



Dustin Morgan &lt;dustinpmorgan@gmail.com&gt;

---

## bash epel-script.sh fixed

---

Dustin Morgan <dustinpmorgan@gmail.com>  
To: Dustin Morgan <DustinPMorgan@gmail.com>

Mon, Jun 30, 2014 at 9:52 AM

```
#!/bin/bash
#
#
# White Listed EPEL Sync script 1.00
# June 24th, 2014 Dustin.Morgan@noaa.gov
#
# This script is designed to be fully automate syncing approved EPEL packages with
# Satellite
# 1. Sync white listed approved packages from EPEL to a sync location
# 2. Automatically process the packages by signing them and moving them to the
# designated channel in satellite
# 3. Email the group when a package is updated, or in the event of an error
# 4. Applies to all currently supported EPEL channels 6, 6Server, 7, 7Server
# 5. See README for more details
#

#STANDARD VARS
LOG=/var/log/rsync/EPEL_Sync-$(date +%Y-%m-%d).log
MAILME="/tmp/epel_updates.daily.$$"
ERROR="/tmp/epel_errors.email.$$"
MAILWHO="ncep.list.sisb-usa@noaa.gov"
CONFIG="/root/scripts/EPEL/config"
PID=/var/run/epel-rsync.pid

#STANDARD VARS
SATADMIN="nco-epel"
SATPASS=$(cat $CONFIG/nco-epel.pw)

#VAR FOR SIGNED RPMS
export PASSPHRASE=$(cat $CONFIG/sign.key)

#READ CONFIG
readarray -t CHANNELS < $CONFIG/array.channels
readarray -t SYNC_DIR < $CONFIG/array.repo-local
readarray -t EPEL_PATH < $CONFIG/array.mirror-path
readarray -t MIRRORS < $CONFIG/array.repo-mirrors
WHITELIST=$CONFIG/whiteList

#Check PID
if [ -f $PID ]; then kill -9 $(cat $PID) && echo $$ > $PID; else echo $$ > $PID; fi
```

```
#####
#      READY SET      GO
#
#
#      START    OF      CODE
#####

#####
##### This Section loops through the epel mirrors to sync the approved files
##### It will stop if there isn't an error or it exhausts the list of mirrors
#####

mirror_count=${#MIRRORS[@]}
chan_count=${#CHANNELS[@]}
index=0
chan=0
while [ "$index" -lt "$mirror_count" ]; do
    while [ "$chan" -lt "$chan_count" ]; do
        #RSYNC each channel recursively, preserve mod times, and use a rules file
        (whiteList.txt)
        rsync -rvtF --no-motd --log-file=$LOG --exclude-from=$WHITELIST --timeout=30
        "${MIRRORS[$index]}" "${EPEL_PATH[$chan]}" "${SYNCDIR[$chan]}"
        ((chan++))
    done
    if $(tail -3 $LOG | grep -q received); then
        break #if data received, exit loop
    fi
    chan=0
    ((index++))
done

#####
##### This section checks to see if the packages from EPEL are in Satellite
##### and creates a report and does if checksum OK it signs then verifies
##### them before pushing to Satellite
#####

chan=0
# Only when new data is rsync'd do this
if $(tail -3 $LOG | grep -q received); then

    while [ "$chan" -lt "$chan_count" ]; do

        REPORTS="${SYNCDIR[$chan]}/Reports" #set the reports dir for each chan

        #read packages from each chan into a report
        spacewalk-report channel-packages --where-channel_label="${CHANNELS[$chan]}" |awk -F
        ", " '{print $8}'|sed -e "1d"|sort > "$REPORTS/${CHANNELS[$chan]}";

        ls "${SYNCDIR[$chan]}"|grep rpm|rev|cut -c 5-|rev|sort > "$REPORTS/${CHANNELS[$chan]}".
    done
fi

```

```

avail";

    #compare whats in the folder to whats in the repo and create a report
    diff "$REPORTS/${CHANNELS[$chan]}" "$REPORTS/${CHANNELS[$chan]}.avail"|awk '{if ($1 ~
/^>/) print $0, "NEWER AVAILABLE"}' |cut -c 3- > "$REPORTS/${CHANNELS[$chan]}.report";

    comm -1 -2 "$REPORTS/${CHANNELS[$chan]}" "$REPORTS/${CHANNELS[$chan]}.avail"|awk
'{print $0, "OK"}'>> "$REPORTS/${CHANNELS[$chan]}.report";
    rm -f "$REPORTS/${CHANNELS[$chan]}.avail"; #remove temporay diff file

    #verify the checksums of the missing packages
    for new in $(grep NEWER "$REPORTS/${CHANNELS[$chan]}.report"|rev|cut -c 17-|rev);do
        md5sum ${SYNCDIR[$chan]}/${new}.rpm | md5sum -c > $REPORTS/checksum.md5; done

    #use the verified checksums to mark packages ready for signing
    grep OK $REPORTS/checksum.md5|awk -F ":" '{print $1}' > $REPORTS/checksum.good

    #sign each package that has a good checksum with the helpdesk gpg signature
    for pkg in $(cat $REPORTS/checksum.good); do
        export pkg=$pkg
        expect -c '
        spawn rpm --resign $::env(pkg)
        expect "Enter pass phrase:"
        send "$::env(PASSPHRASE)\r"
        expect eof ' ;
        rpm -K $pkg
    done > $REPORTS/key.report

    #upload the packages to satellite with verified gpg signatures on them
    # for pkg in $(grep "gpg OK" $REPORTS/key.report|awk -F ":" '{print $1}'); do
    # nohup rhnpush -u $SATADMIN -p $SATPASS -c ${CHANNELS[$chan]} $pkg --server localhost
    --force >> $ERROR;
    # done
    rm -f $REPORTS/key.report $REPORTS/checksum.good #remove the tmp reports
        ((chan++)) #move to the next channel
    done
fi
#####
##### This section does the emails
#####

if $(grep -q "+" $LOG); then
    subj="EPEL_UPDATE_SYNCED"
    echo "The following packages were updated today:" >> $MAILME;
    echo "" >> $MAILME;
    grep rpm $LOG|grep "+" |awk -F " " '{print $5}' >> $MAILME;
    echo "" >> $MAILME;
fi

if [ -s $ERROR ]; then
    subj="EPEL_RHNPUSH_ERROR"

```

```
        echo "" >> $ERROR;
fi

if $(tail -1 $LOG | grep -q error); then
    subj="EPEL_RSYNC_ERROR"
    echo "There was an issue rsync'ing EPEL today, better check it out." >> $ERROR;
    grep error $LOG|sort|uniq >> $ERROR;
    echo "" >> $ERROR;
fi

if [ -s $MAILME ] && [ ! -s $ERROR ]; then
    mail -s $subj $MAILWHO < $MAILME;
else
    cat $ERROR >> $MAILME
    mail -s $subj $MAILWHO < $MAILME
fi

rm -f $MAILME $ERROR $PID

--
```

