# 用户插件方案

## 背景

用户插件分为两种类型，一种是root权限下的超级开发者开发的用户插件，另一种是普通用户权限下的普通开发者则用户插件。超级开发者暂定义为公司内部开发插件使用。

系统级插件与用户插件的区别：

1. 系统插件一定以root方式运行，用户插件则可以使用root或用户权限运行。
2. 系统插件的配置存放在startup文件中，被mater进程集中管理。用户插件自行存放配置。
3. 系统插件不会接收事件，用户插件则可以，监听事件通知。
4. 系统插件不提供接口给外部进行插件安装，而用户插件必须提供接口对外。

## 用户插件功能点

#### 制作用户插件SD卡环境

* 1. **用户插件目录环境**

用户插件需要安装在持久存储区域中，默认整个环境安装在SD卡挂载的根目录下。

SD

|-Apps

|-sysevent.d

|-Appname

|-脚本文件

|-store\_super.d

|-Appname.script

|-Appname.info

|-store\_user.d

|-Appname.script

|-Appname.info

|-store\_profile.d

|-Appname

|-Appname1

|-etc

|-bin

|-lib

|-html

|-其他

|-AppnameX

Sysevent.d目录：每个插件的事件回调函数放在此目录下自己插件名的目录中。

Store\_super.d目录：以root权限运行的插件管理脚本与信息放在此目录中。

Store\_user.d目录：以用户权限运行的插件管理脚本与信息放在此目录中。

Store\_profile.d目录：插件运行信息保存，命名为插件名保存在此目录下。

文件格式如下：

*Name；Ename；安装路径；运行权限；认证权限*

Appname目录：存放用户插件运行环境。

* 1. **制作过程**
* 建立目录结构的时机有两个：

1. 将制作代码编译到开机脚本中，开机即判断运行。
2. 做成系统插件。（推荐使用）

* 用户插件环境在安装的时候生成。

#### 制作用户插件

* 1. **用户插件结构**

|-Appname.pem

|- Appname.info

|-Appname.script

|-bin文件

|-lib库

|-conf配置文件

|-icon

|-其他

Appname.pem：用户插件认证文件，以root权限运行本地安装插件，此文件可忽略。否则，需要依靠此文件才判断是否有安装权限与系统相关使用权限。

Appname.info：保留文件，存储用户插件信息，如名称、版本号、说明信息、图标文件路径等【详细格式可参见[Appname.info格式](#_Appname.info格式)】。

Appname.script：保留文件，提供给用户插件模块用于插件的安装、卸载、启动、停止、获取状态等操作【详细格式可参见[Appname.script格式](#_Appname.script格式)】。

bin文件：插件的执行代码，可以是C、shell、Lua等语言编写。

Lib库：可选，执行代码依赖的库文件。

Conf配置文件：执行代码依赖的配置文件。

Icon：图标文件，用于网页中插件图标显示，可存于本地或网络URL链接。

其他：其他类型文件。

* 1. **制作过程**

将上述整个插件内容打包成一个tar包即可，取名为Appname.cipk。

#### 插件安装

* 1. **安装流程**



1. UI界面从插件服务器获取可用插件列表。

每个插件至少要从服务器获取到下列信息：

* *插件名*
* *插件版本*
* *插件作者*
* *插件简介与说明*
* *插件图标*
* *插件文件URL下载地址*
* *插件文件MD5校验值*

可参考如下



1. 点击UI界面插件安装按钮。
2. 调用cpkgcli安装插件，传入参数为插件文件的URL路径与MD5值。cpkgcli为用户插件与外界的接口层，用户插件所有接口均通过此命令实现。
3. 下载插件安装包并校验其完整性。
4. 解压安装包，通过Appname.pem文件判断是否具安装权限，并判断是否已root权限运行插件。

6a、获得root权限安装插件

* 拷贝Appname.script Appname.info文件到store\_super.d目录下。
* 通过权限分析和Appname.info生成store\_profile.d下Appname文件。
* 创建Appname目录，运行Appname.script中的install环境安装插件。

6b、以用户权限安装插件

* 以Appname.info中的ename创建用户名
* 将插件解压包中的文件，设置成只有ename可操作权限。
* 拷贝Appname.script Appname.info文件到store\_user.d目录下。
* 通过权限分析和Appname.info生成store\_profile.d下Appname文件。
* 创建Appname目录，设置成只有ename可操作权限。
* 已ename权限运行Appname.script中的install环境安装插件。

7、返回安装结果给UI。

8、将插件显示在UI已经安装插件中。

* 1. **手动安装**

手动安装的情况，可直接调用cpkgcli进行插件安装，传入本地文件路径，可忽略MD5参数，忽略认证文件，直接以root权限运行插件，用于插件调试。

* 1. **。**

#### 插件卸载

* 1. **卸载流程**



1. 点击UI界面中卸载按钮。
2. 调用cpkgcli卸载插件，传入参数为插件ename名。
3. 从Appname.profile中获取插件权限与安装路径。

4a、以root权限卸载插件

* 运行Appname.script中uninstall卸载插件。
* 删除store\_profile.d下Appname文件。
* 删除store\_super.d目录下Appname.script Appname.info文件。
* 删除sysevent目录下Appname目录。
* 删除插件运行环境目录。

4b、以用户权限卸载插件

* 进入ename权限，运行Appname.script中uninstall卸载插件。
* 删除store\_profile.d下Appname文件。
* 删除store\_super.d目录下Appname.script Appname.info文件。
* 删除sysevent目录下Appname目录。
* 删除插件运行环境目录。
* 删除ename用户。

5、返回结果给UI界面。

6、在界面中删除卸载插件。

#### 插件运行

* 1. **开机启动流程**

1. 运行系统启动脚本，启动master进程之后。
2. 遍历store\_supter.d目录下脚本，以root权限运行Appname.script中的start函数。
3. 遍历store\_user.d目录下脚本，以ename权限运行Appname.script中的start函数。
   1. **手动运行插件**

1、调用cpkgcli命令，传入appname，可手动启动插件。

1. 从Appname.profile中获取插件权限。

3a、若为root权限，以root权限运行Appname.script中的start函数。

3b、若为用户权限，以用户权限运行Appname.script中的start函数

#### 监听系统事件

* 1. **订阅事件**
* 使用sysevent命令将脚本注册到需监听的事件中。

Sysevent命令格式

*Sysevent 【事件类型】 【处理脚本】*

* 拷贝【处理脚本】到/sysevent/【事件类型】目录中，命名为Appname文件。
  1. **事件分发**
* 当事件发生时，使用sysevent事件接口接收事件通知。
* 分析事件类型，将时间信息存储在环境中。
* 查找/sysevent/【事件类型】中挂载上的回调脚本。
* 通过回调脚本名Appname，到Appname\_profile下查看插件权限。
* 使用相应权限运行回调脚本。
  1. **分发事件协议格式**

*Action=xxx；Interface=xxx；Devpath=xxx；*

ACTION=ifup # 事件行为；“iface”的有效值为 ifup、ifdown；“block”的有效值为add、remove；

INTERFACE=wan # [iface]网络接口的OpenWrt名称，例如：wan、lan、lan2

DEVPATH=/dev/sda1 # [block]可插拔存储的设备文件名

#### 获取插件状态

* 调用cpkgcli命令，传入appname，可获取插件运行状态。
* 从Appname.profile中获取插件权限。
* 使用相应权限运行Appname.script中的stauts函数。

## 接口定义

#### Appname.info格式

*Name:广告过滤*

*Ename:ADBlock*

*Version: 1.0.0.2*

*Info: 从此，广告已成往事*

*Author: zhangjianfei3@gmail.com*

*Largeicon:/apps/ADBlock/icon/large.jpg*

*Smallicon: /apps/ADBlock/icon/small.png*

Name：插件名，可为中文名。

Ename：关键，环境中Appname与用户名均为此名。只可使用英文、数字。

Info：插件简介。

Author：插件开发者。

Largeicon：缓存插件图标到指定路径。

Smallicon：缓存插件图标到指定路径。

#### Appname.script格式

1. #!/bin/sh
3. start()
4. {
5. # App启动代码
6. # 函数返回0表示启动成功，返回非0表示启动失败
7. }
9. stop()
10. {
11. # App停止代码
12. }
14. status()
15. {
16. # App状态查询代码
17. if <运行中>; then
18. echo "{ \"status\" : \"running\" }"
19. else
20. echo "{ \"status\" : \"stopped\" }"
21. fi
22. }
24. install()
25. {
26. # App安装代码
27. # 函数返回0表示安装成功，返回非0表示安装失败
28. }
30. uninstall()
31. {
32. # App卸载代码
33. }

## 用户插件与权限管理

* 用户插件在安装时，通过权限管理模块获取安装权限、用户权限、系统权限。并将用户权限和系统权限写入到Appname\_profile.d目录下。
* 插件程序运行中，系统通过分析Appname\_profile.d下Appname文件，获取插件系统运行权限。