

EreTik's Box

Разработка, исследование и низкоуровневое программирование

EreTik's Box » [Статьи, исходники](#) » Установка устройства аудио-воспроизведения по умолчанию, используя недокументированный COM-интерфейс IPolicyConfig

[Стартовая страница](#)

[Статьи, исходники](#)

[Заметки о WinDbg](#)

[Скачать](#)

[Внешние ссылки](#)

[Обо всем](#)

[Контакты](#)

[Утилиты](#)

В настоящее время ситуация, когда на машине установлено несколько звуковых плат, уже редкостью не назовешь. Вот и я, став обладателем "внешнего" телевизора, который подключается через HDMI, для своего ноутбука, стал активно пользоваться установкой устройства аудио-воспроизведения по умолчанию. Как оказалось, единственный доступный способ переключить устройство аудио-воспроизведения - **MmSys.cpl**.

Постоянно вызывать свойства звуковых устройств это не решение. Поэтому я решил написать небольшую утилиту **DefSound**. Основное назначение - висеть в системном трее и переключать устройство по-умолчанию.

Как выяснилось, установить устройство воспроизведения по умолчанию документированным способом не возможно. Пришлось вооружиться стандартным боекомплект: **IDA** и **WinDBG**. Тестовой системой выступила Windows **Vista**.

Для начала статически анализируем дизассемблированный листинг **MmSys.cpl**. Немного поковырявшись, приходим к методу **CEndpoint::MakeDefault**. Это метод отвечает за обработку события установки пользователем нового аудиоустройства по умолчанию. Метод иницирует ATL'ное получение COM-интерфейса :

```
1  push    17h                ; dwClsContext
2  push    0                  ; pUnkOuter
3  ; CLSID запрашиваемого интерфейса
4  push    offset __GUID_294935CE_F637_4E7C_A41B_AB255460B862
5  lea     ecx, [ebp+var_10]
6  ; вызов ATL::CComPtrBase<IPolicyConfig>::CoCreateInstance(_GUID const &,IUnknown *,ulong)
7  call    1CD32850h
```

Из приведенного листинга мы получаем интересующий нас **CLSID: GUID{294935CE-F637-4E7C-A41B-AB255460B862}**. Но для получения COM-интерфейса также необходим IID, его мы с легкостью получим, посмотрев дизассемблированный листинг метода **ATL::CComPtrBase::IPolicyConfig::CoCreateInstance** :

```
1  mov     [ebp+ppv], ecx
2  mov     eax, [ebp+ppv]
3  push    eax                ; ppv
4  ; IID запрашиваемого интерфейса
5  push    offset __GUID_568b9108_44bf_40b4_9006_86afe5b5a620 ; riid
6  mov     ecx, [ebp+dwClsContext]
7  push    ecx                ; dwClsContext
8  mov     edx, [ebp+pUnkOuter]
9  push    edx                ; pUnkOuter
10 mov     eax, [ebp+rclsid]
11 push    eax                ; rclsid
12 ; вызов ole32!CoCreateInstance(...)
13 call    ds:__imp__CoCreateInstance@20
```

Вот мы и добрались до **ole32!CoCreateInstance(...)**. Тем самым получаем интересующий нас идентификатор интерфейса **IID: GUID{568b9108-44bf-40b4-9006-86afe5b5a620}**.

Полученных данных не достаточно, так как мы все еще не можем восстановить список виртуальных методов интерфейса. Поэтому продолжаем трассировку метода **CEndpoint::MakeDefault** и приходим к вызову метода полученного интерфейса.

В итоге трассировки мы попадаем в метод **CPolicyConfigClient::SetDefaultEndpoint**, реализованный в динамической библиотеке **AudioSes.dll**.

Загружаем динамическую библиотеку **AudioSes.dll** в IDA, и получаем искомый символ **ATL::CComObject<class CPolicyConfigClient>::`vftable'**: таблицу виртуальных функций интерфейса **IPolicyConfig**

На основе вышеописанного небольшого исследования я написал небольшую утилиту **DefSound**, которая позволяет по правой кнопке мыши, в выпадающем меню, выбрать устройство аудио-воспроизведения по умолчанию.

Updated (22.09.2010)

Получив отзыв об этом небольшом исследовании от **Robert Bacs**, я обнаружил, что часть методов **IPolicyConfig** на Windows 7 возвращают **ERROR_NOT_SUPPORTED(0x32)**. Беглое дизассемблирование показало, что на Windows 7 класс **IPolicyConfig** переименован в **IPolicyConfigVista**, соответственно **CPolicyConfigClient** переименован в **CPolicyConfigVistaClient**. Этот интерфейс присутствует на обеих ОС (Vista и 7) и метод **SetDefaultEndpoint(...)** успешно работает, поэтому утилита **DefSound** оказалась работоспособной на обеих версиях ОС.

Продолжив, я нашел новый интерфейс **CLSID: GUID{870af99c-171d-4f9e-af0d-e63df40c2bc9}** и **IID: GUID{f8679f50-850a-41cf-9c72-430f290290c8}**, который в Windows 7 называется **IPolicyConfig**. Я тоже решил перейти к новому именования интерфейсов. В **IPolicyConfig** реализованы методы, которые в интерфейсе **CPolicyConfigVistaClient** на Windows 7 возвращают **ERROR_NOT_SUPPORTED**:

- **GetMixFormat(...)**
- **GetProcessingPeriod(...)**
- **SetProcessingPeriod(...)**
- **GetShareMode(...)**
- **SetShareMode(...)**
- **SetEndpointVisibility(...)**

Следовательно, эти методы стоит использовать следующим образом: на Windows Vista, где **IPolicyConfig** не зарегистрирован, необходимо вызвать эти методы у интерфейса **IPolicyConfigVista**. На ОС Windows 7 эти методы необходимо вызывать из **IPolicyConfig**.

Из вышесказанного, я дополнил заголовочный файл [PolicyConfig.h](#), в котором старый интерфейс имеет суффикс Vista и присутствует новый интерфейс IPolicyConfig, доступный только в Windows 7.

Скачать утилиту [DefSound](#) и [исходные тексты](#) к ней.

ΞρεTik