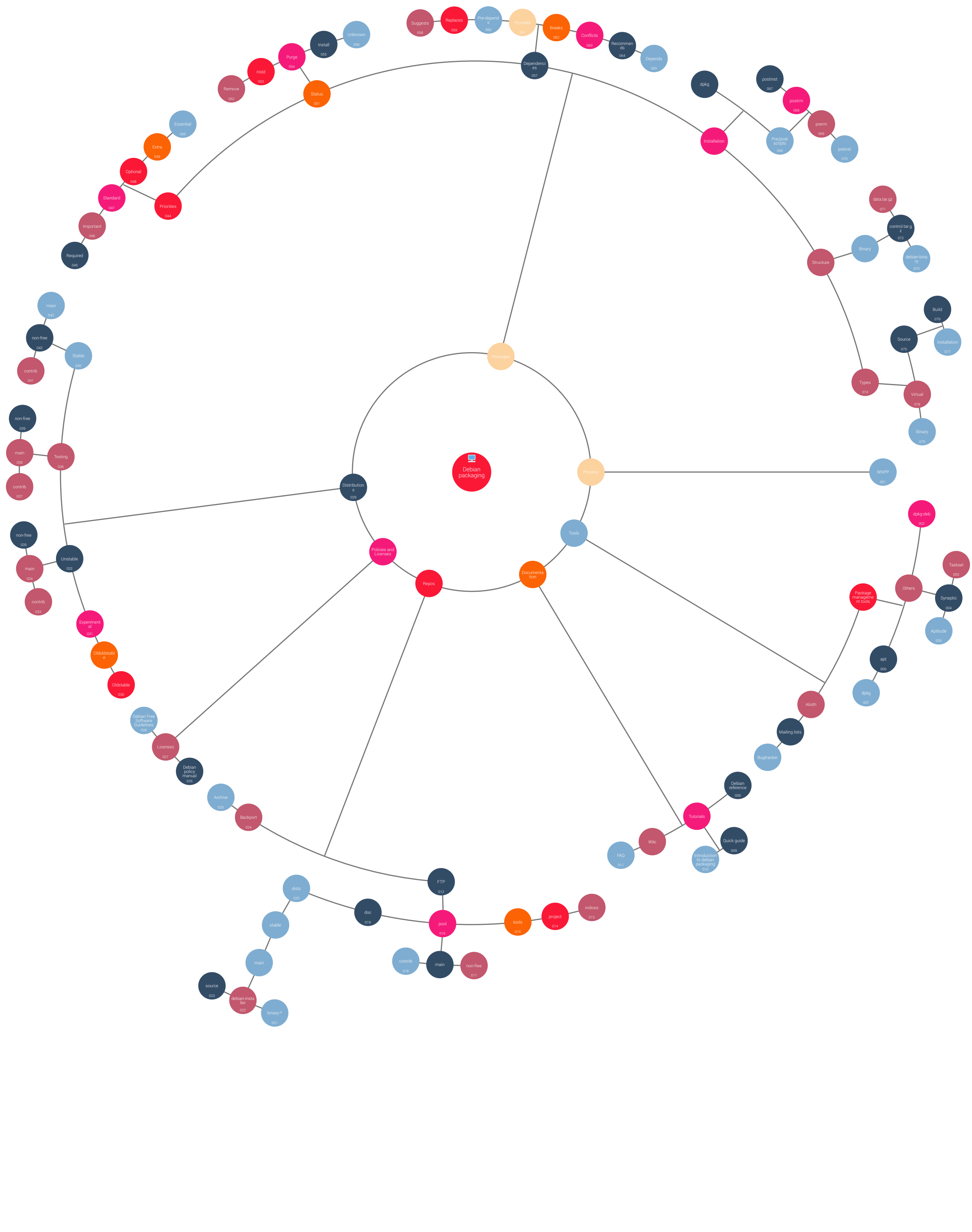
Sep 10, 2017 12:18

Debian packaging



001 WNPP

https://www.debian.org/devel/wnpp/

002 dpkg-deb

dpkg-deb is a package manipulation program that allows the user to, among other things, list the contents of a package, extract the files to a particular directory or extract the control information.

003 Tasksel

A task is a collection of several individual Debian packages all related to a specific activity. Tasks can be installed through the tasksel program or through aptitude.

Typically, the Debian installer will automatically install the task associated with a standard system and a desktop environment. Also, depending on your selections throughout the installation process, tasks might be automatically installed in your system. For example, if you selected a language other than English, the task associated with it will be installed automatically too and if the installer recognises you are installing on a laptop system the laptop task will also be installed.

004 Synaptic

synaptic is a graphical package manager. Along with most of the features offered by aptitude, it also has a feature for editing the list of used repositories, and supports browsing all available documentation related to a package.

http://www.nongnu.org/synaptic/

005 Aptitude

aptitude is a package manager for Debian GNU/Linux systems that provides a frontend to the apt package management infrastructure. aptitude is a text-based interface using the curses library. Actions may be performed from a visual interface or from the command-line.

aptitude provides the functionality of apt-get, as well as many additional features.

006 apt

APT is the Advanced Package Tool, an advanced interface to the Debian packaging system which provides the apt-get and apt programs. It provides commandline tools for searching and managing packages, and for querying information about them, as well as low-level access to all features of the libapt-pkg library.

FAQ:
https://www.debian.org/doc/manuals/debian-faq/ch-pkgtools.en.html#s-apt

man:
https://manpages.debian.org/stretch/apt/apt.8.en.html

007 dpkg

Main package management program. All other package management tools depend on dpkg.

FAQ:
https://www.debian.org/doc/manuals/debian-faq/ch-pkgtools.en.html#s-dpkg

man:
https://manpages.debian.org/stretch/dpkg/dpkg.1.en.html

008 Debian reference

https://www.debian.org/doc/manuals/debian-reference/

009 Quick guide

https://anarc.at/software/debian-development/

010 Introduction to debian

https://www.debian.org/doc/devel-manuals#packaging-tutorial

011 FAQ

https://www.debian.org/doc/manuals/debian-faq/index.en.html#contents

012 FTP

ftp.debian.org us where the packages or the distributions can be downloaded from.

The following subdirectories can be found Theresa.

013 indices

Various indices of the site (the Maintainers file and the override files).

014 project

Mostly developer-only materials and some miscellaneous files.

015 tools

DOS utilities for creating boot disks, partitioning your disk drive, compressing/decompressing files, and booting Linux.

016 pool

The pool directory contains the actual packages.

The source code is distributed in the pool directory together with all the architecture-specific binary directories.

Packages are structured in t'he "pool" directory according to the name of the source package. To make this manageable, the pool is subdivided by section (`main', `contrib' and `non-free') and by the first letter of the source package name. These directories contain several files: the binary packages for each architecture, and the source packages from which the binary packages were generated.

017 non-free

Due to restrictions in their licenses, source code may or may not be available for packages in the "contrib" and "non-free" areas.

018 contrib

Due to restrictions in their licenses, source code may or may not be available for packages in the "contrib" and "non-free" areas.

019 doc

The basic Debian documentation, such as this FAQ, the bug reporting system instructions, etc.

020 dists

The dists directory is short for "distributions", and it is the canonical way to access the currently available Debian releases (and pre-releases).

Inside, the 'stable', 'testing' and 'unstable' directories are symlinks to the corresponding codename.

021 binary-*

Subdirectories that contain index files for binary packages of each available computer architecture, for example binary-i386 for packages which execute on Intel x86 PC machines or binary-sparc for packages which execute on Sun SPARCStations.

The index files in binary-* are called Packages(.gz, .bz2) and they include a summary of each binary package that is included in that distribution. The actual binary packages reside in the top level pool directory.

022 debian-installer

Contains a set of subdirectories (binary-architecture) meant for the installation system index files.

023 source

Contains index files for source packages included in the distribution. The index file is called Sources(.gz, .bz2).

024 Backport

Packages in testing and unstable adapted to work on stable.

025 Archive

Archive is the repo where all previous stable distributions go when the new stable is released (also known as oldstables).

026 Debian policy manual

https://www.debian.org/doc/debian-policy/

027 Licenses

FS guidelines
GPL, BSD, Artistic,...

028 Debian Free Software

https://www.debian.org/doc/debian-policy/ch-archive.html#s-dfsg

029 Distributions

https://www.debian.org/doc/manuals/debian-faq/ch-ftparchives

https://www.debian.org/releases/index.en.html

030 Oldstable

Previous stable distributions.

031 Experimental

Experimental is used for packages which are still being developed, and with a high risk of breaking your system. It's used by developers who'd like to study and test bleeding edge software. Users shouldn't be using packages from there, because they can be dangerous and harmful even for the most experienced people.

032 Unstable

sid or unstable is the place where most of the packages are initially uploaded. It will never be released directly, because packages which are to be released will first have to be included in testing, in order to be released in stable later on. sid contains packages for both released and unreleased architectures.

033 contrib

This directory contains packages which are DFSG-free andfreely distributable themselves, but somehow depend on a package that is notfreely distributable and thus available only in the non-free section.

034 main

This directory contains the packages which formally constitute the most recent release of the Debian GNU/Linux system.

These packages all comply with the Debian Free Software Guidelines, and are all freely usable and distributable.

035 non-free

This directory contains packages distribution of which is restricted in a way that requires that distributors take careful account of the specified copyright requirements.

036 Testing

Packages are installed into the `testing' directory after they have undergone some degree of testing in unstable.

They must be in sync on all architectures where they have been built and mustn't have dependencies that make them uninstallable; they also need to have fewer release-critical bugs than the versions currently in unstable.

More information about the status of "testing" in general and the individual packages is available at http://www.debian.org/devel/testing.

037 contrib

This directory contains packages which are DFSG-free andfreely distributable themselves, but somehow depend on a package that is notfreely distributable and thus available only in the non-free section.

038 main

This directory contains the packages which formally constitute the most recent release of the Debian GNU/Linux system.

These packages all comply with the Debian Free Software Guidelines, and

are all freely usable and distributable.

039 non-free

This directory contains packages distribution of which is restricted in a way that requires that distributors take careful account of the specified copyright requirements.

040 Stable

Current stable production release.

041 contrib

This directory contains packages which are DFSG-free andfreely distributable themselves, but somehow depend on a package that is notfreely distributable and thus available only in the non-free section.

042 non-free

This directory contains packages distribution of which is restricted in a way that requires that distributors take careful account of the specified copyright requirements.

043 main

This directory contains the packages which formally constitute the most recent release of the Debian GNU/Linux system.

These packages all comply with the Debian Free Software Guidelines, and are all freely usable and distributable.

044 Priorities

Each Debian package is assigned a priority by the distribution maintainers, as an aid to the package management system.

https://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html#s-priority

045 Required

Packages that are necessary for the proper functioning of the system.

This includes all tools that are necessary to repair system defects. Should not be removed, as this can leave a system broken.

046 Important

Packages that should be found on any Unix-like system.

Other packages which the system will not run well or be usable without will be here. There are no large applications here, they only constitute the bare infrastructure.

047 Standard

Packages that are standard on any Linux system, including a reasonably small but not too limited character-mode system.

This is what will be installed by default if users do not select anything else. Tools are included to be able to send e-mail, download files from FTP servers, a Python interpreter, some server software like OpenSSH and Exim and some common generic documentation.

048 Optional

Packages that you might reasonably want to install if you do not know what they are, or that do not have specialized requirements.

This includes plots of applications.

049 Extra

Packages that either conflict with others with higher priorities, are only likely to be useful if you already know what they are, or have specialized requirements that make them unsuitable for "Optional".

050 Essential

Packages that are absolutely necessary for the proper functioning of the system. The package management tools will refuse to remove these.

051 Status

The package status seen with dpkg indicates what the user wants regarding the package.

https://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html#s-pkgstatus

052 Remove

The user wants the package removed, but does not want to remove any existing configuration file.

053 Hold

The user wants this package not to be processed, i.e. wants to keep the current version with the current status whatever that is.

054 Purge

The user wants the package to be removed completely, including its configuration files.

055 Install

The user wants the package installed or upgraded.

056 Unknown

The user has never indicated whether the package is wanted.

057 Dependencies

Dependencies are designed to indicate (in a single flag) the level at which Program A can operate independently of the existence of Program B on a given system.

This information us declared in the package's control file.

https://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html#s-depends

058 Suggests

Package A suggests Package B if B contains files that are related to (and usually enhance) the functionality of A.

059 Replaces

Package A replaces Package B when files installed by B are removed and (in some cases) over-written by files in A.

060 Pre-depends

For some packages, dpkg will refuse even to unpack them until certain dependencies are resolved. Such packages are said to "Pre-depend" on the presence of some other packages.

061 Provides

Package A provides Package B when all of the files and functionality of B are incorporated into A. This mechanism provides a way for users with constrained disk space to get only that part of package A which they really need.

062 Breaks

Package A breaks Package B when both packages cannot be simultaneously configured in a system. The package management system will refuse to install one if the other one is already installed and configured in the system.

063 Conflicts

Package A conflicts with Package B when A will not operate if B is installed on the system. Most often, conflicts are cases where A contains files which are an improvement over those in B. "Conflicts" are often combined with "replaces".

064 Recommends

Package A recommends Package B, if the package maintainer judges that most users would not want A without also having the functionality provided by B.

065 Depends

Package A depends on Package B if B absolutely must be installed in order to run A. In some cases, A depends not only on B, but on a version of B. In this case, the version dependency is usually a lower limit, in the sense that A depends on any version of B more recent than some specified version.

066 Pre/post scripts

https://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html#s-maintscripts

067 postinst

This script typically completes any required configuration of the package foo once foo has been unpacked from its Debian archive (".deb") file. Often, 'postinst' scripts ask users for input, and/or warn them that if they accept default values, they should remember to go back and re-configure that package as needed. Many 'postinst' scripts then execute any commands necessary to start or restart a service once a new package has been installed or upgraded.

068 postrm

This script typically modifies links or other files associated with foo, and/or removes files created by the package.

069 prerm

This script typically stops any daemons which are associated with a package. It is executed before the removal of files associated with the package.

070 preinst

This script is executed before the package it belongs to is unpacked from its Debian archive (".deb") file. Many 'preinst' scripts stop services for packages which are being upgraded until their installation or upgrade is completed (following the successful execution of the 'postinst' script).

071 data.tar.gz

data files of the package

072 control.tar.gz

metadata about the package
control, md5sums, (pre|post)(rm|inst), triggers, shlibs, . . .

073 debian-binary

version of the deb file format, "2.0\n"

074 Types

There are two types of packages: binary and source.

https://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html#s-package

075 Source

Source packages contain the source code and consist of a .dsc file describing the source package (including the names of the following files), a.orig.tar.gz file that contains the original unmodified source in gzip-compressed tar format and usually a .diff.gz file that contains the Debian-specific changes to the original source. The utility dpkg-source packs and unpacks Debian source archives.

https://www.debian.org/doc/debian-policy/ch-source.html

076 Build

https://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html#s-sourcebuild

077 Installation

To unpack a source package it is first necessary to configure the "deb-src" lines in apt's sources.list config file.

Afterwards it's just:

apt-get source foo

078 Virtual

A virtual package is a generic name that applies to any one of a group of packages, all of which provide similar basic functionality.

Formatge
example, exim4 and sendmail both provide the functionality of a mail transport agent. They are therefore said to provide the virtual package "mail-transport-agent". If either one is installed, then any program depending on the installation of a mail-transport-agent will be satisfied by the presence of this virtual package.

If more than one package which provide the same virtual package is installed on a system, then system administrators can set one as the preferred package with the update-alternatives command.

https://www.debian.org/doc/manuals/debian-faq/ch-customizing.en.html#s-diverse

https://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.en.html#s-virtual

079 Binary

Binary packages contain executables, configuration files, man/info pages, copyright information, and other documentation. These packages are distributed in a Debian-specific archive format and they are usually characterized by having a '.deb' file extension. Binary packages can be unpacked using the Debian utility dpkg

https://www.debian.org/doc/debian-policy/ch-binary.html