



ООО «Цифрасофт»

Описание функциональных характеристик программного обеспечения и информация, необходимая для установки и эксплуатации программного обеспечения

**Программное обеспечение
«SoundCode SDK: набор библиотек для вставки акустических
водяных знаков в аудиовизуальный контент и их
декодирования»**

(версия 3.0)

Оглавление

1 Аннотация.....	4
2 Термины и сокращения.....	5
3 Общие сведения.....	6
3.1 Обозначение и наименование программного продукта.....	6
3.2 Инструменты разработки.....	6
3.3 Функциональные характеристики и сфера применения.....	6
3.3.1 Функциональные характеристики.....	6
3.3.2 Сфера применения.....	7
3.4 Использование SoundCode SDK совместно с FFmpeg и GStreamer.....	7
4 Функциональная архитектура программного обеспечения.....	8
4.1 Состав ПО.....	8
4.2 Функции частей ПО.....	8
4.2.1 Схема архитектуры ПО (Рис. 1).....	8
4.2.2 Тип реализующий ЭВМ.....	9
4.2.3 Виды и версии операционной системы.....	9
4.2.4 Минимальные требования для установки части 1.....	9
4.2.5 Минимальные требования для установки части 2.....	9
4.2.6 Установка части 1.....	9
4.2.7 Установка части 2 (настольные ПК).....	10
4.2.8 Установка части 2 (мобильные ОС).....	10
5 Эксплуатация ПО SoundCode SDK.....	11
5.1 Эмбедирование водяных знаков на сервере или ПК.....	11
5.1.1 Особенности использования различных аудио кодеков.....	12
5.2 Декодирование водяных знаков на сервере или ПК.....	12

5.3 Декодирование водяных знаков в мобильных приложениях.....	13
5.3.1 Использование SoundCode SDK в приложениях Android.....	13
5.3.2 Использование SoundCode SDK в приложениях iOS.....	15
5.3.3 Демо-приложения для мобильных платформ.....	17
6 История изменений.....	19

1 Аннотация

Набор библиотек SoundCode SDK реализует функционал вставки цифровой информации (акустических водяных знаков) в аудиодорожку любого аудиовизуального контента. Кроме того, набор библиотек SoundCode SDK позволяет извлечь (декодировать) акустические водяные знаки из предварительно промаркированного аудиосигнала. Акустические водяные знаки SoundCode позволяют незаметно для слушателя передавать в аудиосигнале небольшой объем информации (несколько десятков бит) и выдерживают все виды искажений, возникающих при распространении аудиосигнала по цифровым и аналоговым каналам передачи информации, при сжатии аудио с потерями, а так же при передаче акустического сигнала «по воздуху». Набор библиотек SoundCode SDK поддерживает операционные системы Linux, Windows, а модули декодирования также могут быть встроены в приложения для мобильных операционных систем Android и iOS.

Область применения – маркирование медиа контента, панельные исследования для анализа аудитории телевизионных каналов и радиостанций, исследование аудитории аудио- и видео- контента в сети Интернет, разработка приложений типа «второй экран», а также приложений для предотвращения несанкционированной утечки медиа-контента и отслеживания источников такой утечки

Данный документ содержит:

- описание функциональных характеристик программного обеспечения;
- информацию, необходимую для установки (интегрирования) программного обеспечения в те или иные программно-аппаратные комплексы и информационные системы;
- информацию, необходимую для эксплуатации программного обеспечения

2 Термины и сокращения

В документе использованы следующие термины:

Термин	Определение
Правообладатель (Лицензиар)	Лицо или организация, имеющие право, долю, требование или интерес в системе или в обладании ее характеристиками, удовлетворяющими ее потребности и ожидания
Приобретающая сторона (Лицензиат)	Сторона, которая приобретает или получает продукт или услугу от поставщика
Приобретение (Лицензирование)	Процесс получения системы, программного продукта или программной услуги
ОС	Операционная система, комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.
Фреймворк	Программная платформа, определяющая системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.
ПО SoundCode SDK	Программное обеспечение «SoundCode SDK: набор библиотек для вставки акустических водяных знаков в аудиовизуальный контент и их декодирования»
ПО	Программное обеспечение
Вотермарка	Сокращенное название акустического водяного знака

3 Общие сведения

3.1 Обозначение и наименование программного продукта

Полное наименование: **Программное обеспечение «SoundCode SDK: набор библиотек для вставки акустических водяных знаков в аудиовизуальный контент и их декодирования»**

Сокращенное название: **SoundCode SDK**

Альтернативное название версий: AudioTag SDK, SoundMatch SDK

3.2 Инструменты разработки

Для разработки **SoundCode SDK** использовались языки программирования: C/C++, Java, Objective-C, Swift

3.3 Функциональные характеристики и сфера применения

3.3.1 Функциональные характеристики

- ПО SoundCode SDK позволяет встраивать в аудиодорожку аудиовизуального контента небольшой объем цифровой информации не внося при этом в аудиосигнал заметных на слух искажений и иных модификаций.
- ПО SoundCode SDK также позволяет извлекать из ранее промаркированной аудиодорожки встроенную цифровую информацию даже если аудио сигнал претерпел типовые искажения, возникающие при распространении звуковых сигналов по цифровым и аналоговым каналам телерадиовещания, при сжатии аудиодорожки «с потерями», а так же при передаче акустического сигнала «по воздуху» и его последующей записи с микрофона.
- ПО SoundCode SDK предоставляет возможность тонкой настройки параметров маркирования для балансирования уровня акустической заметности и помехоустойчивости декодирования водяных знаков.
- ПО SoundCode SDK может быть использовано в виде отдельных утилит для эмбедирования и декодирования аудиофайлов без сжатия (в формате WAV), а также предоставляет набор библиотек, которые могут быть использованы совместно с популярным сторонним программным обеспечением для транскодирования аудиовидеофайлов, в частности, фреймворком FFmpeg (<https://ffmpeg.org/>) и фреймворком GStreamer (<https://gstreamer.freedesktop.org/>).
- При использовании совместно с мультимедиа фреймворками FFmpeg и GStreamer ПО SoundCode SDK обеспечивает возможность эмбедирования акустических водяных знаков в аудио и видео файлы с любым форматом видео потока и с аудио потоком сжатым кодеком «с потерями», при условии, что: скорость аудио потока составляет не менее 48 кбит/с для формата MP3 или ее эквивалент для других форматов, таких как AAC, AC3, OGG и др, частота дискретизации аудио сигнала не менее 11 кГц, а разрядность отсчетов аудио сигнала не менее 16 бит.
- Набор библиотек SoundCode SDK может быть также использован для реализации функционала декодирования акустических водяных знаков в реальном времени в мобильных приложениях на базе ОС Android и iOS.

3.3.2 Сфера применения

- Панельные исследования аудитории медиа, проводимые исследовательскими компаниями по заказу производителей медиа контента
- Решение узконаправленных отраслевых задач в области медиа измерений
- Решение узконаправленных отраслевых задач в области маркировки медиа контента
- Решение узконаправленных отраслевых задач в области отслеживания каналов утечки медиа контента

3.4 Использование SoundCode SDK совместно с FFmpeg и GStreamer

Для поддержки различных форматов аудио и видео кодеков и файловых контейнеров мультимедиа файлов ПО SoundCode SDK можно использовать в составе популярных открытых фреймворков таких как FFmpeg и GStreamer. Оба фреймворка представляют собой свободное ПО, лицензируемое в соответствии с открытой лицензией LGPLv2.1, см. <https://www.gnu.org/licenses/old-licenses/lgpl-2.1.ru.html>. Данная лицензия позволяет использовать и распространять ПО собранное на базе фреймворков FFmpeg и GStreamer совместно с проприетарными библиотеками (плагины), в частности с такими как SoundCode SDK. Для удобства пользователей в комплект поставки ПО SoundCode SDK входят бинарные утилиты ffmpeg и плагин gstreamer с интегрированными библиотеками SoundCode SDK собранными в режиме статической компиляции. Помимо этого в комплект поставки ПО SoundCode SDK входят объектные версии библиотек SoundCode SDK для FFmpeg и GStreamer и инструкции по сборке, с помощью которых пользователи могут самостоятельно скомпилировать утилиты ffmpeg и плагины gstreamer с использованием любых версий данных мультимедиа фреймворков. Такой подход с одной стороны обеспечивает выполнение условий распространения, которые указаны в п.6 лицензии LGPL, с другой стороны - позволяет пользователю SoundCode SDK самостоятельно обновлять и актуализировать версии FFmpeg и GStreamer, если это необходимо.

Исходный код фреймворка GStreamer доступен на <https://cifrasoft.ru/files/gstreamer-main.zip>

Исходный код фреймворка FFmpeg доступен на <https://cifrasoft.ru/files/FFmpeg-master.zip>

4 Функциональная архитектура программного обеспечения

4.1 Состав ПО

ПО SoundCode SDK имеет две составляющие (см. п.4.2.1):

Часть 1: Набор библиотек и утилит для эмбедирования акустических водяных знаков в аудио/видео файлы или цифровой аудио/видео поток

Часть 2: Набор библиотек для интеграции функционала декодирования водяных знаков в настольные ОС Linux и Windows и мобильные приложения на базе мобильных ОС Android, iOS

4.2 Функции частей ПО

А) Часть 1 обладает такими функциями, как:

- встраивание водяных знаков в аудио дорожку аудио/видео файлов и цифровых аудио/видео потоков

Б) Часть 2 обладает такими функциями, как:

- декодирование и идентификация водяных знаков в маркированном медиаконтенте как «по воздуху», так и посредством обработки маркированного медиафайла

4.2.1 Схема архитектуры ПО (Рис. 1)



4.2.2 Тип реализующий ЭВМ

- IBM PC совместимый компьютер с архитектурой процессоров x86
- Мобильные и настольные устройства на базе процессоров с архитектурой ARM

4.2.3 Виды и версии операционной системы

- Семейство операционных системы Linux, Windows
- Мобильные операционные системы Android и iOS.

4.2.4 Минимальные требования для установки части 1

- Набор библиотек/утилит для эмбедрования акустических водяных знаков может быть установлен на сервер, персональный компьютер или ноутбук на базе ОС:
 - Linux Debian 11 и выше, а также совместимых ОС
 - Windows 7 и выше, а также совместимых ОС
- Набор библиотек и утилиты эмбедрования не требуют наличия на сервере, персональном компьютере или ноутбуке интернет соединения и дополнительных библиотек помимо стандартных библиотек, поставляемых с указанными ОС; для вставки акустических водяных знаков в аудио/видео- файлы в формате отличном от несжатого формата WAV, требуется наличие дополнительных мультимедиа фреймворков с открытым исходным кодом FFmpeg (<https://ffmpeg.org/>) или GStreamer (<https://gstreamer.freedesktop.org/>).

4.2.5 Минимальные требования для установки части 2

- Набор библиотек/утилит для декодирования акустических водяных знаков может быть установлен на сервер, персональный компьютер или ноутбук на базе ОС:
 - Linux Debian 11 и выше, а также совместимых ОС
 - Windows 7 и выше, а также совместимых ОС
- Набор библиотек и утилиты декодирования не требуют наличия на сервере, персональном компьютере или ноутбуке интернет соединения и дополнительных библиотек помимо стандартных библиотек, поставляемых с указанными ОС; для декодирования аудио/видео- файлов в формате отличном от несжатого формата WAV, требуется наличие дополнительных утилит с открытым исходным кодом FFmpeg (<https://ffmpeg.org/>) или мультимедиа фреймворка GStreamer (<https://gstreamer.freedesktop.org/>).
- Набор библиотек для декодирования акустических водяных знаков может быть также интегрирован в нативные приложения, работающие на базе операционных систем:
 - iOS 8 и выше
 - Android 8 и выше

4.2.6 Установка части 1

- Бинарная утилита командной строки для эмбедрования вотермарок в несжатые аудио-файлы формата WAV (wmen) не требует дополнительной установки —

достаточно распаковать файловый архив с набором библиотек и утилит SoundCode SDK в любую пользовательскую папку на сервере или настольном компьютере, дополнительных зависимостей не требуется.

- Для установки библиотек SoundCode SDK в качестве аудио-фильтра в мультимедиа фреймворке FFmpeg требуется пересобрать утилиту ffmpeg с подключением библиотеки af_audiotag.o из состава ПО SoundCode SDK. Для этого:
 - Зарегистрируйте новый аудиофильтр добавив в файл ffmpeg/libavfilter/allfilters.c строку:
REGISTER_FILTER(AUDIOTAG, audiotag, af);
 - Добавьте в файл: ffmpeg/libavfilter/Makefile строку:
OBJS-\$(CONFIG_AUDIOTAG_FILTER) += af_audiotag.o
 - Перекомпилируйте утилиту ffmpeg с необходимыми флагами
- Для удобства использования в составе ПО SoundCode SDK распространяется статическая сборка утилиты ffmpeg с уже подключенной библиотекой af_audiotag.o.

4.2.7 Установка части 2 (настольные ПК)

- Бинарная утилита командной строки для декодирования вотермарок (wmdd) не требует дополнительной установки — достаточно распаковать файловый архив с набором библиотек и утилит SoundCode SDK в пользовательскую папку на сервере или настольном компьютере. Для декодирования несжатых аудиофайлов формата WAV дополнительных зависимостей не требуется. Для декодирования сжатых аудиофайлов в формате (MP3, AAC, OGG и др), а также аудиодорожки видео файлов требуется наличие установленной в системе библиотеки FFmpeg (для работы утилиты декодирования может быть использована утилита ffmpeg с подключенной библиотекой SoundCode SDK).

4.2.8 Установка части 2 (мобильные ОС)

- Для добавления функционала SoundCode SDK в мобильное приложение Android добавьте в проект Android Studio в папку /app/libs/ библиотеки в формате AAR:
 - audioRecordLib-sdk-release.aar
 - soundCode3Lib-sdk-release.aar
- Для добавления функционала SoundCode SDK в мобильное приложение iOS добавьте в проект Xcode добавьте:
 - заголовочные файлы библиотеки (\SCS\SCSCore\Include):
 - ac_typedef.h
 - ac_vec16.h
 - ac_vec32.h
 - soundCode.h
 - заголовочные и исходные файлы сервисов (\SCS):
 - SCSSettings.h
 - SCSAudioManager.h
 - SCSAudioManager.m
 - SCSManager.h

- SCSManager.m
- SCSSearchManager.h
- SCSSearchManager.m
- нативную библиотеку (\SCS\SCSCore):
 - SoundCodeLibU.a

5 Эксплуатация ПО SoundCode SDK

5.1 Эмбедирование водяных знаков на сервере или ПК

Для эмбедирования водяных знаков на платформах Linux и Windows может быть использована утилита `wmen`, которая позволяет внедрять вотермарки в аудиофайлы формата WAV с параметрами: PCM, частота дискретизации 44100 Гц, разрядность 16-бит, моно режим. Параметры командной строки:

```
wmen -i input_file.wav -id wm_id -r robust -k api_key -o output_file.wav
```

где:

`input_file.wav` — имя входного wav-файла

`wm_id` — числовой идентификатор вотермарки (1... 134217727)

`robust` — уровень робастности (2...5)

`api_key` — ключ пользователя

`output_file.wav` — имя выходного wav-файла

Для эмбедирования вотермарок в аудио файлы других форматов, а также эмбедирования вотермарок в видео файлы можно использовать утилиту `FFmpeg` с кастомным аудио фильтром из состава ПО SoundCode SDK.

Для этого в параметрах командной строки используется флаг `(-af)`, а сам фильтр имеет обозначение `atag` и следующие параметры:

`atag=id` - идентификатор вотермарки (1... 134217727)

`atag=r` - робастность (2...5)

`atag=key` (ключ пользователя)

При кодировании с помощью `FFmpeg` можно также использовать другие стандартные параметры утилиты `FFmpeg`, например, `"-vcodec copy"`, чтобы избежать модификации видео потока в процессе эмбедирования.

Пример запуска утилиты эмбедирования:

```
ffmpeg -i input_file.mkv -vcodec copy -af atag=id=33333:r=5:key=USER_KEY output_file.mkv
```

где

`input_file.mkv` - исходный видео файл,

`output_file.mkv` - файл с эмбедированной вотермаркой с параметрами: `id=33333` и робастностью `r=5`.

Для того, чтобы избежать модификации формата аудио и видео потоков в выходном файле, рекомендуется явно указывать в командной строке флаги копирования видео потока и параметры выходного аудио кодека, например:

`-vcodec copy` - указывает, что видеопоток должен быть скопирован в выходной файл без изменений

`-ar 48000` - указывает, что выходной аудио кодек должен использовать частоту дискретизации 48кГц

-acodec pcm_s24le - указывает, что выходной кодек должен использовать 24-битное квантование
-map a:? - указывает, что параметры аудио применяются для всех аудио-стримов

Пример:

```
ffmpeg -i input.mxf -map v:? -vcodec copy -map a:?
        -af atag=id=33333:r=5:key=USER_KEY
        -ar 48000 -acodec pcm_s24le output.mxf
```

С помощью FFmpeg также поддерживается вставка водяных знаков в непрерывный видео поток. В этом случае в параметрах командной строки необходимо указать IP-адрес (доменное имя) и протокол входного и выходного потоков и указать формат выходного потока с помощью ключа -f, например:

```
ffmpeg -i udp://239.1.4.4:1234 -map v:? -vcodec copy -map a:?
        -af atag=id=888:r=3:key=USER_KEY -f flv rtmp://rtmp.vk.com/live2/api_key
```

В случае необходимости использования дополнительных ключей командной строки утилиты FFmpeg смотрите справку на: <https://ffmpeg.org/ffmpeg-all.html>

5.1.1 Особенности использования различных аудио кодеков

Рекомендуется по возможности эмбедировать вотермарки в медиафайлы и потоки, аудиодорожка которых закодирована кодеком без потери качества (PCM, FLAC, MP4 Audio lossless codec и т.д.), и при необходимости после эмбедирования конвертировать аудио дорожку в требуемый lossy-аудио формат (это также можно сделать установив соответствующие параметры командной строки утилиты ffmpeg). Такой подход позволит избежать дополнительных потерь качества аудио из-за процедуры транскодирования.

5.2 Декодирование водяных знаков на сервере или ПК

Для декодирования водяных знаков на ОС Linux и Windows используйте утилиту wmdd. Пример запуска утилиты:

```
wmdd output_file.mkv
```

где

output_file.mkv - проверяемый файл (аудиодорожка содержит эмбедированную вотермарку)

В случае обнаружения вотермарки утилита выдаст информацию в стандартный вывод в следующем виде:

```
01.10.2024 20:13:14 wmdd starting in single param mode.
->> Please wait... (compilation time: Oct 01 2024 09:55:47)
2024.10.01 20:13:36;detected;1111111;1
```

2024.10.01 20:13:36;detected;1111111;2
2024.10.01 20:13:36;detected;1111111;3

данный вывод означает, что в анализируемом файле трижды была обнаружена вотермарка с идентификатором id=1111111 в 1-ю, 2-ю и 3-ю секунды. Утилита `wmdd` поддерживает как аудио файлы формата WAV, 44100 Гц, 16-бит (в этом случае никаких зависимостей не требуется), так и большинство других форматов аудио/видео файлов, если в системе установлен фреймворк FFmpeg или бинарная утилита `ffmpeg` лежит в той же директории, что и утилита `wmdd`.

5.3 Декодирование водяных знаков в мобильных приложениях

5.3.1 Использование SoundCode SDK в приложениях Android

Высокоуровневое взаимодействие с декодером вотермарок ПО SoundCode SDK реализовано через сервис `com.cifrasoft.services.SoundCode3`.

Чтобы начать слушать аудио сигнал с микрофона и пытаться декодировать вотермарки приложение должно инициализировать и запустить сервис `SoundCode3` с объектом класса `SoundCode3Settings`. Чтобы изменить параметры по умолчанию приложение может изменить соответствующие значения в экземпляре класса `SoundCode3Settings`. Для взаимодействия с сервисом `SoundCode3` используется интерфейс `SoundCodeListener`. У данного интерфейса есть две коллбэк-функции:

```
public void onDetectedId (long[] result)
```

вызывается когда в аудио сигнале обнаружено наличие водяных знаков SoundCode. В этом случае, результат декодирования будет возвращен в массиве `result` (long array), в частности:

`result[1]` — содержит идентификатор распознанной вотермарки. Коллбэк-функция:

```
public void onAudioInitFailed ()
```

вызывается в том случае, если сервис не смог инициализировать аудио запись (например, микрофон недоступен по тем или иным причинам).

Для непосредственного использования SoundCode SDK в приложении необходимо выполнить следующие шаги:

Шаг 1) Импортируйте в приложении сервис `SoundCode3`, классы `SoundCode3Settings` и `AudioRecordServiceSettings` и интерфейс `SoundCode3Listener`:

```
import com.cifrasoft.services.SoundCode3;  
import com.cifrasoft.services.SoundCode3Listener;  
import com.cifrasoft.services.SoundCode3Settings;  
import com.cifrasoft.services.AudioRecordServiceSettings;
```

Шаг 2) Создайте обработчик событий *SoundCodeListener* и добавьте свои собственные методы обработки событий:

```
final SoundCodeListener scListener = new SoundCodeListener() {  
    @Override  
    public void onDetectedId (long[] result) {  
        // Do something with result:  
        // Get ContentID via result[1]  
    }  
    @Override  
    public void onAudioInitFailed () {  
        // Audio initialization has failed  
    }  
};
```

Шаг 3) Инициализируйте контекст для сервиса *SoundCode3* (обычно в методе *onCreate()*):

```
SoundCode3.instance(this);
```

где первый параметр — это контекст приложения или *activity*.

Шаг 4) Создайте объекты классов *SoundCode3Settings* и *AudioRecordServiceSettings* с параметрами «по умолчанию» или со своими собственными параметрами, например:

```
SoundCode3Settings scSettings = new SoundCode3Settings();  
AudioRecordServiceSettings arSettings = new AudioRecordServiceSettings();  
arSettings.soundSource = AudioRecordServiceSettings.SOUND_SOURCE.FROM_MIC;  
arSettings.channelConfig = AudioFormat.CHANNEL_IN_MONO;
```

Шаг 5) Инициализируйте сервис *SoundCode3*:

```
SoundCode3.instance().prepare(scSettings, scListener, true, arSettings);
```

где

первый параметр — объект класса *SoundCode3Settings* с настройками сервиса *SoundCode3*

второй параметр — объект класса *SoundCode3Listener*

третий параметр — флаг автоматического запуска сервиса после создания

четвертый параметр — объект класса *AudioRecordServiceSettings* с настройками аудиозаписи для сервиса *SoundCode3*

Шаг 6) Чтобы начать запись аудио в сервисе *SoundCode3* используйте метод *start()*:

```
SoundCode3.instance().start();
```

* этот шаг можно пропустить, если третий параметр при вызове метода `prepare()` был установлен в `true`.

Шаг 7) Чтобы начать декодирование воте́рмарок в аудио потоке, поступающем с микрофона, используйте метод `startSearch()`:

```
SoundCode3.instance().startSearch();
```

Шаг 8) Чтобы остановить распознавание воте́рмарок в аудио потоке, поступающем с микрофона, используйте метод `stopSearch()`:

```
SoundCode3.instance().stopSearch();
```

Вы можете запускать и останавливать сервис несколько раз. Обратите внимание, что при повторном запуске сервиса с помощью метода `startSearch()` декодер будет пытаться использовать аудио данные накопленные ранее, чтобы очистить внутренние буферы аудио и декодера воте́рмарок используйте метод:

```
SoundCode3.instance().reset();
```

Проверить работает ли сейчас сервис распознавания или нет можно с помощью метода:

```
SoundCode3.instance().isSearchRunning()
```

Шаг 9) После завершения использования сервиса *SoundCode3* в приложении используйте метод:

```
SoundCode3.release();
```

чтобы завершить использование микрофона устройства и полностью остановить сервис.

5.3.2 Использование SoundCode SDK в приложениях iOS

В классе *SCSManager* реализованы основные функции для работы с декодером воте́рмарок *SoundCode*. Объект созданный от класса *SCSManager* может существовать в единственном экземпляре (singleton).

- (void) *startSearching* - старт треда поиска воте́рмарок *SoundCode* в непрерывном режиме. Тред поиска запускается с интервалом *userSearchInterval*.
- (void) *singleSearching* - однократный запуск поиска воте́рмарок *SoundCode*.
- (void) *pauseSearching* - поисковой тред останавливается, аудио тред продолжает работать.
- (void) *stopSearching* - остановка треда поиска воте́рмарок *SoundCode*.
- (void) *resetSearching* - сброс внутренних буферов декодера воте́рмарок *SoundCode*.

- (void) *settingsSearching:(scsSettingsStruct*) scsSettings* - инициализация объекта с начальными параметрами *scsSettingsSearchManagerStruct*. По умолчанию структура инициализируется константами из заголовочного файла *SCSSettings.h*. Все константы SoundCode SDK находятся в файле *SCSSettings.h*.

Результаты работы передаются пользовательскому приложению с помощью уведомлений (notification). В пользовательском приложении регистрируется наблюдатель (observer), который принимает сообщения:

- о состояниях: *SCS_SEARCH_MANAGER_NOTIFICATION_STATE_CHANGE_KEY*,
- об ошибках: *SCS_SEARCH_MANAGER_NOTIFICATION_ERROR_KEY*,
- о результатах работы: *SCS_SEARCH_MANAGER_NOTIFICATION_RESULT_KEY*.

SCS_SEARCH_MANAGER_NOTIFICATION – отправляется каждый раз при изменении состояния, обнаружении ватермарки SoundCode, возникновении ошибок.

UserInfo:

SCS_SEARCH_MANAGER_NOTIFICATION_STATE_CHANGE_KEY – отправляется если объект от класса *SCSSearchManager* изменил свое состояние. Содержит перечисление *SCSSearchManagerState* типа *NSUInteger*.

UserInfo:

SCS_SEARCH_MANAGER_NOTIFICATION_RESULT_KEY – содержит ID обнаруженного идентификатора ватермарки (*NSNumber*).

UserInfo:

SCS_SEARCH_MANAGER_NOTIFICATION_ERROR_KEY – отправляется при возникновении ошибок. Содержит перечисление *SCSSearchManagerError* типа *NSUInteger* с кодом ошибки.

Для непосредственного использования SoundCode SDK в приложении необходимо выполнить следующие шаги:

Шаг 1) Импортируем в проект заголовочный файл *SCSManager.h*.

```
#import "SCSManager.h"
```

Шаг 2) На каждое событие регистрируем приемник сообщений (observer):

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(eventsSCSManger:) name:SCS_MANAGER_NOTIFICATION object:nil];
```

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(eventsAudioManger:) name:SCS_AUDIO_MANAGER_NOTIFICATION  
object:nil];
```

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(eventsSearchManger:) name:SCS_SEARCH_MANAGER_NOTIFICATION  
object:nil];
```

Шаг 3) Реализуем обработчик событий.

```
- (void) eventsSearchManger:(NSNotification*) notification {  
    id message;  
    if ((message = [notification.userInfo objectForKey:  
        SCS_SEARCH_MANAGER_NOTIFICATION_STATE_CHANGE_KEY])) {  
        // Do something with result:  
    }  
}
```

Шаг 4) Используем основные функции SoundCode API, для управления режимом поиска воте́рмарок SoundCode: *startSearching*, *singleSearching*, *pauseSearching*, *stopSearching*, *resetSearching*.

5.3.3 Демо-приложения для мобильных платформ

Для более детального знакомства с библиотеками SoundCode SDK и возможностями их использования в мобильных приложениях Android и iOS в комплект поставки ПО SoundCode SDK входят исходные коды двух демонстрационных проектов для Android Studio и XCode с примерами полнофункциональных приложений для декодирования в реальном времени водяных знаков SoundCode через микрофон мобильного устройства, разработанных с использованием библиотек SoundCode SDK.

6 История изменений

Версия	Дата	Описание
1.0	22.06.2025	Первоначальный релиз