

Содержание

- deb пакет** 3
- DEBIAN/** 3
 - Обязательное содержимое 3
 - Опциональное содержимое 4
 - md5sums 4
 - Скрипты для установки/удаления пакета 4
 - templates 6
 - conffiles 6
 - Утилиты для работы/генерации содержимого DEBIAN/ 6
 - dpkg-gencontrol 6
 - dpkg-shlibdeps 6
 - dpkg-parsechangelog 7
 - dpkg-architecture 7
 - Проверка соответствия пакета современным требованиям Debian 7

deb пакет

deb пакеты делятся на бинарные (.deb) и исходные (.tar.gz).

Deb-пакет это обычный архив файлов, содержащий файлы, предназначенные для установки в систему, а так же некоторые служебные файлы, необходимые для того чтобы эту установку сделать гибкой. При помощи программы ar упаковано в один файл:

- Архив control.tar.gz, содержащий скрипты, написанные майнтенером пакета, использующиеся при установке/удалении пакета, а так же другие служебные файлы;
- Архив data.tar.gz, содержащий двоичные файлы (в случаи с бинарным пакетом) программы, ради которой создан пакет;
- Файл debian-binary.

Собирается пакет программой dpkg из специально подготовленной структуры каталогов:

```
mypkg/  
  DEBIAN/  
    path/to/dir/file1  
    path/to/dirX/fileX
```

то есть:

- Файлы и каталоги, предназначенные для установки в систему. Их расположение в архиве соответствует положению их в файловой системе если считать от корня. Например файл `usr/share/doc/package/copyright` в deb-архиве после установки будет находиться в `/usr/share/doc/package/copyright` (все они будут упакованы в архив data.tar.gz);
- Каталог DEBIAN/, содержащий служебную информацию о пакете (о ней пойдет речь ниже). Содержимое этого каталога при сборке будет упаковано в архив control.tar.gz;

DEBIAN/

Обязательное содержимое

Файл control - этот файл содержит всю описательную информацию на пакет, как то:

- Короткое описание пакета (одна строка);
- Полное описание пакета (один или несколько абзацев);
- Имя и координаты майнтенера, создавшего пакет;
- Номер версии программы (пакета);
- Имя раздела которому принадлежит пакет;
- Перечень пакетов от которых зависит пакет (несколько видов зависимостей: строгие, нестрогие, конфликты итп);
- Уровень важности пакета;
- Имя и версию src-пакета из которого собран данный deb-пакет (если есть таковой);
- Архитектуру для которой данный пакет собран;
- Некоторые другие параметры.

[подробнее о control-файле](#)

Опциональное содержимое

md5sums

md5sums - Содержит md5 хеши для всех файлов кроме файлов находящихся в каталоге DEBIAN/. Данный файл необязателен для deb-пакета, однако программы верификации пакетов считают пакеты, несодержащие этот файл ошибочными. Может использоваться некоторыми программами администрирования системы для верификации изменений в файловой системе.

Скрипты для установки/удаления пакета

Пакет может содержать несколько скриптов (или программ), которые будут вызываться при установке/удалении пакета. Эти скрипты позволяют майнтенеру выполнять некоторые действия при установке/удалении. Например этими скриптами могут создаваться/удаляться каталоги в которых программа будет хранить свои временные данные, может производиться добавление/удаление программы в меню вашего оконного менеджера, первичное конфигурирование программы и так далее.

Виды скриптов:

- `preinst` - скрипт, вызывающийся перед распаковкой пакета в файловую систему. Обычно при помощи этого скрипта пользователю задаются вопросы, позволяющие позже сконфигурировать программу. Так же возможно производится сбор информации, которая потребуется позже для конфигурирования программы. Если скрипт завершится с кодом ошибки, то пакет не будет установлен в систему;
- `postinst` - скрипт, вызывающийся после распаковки пакета. При помощи этого скрипта производится конфигурирование уже установленной программы. Конфигурирование происходит с использованием информации полученной на предыдущем этапе от пользователя. Задавать вопросы пользователю из скрипта `postinst` не принято.;
- `prerm` - скрипт, вызывающийся перед удалением пакета. Может задать пользователю некоторые вопросы, связанные с удалением. Например вопрос что делать с лог-файлами итп. Если скрипт завершится с кодом ошибки, пакет останется установленным в систему;
- `postrm` - скрипт, вызывающийся после удаления пакета;
- `config` - скрипт, вызывающийся на стадии конфигурирования и переконфигурирования пакета. Это дополнительный скрипт, который используется системой `Debconf`. Вызывается перед скриптом `postinst`, используется для того чтобы задать пользователю вопросы в системе диалогов `debconf`.

В разных ситуациях эти скрипты вызываются с разными опциями. В общем виде пакет может:

- Устанавливаться в систему (первичная установка);
- Устанавливаться новая версия пакета (`upgrade`);
- Устанавливаться старая версия пакета (`downgrade`);
- Удаляться из системы.

Соответственно порядок вызова скриптов и опции их вызова может быть разный. Скрипты

могут вызываться со следующими параметрами (полный перечень можно найти в [Debian-policy](#)):

параметры

Параметр	Доп. параметры	Описание
preinst		
install	Номер версии*	Вызывается перед распаковкой пакета. Номер версии указывает на пакет, который стоял ранее, но был удален без ключа <code>-purge</code> , то есть конфигурационные файлы сохранены. С данным параметром скрипт вызывается если в системе пакет не установлен.
upgrade	Номер версии	Вызывается перед распаковкой пакета. Номер версии указывает на пакет, который стоял ранее. Выполняется <code>upgrade</code> или <code>downgrade</code> для пакета. Зная текущую версию пакета и сравнив её с передаваемым здесь значением мы можем определить что происходит <code>upgrade</code> или <code>downgrade</code>
abort-upgrade	Номер версии	Вызывается если <code>preinst upgrade</code> вызванное для нового пакета завершилось неудачей. Номер версии соответствует номеру версии нового пакета, который пытались установить. Таким вызовом установленный пакет информируется о том что его попытались неудачно проапгрейдить.
postinst		
configure	Номер версии*	Вызывается после распаковки пакета. Номер версии содержит версию пакета корректно сконфигурированного ранее. То есть того для которого данный вызов завершился без ошибки. По номеру версии скрипт конфигурации может принять решение о том каким образом необходимо производить апгрейд конфигурационных файлов программы.
reconfigure	Номер версии	Вызывается системой <code>Debconf</code> после переконфигурации пакета. Выполняет все необходимые действия по переконфигурации.
abort-upgrade	Номер версии	Вызывается если <code>preinstall upgrade</code> нового пакета завершился неудачей. Новый пакет не устанавливается, а старый при этом может провести собственную переконфигурацию для того чтобы сохранить работоспособность.
prerm		
remove		Вызывается перед удалением пакета.
upgrade	Номер версии	Вызывается перед удалением пакета при его апгрейде. Номер версии указывает на версию устанавливаемого пакета.
failed-upgrade	Номер версии	Вызывается для устанавливаемого пакета, если <code>prerm upgrade</code> удаляемого пакета завершился с кодом ошибки. Номер версии указывает на удаляемый пакет.
postrm		
remove		Вызывается после удаления пакета
purge		Вызывается после удаления пакета. Если вызван <code>dpkg -purge</code>
upgrade	Номер версии	Вызывается после удаления пакета при его апгрейде. Номер версии указывает на новый пакет.
failed-upgrade	Номер версии	Вызывается после удаления пакета при его апгрейде. Вызывается в случае если <code>postrm upgrade</code> вернул код ошибки. Номер версии указывает на удаляемый пакет. Вызов скрипта происходит во вновь устанавливаемом пакете.

Параметр	Доп. параметры	Описание
abort-install	Номер версии*	Вызывается если preinst install вернул код ошибки. Номер версии соответствует номеру версии передаваемому preinst install
config		
configure	Номер версии	Вызывается во время предварительной настройки пакета из dpkg-preconfigure. Этот вызов проходит во время установки/апгрейда пакета. Номер версии указывает на версию установленного на данный момент пакета.
reconfigure	Номер версии	Вызывается при повторной настройке пакета. Номер версии - номер установленной версии пакета. Инициировать вызов можно дав команду dpkg-reconfigure package из командной строки. После этого вызова системой Debconf будет вызван сценарий postinst reconfigure

templates

Если используется возможность конфигурации/реконфигурации пакета в системе Debconf (скрипт config), то этот файл содержит шаблоны диалогов с пользователем.

conffiles

Содержит перечень файлов пакета, которые являются конфигурационными. По одному файлу на одну строку. Эти файлы при апгрейдах пакета заменяться не будут (или же будут задаваться вопросы с предложением о замене). Подробнее о содержимом см. man debconf-devel.

Утилиты для работы/генерации содержимого DEBIAN/

dpkg-gencontrol

Осуществляет генерацию файла control на базе шаблона этого файла, составляемого майнтенером, а так же дополнительных параметров, передаваемых из командной строки. В частности, устанавливает номер версии пакета, архитектуру итп. Номер версии обычно берется из файла changelog, однако иногда бывает необходимо из одного src-пакета собрать несколько deb-пакетов с разными номерами версий. Опция -v поможет Вам в этом.

dpkg-shlibdeps

Вычисляет зависимости для исполняемых файлов и библиотек. Майнтенер обычно указывает Build-зависимости (зависимости сборки), а Depend-зависимости (зависимости необходимые для работы) вычисляются с помощью этой (или подобных) утилит. Такой подход дает возможность не привлекать майнтенера при смене имен библиотек от которых зависит пакет.

dpkg-parsechangelog

Позволяет извлекать из changelog-файла некоторые параметры, вроде номера версии, координат и имени майнтенера итп. Результаты работы этой утилиты могут использоваться как входные параметры для утилит вроде dpkg-gencontrol.

dpkg-architecture

Позволяет извлекать информацию (манипулировать ей) об архитектуре системы для которой собирается пакет или на которой собирается пакет. Выходные данные так же могут использоваться для использования в других утилитах. Например при генерации файла control утилитой dpkg-gencontrol.

Проверка соответствия пакета современным требованиям Debian

После того как пакет создан, можно получить информацию о его содержимом с помощью вышеупомянутой утилиты dpkg:

- `dpkg -I package.deb` - выведет общую информацию о пакете, включая содержимое control-файла;
- `dpkg -c package.deb` - выведет полный перечень файлов пакета.

После этих двух проверок (все файлы на месте, информация в финальном control корректна), можно запустить одну из двух проверки пакета на соответствие текущему полиси.

Такие проверки осуществляются при помощи программы lintian. Проверяются такие параметры, как правильность расположения двоичных файлов, наличие man-страниц, проверяется на наличие ошибок описание пакета в control-файле, просматривается файл copyright и так далее. Идеальный вариант - добиться от создаваемого пакета такого состояния, что lintian не выводит предупреждений/сообщений об ошибках:

- `lintian -is package.deb` - покажет подробную информацию о проблемах в пакете.

From:

<https://wiki.radi0.cc/> - radi0wiki

Permanent link:

https://wiki.radi0.cc/glossary:gnu_linux:deb_pkg

Last update: **2026/03/03 14:07**

