

# DoDoc

Руководство пользователя

Москва, 2011 г.

## ОГЛАВЛЕНИЕ

1.	Введение.....	3
2.	Входные файлы.....	3
3.	Правила оформления шаблонов.....	3
3.1	Простая подстановка текста.....	3
3.2	Строка таблицы.....	3
3.3	Изображение.....	4
3.4	Раздел условного включения.....	4
4.	Правила формирования данных.....	5
4.1	Простая подстановка текста.....	5
4.2	Дерево представления таблицы.....	5
4.3	Список путей к подставляемым изображениям.....	6
4.4	Форматирование подставляемого текста.....	6
5.	Инструментарий.....	7
5.1	odt2pdf.....	7
5.2	odg2wmf.....	7
5.3	DoXML.....	7
5.4	DoDoc_folder_printer.....	7
5.5	ods_parser.....	8
6.	Полезные мелочи.....	8
6.1	«Исходники» .odt.....	8
6.2	Удаление абзацев, которые «не удаляются».....	9
6.3	Если маркер не заменяется.....	10

# 1. ВВЕДЕНИЕ

DoDoc – это инструмент для автоматизированного формирования шаблонной документации. DoDoc реализован на языке python. В качестве офисного пакета используется Open Office.

В данный момент DoDoc является консольной утилитой. Для получения справки по использованию, выполните команду:

```
python DoDoc.py --help
```

# 2. ВХОДНЫЕ ФАЙЛЫ

DoDoc работает по принципу подстановки *данных* в *шаблон*. *Шаблоном* является специальным образом размеченный .ODT файл. Для передачи *данных* используется формат XML.

Для работы DoDoc необходим установленный пакет Open Office версии не ниже 3.3.0. На ОС Windows в переменной среды PATH должен быть прописан путь к папке program. Например:

```
C:\Program Files\OpenOffice.org 3\program
```

# 3. ПРАВИЛА ОФОРМЛЕНИЯ ШАБЛОНОВ

Для определения мест в шаблоне, в которые требуется подставить *данные* необходимо использовать *маркеры*. *Маркером* является слово, заключенное в фигурные скобки. Разрешается использовать латинские буквы в нижнем регистре, цифры, типографскую землю («\_») и точку. Подставленный текст будет иметь такой же формат, как и маркер.

*Маркеры* можно разделить на 4 типа:

1. Простая подстановка текста;
2. Строка таблицы;
3. Изображение;
4. Раздел условного включения.

## 3.1 Простая подстановка текста

Используется, когда необходимо вставить текст в документ в одном, или нескольких местах.

Например:

```
Название документа: {document_title}.
```

## 3.2 Строка таблицы

Когда требуется вставлять в документ переменное количество однотипных данных, рекомендуется использовать таблицы. Табличные маркеры имеют следующий формат:

`{table_name.variable_name}`

Где:

- `table_name` — имя таблицы;
- `variable_name` — имя элемента таблицы.

Например, в шаблоне может находиться такая таблица:

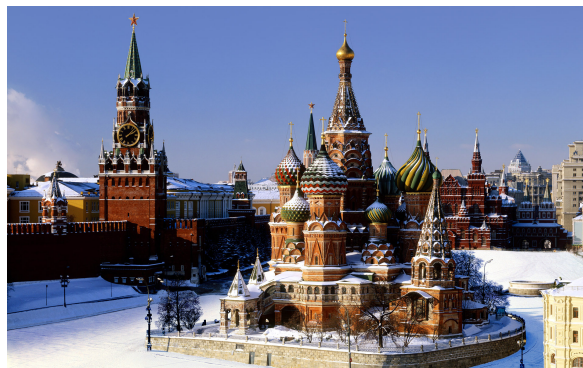
Идентификатор	Наименование	Масштаб
<code>{variables.identifier}</code>	<code>{variables.name}</code>	<code>{variables.scale}</code>

DoDoc заменит строку таблицы, содержащую табличные маркеры на данные из XML файла.

Допускается использовать вложенные таблицы. В ячейки таблиц можно вставлять разделы условного включения и маркеры изображений.

### 3.3 Изображение

Для подстановки в документ изображений используется специальный маркер. В шаблоне необходимо вставить картинку (Вставка → Изображение → Из файла...), и в её свойствах задать имя в формате маркера, но без фигурных скобок, например «radius\_vector».



Все параметры отображения картинки в шаблоне (положение, размер, обтекание, привязка, и т. д.) перейдут на вставленные картинки.

### 3.4 Раздел условного включения

Если требуется вставлять часть документа только при наличии некоторых данных, можно использовать маркеры разделов условного включения. Для вставки такого маркера в шаблон, выберите в главном меню Вставка → Раздел... и задайте имя новому разделу в следующем формате:

`have_marker`

или

`have_no_marker`

Где `marker` — имя маркера, в зависимости от наличия которого в наборе данных

будет вставляться данный раздел.

## 4. ПРАВИЛА ФОРМИРОВАНИЯ ДАННЫХ

Корневой узел XML файла должен называться DoDoc и может содержать следующие элементы:

1. Простая подстановка текста;
2. Дерево представления таблицы;
3. Список путей к подставляемым изображениям.

### 4.1 Простая подстановка текста

Узел, содержащий подстановочный текст.

```
<document_title>
    Технические требования к алгоритму навигации
</document_title>
```

### 4.2 Дерево представления таблицы

Структуру представления таблицы удобно описать на примере. Пусть требуется вставить следующие данные в таблицу из примера главы:

Идентификатор	Наименование	Масштаб
ri[3]	Радиус-вектор разгонного блока	2 <sup>3</sup>
vi[3]	Вектор скорости разгонного блока	2 <sup>17</sup>

Тогда XML будет выглядеть следующим образом:

```
<DoDoc>
  <variables>
    <ROW>
      <identifier>
        ri[3]
      </identifier>
      <name>
        Радиус-вектор разгонного блока
      </name>
      <scale>
        2^3
      </scale>
    </ROW>
    <ROW>
      <identifier>
        vi[3]
      </identifier>
      <name>
        Вектор скорости разгонного блока
      </name>
```

```

        <scale>
            2^17
        </scale>
    </ROW>
</variables>
</DoDoc>

```

Узел с именем, соответствующим названию таблицы в шаблоне, содержит элементы ROW (регистр важен), которые, в свою очередь, содержат узлы с подстановочным текстом.

### 4.3 Список путей к подставляемым изображениям

Структура дерева подстановки изображений описывается следующим примером:

```

<radius_vector>
    <IMAGE>ri_1.png</IMAGE>
    <IMAGE>ri_2.png</IMAGE>
</radius_vector>

```

Узел с именем, соответствующим имени Изображения в шаблоне, содержат узлы IMAGE (регистр важен), которые, в свою очередь, содержат узлы с путями к графическим файлам. В данный момент поддерживаются следующие форматы:

- gif
- png
- wmf
- odg (Каждый лист вставляется отдельной картинкой)

### 4.4 Форматирование подставляемого текста

По умолчанию подставляемый текст принимает тот же стиль, что был у маркера. Для более тонкого выделения слов используется HTML-подобный язык разметки. В текущей версии поддерживаются следующие тэги:

- **<b>** – **жирное начертание;**
- **<i>** – *наклонное начертание;*
- **<u>** – подчеркивание;
- **<red>** – **красный цвет текста;**
- **<green>** – **зеленый цвет текста;**
- **<blue>** – **синий цвет текста;**
- **<sup>** – **верхний** индекс;
- **<sub>** – **нижний** индекс.

Для вставки разрыва строки в подставляемый текст необходимо вставить в текст следующую последовательность:

```
#BR#
```

## 5. ИНСТРУМЕНТАРИЙ

### 5.1 odt2pdf

Консольная утилита odt2pdf служит для конвертации odt файлов в pdf. Скрипт принимает 2 аргумента: имя входного odt файла и имя выходного pdf файла.

### 5.2 odg2wmf

Консольный конвертор .odg в набор .wmf файлов (по 1 на каждый лист). Также существует аналогичная утилита odg2png для получения .png файлов, однако рекомендуется использовать именно формат WMF, так как файлы занимают меньше места на диске и отсутствуют проблемы со сглаживанием шрифтов.

### 5.3 DoXML

Как правило, данные перед подстановкой в шаблон необходимо преобразовать. Программа-конвертер, формирующая входной XML -файл для DoDoc может быть написана на любом языке программирования. Однако, для языка Python существует специализированный вспомогательный класс — `DoDoc.DoXML`.

`DoDoc.DoXML` представляет из себя класс-обертку для универсального `xml.dom.minidom`. С его помощью, пример из главы 4.2 можно получить следующим кодом:

```
import DoDoc
doc = DoDoc.DoXML.DoXML() # Создать объект класса
doc.table(u'variables')   # Создать таблицу
doc.row()                 # Добавить строку таблицы
doc.text(u'identifier', u'ri[3]') # Добавить значение
doc.text(u'name', u'Радиус-вектор разгонного блока')
doc.text(u'scale', u'2^3')
doc.row()                 # Добавить следующую строку
doc.text(u'identifier', u'vi[3]')
doc.text(u'name', u'Вектор скорости разгонного блока')
doc.text(u'scale', u'2^17')
doc.save(u'variables.xml') # Сохранить XML в файл
```

### 5.4 DoDoc\_folder\_printer

После генерации большого количества документов, иногда возникает задача распечатать их. Утилита `DoDoc_folder_printer` позволяет последовательно отправить на принтер все документы в выбранной папке. Использование:

```
python DoDoc_folder_printer.py <folder_path>
```

## 5.5 ods\_parser

Класс для получения данных из таблиц OpenOffice Calc. Метод `parseODS` возвращает словарь (dict) листов .ods документа. Доступ к содержимому ячеек осуществляется по индексам двумерного массива. Пример использования:

```
import DoDoc.ods_parser
# CARD_all.ods - карточки ПЗ для РБ «Бриз-М»
tables = DoDoc.ods_parser.parseODS('CARD_all.ods')
table_bocp = tables[u'Карточка БОЦП']
bocp_row_13 = table_bocp[13]
bocp_cell_B13 = bocp_row_13[1]
print bocp_cell_B13
```

Результатом выполнения данного скрипта будет следующая строка:

```
>>> Географическая широта места старта
```

## 6. ПОЛЕЗНЫЕ МЕЛОЧИ

### 6.1 «Исходники» .odt

Open Document Format (ODF), используемый в файлах .odt представляет собой zip-архив, примерно такой структуры:

```
Configurations2\
...
META-INF\
    manifest.xml
Pictures\
    100000000000007800000004B07D961EC8.jpg
Thumbnails\
    thumbnail.png
layout-cache
mimetype
manifest.rdf
content.xml
meta.xml
settings.xml
styles.xml
```



Файл `content.xml` содержит почти все данные документа. Кратко, его формат выглядит следующим образом:

```
<office:document-content>
  <office:font-face-decls>
    <!--Объявление шрифтов-->
  </office:font-face-decls>
  <office:automatic-styles>
    <!--Объявление стилей-->
  </office:automatic-styles>
  <office:body>
    <!--Содержимое документа-->
  </office:body>
</office:document-content>
```

Иногда, при формировании *шаблона* может понадобиться поправить этот XML файл, хотя такие случаи редки и средств Open Office достаточно, чтобы выполнить практически все возможные операции над документом.

## 6.2 Удаление абзацев, которые «не удаляются»

При вставки таблицы Open Office добавляет абзац после таблицы. Зачастую этот абзац может привести к неожиданным последствиям в сформированном документе. Например, если после таблицы стоит разрыв (переход на новую страницу) и таблица заканчивается в конце листа, то абзац окажется на следующей странице, где будет вставлен разрыв. В результате в документе окажется пустая страница.

Для удаления этого абзаца необходимо установить курсор в последнюю ячейку таблицы и нажать комбинацию клавиш:

`Ctrl↓ Shift↓ Delete↓ ↑↑↑`

Если необходимо, наоборот, добавить абзац, используется комбинация

`Alt↓ Enter↓ ↑↑`

при курсоре установленном в последнюю ячейку. Аналогично добавляется абзац перед таблицей, но в этом случае курсор должен быть установлен в первой ячейке таблицы.

В случае, когда в шаблоне используются вложенные таблицы могут возникнуть трудности: абзац после вложенной таблицы таким способом удалить не получится. В этом случае необходимо править `content.xml`. Просто найдите нужное место и удалите тэг

```
<text:p text:style-name="Text_body" />
```

Затем, сохраните исправленный `content.xml` в архив `.odg` файла.

### 6.3 Если маркер не заменяется

Иногда бывает так, что маркер в шаблоне стоит, и в XML есть замена для него, но подстановка не происходит. Скорее всего, это связано с тем, что в форматировании маркера в *шаблоне* используется несколько стилей. Open Office грешит вставкой стилей символов там, где пользователь этого не ожидает. Если вы столкнулись с такой проблемой, у вас есть 2 способа исправить *шаблон*:

1. Удалить маркер и напечатать его снова (При непрерывной печати стили к символам не применяются);
2. Открыть `content.xml` и удалить лишние тэги форматирования.