# 概述

## 文档简介

本文档主要提供鸿蒙版HiLinkSvc Kit的集成说明。包含设备管理AA、设备添加AA的集成指导。

## 基本概念及功能说明

| 名称 | 说明 |
| --- | --- |
| 设备管理AA | HiLink Service（以下简称HiLinkSvc）设备管理原子能力（Atomic Ability），向FA（Feature Ability）提供可获取设备列表、控制设备等设备管理的接口能力。 |
| 设备配网AA | HiLinkSvc配网原子能力（Atomic Ability），向FA（Feature Ability）提供可对设备进行配网的接口能力。 |

# 软件包集成说明

## 注意事项

为保证FA正常调用文档中的接口，需要提供FA的BundleName和证书指纹信息，且此信息不建议随意变更。

## 设备管理

### 导入JS API包

将fa-DeviceManager.rar解压到entry/node\_modules目录下，在js/app.js中导入设备管理模块：

var deviceManager = require('fa-deviceManager');

export default {

onCreate() {

console.info('AceApplication onCreate');

},

onDestroy() {

console.info('AceApplication onDestroy');

},

DeviceManager: deviceManager.FADeviceManager

}

### 导入Har包

将Har包放到entry/libs下，修改build.gradle文件，在dependencies中加载Har文件：

dependencies {  
 implementation fileTree(dir: 'libs', include: ['\*.har'])  
}

在MainAbility的onStart()中注册，且要放在super.onStart(intent)前：

public class MainAbility extends AceAbility {

@Override  
public void onStart(Intent intent) {

……

DeviceMgrAbility.register(this);

super.onStart(intent);

}

在FA根据业务场景不再需要HiLinkSvc Kit功能时需要去注册：

DeviceMgrAbility.deregister();

### 接口调用

使用this.$app.$def.DeviceManager.XXX调用指定接口。

命令发送响应消息体code与data取值如下：

|  |  |  |  |
| --- | --- | --- | --- |
| code | data | 说明 | 处理建议 |
| 0 | 0 | 消息下发成功 |  |
| -1 | 1 | 接口参数错误 | 请检查接口参数 |
| 2 | IDL与AIDL连接失败 | 重试 |
| 3 | 接口code值错误 | 请检查接口参数code值 |
| 4 | 设备管理Service不支持 | 确认设备管理Service版本 |
| 5 | 请求超时 | 延迟重试 |

回调响应消息体code取值如下：

|  |  |
| --- | --- |
| code | 说明 |
| 0 | 接口回调返回成功，data数据见对应接口回调数据说明。 |
| -1 | 接口回调返回失败。 |

## 设备配网

### 导入JS API包

将fa-netconfig.rar解压到entry/node\_modules目录下，在js/app.js中导入配网模块：

var netConfig = require('fa-netconfig');

export default {

onCreate() {

console.info('AceApplication onCreate');

},

onDestroy() {

console.info('AceApplication onDestroy');

},

NetConfig: netConfig.FANetConfig

}

### 导入Har包

将Har包放到entry/libs下，修改build.gradle文件，在dependencies中加载Har文件：

dependencies {  
 implementation fileTree(dir: 'libs', include: ['\*.har'])  
}

在MainAbility的onStart()中注册，且要放在super.onStart(intent)前：

public class MainAbility extends AceAbility {

@Override  
public void onStart(Intent intent) {

……

NetConfigAbility.register(this, sessionId);

super.onStart(intent);

}

注：sessionId为HiLinkSvc拉起FA时的Intent携带的sessionId值，如果FA启动时intent没有携带sessionId，则设置为“”。

Intent携带的数据：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 说明 | Key |
| sessionId | String | NAN连接会话ID | “nanSessionId” |
| nfcTagId | byte[] | NFC标签ID | “tagUid” |
| isBound | boolean | 设备是否绑定标识，true标识已绑定 | “isBound” |

其解析方法，以解析sessionId为例：

String sessionId = intent.getStringParam("nanSessionId");

在FA根据业务场景不再需要HiLinkSvc Kit功能时需要去注册：

NetConfigAbility.deregister();

### 接口调用

使用this.$app.$def.NetConfig.XXX调用指定接口。

命令发送响应消息体code与data取值如下：

|  |  |  |  |
| --- | --- | --- | --- |
| code | data | 说明 | 处理建议 |
| 0 | 0 | 消息下发成功 |  |
| -1 | 1 | 接口参数错误 | 请检查接口参数 |
| 2 | IDL与AIDL连接失败 | 重试 |
| 3 | 接口code码错误 | 请检查接口参数code码 |
| 4 | 配网Service不支持 | 确认配网Service版本 |

回调响应消息体code取值如下：

|  |  |
| --- | --- |
| code | 说明 |
| 0 | 接口回调返回成功，data数据见对应接口回调数据说明。 |
| -1 | 接口回调返回失败。 |

# 功能说明

## 设备管理

### 接口说明

#### 接口：获取设备列表

##### 接口定义

| 接口定义 |
| --- |
| DeviceManager.getSummaryDevices(source, callback) |

##### 接口描述

用于获取已经登录的华为账号关联的智慧生活设备简要信息列表。

##### 参数说明

| 参数 | 类型 | M(必选)/O(可选） | 描述 |
| --- | --- | --- | --- |
| source | String | M | 数据来源：  “local”：从本地获取缓存数据  “cloud”：从云端获取最新数据 |
| callback | function | M | 回调函数。 |

##### 返回值说明

方法无返回值，结果由callback返回：

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| errorCode | int | M | 错误码：  0：成功  -1：失败 |
| message | String | M | 错误原因简要描述。 |
| data | String | O | 回调成功的情况下，即errorCode等于0时，data为返回的数据结果，格式为json字符串，在该回调函数中解析数据。如下：  [  {  "roomId": 1010,  "roomName": "room1",  "deviceId": "xxxxx",  "deviceName": "device1",  "productId": "101C",  "status": "online",  "role": "owner",  "homeId": "84959d9",  "deviceIconUrl": "https://xxxx",  "quickMenuInfo": {  "text": "owner",  "icon": "84959d9",  "openStatus": 1  }  }  ] |
|  |  |  |  |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：获取设备详情

##### 接口定义

| 接口定义 |
| --- |
| DeviceManager.getProfileData(deviceId, callback) |

##### 接口描述

用于获取已经登录的华为账号关联的智慧生活设备详情。

##### 参数说明

| 参数 | 类型 | M(必选)/O(可选） | 描述 |
| --- | --- | --- | --- |
| deviceId | String | M | 设备标识，获取方式见3.1.1.1。  最大长度40个字符。 |
| callback | function | M | 回调函数。 |

##### 返回值说明

方法无返回值，结果由callback返回：

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| errorCode | int | M | 错误码：  0：成功  -1：失败 |
| message | String | M | 错误原因简要描述。 |
| data | String | O | 回调成功的情况下，即errorCode等于0时，data为返回的数据结果，格式为json字符串，在该回调函数中解析数据。数据格式基于profile呈现，如下：  [  {  "deviceId": "xxxxx",  "status": "online",  "services": [{  "st": "light",  "ts": "20151212T121212Z",  "sid": "light1",  "data": {  "characteristicName1": "value1",  "characteristicName2": "value2"  }  }],  "devInfo": {  "sn": "00E0FC018008",  "model": "SmartSpeaker",  "devType": "004",  "mac": "",  "protType": 1,  "prodId": "000b"  }  }  ] |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：获取设备配置文件

##### 接口定义

| 接口定义 |
| --- |
| DeviceManager.getDeviceMetaData (productId, metaType, callback) |

##### 接口描述

用于获取设备的配置文件。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| productId | String | M | 产品标识。获取方式参3.1.1.1。  最大长度32个字符。 |
| metaType | String | M | 数据类型，如下：  “mainhelp”：白名单配置文件  “profile”：设备功能清单文件 |
| callback | function | M | 回调函数。 |

##### 返回值说明

方法无返回值，结果由callback返回：

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| errorCode | int | M | 错误码：  0：成功  -1：失败 |
| message | String | M | 错误原因简要描述。 |
| data | String | O | 回调成功的情况下，即errorCode等于0时，data为返回的数据结果，格式为json字符串，在该回调函数中解析数据。mainhelp数据如：  {  "DeviceTypeName": "网络摄像头",  "DeviceId": "2BBH",  "ManufacturerId": "04D",  "icon": "2BBH/iconB.png",  "DeviceNameEn": "HUAWEI Hilink Smart Camera Pro",  "deviceSn": "",  "mac": "",  "DeviceNameSpreading": "华为智选 海雀智能摄像头Pro 32GB",  "ManufacturerFullName": "深圳市海雀科技有限公司",  "deviceVersionCode": "11",  "helpDesc": "2BBH/help.html",  "DeviceModel": "GD01",  "DeviceTypeNameEn": "WiFi Camera",  "rssi": 1,  "ManufacturerName": "海雀",  "deviceIcon": "2BBH/iconD.png",  "profile": "2BBH/2BBH.json",  "deviceIcon1": "2BBH/iconA.png",  "ManufacturerNameEn": "Haique ",  "ManufacturerFullNameEn": "Haique Inc",  "onlineIcon": "2BBH/2BBH\_F.png",  "DeviceNameSpreadingEn": "HUAWEI Hilink Smart Camera Pro",  "offlineIcon": "2BBH/2BBH\_E.png",  "category": "NOT\_SHOW",  "platform": "HuaweiEco",  "DeviceName": "华为智选 海雀智能摄像头Pro 32GB",  "DeviceTypeId": "008",  "desc": "",  "status": 0,  "uiType": "PLUGIN",  "language": "zh-CN,en-UK",  "isSupportSoftAP": 1,  "deviceSsid": "Hi- Alcidae-WiFiCamera-12BBHX",  "devicePin": 1,  "template": "semiModal"  }  profile数据如：  {  "prodId": "101C",  "deviceModel": "Reading lamp-CW",  "deviceTypeId": "046",  "deviceTypeName": "书写台灯",  "deviceTypeNameEn": "Reading lamp",  "deviceName": "欧普读写台灯",  "deviceNameEn": "Reading lamp",  "manufacturerId": "013",  "manufacturerName": "欧普照明",  "manufacturerNameEn": "OPPLE",  "description": "",  "protocolType": "WiFi",  "uiType": "H5",  "uriInfo": {  "uri":"101C/h5\_001/index.html"  },  "voiceControl": 1,  "services": [{  "serviceId": "light",  "serviceType": "light",  "characteristics": [{  "characteristicName": "brightness",  "characteristicType": "int",  "min": 0,  "max": 255,  "method": "RW"  }]  }]  } |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：修改设备属性

##### 接口定义

| 接口定义 |
| --- |
| DeviceManager.modifyDeviceProperty(deviceId, serviceId, keyValue, callback) |

##### 接口描述

用于修改设备的属性，需要指定待修改的设备、待修改的服务标识、以及特征取值。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| deviceId | String | M | 设备标识。获取方式见3.1.1.1。  最大长度40个字符。 |
| serviceId | String | M | 设备服务标识。  参见设备的profile定义，如：  "light"。  最大长度128个字符。 |
| keyValue | String | O | 服务属性及取值的键值对。  参见设备的profile定义，如：  {"mode" : 2}。  最大长度1024个字符。 |
| callback | function | O | 回调函数。 |

##### 返回值说明

方法无返回值，结果由callback返回：

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| errorCode | int | M | 错误码：  0：成功  -1：失败 |
| message | String | M | 错误原因简要描述。 |
| data | String | O | 回调成功的情况下，即errorCode等于0时，data为返回的数据结果，格式为json字符串，以单品返回数据为准。 |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：订阅事件上报

##### 接口定义

| 接口定义 |
| --- |
| DeviceManager.registerEventCallback(eventTypes, callbackFunc) |

##### 接口描述

订阅事件上报，当订阅的事件触发时，则会通过回调接口通知订阅者。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| eventTypes | String | M | 由一个或多个订阅事件组成的数组，格式为json字符串。  订阅的事件类型，如下：  "deviceAdded":添加新设备通知  "deviceDeleted":删除设备通知  "deviceNameUpdated":设备名称修改  "deviceMoved":设备位置更新  "deviceDataChanged":设备数据变化 |
| callback | function | M | 回调函数。 |

##### 返回值说明

方法无返回值，结果由callback返回：

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| eventType | String | M | 事件类型，如下：  "deviceAdded":添加新设备通知  "deviceDeleted":删除设备通知  "deviceNameUpdated":设备名称修改  "deviceMoved":设备位置更新  "deviceDataChanged":设备数据变化 |
| data | String | O | data为返回的数据结果，格式为json字符串。 |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：取消订阅事件上报

##### 接口定义

| 接口定义 |
| --- |
| DeviceManager.unregisterEventCallback(eventTypes) |

##### 接口描述

取消订阅事件上报，取消订阅后，不再上报取消的事件。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| eventTypes | String | M | 由一个或多个订阅事件组成的数组，格式为json字符串。  订阅的事件类型，如下：  "deviceAdded":添加新设备通知  "deviceDeleted":删除设备通知  "deviceNameUpdated":设备名称修改  "deviceMoved":设备位置更新  "deviceDataChanged":设备数据变化 |

##### 返回值说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| NA | NA | NA | NA |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

## 设备添加

### 接口说明

#### 接口：是否支持WifiAware

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.isSupportWifiAware() |

##### 接口描述

用于判断手机是否支持WifiAware

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| NA | NA | NA | NA |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（支持）/ -1（不支持） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：通过NAN发现设备

##### 接口定义

| 接口定义 |
| --- |
| NetConfig. discoveryByNAN (scanNanInfo, callbackFunc) |

##### 接口描述

通过NAN广播服务发现设备

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| scanNanInfo | Object | M | NAN扫描信息：  scanNanInfo {  int duration; 在指定的duration时间内未建立起连接，停止广播，单位为秒，范围[0,100],0表示不限制时长。  int lockTime; 设备收到NAN互信消息后在NAN模式的锁定时长，单位为秒，范围[0,100],0表示不限制时长；在碰一碰交互机制下，该字段功能暂不生效。  String sessionId; 值为service拉起FA时的Intent携带的sessionId值，如果FA启动时intent没有携带sessionId，则设置为“”。  } |
| callbackFunc | function | M | 回调函数，回调成功的情况下，data为返回数据结果，在该回调函数中解析使用数据结果。  返回：DeviceInfo{  String productId,  String sn,  String sessionId  } |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：通过SoftAP发现设备

##### 接口定义

| 接口定义 |
| --- |
| NetConfig. discoveryBySoftAp (callbackFunc) |

##### 接口描述

扫描获取AP信息

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| callbackFunc | function | M | 回调函数，回调成功的情况下，data为返回数据结果，在该回调函数中解析使用数据结果。  返回：List<SoftAPInfo>  SoftAPInfo{  String ssid,  boolean usePassword 是否需要使用密码  } |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：通过蓝牙发现设备

##### 接口定义

| 接口定义 |
| --- |
| NetConfig. discoveryByBle (scanInfo, callbackFunc) |

##### 接口描述

扫描获取蓝牙信息

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| scanInfo | Object | M | 扫描信息：  scanInfo {  int duration; 在指定的duration时间内未建立起连接，停止扫描，单位为秒，范围（0, 20]。  } |
| callbackFunc | function | M | 回调函数，发现设备会立即返回，会回调多次，和回调成功的情况下，data为返回数据结果，在该回调函数中解析使用数据结果。  返回：  commonInfo{  String sessionId，  BleInfo bleInfo  }  BleInfo{  String deviceMac,  String deviceName 设备名称  List<String> manufacturerData 经Base64编码后的蓝牙广播manufacturer数据  String rawBytes 经Base64编码后的蓝牙广播rawBytes数据  } |
|  |  |  |  |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：停止蓝牙发现设备

##### 接口定义

| 接口定义 |
| --- |
| NetConfig. stopDiscoveryByBle (stopInfo) |

##### 接口描述

不能频繁调用开始和停止，避免手机系统接口抑制（30秒5次）

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| stopInfo | Object | M | 停止扫描  stopInfo{  String sessionId；为discoveryByBle返回的sessionId。  } |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
|  |  |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：连接设备

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.connectDevice(connectInfo, callbackFunc) |

##### 接口描述

连接设备

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| connectInfo | Object | M | connectInfo {  string targetDeviceId; 待连接设备的标识，BLE模式下为设备mac地址。  number type; 配网类型0（NAN）/ 1（SoftAp）/ 2 (BLE)。  boolean isNeedBond; BLE模式有效，标识是否需要绑定设备，绑定设备采用justwork方式。  string pin; 设备pin。  string password; 设备密码，若设备需要密码，则设置为所需密码；否则设置为“”。  string sessionId; 会话ID，若type为NAN或BLE，则设置为discoverDevByNAN或discoveryByBle返回的sessionId；否则设置为“”。  } |
| callbackFunc | function | M | 回调函数，回调成功的情况下，data为返回数据结果，在该回调函数中解析使用数据结果。  返回：String vendorData，  Softap或BLE模式下返回  {  String productId，  String sn，  String vendorData  }的JSON格式字符串；NAN模式下不返回数据。 |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：获取wifi列表

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.getWifiList(callbackFunc) |

##### 接口描述

获取周边wifi列表

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| callbackFunc | function | M | 回调函数，回调成功的情况下，data为返回数据结果，在该回调函数中解析使用数据结果。  返回：List<WifiApInfo>  WifiApInfo {  String ssid;  boolean hasDefaultPassword;  String securityType;  int signalLevel;  int channel;  boolean is5G;  String wifiApId;  }  2.4G和5G都返回，FA只需通过is5G是否支持5G进行筛选过滤；  周边范围内的  已配置过，并且能够成功获取密码的ssid，hasDefaultPassword才为true； |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：对指定的设备进行配网

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.configDeviceNet(deviceInfo,accountInfo,netConfigInfo,callbackFunc) |

##### 接口描述

对指定的设备进行配网，发送SSID和密码，以及设备信息等

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| deviceInfo | string | M | 设备云为设备分配的信息，如deviceID等 |
| accountInfo | string | M | 账号信息 |
| netConfigInfo | Object | M | netConfigInfo {  string ssid; 所配目标路由的SSID；  string ssidPassword; 不需要下发密码时，填写“”；  boolean isDefaultPassword;  int channel;  string sessionId; 若配网模式为NAN或BLE时填discoverDevByNAN或discoveryByBle返回的sessionId；配网模式为SoftAp时填“”；  int type; 配网模式，0（NAN）/ 1（SoftAp）/ 2（BLE）；  String wifiApId; Wifi列表返回信息中的对应字段；  String vendorData; 产品数据  int timeout; 配网超时时间，在NAN模式下设置，范围[1,90]s；  }  说明:如果用户输入密码，则isDefaultPassword为false, ssidPassword为“输入的密码”；否则按使用默认密码处理，isDefaultPassword为true, ssidPassword为“”。 |
| callbackFunc | function | M | 回调函数，NAN配网模式中，回调成功表示模组上报配网状态结果，其结果会转换成整型数值放到回调函数入参data的data.code，字符串类型会放到data.msg; SoftAp/BLE配网中，返回0/ -1表示配网成功/失败； |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：秒控接口

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.sendMessage(commonInfo, message, callbackFunc) |

##### 接口描述

秒控接口，通过该接口发送秒控信息，在NAN模式或者BLE模式下可使用，SoftAp模式下不支持。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| commonInfo | Object | M | commonInfo{  String sessionId; 若配网模式为NAN或BLE，则设置为discoverDevByNAN或discoveryByBle返回的sessionId。  int type; 配网模式，0（NAN）/ 1（SoftAp，暂无softAp秒控，占位使用）/ 2（BLE）；可选，不带是NAN  } |
| message | String | M | 消息内容，最大长度建议不超过512。 |
| callbackFunc | function | M | 回调函数，返回命令发送结果。  返回：0（成功）/ -1（失败） |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：释放配网通道

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.disconnectDevice(commonInfo, callbackFunc) |

##### 接口描述

释放配网通道，在FA不再使用NAN通道、FA退出、FA切后台等情况需要FA主动调用该接口。

在SoftAp模式下，当配网结束后需要调用此接口以断开与设备的连接。

在BLE模式下，当配网结束后需要调用此接口以断开设备的连接。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| commonInfo | Object | M | commonInfo{  String sessionId; 若配网模式为NAN或BLE，则设置为discoverDevByNAN或discoveryByBle返回的sessionId；否则设置为“”。  int type; 配网模式，0（NAN）/ 1（SoftAp，暂无softAp秒控，占位使用）/ 2（BLE）；可选，不带是NAN  } |
| callbackFunc | function | M | 回调函数，返回数据结果，在该回调函数中解析使用数据结果。  返回：0（成功）/ -1（失败） |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：接收设备侧消息

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.registerMsgReceive(commonInfo, callbackFunc) |

##### 接口描述

获取设备侧消息，在NAN配网模式下连接设备接口前调用。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| commonInfo | Object | M | commonInfo{  String sessionId; 若配网模式为NAN或BLE，则设置为discoverDevByNAN或discoveryByBle返回的sessionId。  int type; 配网模式，0（NAN）/ 1（SoftAp，暂无softAp秒控，占位使用）/ 2（BLE）；可选，不带是NAN  } |
| callbackFunc | function | M | 回调函数，回调成功的情况下，data为返回数据结果，在该回调函数中解析使用数据结果。  返回：String message |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：设备断链通知回调消息

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.registerDisconnectCallback (commonInfo, callbackFunc) |

##### 接口描述

注册连接断开事件回调，在NAN或者BLE配网模式下连接设备接口前调用。当因某些原因造成设备主动与手机断开连接时，会通过该回调通知。当调用3.1.9释放配网通道接口主动断开与设备的连接时，此断连回调函数不会执行。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| commonInfo | Object | M | commonInfo{  String sessionId; 若配网模式为NAN或BLE，则设置为discoverDevByNAN或discoveryByBle返回的sessionId。  int type; 配网模式，0（NAN）/ 1（SoftAp，暂无softAp秒控，占位使用）/ 2（BLE）；可选，不带是NAN  } |
| callbackFunc | function | M | 回调函数，当收到该回调时表示设备与手机连接断开。  返回状态码：0。 |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：设备绑定关系设定接口（废弃，需要使用3.2.1.14接口替代此接口）

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.setDeviceBindInfo(bindInfo, callbackFunc) |

##### 接口描述

设置设备绑定信息，设备配网结束成功注册到自有云后调用该接口可将绑定信息设置到华为云。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| bindInfo | Object | M | bindInfo{  String nfcTagId; 设备标签ID，为HexString格式  String deviceId; 设备ID  } |
| callbackFunc | function | M | 回调函数，返回状态码表示设定结果。  返回状态码：0 设定成功 / -1 设定失败。 |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：设备绑定关系设定接口（代替setDeviceBindInfo接口）

##### 接口定义

| 接口定义 |
| --- |
| NetConfig. setDeviceBindInfoWithThirdDevId (bindInfo, callbackFunc) |

##### 接口描述

设置设备绑定信息，设备配网结束成功注册到自有云后调用该接口可将绑定信息设置到华为云。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| bindInfo | Object | M | bindInfo{  String nfcTagId; 设备标签ID，为HexString格式  String deviceId; 设备ID  String thirdPartyId; 设备ID  String productId; 设备产品ID  DeviceTokenKey deviceToken 需要与kitFramework使用的保持一致  }  DeviceTokenKey {  String manufacture, 厂商信息  String productModel, 产品信号  String serial 序列号  } |
| callbackFunc | function | M | 回调函数，返回状态码表示设定结果。  返回状态码：0 设定成功 / -1 设定失败。 |

##### 返回值说明

| 参数类型 | 描述 |
| --- | --- |
| int | 0（执行成功）/ -1（执行失败） |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：开启协商

##### 接口定义

| 接口定义 |
| --- |
| NetConfig.startNegotiate (negotiateInfo, callbackFunc) |

##### 接口描述

开启协商

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| negotiateInfo | Object | M | negotiateInfo {  int infoType; 信息类型，调用该接口时设置为0。  string pin; 设备pin。  string flag; 设备标识，建议使用设备sn/mac。  string data; 调用该接口时，该字段传“”。  } |
| callbackFunc | function | M | 回调函数。  其中code值为：  -1协商失败；  0 协商成功，此时 data为sessionId，即安全会话ID，在调用wifi加密接口时需要传入。  2 协商中，此时data为协商消息，需要发送给对端。 |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：处理协商过程中消息

##### 接口定义

| 接口定义 |
| --- |
| NetConfig. processNegotiateData (negotiateInfo) |

##### 接口描述

处理协商过程中的消息，即收到对端的协商消息后调用该接口。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| negotiateInfo | Object | M | negotiateInfo {  int infoType; 信息类型，调用该接口时设置为1。  string pin; 调用该接口时，该字段设置为“”。  string flag; 设备标识，建议使用设备sn/mac。  string data; 协商消息，即对端发来的协商消息。  } |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

#### 接口：wifi信息加密

##### 接口定义

| 接口定义 |
| --- |
| NetConfig. wifiDataEncrypt (commonInfo, callbackFunc) |

##### 接口描述

对当前连接的Wi-Fi或用户输入的Wi-Fi信息进行加密，返回加密后的数据。调用成功后，秘钥失效。

##### 参数说明

| 参数 | 类型 | M/O | 描述 |
| --- | --- | --- | --- |
| wifiEncryptInfo | Object | M | wifiEncryptInfo {  String flag; 设备标识  String sessionId；安全会话ID，由协商成功后返回  String ssid；所配目标路由的SSID  String pwd；不需要下发密码时，填写“”  boolean isDefaultPassword;是否使用默认密码  }  说明:如果用户输入密码，则isDefaultPassword为false, pwd为“输入的密码”；若使用默认密码，isDefaultPassword为true, pwd为“”。 |
| callbackFunc | function | M | 回调函数，返回数据结果。  返回：0（成功）/ -1（失败）  成功时，data为加密后的数据byte[]经过base64编码后的字符串。 |

##### 异常说明

| 异常类型 | 描述 |
| --- | --- |
| NA | NA |

##### 补充：

回调成功时，data为加密后的数据byte[]经过base64编码后的字符串，若要获取byte[]格式需要对data进行base64解码。方法可参考：

通过npm安装三方base64-js，在FA工程entry/node\_modules/目录下执行：

npm install base64-js

安装完成后，在js/app.js文件中引入即可：

var netConfig = require('fa-netconfig');

**var base64js = require('base64-js')**

export default {

onCreate() {

console.info('AceApplication onCreate');

},

onDestroy() {

console.info('AceApplication onDestroy');

},

NetConfig: netConfig.FANetConfig,

**base64: base64js**

};

使用：

this.$app.$def.base64.toByteArray()