

Automatische Updates Debian Ubuntu

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen 🙏. Gerne. Wir bedanken uns bei dir für deine Spende! ☐

Spenden

Zum frei verfügbaren [Apt-Repository](#)



GITLAB:

Die automatische Installation der Updates wird von unattended-upgrades übernommen. Dieses speichert Logdateien im Verzeichnis /var/log/unattended-upgrades/. Sollte versucht werden den Computer herunterzufahren, während unattended-upgrades arbeitet, verzögert der gleichnamige Dienst dies so lange wie nötig, sodass das Herunterfahren länger dauert. Das ist wichtig, damit keine unvollständig installierten oder konfigurierten Pakete hinterlassen werden.

Um unattended-upgrades manuell (ohne GUI) zu aktivieren, muss man die Datei /etc/apt/apt.conf.d/10periodic oder je nach Release /etc/apt/apt.conf.d/20auto-upgrades bearbeiten, sodass sie mindestens diese Einträge enthält: Anzahl in Tagen.

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::AutocleanInterval "30";
APT::Periodic::Unattended-Upgrade "7";
```

Die Zeilen erklären sich wohl von alleine. Die Configdatei hierfür wäre z.B. (Ubuntu 16.04.2) das:
nano /etc/apt/apt.conf.d/50unattended-upgrades

[50unattended-upgrades](#)

```
// Automatically upgrade packages from these (origin:archive) pairs
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but
if
    // available, the policy for updates is such that unattended-
upgrades
    // should also install from here by default.
    "${distro_id}ESM:${distro_codename}";
    "${distro_id}:${distro_codename}-updates";
//
    "${distro_id}:${distro_codename}-proposed";
    "${distro_id}:${distro_codename}-backports";
};

// List of packages to not update (regex are supported)
```

```
Unattended-Upgrade::Package-Blacklist {
//      "vim";
//      "libc6";
//      "libc6-dev";
//      "libc6-i686";
};

// This option allows you to control if on a unclean dpkg exit
// unattended-upgrades will automatically run
// dpkg --force-confold --configure -a
// The default is true, to ensure updates keep getting installed
Unattended-Upgrade::AutoFixInterruptedDpkg "true";

// Split the upgrade into the smallest possible chunks so that
// they can be interrupted with SIGUSR1. This makes the upgrade
// a bit slower but it has the benefit that shutdown while a upgrade
// is running is possible (with a small delay)
//Unattended-Upgrade::MinimalSteps "true";

// Install all unattended-upgrades when the machine is shutting down
// instead of doing it in the background while the machine is running
// This will (obviously) make shutdown slower
Unattended-Upgrade::InstallOnShutdown "false";

// Send email to this address for problems or packages upgrades
// If empty or unset then no email is sent, make sure that you
// have a working mail setup on your system. A package that provides
// 'mailx' must be installed. E.g. "user@example.com"
// Unattended-Upgrade::Mail "test@bla.com";

// Set this value to "true" to get emails only on errors. Default
// is to always send a mail if Unattended-Upgrade::Mail is set
Unattended-Upgrade::MailOnlyOnError "true";

// Do automatic removal of new unused dependencies after the upgrade
// (equivalent to apt-get autoremove)
Unattended-Upgrade::Remove-Unused-Dependencies "true";

// Automatically reboot *WITHOUT CONFIRMATION*
// if the file /var/run/reboot-required is found after the upgrade
Unattended-Upgrade::Automatic-Reboot "true";

// If automatic reboot is enabled and needed, reboot at the specific
// time instead of immediately
// Default: "now"
Unattended-Upgrade::Automatic-Reboot-Time "06:30";

// Use apt bandwidth limit feature, this example limits the download
// speed to 70kb/sec
```

```
//Acquire::http::Dl-Limit "70";
```

Hier kann man sich die Dinge einstellen die man benötigt.

Automatisches entfernen alter Kernels Ubuntu 16.04

Bei 18.04 passiert das schon automatisch. Kernels werden von Debian basierendes Distributionen leider nie automatisch entfernt. Es gibt hierfür auch keine Option. War vom Hersteller bis jetzt nicht vorgesehen. Abhilfe zusammen mit dem obigen Updateverfahren verschafft uns DPKG mit dem wir fast alles bewerkstelligen können.

Dieses kleine Script speichern wir unter „**/usr/local/bin/kernelpurge**“ ab, machen es ausführbar, und stellen es im Crontab so ein das es bei jedem Boot ausgeführt wird. Das macht Sinn da wir ja bei den automatischen Updates auch Autoreboot aktiviert haben. Somit bleibt das System auch vor zu vielen alten Kernels verschont.

[kernelpurge](#)

```
#!/bin/bash
dpkg -l 'linux-*' | sed '/^ii/!d;/'"$(uname -r | sed "s/\(.*)-\n([\^0-9]+\)/\1/")"'/d;s/^[^ ]* [^ ]* \([^\ ]*\).*/\1;/[0-9]!/d' |
xargs apt-get -y purge
```

```
chmod +x /usr/local/bin/kernelpurge
echo "@reboot root /usr/local/bin/kernelpurge #Löscht alle alten
Linuxkernels" >> /etc/crontab
```

Automatisches Upgrade Ubuntu 18.04 inkl. Kernel autoremove

nano /etc/apt/apt.conf.d/50unattended-upgrades

[50unattended-upgrades](#)

```
// Automatically upgrade packages from these (origin:archive) pairs
//
// Note that in Ubuntu security updates may pull in new dependencies
// from non-security sources (e.g. chromium). By allowing the release
// pocket these get automatically pulled in.
Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
```

```
"${distro_id}:${distro_codename}-security";
// Extended Security Maintenance; doesn't necessarily exist for
// every release and this system may not have it installed, but
if
    // available, the policy for updates is such that unattended-
    upgrades
    // should also install from here by default.
"${distro_id}ESM:${distro_codename}";
"${distro_id}:${distro_codename}-updates";
// "${distro_id}:${distro_codename}-proposed";
// "${distro_id}:${distro_codename}-backports";
};

// List of packages to not update (regexp are supported)
Unattended-Upgrade::Package-Blacklist {
//     "vim";
//     "libc6";
//     "libc6-dev";
//     "libc6-i686";
};

// This option will controls whether the development release of Ubuntu
// will be
// upgraded automatically.
Unattended-Upgrade::DevRelease "false";

// This option allows you to control if on a unclean dpkg exit
// unattended-upgrades will automatically run
// dpkg --force-confold --configure -a
// The default is true, to ensure updates keep getting installed
Unattended-Upgrade::AutoFixInterruptedDpkg "true";

// Split the upgrade into the smallest possible chunks so that
// they can be interrupted with SIGTERM. This makes the upgrade
// a bit slower but it has the benefit that shutdown while a upgrade
// is running is possible (with a small delay)
Unattended-Upgrade::MinimalSteps "false";

// Install all unattended-upgrades when the machine is shutting down
// instead of doing it in the background while the machine is running
// This will (obviously) make shutdown slower
Unattended-Upgrade::InstallOnShutdown "false";

// Send email to this address for problems or packages upgrades
// If empty or unset then no email is sent, make sure that you
// have a working mail setup on your system. A package that provides
// 'mailx' must be installed. E.g. "user@example.com"
Unattended-Upgrade::Mail "meine Email";

// Set this value to "true" to get emails only on errors. Default
```

```
// is to always send a mail if Unattended-Upgrade::Mail is set
Unattended-Upgrade::MailOnlyOnError "true";

// Remove unused automatically installed kernel-related packages
// (kernel images, kernel headers and kernel version locked tools).
Unattended-Upgrade::Remove-Unused-Kernel-Packages "true";

// Do automatic removal of new unused dependencies after the upgrade
// (equivalent to apt-get autoremove)
Unattended-Upgrade::Remove-Unused-Dependencies "true";

// Automatically reboot *WITHOUT CONFIRMATION*
// if the file /var/run/reboot-required is found after the upgrade
Unattended-Upgrade::Automatic-Reboot "true";

// If automatic reboot is enabled and needed, reboot at the specific
// time instead of immediately
// Default: "now"
Unattended-Upgrade::Automatic-Reboot-Time "03:00";

// Use apt bandwidth limit feature, this example limits the download
// speed to 70kb/sec
//Acquire::http::Dl-Limit "70";

// Enable logging to syslog. Default is False
// Unattended-Upgrade::SyslogEnable "false";

// Specify syslog facility. Default is daemon
// Unattended-Upgrade::SyslogFacility "daemon";
```

Automatische Updates für alle Quellen

Möchte man für alle Quellen, PPA's usw. die Updates aktivieren fügt man folgenden Teil ein

```
Unattended-Upgrade::Origins-Pattern {
    "origin=*";
};
```

Beispiel:

```
// Automatically upgrade packages from these (origin:archive) pairs
//
// Note that in Ubuntu security updates may pull in new dependencies
// from non-security sources (e.g. chromium). By allowing the release
// pocket these get automatically pulled in.

Unattended-Upgrade::Origins-Pattern {
    "origin=*";
```

```
};

Unattended-Upgrade::Allowed-Origins {
    "${distro_id}:${distro_codename}";
    "${distro_id}:${distro_codename}-security";
    // Extended Security Maintenance; doesn't necessarily exist for
    // every release and this system may not have it installed, but if
    // available, the policy for updates is such that unattended-
    upgrades
    // should also install from here by default.
    "${distro_id}ESMA:${distro_codename}-apps-security";
    "${distro_id}ESM:${distro_codename}-infra-security";
    "${distro_id}:${distro_codename}-updates";
    // "${distro_id}:${distro_codename}-proposed";
    "${distro_id}:${distro_codename}-backports";
};

// Python regular expressions, matching packages to exclude from upgrading
```

From:
<http://www.osit.cc/dokuwiki/> - DEEPDOC.AT - enjoy your brain

Permanent link:
http://www.osit.cc/dokuwiki/doku.php?id=server_und_serverdienste:automatische_updates_debian_ubuntu&rev=1747556768

Last update: 2025/11/29 22:06

