# 简介

当前架构上完整的CA功能由三部分实现：CA App、CA Framework、CA Plugin。

## CA App

CA App用于处理应用与CA Plugin的交互。主要包括两方面：

1、响应CA Plugin的消息，包括显示CA Plugin通知和执行CA Plugin请求的操作。

2、获取和设置CA Plugin数据，包括邮件、卡信息、成人级等。

## CA Framework

CA Framwwork实现对CA Plugin加载和管理，同时规范用于CA App、DVB Core、CA Plugin之间交互的统一接口。

## CA Plugin

CA Plugin包含三部分：

1. 实现CA业务逻辑的CA Core库。
2. CA Core库的底层移植接口。
3. CA Core、CA Framework、CA App交互的适配接口

# CA Framework适配实现

CA Plugin需要通过继承CAEngineBase实现与DVB Core、CA App交互的基本接口。CA Framework会在首次加载时通过create接口创建CA Plugin对象，并调用init接口初始CA Plugin。

注意：Plugin对像构造时需要初始化当前CA Plugin的基本属性，如CA\_MIME\_TYPE等，此属性由CA Framework分配和管理，配置错误可能会导致CA Plugin加载失败。

## 与CA App交互实现

CA Plugin与CA App之间的交互由两部分组成：消息、接口。分别由onInfo和getProperty两个接口提供。而消息和接口俱体定义在向CA App提供的下面两个文件中定义：

1. SDMCCAPlublic.h中定义了各CA Plugin中通用的消息和统一的消息参数结构。
2. XXX\_ext.h(XXX为CA的名称)定义了该CA私有的消息和CA App所能调用的接口。

### 消息定义及实现

CA Plugin需要通过CAPlugin：void onInfo(SDMCCA\_INFO\_EVENT event)向CA App发送消息。SDMCCA\_INFO\_EVENT定义如下：

typedef struct {

int mType; //消息的类型分为：Public和Private

int mEventID;//消息的唯一标志

void \*mEvent;//消息附加的数据

} SDMCCA\_INFO\_EVENT;

1. mType: Public、各CA Private消息的起始值在SDMCCAPlublic.h SDMCCA\_EVENT\_TYPE\_E定义。
2. mEentID:由mType的值+偏移组成，不管是否是同一个CA Plugin都不能重叠。
3. mEvent默认为指向CAEvent结构的指针，CA Plugin根据需要把需要传给CA App的数据存储在该数据结构中。

注意：当传送的数据为结构体时需要注意对齐问题，如果结构体中有填充会导致应用解析的数据错误。

建议：如果传送的数据为结构体时，结构体的所有成员都定义为unsigned char型或者unsigned char型的数组。

### 接口定义及实现

CA App通过CAPlugin: SDMCCA\_ERROR\_E getProperty(SDMCCA\_PROPERTY\_E type, void \*pPropParam, void \*pPropRes)接口获取或设置CA Plugin数据。CA Plugin在实现时接口统一定义为：typedef int (\*GetDateFun\_FN)(void \*pInParam, void \*pOutParam);CA App通过getProperty接口和type来选择调用特定的接口，pPropParam和pProgRes对应GetDateFun的pInParam、pOutParam参数。CA Plugin需要通过XXX\_ext.h文件向CA App说明接口作用，返回值和参数结构。

## 与DVB Core交互

CA Framework提供四个函数与DVB Core交互：

1. start: DVB Core在播放节目时会调用此函数，并通过此接口提供当前播放节目的信息
2. stop: DVB Core在停止当前节目播放时会调用此接口
3. notift: DVB Core通过此接口来通知CA Plugin PSI/SI信息的改变并提供当前PSI/SI信息
4. GetDvbInfo：CA Plugin通过此接口来主动获取PSI/SI信息

# CA Plugin集成步骤

参考：sdmc\_ca\_plugin\_implement\_flow.docx