DSS-Script zur Kontrolle des DSS EdooSYS-Servers

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:              dss
# chkconfig: 345 99 05
# Required-Start:       postgresql
# Required-Stop:        postgresql
# Default-Start:        2 3 4 5
# Default-Stop:         0 1 6
# Short-Description:    Startet den DSS mit dem Systemstart
# Description: Java deamon script
### END INIT INFO
#
# A non-SUSE Linux start/stop script for Java daemons.
#
# Derived from -
# Home page: http://www.source-code.biz
# License:   GNU/LGPL (http://www.gnu.org/licenses/lgpl.html)
# Copyright 2006 Christian d'Heureuse, Inventec Informatik AG, Switzerland.
#
# History:
# 2018-01-22 Alexander Scheib: Several changes for running dss with openjdk 1.8 and opensuse leap 42.3
# 2013-07-01 Christian Schütz: Several changes for running dss
# 2012-10-30 Christian Schütz: Changes for running with java7
# 2011-11-23 Christian Schütz: Modifications for running ASV as daemon
# 2010-09-21 Josh Davis: Changed 'sudo' to 'su', fix some typos, removed unused variables
# 2009-03-04 Josh Davis: Ubuntu/Redhat version.
# 2006-06-27 Christian d'Heureuse: Script created.
# 2006-07-02 chdh: Minor improvements.
# 2006-07-10 chdh: Changes for SUSE 10.0.

### Passen Sie die folgenden Zeilen an Ihre Bedürfnisse an
 # Set this to your Java installation
JAVA_HOME=/usr/lib64/jvm/java-1.8.0-openjdk-1.8.0/jre

serviceNameLo="dss"                                     # service name with the first letter in lowercase
serviceName="DSS"                                       # service name
serviceUser="root"                                    # OS user name for the service
serviceGroup="nogroup"                                  # OS group name for the service
applDir="/opt/edoosys/server"                                       # home directory of the service application
serviceUserHome=""                                    # home directory of the service user
serviceLogFile="/var/log/$serviceNameLo-service.log"        # log file for StdOut/StdErr
maxShutdownTime=15                                      # maximum number of seconds to wait for the daemon to terminate normally
pidFile="/opt/edoosys/server/$serviceNameLo.pid"                        # name of PID file (PID = process ID number)
javaCommand="java"                                    # name of the Java launcher without the path
javaExe="$JAVA_HOME/bin/$javaCommand"                       # file name of the Java application launcher executable
```

```
javaArgs="-Declipse.ignoreApp=true -Dosgi.noShutdown=true -Xmx512m -Xms128m -Xverify:none -XX:PermSize=128m -XX:MaxPermSize=160m -XX:
+UseBiasedLocking -XX:+CMSClassUnloadingEnabled -XX:+UnlockDiagnosticVMOptions -XX:+UnsyncloadClass -jar startup.jar"
# arguments for Java launcher
javaCommandLine="$javaExe $javaArgs"                    # command line to start the Java service application
javaCommandLineKeyword="java"
# a keyword that occurs on the commandline, used to detect an already running service process and to distinguish it from others
##############################################################################################
#### Ab hier sind keine Änderungen mehr nötig
##############################################################################################
# Makes the file $1 writable by the group $serviceGroup.
function makeFileWritable {
  local filename="$1"
  touch $filename || return 1
  chgrp $serviceGroup $filename || return 1
  chmod g+w $filename || return 1
  return 0; }

# Returns 0 if the process with PID $1 is running.
function checkProcessIsRunning {
  local pid="$1"
  if [ -z "$pid" -o "$pid" == " " ]; then return 1; fi
  if [ ! -e /proc/$pid ]; then return 1; fi
  return 0; }

# Returns 0 if the process with PID $1 is our Java service process.
function checkProcessIsOurService {
  local pid="$1"
  if [ "$(ps -p $pid --no-headers -o comm)" != "$javaCommand" ]; then return 1; fi
  grep -q --binary -F "$javaCommandLineKeyword" /proc/$pid/cmdline
  if [ $? -ne 0 ]; then return 1; fi
  return 0; }

# Returns 0 when the service is running and sets the variable $pid to the PID.
function getServicePID {
  if [ ! -f $pidFile ]; then return 1; fi
  pid="$(<$pidFile)"
  checkProcessIsRunning $pid || return 1
  checkProcessIsOurService $pid || return 1
  return 0; }

function startServiceProcess {
  cd $applDir || return 1
  rm -f $pidFile
  makeFileWritable $pidFile || return 1
  makeFileWritable $serviceLogFile || return 1
  cmd="nohup $javaCommandLine >>$serviceLogFile 2>&1 & echo \$! >$pidFile"
  su -m $serviceUser -s $SHELL -c "$cmd" || return 1
  sleep 0.1
```

```
  pid="$(<$pidFile)"
  if checkProcessIsRunning $pid; then :; else
    echo -ne "\n$serviceName start failed, see logfile."
    return 1
  fi
  return 0; }

function stopServiceProcess {
  kill $pid || return 1
  for ((i=0; i<maxShutdownTime*10; i++)); do
    checkProcessIsRunning $pid
    if [ $? -ne 0 ]; then
      rm -f $pidFile
      return 0
      fi
    sleep 0.1
    done
  echo -e "\n$serviceName did not terminate within $maxShutdownTime seconds, sending SIGKILL..."
  kill -s KILL $pid || return 1
  local killWaitTime=15
  for ((i=0; i<killWaitTime*10; i++)); do
    checkProcessIsRunning $pid
    if [ $? -ne 0 ]; then
      rm -f $pidFile
      return 0
      fi
    sleep 0.1
    done
  echo "Error: $serviceName could not be stopped within $maxShutdownTime+$killWaitTime seconds!"
  return 1; }

function startService {
  getServicePID
  if [ $? -eq 0 ]; then echo -n "$serviceName is already running"; RETVAL=0; return 0; fi
  echo -n "Starting $serviceName   "
  startServiceProcess
  if [ $? -ne 0 ]; then RETVAL=1; echo "failed"; return 1; fi
  echo "started PID=$pid"
  RETVAL=0
  return 0; }

function stopService {
  getServicePID
  if [ $? -ne 0 ]; then echo -n "$serviceName is not running"; RETVAL=0; echo ""; return 0; fi
  echo -n "Stopping $serviceName   "
  stopServiceProcess
  if [ $? -ne 0 ]; then RETVAL=1; echo "failed"; return 1; fi
  echo "stopped PID=$pid"
```

```
  RETVAL=0
  return 0; }

function checkServiceStatus {
  echo -n "Checking for $serviceName:    "
  if getServicePID; then
   echo "running PID=$pid"
   RETVAL=0
  else
   echo "stopped"
   RETVAL=3
  fi
  return 0; }

function main {
  RETVAL=0
  case "$1" in
     start)                                  # starts the Java program as a Linux service
        startService
        ;;
     stop)                                   # stops the Java program service
        stopService
        ;;
     restart)                                # stops and restarts the service
        stopService && startService
        ;;
     status)                                 # displays the service status
        checkServiceStatus
        ;;
     *)
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
        ;;
     esac
  exit $RETVAL
}

main $1

main $1
```