

openSUSE - Upgrade der Distribution mit einem Skript

Erst vor kurzem ist die neueste openSUSE-Distribution herausgekommen und nun will man die Distribution via "zypper dup" upgraden. Die meisten openSUSE-User müssen erstmal nach einer Anleitung im Internet suchen und diese dann Schritt für Schritt durchgehen. Das kann eine einfache oder auch eine komplizierte Angelegenheit werden.

Aus diesem Grund habe ich ein Skript `upgrade-opensuse.sh` entwickelt, das alle notwendigen Schritte eines Distributionsupgrades automatisch durchführt. Die Vorgehensweise des Skript ist ganz grob an das Upgrade-Tool `do-release-upgrade` von Ubuntu angelehnt. Wenn alle Pakete von zypper korrekt aufgelöst werden kann, ist es sogar möglich, dass der Upgrade-Prozess in einem Rutsch durchläuft und man am Ende nur noch neustarten muss. Das Skript merkt sich auch die Stelle, an der der Upgrade-Prozess abgebrochen wurde und wird beim erneuten Ausführen an der letzten Stelle fortfahren. So kann man zwischendurch ein Problem beheben und anschließend mit dem Upgrade-Prozess fortfahren.

Folgende Schritte werden durchgeführt:

- Ermittlung der eingesetzten openSUSE-Version.
- Überprüfung der Internetverbindung.
- Ermittlung der neuesten openSUSE-Version.
- Backup vom `/etc` Verzeichnis.
- Umbenennung des Verzeichnis der eingebunden Repos `/etc/zypp/repos.d` nach `/etc/zypp/repos.d.upgrade`.
- Hinzufügen der Online-Repos (OSS, NON-OSS, OSS Update, NON-OSS Update) von der neuesten openSUSE-Version.
- Upgrade der Distribution via `zypper dup` (Ohne Community-Repos, um ungewollte `VendorChanges` zu vermeiden).
- Hinzufügen aller vormals aktivierten Community-Repos.
- Temporäre Modifizierung der zypper Konfiguration, um `VendorChanges` zu erlauben.
- Überprüfung von alten openSUSE-Pakete im System. Es wird versucht, die alten Pakete durch neuere Pakete zu ersetzen.
- Rückgängig machen der temporäre Modifizierung der zypper Konfiguration.
- Alte openSUSE-Pakete, die nicht aktualisiert werden konnten, werden endgültig entfernt.
- Auflistung aller neuen bzw. modifizierten Konfigurationsdateien (`*.rpmnew`, `*rpmsave`).

Alle Vorgänge werden protokolliert, um später nachzuvollziehen, was genau am System verändert wurde.

Folgende selbsterklärenden Logdateien werden erzeugt:

- `upgrade-opensuse.zypper-dup-output`
- `upgrade-opensuse.old-packages-output`
- `upgrade-opensuse.zypper-reinstall-packages-output`
- `upgrade-opensuse.remove-old-packages-output`
- `upgrade-opensuse.zypper-rm-packages-output`
- `upgrade-opensuse.list-new-and-old-config-files`

Wichtiger Hinweis: Vor einem Distributionsupgrade bitte unbedingt ein Backup machen, um im Bedarfsfall auf ein aktuelles Backup zurückgreifen zu können! Außerdem gibt es z.B. RPM Pakete von Drittanbietern wie z.B. AMD Catalyst, NVIDIA, VirtualBox, CrossOver, HumbleBundle-Games, usw., die während des Upgrade-Prozess nicht angerührt werden und von Hand aktualisiert werden müssen.

Downloads:

openSUSE

- Skript: [upgrade-opensuse.sh](#)
- SHA1: [upgrade-opensuse.sh.sha1](#)

Das Skript wird via root ausgeführt und fängt sofort mit der Arbeit an. Es gibt zu Beginn ein Zeitfenster von 5 Sekunden, in der noch ein unkritischer Abbruch mit STRG+C möglich ist.

```
sudo sh upgrade-opensuse.sh
```

-h	Die Hilfe anzeigen lassen
-n/-non-interactive	Keine Fragen stellen, benutze automatisch Standard-Antworten. (zypper Option)
-r/-reset	Beginne das Disributionsupgrade von vorne (Die Option bitte vorsichtig verwenden!)
-V	Version des Skript anzeigen

Das Script:

```
#!/bin/bash
```

```
# Copyright (c) 2013-2015, Sebastian Siebert (mail@sebastian-siebert.de)
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
# * Redistributions of source code must retain the above copyright
# notice, this list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions and the following disclaimer in the
# documentation and/or other materials provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY SEBASTIAN SIEBERT "AS IS" AND ANY
# EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL SEBASTIAN SIEBERT BE LIABLE FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
# LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
# ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
# (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
# Changelog:
```

```
# Version 1.3 - Add parameter "-ov/--opensuseversion" for forcing upgrade another openSUSE
version
# Version 1.2 - Bugfix
# Version 1.1 - /etc/SuSE-release is deprecated, read instead from lsb_release
# Version 1.0 - Initial release
```

```
SCRIPT_VERSION="1.3"
```

openSUSE

```
OPENSUSE_DOWNLOAD_URL="http://download.opensuse.org/distribution/"
```

```
RESET_UPGRADE_PROGRESS="false"  
ZYPPER_GLOBAL_OPTION="-v"  
FORCE_SUSE_VERSION=""
```

```
function OutputUsage() {  
echo "Usage : $(basename $0) [options...]"  
echo "Options:"  
echo " -h/--help this help text"  
echo " -n/--non-interactive do not ask anything, use default answers automatically. (zypper option)"  
echo " -ov/--opensuseversion VERSION set openSUSE version for distribution upgrade (for example:  
13.1)"  
echo " -r/--reset start the distribution upgrade from beginning (use this option carefully)"  
echo " -V/--version show version number"  
}
```

```
while [ "$#" -gt "0" ]; do  
case $1 in  
-h|--help)  
OutputUsage  
exit 0  
;;  
-n|--non-interactive)  
ZYPPER_GLOBAL_OPTION="{ZYPPER_GLOBAL_OPTION} -n"  
shift 1  
;;  
-r|--reset)  
RESET_UPGRADE_PROGRESS="true"  
shift 1  
;;  
-ov|--opensuseversion)  
shift 1  
FORCE_SUSE_VERSION="$1"  
shift 1  
;;  
-V|--version)  
echo -e "$(basename $0) - Version ${SCRIPT_VERSION}\n"  
echo "Copyright (c) `date +%Y`, Sebastian Siebert (mail@sebastian-siebert.de)"  
echo "All rights reserved."  
echo "This script is under the modified BSD License (2-clause license)"  
exit 0  
;;  
-*)  
echo "Error: Option \"$1\" is unknown"  
echo "try '$(basename $0) -h' or '$(basename $0) --help' for more information"  
exit 1  
;;  
esac  
done
```

```
# Check the size of the console  
set -- $(stty size 2> /dev/null || echo 0 0)  
LINES=$1  
COLUMNS=$2
```

openSUSE

```
if [ ${LINES} -eq 0 ]; then
LINES=24
fi
if [ ${COLUMNS} -eq 0 ]; then
COLUMNS=80
fi

print_okay() {
echo -e "\033[${COLUMNS}C\033[15D[\e[1;32m OK \e[0m]"
}
print_failure() {
echo -e "\033[${COLUMNS}C\033[15D[\e[1;31m FAILURE \e[0m]"
}
print_missing() {
echo -e "\033[${COLUMNS}C\033[15D[\e[1;33m MISSING \e[0m]"
}
print_aborted() {
echo -e "\033[${COLUMNS}C\033[15D[\e[1;33m ABORTED \e[0m]"
}

print_pkg_available() {
echo -e "\e[1;32m AVAILABLE \e[0m]"
}
print_pkg_missing() {
echo -e "\e[1;31m MISSING \e[0m]"
}

box_full() {
printf '*%.0s' $(seq 1 67)
echo ""
}

box_content() {
echo -n "*"
if [ $# -eq 0 ]; then
printf ' %.0s' $(seq 1 65)
elif [ $# -eq 1 ]; then
RIGHT="$1"
printf ' %.0s' $(seq 1 17)
echo -n "${RIGHT}"
printf ' %.0s' $(seq 1 ${48-${#RIGHT}})
elif [ $# -eq 2 ]; then
LEFT="$1"
RIGHT="$2"
printf ' %.0s' $(seq 1 3)
echo -n "${LEFT}"
printf ' %.0s' $(seq 1 ${14-${#LEFT}})
echo -n "${RIGHT}"
printf ' %.0s' $(seq 1 ${48-${#RIGHT}})
fi
echo "*"
}

load_step() {
```

openSUSE

```
if [ ! -f upgrade-opensuse.step ]; then
save_step 0 "Start"
fi
. ./upgrade-opensuse.step
}
```

```
set_variable_step() {
echo "$1=\"$2\" >>upgrade-opensuse.step
}
```

```
save_step() {
# $1 = Step number
# $2 = Step reason
echo "STEP_NUMBER=$1" >>upgrade-opensuse.step
echo "STEP_INFO=\"$2\" >>upgrade-opensuse.step
echo -e "\n#####\n" >>upgrade-
opensuse.step
}
```

```
box_full
box_content
box_content "Script:" "$(basename $0)"
box_content "Version:" "${SCRIPT_VERSION}"
box_content "Written by:" "Sebastian Siebert (mail@sebastian-siebert.de)"
box_content
box_content "Description:" "This script helps you to upgrade openSUSE"
box_content "and logged all upgrade activities."
box_content
box_content "License:" "This script is under the"
box_content "modified BSD License (2-clause license)"
box_content
box_full
```

```
echo -n "Check for running this script as root ..."
if [ "$(id -u)" != "0" ]; then
echo -n -e "\n Error: This script needs to run as root!"
print_failure
exit 1
fi
print_okay
```

```
if [ "${RESET_UPGRADE_PROGRESS}" = "true" ]; then
echo -n "Reset the distribution upgrade progress ..."
rm upgrade-opensuse.list-new-and-old-config-files
rm upgrade-opensuse.old-packages-output
rm upgrade-opensuse.remove-old-packages-output
rm upgrade-opensuse.step
rm upgrade-opensuse.zypper-dup-output
rm upgrade-opensuse.zypper-reinstall-packages-output
rm upgrade-opensuse.zypper-rm-packages-output
print_okay
fi
```

openSUSE

load_step

```
if [ ${STEP_NUMBER} -lt 1 ]; then
echo -n -e "Start the distribution upgrade in 5 seconds (Abort it with CTRL+C)\n"
echo -n -e " 5 ..."
sleep 1s
echo -n -e " 4 ..."
sleep 1s
echo -n -e " 3 ..."
sleep 1s
echo -n -e " 2 ..."
sleep 1s
echo -n -e " 1 ..."
sleep 1s
echo -n -e " GO!\n"
fi
```

```
if [ ${STEP_NUMBER} -lt 1 ]; then
echo -n "Check for existing lsb_release ..."
which lsb_release 1>/dev/null 2>&1
if [ $? -ne 0 ]; then
print_missing
echo -n -e "\n missing \"lsb\"-package: try to run \"zypper install lsb\" now\n"
zypper ${ZYPPEER_GLOBAL_OPTION} in lsb
if [ $? -ne 0 ]; then
echo -n " Error: zypper could not install the package \"lsb\" ..."
print_failure
exit 1
fi
fi
print_okay
save_step 1 "Check lsb_release"
fi
```

```
if [ ${STEP_NUMBER} -lt 2 ]; then
echo -n "Get openSUSE Version..."
SUSE_NAME=`lsb_release -d | head -n1 | cut -f2 | cut -f1 -d" "`
SUSE_VERSION=`lsb_release -r | cut -f2`
if [ -n "${SUSE_NAME}" -a -n "${SUSE_VERSION}" ]; then
echo -n -e "\n ${SUSE_NAME} ${SUSE_VERSION}"
print_okay
else
echo -n -e "\n No openSUSE Version found!"
print_aborted
exit
fi
set_variable_step "SUSE_NAME" "${SUSE_NAME}"
set_variable_step "SUSE_VERSION" "${SUSE_VERSION}"
save_step 2 "openSUSE Version"
fi
```

```
if [ ${STEP_NUMBER} -lt 3 ]; then
echo -n "Check internet connection to ${OPENSUSE_DOWNLOAD_URL} ..."
curl --progress-bar -s --retry 10 --connect-timeout 5 -f ${OPENSUSE_DOWNLOAD_URL} >/dev/null
if [ $? -ne 0 ]; then
```

openSUSE

```
echo -n -e "\n Error: Internet connection is not available!"
print_failure
exit 1
fi
save_step 3 "Internet connection"
print_okay
fi
```

```
if [ ${STEP_NUMBER} -lt 4 ]; then
if [ -n "${FORCE_SUSE_VERSION}" ]; then
echo -n "Forcing openSUSE version:"
SUSE_VERSION_LATEST="${FORCE_SUSE_VERSION}"
echo -n -e "\n ${SUSE_NAME} ${SUSE_VERSION_LATEST}"
else
echo -n "Try to figure out the latest openSUSE Version ..."
SUSE_VERSION_LATEST=`curl -s -f --list-only ${OPENSUSE_DOWNLOAD_URL}/openSUSE-current/iso/
2>/dev/null | sed -n -e 's/.*openSUSE-\([0-9]*\.[0-9]*\)-DVD-x86_64.iso<.*\1/mp`
echo -n -e "\n Latest openSUSE Version: ${SUSE_VERSION_LATEST}"
fi
set_variable_step "SUSE_VERSION_LATEST" "${SUSE_VERSION_LATEST}"
save_step 4 "Latest openSUSE Version"
print_okay
fi
```

```
if [ "${SUSE_VERSION}" = "${SUSE_VERSION_LATEST}" ]; then
echo -n -e "This is already the latest openSUSE version!"
print_aborted
exit 1
fi
```

```
if [ ${STEP_NUMBER} -lt 5 ]; then
CURRENT_DATE=`date '+%Y%m%d'`
echo -n "Create a tarball backup etc_${CURRENT_DATE}.tar.bz2 from /etc ..."
tar -cjSP --numeric-owner --atime-preserve -f etc_${CURRENT_DATE}.tar.bz2 /etc
if [ $? -ne 0 ]; then
echo -n -e "\n Error: Can not create a backup!"
print_failure
exit 1
fi
save_step 5 "Create a tarball backup etc_${CURRENT_DATE}.tar.bz2 from /etc"
print_okay
fi
```

```
if [ ${STEP_NUMBER} -lt 6 ]; then
echo -n -e "Move the directory /etc/zypp/repos.d to /etc/zypp/repos.d.upgrade ... \n"
mv -v /etc/zypp/repos.d /etc/zypp/repos.d.upgrade
if [ $? -ne 0 ]; then
echo -n -e "\n Error: Can not move the directory!"
print_failure
exit 1
fi
save_step 6 "Move the directory /etc/zypp/repos.d to /etc/zypp/repos.d.upgrade"
print_okay
fi
```

openSUSE

```
if [ ${STEP_NUMBER} -lt 7 ]; then
echo -n -e "Create new directory /etc/zypp/repos.d ...\n"
mkdir -v /etc/zypp/repos.d
if [ $? -ne 0 ]; then
echo -n -e "\n Error: Can not create the directory!"
print_failure
exit 1
fi
save_step 7 "Create new directory /etc/zypp/repos.d"
print_okay
fi
```

```
if [ ${STEP_NUMBER} -lt 8 ]; then
echo -n -e "Add new openSUSE standard repositories ...\n"
```

```
zypper --gpg-auto-import-keys ${ZYPPER_GLOBAL_OPTION} ar -t yast2 -f -c
http://download.opensuse.org/distribution/${SUSE_VERSION_LATEST}/repo/oss/ "openSUSE
${SUSE_VERSION_LATEST} OSS"
if [ $? -ne 0 ]; then
echo -n -e "\n Error: zypper could not add the standard repository! [openSUSE
${SUSE_VERSION_LATEST} OSS]"
print_failure
exit 1
fi
```

```
zypper --gpg-auto-import-keys ${ZYPPER_GLOBAL_OPTION} ar -t yast2 -f -c
http://download.opensuse.org/distribution/${SUSE_VERSION_LATEST}/repo/non-oss/ "openSUSE
${SUSE_VERSION_LATEST} NON-OSS"
if [ $? -ne 0 ]; then
echo -n -e "\n Error: zypper could not add the standard repository! [openSUSE
${SUSE_VERSION_LATEST} NON-OSS]"
print_failure
exit 1
fi
```

```
zypper --gpg-auto-import-keys ${ZYPPER_GLOBAL_OPTION} ar -t rpm-md -f -c
http://download.opensuse.org/update/${SUSE_VERSION_LATEST}/ "openSUSE
${SUSE_VERSION_LATEST} OSS Update"
if [ $? -ne 0 ]; then
echo -n -e "\n Error: zypper could not add the standard repository! [openSUSE
${SUSE_VERSION_LATEST} OSS Update]"
print_failure
exit 1
fi
```

```
zypper --gpg-auto-import-keys ${ZYPPER_GLOBAL_OPTION} ar -t rpm-md -f -c
http://download.opensuse.org/update/${SUSE_VERSION_LATEST}-non-oss/ "openSUSE
${SUSE_VERSION_LATEST} NON-OSS Update"
if [ $? -ne 0 ]; then
echo -n -e "\n Error: zypper could not add the standard repository! [openSUSE
${SUSE_VERSION_LATEST} NON-OSS Update3]"
print_failure
exit 1
fi
```

openSUSE

```
zypper ${ZYPPER_GLOBAL_OPTION} clean -a
if [ $? -ne 0 ]; then
echo -n -e "\n Error: zypper could not clean up the repository cache!"
print_failure
exit 1
fi
```

```
zypper ${ZYPPER_GLOBAL_OPTION} ref
if [ $? -ne 0 ]; then
echo -n -e "\n Error: zypper could not refresh the repository cache!"
print_failure
exit 1
fi
```

```
save_step 8 "Add new openSUSE standard repositories"
print_okay
fi
```

```
if [ ${STEP_NUMBER} -lt 9 ]; then
echo -n -e "Upgrade the openSUSE distribution now (The output of zypper is logged into upgrade-
opensuse.zypper-dup-output) ...\n"
zypper ${ZYPPER_GLOBAL_OPTION} dup -l 2>&1 | tee upgrade-opensuse.zypper-dup-output
if [ ${PIPESTATUS[0]} -ne 0 ]; then
echo -n -e "\n Error: zypper could not upgrade the openSUSE distribution!"
print_failure
exit 1
fi
```

```
save_step 9 "Upgrade the openSUSE distribution"
print_okay
fi
```

```
if [ ${STEP_NUMBER} -lt 10 ]; then
echo -n -e "Add back the formerly activated repositories and replace old openSUSE Version in the url
...\n"
IFS_BACKUP="${IFS}"
IFS=$'\n'
for REPOFILE in `grep -rl 'enabled=1' /etc/zypp/repos.d.upgrade | sort`
do
if [ -z `grep -rliE '/distribution/|/update/' ${REPOFILE}` ]; then
echo " Add /etc/zypp/repos.d/`basename ${REPOFILE}`"
cat ${REPOFILE} | sed -e
"s/^\(baseurl=.*)[0-9]\{2,\}\.[0-9]\{1,\}\(.*\)$\1${SUSE_VERSION_LATEST}\2/g"
>/etc/zypp/repos.d/`basename ${REPOFILE}`
fi
done
IFS="${IFS_BACKUP}"
```

```
zypper ${ZYPPER_GLOBAL_OPTION} ref
if [ $? -ne 0 ]; then
echo -n -e "\n Error: zypper could not refresh the repository cache!"
print_failure
```

openSUSE

```
exit 1
fi
```

```
save_step 10 "Add back the formerly activated repositories"
print_okay
fi
```

```
if [ ${STEP_NUMBER} -lt 11 ]; then
echo -n -e "Set temporarily zypper setting to allowed vendor change to prevent issues during
package update...\n"
cp -p /etc/zypp/zypp.conf /etc/zypp/zypp.conf.orig
sed -i 's/^[#\s]*solver.allowVendorChange.*\/solver.allowVendorChange = true/g' /etc/zypp/zypp.conf
save_step 11 "Set temporarily zypper setting to allowed vendor change"
print_okay
fi
```

```
if [ ${STEP_NUMBER} -lt 12 ]; then
echo -n -e "Check for old packages and try to reinstall those packages ...\n"
IFS_BACKUP="${IFS}"
IFS=$'\012'
OPENSUSE_MAJOR_VERSION="`lsb_release -r | cut -f2 | cut -d"." -f1`"
OPENSUSE_MINOR_VERSION="`lsb_release -r | cut -f2 | cut -d"." -f2`"
PACKAGE_LIST=""
if [ -f ./upgrade-opensuse.old-packages-output ]; then
rm upgrade-opensuse.old-packages-output
fi
for PACKAGES in `rpm -qa --qf '%{name}#{version}#{release}#{distribution}\n' | sort`
do
PKGDISTRO="`echo "${PACKAGES}" | cut -d'#' -f4`"
if [ -n "`echo "${PKGDISTRO}" | grep -v -E
"${OPENSUSE_MAJOR_VERSION}\.${OPENSUSE_MINOR_VERSION}|(none)|openSUSE|Tumbleweed`"
]; then
PKGNAME="`echo "${PACKAGES}" | cut -d'#' -f1`"
PKGVERSION="`echo "${PACKAGES}" | cut -d'#' -f2`"
PKGRELEASE="`echo "${PACKAGES}" | cut -d'#' -f3`"
echo " ${PKGNAME}-${PKGVERSION}-${PKGRELEASE} -> ${PKGDISTRO}" | tee -a upgrade-
opensuse.old-packages-output
PACKAGE_LIST="${PACKAGE_LIST} ${PKGNAME}"
fi
done
IFS="${IFS_BACKUP}"
```

```
zypper ${ZYPPER_GLOBAL_OPTION} in --oldpackage -n -l ${PACKAGE_LIST} 2>&1 | tee upgrade-
opensuse.zypper-reinstall-packages-output
if [ ${PIPESTATUS[0]} -ne 0 ]; then
echo -n -e "\n Error: zypper could not reinstall old packages!"
print_failure
exit 1
fi
```

```
save_step 12 "Check for old packages and try to reinstall those packages"
print_okay
fi
```

openSUSE

```
if [ ${STEP_NUMBER} -lt 13 ]; then
echo -n -e "Restore the zypper setting ...\n"
mv /etc/zypp/zypp.conf.orig /etc/zypp/zypp.conf
save_step 13 "Restore the zypper setting"
print_okay
fi

if [ ${STEP_NUMBER} -lt 14 ]; then
echo -n -e "Check for old packages that no longer exist in the repositories and remove them ...\n"
IFS_BACKUP="${IFS}"
IFS=$'\012'
OPENSUSE_MAJOR_VERSION=`lsb_release -r | cut -f2 | cut -d"." -f1`
OPENSUSE_MINOR_VERSION=`lsb_release -r | cut -f2 | cut -d"." -f2`
PACKAGE_LIST=""
if [ -f ./upgrade-opensuse.remove-old-packages-output ]; then
rm upgrade-opensuse.remove-old-packages-output
fi
for PACKAGES in `rpm -qa --qf '%{name}#%{version}#%{release}#%{distribution}\n' | sort`
do
PKGDISTRO=`echo "${PACKAGES}" | cut -d'#' -f4`
if [ -n "`echo "${PKGDISTRO}" | grep -v -E
"${OPENSUSE_MAJOR_VERSION}\.${OPENSUSE_MINOR_VERSION}|(none)|openSUSE|Tumbleweed`"
]; then
PKGNAME=`echo "${PACKAGES}" | cut -d'#' -f1`
PKGVERSION=`echo "${PACKAGES}" | cut -d'#' -f2`
PKGRELEASE=`echo "${PACKAGES}" | cut -d'#' -f3`
echo " ${PKGNAME}-${PKGVERSION}-${PKGRELEASE} -> ${PKGDISTRO}" | tee -a upgrade-
opensuse.remove-old-packages-output
PACKAGE_LIST="${PACKAGE_LIST} ${PKGNAME}-${PKGVERSION}-${PKGRELEASE}"
fi
done
IFS="${IFS_BACKUP}"

zypper ${ZYPPEER_GLOBAL_OPTION} rm --force-resolution ${PACKAGE_LIST} 2>&1 | tee upgrade-
opensuse.zypper-rm-packages-output
if [ ${PIPESTATUS[0]} -ne 0 ]; then
echo -n -e "\n Error: zypper could not remove old packages!"
print_failure
exit 1
fi

save_step 14 "Check for old packages that no longer exist in the repositories and remove them"
print_okay
fi
```

```
LIST_OLD_CONFIG_FILES=`find /etc -iname "*rpmnew" -o -iname "*rpmsave" | sort`
if [ -n "${LIST_OLD_CONFIG_FILES}" ]; then
echo -n -e "IMPORTANT: Please check manually the new or updated configuration files during
distribution upgrade ...\n"
echo -n -e " Remember:\n"
echo -n -e " - *.rpmnew = new configuration file that should be replaced the old configuration file.\n"
echo -n -e " - *.rpmsave = old configuration file that was backed up because the new configuration
file was replaced.\n"
echo ""
if [ -f upgrade-opensuse.list-new-and-old-config-files ]; then
```

openSUSE

```
rm upgrade-opensuse.list-new-and-old-config-files
fi
for LIST_OLD_CONFIG_FILE in ${LIST_OLD_CONFIG_FILES}
do
echo -n -e " ${LIST_OLD_CONFIG_FILE}\n" | tee -a upgrade-opensuse.list-new-and-old-config-files
done
echo ""
fi
```

```
echo -n -e "Finish! Please restart your computer and have a lot of fun! ;-)"
print_okay
```

SHA1: 38f9d02bb189bd21853f40c90491d00f8a111229 upgrade-opensuse.sh

Eindeutige ID: #1009

Verfasser: n/a

Letzte Änderung: 2020-10-10 12:44