

## Генерация отчётов с использованием Aspose в приложении на платформе .NET

*Фатеенков Данила Витальевич*

*Приамурский государственный университет имени Шолом-Алейхема  
Студент*

### Аннотация

В статье рассмотрен модуль Aspose для генерации текстовых документов в формате docx (а также других форматах). Данный модуль используется при разработке программного обеспечения на платформе .NET и языке программирования C#. Aspose позволяет автоматизировать процесс генерации отчётов и упростить работу с документами различных форматов (в том числе doc, docx и pdf).

**Ключевые слова:** C#, .NET, Aspose, отчёты, генерация отчётов

### Generating reports using Aspose in a .NET application

*Fateenkov Danila Vitalievich*

*Sholom-Aleichem Priamursky State University  
Student*

### Abstract

This article describes the Aspose module for generating text documents in docx format. This module is used when developing software on the .NET platform and C# programming language. Aspose allows to automate the process of report generation and simplify the work with documents of different formats (including doc, docx and pdf).

**Keywords:** C#, .NET, Aspose, reports, report generation

## 1. Введение

### 1.1 Актуальность

В настоящее время работа с информацией, которая содержится в базах данных на стороне сервера, зачастую сопровождается и созданием отчётности по этой же информации. Отчётность может быть представлена в разных видах: это могут быть таблицы, содержащие информацию из базы данных; могут быть различные статистические данные о работе БД и информации, содержащейся в базе; могут быть логи работы сервера при отправке запроса к базе данных.

Какой бы отчёт не формировался в итоге, процесс генерации необходимо автоматизировать. Необходимость заключается в том, что потребность в отчётах может быть еженедельной или ежедневной (зависит от требований к системе), из-за чего вручную составлять документы очень сложно.

Для таких целей уже созданы различные библиотеки или модули, которые позволяют создать отчёт либо на основе заранее заготовленных шаблонов, либо на основе функций генерации стилей внутри документа. Примером такого модуля является Aspose, который предлагает широкий функционал: от создания таблиц в формате xls до обработки запросов к различным протоколам.

## 1.2 Обзор исследований

Р.А. Соболевский и Т.А. Крамаренко в своей статье рассмотрели использование объектной модели MS Word для работы с документами расширения doc из приложения, которое было разработана с применением .NET архитектуры [1].

А.С. Вискова провела анализ библиотек для работы с Word документами в .NET приложении [2]. В статье рассматривались следующие библиотеки: Spire.Doc, Open XML SDK, EasyDox и др.

С.Г. Глушко и Ю.Ю. Шатилов описали процесс разработки модуля для формирования документов с расчётами стержневых металлических конструкций [3]. Для разработки использовались Spire.Doc netDxf, а также модуль реализован на платформе .NET.

## 1.3 Цель исследования

Цель – автоматизировать процесс генерации отчётов в приложении, созданном с помощью языка программирования C#, по работе с информацией из базы данных.

## 2. Материалы и методы

Для реализации поставленной цели используется язык программирования C# и модуль Aspose.

## 3. Результаты и обсуждения

Aspose предлагает большое количество модулей для работы с различными форматам:

1. Words – модуль для работы с docx;
2. PDF – модуль для работы с pdf;
3. Cells – модуль для работы с таблицами формата xls;
4. Slides – модуль для работы с презентациями PowerPoint;
5. HTML – модуль для работы с разметкой HTML.

Выше представлен неполный список, так как количество модулей Aspose в NuGet довольно большое и нет необходимости описывать каждый в контексте данной статьи. Для создания отчётов достаточно использования Words модуля.

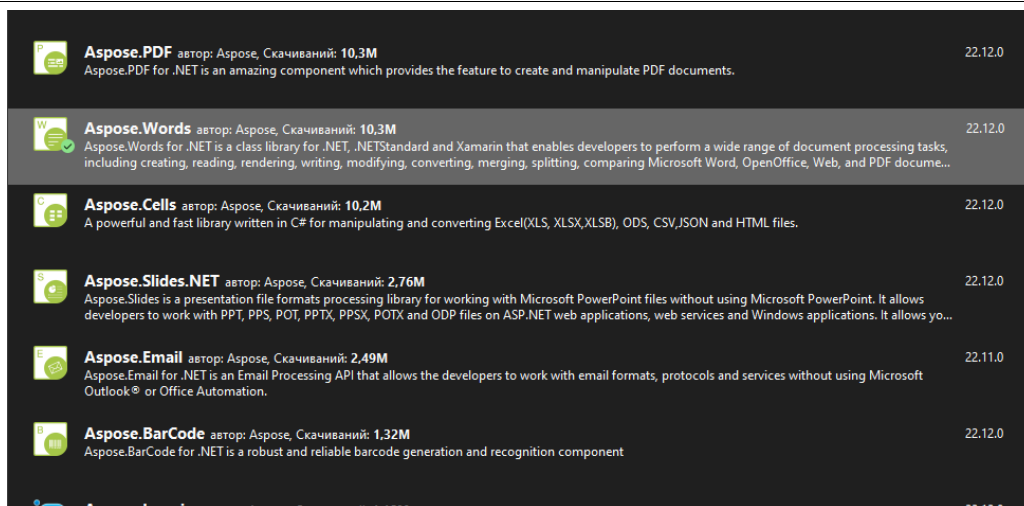


Рисунок 1. Необходимый модуль в NuGet (Visual Studio)

Создание отчёта начинается с создания документа. Необходимо создать объект класса Document следующим образом:

```
var doc = new Document ();
```

Объект данного класса нужен в первую очередь для того, чтобы была возможность записать отчёт в него. В документе можно изменять метаданные, добавлять и удалять (а также изменять) содержимое, работать со стилями наполнения документа, искать необходимые данные и отслеживать изменения, внесённые в документ. Любое взаимодействие с документом заканчивается выполнением функции «Save»:

```
doc.Save ("*название документа*");
```

Объект класса Document содержит следующие методы для работы с информацией:

1. AcceptAllRevisions – сохраняет все внесённые изменения в документ;
2. AppendChild – добавляет узел в конец листа узлов, принадлежащих указанному узлу;
3. AppendDocument – добавляет информацию из указанного документа в конец текста документа, с которым ведётся работа;
4. Cleanup – очищается неиспользуемая информация из документа;
5. Clone – создаётся копия документа;
6. Compare – сравниваются 2 документа;
7. CopyStylesFromTemplate – копируются элементы стиля из другого документа (шаблона);
8. CreateNavigator – создаётся элемент IXPathNavigable. Нужно для операционной системы и данная функция не используется часто;
9. EnsureMinimum – проверка на наличие информации в документе (если она отсутствует, то создаётся один параграф);

10. ExtractPages – возвращает объект класса Document, представляющий информацию с указанного диапазона страниц;
11. GetAncestor – получение первого «предка» того или иного объекта;
12. GetChild – получение узла, который связан с тем или иным типом или объектом;
13. GetEnumerator – обеспечивает поддержку итерации для каждого стиля над дочерними узлами этого узла;
14. GetPageInfo – получает информацию о странице (размер, ориентация и т.д.);
15. GetText – получает текст заданного узла;
16. ImportNode – импортирует узел из другого документа в документ, который на данный момент редактируется;
17. IndexOf – возвращает индекс указанного ребёнка узла в массиве детей этого же узла;
18. InsertAfter – добавляет узел после референсного узла;
19. InsertBefore – добавляет узел перед референсным узлом;
20. Print – печатает документ;
21. Remove – удаляет указанных детей из узла (либо сам документ);
22. Save – сохраняет документ;

Методов много, но не со всеми есть необходимость работать, если необходимо составить отчёт, в котором неважно оформление.

После создания документа, необходимо создать объект класса DocumentBuilder, который предназначен для составления документа:

```
DocumentBuilder builder = new DocumentBuilder(doc);
```

Это один из самых больших классов в модуле Words. Он позволяет настраивать стиль содержимого, добавлять и удалять новую информацию различного формата (начиная с обычного текста и заканчивая изображениями и таблицами). Также класс позволяет работать с HTML-кодом и обрабатывать информацию на графиках. На данный момент DocumentBuilder работает со следующими данными:

1. Текст;
2. Таблица;
3. Диаграмма;
4. Чекбокс и комбобокс;
5. Изображение;
6. HTML;
7. OLE объект;

Для всех объектов, перечисленных выше, реализованы методы создания, перемещения и удаления. Также DocumentBuilder позволяет настроить шрифт, формат таблиц, курсив, жирный шрифт, формат листов. Для этого выделены отдельные свойства класса. В контексте рассматриваемой темы будут использоваться преимущественно таблицы.

Для того, чтобы создать таблицу, необходимо вызвать следующий метод:

```
builder.StartTable();
```

Для того, чтобы создать строку, достаточно вставить клетку в таблицу:

```
builder.InsertCell();
```

Для записи информации в ячейки (записывать информацию нужно строго перед вызовом новой ячейки) используется метод Write:

```
builder.Write("Some Info");
```

Чтобы перейти на следующую строку, нужно закончить редактирование нынешней через метод EndRow:

```
builder.EndRow();
```

После завершения составления таблицы, необходимо вызвать метод EndTable:

```
builder.EndTable();
```

Этого функционала хватает для составления таблиц в документах doc формата. Пример создания таблицы ниже:

```
Document doc = new Document();
DocumentBuilder builder = new DocumentBuilder(doc);
builder.StartTable();
for (int i=0;i<rows;i++) {
    for (int j=0;j<cols;j++) {
        builder.InsertCell();
        builder.Write(i + j);
    }
    builder.EndRow();
}
builder.EndTable();
doc.Save(name);
```

В методе Write можно указать информацию из БД, которую необходимо сохранить в документе. Таким образом будет создан отчёт об информации, хранящейся в той или иной таблице базы данных, с которой работает приложение. Ниже представлен код, создающий таблицу с информацией о студентах:

```
builder.WriteLine("Студенты");
builder.InsertCell(); builder.Write("id");
builder.InsertCell(); builder.Write("Имя");
builder.InsertCell(); builder.Write("Возраст");
builder.InsertCell(); builder.Write("Номер
телефона");
builder.InsertCell(); builder.Write("Номер
группы");
builder.EndRow();
foreach(var student in slist)
{
    builder.InsertCell();
builder.Write(student.id.ToString());
    builder.InsertCell();
builder.Write(student.name.ToString());
    builder.InsertCell();
builder.Write(student.age.ToString());
    builder.InsertCell();
builder.Write(student.number.ToString());
    builder.InsertCell();
builder.Write(student.groups_.id.ToString());
    builder.EndRow();
}
builder.EndTable();
```

Главный недостаток `DocumentBuilder` – большой объём кода на выходе. Можно указать заранее созданный шаблон, что сократит длину кода. `Aspose` также поддерживает кириллицу и на выходе, как правило, нет проблем с кодировкой.

Таким образом, получилось создать таблицу с данными из таблицы `MySQL` и сохранить в документе формата `doc`.

В статье был рассмотрен `Aspose`, основные методы и параметры классов `Document` и `DocumentBuilder`, а также рассмотрен процесс составления таблиц в документах с данными из таблицы базы данных.

### Библиографический список

1. Соболевский Р.А., Крамаренко Т.А. MICROSOFT OFFICE и платформа .NET FRAMEWORK // Информационное общество: современное состояние и перспективы развития: Сборник материалов IX студенческого международного форума, Краснодар, 26–30 июня 2017 года. Краснодар: Кубанский государственный аграрный университет имени И.Т. Трубилина, 2017. С. 339-342.
2. Вискова А.С. Сравнительный анализ библиотек .NET для генерации

- документов Word // Россия молодая: Сборник материалов XIII Всероссийской научно-практической конференции с международным участием, Кемерово, 20–23 апреля 2021 года / Редколлегия: К.С. Костиков (отв. ред.) [и др.]. – Кемерово: Кузбасский государственный технический университет имени Т.Ф. Горбачева, 2021. С. 31505.1-31505.4.
3. Глушко С.Г., Шатилов Ю.Ю. Разработка программного модуля для формирования чертежей и отчетов по результатам проверки прочности металлических конструкций в двумерной постановке // Инженерный вестник Дона. 2021. № 4 (76). С. 97-104.
  4. Aspose API Reference URL: <https://reference.aspose.com> (дата обращения: 31.01.2023).
  5. Aspose URL: <https://www.aspose.com> (дата обращения: 31.01.2023).