

1. Список файлов:

MAG-CAS-plugin.h - заголовочный файл с описанием интерфейса CAS плагина.

test-cas.c - пример реализации интерфейса CAS плагина.

mk.sh - скрипт для сборки примера.

2. Использование.

Для использования необходимо собрать плагин и положить его в каталог `/home/default`

Максимальное количество одновременных плагинов - 7, с типами CAS от 4 до 10 включительно.

При установке такого типа плеер будет искать библиотеку с именем `/home/default/libCasCustom_x.so`, где `_x` соответствует типу CAS.

Например при вызове:

```
STB_SetCASType(hPlayer, STB_CAS_TYPE_CUSTOM4)
```

или в JavaScript

```
stb.SetCASType(4)
```

будет искаться библиотека `/home/default/libCasCustom4.so`

3. Описание интерфейса

В динамической библиотеке необходимо реализовать функции `CreateCasPlugin`, `GetCasApiVersion`, `GetCasPluginDescription`, а также содержимое структуры `STB_MAG_Cas_t`:

`STB_MAG_Cas_t* CreateCasPlugin(STB_MAG_Cad_t* mag_interface, const char* ini_filename)` - инициализация системы условного доступа.

`int GetCasApiVersion(void)` - Версия CAS API, реализованная плагином.

<code>const char* GetCasPluginDescription(void)</code>	Описание CAS плагина.
<code>void (*OnPmtChange)(byte* buffer, int length)</code>	Вызывается при смене таблицы PMT в потоке, необязателен для реализации.
<code>void (*OnEcm)(byte* ecm, int length)</code>	Вызывается при получении ECM пакета соответствующего заданной CAS. Реализация обязательна только для CAS модулей работающих в режиме <code>CAS_Flags_PMT_ECM</code> .

int (*Decrypt)(byte* in_buffer, int length)	Вызывается для того, чтобы CAS модуль дешифровал блок данных из зашифрованного транспортного потока. Данные в in_buffer всегда выровнены на начало транспортного пакета. Результат дешифрования сохраняется в in_buffer. Возвращает количество дешифрованных байт. Плагин обязан очистить биты Transport Scrambling Control в случае успешного дешифрования транспортного пакета.
void (*Deinit)(void)	Вызывается при окончании работы с CAS для того, чтобы освободить ресурсы библиотеки.
void (*ResetStream)(void)	Вызывается при смене зашифрованного канала.
int (*GetSysID)(void)	Получить SysID соответствующий вендору CAS. Реализация обязательна только для CAS модулей работающих в режиме CAS_Flags_PMT_ECM.
int (*GetSoID)(void)	Получить SoID соответствующий вендору CAS. Реализация обязательна только для CAS модулей работающих в режиме CAS_Flags_PMT_ECM.
CAS_Flags_e (*GetCasFlags)(void)	Получить флаги, указывающие в каком режиме должен работать CAS модуль (см. описание CAS_Flags_e).
ECM_Format_e (*GetEcmFormat)(void)	Получить формат, в котором CAS плагин будет получать ECM пакеты: ECM_FORMAT_SECTION - только секцию ECM ECM_FORMAT_TS - полный транспортный пакет, содержащий ECM.
void (*SetAdditionalParam)(const char* name, const char* value)	Установить дополнительные параметры CAS.
CAS_Flags_e	определяет режим работы CAS модуля: CAS_Flags_PMT_ECM - плагину не посылается весь транспортный поток, а только вызываются функции OnPmtChange и OnEcm по соответствующим событиям. Модуль должен по результатам обработки ECM устанавливать ключи и тип зашифрования с помощью функций SetScramblingType и SetScramblingKey из STB_MAG_Cad_t. CAS_Flags_DecodeAll - плагину передается весь поток для дешифрования через функцию Decrypt. CAS_Flags_DecodeAll_PMT - плагину передается весь поток для дешифрования через функцию Decrypt. Но при этом PMT также зашифрован. Если в STB_MAG_Cas_t некоторые функции не реализованы, то соответствующие поля должны быть NULL. Структура STB_MAG_Cad_t определяет функции, которые необходимо вызывать для установки алгоритма и ключей шифрования.

From:

<https://docs.infomir.com.ua/> -

Permanent link:

https://docs.infomir.com.ua/doku.php?id=knowledge_base:test_cas_ap_i

Last update: **2021/12/15 15:02**

