
Manutenzione e Aggiornamenti: gli strumenti di APT

Capitolo 6

6. Manutenzione e Aggiornamenti: gli strumenti di APT	pag. 107
1. Come compilare il file sources.list	pag. 108
1. Sintassi	pag. 108
2. I Repositories per gli utenti della versione Stable	pag. 110
1. Gli Aggiornamenti di Sicurezza	pag. 111
2. Gli Aggiornamenti della versione Stable	pag. 111
3. Proposed Updates	pag. 112
4. Stable Backports	pag. 112
3. I Repositories per gli utenti della versione Testing/Unstable	pag. 112
1. Experimental Repository	pag. 113
4. Come utilizzare dei Mirrors Alternativi	pag. 114
5. Risorse NON Ufficiali: mentors.debian.net	pag. 114
6. Come usare un server Cache Proxy per i Pacchetti Debian	pag. 115
2. I comandi aptitude, apt-get e apt	pag. 116
1. Inizializzazione	pag. 116
2. Installazione e Rimozione	pag. 117
3. Aggiornamento del Sistema	pag. 120
4. Le Opzioni per la Configurazione	pag. 121
5. Gestione delle Package Priorities	pag. 122
6. Come utilizzare più distribuzioni	pag. 124
7. Come resocontare i pacchetti installati automaticamente	pag. 125
3. Il comando apt-cache	pag. 126
4. Il comando apt-file	pag. 128
5. Frontends: aptitude, synaptic	pag. 129
1. aptitude	pag. 129
1. Gestire i Reccomendations, i Suggestions e le Tasks	pag. 131
2. I migliori Solver Algorithms	pag. 132
2. synaptic	pag. 132
6. Verifica dell'autenticità del pacchetto	pag. 133
7. Aggiornamento di una distribuzione alla versione corrente Stable	pag. 135
1. Le procedure raccomandate	pag. 135
2. Come gestire i problemi dopo un aggiornamento	pag. 136
3. "Pulizia," dopo un aggiornamento	pag. 137
1. I pacchetti rimossi dall'archivio Debian	pag. 138
2. Pacchetti Dummy e Transitional	pag. 138
3. File di configurazione obsoleti o non utilizzati	pag. 138
4. I files che non appartengono ad alcun pacchetto	pag. 138
8. Mantenere un Sistema Aggiornato	pag. 139
9. Aggiornamenti Automatici	pag. 141
1. Come configurare dpkg	pag. 141
2. Come configurare APT	pag. 141
3. Come configurare debconf	pag. 141
4. Come comportarsi con le interazioni da riga di comando	pag. 141
5. La combinazione miracolosa	pag. 142
10. Come ricercare i Pacchetti	pag. 142

<<Ciò che rende Debian così popolare tra gli amministratori è la facilità con cui è possibile installare il software ed aggiornare l'intero sistema. Questo straordinario vantaggio è in gran parte dovuto al programma APT, uno strumento che gli amministratori della Falcot Corp hanno studiato con entusiasmo>>

[Per inizializzazione in informatica si intende l'insieme di operazioni necessarie per predisporre un dispositivo, un programma o una risorsa nel suo stato iniziale, a partire dal quale può essere utilizzato.

Per solver, in informatica, si intende una parte del software, che fa uso di algoritmi matematici, dedicata alla ricerca di soluzioni a problemi generici, in modo che tali soluzioni siano applicabili ad ogni situazione si presenti.]

APT è l'abbreviazione di Advanced Package Tool. Ciò che rende questo programma "avanzato" è il suo approccio ai pacchetti. Non solo valuta i pacchetti individualmente, ma li considera nel loro insieme e ottiene la loro migliore combinazione possibile tenendo in considerazione tutto ciò che è disponibile e compatibile (in termini di dipendenze).

DIZIONARIO

Pacchetto
sorgente e la
sorgente dei
pacchetti

Il termine sorgente può essere fuorviante. Non dovete confondere un pacchetto sorgente - un pacchetto contenente il codice sorgente di un programma - con la sorgente di un pacchetto - un repository (un sito Web, un server FTP, un CD-Rom, una local directory, ecc.) che contiene i pacchetti.

APT necessita che gli sia fornita una "lista di sorgenti dei pacchetti (repositories)": il file `/etc/apt/sources.list` elenca i diversi repositories che distribuiscono i pacchetti Debian. APT dovrà quindi importare l'elenco dei pacchetti pubblicati da ciascuna di queste sorgenti, nonché le relative intestazioni. APT realizza questa operazione [l'insieme dei processi e delle attività che utilizzano risorse per trasformare input in output] scaricando ed analizzandone il contenuto: dalle sorgenti dei pacchetti binari i files `Packages.xz` (o in altre varianti che utilizzano differenti metodi di compressione tra cui `.gz`, `.bz2`) e dalle sorgenti dei pacchetti sorgente i files `Sources.xz` (o in altre varianti che utilizzano differenti metodi di compressione). APT è in grado di aggiornare le vecchie copie di questi files già in suo possesso scaricando solo le diseguaglianze (per maggiori dettagli andate a leggere la casella di testo "Aggiornamento incrementale" a pagina 117).

BASILARE

La compressione
attraverso tools
`gzip`, `bzip2`,
`LZMA` e `XZ`

Un'estensione `.gz` di un file compresso è ottenuta attraverso l'utility `gzip`. `gzip` è un'utility rapida ed efficiente per la compressione dei files di tradizione UNIX. Tools più moderni riescono a realizzare le compressioni in tempi più rapidi, ma necessitano di più risorse per comprimere e decomprimere i files (in termini di computation time e memoria). Occorre citare, in un ordine basato sulla data di pubblicazione: `bzip2` (che genera files con estensione `.bz2`), `LZMA` (che genera files con estensione `.lzma`) e `xz` (che genera files con estensione `.xz`).

6.1 Come compilare il file sources.list

6.1.1 Sintassi

Il file `/etc/apt/sources.list` contiene su ogni riga attiva una descrizione della sorgente dei pacchetti (repository), che è divisa in 3 parti separate da spazi. Per maggiori informazioni sulla sintassi del file e sulle voci ammesse fate riferimento a `sources.list(5)`.

Esempio 6.1 Esempio della sintassi delle voci del file `/etc/apt/sources.list`

```
deb url distribution component1 component2 component3 [...] componentx
deb-src url distribution component1 component2 component3 [...] componentx
```

Il primo campo indica il tipo di sorgente:

- **deb** sorgente (repository) dei pacchetti binari.

- **deb-src** sorgente (repository) dei pacchetti sorgente.

Il secondo campo indica l'URL "base" della sorgente che deve essere poi combinato con i filenames contenuti nei files Packages.gz per fornire un URL completo e valido: può essere un mirror Debian o qualsiasi altro archivio di pacchetti impostato da terze parti. L'URL può iniziare con: `file://` per indicare una sorgente locale installata nella gerarchia del file system; `http://` o `https://` per indicare una sorgente accessibile da un server Web; `ftp://` o `ftps://` per indicare una sorgente disponibile su un server FTP.

L'URL può iniziare anche con `cdrom:` per indicare le installazioni da CD-Rom/DVD-Rom/Blu-ray, ma dato che i metodi di installazione via rete sono sempre più diffusi, ciò sarà per voi meno frequente.

La sintassi dell'ultimo campo dipende dalla struttura del repository. Nel più semplice dei casi, potrete solo indicare il nome della subdirectory (ed è d'obbligo una barra slash alla fine) contenente la sorgente desiderata (se non ci sono subdirectory è `./`, – in questo caso i pacchetti saranno quindi disponibili direttamente all'URL specificato). Ma capita comunemente che i repositories siano strutturati come i mirrors Debian ufficiali, con molteplici distribuzioni suddivise ciascuna in molteplici componenti. In questo caso, è necessario che indichiate prima la distribuzione scelta (o attraverso il suo "nome in codice" - andate a leggere la casella di testo "Bruce Perens, un leader controverso" a pagina 9 - o attraverso la sua "suite" - stable, testing, unstable), e poi le componenti (o le sezioni) da abilitare (su un mirror Debian standard, potrete indicare singolarmente o collettivamente `main` e/o `contrib` e/o `non-free`).

DIZIONARIO

Gli archivi Main (principale), Contrib e Non-free

Debian fornisce tre sezioni per differenziare i pacchetti in base alle licenze scelte dagli autori dei rispettivi programmi. Main (archivio principale) raccoglie tutti i pacchetti che soddisfano pienamente le Debian Free Software Guidelines (https://www.debian.org/social_contract.html#guidelines).

L'archivio `non-free` contiene il software che non soddisfa (completamente) le suddette linee guida ma è comunque liberamente fruibile. Questo archivio, che non fa ufficialmente parte di Debian, è un servizio per gli utenti che potrebbero aver bisogno di questi programmi - anche se Debian consiglia sempre di scegliere il software libero. L'esistenza di questa sezione è motivo di attrito con Richard M. Stallman ed impedisce alla Free Software Foundation di raccomandare agli utenti l'uso di Debian.

`Contrib` è uno stock di software libero che non funziona senza elementi `non-free`. Questi elementi possono far parte del software della sezione `non-free` o di files `non-free` come ROM (giochi), BIOS di console, ecc. Inoltre la sezione `Contrib` potrebbe includere software liberi, che per essere compilati potrebbero necessitare di elementi proprietari. La suite per ufficio OpenOffice.org, che inizialmente aveva bisogno di un ambiente proprietario Java, rientrava nel suddetto caso.

SUGGERIMENTO

Il file `/etc/apt/sources.list.d/*.list`

Se fate riferimento a molteplici sorgenti dei pacchetti, può esservi utile separarle in più files. Ciascuna di queste deve essere poi inclusa in `/etc/apt/sources.list.d/filename.list` (andate a leggere la casella di testo "Le Directories che terminano in `.d`" a pagina 121).

Le voci cdrom descrivono il CD/DVD-Rom Debian utilizzato come sorgente. A differenza delle altre voci, di questa sorgente non si può sempre disporre in quanto il disco deve essere presente nel lettore e può essere letto solo una volta: queste sorgenti sono quindi gestite in modo leggermente diverso e sono aggiunte utilizzando il programma apt-cdrom, di solito eseguito con il parametro add. Il suddetto programma vi chiederà quindi di inserire il disco nell'unità e scansionerà il suo contenuto alla ricerca dei files Packages. Dopodiché utilizzerà i files trovati per aggiornare il suo database di pacchetti disponibili (questa operazione è regolarmente realizzabile attraverso il comando apt update). Inoltre come conseguenza a ciò APT potrà richiedervi il disco al bisogno di uno dei pacchetti contenuti dallo stesso disco.

6.1.2 I Repositories per gli utenti della versione Stable

Ecco il contenuto standard del file "sources.list" di un sistema che esegue la versione stabile di Debian:

Esempio 6.2 Il file /etc/apt/sources.list della versione Stable di Debian

```
# Security updates
deb http://security.debian.org/ buster/updates main contrib non-free
deb-src http://security.debian.org/ buster/updates main contrib non-free

## Debian mirror

# Base repository
deb http://deb.debian.org/debian buster main contrib non-free
deb-src http://deb.debian.org/debian buster main contrib non-free

# Stable updates
deb http://deb.debian.org/debian buster-updates main contrib non-free
deb-src http://deb.debian.org/debian buster-updates main contrib non-free

# Stable backports
deb http://deb.debian.org/debian buster-backports main contrib non-free
deb-src http://deb.debian.org/debian buster-backports main contrib non-free
```

Questo file elenca tutte le sorgenti dei pacchetti relative alla versione Buster di Debian (la versione Stable al momento della stesura di questo libro). Nell'esempio è stato scelto di indicare il code-name buster in modo esplicito piuttosto che usare il suo corrispondente alias "stable" (stable, stable-updates, stable-backport) per evitare che la soprastante distribuzione cambi autonomamente e senza alcun controllo da parte dell'utente a causa del successivo rilascio della versione stable.

La maggior parte dei pacchetti proviene dal "base repository", che contiene tutti i pacchetti ma viene aggiornato solo raramente (circa una volta ogni due mesi per il point release) [il point release è una versione minore di un progetto software, destinato a correggere bug o eseguire small cleanups anziché aggiungere funzionalità significative. Spesso la necessità di un point release deriva dal fatto che ci sono troppi bug per essere risolti in una singola versione principale o secondaria]. Gli altri repositories sono parziali (non contengono tutti i pacchetti) ma contengono gli aggiornamenti (delle versioni dei pacchetti più recenti) che APT è in grado di installare. I paragrafi seguenti descrivono in dettaglio gli scopi e le regole che regolano ciascuno di questi repositories.

Si precisa che quando la versione di un pacchetto da voi desiderata è disponibile su più repositories, il pacchetto verrà scaricato dal primo repository utile menzionato nel file sources.list. È per questo motivo che le sorgenti non ufficiali sono generalmente aggiunte alla fine del file.

Si precisa inoltre che la maggior parte di quello che è stato menzionato in questo paragrafo sulla Stable si applica anche sulla Oldstable, in quanto quest'ultima distribuzione è semplicemente una versione precedente della Stable che continua ad essere manutenuta in parallelo.

6.1.2.1 Gli Aggiornamenti di Sicurezza

Debian prende molto seriamente la sicurezza. Le vulnerabilità software note vengono registrate nel Security Bug Tracker (<https://security-tracker.debian.org>), per poi essere risolte in un timeframe ragionevole [timeframe è un lemma tipico nella statistica. Si tratta dell'arco temporale entro cui viene registrata una variazione della curva del grafico per il relativo calcolo al margine]. Gli aggiornamenti di sicurezza (Security Updates) non sono conservati sulla normale rete mirror Debian, ma su security.debian.org (che è un piccolo gruppo di servers manutenuti dal team Debian degli Amministratori di Sistema). Questo archivio contiene gli aggiornamenti di sicurezza (preparati dal Team Debian di Sicurezza e/o dai manutentori del pacchetto) per le distribuzioni Stable e Oldstable.

Questo server può anche conservare gli aggiornamenti di sicurezza della Testing, ma ciò accade più raramente in quanto per la Testing gli aggiornamenti di sicurezza tendenzialmente provengono dal flusso regolare degli aggiornamenti della Unstable.

Di fronte ad un caso di grave serietà il security team emette un Debian Security Advisory (DSA) e rilascia un aggiornamento di sicurezza rendendolo noto attraverso la mailing list debian-security-announce@lists.debian.org (l'archivio degli annunci di sicurezza si trova in <https://lists.debian.org/debian-security-announce/>).

6.1.2.2 Gli Aggiornamenti della versione Stable

Gli aggiornamenti della Stable (Stable Updates) non sono necessariamente dovuti a problemi di sicurezza, ma sono considerati così importanti da essere resi disponibili agli utenti prima della successiva point release della versione stable.

Il repository Stable Updates contiene solitamente fixes di bug critici che non potevano essere corretti prima del rilascio ufficiale della versione stabile di Debian o che si sono presentati in aggiornamenti successivi. A seconda dell'urgenza può contenere anche aggiornamenti per i pacchetti che devono essere migliorati costantemente ad esempio: **spamassassin**, a causa delle sue regole di rilevamento dello spam; **clamav**, a causa del suo database virus; **tzdata** a causa del cambio dell'ora da solare a legale di tutti i fusi orari [il cambio dell'ora da solare a legale in inglese è definito: per USA e Canada, Daylight saving time - DST o daylight savings time o daylight time; mentre per UK e UE, summer time]; la versione **ESR** di Firefox ([firefox-esr](https://www.mozilla.org/en-US/firefox/organizations/)); l'archivio delle chiavi crittografate, definito **keyrings** o portachiavi ([debian-archive-keyring](https://www.debian.org/doc/manuals/debian-reference/ch05.html#keyrings)).

In pratica, questo repository altri non è che una raccolta di proposed-updates, accuratamente selezionati dai Stable Release Managers. Tutti gli aggiornamenti sono resi noti attraverso la mailing list debian-stable-announce@lists.debian.org (l'archivio degli annunci degli aggiornamenti si trova in <https://lists.debian.org/debian-stable-announce/>).

```
deb https://deb.debian.org/debian buster-updates main contrib non-free
```

6.1.2.3 Proposed Updates

Una volta pubblicata, la distribuzione Stable viene aggiornata solo ogni 2 mesi circa. Nel repository Proposed Updates gli aggiornamenti previsti per l'inclusione in Stable sono elaborati, sotto la supervisione dei Stable Release Managers.

I Security Updates e gli Stable Updates, trattati nei paragrafi precedenti, sono anch'essi inclusi nel suddetto repository, ma affermare ciò non è abbastanza; difatti i manutentori dei pacchetti hanno la possibilità di correggere li stesso i significativi bugs che non giustificano comunque un rilascio immediato.

Si precisa inoltre che chiunque può attivare questo repository se desidera testare queste fixes prima della loro pubblicazione ufficiale. Nell'esempio già in basso abbiamo usato l'alias buster-proposed-updates in quanto più esplicito e meno ambiguo considerato che esistono anche i proposed-updates per stretch, essendo quest'ultima la Oldstable corrente al momento della stesura di quest'opera:

```
deb http://deb.debian.org/debian buster-proposed-updates main contrib non-free
```

6.1.2.4 Stable Backports

Il repository stable-backports contiene i "packages backports". Questo termine si riferisce ad un pacchetto relativo a qualche recente software che è stato ricompilato per una distribuzione antecedente, generalmente la Stable.

Quando la distribuzione Stable incomincia ad essere un po' datata, capita che le nuove versioni rilasciate da numerosi progetti software non sono ancora state integrate (infatti la Stable viene modificata per risolvere i problemi più critici, come i problemi di sicurezza). Dal momento che le distribuzioni Testing e Unstable possono assumersi la responsabilità di rischi maggiori, i manutentori dei pacchetti a volte offrono ricompilazioni per la Stable di applicazioni software recenti, il che consente di limitare una potenziale instabilità ad un numero ridotto e selezionato di pacchetti. Per maggiori informazioni:

♦ <https://backports.debian.org>

I backports dei stable-backports sono creati dai pacchetti disponibili nella distribuzione Testing, in modo da garantire che tutti i backports installati possano essere aggiornati alla corrispondente futura versione Stable di Debian quando sarà rilasciata.

Sebbene questo repository fornisca nuove versioni dei pacchetti, APT procederà alla loro installazione solo su istruzioni esplicite (oppure se avete già installato una versione antecedente dei pacchetti in questione attraverso i backports):

```
$ sudo apt-get install package/buster-backports  
$ sudo apt-get install -t buster-backports package
```

6.1.3 I Repositories per gli utenti della versione Testing/Unstable

Ecco un file sources.list standard per un sistema che esegue la versione Testing o Unstable di Debian:

Esempio 6.3 Il file /etc/apt/sources.list della versione Testing/Unstable di Debian

```
# Unstable
deb http://deb.debian.org/debian unstable main contrib non-free
deb-src http://deb.debian.org/debian unstable main contrib non-free
# Testing
deb http://deb.debian.org/debian testing main contrib non-free
deb-src http://deb.debian.org/debian testing main contrib non-free
# Stable
deb http://deb.debian.org/debian stable main contrib non-free
deb-src http://deb.debian.org/debian stable main contrib non-free
# Security updates
deb http://security.debian.org/ stable/updates main contrib non-free
deb http://security.debian.org/ testing/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ testing/updates main contrib non-free
```

DIZIONARIO

La sintassi dei repositories di sicurezza

Onde evitare di creare confusione con la sintassi dei repositories degli aggiornamenti della stable "codename-updates", da Debian 11 Bullseye in poi la sintassi dei repositories degli aggiornamenti di sicurezza verrà cambiata da "codename/updates" a "codename-security", pertanto prendetene nota (per maggiori informazioni leggete il paragrafo 6.1.2.2 Gli Aggiornamenti della versione Stable a pagina 111).

Attraverso il file sources.list dell'esempio soprastante, APT installerà i pacchetti dalla versione Unstable. Se non desiderate ciò, dovete utilizzare l'opzione di configurazione **APT::Default-Release "testing";** (per maggiori dettagli leggete il paragrafo 6.2.3, "Aggiornamento del Sistema" a pagina 120) per imporre ad APT di prelevare i pacchetti da un'altra distribuzione (presumibilmente in questo esempio dalla Testing).

Non è criticabile che un utente possa desiderare di includere tutti questi repositories nel file sources.list, anche se uno sarebbe stato sufficiente. Difatti gli utenti della testing apprezzeranno tale possibilità ovvero di poter scegliere manualmente un fixed package [pacchetto corretto] dalla Unstable quando la versione Testing dello stesso pacchetto è affetta da un fastidioso bug. Al contrario, gli utenti dell'Unstable danneggiati da insospettabili regressioni potranno eseguire il downgrade dei loro pacchetti (a patto che poi continuino a funzionare) alla versione Testing.
L'inclusione del repository Stable è più discutibile, ma potrebbe essere dovuta al fatto che il repository Stable fornisce l'accesso a quei pacchetti che sono già stati rimossi dalle versioni di sviluppo. Inoltre tale inclusione garantisce l'accesso agli ultimi aggiornamenti dei pacchetti che, dal rilascio dell'ultima versione stabile, non sono stati ancora modificati.

[Una software regression (una regressione del software) è un bug del software che interrompe il funzionamento di una funzione. Può presentarsi a seguito di un determinato evento (un aggiornamento del sistema, un system patching o con il cambio dell'ora legale). Una software performance regression (una regressione delle prestazioni del software) si verifica quando il software funziona ancora correttamente, ma più lentamente o utilizza più memoria o risorse rispetto a prima. Le regressioni sono spesso causate anche da hotfix dette anche quick-fix engineering update (QFE update) incluse nelle software patches. Un approccio per evitare questo tipo di problema è il regression testing (il test di regressione). Un test plan (un documento che descrive gli obiettivi, le risorse ed i processi di un test specifico per un prodotto software o hardware) ben progettato dovrebbe essere in grado di prevenire le regressioni prima del rilascio del software. Gli automated testing (test automatici di collaudo del software attraverso un software separato dal software da collaudare) ed i test cases (test di conformità a dei modelli prefissati) ben scritti possono ridurre le probabilità di una regressione. Una regressione del software può essere di tre tipi:

- Local: una modifica introduce un nuovo bug nel modulo o componente modificato.
- Remote: una modifica in una parte del software "interrompe" le funzionalità di un altro modulo o componente.
- Unmasked: una modifica "smaschera" un bug già esistente che non aveva prodotto alcun effetto prima della modifica.]

6.1.3.1 Experimental Repository

L'archivio dei pacchetti Experimental, presente su tutti i mirror Debian, contiene i pacchetti che non appartengono ancora alla versione Unstable a causa delle loro qualità al di sotto degli standard previsti: spesso si tratta di versioni in fase di sviluppo o pre-versioni (alfa, beta, release candidate ...).

Difatti capita spesso che un pacchetto sia inviato nell'Experimental come conseguenza a dei cambiamenti significativi subiti, che potrebbero avere generato problemi. Poi il manutentore cercherà di attenuarne gli effetti con l'aiuto di utenti avanzati che sono in grado di gestirne le anomalie più evidenti. Dopo questa prima fase, il pacchetto sarà trasferito nel repository dell'Unstable, dove avrà modo di giungere ad un pubblico molto più ampio e dove sarà quindi sottoposto a test molto più severi.

Pertanto l'Experimental è solitamente riservata agli utenti che non hanno paura di "rompere" il loro sistema e di essere poi costretti a ripararlo. Questa distribuzione consente inoltre di importare un pacchetto che un utente desidera testare o utilizzare per necessità. Ed è per quest'ultimo suo aspetto che Debian ne usufruisce, in quanto la mera voce del correlato repository nel file sources.list di APT non comporta l'utilizzo sistematico dei suoi pacchetti. Qualora desideriate usufruirne anche voi la voce da aggiungere nel file sources.list è:

```
deb https://deb.debian.org/debian experimental main contrib non-free
```

6.1.4 Come utilizzare dei Mirrors Alternativi

Il file sources.list utilizzato come modello in questo capitolo definisce i repositories resi disponibili da deb.debian.org (<https://deb.debian.org/>). Verrete reindirizzati ai servers che si trovano geograficamente quanto più vicini rispetto alla vostra posizione dal Content Delivery Network, che si occupa appunto delle gestione dei suddetti servers, nonché della memorizzazione e della distribuzione, nei tempi più rapidi possibili, delle copie dei files in tutto il mondo. Le società che contribuiscono al CDN collaborano con Debian offrendo i loro servizi gratuitamente. Ciò nonostante nessun server viene direttamente gestito da Debian e, senza alcuna esternalità, la sicurezza degli archivi si basa esclusivamente sulle firme GPG.

Qualora non vi riteniate soddisfatti delle prestazioni di deb.debian.org potrete fare riferimento alla lista ufficiale dei mirrors, disponibile in:

- ♦ <https://www.debian.org/mirror/list>

Se non siete in grado autonomamente di stabilire quale mirror sia più adatto a voi l'elenco soprastante sarà inutile. Pertanto potrete fare riferimento alle voci DNS che Debian gestisce nel seguente formato: <ftp.country-code.debian.org> (ad esempio <ftp.us.debian.org> per gli USA, <ftp.fr.debian.org> per la Francia, ecc.); Debian attua tale politica allo scopo di poter offrire i propri servizi in diversi paesi nel mondo con i migliori mirrors disponibili.

Infine come alternativa in passato avreste potuto usufruire di <http://predir.debian.org>. Tale servizio attraverso la vostra geolocalizzazione (GeoIP) identificava il mirror quanto più prossimo alla vostra posizione ed inoltrava di conseguenza le istanze APT. Purtroppo è stato deprecato per ragioni di affidabilità ed ormai si basa sul CDN come deb.debian.org.

6.1.5 Risorse NON Ufficiali: mentors.debian.net

Ci sono molte sorgenti non ufficiali di pacchetti Debian preparate da utenti esperti che hanno ricompilato determinati software (Ubuntu ha reso questa pratica diffusa con il loro servizio Personal Package Archive), dai programmati che hanno reso le loro creazioni disponibili a tutti, sino agli sviluppatori Debian che distribuiscono pre-versioni del loro pacchetto online.

Merita di essere menzionato il sito mentors.debian.net - <https://mentors.debian.net/> - (anche se distribuisce solo pacchetti sorgente), che raccoglie i pacchetti realizzati da pretendenti allo status di sviluppatore ufficiale Debian o da volontari che desiderano creare pacchetti Debian senza passare attraverso questo processo di integrazione. Questi pacchetti sono quindi distribuiti senza alcuna garanzia di qualità; pertanto prima di prenderli in considerazione per il vostro sistema assicuratevi di verificarne l'origine e l'integrità e di testarli accuratamente.

COMUNITÀ

I siti di
debian.net

Il dominio debian.net non è una risorsa ufficiale del progetto Debian. Ciascun sviluppatore Debian ha la possibilità di usarne il domain name per uso personale. Questi siti web possono contenere servizi non ufficiali (a volte anche siti personali) ospitati su una macchina non appartenente al progetto anche se configurata da sviluppatori Debian, oppure servizi pilota (prototypes) in attesa di essere trasferiti su debian.org. Sono due le ragioni che possono giustificare come mai alcuni di questi servizi pilota rimangono su debian.net: la prima ragione è che nessuno desidera fare lo sforzo necessario per trasformare il servizio pilota in un servizio ufficiale (ospitato sul dominio debian.org e con una certa garanzia di manutenzione); la seconda ragione è che il servizio pilota è troppo controverso per essere formalizzato.

L'effetto dell'installazione di un pacchetto è di fatto quello di concedere i privilegi di root al suo creatore, poiché quest'ultimo decide il contenuto degli scripts di inizializzazione eseguiti sotto l'account amministratore. I pacchetti ufficiali Debian sono realizzati da volontari selezionati per cooptazione, valutati ed in grado di siglare i loro pacchetti in modo che chiunque possa verificarne l'origine e l'integrità.

In generale fate attenzione a priori quando desiderate usufruire di un pacchetto la cui origine è incerta e che non è ospitato su uno dei server ufficiali del progetto Debian: valutate fino a che punto siete disposti a concedere fiducia al suo creatore e verificate l'integrità del pacchetto.

ANDANDO OLTRE

Le precedenti
versioni dei
pacchetti:
snapshot.debian.org

Il servizio snapshot.debian.org (<https://snapshot.debian.org>), introdotto nell'aprile 2010, può essere utilizzato "per andare a ritroso nel tempo" e trovare una vecchia versione di un pacchetto non più disponibile negli archivi Debian. Infatti può essere utilizzato per identificare quale versione di un pacchetto ha introdotto una regressione, ossia più concretamente, per ripristinare la versione precedente in attesa della fix della regressione.

6.1.6 Come usare un server Cache Proxy per i Pacchetti Debian

Quando un'intera rete di macchine è configurata per scaricare gli stessi pacchetti aggiornati dallo stesso server remoto, qualsiasi amministratore, conoscendone i benefici, preferisce utilizzare un proxy configurato come ad esempio un network-local-cache (per maggiori dettagli leggete la casella di testo "Cache" a pagina 126) in modo da limitare il traffico indotto da download multipli.

APT può essere configurato per utilizzare un proxy "standard" (per maggiori dettagli leggete il paragrafo 6.2.4, "Le Opzioni per la Configurazione" a pagina 121 per la configurazione di APT ed il paragrafo 11.6, "Proxy HTTP/FTP" a pagina 309 per la configurazione del proxy), ma l'ecosistema Debian dispone di soluzioni migliori per risolvere questo inconveniente [ossia, si reitera, un'intera rete di macchine configurata per scaricare gli stessi pacchetti aggiornati dallo stesso server remoto]. Il software dedicato descritto in questo paragrafo è più efficiente di un proxy generico poiché si basa sulla struttura particolareggiata dei repositories APT (ad esempio questi sono in grado di capire

quando i singoli files diventeranno più o meno obsoleti e quindi stabilire per quanto tempo mantenerli).

apt-cacher e apt-cacher-ng si comportano come un server proxy cache standard. Il file sources.list di APT rimane invariato, ma APT viene configurato per utilizzare questi software come proxy durante le richieste in uscita.

Diversamente approx, si comporta come un server HTTP che funge da mirror per altri remote repositories [repositories esterni], resi disponibili attraverso i suoi top level URLs. [In generale il mapping è una tecnica di gestione dei dati (per poter eseguire una migrazione, integrazione, ecc), che consente l'estrazione dei dati dei campi (in inglese fields) di un database ed il loro indirizzamento ai campi corrispondenti di altri database targets. Pertanto, genericamente, i databases i cui dati vengono fra loro indirizzati attraverso questa tecnica possono essere definiti mapping databases] Il mapping tra queste top level directories ed i remote URLs dei repositories è definita nel file di configurazione /etc/approx/approx.conf:

```
# <name>      <repository-base-url>
debian       http://deb.debian.org/debian
security     http://security.debian.org
```

approx per impostazione predefinita svolge le sue funzioni attraverso il port 9999 e tramite il demone inetd. Gli utenti dovranno poi configurare il loro file sources.list in modo che utilizzi il server approx:

```
# Sample sources.list pointing to a local approx server
deb http://apt.falcot.com:9999/security buster/updates main contrib non-free
deb http://apt.falcot.com:9999/debian buster main contrib non-free
```

6.2 I comandi aptitude, apt-get e apt

APT è un vasto progetto, che inizialmente prevedeva un'interfaccia grafica. Si basa su una libreria che racchiude il nucleo elaborativo dell'applicazione ed apt-get è stato il primo front-end, a riga di comando, che è stato sviluppato nell'ambito del progetto. Mentre apt è stato il secondo front-end a riga di comando previsto per APT che corregge alcuni errori di progettazione di apt-get.

Entrambi i summenzionati strumenti si basano sulla stessa libreria, tanto da renderne le funzionalità simili, ma in apt è stata migliorata l'interazione con l'utente in modo che potesse soddisfarlo pienamente. Si precisa però che gli sviluppatori di APT si riservano il diritto di poter cambiare la public interface del loro tool in qualsiasi momento pur di migliorarlo [per "public interface" si intende la logica attraverso la quale un'entità software interagisce con altre entità software all'interno di un singolo computer oppure attraverso la rete]. Diversamente la public interface di apt-get è stata ben definita in modo da renderla retrocompatibile. Di conseguenza è probabile che preferiate apt-get per l'installazione di un pacchetto.

Sono state realizzate molte altre interfacce grafiche come progetti esterni: synaptic (interfaccia grafica), aptitude (che include sia un'interfaccia testuale che un'interfaccia grafica, anche se quest'ultima, al momento della stesura di quest'opera, non è ancora completa), wajig, ecc. L'interfaccia che noi ci sentiamo di raccomandarvi è apt, ovvero quella che utilizzeremo per gli esempi in questo capitolo. Si precisa, tuttavia, che la sintassi da riga di comando di apt-get e aptitude è molto simile. Pertanto i casi in cui sono evidenti le differenze tra apt, apt-get e aptitude saranno trattati con maggiori dettagli.

6.2.1 Inizializzazione

Un prerequisito per qualsiasi interazione con APT è quello di aggiornare l'elenco dei pacchetti disponibili, che viene eseguito con il semplice comando apt-update. Questa operazione può

richiedere più o meno tempo, a seconda della velocità della vostra connessione, in quanto scarica un certo numero di files (Packages, Sources, Translation-language-code) solitamente compressi, che crescono di pari passo con lo sviluppo di Debian (con almeno 10 MB per la sezione main). Certamente l'installazione da un set di CD-ROM/DVD non richiede alcun download ed in questo caso l'aggiornamento dovrebbe essere piuttosto veloce.

SUGGERIMENTO

Aggiornamento incrementale

Come già espresso precedentemente, lo scopo del comando apt update è quello di scaricare i corrispondenti files Packages (o Sources) di ciascuna sorgente dei pacchetti. Tuttavia questi files, anche dopo essere stati compressi con xz, rimangono di grandi dimensioni (i Packages.xz della sezione main di Buster occupano più di 7 MB). Ne consegue che qualora siate costretti ad effettuare aggiornamenti regolarmente, questi downloads potrebbero farvi perdere molto tempo.

Pertanto invece di scaricare un intero file, per accelerare il processo, APT è in grado di scaricare i files "diff" che elencano singolarmente i cambiamenti del correlato pacchetto dal precedente aggiornamento [tale metodo viene definito incrementale]. Per consentire la suddetta operazione i mirrors ufficiali di Debian assegnano numerosi files che elencano le differenze da una versione all'altra di ciascun pacchetto. Inoltre questi files sono creati durante l'aggiornamento degli archivi e viene conservata una copia della cronologia di ogni settimana. Ciascun file "diff" della Unstable generalmente pesa solo una dozzina di kilobyte in modo tale che l'ammontare di dati da ricevere settimanalmente via apt update sia di fatto un decimo. Nelle distribuzioni come la Stable e la Testing, con minori cambiamenti, il guadagno è ancora più evidente.

Ciò nonostante potrete avere comunque bisogno di forzare il download del formato intero del file dei pacchetti, specialmente nei casi in cui l'ultimo aggiornamento è troppo obsoleto e di conseguenza la procedura incrementale non vi può essere d'aiuto. Inoltre, si reitera, potrete essere costretti a forzare il download del formato intero del file dei pacchetti nei casi in cui, pur essendo l'accesso alla rete molto veloce, il processore della macchina da aggiornare è relativamente lento. Di conseguenza il tempo che risparmierete per il download dei files verrà poi perso nell'elaborazione delle nuove versioni di questi files a partire dalle vecchie versioni (incluse le differenze da applicare). Per disattivare l'aggiornamento incrementale, dovete impostare il parametro di configurazione Acquire::Pdiffs su false.

```
$ sudo apt -o "Acquire::Pdiffs=false" update
```

Attraverso l'opzione Acquire::* potrete gestire sia altri aspetti del download, nonché i metodi. Acquire::Languages può limitare o disabilitare il download dei files Translation-language-code e farvi risparmiare ulteriore tempo. Per maggiori informazioni consultate apt.conf(5).

6.2.2 Installazione e Rimozione

Attraverso APT potrete aggiungere o rimuovere pacchetti nel vostro sistema, rispettivamente con apt install package e apt remove package. In entrambi i casi citati, APT installerà automaticamente le dipendenze necessarie o rimuoverà i pacchetti che dipendono dal pacchetto che si sta disinstallando. Il comando apt purge package esegue una disinstallazione completa ovvero verranno eliminati anche i files di configurazione.

SUGGERIMENTO

Come installare la stessa dpkg's selection di pacchetti più volte

Talvolta può essere necessario installare sistematicamente lo stesso elenco di pacchetti su più computers. Tale operazione può essere svolta facilmente. Innanzitutto, dovete ottenere l'elenco dei pacchetti installati sul computer che fungeranno come "modello" da duplicare.

```
$ dpkg --get-selections >elenco-pkg
```

Il file elenco-pkg contiene l'elenco dei pacchetti già installati da usare come modello. Dovrete, quindi, prima trasferire il file elenco-pkg su tutti i computers da aggiornare e poi utilizzare i seguenti comandi singolarmente su ogni singolo computer:

```
## Update dpkg's database of known packages  
# avail='mktemp' [REDACTED]  
# apt-cache dumpavail > "$avail"  
# dpkg --merge-avail "$avail"  
# rm -f "$avail"  
## Update dpkg's selections  
# dpkg --set-selections < elenco-pkg  
## Ask apt-Gent to install the selected packages  
# apt-get dselect-upgrade
```

Il primo comando registra l'elenco dei pacchetti disponibili nel database dpkg, mentre dpkg --set-selections ripristina la selezione dei pacchetti di dpkg in base al vostro elenco di pacchetti da voi scelti per l'installazione su tutti i computers. Tale operazione potrà essere eseguita individualmente su ogni singolo computer con la semplice invocazione di apt-get! aptitude non dispone di questo comando.

SUGGERIMENTO

Rimuovere ed installare simultaneamente

Potrete imporre ad apt (oppure ad apt-get o ad aptitude), aggiungendo un suffisso, di installare determinati pacchetti e di rimuovere altri dalla stessa riga di comando. Per fare ciò, quando utilizzate il comando apt install, dovete aggiungere un “-” (meno) ai nomi dei pacchetti che desiderate rimuovere.

Diversamente, quando utilizzate il comando apt remove dovete aggiungere un “+” (più) ai nomi dei pacchetti che desiderate installare.

L'esempio sottostante mostra come installare package1 e rimuovere package2.

```
# apt install package1 package2-  
[...]  
# apt remove package1+ package2  
[...]
```

Questo metodo consente anche di escludere pacchetti che senzò verrebbero installati, ad esempio a causa di un campo Recommends. Il dependency solver, generalmente tratta queste informazioni come se fossero un suggerimento e di conseguenza ricerca soluzioni alternative.

SUGGERIMENTO

```
apt --  
reinstall e  
aptitude  
reinstall
```

A volte il sistema può danneggiarsi a causa dell'eliminazione o della modifica dei files di un pacchetto. Il modo più semplice con cui potete recuperare questi files è reinstallare il pacchetto stesso. Purtroppo generalmente il packaging system tende a considerare il pacchetto danneggiato già installato, pertanto si rifiuterà pacificamente di reinstallarlo; attraverso l'opzione `--reinstall` dei comandi `apt` e `apt-get` potrete raggiungere il summenzionato disagio noto. Il seguente comando reinstalla postfix anche se già installato.

```
# apt --reinstall install postfix
```

Il comando di `aptitude` è leggermente diverso, ma raggiunge lo stesso risultato:

```
aptitude reinstall postfix.
```

Il suddetto problema non si presenta con `dpkg`, ma raramente un amministratore si serve di questo strumento direttamente.

Fate attenzione! `apt --reinstall` non sarà mai in grado di ripristinare i pacchetti modificati da un attacco informatico e di conseguenza di restituirci un sistema identico a quello che avevate prima. Il paragrafo 14.7, "Come comportarsi con una macchina compromessa" a pagina 440 tratta l'argomento in maniera dettagliata. Questi comandi non ripristineranno i files di configurazione, ma come già accennato nel paragrafo 5.2.3, Checksum, Elenco dei Files di Configurazione a pag. 89 (e nella casella di testo Come forzare `dpkg` a porre le domande per il file di configurazione a pag. 89) potrete richiedere al seguente comando di installare una versione non modificata e persino di ripristinare qualsiasi file di configurazione rimosso.

```
# apt --reinstall -o Dpkg::Options::="--force-confask, confmiss"  
install package
```

Diversi pacchetti non includono il file di configurazione riscontrabile in `/etc`. Ma lo creano durante l'installazione basandosi su uno skeleton [uno stile di programmazione che si basa su strutture di linguaggio di programmazione di alto livello denominate dummy code] oppure attraverso uno script. Ad esempio il file `/etc/inputrc` è la copia di `/usr/share/debian/inputrc`. Pertanto in questi casi il comando sopra citato non potrà funzionare.

Se nel file `sources.list` sono citate diverse distribuzioni, è possibile specificare la versione del pacchetto da installare. È possibile invocare un numero di una specifica versione con `apt install package=version`, anche se generalmente è preferibile indicare solo la distribuzione da cui estrarre il pacchetto (Stable, Testing o Unstable) con la sintassi `apt install package/distribution`. Attraverso quest'ultimo comando, potrete quindi ripristinare una vecchia versione di un pacchetto (qualora siate certi che ancora funzioni bene), a condizione che sia ancora disponibile in una delle sorgenti che cita il file `sources.list`. Altrimenti, qualora ne abbiate bisogno, potrete sempre utilizzare l'archivio di `snapshot.debian.org` (per maggiori dettagli andate a leggere la casella di testo "Le precedenti versioni dei pacchetti: `snapshot.debian.org`" a pagina 115).

Esempio 6.4 Installazione della versione Unstable di spamassassin

```
# apt install spamassassin/unstable
```

Se dovete installare un pacchetto reso disponibile come file `.deb` a cui non è associato alcun repository, potrete comunque usare APT per installarlo con le sue dipendenze (a condizione che le

dipendenze siano disponibili nei repositories da voi configurati) con il comando: `apt install ./path-to-the-package.deb`. Occorre precisare che `./` indica che si tratta di un file e non di un pacchetto disponibile in un repository.

ANDANDO OLTRE La cache dei files .deb

APT conserva nella directory `/var/cache/apt/archives/` una copia di ogni file `.deb` scaricato. Nel caso di aggiornamenti frequenti, con diverse versioni di ciascun pacchetto, questa directory può occupare rapidamente molto spazio sul disco; pertanto dovreste ordinarla regolarmente onde evitare il loro accumulo. Esistono due comandi per fare ciò: `apt-get clean` che svuota completamente la directory; mentre `apt-get autoclean` rimuove solo i pacchetti non più scaricabili (perché non sono più presenti nel mirror Debian) e quindi chiaramente inutili (il parametro di configurazione `APT::Clean-Installed=false` consente di impedire la cancellazione dei files `.deb` attualmente installati ma non più scaricabili).

```
# apt-get -o APT::Clean-Installed=false autoclean
```

6.2.3 Aggiornamento del Sistema

Vi raccomandiamo di effettuare degli aggiornamenti regolari per ricevere gli aggiornamenti di sicurezza più recenti. Per effettuare l'aggiornamento dovrete utilizzare a vostra scelta uno dei seguenti comandi (ovviamente preceduto da `apt update`): `apt upgrade`, `apt-get upgrade` o `aptitude safe-upgrade`. Questi comandi verificano la presenza di aggiornamenti per i pacchetti installati, che possono essere eseguiti senza la necessità di rimuovere alcun pacchetto. In altre parole, il loro obiettivo è quello di garantire l'aggiornamento meno invasivo possibile. Per questa operazione, `apt-get` è un po' più esigente di `aptitude` o di `apt` in quanto si rifiuta di installare i pacchetti mai installati prima.

Si precisa nondimeno che `apt` seleziona solitamente i pacchetti scremandoli in base al numero di versione più recente (ed ignorerà per impostazione predefinita ed indipendentemente dal loro numero di versione i pacchetti dei repositories `Experimental` e `stable-backports`). Se avete inserito nel vostro `sources.list` le voci correlate alla `Testing` o all'`Unstable`, `apt upgrade` convertirà gran parte del vostro sistema `Stable` rispettivamente in `Testing` o in `Unstable`, con conseguenze indesiderate.

Per imporre ad `apt` di utilizzare una specifica distribuzione durante l'aggiornamento e la ricerca dei pacchetti, dovrete utilizzare l'opzione `-t` o `--target-release` seguita dal nome della distribuzione in questione (ad esempio: `apt -t stable upgrade`). Per imporre ad `apt` questa opzione sistematicamente, dovrete aggiungere nel file `/etc/apt/apt.conf.d/local` la seguente voce:

`APT::Default-Release "stable";`

Per aggiornamenti più imponenti, come l'aggiornamento di un'intera versione di Debian alla corrispondente successiva, dovrete utilizzare il comando `apt full-upgrade`. Con questo comando `apt` eseguirà l'aggiornamento anche se ci sono pacchetti obsoleti da rimuovere e nuove dipendenze da installare. Questo comando è valido anche per coloro i quali usufruiscono quotidianamente della versione `Unstable` di Debian e seguono i suoi sviluppi giorno dopo giorno. Inoltre, si reitera, questo comando è così semplice che non necessita di ulteriori delucidazioni: è questa caratteristica che ha reso famoso il tool APT.

A differenza di `apt` ed `aptitude`, `apt-get` non riconosce l'opzione `full-upgrade`. Di conseguenza, dovrete utilizzare la storico e ben noto comando `apt-get dist-upgrade` (aggiornamento della distribuzione), riconosciuto anche da `apt` e `aptitude` per agevolare gli utenti che sono ancora abituati ad usarlo.

I risultati delle suddette operazioni vengono registrati (in ingl. `logged`) in `/var/log/apt/history.log` e `/var/log/apt/term.log`, diversamente `dpkg` dispone di un log file particolareggiato denominato `/var/log/dpkg.log`.

6.2.4 Le Opzioni per la Configurazione

È possibile configurare anche altri aspetti di APT, oltre agli elementi di configurazione già menzionati, aggiungendo delle direttive ad un file nella directory `/etc/apt/apt.conf.d` o direttamente nel file `/etc/apt.conf`. Ad esempio, APT è in grado imporre a `dpkg` di ignorare gli errori dovuti ai conflitti dei files specificando `DPkg::options { "--force-overwrite"; }`. Se l'accesso al Web avviene solo tramite un proxy, dovete aggiungere una riga simile a `Acquire::http::proxy "http://ilvostroproxy:3128"`. Per un proxy FTP, dovete scrivere invece `Acquire::ftp::proxy "ftp://ilvostroproxy"`. Per scoprire le altre opzioni di configurazione consultate la pagina man di `apt.conf(5)`, con il comando `man apt.conf` (per maggiori dettagli sulle pagine man, andate a leggere il paragrafo 7.1.1, "Manual Pages" a pagina 148).

BASILARE

Le Directories che terminano in .d

Le directories dei suffissi `.d` vengono utilizzate molto spesso. Ciascuna di queste directories rappresenta un file di configurazione, che a sua volta è diviso in diversi files di configurazione. Pertanto, tutti i files contenuti in `/etc/apt/apt.conf.d` rappresentano le istruzioni per la configurazione di APT. APT computerà questi files di configurazione in ordine alfabetico, in modo da consentire ai files di configurazione che sono ultimi in ordine alfabetico di modificare gli elementi di configurazione definiti dai files di configurazione che li precedono nell'ordine alfabetico.

Questa conformazione conferisce flessibilità sia all'amministratore della macchina, sia ai manutentori del pacchetto. Infatti, l'amministratore della macchina può facilmente modificare la configurazione del software aggiungendo un file, realizzato ad hoc per una sua specifica esigenza, nella directory in questione, senza dover modificare un file esistente. Ed i manutentori dei pacchetti possono usufruire dello stesso approccio quando, per garantire la perfetta coesistenza con i loro software, devono modificare la configurazione di terzi. La Debian policy proibisce esplicitamente qualsiasi modifica ai files di configurazione appartenenti ad altri pacchetti - solo gli utenti sono autorizzati a modificarli. Difatti durante gli aggiornamenti `dpkg`, quando rileva una modifica di un file di configurazione, invita l'utente a sceglierne una versione così da salvarla. Qualsiasi modifica esterna dei files di configurazione comporta tale richiesta, richiesta che di certo inquieta gli amministratori sicuri di non aver alterato nulla. In assenza di una directory `.d`, è impossibile per un pacchetto esterno modificare le impostazioni di un programma senza modificare il file di configurazione. Pertanto, in questo caso, un pacchetto esterno deve invitare gli utenti ad effettuare loro stessi le modifiche, documentando le operazioni da eseguire nel file `/usr/share/doc/package/README.Debian`.

La directory `.d`, a seconda dei casi e dei programmi, può essere utilizzata direttamente o gestita indirettamente da uno script esterno che concatena tutti i files per realizzare un unico file di configurazione.

Inoltre, non potrete modificare direttamente il file di configurazione generato automaticamente dallo script, poiché tali modifiche andrebbero comunque perse con la successiva esecuzione dello script. La scelta di uno dei suddetti metodi (ossia l'utilizzo diretto della directory `.d` o la gestione indiretta attraverso un file generato dal contenuto della stessa directory `.d`) è solitamente dettata da vincoli di implementazione, ma entrambi i metodi garantiscono più profitti in termini di flessibilità della configurazione che costi in termini di piccole complicatezze.

Ad esempio Exim 4 per mail server, usufruisce della gestione indiretta della directory `.d`: i files di configurazione della cartella `/etc/exim4/conf.d/*` sono infatti concatenati in unico file nella cartella `/var/lib/exim4/config autogenerated` attraverso il comando `update-exim4.conf`

6.2.5 Gestione delle Package Priorities

Una delle questioni più importanti nella configurazione di APT è la gestione delle priorità delle diverse sorgenti dei pacchetti. Ad esempio, potrete desiderare di aggiungere ad una distribuzione uno o due pacchetti più recenti estraendoli dalla Testing, dalla Unstable o dalla Experimental. Per agevolare il suddetto intento è possibile assegnare una priorità a ciascun pacchetto disponibile (lo stesso pacchetto può ricevere diverse priorità, a seconda della sua versione o della sua distribuzione di appartenenza). Queste priorità influenzano il modo di procedere di APT: ovvero APT per ciascun pacchetto, selezionerà sistematicamente la versione con la massima priorità (a meno che questa versione non sia più obsoleta rispetto a quella installata e con priorità associata inferiore a 1000).

APT definisce diversi livelli predefiniti di priorità. Ad ogni versione del pacchetto già installata viene assegnato un livello di priorità pari a 100; ad una versione disinstallata del pacchetto viene assegnato un livello di priorità pari a 500, ma se tale versione disinstallata è inclusa nella target release, specificata dall'utente con l'opzione `-t` o attraverso la direttiva `APT::Default-Release`, le viene assegnato un livello di priorità predefinito pari a 990.

Potrete usufruire dei vantaggi delle priorità aggiungendo delle voci correlate in un file nella directory `/etc/apt/preferences.d` o direttamente nel file `/etc/apt/preferences`, voci che dovranno elencare il nome dei pacchetti a cui voi siete interessati, la loro versione, la loro origine e la loro nuova priorità.

Si reitera che APT si rifiuterà sempre di installare la versione più obsoleta di un pacchetto (con un numero di versione inferiore rispetto a quello della versione corrente), a meno che la priorità del pacchetto in questione non sia maggiore di 1000 (o sia esplicitamente richiesto dall'utente come già trattato nel paragrafo 6.2.2 Installazione e Rimozione a pagina 117). Di conseguenza APT prediligerà sempre installare la versione di un pacchetto con il livello di priorità più alto e che rispetta le sue istruzioni. Se due versioni hanno la stessa priorità, APT installerà la più recente (ovvero quella con il numero di versione più alto). Se due pacchetti hanno la stessa versione e lo stesso livello di priorità, ma differiscono nel loro contenuto, APT installerà la versione che non è ancora stata installata (questa regola predefinita è stata creata per rispondere concretamente ad un aggiornamento di un pacchetto larvato dell'aggiornamento incrementale del suo numero di revisione).

Di fatto, un pacchetto con un livello di priorità **inferiore a 0** non verrà mai installato. Un pacchetto con un livello di priorità **compreso tra 1 e 99** verrà installato solo se non è già installata un'altra versione dello stesso pacchetto. Con un livello di priorità **compreso tra 100 e 499**, il pacchetto verrà installato solo se non è già installata una versione più recente o non sia disponibile una versione più recente in un'altra distribuzione. Un pacchetto con un livello di priorità **tra 500 e 989** verrà installato solo in assenza di una versione più recente già installata o non se non è disponibile una versione più recente nella target release. Con un livello di priorità **compreso tra 990 e 1000** APT installerà il pacchetto, a condizione che non sia già installata una sua versione più recente. Un livello di priorità **superiore a 1000** imporrà ad APT l'installazione del pacchetto, anche se la versione del pacchetto è più obsoleta rispetto alla versione corrente (con conseguente downgrade).

Quando APT consulta il file `/etc/apt/preferences` e la directory `/etc/apt/preferences.d/`, prima considera le voci con maggiori dettagli (ovvero quelle riferite specificatamente al pacchetto in questione) e dopo le più generiche (includendo ad esempio tutti i pacchetti di una distribuzione). Se esistono più voci generiche riferite ad un pacchetto, viene utilizzata la prima voce corrispondente al pacchetto. I criteri della selezione stabiliscono che devono essere inclusi sia il nome del pacchetto,

sia la sorgente da cui proviene il pacchetto. Ogni sorgente di ogni pacchetto è identificata da una serie di informazioni contenute nel `Release` file, che APT scarica insieme ai files `Packages.gz`. Il `Release` file del pacchetto enuncia: l'origine (generalmente è "Debian" per i pacchetti scaricati dai mirror ufficiali, ma può anche trattarsi di un repository di terze parti identificato attraverso il nome di una persona od un'organizzazione); il nome della distribuzione (di solito, per le distribuzioni standard fornite da Debian, `Stable`, `Testing`, `Unstable` o `Experimental`) e la sua versione (ad esempio 10 per Debian `Buster`). Date un occhiata alla sua sintassi attraverso alcuni esempi che illustrano in termini pratici la procedura predefinita da seguire.

SPECIFIC CASE

La priorità
assegnata alla
`experimental`

Se avete inserito l'`Experimental` nel vostro file `sources.list`, i suoi pacchetti non verranno quasi mai installati nel vostro sistema, in quanto il loro livello di priorità APT, assegnato per impostazioni predefinite, è pari ad 1. Questo è certamente uno specific case (un caso particolare) designato per proteggere gli utenti ed impedire loro di installare erroneamente i pacchetti `Experimental`. Gli utenti consapevoli dei rischi possono raggiungere il suddetto ostacolo attraverso il comando `aptitude install nomedelpacchetto/experimental`. È possibile, ma non è consigliabile, usufruire dei pacchetti dell'`Experimental` e di altre distribuzioni assegnando loro un livello di priorità pari a 500 tramite una voce correlata nel file `/etc/apt/preferences`:

```
Package: *
Pin: release a=experimental
Pin-Priority: 500
```

Supponiamo che vogliate usare esclusivamente i pacchetti provenienti dalla versione `stable` di Debian. Gli stessi pacchetti potrebbero essere distribuiti anche da altre versioni, ma non verrebbero installati, eccetto su vostra esplicita richiesta. Per fare ciò vi basterà scrivere quanto segue nel file `/etc/apt/preferences`:

```
Package: *
Pin: release a=stable
Pin-Priority: 900
```

```
Package: *
Pin: release o=Debian
Pin-Priority: -10
```

`a=stable` specifica il nome della distribuzione da voi prescelta. `o=Debian` stringe il campo ai pacchetti la cui origine è "Debian".

[Il termine `pin` in inglese significa letteralmente "puntina", ma in questo contesto è più conveniente intenderlo come "etichettatura", perché il pin consente di etichettare un pacchetto per imporre ad APT da quale distribuzione deve scaricarlo].

Supponiamo ora che abbiate un server con installati diversi programmi che si basano sulla versione 5.24 di Perl e che vogliate assicurarvi che nessun aggiornamento installi un'altra versione. Per fare ciò vi basterà scrivere quanto segue:

```
Package: perl
Pin: version 5.24*
Pin-Priority: 1001
```

Per raggiungere una migliore consapevolezza dei meccanismi inerenti la priorità, la distribuzione o le proprietà dei repositories che potrete definire, non esitate ad eseguire i seguenti comandi: `apt-cache policy` per conoscere la priorità predefinita della sorgente dei pacchetti; `apt-cache policy package` per conoscere di un pacchetto sia la priorità di ciascuna versione disponibile, sia la priorità

della sorgente del pacchetto, come ampiamente trattato nella casella di testo apt-cache policy a pag. 128.

La documentazione di riferimento per il file /etc/apt/preferences e la directory /etc/apt/preferences.d/ è disponibile nella manual page `apt_preferences(5)`, accessibile attraverso il comando `man apt_preferences`

SUGGERIMENTO	Non esiste una sintassi ufficiale standard per l'inserimento dei commenti nel file /etc/apt/preferences, ma è possibile inserire il ruolo di ciascuna voce utilizzando uno o più fields "Explanation" (spiegazione) posizionati all'inizio di ogni blocco:
I commenti in /etc/apt/preferences/	<code>Explanation: The package server-xorg-video-intel provided</code> <code>Explanation: in experimental can be used safely</code> <code>Package: xserver-xorg-video-intel</code> <code>Pin: release a=experimental</code> <code>Pin-Priority: 500</code>

6.2.6 Come utilizzare più distribuzioni

apt è uno strumento così efficiente che incentiva l'installazione di pacchetti provenienti da altre distribuzioni. Ad esempio dopo aver installato una versione stable, potreste desiderare di testare un pacchetto software disponibile nella Testing o Unstable, ma senza far divergere del tutto il vostro sistema dal suo stato originale.

Anche se non sarete mai del tutto esenti da anomalie dovute alla presenza di pacchetti diversi provenienti da distribuzioni diverse, apt gestirà comunque la loro coesistenza nel modo più efficace possibile, limitandone i rischi. Per procedere in tal senso occorre innanzitutto specificare tutte le distribuzioni che volete utilizzare nel file /etc/apt/sources.list (diversi utenti spesso indicano tutte le tre distribuzioni, pur essendo la Unstable riservata agli utenti esperti) e specificare la distribuzione di riferimento con il parametro `APT::Default-Release` (andate a vedere il paragrafo 6.2.3 "Aggiornamento del Sistema" a pagina 120).

Supponiamo che la Stable sia la vostra distribuzione di riferimento, ma che abbiate anche inserito la Testing e la Unstable nel vostro file sources.list. In questo caso, è possibile utilizzare `apt install nome_delpacchetto/testing` per installare un pacchetto dalla Testing. Se l'installazione non dovesse andare a buon fine a causa di alcune dipendenze non soddisfatte, potrete autorizzare la loro risoluzione attraverso la loro installazione aggiungendo il parametro `-t testing`. Potrete utilizzare questo metodo ovviamente anche con la Unstable.

Gli aggiornamenti ("upgrade" e "full-upgrade"), nei suddetti casi, vengono comunque eseguiti senza divergere dalla Stable, ad eccezione dei pacchetti già aggiornati da un'altra distribuzione: questi infatti verranno aggiornati in base agli ultimi aggiornamenti disponibili della distribuzione da cui derivano. In seguito tratteremo meglio tale procedura grazie alle priorità predefinite utilizzate da APT. Non esitate a utilizzare il comando `apt-cache policy` per verificare le priorità assegnate (per maggiori dettagli andate a leggere la casella di testo "apt-cache policy" a pagina 128).

Ogni effetto o conseguenza sopra delineata è dovuta al fatto che APT considera sempre solo i pacchetti con una versione maggiore o uguale alla versione installata (presumendo che in /etc/apt/preferences non siano state impostate delle priorità di qualche pacchetto maggiori di 1000).

Poniamo il caso che abbiate installato la versione 1 di un pacchetto dalla Stable e che le sue relative versioni 2 e 3 si trovino rispettivamente nella Testing e nella Unstable. La versione installata avrà una priorità di 100, ma la versione disponibile nella Stable (ossia la stessa identica) avrà una priorità di 990 (in quanto fa parte della target release). I pacchetti della Testing e della Unstable avranno una priorità di 500 (la priorità predefinita di una versione non installata). Il vincitore, ovvero la versione del pacchetto selezionata da APT, sarà quindi la versione 1 essendole stata assegnata una priorità di 990. Di conseguenza il pacchetto "rimarrà della versione Stable".

Prendiamo come esempio il caso di un altro pacchetto, di cui avete installato la versione 2 dalla Testing la versione 1 è disponibile nella stable e la versione 3 nella Unstable. La versione 1 (con una priorità 990 - e quindi inferiore a 1000) sarà ignorata perché è più vecchia della versione installata. Rimangono quindi le versioni 2 e 3, entrambe con priorità 500. Di fronte a queste alternative, APT sceglierà la versione più recente, ossia quella della distribuzione Unstable.

Se non desiderate che un pacchetto installato dalla Testing si aggiorni alla sua versione Unstable, è necessario che associate alla sua corrispettiva versione Unstable una priorità inferiore a 500 (ad esempio, 490). Per fare ciò dovete modificare nel seguente modo il file /etc/apt/preferences:

```
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

6.2.7 Come resocontare i pacchetti installati automaticamente

Una delle funzionalità indispensabili di apt è il monitoraggio dei pacchetti installati attraverso le dipendenze. Questi pacchetti sono chiamati "automatici" e spesso includono librerie.

Quando i pacchetti vengono rimossi, i sistemi di gestione dei pacchetti con queste informazioni possono computare un elenco di pacchetti automatici che non sono più necessari (in quanto non esistono più i pacchetti installati "manualmente" che dipendono da loro). apt autoremove e apt-get autoremove sono i comandi per rimuovere questi pacchetti. aptitude non dispone di un comando equivalente, poiché elimina automaticamente questi pacchetti non appena vengono identificati come non necessari. In ogni caso, gli strumenti menzionati elencano attraverso un messaggio chiaro i pacchetti influenzati dalle loro operazioni.

È buona idea prendere l'abitudine di contrassegnare come pacchetti automatici i pacchetti di cui non necessitate direttamente, in modo che vengano automaticamente eliminati quando non sono più necessari. Il comando apt-mark auto nomedelpacchetto contrassegna il pacchetto indicato come pacchetto automatico mentre apt-mark manual nomedelpacchetto fa il contrario. aptitude markauto e aptitude unmarkauto funzionano allo stesso modo, ma offrono più funzionalità per contrassegnare più pacchetti contemporaneamente (andate a leggere al riguardo il paragrafo 6.5.1, "aptitude" a pagina 129). L'interfaccia interattiva di aptitude facilita anche la revisione degli "automatic flag" su molti pacchetti.

Potreste desiderare sapere perché sul sistema è presente un pacchetto installato automaticamente. Per ottenere queste informazioni direttamente dalla riga di comando, potrete utilizzare il comando aptitude why nomedelpacchetto (apt e apt-get non dispongono di questa funzionalità):

```
$ aptitude why python-debian
i aptitude Suggests apt-xapian-index
p apt-xapian-index Depends python-debian (>= 0.1.14)
```

[In informatica, un "reference" (riferimento) è un valore che consente ad un programma di accedere indirettamente ad un dato particolare, come il valore di una variabile o un record, nella memoria del computer o in qualche altro dispositivo di archiviazione. Invece "l'indirezione" (detta anche "riferimento indiretto") è la tecnica che consente di indicare un oggetto o un valore mediante un suo riferimento invece che direttamente. Il termine viene utilizzato frequentemente nei linguaggi di programmazione per indicare l'uso dei puntatori. I puntatori sono tipi di dati che rappresentano la posizione (usando indirizzi di memoria) di elementi del programma come variabili, oggetti, strutture dati, sottoprogrammi. I files che contengono l'estensione .desktop appartengono ai sistemi operativi basati su Linux e Unix. Questi files consentono ai sistemi Linux e Unix provvisti di una GUI di sapere come avviare l'applicazione a loro associate.]

ALTERNATIVA deborphan e debfoster

In passato quando apt, apt-get e aptitude non erano ancora in grado di effettuare il monitoraggio automatico dei pacchetti, esistevano due utilità per elaborare un elenco di pacchetti non necessari: deborphan e debfoster. Entrambi questi strumenti sono ancora utili.

deborphan si limita per impostazione predefinita a scansionare semplicemente le sezioni `libs` e `oldlibs` (se non ci sono istruzioni aggiuntive) per cercare i pacchetti attualmente installati da cui non dipende nessun altro pacchetto installato. Questo elenco può quindi essere utilizzato come punto di partenza per la rimozione di pacchetti non necessari.

debffoster ha un approccio più avanzato, simile a quello di APT: mantiene un elenco dei pacchetti installati esplicitamente e memorizza quali pacchetti sono di fatto effettivamente necessari durante ogni invocazione.

Se sul sistema compaiono nuovi pacchetti e se debffoster non li riconosce come “explicitly required” (pacchetti necessari), verranno visualizzati sullo schermo, insieme ad un elenco delle loro dipendenze. Il programma vi consentirà quindi di scegliere di: eliminare il pacchetto (e di eliminare anche i pacchetti che possibilmente dipendono da questi), contrassegnarlo esplicitamente come necessario (`explicitly required`) o ignorarlo temporaneamente.

6.3 Il comando apt-cache

Il comando apt-cache consente di consultare una determinata quantità di informazioni memorizzate nel database interno di APT. Tutta questa informazione — che costituisce una specie di cache — viene raccolta dalle diverse sorgenti elencate nel file `sources.list` durante l'operazione di `apt update`.

DIZIONARIO Cache

Una cache è un sistema di archiviazione temporanea utilizzato per accelerare l'accesso frequente ai dati quando il normale metodo di accesso richiede maggiori costi (in termini di prestazioni). Questo concetto si applica in moltissimi contesti e con differenti rilevanze, dal core dei microprocessori ai sistemi di archiviazione di fascia alta.

Nel caso di APT i reference Packages files sono collocati nei mirrors Debian. Tuttavia, non sarebbe molto efficiente doversi collegare alla rete per ogni ricerca che si desidera fare nel database dei pacchetti disponibili. APT pertanto memorizza una copia di questi files (in `/var/lib/apt/lists/`) in modo che le ricerche possano essere eseguite localmente (ossia attraverso questi files locali). Allo stesso modo `/var/cache/apt/archives/` contiene una cache dei pacchetti già scaricati, che vi eviterà di scaricarli nuovamente se desiderate reinstallarli dopo averli eliminati.

Dovrete eseguire regolarmente il comando `apt update` per aggiornare la cache. Altrimenti non potrete beneficiare degli ultimi aggiornamenti distribuiti dai mirrors Debian fra i risultati di ricerca dei pacchetti.

Il programma apt-cache consente di cercare i pacchetti utilizzando delle parole chiave con `apt-cache search keyword`. Potete anche consultare gli headers dei pacchetti (e delle loro diverse versioni disponibili) con `apt-cache show package`. Questo comando vi fornirà la descrizione del pacchetto, le sue dipendenze, il nome del suo manutentore, ecc. Si precisa che i comandi APT (`apt search` e `apt show`) funzionano in modo simile ai loro corrispettivi APTITUDE (`aptitude search` e `aptitude show`).

ALTERNATIVA axi-cache

`apt-cache` serch è uno strumento rudimentale, di ricerca come grep ma dedicato alle descrizioni dei pacchetti. Restituisce spesso troppi risultati se non addirittura nessuno quando vengono fornite troppe parole chiave.

Al contrario, `axi-cache` serch term fornisce risultati migliori, ordinati per attinenza. Questo strumento utilizza il motore di ricerca Xapian; fa parte del pacchetto `apt-xapian-index`, che indicizza tutte le informazioni sul pacchetto (ma non solo: indicizza anche i files .desktop di tutti i pacchetti debian). È in grado di riconoscere e gestire le tags (per maggiori dettagli leggete la casella di testo "Il field Tag" a pagina 86) ed il suo tempo di risposta è spesso di pochi millisecondi.

```
$ axi-cache search package use::searching

100 results found.
Results 1-20:
100% packagesearch - GUI for searching packages and viewing
-> package information
99% apt-utils - package management related utility programs
98% whohas - query multiple distributions' package archives
98% dpkg-awk - Gawk script to parse /var/lib/dpkg/{status,
-> available} and Packages
97% apt-file - search for files within Debian packages (
-> command-line interface)
[...]
90% wajig - unified package management front-end for Debian
More terms: debtags debian paket dpkg search pakete tools
More tags: role::program interface::commandline works-with
-> ::software:package suite::debian admin::package-
-> management scope::utility network::client
'axi-cache more' will give more results
```

Alcune funzionalità sono utilizzate raramente. Ad esempio, `apt-cache policy` consente di consultare le priorità delle diverse sorgenti dei diversi pacchetti nonché le priorità delle sorgenti dei singoli pacchetti. Un altro esempio è `apt-cache dumpavail` che mostra gli headers di tutte le versioni disponibili di tutti i pacchetti. `apt-cache pkgnames` visualizza un elenco di tutti i pacchetti esistenti nella cache.

SUGGERIMENTO apt-cache policy

Utilizzate il comando `apt-cache policy` per conoscere la priorità predefinita della sorgente e della distribuzione dei pacchetti come spiegato nel paragrafo 6.2.5 Gestione delle Package Priorities a pag. 122. Potrete inoltre eseguire `apt-cache policy` per consultare le priorità di tutte le versioni e le sorgenti disponibili dei pacchetti. In base al file `sources.list` utilizzato nell'esempio 6.2 Il file `/etc/apt/sources.list` della versione Stable di Debian a pag. 110 ed alla direttiva APT::Default-Release configurata con "buster", avrete il seguente output:

```
$ apt-cache policy
Package files:
100 /var/lib/dpkg/status
release a=now
100 https://deb.debian.org/debian buster-backports/contrib amd64 Packages
release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian
Backports,c=contrib, b=amd64 origin deb.debian.org
100 https://deb.debian.org/debian buster-backports/main amd64 Packages
release o=Debian Backports,a=buster-backports,n=buster-backports,l=Debian
Backports,c=main,b=amd64 origin deb.debian.org
990 https://deb.debian.org/debian buster/non-free amd64 Packages
release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=non-free,b=amd64 origin
deb.debian.org
990 https://deb.debian.org/debian buster/contrib amd64 Packages
release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=contrib,b=amd64 origin
deb.debian.org
990 https://deb.debian.org/debian buster/main amd64 Packages
release v=10.0,o=Debian,a=stable,n=buster,l=Debian,c=main,b=amd64 origin
deb.debian.org
990 http://security.debian.org buster/updates/main amd64 Packages
release v=10.0,o=Debian,a=stable,n=buster,l=Debian-Security,c=main,b=amd64 origin
security.debian.org
```

Attraverso `apt-cache policy` potrete consultare le priorità di tutte le versioni e le sorgenti disponibili di un dato pacchetto.

```
$ apt-cache policy iptables
iptables:
Installed: 1.8.2-4
Candidate: 1.8.2-4
Version table:
1.8.3-2-bpo10+1 100
100 https://deb.debian.org/debian buster-backports/main amd64
Packages
*** 1.8.2-4 990
990 https://deb.debian.org/debian buster/main amd64 Packages
100 /var/lib/dpkg/status
Dall'esempio soprastante pare evidente che nonostante sia disponibile una versione più aggiornata del pacchetto di iptables attraverso il repository buster-backports APT non la installerà automaticamente per rispettare le priorità. Di conseguenza per installare la versione più aggiornata dovete eseguire il comando apt install iptables/buster-backports oppure assegnare una priorità più alta a /etc/apt/preferences.d/iptables:
```

Package: iptables
Pin: release o=Debian Backports, a=buster-backports
Pin-Priority: 1001

6.4. Il comando apt-file

Potrebbe capitarsi di cercare informazioni su un file o su un comando e di non conoscere il pacchetto che le include. Fortunatamente i repositories Debian distribuiscono non solo le informazioni riguardo a tutti i pacchetti binari supportati, ma anche su tutti i files che gli stessi pacchetti supportati dispongono. Tali informazioni vengono memorizzate nei files denominati `Contents-arch.gz` e `Contents-udeb-arch.gz`. Occorre precisare che le suddette informazioni non vengono

scaricate automaticamente da APT. Dovrete pertanto utilizzare il comando `apt-file update` (incluso nel pacchetto omonimo) per recuperare i contenuti da tutte le sorgenti dei pacchetti definite in `/etc/apt/sources.list`. Se desiderate aggiornare il database settimanalmente, dovete aggiungere la seguente voce a `/etc/crontab`:

```
@weekly root test -x /usr/bin/apt-file && /usr/bin/apt-file update >> /dev/null 2>&1
```

Dopo che avrete aggiornato il database, il comando `apt-file search pattern` elencherà tutti i pacchetti con un filename o un path come quello da voi definito attraverso il pattern (filtro).

```
$ apt-file search bin/axi-cache  
apt-xapian-index: /usr/bin/axi-cache
```

Diversamente il comando `apt-file list package` elencherà tutti i files distribuiti attraverso il pacchetto.

SUGGERIMENTO
Come ottenere
l'elenco dei
contenuti di un
pacchetto e
trovarne tutti i files

Anche il comando `dpkg -L package` elenca tutti i files come `apt-file list`, ma `dpkg`, diversamente da `apt-file`, elenca tutti i files del pacchetto richiesto. Inoltre potrete cercare il pacchetto a cui appartiene un file locale, utilizzando il comando `dpkg -S file` (per maggiori informazioni andate a leggere il paragrafo 5.4.3 Querying (Richieste) per il Database di `dpkg` ed ispezione dei Files .deb a pagina 97). Invece per ottenere l'elenco di tutti i files locali che non appartengono a nessun pacchetto installato, dovete fare riferimento a `cruft` o al pacchetto `cruft-ng`.

6.5 Frontends: aptitude, synaptic

APT è un programma C++, di cui la cui maggior parte del codice si trova nella libreria condivisa `libapt-pkg`. L'utilizzo di una libreria condivisa rende relativamente semplice la creazione di un'interfaccia utente (un "front-end"), poiché è sufficiente richiamare il codice contenuto dalla libreria. `apt-get` era stato sviluppato originariamente per testare `libapt-pkg` come front-end, ma il suo successo ha contribuito ad offuscare il compito a cui era stato inizialmente destinato.

6.5.1 aptitude

`aptitude` è un programma interattivo che può essere utilizzato in modalità consolle (semi grafica). Attraverso `aptitude` potrete sfogliare l'elenco dei pacchetti installati e disponibili, consultare tutte le informazioni disponibili e contrassegnare i pacchetti per eseguire una loro installazione o una loro rimozione. Questo programma è stato progettato specificatamente per essere usato dagli amministratori, difatti i suoi processi predefiniti sono più raffinati di `apt-get` e la sua interfaccia molto più facile da comprendere.

```

Actions Undo Package Resolver Search Options Views Help
C-T: Menu ?: Help q: Quit u: Update g: Preview/Download/Install/Remove Pkgs
aptitude 0.8.12 @ vagon
i A apg 2.2.3.dfsg.1-5 2.2.3.dfsg.1-
i A apparmor 2.13.3-7 2.13.3-7
i A appstream 0.12.9-1 0.12.9-1
i A apt 1.8.4 1.8.4
i A apt-file 3.2.2 3.2.2
i A apt-listbugs 0.1.30 0.1.30
i A apt-show-versions 0.22.11 0.22.11
i A apt-utils 1.8.4 1.8.4
i A apt-xapian-index 0.50 0.50
i A aptitude 0.8.12-1 0.8.12-1
i A aptitude-common 0.8.12-1 0.8.12-1
A arch-test 0.16-2 0.16-2
A at 3.1.23-1+b1 3.1.23-1+b1
base-files 11 11
terminal-based package manager
aptitude is a package manager with a number of useful features, including: a mutt-like syntax for
atching packages in a flexible manner, dselect-like persistence of user actions, the ability to
etrieve and display the Debian changelog of most packages, and a command-line mode similar to that
f apt-get.

aptitude is also Y2K-compliant, non-fattening, naturally cleansing, and housebroken.
omepage: https://wiki.debian.org/Aptitude
ags: admin::configuring, admin::package-management, implemented-in::c++, interface::commandline,
interface::text-mode, role::program, scope::application, suite::debian, uikit::ncurses,
use::browsing, use::configuring, use::downloading, use::searching,
works-with::software:package

```

Figura 6.1 aptitude package manager

L'interfaccia di aptitude consente la visualizzazione dell'elenco suddiviso dei pacchetti in base al loro status corrente (installato, non installato o installato ma non disponibile sui mirrors – altre sezioni mostrano le attività, i virtual packages ed i pacchetti comparsi recentemente sui mirrors). Per facilitare la consultazione tematica, sono disponibili altre modalità di visualizzazione dei contenuti. In ogni caso, aptitude è in grado di visualizzare in una schermata un elenco ottenuto dalla combinazione di categorie e pacchetti. Le categorie sono organizzate attraverso una struttura ad albero, di cui è possibile espandere o chiudere "i rami" rispettivamente con i tasti Enter, [e]. Il tasto + consente di contrassegnare un pacchetto per poterne eseguire l'installazione, il tasto - consente di contrassegnare un pacchetto per poterne eseguire la rimozione ed il tasto _ per "purgarlo" definitivamente (si precisa che queste scorciatoie possono essere applicate anche sulle categorie e, nel qual caso, comporteranno l'applicazione delle correlate azioni su tutti i pacchetti della categoria contrassegnata). Il tasto u aggiorna gli elenchi dei pacchetti disponibili e la combinazione dei tasti Shift ed u (ossia Shift+u) prepara un aggiornamento globale del sistema. Il tasto g consente la visualizzazione di un riepilogo sommario delle richieste di modifica (tale modifiche saranno avviate in risposta ad una seconda digitazione del tasto g) mentre il tasto q consente di uscire dalla visualizzazione in corso. Di conseguenza se state visualizzando la schermata iniziale, la digitazione del tasto q di fatto chiuderà aptitude.

DOCUMENTAZIONE aptitude

Questo paragrafo non esplica tutti i dettagli sull'utilizzo di aptitude, ma si limita a fornire delle istruzioni essenziali. Su aptitude è comunque disponibile una correlata documentazione, quindi vi invitiamo a consultarne il manuale completo disponibile con il pacchetto aptitude-doc-en (/usr/share/doc/aptitude/html/en/index.html) o <https://www.debian.org/doc/manual/aptitude/>

Per cercare un pacchetto attraverso un pattern (filtro) di ricerca utilizzate il tasto /. Come filtro si può utilizzare il nome del pacchetto, oppure il filtro può essere riferito alla sua descrizione (se preceduto da ~d) o alla sua sezione (se preceduto ~s) o ad altre caratteristiche descritte nella documentazione. I summenzionati patterns possono essere utilizzati per filtrare l'elenco dei pacchetti visualizzati e tale funzionalità è accessibile attraverso la digitazione del tasto l (inteso come limit) seguito dal pattern.

aptitude facilita notevolmente la gestione degli "automatic flag" dei pacchetti Debian (a tal proposito consultate il paragrafo 6.2.7, "Come resocontare i pacchetti installati automaticamente" a pagina 125). Vi permetterà di consultare l'elenco dei pacchetti installati, contrassegnarli come

automatici con la combinazione dei tasti Shift e m (ossia Shift+m) e di rimuovere la contrassegnazione con il tasto m. Gli "automatic packages" sono segnalati all'utente attraverso una "A" nell'elenco dei pacchetti.

Questa funzionalità consente inoltre di visualizzare semplicemente i pacchetti in uso su una macchina, senza elencare tutte le librerie e le dipendenze di cui potreste non essere interessati.

Il correlato filtro di ricerca, che può essere utilizzato in combinazione con l (per attivare la modalità filtro), è ~i!~M. Tale filtro consente di visualizzare solo i pacchetti installati (~i) non contrassegnati come automatici (~M).

STRUMENTI TOOLS

Come utilizzare aptitude da riga di comando

La maggior parte delle funzionalità di aptitude sono accessibili sia attraverso l'interfaccia interattiva, sia da riga di comando. Gli utenti che utilizzano regolarmente apt-get e apt-cache avranno familiarità con le funzionalità di aptitude da riga di comando.

Anche le funzionalità avanzate di aptitude sono accessibili da riga di comando. Potrete quindi usare gli stessi patterns della versione interattiva per la ricerca dei pacchetti. Per esempio, qualora vogliate sistemare l'elenco dei pacchetti "installati manualmente", a condizione che siate certi che nessun programma installato localmente richieda specifiche librerie o moduli Perl, potrete contrassegnare i pacchetti corrispondenti come automatici con un singolo comando:

```
# aptitude markauto '~slibs|~perl'
```

Quanto appena espresso vi consentirà di capacitarvi dell'efficacia del sistema di ricerca di aptitude attraverso i patterns, che a loro volta vi metteranno in grado di selezionare simultaneamente tutti i pacchetti nelle sezioni libs e perl.

Prestate attenzione che i pacchetti contrassegnati come automatici non abbiano altri pacchetti che dipendono da loro, in quanto altrimenti verranno eliminati immediatamente (dopo una richiesta di conferma).

6.5.1.1 Gestire i Recommendations, i Suggestions e le Tasks

Un altro aspetto di aptitude meritevole di essere menzionato è che rispetta le raccomandazioni (ed il conseguente rapporto tra i pacchetti) conferendo allo stesso tempo agli utenti l'opportunità di valutare caso per caso di non installarli. Per esempio, il pacchetto gnome raccomanda (anche) transmission-gtk. Quindi se selezionerete gnome per l'installazione, sarà selezionato anche transmission-gtk (e sarà contrassegnato come automatico se non è già installato sul sistema). Attraverso il tasto g potrete accertarvene: transmission-gtk apparirà nella schermata di riepilogo delle azioni in sospeso nell'elenco dei pacchetti installati automaticamente per soddisfare le dipendenze. Tuttavia potrete decidere di non installarlo, deselectandolo prima di convalidare le operazioni.

Si precisa però che la funzionalità di monitoraggio delle raccomandazioni non viene eseguita durante gli aggiornamenti. Pertanto, se una nuova versione di gnome raccomanda un pacchetto che non aveva raccomandato in precedenza, non verrà contrassegnato per l'installazione. Tuttavia, verrà inserito nell'elenco della schermata di aggiornamento, in modo che l'amministratore possa installarlo comunque.

Anche i suggerimenti (ed il conseguente rapporto tra i pacchetti) sono tenuti in considerazione, ma con un metodo adeguato al loro status. Per esempio, gnome (al momento della stesura di questo testo) suggerisce empathy, quindi quest'ultimo sarà elencato nella schermata di riepilogo delle azioni in sospeso (nella sezione dei pacchetti suggeriti da altri pacchetti). In questo modo sarà visibile e l'amministratore potrà decidere se tenere conto del suggerimento. Ma essendo un mero suggerimento e non una dipendenza o una raccomandazione, il pacchetto non sarà selezionato automaticamente – la sua selezione richiederà un intervento manuale dell'utente (il pacchetto non verrà quindi contrassegnato come automatico).

Per le soprammenzionate ragioni, si ricorda anche che aptitude ha reso proprio in modo efficiente il concetto di task (attività). Dal momento che le attività sono visualizzate sotto forma di categorie nell'elenco dei pacchetti (e nelle correlate schermate), avrete l'opportunità di selezionare un'attività completa da installare o rimuovere, oppure di consultare l'elenco dei pacchetti inclusi nell'attività per personalizzarne la selezione e realizzarne un subset (sottoinsieme) meno esteso.

6.5.1.2 I migliori Solver Algorithms

Infine, per concludere questo paragrafo, si precisa che aptitude, per trovare la soluzione a dei contesti non facili, include degli algoritmi più complessi di apt-get. Per esempio, quando viene richiesta una configurazione di azioni che, se combinate, possono rendere un sistema incoerente, aptitude valuta le potenziali conseguenze e le espone in ordine di rilevanza decrescente. Nonostante ciò, questi algoritmi non sono infallibili. Fortunatamente esiste sempre la possibilità di selezionare manualmente le azioni da eseguire. Ad esempio quando le azioni selezionate comportano delle contraddizioni, la parte superiore dello schermo indica il numero di pacchetti "broken" [rotti - non funzionanti] (e potrete giungere a questi pacchetti attraverso il tasto b). Pertanto potrete voi stessi realizzare manualmente una soluzione ai problemi rilevati. Nello specifico potrete accedere alle diverse versioni disponibili di un pacchetto, semplicemente selezionando un pacchetto con Enter. Se tale selezione ovvero la scelta di una particolare versione rispetto ad un'altra risolve il problema, non esitate ad utilizzare questa funzionalità. Quando il numero di pacchetti rotti scende a zero, recatevi senza remore nella schermata di riepilogo delle azioni in sospeso per un ultimo controllo prima della loro esecuzione.

NOTA	
Il log di aptitude	Come dpkg, aptitude tiene traccia delle azioni eseguite nel suo logfile (/var/log/aptitude). Tuttavia, dal momento che i due summenzionati comandi svolgono le loro funzioni a livelli differenti, non potrete trovare le stesse informazioni nei rispettivi logfiles. Anche se dpkg è in grado registrare [in inglese "to log"] passo dopo passo tutte le operazioni eseguite su ogni singolo pacchetto, aptitude offre una panoramica delle operazioni di livello superiore come ad esempio un aggiornamento dell'intero sistema. Fate attenzione: il registro di aptitude contiene solo il riepilogo delle operazioni avviate dallo stesso aptitude. Se utilizzate occasionalmente altri front-end (come lo stesso dpkg), potrete avere dal registro di aptitude solo una visione parziale delle operazioni e non potrete ricostruire una cronologia attendibile del sistema.

6.5.2 synaptic

synaptic è un sistema di gestione dei pacchetti Debian in modalità grafica che si contraddistingue per la chiara ed efficiente interfaccia grafica basata su GTK +/GNOME. I suoi numerosi filtri pronti all'uso consentono di visualizzare rapidamente i nuovi pacchetti disponibili, i pacchetti installati, i pacchetti che possono essere aggiornati, i pacchetti obsoleti, ecc. Attraverso questi elenchi, e la loro mera consultazione, potrete selezionare le operazioni da eseguire (install, upgrade, remove, purge). Queste operazioni non sono eseguite istantaneamente, ma vengono incluse in un elenco. Poi sarà sufficiente un semplice clic su un apposito bottone per convalidarle ed eseguirle tutte in una volta.

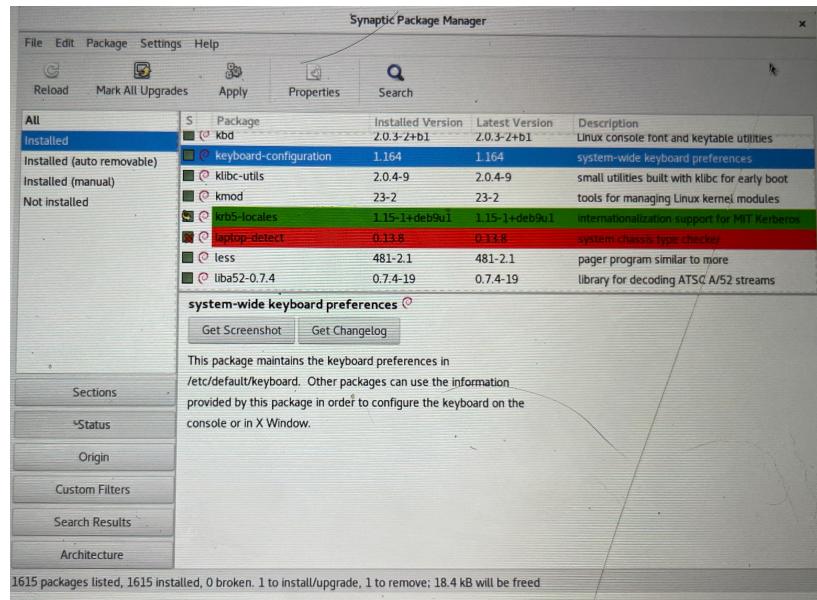


Figura 6.2 synaptic package manager

6.6 Verifica dell'autenticità del pacchetto

Gli amministratori della Falcot Corp attribuiscono molta importanza alla sicurezza. Pertanto desiderano avere la certezza di installare solo pacchetti garantiti da Debian che non siano stati in alcun modo manomessi. Difatti un cracker [un pirata informatico – da distinguersi dall'hacker, che è un esperto informatico che sfrutta le proprie competenze per migliorare i sistemi programmabili ed estenderne l'utilizzo] potrebbe aver aggiunto del codice malevolo ad un pacchetto non legittimo. Quest'ultimo alla stessa stregua di un pacchetto legittimo, se installato, potrà svolgere qualsiasi azione malevola per cui è stato designato dal cracker, come ad esempio rubare le password o le informazioni confidenziali. Per scongiurare tale rischio, Debian fornisce un *tamper-proof seal* [tradotto letteralmente dall'inglese "sigillo o marchio antimanomissione"] per garantire che il pacchetto, che si intende installare, provenga dal suo manutentore e che non sia stato modificato da terzi.

Questo sigillo è ottenuto dalla combinazione di funzioni crittografiche di hashes ed una firma; troverete maggiori dettagli al riguardo in `apt-secure(8)`. A partire da Debian 10 Buster il file firmato è il file `InRelease`, reso disponibile dai mirrors Debian. Esiste anche un file `legacy` denominato `Release`. Entrambi i summenzionati files contengono una lista di files `Packages` (lista che include il loro formato di compressione, `Packages.gz` e `Packages.xz`, ed il loro numero incrementale di versione) ed i corrispettivi hashes [somme di controllo o checksums] in SHA256, che garantiscono che i files sono esenti da manomissione. Questi files `Packages` a loro volta contengono un elenco di pacchetti Debian con i loro hashes, per assicurare che anche il loro contenuto non sia stato alterato, e sono disponibili attraverso i mirrors. La differenza tra il file `InRelease` ed il file `Release` è che il primo viene crittografato in-line [ovvero include direttamente `proper` e `firma`], mentre il secondo viene accompagnato da un file `Release.gpg` in cui la firma è inclusa in un format dedicato.

NOTA Il futuro del file <code>Release</code> e di <code>Release.gpg</code>	Probabilmente a partire dal rilascio di Debian 11 Bullseye APT rimuoverà il supporto ai files <code>legacy</code> <code>Release</code> e <code>Release.gpg</code> , utilizzati a partire da APT 0.6 che a sua volta ha introdotto un archivio di autenticazione.
--	--

APT necessita di un set di chiavi pubbliche GnuPG fidate per verificare le firme in `InRelease` e `Release`. I files `.gpg` sono accessibili attraverso i mirrors. Inoltre APT dispone di tali files attraverso `/etc/apt/trusted.gpg.d/` e dal keyring `/etc/apt/trusted.gpg` (le `trusted keys`, traducibili letteralmente dall'inglese come "chiavi attendibili", sono gestite attraverso il comando `apt-key`). [Questo programma gestisce un `keyring` ["portachiavi"] di chiavi pubbliche GnuPG, che sono utilizzate per verificare le firme dei files `.gpg` disponibili tramite i mirrors Debian. Inoltre sempre attraverso il summenzionato programma è possibile aggiungere delle nuove chiavi manualmente (quando sono anche necessari i mirrors non ufficiali Debian). Anche se, solitamente, sono indispensabili solo le chiavi dei mirrors ufficiali di Debian.] Queste chiavi sono mantenute aggiornate dal pacchetto `debian-archive-keyring` (che installa i corrispettivi keyrings in `/etc/apt/trusted.gpg.d/`). Nonostante ciò, anche la prima installazione di questo particolare pacchetto richiede cautela in quanto, pur essendo firmato come tutti gli altri pacchetti, la sua firma non può essere verificata al di fuori di esso. Gli amministratori prudenti pertanto dovrebbero verificare le fingerprints ["impronte digitali"] delle chiavi importate, prima di porre la loro fiducia su di esse ed installare i corrispondenti nuovi pacchetti:

```
# apt-key fingerprint
/etc/apt/trusted.gpg.d/debian-archive-buster-automatic.gpg
-----
pub rsa4096 2019-04-14 [SC] [expires: 2027-04-12]
  80D1 5823 B7FD 1561 F9F7 BCDD DC30 D7C2 3CBB ABEE
uid [ unknown] Debian Archive Automatic Signing Key (10/buster) <ftpmaster@debian.org>
sub rsa4096 2019-04-14 [S] [expires: 2027-04-12]
/etc/apt/trusted.gpg.d/debian-archive-buster-security-automatic.gpg
-----
pub rsa4096 2019-04-14 [SC] [expires: 2027-04-12]
  5E61 B217 265D A980 7A23 C5FF 4DFA B270 CAA9 6DFA
uid [ unknown] Debian Security Archive Automatic Signing Key (10/buster) <ftpmaster@debian.org>
sub rsa4096 2019-04-14 [S] [expires: 2027-04-12]
/etc/apt/trusted.gpg.d/debian-archive-buster-stable.gpg
-----
pub rsa4096 2019-02-05 [SC] [expires: 2027-02-03]
  6D33 866E DD8F FA41 C014 3AED DCC9 EFBF 77E1 1517
uid [ unknown] Debian Stable Release Key (10/buster) <debian-release@lists.debian.org>
/etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg
-----
pub rsa4096 2014-11-21 [SC] [expires: 2022-11-19]
  126C 0D24 BD8A 2942 CC7D F8AC 7638 D044 2B90 D010
uid [ unknown] Debian Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>
/etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg
-----
pub rsa4096 2014-11-21 [SC] [expires: 2022-11-19]
  D211 6914 1CEC D440 F2EB 8DDA 9D6D 8F6B C857 C906
uid [ unknown] Debian Security Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>
/etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg
-----
pub rsa4096 2013-08-17 [SC] [expires: 2021-08-15]
  75DD C3C4 A499 F1A1 8CB5 F3C8 CBF8 D6FD 518E 17E1
uid [ unknown] Jessie Stable Release Key <debian-release@lists.debian.org>
/etc/apt/trusted.gpg.d/debian-archive-stretch-automatic.gpg
-----
pub rsa4096 2017-05-22 [SC] [expires: 2025-05-20]
  E1CF 20DD FFE4 B89E 8026 58F1 E0B1 1894 F66A EC98
uid [ unknown] Debian Archive Automatic Signing Key (9/stretch) <ftpmaster@debian.org>
sub rsa4096 2017-05-22 [S] [expires: 2025-05-20]
/etc/apt/trusted.gpg.d/debian-archive-stretch-security-automatic.gpg
-----
pub rsa4096 2017-05-22 [SC] [expires: 2025-05-20]
  6ED6 F5CB 5FA6 FB2F 460A E88E EDA0 D238 8AE2 2BA9
uid [ unknown] Debian Security Archive Automatic Signing Key (9/stretch) <ftpmaster@debian.org>
sub rsa4096 2017-05-22 [S] [expires: 2025-05-20]
/etc/apt/trusted.gpg.d/debian-archive-stretch-stable.gpg
-----
pub rsa4096 2017-05-20 [SC] [expires: 2025-05-18]
  067E 3C45 6BAE 240A CEE8 8F6F EF0F 382A 1A7B 6500
uid [ unknown] Debian Stable Release Key (9/stretch) <debian-release@lists.debian.org>
```

IN PRATICA

Come aggiungere le trusted keys

Se aggiungerete al file sources.list una sorgente di pacchetti di terze parti, APT dovrà essere in grado di trovare la trusted key per la corrispondente autenticazione GPG (altrimenti vi reclamerà incessantemente di non poter verificare l'autenticità di pacchetti provenienti dal repository in questione). Quindi innanzitutto dovrete recuperarne la chiave pubblica: la maggior parte delle volte sarà fornita sotto forma di un file di testo, che denominiamo nei seguenti esempi key.asc . L'amministratore potrà quindi aggiungere la suddetta trusted key in un file *.asc in /etc/apt/trusted.gpg.d/. Questo metodo è stato supportato a partire da Stretch. Per i rilasci antecedenti potrete eseguire il comando apt-key add < key.asc con diritti di amministrazione.

Una volta che installerete le chiavi appropriate nel keyring, APT eseguirà sistematicamente i controlli delle firme prima di qualsiasi attività rischiosa e qualora l'autenticità di un pacchetto non possa essere verificata i front-ends vi comunicheranno un apposito avviso.

6.7 Aggiornamento di una distribuzione alla versione corrente Stable

Una delle caratteristiche più popolari di Debian è la sua capacità di aggiornare un sistema da una distribuzione stable alla successiva: attraverso il famoso comando apt-get dist-upgrade, che ha largamente contribuito alla reputazione del progetto. Con poche precauzioni pertanto potrete aggiornare un computer, in base alla velocità dei downloads dalle sorgenti dei pacchetti, da un minimo di pochi minuti ad un massimo di circa una decina di minuti.

6.7.1 Le procedure raccomandate

Dato che il sistema Debian può subire dei cambiamenti rilevanti tra un rilascio della Stable e l'altro, dovete avere cura di leggere le "release notes" [note di rilascio] prima di procedere con l'aggiornamento.

BASILARE

Le Release notes

Le release notes di un sistema operativo (e più genericamente di un software) sono dei documenti, che descrivono una visione d'insieme del software in questione, attraverso diversi dettagli in merito alla versione a cui sono riferiti. Questi documenti sono di breve lunghezza, comparati alla documentazione completa e contengono, solitamente, un riepilogo delle nuove funzionalità introdotte rispetto alle versioni precedenti. Inoltre forniscono delle istruzioni sulle procedure di aggiornamento, degli avvisi per gli utenti delle precedenti versioni e talvolta un elenco degli errori. Le Release notes sono disponibili sul Web: la corrente versione stable ha un URL dedicato, mentre le precedenti possono essere reperite attraverso il loro nome in codice.

- ♦ <https://www.debian.org/releases/stable/releasenotes>
- ♦ <https://www.debian.org/releases/stretch/releasenotes>

In questo paragrafo tratteremo l'aggiornamento da Stretch a Buster. Questa procedura è la più pesante su un sistema; inoltre come qualsiasi altra procedura non è esente al 100% da rischi e non dovrà quindi essere messa in atto prima di un backup dei dati più importanti.

Vi consigliamo, per accelerare i tempi dell'aggiornamento e facilitarne l'esecuzione, di censire i pacchetti installati e di mantenere solo quelli che sono realmente necessari. Gli strumenti per effettuare quanto sopra espresso sono aptitude, deborphan e debfoster (leggete il paragrafo 6.2.7, "Come resocontare i pacchetti installati automaticamente" a pagina 125). Ad esempio potete usare il seguente comando per poi utilizzare la modalità interattiva di aptitude ed effettuare un ulteriore controllo e perfezionare le rimozioni che avete programmato:

```
# deborphan | xargs aptitude --schedule-only remove
```

SUGGERIMENTO

Come "rilevare" i files modificati

Il comando debsums può verificare se i files sul sistema locale, appartenenti ad un pacchetto installato, sono stati modificati. Tale comando impiega un semplice algoritmo hashsum e l'informazione in /var/lib/dpkg/info/package.md5sums (andate a leggere al riguardo il paragrafo 5.2.3 Checksum, Elenco dei Files di Configurazione a pag. 89). Per rilevare i files di configurazione modificati utilizzate il comando debsums -ec. Per verificare l'intero sistema utilizzate debsums -c.

Passiamo ora all'aggiornamento vero e proprio. Innanzitutto dovrete modificare il file /etc/apt/sources.list per impostare ad APT di utilizzare i pacchetti provenienti da Buster e non da Stretch. Se questo file contiene solo riferimenti a Stable e non ad un nome in codice, non dovete effettuare alcuna modifica, dal momento che Buster, al momento della stesura di questo manuale, è la corrente versione Stable di Debian. In ogni caso dovete comunque effettuare un refresh del database dei pacchetti disponibili (attraverso il comando apt update oppure attraverso synaptic ed il pulsante refresh).

NOTA

Se l'informazione del repository cambia ...

Quando viene rilasciata una nuova versione di Debian, alcuni fields, come ad esempio Suite, dei files Release e di InRelease del repository possono essere stati modificati. L'effetto di ciò è che lo scaricamento dati dal repository viene negato sino a quando non ne viene confermata l'informazione alterata in modo che l'utente sia consapevole. Per confermare i cambiamenti dovete utilizzare le opzioni --allow-releaseinfo-change o --allow-releaseinfo-change-field di apt-get oppure l'opzione di configurazione Acquire::AllowReleaseInfoChange.

Dopo aver salvato le nuove sorgenti dei pacchetti, dovete effettuare un aggiornamento minimale con apt upgrade.

Così facendo semplificherete il carico di lavoro degli strumenti di gestione dei pacchetti ed in particolare, garantirete che i suddetti strumenti siano aggiornati alla loro ultima versione e che quindi ricevano quelle bugfixes e quei miglioramenti che potrebbero rivelarsi necessari durante l'aggiornamento completo della distribuzione.

Una volta completata la precedente fase, potrete passare all'aggiornamento effettivo, con apt full-upgrade, aptitude oppure con synaptic. Dovrete verificare le operazioni prima di eseguirle per aggiungere eventualmente dei pacchetti suggeriti o deselezionare dei pacchetti raccomandati, ma che siete certi non siano indispensabili. In ogni caso il front-end dovrebbe comunque giungere ad uno scenario che si conclude in un sistema (Buster) coerente ed aggiornato. [In informatica lo scenario è una serie di interazioni prevedibili degli utenti e dei loro ruoli con un sistema tecnologico che, di solito, comprende hardware e software per computer. Gli utenti (ed i loro ruoli) sono definiti "attori" nel Unified Modeling Language (UML). Uno scenario descrive come un sistema può o potrebbe essere utilizzato nel contesto di un'attività in un determinato intervallo di tempo. Tale intervallo di tempo può essere una singola transazione, un'operazione commerciale, un giorno o un altro periodo oppure l'intero ciclo di vita di un sistema operativo.] Poi dovete semplicemente stare seduti mentre APT compie la magia, attendere il download dei pacchetti e rispondere alle domande di Debconf (in particolare alle domande sui files di configurazione che potrebbero essere stati modificati localmente).

6.7.2 Come gestire i problemi dopo un aggiornamento

Nonostante tutti gli sforzi dei manutentori Debian, un importante aggiornamento del sistema operativo non ha sempre un esito così prevedibile come si potrebbe sperare. Le nuove versioni di

alcuni software sono talvolta incompatibili con quelle precedenti (per esempio a causa del cambiamento del loro funzionamento di default o del loro formato dati). Inoltre alcuni bug riescono comunque a passare inosservati nonostante la fase testing che precede il rilascio di una nuova versione Debian.

Per conoscere anticipatamente alcune di queste anomalie, potrete installare il pacchetto `apt-listchanges`, che vi mostrerà, all'inizio di un aggiornamento del pacchetto, tutte le informazioni inerenti i problemi noti. Queste informazioni, a beneficio degli utenti, sono redatte dai manutentori dei pacchetti ed incluse nel file `/usr/share/doc/package/NEWS`. Leggere questi documenti (anche attraverso `apt-listchanges`) vi eviterà di incorrere in brutte sorprese.

A volte la nuova versione di un software potrebbe non funzionare affatto. Questo solitamente avviene quando l'applicazione non è molto popolare e non è stata sufficientemente testata; inoltre un aggiornamento last-minute potrebbe anche aver introdotto regressioni che sono state scoperte solo dopo il rilascio della Stable. In ogni caso, dovete per prima cosa consultare il bug tracking system su <https://bugs.debian.org/package> (<https://bugs.debian.org>) e verificare se il problema è già noto ed è stato già segnalato. Qualora nel vostro sistema fosse già installato `apt-listbugs`, verrà elencato prima dell'inizio dell'aggiornamento. In caso contrario, sarebbe bene che lo segnalaste voi attraverso `reportbug`. Altrimenti se il bug è già noto, la mera lettura del bug report e dei relativi messaggi saranno un eccellente fonte di informazioni sull'anomalia:

- Capita qualche volta che una patch pronta all'uso sia già disponibile sul bug report; pertanto potrete ricompilare localmente una versione corretta del pacchetto Debian danneggiato (leggete al riguardo il paragrafo 15.1, "Ricompilazione di un Pacchetto dalle sue Sorgenti" a pagina 448).
- Certe volte sono gli stessi utenti a trovare un modo per aggirare il problema ed a condividere la loro esperienza nei loro replies al report.
- In altri casi ancora, il manutentore potrebbe aver preparato un fixed package ed averlo reso disponibile al pubblico.

A seconda della severità del bug, una nuova versione del pacchetto potrebbe essere stata realizzata specificatamente per la nuova revisione della stable release. In tal caso, il fixed package potrebbe essere stato reso disponibile nella sezione proposed-updates dei mirrors Debian (leggete al riguardo il paragrafo 6.1.2.3, "Proposed Updates" a pagina 112). Potrete quindi aggiungere temporaneamente la voce corrispondente nel file `sources.list` ed installare l'aggiornamento con `apt` o `aptitude`.

Qualora il pacchetto non sia ancora stato reso disponibile nella suddetta sezione, potrete verificare se è in attesa della validazione da parte degli SRM – Stable Release Managers. Potrete in tal senso consultare la loro pagina web. I pacchetti elencati nella loro pagina, pur non essendo ancora disponibili al pubblico, vi confermeranno alla fine che il loro processo di pubblicazione è in corso.

♦ <https://release.debian.org/proposed-updates/stable.html>

6.7.3. “Pulizia” dopo un aggiornamento

APT solitamente è in grado di garantire un aggiornamento pulito, installando dipendenze nuove e aggiornate o rimuovendo i pacchetti in conflitto. Ciò nonostante non può gestire autonomamente tutte le attività che gli utenti e gli amministratori dovranno affrontare dopo un aggiornamento, in quanto richiedono una decisione umana.

6.7.3.1 I pacchetti rimossi dall'archivio Debian

A volte i Debian FTP Master rimuovono i pacchetti dall'archivio Debian, perché contengono bugs critici per la release oppure perché sono stati abbandonati dall'autore a monte o dal manutentore dei pacchetti, o semplicemente in quanto hanno raggiunto la fine del loro ciclo di vita. Di conseguenza verranno esclusi nella nuova versione di Debian. Per trovare tutti i pacchetti che non hanno una sorgente dei pacchetti dovete utilizzare il comando apt-show-versions:

```
$ apt-show-versions | grep "No available version"
```

In alternativa potrete utilizzare per la ricerca aptitude search ~o. Se non avete più necessità dei pacchetti trovati vi suggeriamo di eliminarli dal sistema, perché non riceveranno più alcun aggiornamento per bug critici o relativi alla sicurezza.

6.7.3.2 Pacchetti Dummy e Transitional

A volte, un pacchetto necessita di un nuovo nome. In questo caso solitamente il vecchio pacchetto permane sotto forma di pacchetto (quasi) vuoto, dipendente dal nuovo pacchetto e che comporta l'installazione soltanto dei files obbligatori in /usr/share/doc/package/. Tali pacchetti sono denominati pacchetti "dummy" o "transitional". Se anche il responsabile della manutenzione dei pacchetti ha modificato la sezione del pacchetto in oldlibs, strumenti come aptitude, deborphan o debfoster (andate a leggere al riguardo la casella di testo "deborphan e debfoster" a pagina 126) saranno in grado di individuarli suggerendone la rimozione.

Sfortunatamente al momento non esiste un metodo infallibile per assicurarvi che questi pacchetti vengano automaticamente rimossi o individuati dagli strumenti sopra menzionati. Per accertarvi se nel sistema sono ancora installati alcuni dei suddetti pacchetti, potrete controllare le descrizioni dei pacchetti installati e verificarne i risultati. Vi suggeriamo di non programmare la rimozione automatica in base ai risultati scaturenti, in quanto tale metodo può giungere a dei risultati falsi-positivi:

```
$ dpkg -l | grep ^ii | grep -i -E "(transition|dummy)"
```

Il nuovo pacchetto verrà contrassegnato come dipendenza del pacchetto in transizione e di conseguenza come installazione automatica, pertanto potrebbe esserne programmata la rimozione qualora il pacchetto in transizione fosse eliminato con opzione purge. In questo caso potrete utilizzare uno degli approcci descritti nella casella di testo "Rimuovere ed installare simultaneamente" a pagina 118 e nel paragrafo 6.2.7, "Come resocontare i pacchetti installati automaticamente" pagina 125 per rimuovere selettivamente il pacchetto in transizione.

6.7.3.3 File di configurazione obsoleti o non utilizzati

Dopo un aggiornamento potrebbero essere rimasti dei files di configurazione di tipo cruft [espressione gergale che significa inutili] di: dpkg (andate a leggere il paragrafo 5.2.3, Checksum, Elenco dei Files di Configurazione a pagina 89), ucf [Update Configuration Files - utilizzati per preservare durante gli aggiornamenti le personalizzazioni degli utenti] o provenienti dai pacchetti rimossi. Questo tipo di files potranno essere eliminati con opzione purge attraverso il comando apt autoremove --purge. I files di configurazione, che sono stati gestiti da dpkg o ucf durante il processo di aggiornamento, si presentano con un suffisso dedicato (ad esempio .dpkg-dist, .dpkg-old, .ucf-old). Potrete rintracciarli attraverso i comandi find e locate. Se inutilizzati, possono essere cancellati.

6.7.3.4 I files che non appartengono ad alcun pacchetto

La Debian Policy impone che i pacchetti rimossi con opzione purge non debbano lasciare nel sistema alcun file. La violazione di questo principio rappresenta un bug serio e solitamente dovrà incontrarlo raramente. Qualora riscontriate la suddetta violazione nel vostro sistema vi invitiamo ad effettuarne il report; se siete curiosi e volete accertarvene immediatamente, potrete impiegare il pacchetto `cruft` o `cruft-ng` per verificare se nel vostro sistema sono presenti dei files che non appartengono ad alcun pacchetto.

6.8 Mantenere un Sistema Aggiornato

Debian è una distribuzione dinamica che evolve continuamente nel tempo. Sebbene la maggior parte delle modifiche siano effettuate nelle versioni Testing e Unstable, anche la versione Stabile viene aggiornata di tanto in tanto, per lo più per fixes di sicurezza. Indipendentemente dalla versione di Debian installata nel sistema, solitamente conviene mantenerla aggiornata, così che possiate beneficiare degli ultimi sviluppi e delle ultime bug fixes.

Pur essendo possibile verificare la presenza di aggiornamenti periodicamente attraverso uno strumento di verifica associato e quindi avviare la procedura di aggiornamento, tale attività, come tutte le attività ripetitive, rimane un compito noioso, soprattutto se avrete la necessità di eseguirla su diverse macchine. Fortunatamente, come in tutte le attività ripetitive, esistono degli strumenti già pronti all'uso e sviluppati per automatizzare parte delle operazioni.

Il primo strumento che merita di essere citato fra questi è `apticron`, incluso nel pacchetto omonimo. Tale strumento in sostanza è in grado di avviare giornalmente lo script (attraverso cron). Lo script aggiorna l'elenco dei pacchetti disponibili e, se sono installati nel sistema dei pacchetti non aggiornati alla loro ultima versione, invia un'e-mail con l'elenco dei pacchetti non aggiornati, nonché una descrizione delle modifiche incluse nelle loro versioni più recenti. Ovviamente questo script è principalmente rivolto agli utenti di Debian Stable in quanto le e-mails giornaliere sarebbero molto più estese per le versioni di Debian con ritmi di sviluppo più incalzanti. Quando gli aggiornamenti sono disponibili, `apticron` li scarica automaticamente, ma non li installa. L'amministratore a sua scelta potrà quindi eseguire l'aggiornamento, ma più rapidamente in quanto i pacchetti saranno già disponibili localmente nella cache di APT.

Gli amministratori incaricati alla gestione di diverse macchine apprezzeranno certamente di essere avvisati sugli aggiornamenti disponibili, ma questa operazione in sé può rimanere comunque noiosa. Pertanto potrete attivare gli aggiornamenti periodici: questi sfruttano `systemd timer unit` o `cron`. Qualora non fosse installato `systemd`, lo script `/etc/cron.daily/apt-compat` (incluso nel pacchetto `apt`) potrebbe tornarvi utile. Anche questo script è avviato giornalmente (e non con interfaccia interattiva) attraverso `cron`. Per controllarne il funzionamento, dovete utilizzare le variabili di configurazione APT (che dovete di conseguenza immettere in un file sotto `/etc/apt/apt.conf.d/10periodic`). Le più importanti sono:

APT::Periodic::Update-Package-Lists Questa opzione consente di specificare la frequenza (in giorni) per il refresh dell'elenco dei pacchetti. Gli utenti di `apticron` possono farne a meno, in quanto tale attività è già eseguita dallo strumento summenzionato.

APT::Periodic::Download-Upgradeable-Packages Anche questa opzione consente di specificare una frequenza in giorni, ma si riferisce al download dei pacchetti aggiornabili. Gli utenti di `apticron` possono fare a meno anche di questa variabile.

APT::Periodic::AutocleanInterval Questa opzione rimpiazza `apticron` in una funzionalità che lo stesso `apticron` non possiede. Tale variabile indica la frequenza con cui i pacchetti obsoleti (ossia quelli che non sono più referenziati da nessuna distribuzione) possono essere eliminati dalla cache di

APT automaticamente. Questa task riporta la cache di APT a delle dimensioni ragionevoli consentendovi di non dovervene più preoccupare.

APT::Periodic::Unattended-Upgrade Quando questa opzione è abilitata, lo script eseguirà giornalmente un unattended-upgrade (attraverso il pacchetto unattended-upgrade) che, come suggerisce il nome, può automatizzare il processo di aggiornamento di alcuni pacchetti (per impostazione predefinita, si occupa solo degli aggiornamenti di sicurezza, ma questa attività può essere personalizzata attraverso il file /etc/apt/apt.conf.d/50unattended-upgrades). Si precisa che questa opzione può essere configurata con l'aiuto di debconf, eseguendo il comando dpkg-reconfigure -p low unattended-upgrade. Qualora abbiate installato anche apt-listbugs, questi impedirà l'aggiornamento automatico dei pacchetti di cui esiste il report di bug seri o gravi.

Ci sono anche altre opzioni che vi possono consentire di controllare con maggiore precisione il funzionamento della cleaning cache (pulizia della cache). Sono descritte nello stesso script /usr/lib/apt/apt.systemd.daily, però non ne discuteremo qui.

Questi strumenti funzionano meglio con i servers, piuttosto che con i desktops in quanto, generalmente, gli utenti desktop preferiscono un sistema più interattivo. Il pacchetto gnome-software dispone di una piccola icona nell'area di notifica dell'ambiente desktop quando sono disponibili gli aggiornamenti; con un clic su questa icona si avvia un'interfaccia semplificata per eseguire gli aggiornamenti. Attraverso questa interfaccia potrete: consultare gli aggiornamenti disponibili; leggere una descrizione in breve dei pacchetti più rilevanti e le corrispondenti changelog entries; decidere caso per caso se installare o non installare un aggiornamento.

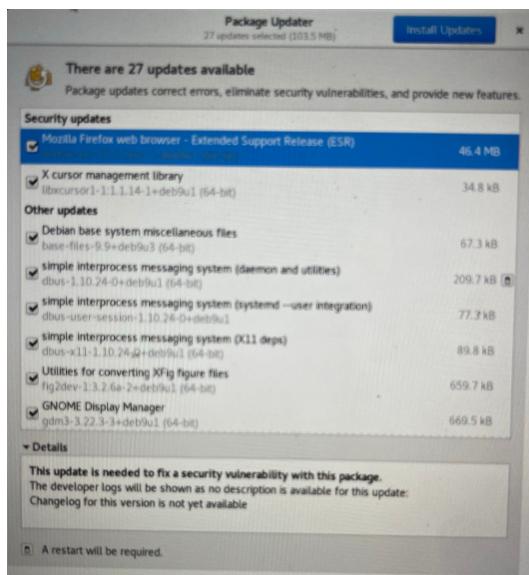


Figura 6.3 Aggiornamento con gpk-update-viewer

Il sopramenzionato strumento non viene più installato nell'ambiente desktop predefinito di GNOME. Tale decisione è dovuta ad una nuova filosofia secondo cui gli aggiornamenti dovrebbero essere automaticamente installati in background o preferibilmente durante lo shutdown del computer.

6.9 Aggiornamenti Automatici

La Falcot Corp, pur possedendo molte macchine, ha risorse umane limitate, pertanto gli amministratori desiderano automatizzare il più possibile gli aggiornamenti. I programmi delegati a questi processi devono di conseguenza essere eseguiti senza l'intervento umano.

6.9.1 Come configurare dpkg

È stato già trattato (andate a leggere al riguardo la casella di testo "Come evitare le domande per il file di configurazione" a pagina 90) come istruire dpkg a non chiedere conferma quando viene sostituito un file di configurazione (con le opzioni `--force-confdef` `--force-confold`). Le interazioni [molto genericamente il rapporto, con reciproche influenze, che si verifica quando l'uomo "guida" la macchina] possono tuttavia presentarsi comunque per le seguenti tre "sorgenti": possono essere generate dallo stesso APT; possono essere gestite da debconf; possono verificarsi da riga di comando come conseguenza agli scripts di configurazione di un pacchetto (qualche volta gestiti da ucf).

6.9.2 Come configurare APT

La configurazione di APT è semplice: l'opzione `-y` (o `--assume-yes`) istruirà APT a rispondere automaticamente "sì" a qualsiasi sua domanda.

6.9.3 Come configurare debconf

La configurazione di debconf merita ulteriori dettagli. Fin dalla sua creazione, questo programma è stato designato per verificare la rilevanza e la quantità delle domande poste all'utente, nonché il metodo attraverso cui vengono poste. Per questo motivo per la sua configurazione occorre che sia definito un livello di priorità minima superato il quale debconf porrà la domanda. debconf utilizza automaticamente la risposta predefinita (definita dal manutentore del pacchetto) per le domande che deve saltare.

Un altro importante elemento che si può configurare è l'interfaccia [ossia, molto genericamente, l'area o la superficie sulla quale due entità qualitativamente differenti si incontrano] ottenuta attraverso il front-end. Se sceglierete la modalità noninteractive disabiliterete qualsiasi interazione utente. Ma le note informative dei pacchetti (non visualizzate nonostante i tentativi degli stessi pacchetti ed a causa della modalità noninteractive) saranno trasmesse all'amministratore da debconf via posta elettronica.

Per riconfigurare debconf dovrete utilizzare lo strumento `dpkg-reconfigure` incluso nel pacchetto debconf; il correlato comando è `dpkg-reconfigure debconf`. Si precisa che i valori di configurazione possono anche essere temporaneamente annullati attraverso le environment variables [le variabili d'ambiente] (per esempio `DEBIAN_FRONTEND` è un'environment variable, che consente di controllare l'interfaccia, come spiegato nella manual page di debconf (?)).

6.9.4 Come comportarsi con le interazioni da riga di comando

Infine, le "sorgenti" di interazioni più difficili da eliminare sono gli scripts di configurazione eseguiti da dpkg. Sfortunatamente non esiste una soluzione standard e nessuna è di fatto migliore di un'altra.

Un approccio genericamente soddisfacente è quello di disabilitare lo standard input attraverso la redirezione del contenuto (vuoto) di /dev/null in esso con il comando </dev/null o intasandolo con un flusso dati continuo di newlines [la newline è un carattere di controllo o una sequenza di caratteri di controllo; la newline viene utilizzata per indicare la fine di una riga di testo e l'inizio di una nuova].

[Lo standard input è un canale ("stream") da cui giunge un flusso di dati, spesso testuali, in ingresso ad un programma. Il programma ne richiede il trasferimento eseguendone le operazioni di lettura. Non tutti i programmi necessitano di stream input: ad esempio i comandi ls o dir (che mostrano i nomi dei files contenuti in una directory) accettano argomenti da riga di comando, eseguendo le loro operazioni senza bisogno di leggere i dati da un canale input. Lo standard input, salvo casi di redirezione, viene ereditato dal Parent Process (processo padre) o è associato ad una tastiera nel caso di una interactive shell – shell in inglese è l'interprete dei comandi. La redirezione è la deviazione verso posizioni specificate dall'utente dei canali standard (standard input, standard output e standard error) ovvero la deviazione dalle loro destinazioni (nel caso dello standard output e dello standard error) o dalle loro sorgenti (nel caso dello standard input).]

Nessuno di questi metodi è affidabile al 100%, ma quanto meno vi consentiranno di ripiegare su delle risposte predefinite, poiché la maggior parte degli scripts interpreta l'assenza di una risposta esplicita come una convalida del valore predefinito.

6.9.5 La combinazione miracolosa

Se mettiamo insieme gli elementi di configurazione esposti nei paragrafi precedenti, è possibile scrivere un piccolo script, ma abbastanza affidabile, in grado di eseguire un aggiornamento automatico.

Esempio 6.5 Script per eseguire un aggiornamento non interattivo

```
export DEBIAN_FRONTEND=noninteractive
yes '' | apt-get -y -o DPkg::options::="--force-confdef" -o DPkg::options::="--force-
confold" dist-upgrade
```

IN PRATICA

Il caso-studio
Falcot Corp

I computers della Falcot costituiscono un sistema eterogeneo con macchine preposte a scopi diversi. Gli amministratori pertanto dovranno scegliere la soluzione più adatta per ogni macchina.
Di fatto, i servers che eseguono Buster sono configurati con la "combinazione miracolosa" sopra menzionata, per essere aggiornati automaticamente. Solo i servers più critici (come ad esempio i firewalls) sono configurati per utilizzare apicron, in modo che gli aggiornamenti si svolgano solo sotto la supervisione di un amministratore.
Anche le workstations dell'ufficio amministrativo eseguono Buster, ma sono state configurate con gnome-packagekit, in modo che gli utenti possano avviare autonomamente gli aggiornamenti. Difatti degli aggiornamenti non richiesti esplicitamente potrebbero rendere il funzionamento del computer inaspettatamente diverso e disorientare i main users.
Infine, i pochi computers in laboratorio utilizzano la Testing - per beneficiare delle ultime versioni software - e non sono aggiornati automaticamente.
Solo gli amministratori sono autorizzati a configurare APT in modo che prepari periodicamente gli aggiornamenti, senza poterli mettere in esecuzione; gli amministratori, così facendo, potranno effettuare gli aggiornamenti (manualmente), evitando le noiose fasi di refresh dell'elenco dei pacchetti e di download, e potranno concentrarsi sulle fasi davvero rilevanti.

6.10 Come ricercare i Pacchetti

L'enorme e sempre crescente quantità di software distribuita da Debian determina un paradosso: per quanto possa esistere un tool ad hoc per qualsivoglia attività, al bisogno è molto difficile trovarne uno in mezzo alla caterva di pacchetti. La mancanza di un appropriato metodo di ricerca (e di un conseguente reperimento) dei tools appropriati è stato per lungo tempo un problema.

Fortunatamente, questo disagio è stato quasi del tutto risolto.

Il metodo di ricerca più banale è sicuramente quello che prevede l'utilizzo del nome esatto del pacchetto. In pratica se apt show package restituisce un risultato, allora il pacchetto esiste. Sfortunatamente, il summenzionato metodo richiede che l'utente conosca il nome del pacchetto o che quanto meno sia in grado di presumerlo, il che non è sempre scontato.

SUGGERIMENTO

Le convenzioni per il package naming

Alcune categorie di pacchetti ricevono il loro nome in base ad uno schema convenzionale di nomenclatura; conoscere lo schema può qualche volta essere d'aiuto per indovinare l'esatto nome del pacchetto Debian. Ad esempio, per i moduli Perl, lo schema convenzionale impone che un modulo upstream con il nome `XML::Handler::Composer` debba essere "impacchettato" come `libxml-handler-composer-perl`. La libreria per l'utilizzo del sistema `gconf` in Python è "impacchettata" come `python-gconf`. Sfortunatamente non è possibile definire uno schema di nomenclatura valido in generale per tutti i pacchetti, anche se i manutentori dei pacchetti cercano solitamente di assecondare le preferenze degli sviluppatori upstream.

Un filtro di ricerca leggermente più efficace, anche se alquanto limitato, è quello in plain-text [in "testo puro"] dei nomi dei pacchetti. Potrete ottenere dei risultati ricercando i pacchetti in base alle loro descrizioni: difatti ciascun pacchetto possiede, oltre ad un nome, una descrizione più o meno dettagliata ed una ricerca che si avvale di parole chiave potrebbe rivelarsi utile. `apt-cache` e `axi-cache` sono gli strumenti utilizzati per questo tipo di ricerca (andate a leggere la casella di testo `axi-cache` a pag. 127); ad esempio, `apt-cache search video` restituirà un elenco di tutti i pacchetti che contengono nel loro nome o nella loro descrizione la parola chiave "video".

Per eseguire ricerche più complesse, è necessario uno strumento più efficace come `aptitude`. `aptitude` consente di effettuare una ricerca nei fields meta-data dei pacchetti che si basa su una logical expression [una espressione logica è il significato (o valore) di una "funzione booleana", che a sua volta può essere rappresentata in una "tabella della verità"]. Ad esempio, attraverso il comando sottostante potrete ottenere l'elenco dei pacchetti che contengono: nel loro nome il termine `kino`; nella loro descrizione il termine `video`; nel nome del loro manutentore il termine `paul`.

```
$ aptitude search kino~dvideo~mpaul
p kino - Non-linear editor for Digital Video data
$ aptitude show kino
Package: kino
Version: 1.3.4+dfsg0-1
State: not installed
Priority: optional
Section: video
Maintainer: Paul Brossier <piem@debian.org>
Architecture: amd64
Uncompressed Size: 8,304 k
Depends: libasound2 (>= 1.0.16), libatk1.0-0 (>= 1.12.4), libavc1394-0 (>= 0.5.3),
-> libavcodec58 (>=7:4.0) | libavcodec-extra58 (>= 7:4.0), libavformat58 (>= 7:4.0),
-> libavutil56 (>= 7:4.0),
libc6 (>= 2.14), libcairo2 (>= 1.2.4), libdv4 (>= 1.0.0), libfontconfig1 (>=
-> 2.12.6),
libfreetype6 (>= 2.2.1), libgcc1 (>= 1:3.0), libgdk-pixbuf2.0-0 (>= 2.22.0),
-> libglade2-0
(>= 1:2.6.4-2~), libglib2.0-0 (>= 2.16.0), libgtk2.0-0 (>= 2.24.32), libice6
-> (>=1:1.0.0),
libiec61883-0 (>= 1.2.0), libpango-1.0-0 (>= 1.14.0), libpangocairo-1.0-0
-> (>= 1.14.0),
libpangoft2-1.0-0 (>= 1.14.0), libquicktime2 (>= 2:1.2.2), libraw1394-11,
-> libsamplerate0
(>= 0.1.7), libsm6, libstdc++6 (>= 5.2), libswscale5 (>= 7:4.0), libx11-6,
-> libxext6,
libxml2 (>= 2.7.4), libxv1, zlib1g (>= 1:1.1.4)
```

```
Recommends: ffmpeg, curl
Suggests: udev | hotplug, vorbis-tools, sox, mjpegtools, lame, ffmpeg2theora
Conflicts: kino-dvtitler, kino-timfx, kinoplus
Replaces: kino-dvtitler, kino-timfx, kinoplus
Provides: kino-dvtitler, kino-timfx, kinoplus
Description: Non-linear editor for Digital Video data
Kino allows you to record, create, edit, and play movies recorded with DV camcorders
-> . This program
uses many keyboard commands for fast navigating and editing inside the movie.
```

```
The kino-timfx, kino-dvtitler and kinoplus sets of plugins, formerly distributed as
-> separate
packages, are now provided with Kino.
Homepage: http://www.kinodv.org/
Tags: field::arts, hardware::camera, implemented-in::c, implemented-in::c++,
-> interface::graphical,
interface::x11, role::program, scope::application, suite::gnome, uikit::gtk
->,
use::editing, use::learning, works-with::video, x11::application
```

La ricerca soprastante ha restituito come risultato un solo pacchetto che si chiama `kino` e che soddisfa tutte le tre condizioni richieste.

Le ricerche multi-criterio, essendo carenti di flessibilità, spesso non sono utilizzate nonostante ve ne sia l'opportunità. Pertanto è stato sviluppato un nuovo tagging system [sistema di "marcatori" o "etichette"], che offre un nuovo approccio alla ricerca. I pacchetti sono dotati di tags che ne consentono una classificazione tematica capillare e diversificata, denominata "facet-based classification". [Un sistema facet-based classification o di classificazione a faccette o di classificazione analitico-sintetica è un sistema di classificazione dove appunto le "faccette" rappresentano le componenti di un soggetto o classe definite in modo chiaro, reciprocamente indipendenti e collettivamente esaustive. La premessa è che qualsiasi soggetto o classe possa essere analizzato in tutte le sue componenti (aspetti, proprietà, caratteristiche, ecc.).] Nell'esempio soprastante, le tags del pacchetto "Kino" lo descrivono come un software per l'editing basato su Gnome che processa dati video.

La consultazione di questa classificazione quindi vi agevererà la ricerca di un pacchetto che soddisfi le vostre esigenze; qualora il sistema di classificazione vi restituiscia come risultato un (moderato) numero di pacchetti, potrete comunque completare la vostra ricerca manualmente. Per effettuare delle ricerche con le tags potrete usare il pattern (filtro) di ricerca `-G` in aptitude, ma troverete sicuramente più agevole consultare il sito dove sono gestiscono le tags:

♦ <https://debtags.debian.org/>

Se utilizzerete le tags `works-with::video` e `use::editing` darete la precedenza nella vostra ricerca ad una manciata di pacchetti, tra i quali i video editors `kino` e `pitivi`. L'impiego di questo sistema di classificazione è destinato a crescere sempre più nel tempo ed i sistemi di gestione dei pacchetti saranno gradualmente provvisti di efficienti search interfaces basate sulla facet-based classification.

In sintesi, il tool più adatto alle vostre esigenze dipende dalla complessità della ricerca che desiderate effettuare:

- `apt-cache` vi consente di cercare delle parole chiave contenute nel nome e nella descrizione dei pacchetti ed è molto utile per trovare rapidamente uno specifico pacchetto che soddisfi poche parole chiave ben mirate;
- se i vostri criteri di ricerca includono anche le relationships tra i pacchetti o altri metadati, come ad esempio il nome del manutentore, potrete usare `synaptic`;

- se desiderate una ricerca basata sulle tags un buon tool è `package-search`, un'interfaccia grafica dedicata alle ricerche nell'elenco dei pacchetti disponibili secondo diversi criteri (compresi i nomi dei files che contengono). Per le ricerche da riga di comando `axi-cache` farà a caso vostro;
- Infine se le vostre ricerche comprendono espressioni logiche complesse con operatori logici [o connettivi logici], lo strumento idoneo è la potente (ma complessa) search pattern sintax di `aptitude`; `aptitude` è disponibile sia da riga di comando, sia in modalità interattiva.

[Parole chiave](#)
[Documentazione](#)
[Solving problems](#)
[Log Files](#)
[README.Debian](#)
[Manual Pages](#)
[Info](#)

