

О.С. Волковский, Я.Ю. Петренко

**КОНВЕРТАЦИЯ МЕДИА ФАЙЛОВ НА ПРИМЕРЕ  
АУДИО, В ОПЕРАЦИОННОЙ СИСТЕМЕ WINDOWS 8 ОС  
ИСПОЛЬЗОВАНИЕМ WINDOWS STORE БИБЛИОТЕКИ.  
СРАВНЕНИЕ С FFMPG**

*Аннотация: В работе исследована непосредственная конвертация аудио файлов с использованием Windows Store библиотек, проведено сравнение с ffmpeg библиотекой. Написана тестовая программа, которая может быть использована для создания приложений по обработке медиа файлов. Проведено сравнение Windows Store библиотеки и ffmpeg библиотеки для определения рациональности использования ресурсов ПК при конвертации медиа.*

*Ключевые слова – конвертация, медиа, аудио, Windows Store, ffmpeg*

**Постановка проблемы:** с выходом новой операционной системы Windows 8 пришло такое понятие как Metro приложения, которые могут быть запущены как на стационарном ПК так и на различных мобильных устройствах(планшетах, мобильных телефонах). Новые технологии требуют написания широко используемых программ под свою платформу, что приводит в свою очередь к необходимости изучения и использования новых библиотек и фреймворков. В начале разработки нового продукта нужно исследовать насколько технология будет рациональна в использовании для данной цели, какая из платформ будет использована и насколько это будет эффективно.

**Анализ последних исследований:** Windows 8 представила новый тип приложений: Windows Store приложения – это приложения в стиле Metro. Windows Store приложения имеют совершенно новый внешний вид, они работают на различных устройствах.

Можно разрабатывать Windows Store приложения на множестве языков:

Если есть знания технологий веб-разработки, можно разрабатывать Windows Store приложения используя HTML5, Cascading Style Sheets, Level 3 (CSS3), и JavaScript.

Если есть опыт в .NET, Windows Presentation Foundation, или Microsoft Silverlight приложениях, можно разрабатывать Windows Store приложения используя XAML, с кодом на C++, C#, или Visual Basic.

Если же есть знания DirectX, то можно разрабатывать DirectX Windows Store приложения используя C++ и HLSL, что бы в полной мере воспользоваться графический оборудованием.

**Формулировка целей работы:** Целью данной работы является разработка программы в ОС Windows 8, на платформе .NET с использованием Windows Store библиотек, которую в последующем можно взять за основу для написания программного продукта конвертации медиа файлов. Сравнение производительности предложенного метода конвертации медиа с уже существующим.

**Основная часть:** Рассмотрим конвертацию цифровых медиа файлов, таких как аудиофайлы, из одного формата в другой. Для начала рассмотрим аудио форматы, с которыми позволяет работать Windows Store библиотека.

Таблица 1

Аудио форматы [1]

Медиа контейнер или формат файла	Расширение файла	Форматы аудио потока (Кодеки)
MPEG-4	.3g2	AAC (LC, HE)
	.3gp2	
	.3gp	
	.3gpp	
	.m4a	AAC (LC, HE)
	.m4v	MP3
	.mp4v	AC3 (DD, DD+)
	.mp4	
.mov		
MPEG-2	.m2ts	MPEG-2 (L1, L2, stereo-only), MPEG-1 (L1, L2), AAC (LC, HE), AC3 (DD, DD+)
ASF	.asf	WMA standard
	.wm	WMA voice
	.wmv	WMA lossless
	.wma	WMA Pro, AC3 (DD, DD+)
ADTS	.aac	AAC (LC, HE)
	.adt	
	.adts	
MP3	.mp3	MP3
WAV	.wav	PCM, MP3, MS ADPCM, IMA ADPCM, MS DDIT G.711, MS GSM 6.10, AC3 (DD, DD+)

AVI	.avi	PCM, MP3, MS ADPCM, IMA ADPCM, MS CCITT G.711, MS GSM 6.10 AC3 (DD, DD+)
AC-3	.ac3 .ec3	AC3 (DD, DD+)

Немаловажной особенностью Windows Store библиотеки является асинхронное программирование, что принципиально отличает реализацию с ее помощью и дает этому способу преимущество. Типичный сегмент кода, написанный на C# или Microsoft Visual Basic, выполняется синхронно, это значит, что выполнение строки заканчивается до начала выполнения следующей. Различные программные модели, такие как Модель асинхронного программирования и Асинхронный паттерн, используются долгое время, и позволяют разработчикам писать асинхронный код. Пока эти модели успешно реализовывали асинхронность, результирующий код сосредотачивался именно на ней, в ущерб выполняемой задаче. Это сделало асинхронный код тяжелым для написания, понимания и поддержки. В Windows Runtime, .NET фреймворк, C# и Visual Basic в компиляторы были добавлены функции, которые позволяли абстрагировать асинхронный механизм от кода, чтобы можно было писать асинхронный код, который сосредотачивается на том что делает код, а не как он это делает. Таким образом можно писать асинхронный код который очень похож на синхронный.

Использование асинхронного программирования помогает приложениям оставаться активными, когда он выполняет работу, занимающую длительный период времени. Например, приложение, которое получает информацию из Интернета, может провести несколько секунд в ее ожидании. Если использовать синхронный метод в потоке пользовательского интерфейса для получения данных, приложение будет заблокировано, пока метод не вернет что-либо. Приложение не будет в таком случае отвечать на действия пользователя. Намного лучше использовать асинхронное программирование, когда приложение продолжает быть активным и отвечать на пользовательский интерфейс, пока оно ждет окончания операции.

Имена асинхронных функций заканчиваются с "Async" чтобы указать, что часть их выполнения может продолжаться после возвращения функции. При использовании асинхронных функций в Windows Store приложениях, код осуществляет неблокирующие вы-

зовы постоянно. При реализации этих асинхронных моделей в функциях код будет понятен и использован предсказуемым образом [2].

Типичные задачи, которые требуют вызов Windows Runtime функций:

- отображение диалогового окна сообщения, - работа с файловой системой, - отправка и получение данных из Интернета.

Используя асинхронные API среды выполнения Windows, не нужно управлять потоками явным образом и напрямую обращаться к базовым функциям. [1]. Рассмотрим простой метод перекодирования аудио файлов. В начале примера используем MP3 и ASF форматы.

Для начала реализуем функцию открытия файла и опишем все его особенности. Что бы открыть аудио файл из системы нам нужно использовать класс `FileOpenPicker` [3] . Алгоритм открытия аудио файла (рис. 1):

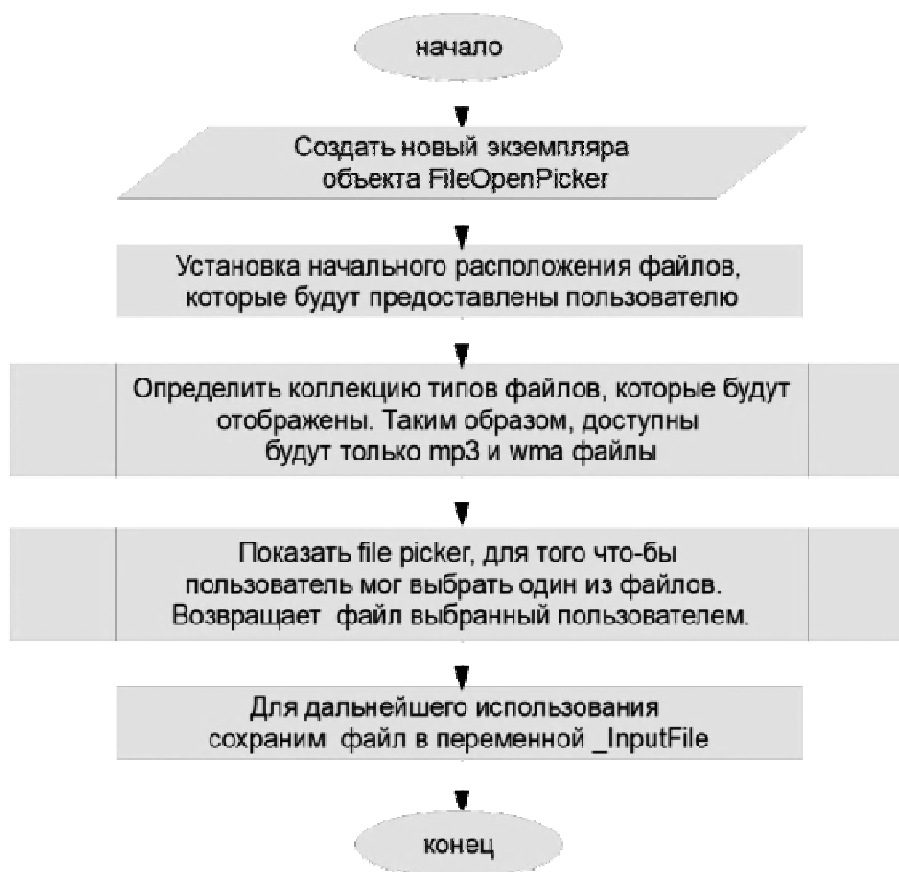


Рисунок 1 – Блок – схема алгоритма открытия аудио файла

Теперь перейдем к функции перекодирования аудио. Алгоритм функции перекодирования аудио (рис. 2):



Рисунок 2 – Блок – схема алгоритма перекодирования аудио

Проанализируем полученные результаты, тестирования медиа конвертации с учетом использования следующей аппаратной конфигурации:

Таблица 2

## Аппаратная конфигурация

ОС	Windows 8 Pro
Процессор	Intel(R) Core(TM) i3 CPU M380 @ 2.53GHz
Установленная память (RAM)	3,00GB
Тип системы	64-bit Operating System, x64-based processor

Конвертация с использованием Windows Store библиотеки из mp3 в wma привела к результатам, представленным на рис.3

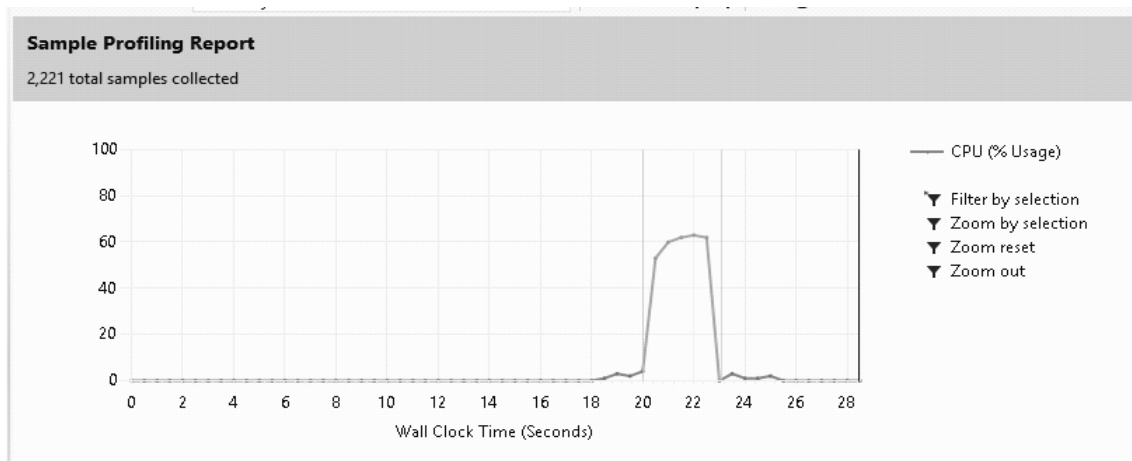
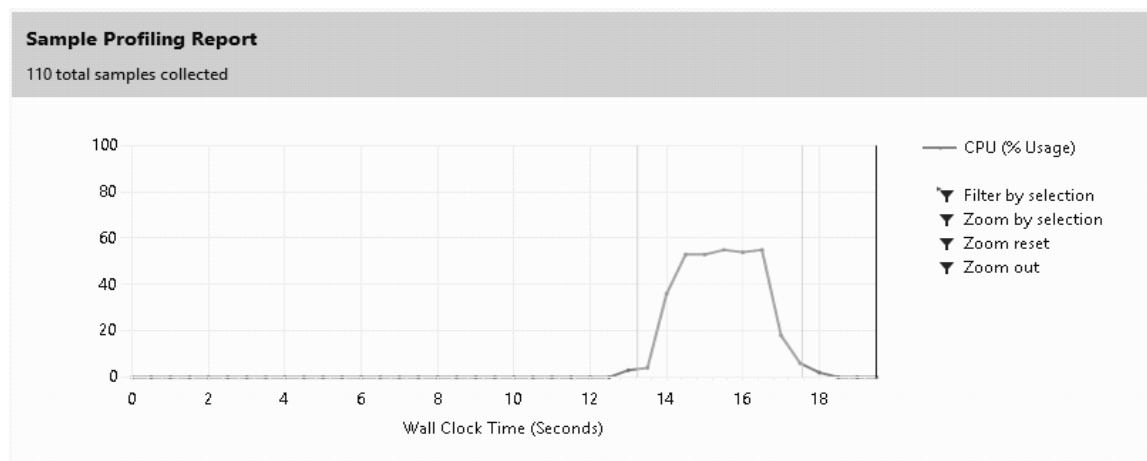


Рисунок 3 - Конвертация из mp3 в wma. Загрузка процессора

Из графика видно, что загрузка процессора находилась на уровне 60%, а время конвертации заняло 3 секунды.

Использование Windows Store библиотеки для конвертации из wma в mp3 привело к следующим результатам (Рис. 4).



Ри

сунк 4 - Конвертация wma в mp3 . Загрузка процессора

В этом случае загрузка процессора составила 55%, а время конвертации около 3 секунд.

Для сравнения полученных результатов была также разработана тестовая программа для конвертации с использованием ffmpeg, которая, в свою очередь, показала следующие результаты (рис.5).

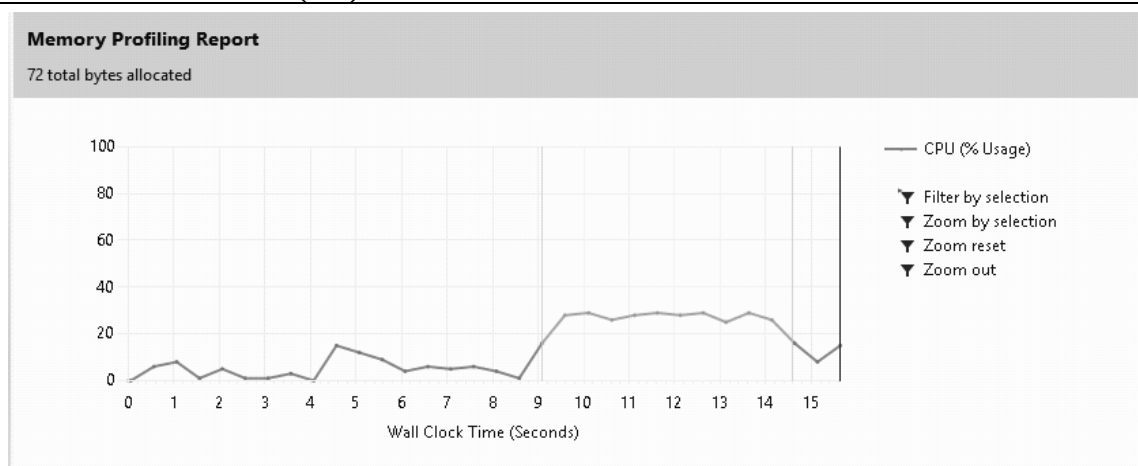
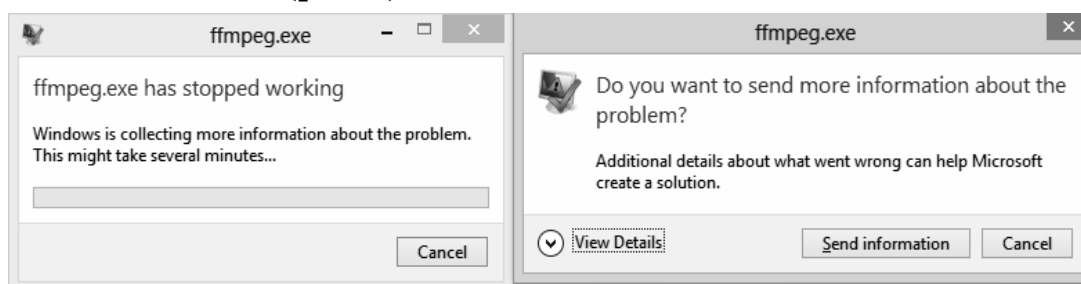


Рисунок 5 - Конвертация wma в mp3. Загрузка процессора

Анализируя график из рис.5 можно увидеть, что загрузка процессора снизилась до 30%, при этом время конвертации возросло до 6 секунд.

Конвертация mp3 в wma с использованием ffmpeg привела к ошибке, таким образом файл не удалось конвертировать с помощью данной библиотеки (рис.6).

Рисунок 6 - Конвертация из mp3 в wma.  
Ошибка во время выполнения

Сравним полученные данные:

Таблица 3

Сравнение данных

Конвертация	Windows Store биб- лиотека		ffmpeg библиотека	
	загрузка процессора	время конверта- ции	загрузка процессора	время конверта- ции
из mp3 в wma	60%	3 с.	error	
из wma в mp3	55%	3 с.	30%	6 с.

Из таблицы видно, что загрузка процессора для библиотеки `ffmpeg` значительно меньше в случае конвертации из `wma` в `mp3`, но время при этом в два раза больше. Учитывая современные технологии загрузка процессора не так критична как время выполнения, таким образом в этом случае можно отдать предпочтение `Windows Store` библиотеке.

Для варианта конвертация из `mp3` в `wma` безусловно лучше использовать библиотеку `Windows Store`, так как `ffmpeg` библиотека не показала положительных результатов для этого случая.

Если же сравнивать эти две библиотеки в перекодировке `lossless` форматов, то сделать это можно лишь с использованием `ffmpeg`, `Windows Store` библиотека в этом случае выдает неизвестную ошибку (рис. 7).

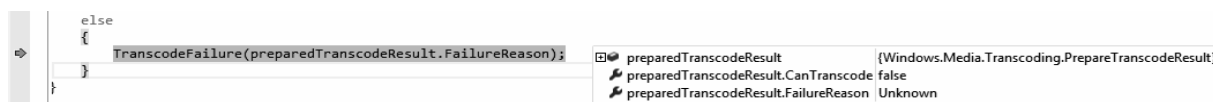


Рисунок 7 - Ошибка при перекодировке `lossless` форматов.  
Windows Store

**Выводы:** Как видно из проведенных исследований `Windows Store` библиотеки лучше для работы с `Windows media formats`, таких как `wma` или `wav`. В случае использования `ffmpeg` для этих целей, могут возникнуть некоторые трудности при перекодировке этих форматов.

Если нужно перекодировать `lossless` форматы (напр. `flac` или `ogg`) – есть возможность сделать это только с использованием `ffmpeg`. `Windows Store` не поддерживают `lossless` форматы.

## ЛИТЕРАТУРА

1. <http://msdn.microsoft.com/ru-ru/windows/apps/>.
2. Лутай С., Байдачный С. «Windows 8 для C# разработчиков», Издательство: Самиздат, 2011г., 278с
3. С. В. Пугачев, А. М. Шериев, К. А. Кичинский «Разработка приложений для Windows 8 на языке C#» — СПб.: БХВ-Петербург, 2013. — 416 с.: ил. —(Профессиональное программирование)