Гурский Станислав Михайлович
бакалавр, студент
Савицкий Вадим Сергеевич
бакалавр, студент
Жаркова Оксана Михайловна
канд. физ.-мат. наук, доцент
ФГБОУ ВО «Кубанский государственный университет»
г. Краснодар, Краснодарский край

ОСОБЕННОСТИ ИНТЕГРАЦИИ БИБЛИОТЕКИ OPENXML В СОВРЕМЕННЫЕ ИКТ-РЕШЕНИЯ

Аннотация: *в статье описываются особенности интеграции библиотеки Open XML SDK, которая служит эффективной альтернативой автоматизации Microsoft Office через COM-объекты при работе с Word. Приведен детальный анализ внутренней структуры формата.docx, который представляет собой ZIP-архив, содержащий XML-файлы с разметкой WordprocessingML. Описаны базовые принципы работы с библиотекой в среде.NET C#, от установки пакета и подключения пространства имен до формирования документа со сложной структурой. Приведены практические примеры создания документа учета оборудования, демонстрирующие типовые сценарии программной генерации документов.*

Ключевые слова: *программная генерация документов, документооборот, автоматизация создания документов, ZIP-архив, серверные приложения, форматирование текста, шаблонизация документов.*

Введение.

Документы формата Microsoft Word занимают центральное место в деятельности организаций. Договоры, отчеты, служебные записки, техническая документация – все это создается и редактируется в файлах формата.docx. Где в свою очередь перед разработчиком встает вопрос по автоматизации создания таких до-

кументов, наполнением их данными из баз данных или генерации сотни однотипных бланков. Решение такого рода задач требует установленного офисного пакета на сервере, от чего возникает огромное число проблем с многопоточностью, утечками памяти и нестабильностью при длительной работе. Более того, подход через COM-объекты категорически не рекомендуется корпорацией Microsoft для серверных решений. Альтернативой в данном случае выступает Open XML SDK – библиотека с открытым исходным кодом, предоставляющая прямой доступ к внутренней структуре документов формата.docx и не только. Этот формат по своей структуре является Zip-архивом, содержащим набор XML-файлов. Библиотека позволяет работать с этими файлами посредством программного кода, без запуска самого Word.

Цель данной статьи – показать возможности библиотеки OpenXML применительно к Word-документам. Объяснить базовые принципы работы с библиотекой, создания документов с нуля, а также разобрать практические примеры в приложениях на платформе.Net C#.

Материал ориентирован на разработчиков, желающих освоить производительный способ программной работы с документами в серверных и десктопных приложениях. Знание основ XML и умение использовать их на практике является важной частью для опытного разработчика.NET C#.

Внутреннее устройство.docx-файлов.

Для понимания принципов работы библиотеки OpenXML необходимо, разобраться в том, как физически устроен документ формата.docx. В отличии от бинарного формата.doc, который хранит в проприетарной, недокументированной структуре, современный формат основан на открытых стандартах и прозрачной архитектуре [1].

Ключевая особенность файлов формата.docx – это обычный ZIP-архив. Если взять любой документ, созданный в Word и переименовать его, заменив расширение на.zip, то этот файл можно открыть любым архиватором, где внутри будет организована строго иерархическая структура папок и XML-файлов, представленный на рисунке 1.

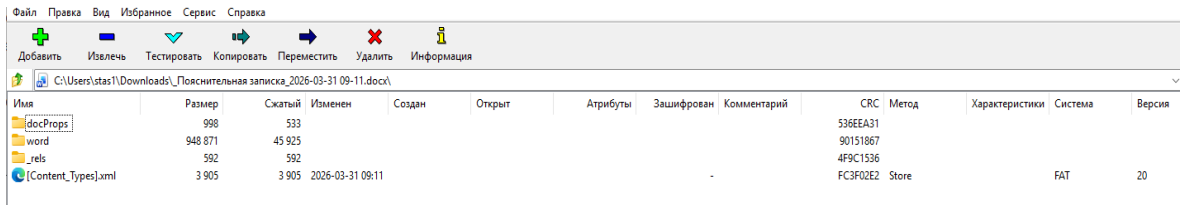

Рис. 1. Иерархическая структура папок и XML-файлов

Использование ZIP-архивов дает несколько преимуществ:

– сжатие данных – все компоненты документа автоматически сжимаются, что делает файлы формата.docx значительно меньше по размеру по сравнению с предшественником.doc;

– модульность – каждый логический компонент документа хранится в отдельном файле архива. Это позволяет обращаться только к нужным частям документа, а не разбирать весь документ [2];

– простота доступа – ZIP является общедоступным форматом, любой разработчик может получить доступ к содержимому документа, просто распаковав архив.

Язык разметки WordprocessingML.

Если ZIP-архив является контейнером, то XML – является языком, на котором написан сам документ. Все файлы с содержимым, такие как Document.xml, styles.xml и другие, представляют собой XML-документы [3].

Каждый элемент документа описывается с помощью специальных XML-элементов из словаря WordprocessingML:

– элемент <w: document> – является корневым для основной части документа;

– элемент <w: body> – содержит основное тело документа;

– абзац кодируется элементом <w: p>;

– отдельный элемент текста с единым форматированием находится внутри элемента <w: r>;

– текст находится внутри элемента <w: t>.

Таким образом, иерархия элементов строго соответствует структуре документа, где абзацы содержат прогоны текста, а прогоны содержат непосредственно текст [4]. Пример структуры документа представлен на рисунке 2.

Рис. 2. Пример структуры документа

Создание документа в C#.

Процесс создания документов с помощью Open XML SDK начинается с установки библиотеки и добавления пакета в проект через команду dotnet add package DocumentFormat.OpenXML или через менеджера пакетов NuGet в Visual Studio [1].

Затем нам необходимо подключить пространство имен, а именно в файл с кодом добавляются необходимые using-директивы:

– using DocumentFormat.OpenXml;

– using DocumentFormat.OpenXml.Packaging;

– using DocumentFormat.OpenXml.Wordprocessing.

Ключевым классом для работы с документом является WordprocessingDocument. Он представляет собой Open XML-пакет и предоставляет доступ ко всем частям документа. Пример создания простейшего документа C# представлен на рисунке 3.

```
// Создание нового документа по указанному пути
using (WordprocessingDocument wordDocument = Wordpro
cessingDocument.Create(filePath, WordprocessingDocum
entType.Document))
{
    // Добавление основной части документа
    MainDocumentPart mainPart = wordDocument.AddMain
DocumentPart();

    // Инициализация структуры документа
    mainPart.Document = new Document();
    Body body = new Body();
    mainPart.Document.Append(body);

    // ... добавление содержимого
}
```

Рис. 3. Пример создания простейшего документа C#

Содержимое документа добавляется в объект Body. Каждый элемент текста – это вложенная иерархия:

– Paragraph paragraph = new Paragraph() – создание абзаца;

– Run run = new Run() – создание прогона текста;

– Text text = new Text(«Привет, OpenXML!") – создание текстового узла [4].

Для форматирования внешнего вида текста используется свойства объекта RunProperties, представленный на рисунке 4.

```
Run run = new Run();

// Жирное начертание
run.RunProperties = new RunProperties(new Bold());

// Курсив
run.RunProperties = new RunProperties(new Italic());

// Цвет текста (в формате RGB)
run.RunProperties = new RunProperties(new Color() {
Val = "FF0000" });

// Размер шрифта (в полупунктах, т.е. 24 = 12pt)
run.RunProperties = new RunProperties(new FontSize()
{ Val = "24" });
```

Рис. 4. Свойства объекта RunProperties

Типовой сценарий формирования документа выглядит так:

– получение данных – из базы данных, API, пользовательской формы;

– загрузка или создание шаблона – создается пустой документ, либо создается документ с готовым шаблоном и местами подстановки данных;

– программное наполнение – данные подставляются в нужные места документа, формируются таблицы, добавляются изображения, применяются стили;

– сохранение документа – может быть реализовано на диске, сервере или на сторонние сервисы [4].

Пример формирования готового документа учета оборудования представлен на рисунке 5.

Номер	Наименование	Код продукции	Тип, марка	Завод изготовитель	Ед. измерения	Кол.	Примечание
1	2	3	4	5	6	7	8
	Оборудование						
1	Насос центробежный	НЦ-100/200	КМ 80-50-200	ООО «Гидромаш», г. Калуга	шт.	3	Для систем отопления
2	Электродвигатель	АИР-132М4	15 кВт, 1500 об/мин	ПАО «Сталь», г. Владимир	шт.	2	С частотным регулятором
3	Клапан шаровой	КШ-50Ф	Ду 50, Ру 16	АО «Арматура», г. Чехов	шт.	10	С электроприводом
4	Теплообменник пластинчатый	ТП-40-1.0	40 кВт, нерж. сталь	ООО «Термосила», г. Самара	шт.	1	Разборный
5	Манометр	МП-100	0-16 кгс/см²	ЗАО «Манометр», г. Москва	шт.	8	С демпферной жидкостью
6	Фильтр сетчатый	ФС-150	Ду 150, Ру 10	ООО «Фильтротех», г. Ярославль	шт.	2	-
7	Задвижка клиновая	ЗКЛ-200	Ду 200, Ру 16	АО «Трубодеталь», г. Санкт-Петербург	шт.	1	С обрезиненным клином
8	Регулятор давления	РД-50	Ду 50, Ру 16	ООО «Регулятор», г. Екатеринбург	шт.	4	Прямого действия

Дата | Подпись

Разработал			20.04.26	Содержимое склада	Стадия	Лист	Листов
Проверил			20.04.26		П	1	1
					ООО "111"		
Нач. смены.			20.04.26				

Рис. 5. Формирование документа учета оборудования

Заключение.

Был проведен детальный анализ внутреннего устройства формата.docx. Понимание этой архитектуры является фундаментом для эффективной работы с библиотекой, с возможностью точечно изменять нужные компоненты документа, не затрагивая остальные части. И на практическом примерах была продемонстрирована работа с библиотекой Open XML SDK в среде.NET C#. Где показывались шаги от установки библиотеки до реализации документов с таблицами подстановками данных из внешних источников.

Список литературы

1. Microsoft Learn. Open XML SDK. URL: https://learn.microsoft.com/en-us/office/open-xml/open-xml-sdk (date of request: 29.04.2026).

2. ISO/IEC 29500:2008. Information technology – Document description and processing languages – Office Open XML File Formats. – В 4 ч. – Geneva: ISO/IEC, 2008.

3. Walkenbach J. Office 2007 Bible / J. Walkenbach, D. Tyree. Indianapolis: Wiley Publishing, 2007. 808 p.

4. Aspose. Format file DOCX. URL: https://docs.aspose.net/file-formats/id/docx/ (date of request: 29.04.2026).