    boolean bRet = false;
    //使用开启服务获得的监听句柄
    if(mServerHandler.longValue() != 0) {
        bRet = LoginModule.netsdk.CLIENT_StopListenServer(mServerHandler);
        mServerHandler.setValue(0);
        System.out.println("Stop server!");
    }
    return bRet;
}

```

# 附录1 法律声明

## 商标声明

- VGA 是 IBM 公司的商标。
- Windows 标识和 Windows 是微软公司的商标或注册商标。
- 在本文档中可能提及的其他商标或公司的名称，由其各自所有者拥有。

## 责任声明

- 在适用法律允许的范围内，在任何情况下，本公司都不对因本文档中相关内容及描述的产品而产生任何特殊的、附随的、间接的、继发性的损害进行赔偿，也不对任何利润、数据、商誉、文档丢失或预期节约的损失进行赔偿。
- 本文档中描述的产品均“按照现状”提供，除非适用法律要求，本公司对文档中的所有内容不提供任何明示或暗示的保证，包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证。

## 隐私保护提醒

您安装了我们的产品，您可能会采集人脸、指纹、车牌、邮箱、电话、GPS 等个人信息。在使用产品过程中，您需要遵守所在地区或国家的隐私保护法律法规要求，保障他人的合法权益。如，提供清晰、可见的标牌，告知相关权利人视频监控区域的存在，并提供相应的联系方式。

## 关于本文档

- 产品请以实物为准，本文档仅供参考。
- 本公司保留随时维护本文档中任何信息的权利，维护的内容将会在本文档的新版本中加入，恕不另行通知。
- 本文档如有不准确或不详尽的地方，或印刷错误，请以公司最终解释为准。
- 本文档供多个型号产品做参考，每个产品的具体操作不逐一例举，请用户根据实际产品自行对照操作。
- 如不按照本文档中的指导进行操作，因此而造成的任何损失由使用方自行承担。
- 如获取到的 PDF 文档无法打开，请将阅读工具升级到最新版本或使用其他主流阅读工具。



## 附录2 网络安全建议

### 保障设备基本网络安全的必须措施：

#### 1. 使用复杂密码

请参考如下建议进行密码设置：

- 长度不小于 8 个字符。
- 至少包含两种字符类型，字符类型包括大小写字母、数字和符号。
- 不包含账户名称或账户名称的倒序。
- 不要使用连续字符，如 123、abc 等。
- 不要使用重叠字符，如 111、aaa 等。

#### 2. 及时更新固件和客户端软件

- 按科技行业的标准作业规范，设备的固件需要及时更新至最新版本，以保证设备具有最新的功能和安全性。设备接入公网情况下，建议开启在线升级自动检测功能，便于及时获知厂商发布的固件更新信息。
- 建议您下载和使用最新版本客户端软件。

### 增强设备网络安全的建议措施：

#### 1. 物理防护

建议您对设备（尤其是存储类设备）进行物理防护，比如将设备放置在专用机房、机柜，并做好门禁权限和钥匙管理，防止未经授权的人员进行破坏硬件、外接设备（例如 U 盘、串口）等物理接触行为。

#### 2. 定期修改密码

建议您定期修改密码，以降低被猜测或破解的风险。

#### 3. 及时设置、更新密码重置信息

设备支持密码重置功能，为了降低该功能被攻击者利用的风险，请您及时设置密码重置相关信息，包含预留手机号/邮箱、密保问题，如有信息变更，请及时修改。设置密保问题时，建议不要使用容易猜测的答案。

#### 4. 开启账户锁定

出厂默认开启账户锁定功能，建议您保持开启状态，以保护账户安全。在攻击者多次密码尝试失败后，其对应账户及源 IP 将会被锁定。

#### 5. 更改 HTTP 及其他服务默认端口

建议您将 HTTP 及其他服务默认端口更改为 1024~65535 间的任意端口，以减小被攻击者猜测服务端口的风险。

#### 6. 使能 HTTPS

建议您开启 HTTPS，通过安全的通道访问 Web 服务。

#### 7. MAC 地址绑定

建议您在设备端将其网关设备的 IP 与 MAC 地址进行绑定，以降低 ARP 欺骗风险。

#### 8. 合理分配账户及权限

根据业务和管理需要，合理新增用户，并合理为其分配最小权限集合。

#### 9. 关闭非必需服务，使用安全的模式

如果没有需要，建议您关闭 SNMP、SMTP、UPnP 等功能，以降低设备面临的风险。

如果有需要，强烈建议您使用安全的模式，包括但不限于：

- SNMP：选择 SNMP v3，并设置复杂的加密密码和鉴权密码。
- SMTP：选择 TLS 方式接入邮箱服务器。
- FTP：选择 SFTP，并设置复杂密码。
- AP 热点：选择 WPA2-PSK 加密模式，并设置复杂密码。

## 10. 音视频加密传输

如果您的音视频数据包含重要或敏感内容，建议启用加密传输功能，以降低音视频数据传输过程中被窃取的风险。

## 11. 安全审计

- 查看在线用户：建议您不定期查看在线用户，识别是否有非法用户登录。
- 查看设备日志：通过查看日志，可以获知尝试登录设备的 IP 信息，以及已登录用户的关键操作信息。

## 12. 网络日志

由于设备存储容量限制，日志存储能力有限，如果您需要长期保存日志，建议您启用网络日志功能，确保关键日志同步至网络日志服务器，便于问题回溯。

## 13. 安全网络环境的搭建

为了更好地保障设备的安全性，降低网络安全风险，建议您：

- 关闭路由器端口映射功能，避免外部网络直接访问路由器内网设备的服务。
- 根据实际网络需要，对网络进行划区隔离：若两个子网间没有通信需求，建议使用 VLAN、网闸等方式对其进行网络分割，达到网络隔离效果。
- 建立 802.1x 接入认证体系，以降低非法终端接入专网的风险。
- 开启设备 IP/MAC 地址过滤功能，限制允许访问设备的主机范围。