

World of Tanks: Mod Packages

version 0.6, 2017-10-18

World of Tanks 9.20.1

Автор		Контактная информация
Anton Bobrov	Wargaming.net	
Mikhail Paulyshka	XVM team	mixail@modxvm.com
Andrey Andruschyshyn	Individual	
Сообщество Koreanrandom.com		

Лицензия

- документация : [CC BY-SA 4.0](#)
- примеры: [WTFPL](#)

1. Общая информация

Пакеты - это способ организации файлов модификаций, в котором весь контент отдельной модификации упакован в один файл.

В случае использования старой схемы распространения файлов, модификации устанавливаются в каталог `<каталог WoT>/res_mods/<версия WoT>/`. При этом, файлы разных модификаций располагаются в одних и тех же каталогах, и поэтому зачастую трудно определить, какой файл относится к какой именно модификации.

Переход на дистрибуцию в пакетах может существенно упростить организацию файлов модификаций: для установки пользователю достаточно скопировать пакет в каталог `<каталог WoT>/mods/<версия WoT>/`, а для деинсталляции - удалить ровно тот же файл.

2. Структура пакета

Пакет представляет собой **zip**-архив со следующими свойствами

- без сжатия
- расширение: `.wotmod`
- максимальный размер архива: 2 ГиБ - 1 байт (2 147 483 647 байт)

Внимание: архивы с компрессией в текущей версии World of Tanks не поддерживаются, поэтому, при создании архивов необходимо устанавливать опцию «уровень сжатия» в значение «без сжатия».

Внимание: архивы размером 2 ГиБ и больше в текущей версии World of Tanks не поддерживаются, поэтому большие пакеты необходимо разбивать на более мелкие, размер каждого из которых не будет превышает 2 ГиБ - 1 байт.

Внутри себя пакет содержит:

- обязательно: каталог `/res/`. Сюда помещаются ресурсы модификации, то есть все те файлы, которые раньше устанавливались в `<каталог WoT>/res_mods/<версия WoT>`;
- опционально: служебный файл `meta.xml` (смотрите **раздел 5**);
- опционально: файл `LICENSE` с лицензионным соглашением;
- опционально: любой иной контент, который может понадобиться автору модификации: ссылка на сайт модификации, документация, список изменений и тому подобное.

Пример структуры пакета:

```
/package.wotmod
  /meta.xml
  /README.md
  /LICENSE
  /res
    /scripts
      /client
        /gui
          /mods
            /mod_example.pyс
```

3. Установка пакета

Пакеты устанавливаются в `<каталог WoT>/mods/<версия WoT>`. Это может быть как ручное копирование, так и установка посредством инсталлятора модификации или сборки модификаций.

При необходимости, пакеты могут быть организованы в подкаталоги, что позволяет авторам сборок модификаций выполнять группировку файлов:

```
mods/
  0.9.17.1/
    MultiHitLog_2.8.wotmod
    DamagePanel/
      Some_common_library_3.14.5.wotmod
      DamagePanel_2.6.wotmod
      DamagePanel_2.8.wotmod
      DamagePanel_2.8_patch1.wotmod
```

4. Рекомендации по именованию пакетов

Для формирования идентификатора пакета (в дальнейшем `package_id`) рекомендуется использовать следующую схему:

```
package_id = author_id.mod_id
```

Где:

- `author_id`: идентификатор автора. Может быть инвертированным доменом разработчика модификации (`com.example`) или же просто его никнеймом (`noname`);
- `mod_id`: идентификатор модификации. Выбирается автором модификации произвольно.

Идентификатор пакета используется в поле `<id>` файла `meta.xml` (смотрите **раздел 5**), а также как часть имени файла пакета.

Пример идентификаторов пакетов:

- `com.example.coolmod`;
- `noname.supermod`.

Имя файла пакета формируется по следующей схеме:

```
<author_id>.<mod_id>_<mod_version>.wotmod
```

Где:

- `mod_version`: версия модификации, задаётся автором в поле `<version>` файла `meta.xml` (смотрите **раздел 5**).

Примеры имён файлов:

- `com.example.coolmod_0.1.wotmod`;
- `noname.supermod_0.2.8.wotmod`.

5. Файл метаданных `meta.xml`

Необязательный файл `meta.xml` содержит опциональные поля, описывающие модификацию.

Пример заполнения:

```
<root>
  <!-- Идентификатор пакета -->
  <id>noname.crosshair</id>

  <!-- Версия пакета -->
  <version>0.2.8</version>

  <!-- Имя пакета понятное для пользователя -->
  <name>Crosshair</name>

  <!-- Описание пакета -->
  <description>New cool Crosshair with feature1.....N</description>
</root>
```

Значения из полей `<id>` и `<version>` используются для определения порядка монтирования пакетов. Значения полей `<name>` и `<description>` в будущем будут использоваться в системе управления модификациями.

6. Загрузка пакетов

6.1 Порядок загрузки

Все пакеты, находящиеся в каталоге `<каталог_WoT>/mods/<версия_WoT>/`, сортируются по значениям узлов `<id>` и `<version>` в файле `meta.xml` и загружаются по порядку. В случае отсутствия файла `meta.xml` внутри пакета, в качестве идентификатора пакета будет использовано имя файла.

Для переопределения порядка загрузки может быть использован файл `load_order.xml`, который должен находиться в вышеуказанном каталоге.

В случае, если все пакеты указаны в `load_order.xml`, то загрузка выполняется в указанном в файле порядке.

В случае, если часть пакетов в `load_order.xml` не указана, то сначала выполняется загрузка пакетов из `load_order.xml`, а затем отсутствующих в нём пакетов в алфавитном порядке.

6.2 Совместное использование пакетов и `res_mods`

С точки зрения клиента игры, корень виртуальной системы формируется из:

- `/res_mods/<версия_WoT>`
- `/mods/<версия_WoT>/<имя_пакета>.wotmod/res/`
- `/res/packages/*.pkg/`
- `/res/`
- Иные пути, указанные в файле `<Каталог_WoT>/paths.xml`

Пути указаны в порядке уменьшения приоритета. То есть, файлы из `/res_mods/<версия_WoT>/` имеют наивысший приоритет вне зависимости от содержания `load_order.xml`

6.3 Разрешение конфликтов при загрузке

В общем случае, система пакетов не допускает ситуации, когда в каталоге `res/` внутри разных пакетов находятся одинаковые файлы. Такая ситуация считается конфликтом.

В случае обнаружения конфликта, вызвавший его пакет не будет загружен, а пользователю будет показано соответствующее уведомление.

Например, если пакеты `a.wotmod` и `b.wotmod` будут содержать внутри себя файл `res/scripts/entities.xml`, то `a.wotmod` будет загружен успешно, а `b.wotmod` вызовет конфликт и загружен не будет.

Для управления обработкой конфликтов можно использовать следующие механизмы:

1. Файл `load_order.xml`

Файл `load_order.xml` должен находиться в каталоге `<каталог_WoT>/mods/<версия_WoT>/` и формируется следующим образом:

```
<root>
  <Collection>
    <pkg>имя_пакета_1.wotmod</pkg>
    <pkg>имя_пакета_2.wotmod</pkg>
    <!-- ... -->
    <pkg>имя_пакета_N.wotmod</pkg>
  </Collection>
</root>
```

Пакеты, перечисленные в этом файле, не считаются конфликтующими и загружаются без анализа на пересечение имен файлов. Приоритет имеет файл из пакета, который был указан последним.

2. Значение узлов `<id>` и `<version>` из `meta.xml`

При наличии узла `<id>` в файле `meta.xml`, имя файла пакета не влияет на порядок загрузки. Пакеты, у которых `<id>` совпадает, считаются разными версиями или частями одной и той же модификации, и конфликты между ними также не учитываются. Данные пакеты загружаются в порядке увеличения версии, которая хранится в узле `<version>`.

Версии пакетов сравниваются посимвольно согласно таблице ASCII. Поведение аналогично функции [strcmp\(\)](#):

- версия `9.0.0` приоритетнее версии `10.0.0`;
- версия `b` приоритетнее версии `B`;
- версия `с<любые символы>` приоритетнее версии `с`;
- если версии совпадают, то приоритет у пакета, имя файла которого будет первым по алфавиту.

При наличии одноименных файлов внутри разных пакетов, конфликты между которыми разрешены посредством `load_order.xml` или `meta.xml`, приоритет имеет файл из пакета, который был подключен последним.

6.4 Исполнение Python-кода

После монтирования всех пакетов и разрешения конфликтов, происходит исполнение всех `.pyc`-файлов из каталога `/scripts/client/gui/mods/` в алфавитном порядке, имя которых начинается с `mod_`.

В пакете данный файл должен быть расположен по следующему пути:

```
<author_id>.<mod_id>_<version>.wotmod/res/scripts/client/gui/mods/mod_<anything>.pyc
```

7. Рекомендованные пути для файлов модификаций

7.1 Файлы конфигурации

Для хранения файлов конфигурации модификаций рекомендованно использовать путь

```
<каталог_WoT>/mods/configs/<author_id>.<mod_id>/
```

Где:

- `author_id` и `mod_id` - идентификаторы, описанные в **разделе 4** данной спецификации.

7.2 Файлы журналов

Помимо штатного файла `python.log`, для хранения журналов рекомендованно использовать путь:

```
<каталог_WoT>/mods/logs/<author_id>.<mod_id>/
```

Где:

- `author_id` и `mod_id` - идентификаторы, описанные в **разделе 4** данной спецификации.

7.3 Временные файлы

Для хранения временных файлов модификации рекомендованно использовать следующий путь:

```
<temp>/world_of_tanks/<author_id>.<mod_id>/
```

Где:

- `temp` - путь к каталогу с временными файлами для текущего пользователя в ОС;
- `author_id` и `mod_id` - идентификаторы, описанные в **разделе 4** данной спецификации.

7.4 Иные файлы модификации

Для хранения внутри пакета контента, к которому необходимо получить доступ из клиента игры, рекомендованно использовать следующий путь:

```
<имя_пакета>.wotmod/res/mods/<author_id>.<mod_id>/
```

Где:

- `author_id` и `mod_id` - идентификаторы, описанные в **разделе 4** данной спецификации.

8. Работа с файлами внутри пакетов

Для работы с файлами внутри пакетов необходимо использовать модуль `ResMgr`.

8.1 Типовые операции

8.1.1 Чтение файла из пакета

```

#импорт
import ResMgr

#функция
def read_file(vfs_path, read_as_binary=True):
    vfs_file = ResMgr.openSection(vfs_path)
    if vfs_file is not None and ResMgr.isFile(vfs_path):
        if read_as_binary:
            return str(vfs_file.asBinary)
        else:
            return str(vfs_file.asString)
    return None

#пример использования
myscript = read_file('scripts/client/gui/mods/mod_mycoolmod.рус')

```

8.1.2 Получение списка элементов в каталоге

```

#импорт
import ResMgr

#функция
def list_directory(vfs_directory):
    result = []
    folder = ResMgr.openSection(vfs_directory)

    if folder is not None and ResMgr.isDir(vfs_directory):
        for name in folder.keys():
            if name not in result:
                result.append(name)

    return sorted(result)

#пример использования
content = list_directory('scripts/client/gui/mods/')

```

8.1.3 Копирование файла из пакета в каталог

```
#импорт
import os
import ResMgr

#функция
def file_copy(vfs_from, realfs_to)
    realfs_directory = os.path.dirname(realfs_to)
    if not os.path.exists(realfs_directory):
        os.makedirs(realfs_directory)

    vfs_data = file_read(vfs_from) #смотрите 8.1.1
    if vfs_data:
        with open(realfs_to, 'wb') as realfs_file:
            realfs_file.write(vfs_data)

#пример использования
file_copy('scripts/client/gui/mods/mod_my.py', 'res/mods/0.9.17.1/scripts/client/gui/mods/mod_my.py')
```

9. Известные проблемы

9.1 Исполнение `.py` файлов

Описание проблемы

На данный момент невозможно исполнение `.py`-файлов, которые размещены внутри пакета.

Временное решение

Размещать в пакете не только `.py`, но и скомпилированные в байткод `.pyc`-файлы.

9.2 Неполная поддержка формата ZIP

Описание проблемы

На данный момент невозможно использование `.wotmod`-файлов, которые не содержат структуры `ZIPDIRENTRY` и `ZIPFILERECORD` для всех каталогов внутри архива.

Временное решение

Использовать для генерации архива совместимые архиваторы, например:

- 7-Zip <http://7-zip.org> ;
- Info-ZIP <http://info-zip.org/> .

Приложение А. Список изменений

v 0.6 (2017-10-18)

- Добавлена лицензия для примеров кода

v 0.5 (2017-10-12)

- Убраны пункты 9.2 и 9.3 (исправлено в World of Tanks 9.20.1)
- Добавлено описание проблемы, которая связана с неполной поддержкой формата ZIP.

v 0.4 (2017-05-04)

- Переработано описание разрешения конфликтов между пакетами с помощью файла `load_order.xml`.

v 0.3 (2017-05-03)

- Добавлена информация про ограничения формата файлов `.wotmod`;
- дополнено описание алгоритма разрешения конфликтов для пакетов с одинаковым идентификатором

v 0.2 (2017-04-10)

- Переработка оформления: новая вёрстка, разбиение на главы;
- переработано описание схемы именования пакетов;
- переработано описание порядка подключения пакетов;
- добавлены рекомендации насчёт мест хранения журналов и файлов конфигурации;
- добавлены примеры исходного кода для работы с файлами внутри пакетов;
- добавлено описание известных на текущий момент проблем.

v 0.1 (2017-01-13)

- Первая версия.