

# UNISOC Power Hint User Guide

适用产品信息	T7510 , SC7731E , SC9832E , SC9863A , SC9820E
适用版本信息	Android 7.0/8.0/9.0
关键字	power hint

Unisoc Confidential For hiar

## 声明

本文件所含数据和信息都属于紫光展锐所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

## 版本历史

版本	日期	备注
V1.0	2019/10/22	初稿

Unisoc Confidential For hiar

## 目录

<b>1 前言</b>	<b>1</b>
1.1 范围	1
1.2 缩略语	1
<b>2 Introduce</b>	<b>2</b>
<b>3 Overview</b>	<b>3</b>
<b>4 API Description</b>	<b>5</b>
4.1 powerHint() API	5
4.1.1 Java Layer API	5
4.1.2 Native Layer API	5
<b>5 Power HIDL HAL</b>	<b>6</b>
5.1.1 Java Layer API	6
5.1.2 Native Layer API	6
5.2 Sample	7
5.2.1 powerHint() sample	7
5.2.2 PowerHALManager(Power HIDL HAL)	8
<b>6 Selinux policy</b>	<b>10</b>
<b>7 Power Hint config file introduction</b>	<b>11</b>
7.1 Resource define file	11
7.2 Scene ID and name file	13
7.3 Scene config file	14
<b>8 How to support new sence</b>	<b>15</b>

# 1 前言

## 1.1 范围

本文档主要介绍了在 Andorid 系统中 PowerHint 模块使用及相关场景配置。

## 1.2 缩略语

此章节请列出此文档所涉及的缩略词。

名称	全称	定义
API	应用程序编程接口	是一些预先定义的函数，或指软件系统不同组成部分衔接的约定。
APK	Android 应用程序包	是 Android 操作系统使用的一种应用程序包文件格式
HAL	硬件抽象层	硬件抽象层是位于操作系统 内核与硬件电路之间的接口层，其目的在于将硬件抽象化。
HIDL	HAL 接口定义语言	是用于指定 HAL 和其用户之间的接口的一种接口描述语言 (IDL)
APP	( 应 用 程 序 ， Application 的缩写 ) 一般指手机软件	手机软件，主要指安装在智能手机上的软件。

## 2 Introduce

在进入某个场景或是在某些条件下有需要调整系统性能时，通过相关接口通知 PowerHint 进行相应处理，如调整 DVFS、DFS 等，以达到提升性能或省电目的，提升用户体验，如启动 APP 时，ActivityManagerService 通过 PowerHint 触发系统提升 CPU、DDR 频率，加速 APP 的启动。后面章节将介绍架构和 PowerHint 相关接口。

Unisoc Confidential For hiar

### 3 Overview

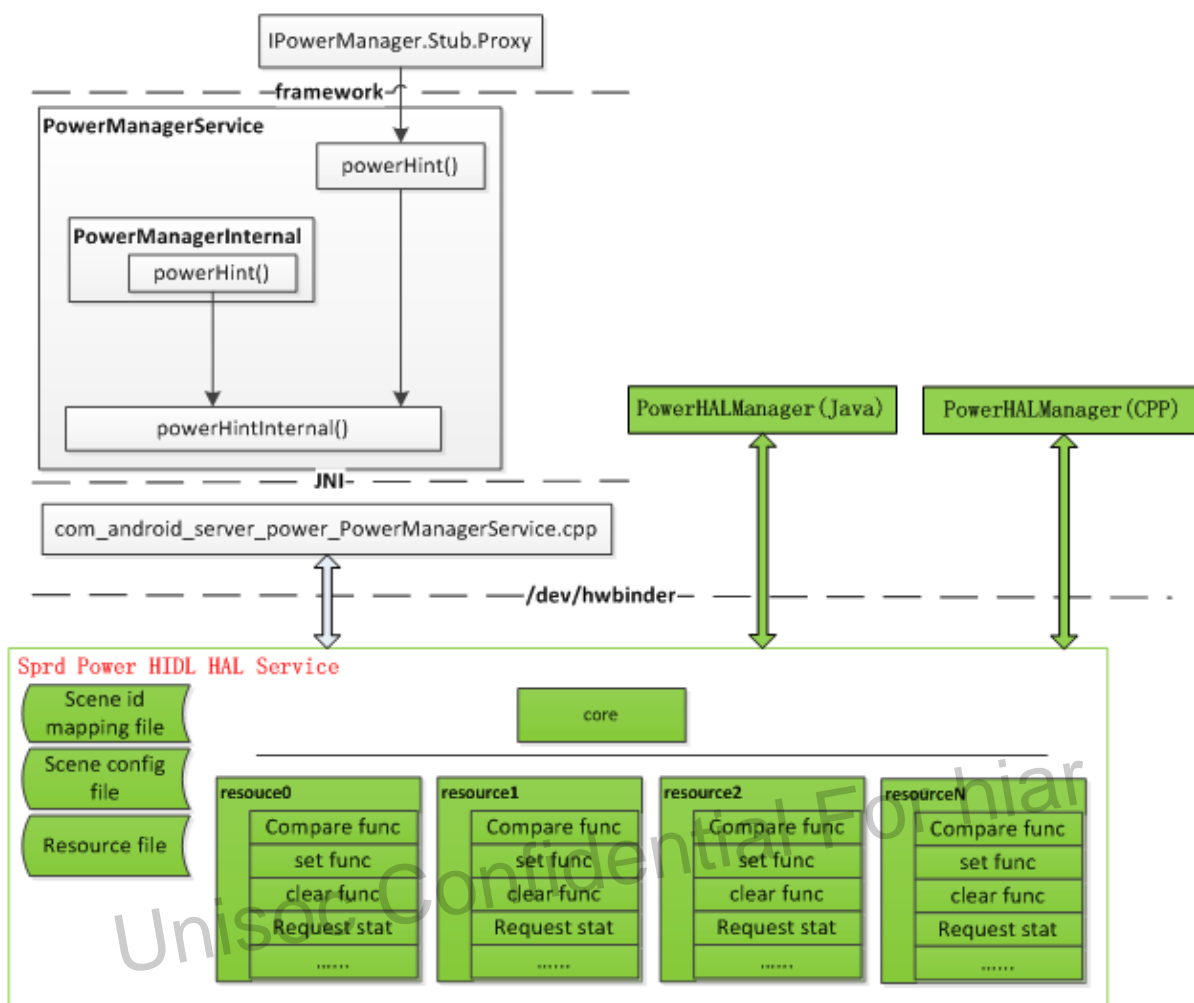


图 3.1 PowerHint 架构

图 3.1 为 PowerHint 整体架构 图中绿色部分为 Sprd 实现和接口封装 如接口类 `PowerHALManager` ;

`PowerManagerInternal::powerHint()` 为 Google 源生接口 .

触发 PowerHint 有三个接口:

- 通过 `PowerManagerService` 的 Binder 接口 `powerHint()` ;
- 通过 `PowerHALManager` , 即 Power HIDL HAL 接口 ;
- 直接通过 Power HIDL HAL , **Deprecated** ;

**Note:** `PowerManagerInternal` 为 `system_server` 中的一个 Local service , 故仅 `system_server` 进程中

可通过 `PowerManagerInternal` 触发 PowerHint .

每个场景对应一个 ID，Android 源生场景 ID 定义为 POWER\_HINT\_XXX，对应场景使用 `PowerManagerInternal::powerHint()` 接口进行 PowerHint；

Sprd 扩展场景使用 `PowerHALManager` 接口，framework 和 Native 层场景 ID 分别定义在 `PowerHintVendorSprd.java` 和 `types.hal` 中，如：

- `POWER_HINT_VENDOR_CAMERA_LOW_POWER(java)`
- `VENDOR_CAMERA_LOW_POWER(cpp)`

Power Hint 所支持的资源和各场景的配置均通过配置文件指定；配置针对产品可定制；

Unisoc Confidential For hiar



## 4 API Description

### 4.1 powerHint() API

此 API 为 Android 原生 API，我们扩展其使用方式。

#### 4.1.1 Java Layer API

```
PowerManagerInternal.java  
public abstract void powerHint(int hintId, int data);
```

@hintId: 场景 ID，可使用 PowerHintVendorSprd.java 中场景 ID；

@data: 1 使能、0 禁止或 timeout；

**注意: Apk 中不会直接使用 PowerHint 相关接口，仅在相关服务中使用。**

#### 4.1.2 Native Layer API

```
frameworks/native/services/powermanager/IPowerManager.cpp::BpPowerManager  
virtual status_t powerHint(int hintId, int param)
```

@hintId: 场景 ID，可使用 types.hal 中场景 ID；

@param: 1 使能、0 禁止或 timeout；

Unisoc Confidential For hiar

## 5 Power HIDL HAL

通过 Power HIDL HAL 触发 Power Hint。

### 5.1.1 Java Layer API

```
Class PowerHALManager:
    public PowerHintScene createPowerHintScene(String tag, int sceneld, String sceneName)

Class PowerHALManager.PowerHintScene :
    public void acquire()
    public void acquire(long duration)
    public void release()
```

**@sceneld:** 场景 ID，定义在 PowerHintVendorSprd.java 中；

**@sceneName:** 场景名字，定义在 power\_scene\_id\_define.txt 中；

**@tag:** 调用者的标识，用于记录调用者信息；

**@duration:** 场景持续时间；

**PS:** sceneld 和 sceneName 指定一个即可，另一个指定为 0 或 NULL。

**使用说明：**

1. 其它模块通过 PowerHALManager 中 createPowerHintScene() 获取 PowerHintScene ；
2. 通过 PowerHintScene 中 acquire() 进行 Power Hint ；当指定 duration 时，持续指定时间后自动停止 Power Hint ；如果不指定 duration，调用者**必须**显示调用 release() 停止 Power Hint 。
3. release() 释放 Power Hint 请求；

### 5.1.2 Native Layer API

```
// power_hal_cli.h
- PowerHALManager:
    void init();
    void deinit();
    PowerHintScene* createPowerHintScene(string tag, int sceneld, string sceneName);
- PowerHintScene:
    void acquire();
    void release();
```

**@sceneld:** 场景 ID (VENDOR\_XXX)，定义在 Power HIDL HAL types.hal 中；

@sceneName: 场景名字，定义在 power\_scene\_id\_define.txt 中；

@tag：调用者的标识，用于记录调用者信息；

**PS:** sceneld 和 sceneName 指定一个即可，另一个指定为 0 或 nullptr.

### 使用说明：

1. 通过 PowerHALManager 对象调用 init()完成初始化操作；
2. 其它模块通过 PowerHALManager 中 createPowerHintScene()获取 PowerHintScene ；
3. 通过 PowerHintScene 中 acquire()进行 Power Hint ；调用者**必须**显示调用 release()停止 Power Hint .
4. release()释放 Power Hint 请求；
5. 通过 PowerHALManager 对象调用 deinit()完成资源回收操作；

## 5.2 Sample

### 5.2.1 powerHint() sample

#### 5.2.1.1 Launch App (Java layer sample)

需求：冷启动(new activity)时触发 Power Hint 以提升 app 启动速度

code 的路径: services/core/java/com/android/server/am/ActivityStackSupervisor.java

```
import android.hardware.power.V1_0.PowerHint;

mPowerManagerInternal = LocalServices.getService(PowerManagerInternal.class);
mPowerManagerInternal.powerHint(PowerHint.LAUNCH, 0/1);
```

#### 5.2.1.2 Camera hdr mode first frame(Native layer sample)

```
1) Android.mk 中引用 libpowermanager.so
2) code
#include <binder/IServiceManager.h>
#include <powermanager/IPowerManager.h>
#include <powermanager/PowerManager.h>

sp<IPowerManager> mPowerManager;
const String16 serviceName("power");
sp<IBinder> bs = defaultServiceManager()->checkService(serviceName);
if (bs == NULL) {
```

```

        return NAME_NOT_FOUND;
    }
    mPowerManager = interface_cast<IPowerManager>(bs);

    mPowerManager->powerHint(POWER_HINT_ENCODE, enabled ? 1:0);

```

## 5.2.2 PowerHALManager(Power HIDL HAL)

### 5.2.2.1 Java layer sample

#### 1. Android.mk

```

LOCAL_STATIC_JAVA_LIBRARIES += \
    sprd.halmanager

```

#### 2. Code

```

import android.os.PowerHALManager;
import android.os.PowerHintVendorSprd;

private PowerHALManager mPowerHALManager;
private PowerHALManager.PowerHintScene mSceneXxx;

mPowerHALManager = new PowerHALManager(mContext, mHandlerXxx);
mSceneXxx = mPowerHALManager.createPowerHintScene("camera", 0,
    "camera_low_power");
Or
sceneXxx = mPowerHALManager.createPowerHintScene("camera",
    PowerHintVendorSprd.POWER_HINT_VENDOR_CAMERA_LOW_POWER, null);

mSceneXxx.acquire();
mSceneXxx.release();
or
mSceneXxx.acquire(500);

```

**PS: 高亮标记根据具体场景进行替换。**

### 5.2.2.2 Native layer sample

#### 1. Android.mk

```

LOCAL_SHARED_LIBRARIES += \
    libpowerhal_cli

```

#### 2. C++ sample code

```

#include <power_hal_cli.h>

::android::sp<::android::PowerHALManager> pm_ =
    new ::android::PowerHALManager();

pm_->init();

::android::sp<::android::PowerHintScene> sceneXxx =
    powerManager->createPowerHintScene(LOG_TAG,

```

```
static_cast<int>(PowerHintVendor::LAUNCH), "");  
Or  
::android::sp<::android::PowerHintScene> sceneXxx =  
    powerManager->createPowerHintScene(LOG_TAG, 0, "launch");  
  
sceneXxx->acquire();  
sceneXxx->release();  
  
pm_->deinit();
```

**PS: 高亮标记根据具体场景进行替换。**

Unisoc Confidential For hiar

## 6 Selinux policy

caller 如果遇到服务访问问题,命令行执行 `setenforce 0`,看下执行状态(在调用接口时要判断服务是否为 null),如果可以正常执行说明是 selinux 问题. 如果为 Selinux 问题, 在对应模块的 TE 文件中添加 `binder_call(xxx, hal_power_default)`, xxx 为调用者所在的 domain,即 scontext。

Unisoc Confidential For hiar

## 7 Power Hint config file introduction

资源、场景配置和场景 I D 的定义通过配置完成：

- **power\_scene\_id\_define.txt**：场景 I D 和名字映射文件；
- **power\_resource\_file\_info.xml**：定义所支持的资源；
- **power\_scene\_config.xml**：定义所支持的模式、每个模式所支持的场景和每个场景对哪些资源进行配置；

Android7.0 配置文件在/system/etc/下，Android8.0 及以后版本配置文件在/vendor/etc/下，修改配置文件后需重启。

下面将以 Android8.0 配置文件为例说明如何使用配置文件。

### 7.1 Resource define file

格式如下：

```
<resources>
  <file path="" file="" no_has_def="" >
    <attr name="comp_func" value="" />
    <attr name="set_func" value="" />
    <attr name="clear_func" value="" />
    <attr name="def_value" value="" />
  </file>
  <file path="subsys" file="xxx" >
    <attr name="comp_func" value="" />
    <attr name="set_func" value="" />
    <attr name="clear_func" value="" />
    <attr name="def_value" value="" />
  </file>
  <subsys name="xxx" >
    <inode path="" file="" def_value="" no_has_def="" />
    <inode path="" file="" def_value="" />
    <conf name="conf_xxx" priority="" >
      <set path=xxxx file=xxx value=xxx />
      <set path=xxxx file=xxx value=xxx />
    </conf>
    .....
  </subsys>
```

```
</resources>
```

Sample:

```
<resources>
  <file path="/sys/devices/system/cpu/cpuhotplug" file="cluster0_core_min_limit" >
    <attr name="comp_func" value="common_comp_ascend_order" />
    <attr name="set_func" value="common_set" />
    <attr name="clear_func" value="common_clear" />
    <attr name="def_value" value="1" />
  </file>
  <file path="/sys/class/devfreq/scene-frequency/sprd_governor" file="scene_boost_dfs" no_has_def="1" >
    <attr name="comp_func" value="common_comp_ascend_order" />
    <attr name="set_func" value="devfreq_ddr_set" />
    <attr name="clear_func" value="devfreq_ddr_clear" />
  </file>
  <file path="subsys" file="cpufreq" >
    <attr name="comp_func" value="common_subsys_comp" />
    <attr name="set_func" value="common_subsys_set" />
    <attr name="clear_func" value="common_subsys_clear" />
    <attr name="def_value" value="conf_3" />
  </file>
  <subsys name="cpufreq" >
    <inode path="/sys/devices/system/cpu/cpu0/cpufreq/interactive" file="hispeed_freq" />
    <inode path="/sys/devices/system/cpu/cpu0/cpufreq/interactive" file="target_loads" />
    <inode path="/sys/devices/system/cpu/cpu0/cpufreq/interactive" file="timer_rate" />
    <inode path="/sys/devices/system/cpu/cpu0/cpufreq/interactive" file="boost" def_value="0" />
    <conf name="conf_1" priority="1" >
      <set path="/sys/devices/system/cpu/cpu0/cpufreq/interactive" file="hispeed_freq" value="768000" />
      <set path="/sys/devices/system/cpu/cpu0/cpufreq/interactive" file="target_loads" value="90 768000:200" />
    </conf>
    <conf name="conf_2" >
      <set path="/sys/devices/system/cpu/cpu0/cpufreq/interactive" file="timer_rate" value="60000" />
    </conf>
    <conf name="conf_3" >
    </conf>
    <conf name="conf_4" >
      <set path="/sys/devices/system/cpu/cpu0/cpufreq/interactive" file="boost" value="1" />
    </conf>
  </subsys>
</resources>
```

图 7.1 资源定义文件

此文件定义资源并指定相关信息：

- **file**：定义一个资源，可以将一个文件或一个子系统定义为一个资源。通过 attr 子节点指定其比较函数、设置函数、请求清理函数和默认值，比较函数、配置函数、请求清理函数必须指定，如果一个子系统最为一个资源必须指定其默认值。

- **Path**：一个文件代表一个资源时，为文件所在的目录；一个子系统代表一个资源时，为 subsys；
- **File**：文件或子系统的名字；
- **No\_has\_def**：指定此资源是否有默认值，可以省略，默认为 0，有默认值。如果有默认值并没有通过 attr 节点显示指定时，系统会自动获取其默认值；

Attr 节点：

- **Comp\_func**：指定请求优先级比较函数。如果文件节点的值越大代表优先级越高，使用 common\_comp\_ascend\_order()；如果文件节点的值越小代表优先级越高，使用 common\_comp\_descend\_order()；如果子系统代表一个资源，使用 common\_subsys\_comp()。
- **Set\_func**：指定资源的设置函数。仅是设置指定的 value 使用 common\_set()即可；子系统代



表一个资源时，使用 `common_subsys_set()`；特殊的资源需单独定义 `set` 函数如 `dfs`。

- **Clear\_func**：指定资源的请求清理函数，仅恢复其默认值使用 `common_clear()`；子系统代表一个资源时，使用 `common_subsys_clear`；特殊的资源需单独定义 `clear` 函数如 `dfs`。

`common.c` 中 `compare_funcs[]`、`clear_funcs[]`和 `set_funcs[]`声明所支持的函数。

➤ **Subsys**：定义子系统，指定此系统包含哪些文件和所支持的配置及其优先级。

- **Inode**：指定子系统包含的文件和文件相关信息，如：是否有默认值或其默认值。
  - **Path**：文件所在目录；
  - **File**：文件名；
  - **Def\_value**：指定此文件的默认值，可省略，省略时，系统会自动获取其默认值；
- **Conf**：定义子系统所支持的配置和配置的优先级。
  - **Name**：配置的名称，唯一并定义后不可改变（不包含优先级信息）；
  - **Priority**：指定配置的优先级，优先级从 1 开始，如果没有显示指定其优先级，逐个加 1，类似 `enum`；
  - **set**：指定配置的具体配置信息，如果 `inode` 所指定的文件没有被显示指定，使用其默认值。

## 7.2 Scene ID and name file

Sample:

```
#id      #sub_id  #scene name
0x00000001 0x00000000 vsync
0x00000002 0x00000000 interaction_other
0x00000003 0x00000000 video_encode
0x00000004 0x00000000 video_decode
0x00000005 0x00000000 low_power
0x00000006 0x00000000 sustained_performance
0x00000007 0x00000000 vr_mode
```

图 7.2 场景名称与 ID 文件

第一列为场景 ID，第二列为子类型(后面将不再使用 subtype)，第三列为场景名。

添加新场景需要修改此文件，并在 `power types.hal` 和 `PowerHALManager.java` 定义其 ID。

## 7.3 Scene config file

格式如下：

```
<power>
  <mode name=xxx>
    <scene name=xxx enable=yyy duration=zzz >
      <set path=xxxx file=xxx value=xxx />
      <set path=xxxx file=xxx value=xxx />
    </scene>
  </mode>
</power>
```

Sample:

```
<power>
  <mode name="normal">
    <scene name="interaction_touch" >
      <set path="subsys" file="cpufreq" value="conf_4" />
    </scene>
    <scene name="interaction_launch" >
      <set path="/sys/devices/system/cpu/cpufreq" file="cluster0_core_min_limit" value="4" />
      <set path="subsys" file="cpufreq" value="conf_4" />
      <set path="/sys/class/devfreq/scene-frequency/sprd_governor" file="scene_boost_dfs" value="max" />
    </scene>
    <scene name="interaction_fling" >
      <set path="subsys" file="cpufreq" value="conf_4" />
    </scene>
    <scene name="interaction_other" >
      <set path="subsys" file="cpufreq" value="conf_4" />
    </scene>
    <scene name="ddr" >
      <set path="/sys/class/devfreq/scene-frequency/sprd_governor" file="scene_boost_dfs" value="max" />
    </scene>
    <scene name="video_sync" >
      <set path="/sys/devices/system/cpu/cpufreq" file="cluster0_core_min_limit" value="2" />
    </scene>
    <scene name="screenof_mp3_playback" >
      <set path="subsys" file="cpufreq" value="conf_2" />
    </scene>
  </mode>
</power>
```

图 7.3 场景配置文件

- Mode：定义不同模式的场景配置。name 为模式的名字，场景 I D 定义文件定义所支持的模式名字，必须支持 normal 模式；
- Scene：包含此场景对每个资源的配置。name 为场景名，场景 I D 定义文件定义所支持的场景名字；enable 控制是否使能此场景，optional；duration 强行指定 boost 时间，optional；
- Set：指定对某个资源的配置。设置子系统所代表的资源时，path 为 subsys，file 为资源文件中所定义的子系统；

## 8 How to support new sence

- 1) 请求使用 PowerHint 的 RD 先在命令行测试配置，确认具体配置后告知 PowerHint FO；
- 2) PowerHint FO 提供场景 ID；
- 3) 需求请求 RD 参考 API 和 Sample code 章节调用 Power Hint 相关接口并添加 Selinux policy;
- 4) PowerHint FO 提供相应 patch，提出需求 RD 验证功能 ok 后，合入 patch；

针对“某个 APP 在前台运行”场景：

1. 提供 APP 包名，如 com.futuremark.pcmark.android.benchmark；
2. 在 power\_scene\_id\_define.txt 添加场景 ID，如 0x7fff0022 0x00000000  
com.futuremark.pcmark.android.benchmark；
3. power\_scene\_config.xml 中添加配置，如：

```
<scene name="com.futuremark.pcmark.android.benchmark" >
  <set path="/dev" file="cluster0_freq_max" value="124F80" />
  <set path="/dev" file="cluster0_freq_min" value="124F80" />
  <set path="/dev" file="cluster1_freq_max" value="17F4D0" />
  <set path="/dev" file="cluster1_freq_min" value="17F4D0" />
</scene>
```

**Note：**Power hint 仅负责确保配置的准确下发，场景配置验证不全面引入的问题，由提供配置的需求

**请求方负责。**