

www.altlinux.ru
www.deepsolver.org

Deepsolver — менеджер пакетов

Руководство пользователя

Содержание

1	Администрирование <i>Deepsolver</i>	3
1.1	Администрирование репозитория пакетов	3
1.1.1	Общие правила использования утилит работы с репозиторием	4
1.1.2	Утилита <i>ds-repo</i>	5
1.1.3	Утилита <i>ds-patch</i>	6
1.1.4	Утилита <i>ds-provides</i>	7
1.1.5	Примеры	7
2	Спецификации	9
2.1	Структура пакета	9
2.2	Алгоритм выбора пакетов для установки	10
2.2.1	Порядок выбора пакета по явному запросу	10
2.3	Репозитории пакетов	11

1 Администрирование *Deepsolver*

Deepsolver для правильной работы требует выполнения некоторых задач администрирования, которые включают в себя задачи обслуживания репозитория пакетов и настройку *Deepsolver* на рабочих местах. Утилиты, предназначенные для администрирования репозитория, ориентированы на опытных пользователей, поэтому некоторые детали реализации отличаются от других компонентов *Deepsolver*. При обслуживании репозитория акцент делается на прозрачность и гибкость выполнения запрошенных действий, интерфейс пользователя на национальные языки не переводится.

1.1 Администрирование репозитория пакетов

Пакеты для установки размещаются на узлах в сети и передаются по одному из распространённых протоколов. Файлы пакетов перед публикацией дополнительной обработки не требуют, но необходимо правильное именование каталогов, в которых они сохраняются (см. разд. 2.3). Все задачи администрирования репозитория сводятся к созданию и поддержке в актуальном состоянии специальной вспомогательной информации, называемой “индексом репозитория”. Индекс используется для хранения подробной информации о наборе пакетов в репозитории. Он доставляется в первую очередь на компьютеры пользователей, и от его актуальности зависит корректность обработки запросов на внесение изменений в состояние ОС. В простейшем варианте индекс хранит перечисление доступных пакетов с необходимыми атрибутами, включая полный список *provides*, из-за чего размер файлов индекса становится чрезмерно большим. Тем не менее, на практике доля *provides*, действительно задействованных в вычислении зависимостей между пакетами, невелика, и это даёт возможность для различных оптимизаций размера хранимой информации.

Правильный подход настройки фильтрации *provides* выбирается исходя из назначения репозитория. Существуют следующие режимы:

1. Фильтрация на основе поиска соответствующих записей *requires/conflicts*. Если репозиторий является единственным источником пакетов для установки на компьютеры пользователей, и нет других сторонних источников ПО, то эффективное сокращение количества *provides* может быть достигнуто за счёт включения режима фильтрации на основе известных *requires/conflicts*. При активации этого режима запись *provides* сохраняется в индексе только в том случае, если известна хотя бы одна запись *requires* или *conflicts*, в которой имя пакета совпадает с именем пакета в *provides*. Информация о версии в этом случае не учитывается. В терминологии *Deepsolver* соответствующие записи *requires/conflicts* иногда называются “ссылками”.

2. Фильтрация на основе каталогов файлов. Режим фильтрации *provides* на основе информации о каталоге файла может применяться только для тех записей *provides*, которые были сгенерированы автоматически на основе списка файлов пакета. *Deepsolver* имеет возможность указания списка каталогов, файлы из которых всегда обрабатываются как допустимые *provides*. Эта возможность полезна для выполнения запросов на установку пакетов по имени файлов. Например, по имени файлов из каталога */usr/bin*.

Если ни один из указанных режимов фильтрации *provides* не используется, то индекс содержит все возможные *provides*. Если указаны оба из них, то в индекс попадает *provides*, если он подходит хотя бы под одно правило.

Deepsolver предоставляет возможность внесения изменений в индекс без полной регенерации, что порождает ряд трудностей при использовании фильтрации *provides* на основе известных *requires/conflicts*. При появлении новых пакетов нарушается общая целостность, поскольку корректное добавление требует восстановления ранее исключённых *provides*. Описанные ниже утилиты позволяют правильно обновлять содержимое индекса и поддерживать его в целостном состоянии.

1.1.1 Общие правила использования утилит работы с репозиторием

Для решения описанных задач *Deepsolver* предлагает три утилиты: *ds-repo*, *ds-patch* и *ds-provides*, назначение которых описывается ниже. Для всех них в качестве одного из аргументов командной строки указывается путь к каталогу с файлами индекса (в выводе справки обозначен именем `INDEX_PATH`). Пользователь должен указать путь непосредственно к каталогу индекса, а не к каталогу репозитория, например, */repo/i586/base.classic*. Аналогично, в каждом случае, когда требуется указание пути к каталогу с пакетами, необходимо указывать каталоги, содержащие файлы пакетов без промежуточных подкаталогов, например, */repo/i586/RPMS.classic*. Несмотря на то, что набор каталогов для организации репозитория пакетов *Deepsolver* строго определён (см. разд. 2.3), для большей гибкости утилиты администрирования его не обрабатывают автоматически. Для всех трёх перечисленных утилит опции командной строки могут быть ключами, управляющими поведением (обычно начинаются с одиночного или двойного тире), и свободными параметрами. Для запрета обработки опции как ключей, т. е. для явного перехода к режиму свободных параметров, пользователь может указать последовательность “- -” (два дефиса). Количество и назначение ключей и свободных параметров для каждой утилиты требуют отдельного описания, но некоторые из них могут использоваться для любой команды. К ним относятся следующие ключи:

- **-h, - -help** — показать экран справки и завершить работу;

- - **-log** — заменить вывод информации о статусе работы на вывод журнала;
- - **-debug** — при выводе журнала на стандартный вывод понизить фильтр записей до уровня отладочной информации.

Без использования режима вывода журнала утилиты предоставляют информацию для пользователя о ходе выполнения запрошенной операции. Если утилиты были вызваны не в интерактивном режиме, то правильнее переключить их вывод на записи журнала и сохранить их в файл.

1.1.2 Утилита *ds-repo*

Утилита *ds-repo* выполняет все действия по созданию индекса для некоторой одной компоненты репозитория. Главными параметрами для неё являются путь к каталогу, где должен быть расположен новый индекс, и множество путей к каталогам с пакетами, которые должны быть зарегистрированы. Целевой каталог создаётся автоматически. Если он уже существует, то не должен содержать других каталогов или файлов. Каталоги с пакетами могут быть перечислены в произвольном порядке и могут содержать как бинарные пакеты, так и пакеты с исходными текстами. Утилита *ds-repo* автоматически распознает их тип и помещает в соответствующий раздел индекса. Имена всех описанных каталогов указываются как свободные параметры командной строки. На первом месте должен быть указан целевой каталог, за которым следует перечисление каталогов с пакетами. Пропуск целевого каталога не допускается. Если ни один из каталогов с пакетами не указан, то просматривается текущий каталог, установленный на момент запуска утилиты.

Поведение утилиты регулируется рядом опций командной строки. Главную роль из них играют опции, направленные на настройку режима фильтрации *provides*, к ним относятся:

- -**r**, - **-references** — включить режим фильтрации *provides* на основе известных *requires/conflicts*;
- -**s**, - **-refs-sources** — указать список дополнительных источников для регистрации *requires/conflicts*. Источники перечисляются через двоеточие и отделяются от ключа пробелом;
- -**d**, - **-dirs** — включить режим фильтрации *provides* на основе каталогов файлов. Каталоги должны быть перечислены через двоеточие и отделяются от ключа пробелом.

Дополнительными источниками для регистрации *requires/conflicts* могут быть другие компоненты репозитория. При перечислении можно указывать каталоги, которые могут содержать как непосредственно файлы пакетов, так

и ранее построенные индексы *Deepsolver*. Ряд ключей управляет другими деталями поведения утилиты *ds-repo*. К ним относятся:

- **-c, - -compression** — указывает алгоритм сжатия файлов индекса;
- **-lb, - -changelog-binary** — включает запись в индекс истории обновления для бинарных пакетов;
- **-ls, - -changelog-source** — включает запись в индекс истории обновления для пакетов с исходными текстами;
- **-nr, - -no-requires** — указывает путь к файлу с регулярными выражениями для исключения записей *requires*;
- **-u, - -user** — добавляет пользовательский параметр в информацию об индексе. Назначение параметра администратор репозитория может определить самостоятельно, им может быть имя поставщика ПО, имя репозитория и т. д. Параметры указываются в форме **имя=значение** и перечисляются через двоеточие.

После создания файлов индекса утилита записывает файл с контрольными суммами. Обратите внимание, что после работы целевой каталог содержит дополнительный файл, не предназначенный для загрузки пользователем. Он содержит вспомогательную информацию, необходимую для правильной работы утилит внесения изменения в построенный индекс.

1.1.3 Утилита *ds-patch*

Утилита *ds-patch* производит включение и исключение файлов из построенного индекса *Deepsolver*. Все параметры, указываемые при вызове *ds-repo*, повторно перечислять не нужно. Они сохраняются в информационном файле индекса репозитория и загружаются автоматически при изменении его содержимого. При запуске *ds-patch* производится проверка контрольных сумм, и если регистрируется факт обнаружения повреждённого файла, работа завершается с ошибкой. После работы суммы изменённых файлов обновляются.

При использовании утилиты *ds-patch* необходимо помнить, что утилита только изменяет набор файлов, перечисленных в индексе, но не обрабатывает множество *provides*, которое после изменения теряет целостность и должно быть отдельно обработано утилитой *ds-provides*, описанной ниже. Целостность списка *provides* нарушается не только в изменённом индексе, но и во всех других взаимосвязанных компонентах репозитория. Таким образом, для них также должна быть произведена обработка утилитой *ds-provides*.

Утилита *ds-patch* ожидает указания одного свободного параметра — каталога с индексом для изменения. Если он не указан, то используется текущий каталог. Файлы для включения или исключения должны перечисляться после ключей **- -add** и **- -del** соответственно. Файлы для включения должны

быть перечислены с указанием абсолютного пути к файлу пакета, в то время как файлы для удаления перечисляются просто по своему имени. Как и утилита *ds-repo*, утилита *ds-patch* производит автоматическое определение типа пакета, и регистрирует его в соответствующем разделе индекса. Обратите внимание, для отделения указания целевого каталога от предшествующего перечисления списка файлов необходимо использовать последовательность “_”.

1.1.4 Утилита *ds-provides*

Утилита *ds-provides* производит исправление множества *provides* в индексе некоторой компоненты репозитория после внесения изменений в неё или в связанные с ней компоненты. Производить запуск этой утилиты требуется только в том случае, если выбран режим фильтрации *provides* на основе списка известных *requires/conflicts*. В качестве свободного параметра при вызове указывается путь к каталогу с файлами индекса. Если он отсутствует, то используется текущий каталог, установленный на момент вызова. В начале работы утилита *ds-provides* проверяет контрольные суммы, и если они не совпадают, то завершает работу аварийно. После выполнения всех необходимых действий файл с контрольными суммами обновляется. В качестве дополнительного параметра может быть указан параметр - **-ref-sources**, назначение которого полностью аналогично назначению одноимённого параметра для утилиты *ds-repo*.

1.1.5 Примеры

Ниже приводится тривиальный пример создания индекса репозитория только для одной архитектуры *i586*, только для одной компоненты *classic* и с фильтрацией *provides* только для файлов из каталога */usr/bin*:

```
# ds-repo -d /usr/bin i586/base.classic i586/RPMS.classic i586/SRPMS.classic
```

Пример создания индексов репозитория с одной компонентой *classic* и для трёх архитектур: *i586*, *noarch*, *x86_64*. Включены оба режима фильтрации *provides*, т. е. используются записи, для которых есть известные *requires/conflicts*, а также файлы из каталогов, перечисленных в переменной окружения *\$PATH*. Используются регулярные выражения для фильтрации *requires* из файла *~exclude*, записи историй обновлений пакетов сохраняются как для пакетов с исходными текстами, так и для бинарных пакетов.

```
# ds-repo -lb -ls -r -s noarch/RPMS.classic -d "$PATH" \  
-nr ~/exclude i586/base.classic i586/RPMS.classic i586/SRPMS.classic
```

```
# ds-repo -lb -ls -r -s noarch/RPMS.classic -d "$PATH" \  
-nr ~/exclude x86_64/base.classic x86_64/RPMS.classic x86_64/SRPMS.classic
```

```
# ds-repo -lb -ls -r -s i586/base.classic:x86_64/base.classic -d "$PATH" \
-nr ~/exclude noarch/base.classic noarch/RPMS.classic noarch/SRPMS.classic
```

Пример аналогичный предыдущему, но без исключения *requires*, без истории обновления пакетов с исходными текстами и с фильтрацией *provides* только по известным *requires/conflicts*:

```
# ds-repo -lb -r -s noarch/RPMS.classic \
i586/base.classic i586/RPMS.classic i586/SRPMS.classic
```

```
# ds-repo -lb -r -s noarch/RPMS.classic \
-nr ~/exclude x86_64/base.classic x86_64/RPMS.classic x86_64/SRPMS.classic
```

```
# ds-repo -lb -r -s i586/base.classic:x86_64/base.classic \
-nr ~/exclude noarch/base.classic noarch/RPMS.classic noarch/SRPMS.classic
```


2 Спецификации

2.1 Структура пакета

Deepsolver обрабатывает структуру пакета так, как это описано ниже. Информация о версии пакета подразумевает наличие следующих компонентов:

- Эпоха;
- Версия;
- Релиз.

Эпоха представляет из себя целое неотрицательное число. Версия и релиз могут быть произвольными наборами символов, за исключением использования символа “-”. При ссылке на некоторую версию другого пакета указание эпохи и релиза необязательно, что подразумевает их произвольное значение (отсутствия указания эпохи подразумевает пакет требуемой версии любой эпохи).

Пакет может предоставлять функциональность других пакетов, указывая информацию об этом в тэге *provides*. Имя *provides* допускает произвольное значение, не обязательно совпадающее с именем какого-либо существующего пакета, и является, скорее, соглашением, что пакет обладает некоторой совместимостью. Для *provides* допускается указание подмножества версии. Неявными *provides* считаются имена всех файлов, хранимых в пакете.

Следующие типы отношений допускаются на множестве пакетов:

Requires: пакет требует обязательное наличие другого пакета, указанного по его имени или по одному из его *provides*. Допускается указание подмножества версии требуемого пакета. В случае указания ограничения версии под *requires* может подходить *provides* только дополненный информацией о версии.

Conflicts: пакет запрещает наличие другого пакета, указанного по его имени или по одному из его *provides*. Допускается указание подмножества версии конфликтующего пакета. В случае указания ограничения версии под *conflicts* может подходить *provides* только дополненный информацией о версии.

Obsoletes: Пакет может указывать, что является обновлением некоторого множества пакетов. При установке такого пакета все пакеты, обновлением которых он является, удаляются из ОС. Попытка их установки после установки обновляющего пакета приводит к ошибке типа “установлена более свежая версия”. Допускается указание подмножества версии обновляемых пакетов. В случае указания ограничения версии под *obsoletes* может подходить *provides* только дополненный информацией о версии.

Установка двух пакетов является невозможной, если для них обнаружены файловые конфликты. Файловыми конфликтами считаются:

- хранение файлов с одинаковыми именами, но с различной *md5*-суммой или с различными атрибутами (права доступа, идентификаторы владельца и группы, отметка времени создания);
- хранение каталогов с одинаковыми именами, но с разными атрибутами (права доступа, идентификаторы владельца и группы, отметка времени создания).

Все записи о конфликтах в пакете применяются только к другим пакетам, т. е. допускается установка в систему самоконфликтующих пакетов. Предполагается, что два пакета из разных источников, но с одинаковыми именами, эпохами, версиями, релизами и отметками времени их создания являются одним и тем же пакетом.

2.2 Алгоритм выбора пакетов для установки

2.2.1 Порядок выбора пакета по явному запросу

При установке пакета, перечисленного в команде пользователя, поиск подходящего кандидата ведётся следующим образом:

1. Если пользователь указал имя файла или запросил установку на основе *URL*, то устанавливается запрошенный пакет без рассмотрения дополнительных вариантов, описанных ниже. Если запрошенный пакет уже установлен, но установленная версия новее запрошенной, то операция не отменяется.
2. Если существует пакет, имя которого точно соответствует запрошенному, то выбирается его самая свежая версия, удовлетворяющая ограничениям. Прочие варианты, описанные ниже, не рассматриваются.
3. Если существуют только пакеты, подходящие под запрос на основе записей *provides*, то выбирается только один из них следующим образом:
 - если пользователь наложил ограничение версии, то рассматриваются только пакеты, соответствующие записи *provides* в которых содержат информацию о версии. Если таких пакетов несколько, то в указанном порядке последовательно выполняется обработка предустановленного списка приоритетов *provides*, сортировка по версии записи *provides* и сортировка по основному имени до тех пор, пока один из методов не исключит неоднозначность выбора;
 - если пользователь ограничения версии не наложил, но все подходящие записи *provides* имеют информацию о версии, выбор осуществляется таким же способом, как описано в предыдущем пункте;

- если пользователь ограничения версии не наложил, и не все подходящие записи *provides* имеют информацию о версии, то выбор производится на основе предустановленного списка предпочтений или, если запись в списке предпочтений отсутствует, путём выбора последнего элемента в списке имён пакетов после сортировки по возрастанию.

2.3 Репозитории пакетов

Репозиторий пакетов *Deepsolver* может содержать два уровня разбиения. Назначение каждого из них администратор должен определить самостоятельно. Если в качестве одного из них выбрано разделение по архитектуре процессора, то рекомендуется использовать для этого первый уровень.

На первом уровне разделение ведётся по имени каталога. Каждый каталог должен иметь имя, содержащее только латинские буквы, цифры, дефис и знак подчёркивания. На втором уровне разбиение производится путём создания нескольких каталогов, имена которых присвоены по следующим правилам:

1. *base.имя* — каталог для индексов компоненты репозитория;
2. *RPMS.имя* — каталог для бинарных пакетов;
3. *SRPMS.имя* — каталог для пакетов с исходными текстами.

Например, если требуется создать репозиторий для архитектур *noarch* и *i586*, для двух компонент *main* и *debuginfo*, то перечень каталогов должен быть следующим:

- *i586/base.main*;
- *i586/base.debuginfo*;
- *i586/RPMS.main*;
- *i586/RPMS.debuginfo*;
- *i586/SRPMS.main*;
- *i586/SRPMS.debuginfo*;
- *noarch/base.main*;
- *noarch/base.debuginfo*;
- *noarch/RPMS.main*;
- *noarch/RPMS.debuginfo*;
- *noarch/SRPMS.main*;
- *noarch/SRPMS.debuginfo*.

Каталог *base.**, предназначенный для хранения индекса, должен содержать следующие файлы:

- *info* — информационный файл с параметрами индекса;
- *.rpms.complete.data* — вспомогательный файл, не предназначенный для загрузки пользователями, с информацией для повторной фильтрации *provides*;
- *rpms.data* — основной список пакетов с информацией о зависимостях между ними;
- *rpms.descr.data* — список пакетов с расширенными описаниями;
- *rpms.filelist.data* — списки файлов для каждого бинарного пакета;
- *srpms.data* — основная информация о пакетах с исходными текстами;
- *srpms.descr.data* — список пакетов с исходными текстами, содержащий расширенную информацию.