

# World of Tanks: Mod Packages

---

version 0.2, 2017-04-10

- Anton Bobrov, Wargaming.net
- Mikhail Paulyshka, XVM team
- Andrey Andruschyshyn, Independent
- Koreanrandom.com community

License: [CC BY-SA 4.0](#)

## 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 без сжатия с расширением `.wotmod`.

**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».

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.pyc
```

## 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 имени пакета (в дальнейшем `package_id`) рекомендована следующая схема формирования:

```
package_id = author_id.mod_id
```

Где:

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

Данное имя используется в файле `meta.xml` (смотрите **раздел 5**), а также как часть имени файла пакета.

Пример имён пакетов:

```
com.example.coolmod
noname.supermod
```

Имя файла пакета формируется следующим образом:

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

Где:

- `mod_version`: версия модификации, задаётся автором в `meta.xml` (смотрите [раздел 5](#)).

Примеры:

```
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>  
  <!-- Technical MOD ID -->  
  <id>noname.crosshair</id>  
  
  <!-- Package version -->  
  <version>0.2.8</version>  
  
  <!-- Human readable name -->  
  <name>Crosshair</name>  
  
  <!-- Human readable description -->  
  <description>New cool Crosshair with feature1.....N</description>  
</root>
```

Значения из указанных полей 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 значению узла `<id>` 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.

**Внимание:** в данный момент использование файла `load_orders.xml` затруднено (смотрите [раздел 9.4](#)).

## 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, пакет, его вызвавший, целиком исключается из обработки, о чем пользователю выдается соответствующее уведомление.

То есть, в случае если оба пакета `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:

1. Файл `load_order.xml`. Пакеты, перечисленные в этом файле, не считаются конфликтующими и загружаются без анализа на пересечение имен файлов.
2. Значение узла `<id>` из `meta.xml`. Пакеты, у которых `<id>` совпадает, считаются разными версиями или частями одной и той же модификации, и конфликты между ними также не учитываются.

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.псч')
```

## 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.рус', 'res_mods/0.9.17.1/scripts/client/gui/mods/mod_my.рус')

```

## 9. Известные проблемы

### 9.1 Регистрозависимость имён файлов

#### Описание проблемы

На данный момент, при монтировании файлов в виртуальную файловую систему:

- файлы из пакетов монтируются в нижнем регистре
- файлы из `<каталог_WoT>/res_mods/` монтируются как есть

В результате, если файл содержится как в пакете, так и в `res_mods`, и при этом содержит хоть один символ в имени в верхнем регистре, то есть вероятность повторной загрузки файла.

#### Временное решение

Использовать для `<каталог_WoT>/res_mods` только имена файлов и каталогов в нижнем регистре.

### 9.2 Работа с файлами GNU Gettext

#### Описание проблемы

На данный момент невозможно переопределить `.mo`-файлы из `<каталог_WoT>/res/text/LC_MESSAGES/` файлами из пакета.

#### Временное решение

В качестве временного решения, вы можете использовать пакет `net.openwg.vfsgettext`:

[http://openwg.net/download/vfsgettext/net.openwg.vfsgettext\\_1.0.0.wotmod](http://openwg.net/download/vfsgettext/net.openwg.vfsgettext_1.0.0.wotmod)

### 9.3 Исполнение `.рус` 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.

## 9.4 Переопределение порядка монтирования пакетов

---

### Описание проблемы

На данный момент невозможно переопределить порядок монтирования пакетов используя файл `load_order.xml`.

### Временное решение

Нет известного решения. Решение ожидается в скором времени.

## Приложение А. Список изменений

---

### 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