

---

# OutWiker Documentation

*Выпуск 2.0.0*

Eugene Ilin (aka Jenyay)

июн. 18, 2017



<b>1 English documentation</b>	<b>1</b>
1.1 English documentation . . . . .	1
<b>2 Russian documentation</b>	<b>7</b>
2.1 Документация на русском языке . . . . .	7
<b>3 Indices and tables</b>	<b>51</b>



## English documentation

### Sources structure

---

#### План

Describe the sources structure.

---

### Page storing format

---

#### План

Describe the page storing format

---

## Version History

### Current developing version

#### 2.0.0.816 RC (April 25, 2017)

- Removed Italian localization.

#### 2.0.0.814 beta (April 03, 2017)

- Added the “subpath” parameter to main window title.

- Some GUI improvements.
- Bug fixes.

#### **2.0.0.812 beta (March 18, 2017)**

- Hot keys GUI improvements.
- Update help.
- Add Ubuntu 17.04 Zesty Zapus supporting.

#### **2.0.0.810 beta (February 24, 2017)**

- Splitted the long toolbars.
- Bug fixes.

#### **2.0.0.808 beta (January 26, 2017)**

- Added the ability to add alias to page.
- Changed GUI to set hot keys.
- Added options to change editor margin color.
- Added options to change background color of the selected text.

#### **2.0.0.806 beta (November 14, 2016)**

- Added the menu items to move cursor in text.
- Added the menu items to copy / cut the current line to clipboard.
- Bug fixes.

#### **2.0.0.804 dev (October 20, 2016)**

- Fixed deb package for Ubuntu 16.10.

#### **2.0.0.802 dev**

- Internal changes.

#### **2.0.0.800 dev (August 20, 2016)**

- Added the Swedish translation.
- Added the “-page, -p” command line parameter to select page when starting.
- Added the “-normal” command line parameter to disable minimizing when starting.
- Added the “-debug” command line parameter to run in the debug mode.

- Bug fixes.

### **2.0.0.798 dev (July 27, 2016)**

- Restored availability the icon in the system tray on Linux.
- The editor adds the ability to join lines with the hot key or menu item.
- The editor adds the ability to remove word to beginning / ending with the hot key or menu item.
- The editor adds the ability to decrease nesting level of the list items with the hot key or menu item.
- Changed tabs style.

### **2.0.0.796 dev (July 02, 2016)**

- The editor adds the ability to duplicate the current line with hot key or menu item.
- The editor adds the ability to delete the current line with hot key or menu item.
- The editor adds the ability to move lines up / down with hot keys or menu items.
- Improved heading inserting in wiki pages.
- Build system refactoring.
- Bug fixes.

### **2.0.0.794 (May 30, 2016)**

- Fixed errors related with migration to wxPython 3.0.

### **2.0.0.792 (May 10, 2016)**

- The program now uses wxPython 3.0 library.

## **Previous versions**

### **1.9.0.790 (April 09, 2016)**

- Added spell checker.
- Created the groups for the icons.
- Now users can add custom icons.
- The choice of colors for the tag cloud.
- The choice the action when you make left or middle mouse click on the tag.
- The choice the action when you make double mouse click on the attached file.
- Added the new wiki commands for the table creation: [=(:table:), (:row:), (:cell:), (:hcell:)=].
- Improved search hot keys for the actions.
- Added new icons.

- Added new page styles.
- Now in the link inserting dialog you can select an attached file to create link to it.
- Added help.

#### 1.8.1.752 (October 20, 2014)

- Bug fixes

#### 1.8.0.750 (October 11, 2014)

- Added the ability to customize keyboard shortcuts.
- Added the ability to refer to a page using unique identifiers.
- Added the ability to navigate through the links to pages with the anchors (`{+[page://bla-bla-bla/#anchor=]+}`).
- Added the ability to use relative paths in the links on the pages (`{+[../../page 1/page 2=]+}`).
- Added the ability to change the page style for branch of the pages at the same time.
- Added the ability of search and replace on the page.
- Added the buttons “Forward” and “Back” for the return to the previous pages.
- Added the ability to change the editor colors.
- Added the ability to change the behaviour of the Home / End keys in the editor (go to begin / end of the string or the paragraph).
- Added a new tag of the wiki syntax for quoting: `[>...<]=`.
- Added the button and menu item insertion the current date and time
- Added the commands `[:(crdate:)]` and `[:(eddate:)]=` for insertion the creation and edition dates of the page respectively.
- Added the dialogs for the commands `[:(attachlist:)]`, `[:(childlist:)]` and `[:(include:)]=`.
- Added the ability to sort child pages for the creation and edition date in the command `[:(childlist:)]=`.
- On the global search page added the ability to sort child pages for the creation date.
- Added new command line parameter “-r” or “-readonly” for the opening the notes tree readonly.
- Added the popup tooltips for icons in the property dialog for page.
- Added a new styles for page design.
- Added the button and menu item for opening a folder with a attached files in a system file manager.
- Added the saving recent used page style.
- Added the saving cursor position for page before closing.
- Added the Italian localization.
- Now in the attachments panel showed the files icons.
- Changed the hyphenation algorithm (“br” tags instead of “p”).
- Now opening the notes tree and global search conduct in a separate thread.



- Now for every page saved the creation date.
- Now all HTML tags, which the wiki parser create, in a lowercase.
- Now help will be open in a separate window.
- Settings moved from `~/outwiker` more ideologically correct place (depending on the operating system).
- Added new icons for pages.
- Many accelerations.
- Bug fixes and improvements.



## Документация на русском языке

### Часто задаваемые вопросы

#### Contents

- *Часто задаваемые вопросы*
  - *Настройки программы*
    - \* *Где хранятся настройки программы?*
    - \* *Где находится папка профиля программы?*
    - \* *Как переключить OutWiker в портативный режим?*
    - \* *Как установить плагин?*
  - *Сборка программы*
    - \* *Не могу собрать OutWiker под Windows с использованием 64-битной версии Python 2.7*
    - \* *Какие пакеты требуются для запуска OutWiker из исходников под Ubuntu?*
  - *Известные проблемы*
    - \* *В Ubuntu курсор отображается в неправильном месте*
  - *Разное*
    - \* *Планируется ли версия под Android*

### Настройки программы

### Где хранятся настройки программы?

Все настройки программы находятся в файле `outwiker.ini`, который расположен в папке профиля программы. См. раздел *Где находится папка профиля программы?*. Кроме того, плагины могут создавать дополнительные файлы или папки для хранения данных внутри папки профиля программы.

### Где находится папка профиля программы?

Папка профиля программы - это папка, в которой расположен файл настроек `outwiker.ini`, файл лога работы `outwiker.log`, а также папки с плагинами (`plugins`), пользовательскими словарями для проверки орфографии (`spell`), пользовательскими стилями (`styles`) и, возможно, с другими папками, которые используются для хранения пользовательских настроек.

Если программа работает в портативном режиме (см. раздел *Как переключить OutWiker в портативный режим?*), то папка профиля программы - это папка с запускаемым файлом (под Windows это файл `outwiker.exe`). Однако по умолчанию папка профиля программы находится внутри папки профиля пользователя операционной системы.

Под Windows 7/8.x/10 это папка `C:\Users\USERNAME\AppData\Roaming\outwiker`, где “`USERNAME`” - имя пользователя в операционной системе.

Под Linux расположение папки настроек зависит от настроек операционной системы, но скорее всего это будет папка `~/.config/outwiker`, где “`~`” обозначает домашнюю папку пользователя.

### Как переключить OutWiker в портативный режим?

Портативным называется режим работы программы, в котором все настройки хранятся в той же папке, где расположен запускаемый файл программы или во вложенных папках. Наиболее часто такой режим используется под Windows. Чтобы переключить OutWiker в портативный режим, нужно создать файл `outwiker.ini` в той же папке, где расположен запускаемый файл (под Windows это файл `outwiker.exe`). Чтобы не потерять настройки, сохраненные в обычном (не портативном) режиме, файл `outwiker.ini` можно скопировать из папки профиля пользователя - см. раздел *Где находится папка профиля программы?*.

### Как установить плагин?

Для установки плагина необходимо распаковать архив со скачанным плагином в папку `plugins` внутри папки профиля программы (см. раздел *Где находится папка профиля программы?*). Обратите внимание, что в распакованный плагин представляет собой папку со множеством файлов, поэтому папка `plugins` после установки нескольких плагинов будет выглядеть примерно так:

```
outwiker
- plugins
|   - changepageuid
|   - counter
|   - datagraph
|   - ...
|   - ...
|   - ...
- outwiker.ini
- outwiker.log
```

## Сборка программы

### Не могу собрать OutWiker под Windows с использованием 64-битной версии Python 2.7

В данный момент OutWiker собирается с использованием 32-битной версии Python 2.7. Запуск программы под Python 2.7 x64 не гарантируется.

Как минимум для 64-битной версии Python 2.7 нет официальных сборок библиотеки `pyenchant`, поэтому ее придется компилировать из исходником самостоятельно. Возможно, будут другие проблемы с запуском под 64-битной версией Python.

### Какие пакеты требуются для запуска OutWiker из исходников под Ubuntu?

Требуемые пакеты зависят от номера версии Ubuntu.

Для запуска под Ubuntu 16.04 требуются следующие пакеты:

- python2.7
- python-wxgtk3.0
- python-wxversion
- libwxgtk-webview3.0-0v5
- python-pil
- python-enchaut
- python-appindicator
- python-gtk2
- ibus-gtk3

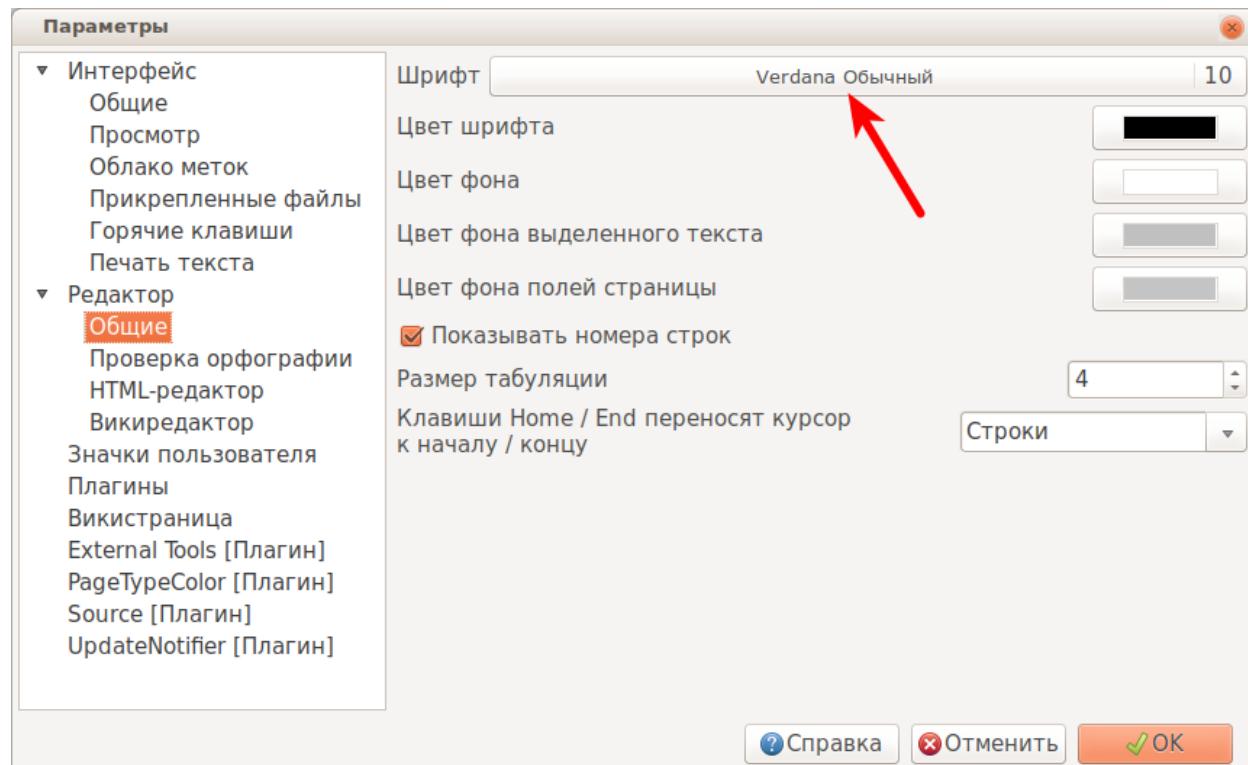
Для запуска под Ubuntu 16.10 и выше требуются следующие пакеты:

- python2.7
- python-wxgtk3.0
- python-wxversion
- libwxgtk-webview3.0-0v5
- python-pil
- python-enchaut
- python-appindicator
- python-gtk2
- python-wxgtk-webview3.0
- ibus-gtk3,

## Известные проблемы

## В Ubuntu курсор отображается в неправильном месте

В Ubuntu Linux есть проблемы с корректным отображением курсора (каретки) в поле ввода заметки с некоторыми шрифтами. Если вы видите, что курсор расположен в одной части строки, а вводимый текст появляется в другой части строки, поменяйте шрифт в окне настроек *Правка* → *Параметры*. Шрифты меняются в разделе “Редактор - Общее”. Данная проблема наблюдается с некоторыми шрифтами.



## Разное

### Планируется ли версия под Android

Очень хочется сделать версию под Android, но пока времени на это не хватает. Если кто-то готов взяться за реализацию данного проекта, то готов оказывать всяческую поддержку.

## Быстрый старт

Для получения исходного кода OutWiker вам понадобится [git](#). Также должны быть установлены [Python 2.7](#) (в данный момент Python 3.x не поддерживается) и [pip](#).

1. Чтобы получить исходный код выполните команду консоли:

```
git clone https://github.com/Jenyay/outwiker
```

Эта команда создаст папку с именем `outwiker` в текущей папке, в которую будут скачены исходные коды программы OutWiker.

2. Зайдите в эту папку с помощью команды

```
cd outwiker
```

3. Установите требуемые библиотеки с помощью команд:

```
pip install --user -r requirements.txt
pip install --user -r requirements_dev.txt
```

Если вы работаете под Windows, также необходимо выполнить следующую команду:

```
pip install --user -r requirements_win.txt
```

Если вы работаете под Linux, убедитесь, что у вас установлены требуемые пакеты:

```
python-wxgtk3.0
python-wxversion
libwxgtk-webview3.0-0v5
python-pil или python-imaging
python-enchaut
python-appindicator
python-gtk2
python-wxgtk-webview3.0
ibus-gtk3
```

4. Если все установилось без ошибок, то следующая команда должна запустить программу OutWiker:

```
fab run
```

5. Также можно запустить тесты (выполнение тестов может занять несколько минут, а под Windows в несколько раз больше времени):

```
fab test
```

## Получение исходников, сборка и тестирование

### Как скачать исходные коды OutWiker

Для получения актуальной версии исходных кодов потребуется система управления версиями [git](#).

Исходные коды расположены на [github](#). Чтобы скачать последнюю версию исходных кодов, выполните команду

```
git clone https://github.com/Jenyay/outwiker
```

В репозитории исходных кодов имеются три “долгоживущие” ветки (branch) в зависимости от степени стабильности версии программы:

**stable** Исходные коды текущей [стабильной версии](#) OutWiker.

**master** Исходные коды, находящиеся в разработке. Скорее всего в этой ветке исходники ушли чуть вперед по сравнению с текущей [нестабильной версией](#) программы.

**dev** Исходные коды, находящиеся в активной разработке, но при этом возможность, над которой идет работа еще не до конца реализована, поэтому исходные коды не влиты в ветку *master*. В этой ветке могут проваливаться некоторые тесты.

Разработка просходит следующим образом. Если коммит полностью реализует какую-то возможность или рефакторинг (возможно, после еще придется что-то дополнять, но новая функциональность появляется), то такой коммит отправляется в ветку *master*.

Если коммит только приближает реализацию возможности или начинает серию большого рефакторинга, но функциональность еще не появляется, то такой коммит отправляется в ветку *dev*. После окончательной реализации возможности ветка *dev* вливается в *master*.

Кроме того, могут создаваться короткоживущие ветки, связанные с реализацией какой-то особо большой возможности или для проверки какой-либо идеи (в последнем случае ветка часто называется *tmp* или *temp*). После удачной реализации данные ветки вливаются в *master*, а сама ветка удаляется.

Для переключения между ветками используется команда:

```
git checkout имя_ветки
```

### Структура исходных кодов

Если вы еще не скачали исходные коды, то см. раздел *Как скачать исходные коды OutWiker*.

### Файлы и папки, расположенные в корне исходных кодов

#### Папки

**.git** Служебная папка, используемая системой контроля версий *git*.

**build** Эта папка создается при сборке OutWiker под какую-либо операционную систему и при сборке архивов с плагинами. Эта папка не индексируется *git*.

**buildtools** Эта папка содержит Python-скрипты, используемые при сборке OutWiker под различные операционные системы.

**doc** Папка с документацией, которую вы сейчас смотрите.

**images** Папка с изображениями, которые не используются непосредственно в программе, но могут быть полезными при разработке. Например, логотипы OutWiker разных размеров и в разных форматах.

**need\_for\_build** Папка с файлами, которые не являются Python-скриптами, но необходимы для сборки под различные операционные системы.

**plugins** Папка с исходными кодами всех плагинов, которые прилагаются к программе OutWiker.

**profiles** Файлы, используемые для измерения скорости работы программы OutWiker и отдельных ее частей.

**src** Самая главная папка, которая содержит исходные коды OutWiker и тестов.

**test** Папка содержит файлы, используемые в тестах.

#### Файлы

**copyright.txt и LICENSE.txt** Файл с описанием лицензии, под которой распространяется программа (GPL 3) и текст лицензии GPL 3.



**fabfile.py** Файл для библиотеки `fabric`. Этот файл нужен для выполнения действий, связанных со сборкой программы под разные операционные системы, запуска тестов, обновления локализаций, выкладывания новых версий программы и плагинов на сайт. По сути это единая “точка входа” для многих действий, которые нужно делать при разработке. Также этот файл заменяет собой `Makefile`. Подробнее о том, что можно делать через этот файл см. в разделе *Выполнение команд Fabric*.

**outwiker\_setup.iss** Скрипт, описывающий создание инсталлятора под Windows с помощью инструмента `Inno Setup`.

**README** Краткое описание программы. Используется при сборке `deb`-пакета (см. раздел *Сборка под Linux*).

**readme.md** Более подробное описание программы, которое показывается в качестве описания проекта на сайте `github`

**requirements.txt** Список общих Python-библиотек, необходимых для запуска и работы программы OutWiker. Этот файл используется для установки требуемых библиотек с помощью `pip`. Подробнее см. в разделе *Инструменты, используемые при разработке*.

**requirements\_win.txt** Список Python-библиотек, необходимых для запуска и работы программы OutWiker под Microsoft Windows. Этот файл используется для установки требуемых библиотек с помощью `pip`. Подробнее см. в разделе *Инструменты, используемые при разработке*.

**requirements\_dev.txt** Список Python-библиотек, используемых при разработке программы OutWiker. Для запуска OutWiker эти библиотеки не требуются. Этот файл используется для установки требуемых библиотек с помощью `pip`. Подробнее см. в разделе *Инструменты, используемые при разработке*.

**Файлы test\_build\_\*.py** Файлы тестов системы сборки. Подробнее см. в разделе *Тестирование OutWiker*.

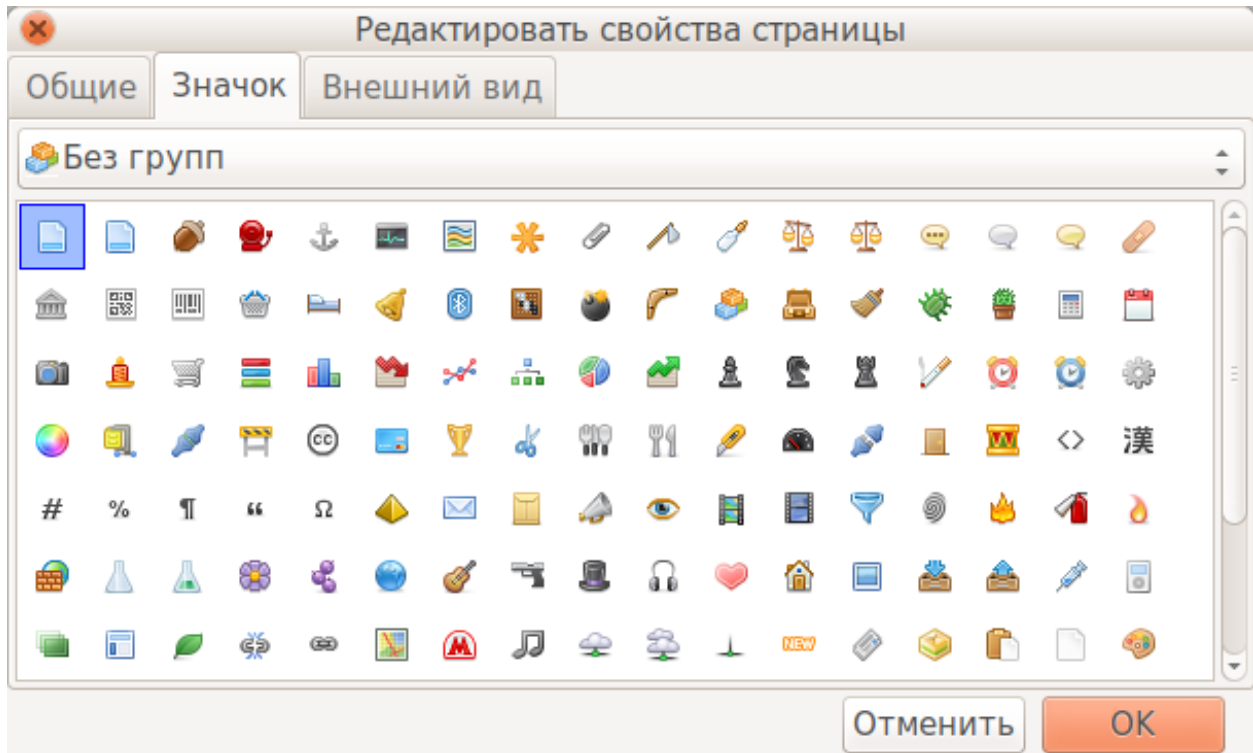
**.gitignore** Файл с описанием файлов и папок, которые не должны попадать в репозиторий исходных кодов (файлы и папки, создаваемые в процессе работы).

## Содержимое папки src

..	tests_plugin_diagrammer.py	</> versions.xml
help	tests_plugin_export2html.py	
iconset	tests_plugin_externaltools.py	
images	tests_plugin_hackpage.py	
locale	tests_plugin_htmlformatter.py	
outwiker	tests_plugin_htmlheads.py	
plugins	tests_plugin_lightbox.py	
profiles	tests_plugin_livejournal.py	
spell	tests_plugin_markdown.py	
styles	tests_plugin_organizer.py	
test	tests_plugin_pagetypecolor.py	
__init__.py	tests_plugin_sessions.py	
profile.py	tests_plugin_snippets.py	
runoutwiker.py	tests_plugin_source.py	
setup_tests.py	tests_plugin_spoiler.py	
setup.py	tests_plugin_statistics.py	
tests_actions.py	tests_plugin_style.py	
tests_core.py	tests_plugin_tableofcontents.py	
tests_gui.py	tests_plugin_texequation.py	
tests_plugin_changepageuid.py	tests_plugin_thumbgallery.py	
tests_plugin_core.py	tests_plugin_updatenotifier.py	
tests_plugin_counter.py	tests_plugin_webpage.py	
tests_plugin_datagraph.py	tests_wiki.py	

**help** Папка содержит справку по программе на данный момент на двух языках - на русском и на английском.

**iconset** Папка содержит значки, которые можно установить для страниц заметок в диалоге свойств страницы. Значки могут загружаться не только из этой папки, но и из пользовательских папок со значками. Эта папка содержит вложенные папки, соответствующие группам значков.



**images** Папка содержит все картинки, которые встречаются в программе (иконки программы и картинки на кнопках). Все иконки взяты из свободно распространяемых коллекций, взятых с сайтов famfamfam, yusukekamiyamane и fatcow.

**locale** Содержит локализации программы OutWiker. Сюда не входят локализации плагинов, они расположены в папке каждого плагина. Для локализации используется технология gettext.

**outwiker** Самая главная папка с исходниками. Именно там содержатся основные классы программы.

**plugins** Пустая папка, предназначенная для хранения плагинов, если OutWiker запускается в портативном режиме.

**profiles** Папка содержит вспомогательный скрипт для измерения скорости википарсера. Запуск измерения скорости осуществляется с помощью скрипта profile.py, расположенного в корне папки src.

**spell** Папка содержит встроенные словари для проверки орфографии. К программе прилагаются словари для английского, русского и украинского языков, причем для русского языка есть два словаря: с буквой “Ё” и без нее. Сюда же входит файл html.dic - словарь со списком HTML-тегов.

**styles** Папка содержит стили страниц.

**test** Папка содержит скрипты с unit-тестами.

Кроме того, папка src содержит следующие файлы:

**versions.xml** Файл содержит список изменений в каждой версии OutWiker на русском и английском языке. Этот также файл используется для определения текущей версии программы. Подробнее см. раздел *Хранение номеров версий в OutWiker и плагинах (не написано)*.

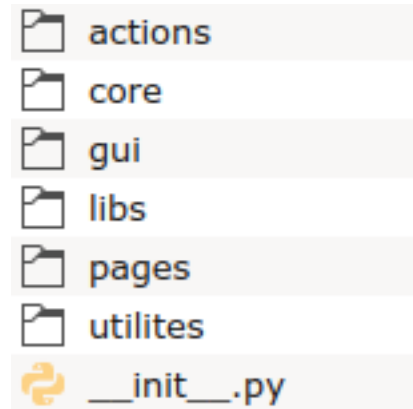
**Файлы tests\_\*.py** Эти файлы предназначены для запуска групп тестов. Подробности см. в *Тестирование OutWiker*.

**runoutwiker.py** Основной файл, запускающий OutWiker.

**setup.py** Скрипт, предназначенный для создания бинарной сборки OutWiker с помощью `sx_freeze`. Подробности см. в *Сборка под Windows* и *Сборка под Linux*.

### Структура папки `src/outwiker`

Папка `src/outwiker` представляет собой Python-пакет (содержит файл `__init__.py`), в который вложены следующие пакеты:



**actions** Эта папка содержит так называемые actions - классы, которые привязываются к элементам меню или к кнопкам на панелях инструментов. Все действия, производимые через интерфейс программы работают через actions, что позволяет настраивать горячие клавиши. Подробности см. в *Actions (не написано)*

**core** Базовые элементы OutWiker. Здесь содержатся классы, по возможности отделенные от интерфейса программы, например, классы, отвечающие за работу с деревом заметок (файл `tree.py`), классы для чтения файлов настроек (`config.py`), классы для работы с плагинами (`pluginsloader.py` и `pluginbase.py`) и другие.

**gui** В этой папке расположены классы, отвечающие за интерфейс программы. Например, класс главного окна содержится в файле `mainwindow.py`.

**libs** Эта папка содержит сторонние библиотеки. В данный момент там содержится единственная библиотека - `pyarsing`, используемая для разбора викинотации.

**pages** Эта папка содержит пакеты, отвечающие за создание страниц различных типов. В ней содержатся пакеты `text` для работы с простыми текстовыми страницами, `html` для работы с HTML-страницами, `wiki` для работы с викистраницами и `search` для работы со страницей глобального поиска.

**utilites** Эта папка содержит скрипты, которые могут быть полезны не только внутри движка OutWiker, но и вне его, например, в системе сборки.

### Инструменты, используемые при разработке

#### Утилиты

**git** Используется для управления версиями исходных кодов.

**fabric** Используется как замена Makefile, для сборки программы под разные операционные системы, запуска тестов (см. *Тестирование OutWiker*), обновления локализаций, сборки архивов плагинов, закачивания новых версий плагинов и программы на сайт. Подробности см. в разделе *Выполнение команд Fabric*.

**Inno Setup** Используется для создания инсталлятора под Windows.

**7zip** Используется для создания архивов в разных операционных системах.

**Sphinx** Используется для создания документации, которую вы сейчас читаете.

## Библиотеки

**Pillow (4.0.0)** Используется для работы с изображениями (создание миниатюр).

**pyenchant (1.6.8)** Используется для проверки орфографии.

**comtypes (1.1.3.post2)** Используется для создания COM-объекта Internet Explorer в операционной системе Microsoft Windows. Под другими операционными системами не требуется.

**cx\_Freeze (4.3.3)** Используется для сборки запускаемого файла под Windows. На момент написания документации вышла cx\_Freeze 5.0.1, но с этой версией возникли проблемы с запуском создаваемого файла outwiker.exe, поэтому на данный момент используется предыдущая версия.

## Выполнение команд Fabric

### Содержание

- *Выполнение команд Fabric*
  - *Основы использования Fabric*
  - *Сборка под Windows*
  - *Сборка под Linux*
  - *Команды, связанные с плагинами*
  - *Команды, помогающие при разработке*
  - *Команды, связанные с локализацией*
  - *Команды, связанные с развертыванием*
  - *Другие команды*

## Основы использования Fabric

Для многих задач, связанных со сборкой, тестированием и выкладыванием новых версий на сайт, используется **Fabric** - удобный инструмент, который позволяет автоматизировать многие задачи как на удаленном сервере, так и на локальном компьютере. На русском языке про **Fabric** можно почитать в статье [Основы использования Fabric](#).

Основная идея использования **Fabric** состоит в том, что все команды (или задачи) описываются в файле `fabfile.py`, расположенном в корневой папке исходных кодов.

Чтобы узнать список имеющихся в данном `fabfile.py` команд, выполните к консоли команду:

```
fab -l
```

Будет выведен следующий список команд:

Available commands:

```

apiversion
apiversions
clear           Remove artifacts after all assemblies
create_tree    Create wiki tree for the tests.
deb            Assemble the deb packages
deb_binary
deb_binary_clear
deb_clear      Remove the deb packages
deb_install    Assemble deb package for current Ubuntu release
deb_single     Assemble the deb package for the current Ubuntu release
deb_sources_included Create files for uploading in PPA(including sources)
deploy_unstable Upload unstable version on the site
doc
linux          Assemble binary builds for Linux
linux_clear    Remove binary builds for Linux
locale         Update the localization file(outwiker.pot)
locale_plugin  Create or update the localization file for pluginname plug-in
outwiker_changelog Generate OutWiker's changelog for the site
plugin_changelog Generate plugin's changelog for the site
plugin_locale  Create or update the localization file for pluginname plug-in
plugins        Create an archive with plugins(7z required)
plugins_clear  Remove an archive with plugins(7z required)
plugins_list
prepare_virtual Prepare virtual machine
run            Run OutWiker from sources
site_versions
sources        Create the sources archives.
sources_clear  Remove the sources archives.
test           Run the unit tests
test_build     Run the build unit tests
upload_plugin  Upload plugin to site
upload_plugins_pack Upload archive with all plugins.
upload_unstable Upload unstable version on the site
win            Build assemblies under Windows
win_clear     Remove assemblies under Windows
    
```

Все эти команды описаны ниже.

Чтобы выполнить команду, в консол нужно написать:

```
fab имя_команды
```

Некоторые команды могут (или должны) принимать некоторые параметры. Параметры передаются после имени команды и символа ":", как показано ниже:

```
fab имя_команды:парам1,парам2
```

Обратите внимание, что после двоеточия и запятой в списке параметров не должно быть пробела. Если параметр должен содержать пробел, то такое значение должно быть заключено в кавычки:

```
fab имя_команды:"параметр с пробелами"
```

**Примечание:** Некоторые команды Fabric принимают булевы параметры. Чтобы в такую задачу передать значение *True*, в качестве параметра в командной строке нужно передать одно из следующих

---

значений: 1 или true (независимо от регистра). Чтобы передать значение False, нужно передать какое-либо другое значение.

---

### Сборка под Windows

**win** Сборка OutWiker под Windows с помощью `cx_Freeze`, а также инсталлятор с помощью `Inno Setup`. Подробнее см. раздел *Сборка под Windows*.

**win\_clear** Удалить все, что создается с помощью команды `win`.

### Сборка под Linux

`deb`

**deb\_binary** Создать deb-пакеты на основе бинарной сборки под Linux для всех поддерживаемых версий Ubuntu.

**deb\_binary\_clear** Удалить все, что создается с помощью команды `deb_binary`

**deb\_sources\_included** Создать deb-пакеты на основе исходных кодов для всех поддерживаемых версий Ubuntu. Используется для загрузки на PPA.

**deb\_install** Создать и установить deb-пакет (на основе исходных кодов) в систему.

**deb\_single** Создать deb-пакет под ту версию Ubuntu, в которой происходит сборка.

**deb\_clear** Удалить все deb-пакеты.

**linux** Создать бинарную сборку под Linux с помощью `cx_Freeze`.

**linux\_clear** Удалить созданную бинарную сборку под Linux.

Подробнее о сборке OutWiker под Linux см. раздел *Сборка под Linux*.

### Команды, связанные с плагинами

**plugins** Создать архивы с плагинами (отдельный архив на каждый плагин и общий архив со всеми плагинами). Эта команда может принимать булево значение. Если оно равно 1, то создаются архивы только для тех плагинов, которые имеют более новые версии по сравнению с теми, что выложены на сайте программы. Общий архив с плагинами создается в любом случае.

**plugins\_clear** Удалить все архивы с плагинами.

### Команды, помогающие при разработке

**run** Запустить OutWiker из исходников.

**apiversion** или **apiversions** Вывести номера версий встроенных пакетов outwiker (см. раздел *Структура папки src/outwiker*).

**test** Запустить юнит-тесты. Подробнее о тестировании см. раздел *Тестирование OutWiker*.

**test\_build** Запустить юнит-тесты, связанные со сборкой. Подробнее о тестировании см. раздел *Тестирование OutWiker*.

### Команды, связанные с локализацией

**locale** Создать файл `src/locale/outwiker.pot`, используемый для создания файлов локализации.

**locale\_plugin** или **plugin\_locale** Создать файл локализации `*.pot` для плагина, указанного в качестве параметра команды.

### Команды, связанные с развертыванием

**deploy\_unstable** Закачать собранную нестабильную версию под Windows, а также собрать deb-пакеты и закачать их на PPA.

**outwiker\_changelog** Вывести список изменений, который нужно будет вставить на сайт. В качестве параметра требуется указать язык: `ru` или `en`.

**plugin\_changelog** Вывести список изменений для плагина. В качестве параметров требуется указать имя плагина и язык: `ru` или `en`.

**site\_versions** Вывести номера версий OutWiker и всех плагинов. Показываются версии, закачанные на сайт и находящиеся в папке с исходниками.

**upload\_plugin** Закачать плагин или плагины на сайт. Для плагинов требуется предварительно создать архивы с плагинами с помощью команды *plugins*.

**upload\_plugins\_pack** Закачать архив со всеми плагинами на сайт. Архив с плагинами требуется предварительно создать с помощью команды *plugins*.

**upload\_unstable** Закачать нестабильную версию OutWiker на сайт.

**plugins\_list** Создать таблицу со списком плагинов для сайта. Требуется указать язык: `ru` или `en`.

### Другие команды

**clear** Удалить все, что создано в папке `build`

**create\_tree** Создать тестовое дерево заметок для тестов.

**doc** Скомпилировать данную документацию.

**prepare\_virtual** Подготовить виртуальную машину с Linux, чтобы в ней можно было бы запустить OutWiker из исходников.

**sources** Создать архив с исходниками. Подробнее см. раздел *Создание архивов с исходниками*.

**sources\_clear** Удалить архив с исходниками.

### Сборка OutWiker

#### Содержание

- *Сборка OutWiker*
  - *Общие моменты, связанные со сборкой*
  - *Сборка под Windows*
    - \* *Использование команды `fab win`*



- \* *Порядок сборки дистрибутивов под Windows*
- \* *Создание бинарной сборки*
- *Сборка под Linux*
  - \* *Порядок сборки deb-пакетов*
- *Создание архивов с исходниками*

## Общие моменты, связанные со сборкой

Все задачи, связанные со сборкой, создают внутри папки `build` папку, имя которой соответствует номеру текущей версии OutWiker в формате `x.x.x.xxx`. Например, `2.0.0.820`. Также внутри папки `build` создается временная папка `tmp`, предназначенная для хранения файлов во время сборки. Данная папка очищается перед каждой сборкой.

Также перед каждой задачей сборки создается папка `tmp/src`, которая содержит копию исходников в соответствии с настройками сборщика: собирается стабильная версия или нестабильная. Если собирается нестабильная версия, то после копирования из исходников удаляется файл `versions_stable.xml`, остается только файл `versions.xml`, описывающий изменения в текущей версии в формате для нестабильной версии.

Если собирается стабильная версия, то после копирования исходников в `tmp/src/` исходный файл `versions.xml` удаляется, а файл `versions_stable.xml` переименовывается в `versions.xml`. Подробнее о формате файла `versions.xml` см. раздел *Хранение номеров версий в OutWiker и плагинах (не написано)*.

Разные классы сборок могут добавлять дополнительные файлы в папку `tmp`.

Описанные возможности реализованы в классе `buildtools.builders.base.BuilderBase`.

Сборка осуществляется с помощью команд Fabric (см. раздел *Выполнение команд Fabric*).

---

**Примечание:** Многие команды сборки принимают булевы параметры. Чтобы в такую задачу передать значение `True`, в качестве параметра в командной строке нужно передать одно из следующих значений: `1` или `true` (независимо от регистра). Чтобы передать значение `False`, нужно передать какое-либо другое значение.

---

## Сборка под Windows

### Использование команды `fab win`

Для того, чтобы собрать все виды дистрибутивов под Windows как нестабильную версию, используется команда

```
fab win
```

Для того, чтобы собрать все виды дистрибутивов под Windows как стабильную версию, используется команда

```
fab win:1
```

Для нестабильной версии будут созданы следующие артефакты в папке `build/{номер версии}/windows`:

- Папка `outwiker_exe`, содержащая все файлы, необходимые для запуска под Windows, включая все плагины в папке `plugins`.
- `outwiker_win_unstable.zip` - zip-архив, содержащий содержимое папки `outwiker_exe`, но без плагинов во вложенной папке `plugins`.
- `outwiker_win_unstable.7z` - 7z-архив, содержащий содержимое папки `outwiker_exe`, но без плагинов во вложенной папке `plugins`.
- `outwiker_win_unstable_all_plugins.zip` - zip-архив, содержащий содержимое папки `outwiker_exe`, включая плагины во вложенной папке `plugins`.
- `outwiker_win_unstable_all_plugins.7z` - 7z-архив, содержащий содержимое папки `outwiker_exe`, включая плагины во вложенной папке `plugins`.
- `outwiker_win_unstable.exe` - инсталлятор, созданный с помощью [Inno Setup](#).
- `versions.xml` - манифест с описанием текущей версии программы OutWiker (см. раздел *Хранение номеров версий в OutWiker и плагинах (не написано)*).

Для стабильной версии будут созданы следующие артефакты в папке `build/{номер версии}/windows`:

- Папка `outwiker_exe`, содержащая все файлы, необходимые для запуска под Windows, включая все плагины в папке `plugins`.
- `outwiker_{x.x.x}_win.zip` - zip-архив, содержащий содержимое папки `outwiker_exe`, но без плагинов во вложенной папке `plugins`.
- `outwiker_{x.x.x}_win.7z` - 7z-архив, содержащий содержимое папки `outwiker_exe`, но без плагинов во вложенной папке `plugins`.
- `outwiker_{x.x.x}_win_all_plugins.zip` - zip-архив, содержащий содержимое папки `outwiker_exe`, включая плагины во вложенной папке `plugins`.
- `outwiker_{x.x.x}_win_all_plugins.7z` - 7z-архив, содержащий содержимое папки `outwiker_exe`, включая плагины во вложенной папке `plugins`.
- `outwiker_{x.x.x}_win.exe` - инсталлятор, созданный с помощью [Inno Setup](#).
- `versions.xml` - манифест с описанием текущей версии программы OutWiker. Этот файл является переименованным файлом `src/versions_stable.xml` (см. раздел *Хранение номеров версий в OutWiker и плагинах (не написано)*).

Команда `fab win` может принимать три булевых параметра.

```
win(is_stable=False, skipinstaller=False, skiparchives=False)
Сборка дистрибутивов под Windows
```

### Параметры

- `is_stable` (*bool*) – Собрать дистрибутивы как стабильную версию (`True`) или как нестабильную (`False`).
- `skipinstaller` (*bool*) – Пропустить шаг создания инсталлятора `outwiker_win_unstable.exe` (если `skipinstaller = True`).
- `skiparchives` (*bool*) – Пропустить шаг создания архивов с собранной версией OutWiker (если `skiparchives = True`).

Чтобы удалить все артефакты, созданные командой `fab win`, предназначена команда:

```
fab win_clear
```

## Порядок сборки дистрибутивов под Windows

Все действия, связанные со сборкой под Windows сосредоточены в классе `buildtools.builders.windows.BuilderWindows`, который является производным от `buildtools.builders.base.BuilderBase`.

В процессе сборки выполняются следующие действия:

Действия, выполняемые классом `buildtools.builders.windows.BuilderBase`.

1. Создание папки `build`.
2. Выполняется команда очистки. Для сборки под Windows это равносильно выполнению команды `fab win_clear`.
3. Удаляется временная папка `build/tmp`, если она существовала.
4. Создается временная папка `build/tmp`.
5. Создается папка `build/{номер версии}/windows`, если она не существовала.
6. Исходники копируются в папку `build/tmp/src`.
7. В папке `build/tmp/src` удаляется файл `versions_stable.xml`, если создается нестабильная версия OutWiker, или `versions_stable.xml` переименовывается в `versions.xml`, если создается стабильная версия OutWiker.

Действия, выполняемые классом `buildtools.builders.windows.BuilderWindows`.

1. В папку `tmp` копируются файлы `copyright.txt` и `LICENSE.txt`.
2. Создается пустая папка для плагинов `tmp/src/plugins`, если она не была создана.
3. Создается бинарная сборка в `tmp/outwiker_exe` (см. раздел *Создание бинарной сборки*).
4. Удаляется и создается заново папка `tmp/outwiker_exe/plugins`.
5. Создаются архивы с бинарной сборкой в формате `zip` и `7z` без плагинов. Созданные архивы помещаются в `build/{номер версии}/windows`.
6. На основе файла шаблона `need_for_build/windows/outwiker_setup.iss.tpl` создается файл `tmp/outwiker_setup.iss`, используемый для создания инсталлятора с помощью `Inno Setup`.
7. Создается инсталлятор в папке `tmp`.
8. Созданный инсталлятор перемещается в папку `build/{номер версии}/windows`.
9. Все плагины копируются в папку `tmp/outwiker_exe/plugins`.
10. Создаются архивы с бинарной сборкой в формате `zip` и `7z` с плагинами. Созданные архивы помещаются в `build/{номер версии}/windows`.
11. Папка `tmp/outwiker_exe` перемещается в `build/{номер версии}/windows`.

## Создание бинарной сборки

Самое важное, что делает команда `fab win` - это создание запускаемого приложения под Windows, чтобы пользователям не требовалось устанавливать интерпретатор Python. Это осуществляется с помощью утилиты `sx_Freeze`. Для создания запускаемых файлов под Windows используется скрипт `src/setup.py` (см. раздел *Классы для создания бинарных сборок с помощью sx\_Freeze*)

В результате выполнения данного скрипта будет создана папка `build/outwiker_exe`, содержащая запускаемый файл `outwiker.exe`, динамически загружаемую библиотеку с интерпретатором Python `python27.dll`, архив `library.zip`, содержащий необходимые Python-библиотеки, а также дополнительные файлы, необходимые для работы с библиотеками и папки, необходимые для работы OutWiker.

..	wxbase30u_vc90.dll	pyexpat.pyd
help	wxmsw30u_adv_vc90.dll	select.pyd
iconset	wxmsw30u_aui_vc90.dll	unicodedata.pyd
images	wxmsw30u_core_vc90.dll	wx._aui.pyd
lib	wxmsw30u_html_vc90.dll	wx._combo.pyd
locale	wxmsw30u_stc_vc90.dll	wx._controls_.pyd
plugins	wxmsw30u_webview_vc90.dll	wx._core_.pyd
spell	<u>outwiker.exe</u>	wx._gdi_.pyd
styles	Microsoft.VC90.CRT.manifest	wx._html_.pyd
libenchant-1.dll	_ctypes.pyd	wx._html2_.pyd
libglib-2.0-0.dll	_elementtree.pyd	wx._misc_.pyd
libgmodule-2.0-0.dll	_hashlib.pyd	wx._stc_.pyd
msvcm90.dll	_socket.pyd	wx._windows_.pyd
msvc90.dll	_ssl.pyd	copyright.txt
MSVCR90.dll	bz2.pyd	LICENSE.txt
<u>python27.dll</u>	PIL._imaging.pyd	</> versions.xml
wxbase30u_net_vc90.dll	PIL._imagingft.pyd	<u>library.zip</u>

Содержимое `library.zip` может выглядеть следующим образом:

..	_MozillaCookieJar.pyc	commands.pyc
_markerlib	_osx_support.pyc	ConfigParser.pyc
comtypes	_strptime.pyc	contextlib.pyc
ctypes	_threading_local.pyc	cookielib.pyc
distutils	argparse.pyc	cx_Freeze__init__.pyc
email	ast.pyc	decimal.pyc
enchant	atexit.pyc	difflib.pyc
importlib	bdb.pyc	dis.pyc
logging	bisect.pyc	doctest.pyc
outwiker	BUILD_CONSTANTS.pyc	dummy_thread.pyc
PIL	calendar.pyc	dummy_threading.pyc
pkg_resources	cgi.pyc	fnmatch.pyc
unittest	cmd.pyc	fractions.pyc
wx	code.pyc	ftplib.pyc
xml	codeop.pyc	functools.pyc
__future__.pyc	collections.pyc	getopt.pyc
_LWPCookieJar.pyc	colorsys.pyc	getpass.pyc

**Предупреждение:** В данный момент для сборки OutWiker под Windows используется `cx_Freeze 4.3.3`. В `cx_Freeze 5.x` возникла проблема с тем, что запускаемое приложение стало гарантированно

виснуть при запуске. Пока проблема не решена, используется предыдущая версия `sx_Freeze`.

**Примечание:** В `sx_Freeze 5.0` изменился способ сохранения необходимых Python-библиотек, и по умолчанию они не архивируются в `library.zip`. С помощью дополнительных параметров можно явно указать, какие библиотеки должны быть включены в `library.zip`. Это нужно будет сделать, если решится проблема с зависимостями, описанная выше. В данный момент эти параметры закомментированы в файле `src/setup.py` (см. раздел *Классы для создания бинарных сборок с помощью `sx_Freeze`*).

Подробное описание работы скрипта `src/setup.py` приводится в разделе *Классы для создания бинарныхборок с помощью `sx_Freeze`*.

## Сборка под Linux

В данный момент для установки под Linux есть возможность создания только deb-пакетов, которые могут быть установлены с помощью команды `sudo dpkg -i {имя пакета}` или закачан на PPA (Personal Packages Archive). В данный момент существуют три PPA-репозитория:

- Для нестабильных версий OutWiker - <https://launchpad.net/~outwiker-team/+archive/ubuntu/unstable>.
- Для стабильных версий OutWiker - <https://launchpad.net/~outwiker-team/+archive/ubuntu/ppa>.
- Для тестирования сборки - <https://launchpad.net/~outwiker-team/+archive/ubuntu/dev>.

Создание deb-пакетов осуществляется через команды Fabric (см. раздел *Выполнение команд Fabric*):

- `fab deb` создает deb-пакеты для всех поддерживаемых версий Ubuntu.
- `fab deb_single` создает единственный deb-пакет под ту версию Ubuntu, в которой запускается данная команда.
- `fab deb_install` создает единственный deb-пакет под ту версию Ubuntu, в которой запускается данная команд, и устанавливает созданный пакет в систему с помощью команды `sudo dpkg -i {имя пакета}`.
- `fab deb_sources_included` создает необходимые файлы для загрузки OutWiker на PPA.

Перечисленные команды могут принимать один булев параметр, который обозначает, что создается сборка в качестве стабильной версии (параметр равен строке, которую можно интерпретировать как True) или нестабильной (параметр равен строке, которую не удастся интерпретировать как True) - см. *примечание*.

Для удаления файлов, созданных с помощью команд `fab deb...`, предназначена команда `fab deb_clear`.

Список поддерживаемых версий Ubuntu содержится в модуле `buildtools.defines` в переменной `UBUNTU_RELEASE_NAMES`.

## Порядок сборки deb-пакетов

Для выполнения задач Fabric `deb`, `deb_single` и `deb_install` предназначен класс `buildtools.builders.Linux.debsource.BuilderDebSource`. Для выполнения задачи `deb_sources_included` предназначен класс `buildtools.builders.Linux.debsource.BuilderDebSourcesIncluded`. Оба этих класса являются производными от класса `buildtools.builders.Linux.debsource.BuilderBaseDebSource`, который в свою очередь, является производным от `buildtools.builders.base.BuilderBase`. Классы

`BuilderDebSource` и `BuilderDebSourcesIncluded` отличаются только параметрами, которые передаются в утилиту сборки deb-пакетов `debuild`.

В классе `BuilderDebSource` используется набор параметров для сборки deb-пакета, предназначенного для непосредственной установки:

```
debuild --source-option=--include-binaries --source-option=--auto-commit
```

В классе `BuilderDebSourcesIncluded` используется набор параметров для создания файлов, предназначенных для закачки пакета на сервер PPA, где будут создаваться необходимые для установки файлы.

```
debuild -S -sa --source-option=--include-binaries --source-option=--auto-commit
```

Основные действия по сборке пакетов под Linux выполняет базовый класс `BuilderBaseDebSource`.

Порядок сборки deb-пакета следующий.

Действия, выполняемые классом `buildtools.builders.windows.BuilderBase` (те же самые действия, что и при сборке под Windows).

1. Создание папки `build`.
2. Выполняется команда очистки. Для сборки под Windows это равносильно выполнению команды `fab win_clear`.
3. Удаляется временная папка `build/tmp`, если она существовала.
4. Создается временная папка `build/tmp`.
5. Создается папка `build/{номер версии}/linux/deb_source`, если она не существовала.
6. Исходники копируются в папку `build/tmp/src`.
7. В папке `build/tmp/src` удаляется файл `versions_stable.xml`, если создается нестабильная версия OutWiker, или `versions_stable.xml` переименовывается в `versions.xml`, если создается стабильная версия OutWiker.

Действия, выполняемые классом `buildtools.builders.linux.debsource.BuilderBaseDebSource` (внутри метода `buildtools.builders.linux.debsource.BuilderBaseDebSource._debuild()`). Действия выполняются для каждой поддерживаемой версии Ubuntu.

1. Создается папка вида `build/{номер версии}/linux/deb_source/outwiker-x.x.x+xxx`, где `x.x.x+xxx` соответствует номеру версии OutWiker.
2. Внутри созданной папки `outwiker-x.x.x+xxx` копируются минимальный набор исходников из `build/tmp/src`. Копирование осуществляется с помощью утилиты `rsync`, которая позволяет задать маски для файлов и папок, которые нужно пропустить при копировании. Также копируются дополнительные файлы, необходимые для сборки. В частности, папка `need_for_build/debian_debsource/{ubuntu_name}/debian`, содержащая инструкции для сборки deb-пакета под конкретную версию Ubuntu. Также копируются другие файлы и папки из `need_for_build/debian_debsource/{ubuntu_name}`. Также копируются файлы `copyright.txt`, `README` и папка `images` из корня исходных кодов.
3. В папке `build/{номер версии}/linux/deb_source` создается архив с “оригинальными” (original) исходниками. Имя архива выглядит следующим образом: `outwiker_2.0.0+817~{ubuntu_name}.orig.tar.gz`, где `{ubuntu_name}` - кодовое имя дистрибутива Ubuntu, для которого создается сборка, число после знака “+” соответствует номеру сборки OutWiker.
4. Создается файл `:file:`build/{номер версии}/linux/deb_source/outwiker-x.x.x+xxx/changelog`, содержащий список изменений для данной версии OutWiker.

5. Выполняется команда `debuild`, соответствующая цели сборки (параметры команды `debuild` показаны *выше*).
6. Удаляется папка `build/{номер версии}/linux/deb_source/outwiker-x.x.x+xxx`.

### Создание архивов с исходниками

Для создания архивов с исходниками предназначена команда `fab sources`, которая может принимать один булев параметр (см. *примечание*), указывающий, будет создаваться архив исходников в виде стабильной или нестабильной версии.

Во время сборки архивов с исходниками создается папка `build/{номер версии}/sources`, в которую будут помещены архивы. Независимо от выбранного режима сборки создается архив `outwiker-src-full-{номер версии}.zip` с полной копией исходников (создается с помощью команды `git archive`).

Если создается архив исходников как нестабильной версии, то создается файл `outwiker-src-min-{номер версии}-unstable.zip`, который содержит минимально необходимый набор файлов, чтобы запустить OutWiker. Если создается архив стабильной версии, то этот файл будет называться `outwiker-src-min-{номер версии}.zip`. Содержимое архивов в двух режимах сборки отличается только текстом файла `versions.xml`.

### Тестирование OutWiker

Для юнит-тестирования в OutWiker используется стандартный пакет `unittest`. Тесты проверяют работу внутренних компонентов OutWiker, а также элементов графического интерфейса.

Все тесты располагаются в папке `src/test`, а запускаемые их скрипты располагаются в папке `src` и имеют имена, соответствующие маске: `tests_*.py` (также см. раздел *Структура папки src/outwiker*). Файлы, соответствующие маске `tests_plugin_*.py` запускают тесты плагинов.

Для запуска тестов предназначена следующая команда `fabfile` (см. раздел *Выполнение команд Fabric*):

```
fab test
```

С помощью дополнительных параметров команды `fab test` можно ограничить наборы тестов, которые должны быть запущены.

`fab test:core` - запуск тестов для ядра движка OutWiker (см. раздел *Пакет outwiker.core (не написано)*).

`fab test:wiki` - запуск тестов парсера викинотации (см. раздел *Парсер викинотации (не написано)*).

`fab test:gui` - запуск тестов графических элементов интерфейса (см. раздел *Пакет outwiker.gui (не написано)*).

`fab test:actions` - запуск тестов actions (см. раздел *Actions (не написано)*).

`fab test:plugins` - запуск тестов для всех плагинов (см. раздел *Создание плагинов для OutWiker (не написано)*).

`fab test:plugins,имя_плагина` - запуск тестов для указанного плагина.

Можно также передавать дополнительные параметры для движка запуска тестов. Например, если требуется прервать тестирование после первого провалившегося теста, то можно выполнить следующую команду:

```
fab test:core,-f
```

В данный момент нельзя использовать дополнительные параметры без ограничения набора выполняемых тестов. Т.е. **следующая команда работать не будет:**

```
fab test:-f
```

Можно запустить тесты выбранного набора тестов (test case) из набора тестов. Например, следующая команда запускает только тесты закладок из набора тестов *core*:

```
fab test:BookmarksTest
```

Можно запустить конкретный тест. Например:

```
fab test:BookmarksTest.testAddToBookmarks
```

---

**Примечание:** Изначально разделение тестов на несколько групп было сделано из-за того, что при выполнении тестов графического интерфейса наблюдается утечка ресурсов GDI (GDI handlers), и после запуска большого количества тестов под Windows количество выделенных GDI-ресурсов превышало 10 000, после чего возникали ошибки. Проблема утечек до сих пор наблюдается, но не влияет на тестирование. Данная утечка скорее всего связана со множественным созданием главного окна OutWiker и не проявляется в работе OutWiker.

---

### Структура папки `src/test`

Все тесты располагаются в папке `src/test`. Имена файлов, содержащие тесты, соответствуют маске `test_*.py`

Тесты для плагинов располагаются в папке `src/test/plugins/`. Для каждого плагина создана отдельная папка внутри `src/test/plugins/`. Например, тесты для плагина `markdown` располагаются в папке `src/test/plugins/markdown`.

## Внутреннее устройство OutWiker

### Формат хранения страниц

### Структура дерева заметок

При создании программы OutWiker был выбран формат хранения заметок, когда каждая страница представлена папкой в файловой системе с некоторыми служебными файлами. Это решение уменьшает вероятность одновременной порчи всех заметок по сравнению со способом хранением всех заметок в одном файле: если какой-нибудь файл, относящийся к заметке попадет на “битый” сектор жесткого диска, то будет испорчена только эта заметка, а не все дерево заметок. Кроме того, поскольку все записи хранятся в текстовом виде, вы можете просматривать их (а при желании и редактировать) даже не имея под рукой OutWiker. Также такой формат хранения позволяет легко организовать “прикрепленные” к заметке файлы, поэтому OutWiker можно использовать и для каталогизации файлов.

Такое хранение заметок подразумевает, что файлы и папки заметок могут быть в любой момент изменены внешними средствами, например, файловыми менеджерами, с помощью них вы можете переносить заметки из одной ветви дерева в другую или даже в другое дерево с заметками, а если вам не хватает возможностей встроенного в OutWiker редактора, то вы можете редактировать заметки с помощью любого текстового редактора, способного открывать и сохранять файлы в кодировке UTF-8, для этого можно воспользоваться плагином [ExternalTools](#).



Однако такой способ хранения, а особенно предположение, что заметки могут изменяться без участия OutWiker, накладывает ряд ограничений. Во-первых, под Windows на данный момент вы ограничены в длине пути до файла, она не может превышать 255 символов (есть пути обхода этого ограничения, возможно, в будущем эта проблема будет преодолена). Во-вторых, поскольку любая заметка может изменяться без участия OutWiker, нет возможности сделать какой-либо кэш с общей информацией обо всех заметках, например, индекс для поиска или информацию о тегах заметок. Поэтому при глобальном поиске приходится каждый раз заново просматривать содержимое каждой заметки, что может занимать достаточно долгое время.

Как уже было сказано, каждая заметка - это просто папка с некоторыми служебными файлами внутри другой папки-заметки, поэтому для создания вложенной заметки достаточно создать папку с именем вложенной заметки, а в ней создать два файла: `__page.opt` и `__page.text` (этот файл не нужен для страниц глобального поиска). Формат хранения каждого типа заметок может отличаться, но файл `__page.opt` должен присутствовать всегда, о его формате будет написано *ниже*.

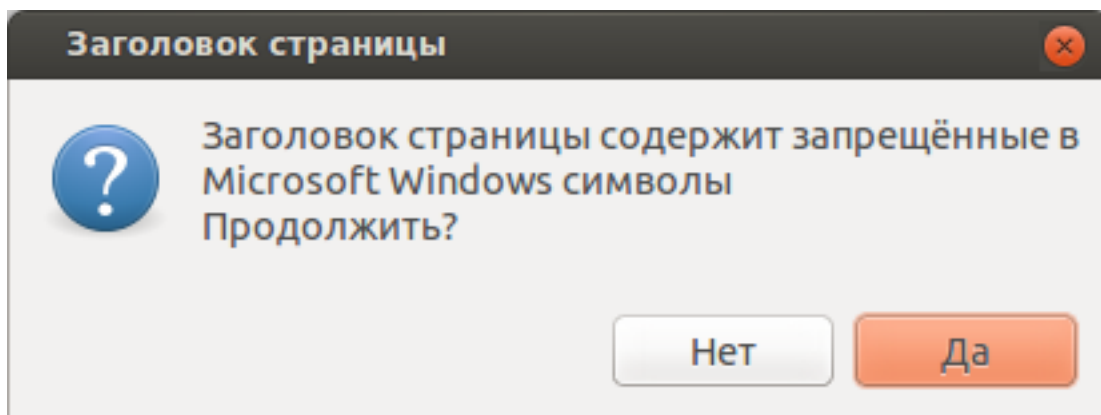
При работе с исходниками OutWiker, например, при создании плагина папки и файлы явно создавать не нужно, для создания заметок и работы с прикрепленными файлами существуют специальные классы внутри API OutWiker (классы, предназначенные для работы со страницами содержатся в файле `src/outwiker/core/tree.py`).

Также в папке с заметкой может храниться значок, установленный для заметки. Файл со значком может записан в различных графических форматах, но рекомендуется использовать формат `png`. В этом случае файл со значком должен иметь имя `__icon.png`. Изображение значка должно иметь размер 16x16 пикселей и может использовать прозрачность. Чтобы внешний вид заметок не зависел от настроек программы OutWiker на различных компьютерах, файл значков копируется в папку с заметкой.

Стиль страниц хранится в файле `__style.html`. Если нужно, в папке `__style` могут находиться дополнительные файлы. Подразумевается, что основные определения стиля хранятся в `__style.html`, а в папке `__style` могут находиться рисунки или скрипты на языке JavaScript, необходимые для отображения стиля. Если для страницы устанавливается стиль по умолчанию, то файл `__style.html` и папка `__style` просто удаляются.

## Ограничения на имена заметок

На имена заметок накладываются некоторые ограничения. Во-первых, ограничения накладывает файловая система. Если под Linux имя директории не может включать в себя только нулевой символ (0), а в остальном вы не ограничены, то под Windows вы не можете создать папку (и, соответственно, заметку) с символами `>`, `<`, `|`, `?`, `*`, `:` и `”`. Если вы работаете под Linux, то при попытке создать заметку, содержащую в имени один из этих символов, вы получите предупреждение, показанное на следующем рисунке, но создать заметку сможете.



Под Windows вам этого сделать не удастся.

Другое ограничение связано со внутренним устройством OutWiker. Все папки внутри папки с заметками, начинающиеся с символов `__` (два символа подчеркивания), программа считает служебными директориями и не отображает в дереве заметок. Например, все прикрепленные файлы содержатся в папке `__attach` внутри каждой заметки (в случае, если прикрепленные файлы есть). Про хранение вложенных файлов будет сказано *ниже*.

Также ограничения выражаются в том, что внутри одной заметки не может быть двух вложенных заметок с одинаковыми именами.

Обойти эти ограничения можно с помощью *псевдонимов*.

### Формат файла `__page.opt`

Файл `__page.opt` является основным файлом в папке заметки, именно он определяет формат страницы, а также различные ее настройки, но этот файл не содержит содержимое страницы, которое видит пользователь (для этого служит файл `__page.text`).

Файл `__page.opt` является текстовым и представляет собой ini-файл. То есть в нем содержатся разделы, обозначенные квадратными скобками, а за названием раздела расположены параметры в формате “имя=значение”. Файл должен быть записан в кодировке UTF-8. Ниже приведен пример файла `__page.opt`, пока не обращайтесь особого внимания на параметры, мы их рассмотрим ниже.

```
[General]
type = wiki
tags = html, софт
order = 0
datetime = 2014-04-12 12:46:55.170000
cursorposition = 121
uid = __a07bd7a7-2be3-41f7-a17d-1ec3997ee988
alias = Отображаемое имя страницы

[Tree]
expand = True

[wiki]
md5_hash = cef7664a01cb932d419a203b2e63b568

[Misc]
pageindex = 0
```

В данном примере содержатся три раздела параметров: *General*, *Tree*, *wiki* и *Misc*. Количество разделов и настроек внутри них зависит от формата страницы, в этом примере приведены параметры для викистраницы.

### Параметры, общие для всех видов страниц

Для начала разберем параметры, общие для всех типов страниц. Начнем с раздела *[General]*.

- *type*. Это единственный обязательный параметр среди всех параметров. Именно он определяет тип страницы. На данный момент существуют 4 типа страниц со следующими типами:
  - *text* - простая текстовая страница.
  - *html* - страница в формате HTML.
  - *wiki* - викистраница.

- *search* - страница глобального поиска.
- *tags*. Этот параметр хранит список меток, которые были применены к странице. Метки перечисляются через запятую (после запятой можно ставить пробел).
- *order*. Задаёт положение заметки в дереве по сравнению с заметками того же уровня. Чем это значение меньше, тем выше располагается заметка в ветви. Если заметки имеют одинаковое значение параметра *order* или оно отсутствует, то такие заметки сортируются по алфавиту.
- *datetime*. Дата и время последнего изменения страницы. Дата и время записывается в формате вида “%Y-%m-%d %H:%M:%S.%f”. За расшифровкой этих символов можете обратиться к документации Python.
- *cursorposition*. Положение курсора, которое было перед закрытием страницы. Это значение используется для того, чтобы при повторном открытии страницы курсор стоял на том же самом месте, как и в тот момент, когда страница закрывалась.
- *uid*. Уникальный идентификатор страницы, который используется для ссылок. В приведенном выше примере идентификатор равен `__a07bd7a7-2be3-41f7-a17d-1ec3997ee988`, это значит, что ссылка на эту страницу будет выглядеть как `page://__a07bd7a7-2be3-41f7-a17d-1ec3997ee988`. При создании заметки странице не присваивается никакой идентификатор, и этот параметр отсутствует, он появляется, когда идентификатор впервые запрашивается, например, при копировании ссылки на страницу в буфер обмена с помощью пункта меню *Инструменты* → *Копировать ссылку на страницу* или одноименного пункта контекстного меню.
- *alias*. Псевдоним страницы или отображаемое имя. Если данный параметр не задан, то в программе OutWiker имя заметки отображается по имени папки, которая содержит данную заметку. Однако, как было написано *выше*, это накладывает некоторые ограничения на имена заметок. С помощью псевдонима можно обойти эти ограничения. Псевдоним - это строка, содержащая любые символы. Если для страницы установлен псевдоним, то псевдоним будет отображаться везде вместо имени заметки. Таким образом, для имени папки сохраняются прежние ограничения, а отображаемое имя заметки может быть произвольным. В основном интерфейсе программы OutWiker нет указаний на возможность установки псевдонимов, чтобы не запутать пользователей. Однако, если вы уверенный пользователь, то можете воспользоваться плагином [HackPage](#), который среди прочего позволяет устанавливать псевдонимы для страниц.
- *expand* из раздела *[Tree]*. Этот параметр имеет булевый тип и может принимать значения *True* или *False*. Этот параметр обозначает, должна ли быть развернута ветвь дерева, начинающаяся с этого узла (заметки). Этот параметр используется для того, чтобы при следующем открытии дерева сохранялась “развернутость” всех его узлов.

### Формат простых текстовых заметок

Формат файла `__page.opt` для простых текстовых заметок не имеет каких-то специфических параметров. При этом параметр *type* в разделе *[General]* должен иметь значение *text*.

Текст заметки хранится в файле `__page.text` в том виде, в котором он отображается в окне редактирования.

Классы, предназначенные для работы с текстовой страницей расположены в папке `src/outwiker/pages/text/`.

### Формат страниц в формате HTML

Страница для заметок в формате HTML должна иметь значение параметра *type* равным *html*.

Кроме того, этот тип страниц в разделе *[General]* имеет дополнительный параметр *linewrap* булевого типа. Если этот параметр имеет значение *True*, то для данной заметки будет включен режим “Автоматический перенос строк”, и для каждого введенного в окне редактора переноса строк будет добавлен тег `<br>`.

Если этот параметр имеет значение *False*, тогда теги `<br>` добавляться не будут, и пользователь сам должен будет писать теги `<br>` или `<p>` для оформления абзацев. В этом случае введенный код HTML будет интерпретироваться точно так, как он введен. Этот режим может быть удобен, например, для подготовки текста для сайта.

Кроме того, у HTML- и *викистраниц* есть дополнительный параметр *pageindex* в разделе *[Misc]*, он определяет, какую вкладку на данной странице в последний раз открывал пользователь. Если пользователь завершил работу со страницей на вкладке “HTML”, это значение будет равно 0, если на вкладке “Просмотр”, то - 1.

При работе с OutWiker пользователь вводит только тело HTML, а именно то, что обычно включается в теги `<body>...</body>`, после чего программа создает полноценный файл `.html`, причем есть возможность посмотреть, что же получается в конечном итоге. Результат полного оформления HTML-страницы OutWiker сохраняет в файл `__content.html` в папке страницы. Этот файл содержит полностью завершённую HTML-страницу со всеми стилями оформления, добавленными переносами строк `<br>` (если включен режим “Автоматический перенос строк”) и т.д.

Например, для HTML-страницы, содержащей фразу “Hello World”, и использующей стиль страницы по умолчанию, файл `__content.html` будет выглядеть следующим образом:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv='X-UA-Compatible' content='IE=edge' />
  <meta http-equiv='content-type' content='text/html; charset=utf-8'/>

  <style type='text/css'>
    body, div, p, table {
      font-size:10pt;
      font-family:Verdana;
    }

    img{border:none}
  </style>
</head>
<body>
Hello world!
</body>
</html>
```

Классы, предназначенные для работы с HTML-страницей расположены в папке `src/outwiker/pages/html/`.

## Формат викистраниц

Формат хранения викистраниц во многом напоминает страницы в формате *HTML*, поскольку в конечном итоге викинотация преобразуется HTML-код, который также сохраняется в файл `__content.html`.

Страница, использующая викинотацию, должна иметь значение параметра *type* равным *wiki*. Все выше сказанное относительно файла `__content.html` относится и к викистраницам с учетом того, что от

викинотации к HTML-странице текст проходит более сложный путь, и перенос строк на викистраницах работает по своим правилам (нет режима “Автоматический перенос строк”).

Кроме того, HTML-код викистраниц проходит дополнительную стадию обработки, чтобы сделать полученный HTML-код более читабельным, расставляя дополнительные переносы строк (не путать с тегом `<br>`) после некоторых тегов (за этот процесс отвечает класс *HtmlImprover* из файла `src/outwiker/core/htmlimprover.py`).

Для примера, если викистраница имеет содержимое:

```
*Hello World!*

||border=1
|| Ячейка таблицы 1 || Ячейка таблицы 2 ||
|| Ячейка таблицы 3 || Ячейка таблицы 4 ||
```

полученный в итоге код HTML будет выглядеть следующим образом:

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv='X-UA-Compatible' content='IE=edge' />
  <meta http-equiv='content-type' content='text/html; charset=utf-8'/>

  <style type='text/css'>
    body, div, p, table {
      font-size:10pt;
      font-family:Verdana;
    }

    img{border:none}

  </style>
</head>

<body>
<b>Hello World!</b><br>
<br>
<table border=1>
<tr>
<td align="center">Ячейка таблицы 1</td>
<td align="center">Ячейка таблицы 2</td>
</tr>
<tr>
<td align="center">Ячейка таблицы 3</td>
<td align="center">Ячейка таблицы 4</td>
</tr>
</table>
</body>
</html>
```

Плагины, такие как, например, [Source](#) или [TeXEquation](#), могут добавлять свои теги между тегами `<head> ... </head>`, а также перед закрывающимся тегом `</body>`.

Поскольку для очень длинных текстов с викинотацией преобразование в формат HTML может быть сравнительно тяжелой операцией, занимающие единицы секунд, викистраница старается кэшировать полученный результат и не создавать код HTML заново при каждом переключении из вкладки *Вики* на вкладку *Просмотр* или “HTML”, как это происходит со страницами в формате *HTML*.

Для того, чтобы определить, изменилась ли страница, OutWiker рассчитывает контрольную сумму по алгоритму MD5 от строки, полученной склеиванием следующих элементов:

- Заголовка страницы.
- Текста страницы.
- Текущего списка установленных плагинов.
- Стиля страницы.
- Списка прикрепленных файлов.
- Списка дочерних страниц.
- Некоторых настроек программы, влияющих на сгенерированный код HTML.

Полученный MD5-хэш записывается в параметр `md5_hash` в разделе `[wiki]`. Он может выглядеть, например, следующим образом:

```
[wiki]
md5_hash = dd03106a9e3705caeba5ac75a83ce518
```

Этот параметр не является обязательным, в случае его отсутствия или, если в нем записана неверная строка, ничего страшного не произойдет, просто данная страница при следующем открытии вкладки “Просмотр” или “HTML” заново создаст файл `__content.html`, после чего в данный параметр будет записана новая контрольная сумма MD5.

Как и у *HTML-страниц*, у викистраниц существует параметр `pageindex` в разделе `[Misc]`, он также определяет, какую вкладку на данной странице в последний раз открывал пользователь. Если пользователь завершил работу со страницей на вкладке “Вики”, это значение будет равно 0, если на вкладке “Просмотр”, то - 1, если на вкладке “HTML”, то - 2.

Классы, предназначенные для работы с викистраницей расположены в папке `src/outwiker/pages/wiki/`.

### Формат страниц глобального поиска

Формат страниц глобального поиска несколько отличается от форматов хранения записей другого типа, поскольку этому типу страниц не нужно хранить большой объем текста. Поэтому в папке страницы поиска содержится обязательный файл `__page.opt`, но нет файла `__page.text`.

Файл `__page.opt` может иметь следующий вид:

```
[General]
type = search
tags = проверка, test
order = 2
datetime = 2014-04-28 20:56:28.363891

[Search]
phrase = искомая фраза
tags = test, html
strategy = 0
sort = 0

[SearchResults]
page_0 = Пример страницы
page_1 = Тесты
page_2 = Тесты/Проверка
page_3 = Тесты/Проверка/Плагины/source
```

```
page_4 = Тесты/Проверка/Абырвалг
page_5 = Тесты/Проверка/Тест
page_6 = HTML5
```

Для данного типа страниц параметр *type* из раздела *[General]* должен быть установлен в значение *search*. Все остальные параметры из этого раздела аналогичны одноименным параметрам страниц, описанных выше.

Рассмотрим теперь параметры, специфические для страницы глобального поиска. Начнем с раздела *[Search]*, где хранятся параметры, описывающие последний поиск:

- *phrase*. Фраза, которая будет искаться по всем страницам дерева.
- *tags*. Метки, которые должны быть проставлены на страницах, чтобы они попали в результат поиска. Теги должны быть разделены запятыми. Если теги не установлены, значит поиск осуществляется по всем страницам дерева. Не путайте этот параметр с одноименным параметром из раздела *[General]*, страница поиска тоже может быть отмечена метками.
- *strategy* определяет поведение поиска в случае выбора двух и более меток. Если параметр *strategy* равен 0 или в нем записано ошибочное значение, значит для того, чтобы страница попала в результаты поиска, необходимо, чтобы в ней была проставлена *хотя бы одна* метка из выбранных (хранимых в параметре *tags*). Если параметр *strategy* равен 1, значит для того, чтобы страница попала в результаты поиска, необходимо, чтобы в ней были проставлены *все* выбранные на странице поиска метки.
- *sort* определяет способ сортировки найденных результатов. Если параметр *sort* равен 0 или имеет ошибочное значение, сортировка найденных страниц будет осуществляться по алфавиту по заголовку страницы (поведение по умолчанию). Если параметр *sort* равен 1, сортировка найденных страниц будет осуществляться по заголовку страницы в обратном порядке. Если параметр *sort* равен 2, сортировка найденных страниц будет осуществляться по дате последнего изменения страницы, причем страницы с наиболее свежими правками будут располагаться вверху. Если параметр *sort* равен 3, сортировка найденных страниц будет осуществляться по дате последнего изменения страницы, причем страницы с наиболее свежими правками будут располагаться внизу.

В разделе *[SearchResults]* сохраняются пути до найденных страниц. Формат хранения результатов поиска видно на приведенном выше примере. Результат сохраняется на тот случай, можно через какое-то время можно было бы вернуться к старому результату поиска без его перезапуска, поскольку эта процедура может быть довольно длительной.

Классы, предназначенные для работы со страницей поиска расположены в папке `src/outwiker/pages/search/`.

## Хранение вложенных файлов

Как уже было сказано выше, прикрепленные к заметке файлы хранятся в папке `__attach`, поэтому для добавления новых файлов вы можете не только использовать интерфейс программы, но и копировать файлы в эту папку, используя любой файловый менеджер. Если вы добавляете файл внешними средствами в выбранную в данный момент заметку, не забудьте нажать кнопку “Обновить” в окне “Прикрепленные файлы”, чтобы новые файлы появились в окне программы.

Когда новая заметка только была создана, она не имеет папки `__attach`, она будет создана только при первой необходимости - когда вы прикрепите какой-нибудь файл, нажмете кнопку “Открыть папку с прикрепленными файлами” в окне “Прикрепленные файлы” или выберете пункт меню `:menuselection:Инструменты -> Копировать путь до прикрепленных файлов`”.

Внутри папки `__attach` могут находиться и вложенные папки, они даже будут показаны в окне “Прикрепленные файлы” но эта особенность программы OutWiker не особо афишируется, поскольку ин-

терфейс для работы с прикрепленными папками пока не очень развит (надо добавить возможность заходить внутрь прикрепленных папок и некоторые другие функции). Но здесь есть одно ограничение - прикрепленные папки не могут начинаться с символов `__` (два подчеркивания), поскольку такие папки считаются служебными и не показываются в списке прикрепленных файлов.

Одно из применений таких служебных папок описывается в следующем разделе.

### Хранение миниатюр

На *викистраницах* есть возможность создавать уменьшенные копии изображений (миниатюры) с помощью команды `%thumb% ... %%`. При использовании этой команды могут быть заданы размеры создаваемых миниатюр. Поскольку миниатюры иногда могут оказаться полезными сами по себе, то они отображаются не просто с помощью тега `<img>` со ссылкой на оригинальный файл и с атрибутами `width` и `height`, задающими размер, а миниатюра создается в виде отдельного файла.

Поскольку в большинстве случаев пользователю незачем видеть созданные миниатюры в списке прикрепленных файлов, а хранить эти файлы где-то надо, для этого в папке `__attach` создается вложенная папка `__thumb`. Так как ее имя начинается символов `__`, она не отображается в списке прикрепленных файлов.

Сами миниатюры хранятся в этой папке и имена их файлов однозначно идентифицируют файл картинки, из которого они были созданы, а также размер миниатюры.

Например, если к викистранице прикреплен файл `image.png`, а на самой странице содержится текст:

```
%thumb width=200%Attach:image.png%%  
%thumb width=400%Attach:image.png%%  
%thumb height=200%Attach:image.png%%  
%thumb maxsize=200%Attach:image.png%%
```

то в папке `attach/__thumb/` будут созданы следующие файлы:

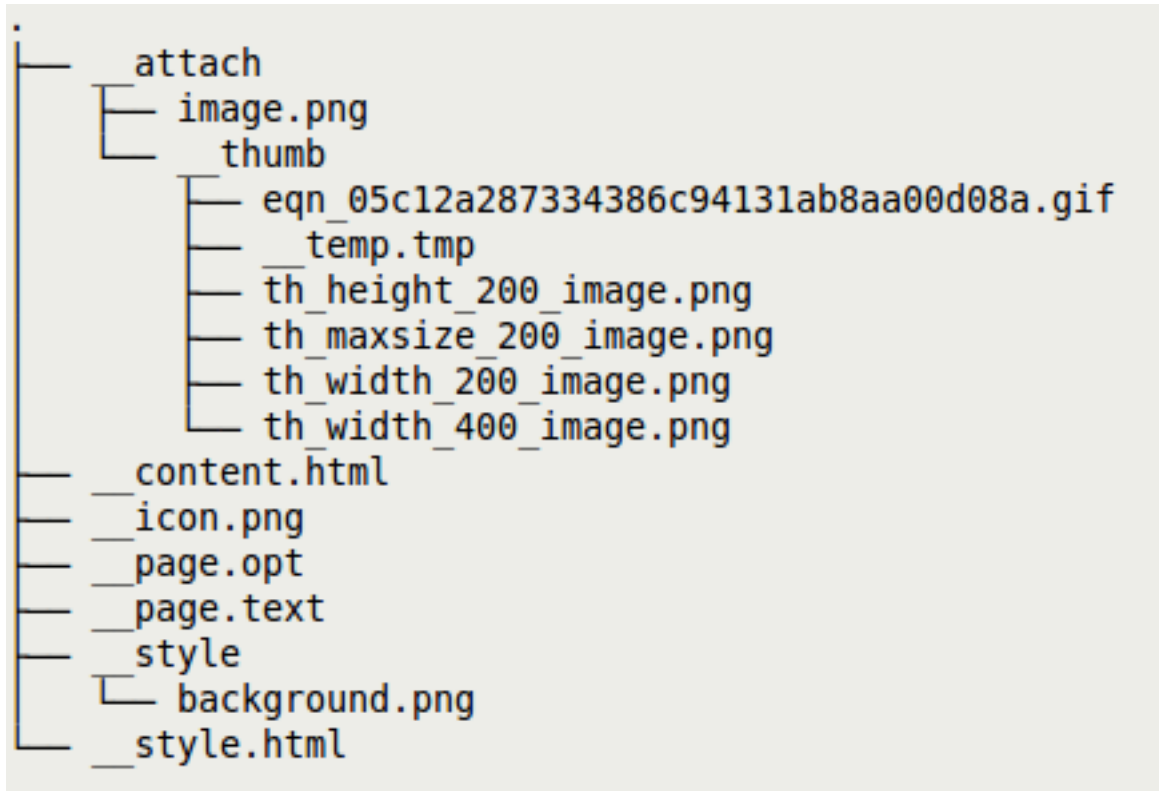
- `th_width_200_image.png`
- `th_width_400_image.png`
- `th_height_200_image.png`
- `th_maxsize_200_image.png`

Таким же образом создаются миниатюры для каждого размера, и в то же время если в тексте страницы встретится еще одно упоминание миниатюры с тем же размером, то оно не будет создано повторно. К тому же имена миниатюр остаются читабельными, можно легко найти нужную миниатюру и скопировать ее с помощью файлового менеджера в другую папку, если этот файл миниатюр понадобится для других целей.

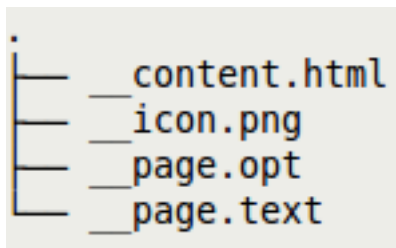
### Заключение

Подводя итоги всего вышесказанного, содержимое папки заметки может выглядеть следующим образом:





Для страницы со стилем по умолчанию и без прикрепленных файлов папка с заметкой будет выглядеть вот так:



## Классы для создания бинарных сборок с помощью `sx_Freeze`

Файл `src/setup.py` содержит классы для создания бинарных сборок под Windows и Linux с помощью `sx_Freeze`.

Базовый класс `BaseBuilder` выполняет общие действия, которые необходимо выполнить с помощью `sx_Freeze` независимо от целевой операционной системы. Производные от `BaseBuilder` классы уточняют параметры, специфические для Windows (класс `WindowsBuilder`) и Linux (класс `LinuxBuilder`).

```
class BaseBuilder
```

```
    build()
```

Единственный публичный метод класса `BaseBuilder` - это метод `build`, который собирает информацию о параметрах сборки и передает их в функцию `setup` из модуля `sx_Freeze`.

```
    _getExecutable()
```

Класс `BaseBuilder` является абстрактным. Данный метод является единственным методом,

который необходимо определить в производных классах. Метод должен возвращать экземпляр класса `cx_Freeze_.Executable` с информацией о создаваемом запускаемом файле.

### `_getExtraIncludeFiles()`

Класс `BaseBuilder` знает, какие файлы требуются для запуска OutWiker после сборки независимо от операционной системы. С помощью данного метода можно указать, какие дополнительные файлы, специфические для данной операционной системы, потребуются после сборки для запуска OutWiker. Метод должен возвращать список кортежей из двух элементов. Первый элемент кортежа - полный путь до файла, который необходимо включить в сборку. Второй элемент кортежа - имя папки, в которую будет скопирован данный файл после сборки. Все файлы, специфичные для операционных систем находятся в папке `need_for_build` в корне папки с исходными кодами. Например, файлы, необходимые для сборки под Windows, находятся в папке `need_for_build/windows`, а под Linux - `need_for_build/linux`.

### `_getPathIncludes()`

С помощью данного метода можно указать `cx_Freeze`, где искать дополнительные бинарные файлы, связанные с библиотеками Python. Данный метод должен возвращать список таких мест. Метод используется только для сборки под Linux, он возвращает список `[u'/usr/lib']`.

### `_getExtraBuildExeOptions()`

С помощью данного метода можно указать дополнительные параметры для сборки. Метод должен возвращать словарь, в котором ключ является именем параметра, а значение - значением данного параметра. Данные параметры добавляются в словарь параметров (параметр `options` функции `cx_Freeze.setup()`), который передается в качестве значения по ключу `build_exe` (см. документацию `cx_Freeze`). В данный момент данный метод используется в классе `WindowsBuilder`, чтобы добавить параметр `include_msucr`, обозначающий, что под Windows необходимо включить в сборку динамически загружаемые библиотеки из Microsoft Visual C++ Runtime Library.

## Как создать свой стиль для страниц заметок (не написано)

---

### План

Написать раздел про создание новых стилей для страниц.

---

## Создание плагинов для OutWiker (не написано)

### Хранение номеров версий в OutWiker и плагинах (не написано)

---

### План

Описать формат хранения изменений в версиях.

---

## Пакет `outwiker.core` (не написано)

### Actions (не написано)

## Пакет `outwiker.gui` (не написано)

Парсер викинотации (не написано)

Локализация OutWiker и плагинов (не написано)

---

## План

Написать раздел про локализации.

---

## История версий

Текущая разрабатываемая версия

### 2.0.0.816 RC (25.04.2017)

- Удалена итальянская локализация.

### 2.0.0.814 beta (03.04.2017)

- Добавлен параметр “subpath” в настройках выбора заголовка главного окна программы.
- Некоторые улучшения в интерфейсе.
- Исправления ошибок.

### 2.0.0.812 beta (18.03.2017)

- Улучшение интерфейса для установки горячих клавиш.
- Обновление справки.
- Добавлена поддержка Ubuntu 17.04 Zesty Zapus.

### 2.0.0.810 beta (24.02.2017)

- Длинные панели инструментов разделены на несколько.
- Исправления ошибок.

### 2.0.0.808 beta (26.01.2017)

- Добавлена возможность создавать псевдонимы для страниц.
- Переделан интерфейс для установки горячих клавиш.
- Добавлена настройка для изменения цвета фона поля редактора.
- Добавлена настройка для изменения цвета фона выделенного текста.

### 2.0.0.806 beta (14.11.2016)

- Добавлены пункты меню и горячие клавиши для перемещения по тексту.
- Добавлены пункты меню и горячие клавиши для копирования / вырезания текущей строки в буфер обмена.
- Исправление ошибок.

### 2.0.0.804 dev (20.10.2016)

- Исправление в сборке deb-пакета в Ubuntu 16.10.

### 2.0.0.802 dev

- Внутренние изменения.

### 2.0.0.800 dev (20.08.2016)

- Добавлена поддержка шведского языка.
- Добавлен параметр командной строки “-page, -p” для выбора страницы при запуске.
- Добавлен параметр командной строки “-normal” для отключения свертывания окна при запуске.
- Добавлен параметр командной строки “-debug” для запуска в режиме отладки.
- Исправление ошибок.

### 2.0.0.798 dev (27.07.2016)

- Восстановлена работоспособность иконки в трее под Linux.
- В редактор добавлена возможность для слияния строк с помощью горячей клавиши или меню.
- В редактор добавлена возможность удаления слов слева / справа с помощью горячей клавиши или меню.
- В редактор добавлена возможность для уменьшения вложенности элементов списка с помощью горячей клавиши или меню.
- Изменен стиль вкладок.

### 2.0.0.796 dev (02.07.2016)

- В редактор добавлена возможность дублировать текущую строку с помощью горячей клавиши или меню.
- В редактор добавлена возможность удалять текущую строку с помощью горячей клавиши или меню.
- В редактор добавлена возможность перемещения строк вверх / вниз с помощью горячих клавиш или меню.
- Улучшена вставка заголовков на викистраницы.

- Рефакторинг системы сборки.
- Исправление ошибок.

### 2.0.0.794 (30.05.2016)

- Исправление ошибок, возникших после перехода на wxPython 3.0.

### 2.0.0.792 (10.05.2016)

- Переход на библиотеку wxPython 3.0.

## Предыдущие версии

### 1.9.0.790 (09.04.2016)

- Добавлена проверка орфографии.
- Иконки для заметок разделены на группы.
- Теперь пользователи могут дополнять стандартный набор иконок своими файлами.
- Возможность выбора цветов для облака тегов.
- Возможность выбора действий при клике левой и средней кнопки мыши по тегам.
- Возможность выбора действия при двойном клике по прикрепленному файлу.
- Добавлены новые викикоманды для создания страниц (:table:), (:row:), (:cell:), (:hcell:)
- Теперь поиск действий для горячих клавиш в настройках осуществляется не только по названию действия, но и по его описанию.
- Добавлены новые иконки.
- Добавлены новые стили страниц.
- Теперь в диалоге вставки ссылок можно выбирать один из прикрепленных файлов, чтобы сделать ссылку на него.
- Дополнена справка.

### 1.8.1.752 (20.10.2014)

- Исправление ошибок

### 1.8.0.750 (11.10.2014)

- Добавлена возможность настройки горячих клавиш.
- Добавлена возможность ссылаться на страницы, используя их уникальные идентификаторы.
- Добавлена возможность перехода по ссылкам на якоря, расположенные на других страницах (page://bla-bla-bla/#anchor).

- Добавлена возможность использовать относительные пути в ссылках на страницы ( ../ ../Страница 1/Страница 2)
- Добавлена возможность установки стиля страницы разом для всех страниц ветки дерева (или всего дерева).
- Добавлена возможность поиска и замены по странице.
- Добавлены кнопки Вперед / Назад для возврата к предыдущим открытым страницам.
- Добавлена возможность настройки цветов редактора.
- Добавлена настройка для изменения поведения клавиш Home / End в редакторе (перескакивать в начало / конец строки или абзаца).
- Добавлен новый тег викисинтаксиса для цитирования: [`>...<`].
- Добавлена кнопка и пункт меню для вставки текущей даты и времени.
- Добавлены команды (`:crdate:`) и (`:eddate:`) для вставки даты создания и редактирования страницы соответственно.
- Добавлены диалоги для вставки команд (`:attachlist:`), (`:childlist:`) и (`:include:`).
- В команде (`:childlist:`) добавлена возможность сортировки дочерних страниц по дате создания и дате последнего изменения.
- На странице глобального поиска добавлена возможность сортировки найденных страниц по дате создания.
- Добавлен новый параметр командной строки “-r” или “-readonly”, позволяющий открывать дерево заметок в режиме “Только для чтения”.
- Добавлены всплывающие подсказки над иконками в окне свойств страниц.
- Добавлены новые стили оформления страниц.
- Добавлена кнопка и пункт меню для открытия папки с прикрепленными файлами в системном файловом менеджере.
- Добавлено сохранение последнего используемого стиля страницы.
- Добавлено сохранение положения курсора на каждой странице перед закрытием.
- Добавлена итальянская локализация.
- Теперь в панели с прикрепленными файлами показываются иконки файлов.
- Изменен алгоритм расстановки переносов строк.
- Открытие заметок и поиск страниц осуществляется в отдельном потоке.
- Теперь для каждой страницы сохраняется дата создания.
- Теперь все HTML-теги, создаваемые википарсером, будут строчными буквами.
- Теперь справка открывается в новом окне.
- Новые иконки.
- Настройки перенесены из `~/outwiker` в более идеологически верные места (зависит от операционной системы).
- Ускорение работы.
- Различные исправления и улучшения.
- Исправление ошибок.

### 1.7.0.684 (01.12.2012)

- Добавлена возможность открывать заметки в отдельных вкладках.
- Панель инструментов разделена на несколько перемещаемых панелей.
- Теперь сохраняется и показывается дата последней правки на странице.
- Значительно ускорены разбор и раскраска викисинтаксиса (по тестам ускорение от 2 до 10 раз).
- Добавлен интерфейс для вставки некоторых элементов викисинтаксиса.
- Добавлена возможность с помощью викисинтаксиса делать более крупный и мелкий шрифт.
- Добавлена возможность сортировки результатов поиска по алфавиту или по дате.
- Добавлена возможность принудительной пересборки HTML-кода по викинотации.
- Исправления в шаблонах страниц.
- В дистрибутив под Linux добавлена векторная иконка и иконки больших размеров.
- Изменения во внешнем виде страницы поиска.
- Создан инсталлятор под Windows.
- Исправлена ошибка, из-за которой не создавались превьюшки на викистраницах, если расширения файлов с картинками было записано заглавными буквами.
- Исправлена ошибка из-за которой не сохранялись размеры внутренних окон (дерево заметок, прикрепленные файлы и т.п.)
- Исправления других ошибок.
- Множество небольших исправлений и улучшений в интерфейсе

### 1.6.0.650 (19.05.2012)

- Добавлена возможность смены стиля страниц.
- Добавлена раскраска викисинтаксиса в редакторе.
- Викикоманды для оформления ссылок на пользователей и сообщества livejournal.com вынесены из программы в отдельный плагин.
- Теперь для выравнивания текста на викистраницах кроме `%center%` и `%right%` можно использовать `%left%` и `%justify%`.
- Добавлена украинская локализация.
- Переделан диалог свойств страницы.
- Множество небольших исправлений.

### 1.5.0.627 (23.02.2012)

- Добавлено облако меток в главное окно.
- Теперь метки можно добавлять, кликая по ним мышкой, а не вводить имена меток полностью.
- Метки для выбранной записи теперь выделяются в общем облаке меток.
- Добавлена возможность группового добавления/удаления/переименования меток.

- Добавлена настройка, позволяющая сворачивать окно программы при нажатии на кнопку закрытия.
- Двойной щелчок на заголовок страницы открывает диалог для редактирования свойств страницы.

### 1.4.0.601 (07.12.2011)

- Добавлена поддержка плагинов.
- Добавлено автосохранение заметок по таймеру (интервал сохранения задается в настройках).
- Добавлена проверка изменения текста заметок сторонними программами.
- Для HTML-страниц добавлена кнопка включения/выключения автоматического переноса строк.
- Версия под Linux теперь устанавливается в /usr, а не в /opt. Теперь из консоли программу можно вызывать с помощью команды outwiker.
- Улучшена обработка викисинтаксиса, касающегося создания списков.
- Теперь поиск по страницам ищет искомую фразу в именах прикрепленных файлов и в названиях тегов.
- Исправления ошибок и небольшие улучшения интерфейса.
- Программа теперь тестируется только под Python 2.7.

### 1.3.2.496 (13.09.2011)

- Новый интерфейс выбора иконок.
- Добавил новые иконки из наборов famfamfam.com и yusukekamiyamane.com.
- Новая иконка программы (спасибо Алексею Набродову).
- В пределах одной страницы теперь работают якоря.
- Новый элемент викинотации для добавления якорей.
- Значительно уменьшено моргание и обновление интерфейса при перемещении по дереву.
- Справка на английском языке (спасибо Илье Чешкову за исправления моего корявого перевода с помощью Google Translate :) ).
- Добавлены горячие клавиши для тега <PRE> на HTML-странице и [ @...@ ] на викистранице.
- Добавлены горячая клавиша и кнопка для [ [ < < ] ] на викистранице.
- Добавлены кнопки для тегов <code> и <blockquote> на HTML-странице.
- Добавлена кнопка “Обновить” для панели с прикрепленными файлами.
- При нажатии на иконку в трее, свернутое окно разворачивается, а развернутое теперь сворачивается.
- Исправлено: под Windows не работали ссылки на страницы, имеющие в своем пути символ “#”.
- Исправлена ошибка, возникавшая при испорченном файле \_\_page.opt в корне вики.
- Другие исправления ошибок в программе и переводе.
- Рефакторинг кода.



**1.3.1.393 (01.07.2011)**

- Под Windows исправлена ошибка, из-за которой не работали ссылки на странице поиска.
- Под Linux исправлена ошибка, из-за которой не открывались страницы, если в пути до них был символ '#’.

**1.3.0.384 (26.06.2011)**

- Теперь в качестве HTML-рендера под Windows используется движок Internet Explorer, под Linux - WebKit.
- Добавлена возможность печати заметок и их исходного текста.
- Добавлен новый вики-тег {-...-} для зачеркнутого текста.
- Добавлена кнопка для зачеркнутого текста на HTML-странице.
- Возможность установки размера шрифта при просмотре заметок через окно настроек.
- Возможность установки дополнительных стилей CSS для заметок.
- Изменение горячей клавиши для переключением между кодом и просмотром на F4.
- Добавлены пункты меню для вики-команд (: ... :).
- Страницы, открытые в режиме “Только для чтения”, в дереве выделяются курсивом.
- Под Windows программа теперь должна работать без установки MS Visual C++ 2008 Redistributable Package.
- При запуске программы под Windows с помощью exe-шника ошибки будут выводиться в файл outwiker.log, который будет создан в той же папке, где расположен файл настроек outwiker.ini.
- Исправлены ошибки, возникавшие при попытке открыть вики, если для какой-то из заметок нет прав на запись.
- Исправлена ошибка, из-за которой под Windows не работал рендеринг формул, если программа была установлена в директорию, содержащей пробелы.

**1.2.0.322 (24.04.2011)**

- Для вики-страниц добавлена возможность ввода формул в нотации TeX (для рендеринга используется mimeTex).
- Ссылка на картинку в виде [[Attach:xxx.png]] или [[http://.../xxx.png]] в вики-страницах вставляет только ссылку на картинку, а не саму картинку.
- Добавлена вики-команда (:childlist:) для вставки списка дочерних заметок.
- Добавлена вики-команда (:attachlist:) для вставки списка прикрепленных файлов
- Добавлена вики-команда (:include:) для вставки содержимого прикрепленных файлов.
- Добавлены вики-команды для вставки ссылок на пользователей Livejournal.com (команда (:ljuser:)) и на ЖЖ-сообщества (команда (:ljcomm:)).
- В меню добавлена функция для преобразования специальных символов HTML (<, > и т.п.) в их HTML-представление.
- Внутри оператора [@ ... @] при разборе вики-нотации символы “<” заменяются на “&lt;”, а “>” - на “&gt;”.

- Добавлена возможность задания шаблона для пустой вики-страницы (см. настройки).
- Из вики-нотации удалена команда `\`, так как ее полностью заменяет команда `[[<<]]` (перевод строки).
- В дереве заметок теперь показывается корень вики.
- Значительно ускорена сортировка страниц по алфавиту.
- Улучшено распознавание ссылок в тексте вики-страниц.
- Улучшено кеширование разбора (парсинга) вики-страниц.
- В заголовках (выделенные нотацией `!!`, `!!!` и т.п.) теперь можно использовать и некоторые другие элементы вики-нотации (выделение полужирным, курсив и т.д, а также формулы и вики-команды (`.....`)).
- Исправлены ошибки, связанные с удалением и перемещением заметок, которые нельзя удалить/переместить из-за блокировки их директорий.
- Исправлены ошибка, возникавшая при попытке сохранить страницу, директорию которой удалили вручную во время работы программы.
- Исправлено падение программы под Linux при переименовании страницы, содержащей картинки, загружаемые из интернета.ч
- Другие небольшие исправления и рефакторинг кода (особенно вики-парсера).
- Подробности в блоге

#### 1.1.0.218 (23.02.2011)

- Переделан интерфейс главного окна. Теперь можно перемещать и закрывать панели с деревом заметок и вложенными файлами.
- Добавлена возможность изменения порядка следования заметок в дереве (горячие клавиши `Ctrl+Shift+Up` / `Ctrl+Shift+Down`).
- Добавлена возможность принудительной сортировки записей по алфавиту.
- Добавлен полноэкранный режим.
- Автосохранение заметок при переключении на другое приложение.
- Добавлена возможность перетаскивать файлы из окна вложений в другие программы.
- Добавлена настройка: всегда показывать значок в трее.
- Добавлена настройка размера табуляции в редакторе.
- Теперь при создании страницы по умолчанию выбирается тип страницы, созданной в прошлый раз.
- Теперь при создании страницы директория `__attach` для вложенных файлов не создается. Она создается при первой необходимости.
- Уменьшено моргание главного окна при различных действиях.
- Исправления ошибок.
- Мелкие исправления в интерфейсе.
- Подробности в блоге

### 1.0.0.108 release (25.12.2010)

- Добавлено окно настроек.
- Добавлена поддержка многоязычности (теперь в программе есть русский и английский интерфейс).
- Добавлена возможность сворачивания в трей (в том числе и при запуске).
- Добавлена возможность автоматического открытия последней вики при запуске программы.
- Добавлена возможность отключения вопроса перед выходом из программы.
- Добавлена возможность установки шрифта для редактора.
- Добавлена возможность включения нумерации строк в окне редактора.
- Добавлена настройка размера по умолчанию для превьюшек картинок на вики-страницах.
- Добавлена вкладка для просмотра полученного по вики-нотации кода HTML.
- Пустые страницы по умолчанию открываются на вкладке для редактирования.
- Если страница пустая, то на вкладке просмотра выводится список прикрепленных файлов (эту особенность можно отключить)
- Добавлена настройка заголовка главного окна. По умолчанию туда выводится имя открытой вики и текущей страницы.
- Новое окно “О программе”, добавил туда вкладку “Donate” (удивительно, но эту “фичу” просили сразу несколько человек :)).
- Исправлена ошибка, связанная с хранением настроек в профиле с русскими буквами.
- Улучшено распознавание интернет-адресов в вики-нотации.
- Двойной клик по заметке в дереве открывает диалог со свойствами заметки.
- Вики-файлы, открытые в режиме “только для чтения” теперь не попадают в список последних открытых файлов.
- Различные мелкие исправления и улучшения в интерфейсе.
- Подробности в блоге

### 1.0 beta 3 (20.10.2010)

- Добавлен режим открытия вики “только для чтения”. Теперь справка открывается именно в этом режиме.
- Добавлена возможность хранить настройки в папке профиля, а не в папке с программой (непортбельный режим).
- Устранено моргание окна при переключении страниц.
- Исходники перенесены с github на launchpad.net (<https://launchpad.net/outwiker>). Теперь вместо git используется Bazaar.
- Исправление нескольких ошибок в вики-парсере.
- Подробности в блоге

### 1.0 beta 2 (22.08.2010)

- Добавлена возможность перемещения страниц по дереву.
- Добавлена возможность переименования страниц без вызова диалога свойств страницы.
- Добавлено контекстное меню для дерева заметок.
- Относительные пути для ссылок на страницы теперь регистронезависимы.
- Изменение горячей клавиши для переключения между кодом и просмотром (было F2, стало F5).
- Если прикрепляется файл с именем, которое уже существует среди прикрепленных файлов, то показывается диалог с вопросом “Что делать?”
- Папка `_thumb` с превьюшками картинок теперь не показывается в списке прикрепленных файлов.
- Теперь дерево заметок сохраняет свое состояние (раскрыт узел или закрыт).
- Добавлена статусная панель, на которой показывается текст ссылки при наведении на нее курсора.
- В программе теперь используются абсолютные пути до папок, поэтому при запуске программы из исходников рабочей папкой не обязательно должна быть папка с исходниками.
- Исправления ошибок и глюков.
- Подробности в блоге

### 1.0 beta 1 (18.07.2010)

- Новый тип страниц - страница в нотации wiki.
- Добавлена справка по программе.
- Добавлен пункт меню для копирования ссылки на страницу в буфер обмена.
- Добавлена возможность открывать вики, передавая путь в командной строке.
- Добавлена горячая клавиша и пункт меню для переключения между кодом и просмотром страницы.
- На панель инструментов добавлены кнопки для выравнивания текста в HTML (по центру, по левому краю и т.п.).
- Пункт меню для копирования заголовка страницы в буфер обмена.
- Перед перезагрузкой вики теперь программа спрашивает, хотите ли вы сохранить текущую страницу.
- Добавлен пункт меню Edit (Undo/Redo/Cut/Copy/Paste).
- Исправлено копирование в буфер обмена из режима просмотра HTML.
- Мелкие исправления и улучшения.
- Подробности в блоге.

### 1.0 alpha 3 (08.06.2010)

- Добавлена возможность поиска по странице.
- Кнопки для основных тегов на странице HTML.

- Добавлена возможность создания ссылок на прикрепленные файлы.
- Добавлена кнопка для сброса флажков со всех тегов при глобальном поиске.
- Добавлены новые иконки для страниц.
- Добавлена возможность прикреплять файлы к странице, перетаскив их в окно программы.
- Улучшен вывод списков на HTML-страницах.
- Исправлены некоторые баги.
- Подробности в блоге.

### 1.0 alpha 2 (19.05.2010)

- Возможность удаления и переименования страниц.
- Глобальный поиск по тексту заметок и по тегам.
- Добавлен новый тип страниц - поиск (search).
- Возможность запуска прикрепленных файлов.
- Возможность создания ссылок между страницами.
- Страницы в дереве теперь всегда сортируются по алфавиту.
- Устранены некоторые глюки.
- Несущественные изменения в интерфейсе.
- Подробности в блоге.

### 1.0 alpha 1 (04.05.2010)

- Первая публичная версия. Реализованы основные возможности.

## Как помочь проекту

Если вы хотите помочь развитию проекта OutWiker, то вы можете сделать это следующими способами:

1. Сделать новую локализацию интерфейса OutWiker или дополнить существующий с помощью сервиса [crowdin.net](http://crowdin.net) (см. раздел *Локализация OutWiker и плагинов (не написано)*).
2. Проверить на ошибки справку или документацию. Особенно требует проверки англоязычная справка.
3. Создать новый стиль для заметок (см. раздел *Как создать свой стиль для страниц заметок (не написано)*).
4. Создать новый плагин (см. раздел *Создание плагинов для OutWiker (не написано)*).
5. Предложить исправление или дополнение в исходный код OutWiker или какого-либо плагина (см. раздел *Как скачать исходные коды OutWiker*).
6. Сообщить о найденной ошибке или предложить новую идею на [Github Issues](https://github.com/OutWiker/OutWiker/issues).
7. Сделать денежный перевод с помощью Яндекс.Деньги на номер *41001580829031* (деньги пойдут на оплату хостинга).

## Ссылки

- Страница программы (стабильная версия)
- Нестабильные версии программы
- Плагины для программы
- Исходные коды
- Сервис для локализации
- Багтрекер и пожелания
- Документация

## OutWiker в социальных сетях

- OutWiker ВКонтакте
- OutWiker в Facebook
- OutWiker в Google+
- OutWiker в Twitter
- OutWiker в Telegram

---

Indices and tables

---

- genindex
- modindex
- search





## Symbols

`_getExecutable()` (метод `BaseBuilder`), 37  
`_getExtraBuildExeOptions()` (метод `BaseBuilder`),  
38  
`_getExtraIncludeFiles()` (метод `BaseBuilder`), 38  
`_getPathIncludes()` (метод `BaseBuilder`), 38

## B

`BaseBuilder` (встроенный класс), 37  
`build()` (метод `BaseBuilder`), 37

## W

`win()` (встроенная функция), 22