

```
#!/bin/sh

#####

##### +-+--+--+--+ +-+--+--+--+ +-+#####

##### Globalteckz - Official Odoo Partners#####

##### +-+--+--+--+ +-+--+--+--+ +-+#####

#####

### BEGIN INIT INFO

# Provides: odoo-server

# Required-Start: $remote_fs $syslog

# Required-Stop: $remote_fs $syslog

# Should-Start: $network

# Should-Stop: $network

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Name: Odoo start/stop script for Ubuntu

# Author: Globalteckz Software Services

# Website: https://globalteckz.com

# Description: Using this script, we can start/stop/restart

# or check status of odoo server.

#

# Copyright(c)-2020-Present Globalteckz.

# All Rights Reserved.

### END INIT INFO

PATH=/bin:/sbin:/usr/bin
```

```

NAME=odoo-server

DESC=ODOO-SERVER

# Specify the daemon path for Odoo server.

# (Default for ODOO >=10: /opt/odoo/odoo-bin)

# (Default for ODOO <=9: /opt/odoo/openerp-server)

DAEMON=/opt/odoo/src/odoo-bin

CONFIGFILE="/etc/odoo-server.conf" # Specify the Odoo Configuration file path.


USER=odoo # Specify the user name (Default: odoo).


PIDFILE=/var/run/$NAME.pid # pidfile


# Additional options that are passed to the Daemon.

DAEMON_ARGS="-c $CONFIGFILE"


display() {

RED=$(tput setaf 1)

GREEN=$(tput setaf 2)

NORMAL=$(tput sgr0)

col=$(tput cols)

case "$#" in

1)

if [ $1 -eq 0 ] ; then

printf '%s%s%s' "$GREEN" $col "[ OK ] " "$NORMAL"

else

```

```
printf '%s%s%s' "$RED" $col "[FAIL] " "$NORMAL"

fi

;;

2)

if [ $1 -eq 0 ] ; then

echo "$GREEN* $2$NORMAL"

else

echo "$RED* $2$NORMAL"

fi

;;

*)

echo "Invalid arguments"

exit 1

;;

esac

}

if ! [ -x $DAEMON ] ; then

echo "Error in ODOO Daemon file: $DAEMON"

echo "Possible error(s):"

display 1 "Daemon File doesn't exists."

display 1 "Daemon File is not set to executable."

exit 0;

fi

if ! [ -r $CONFIGFILE ] ; then
```

```

echo "Error in ODOO Config file: $CONFIGFILE"

echo "Possible error(s):"

display 1 "Config File doesn't exists."

display 1 "Config File is not set to readable."

exit 0;

fi

if ! [ -w $PIDFILE ] ; then

touch $PIDFILE || echo "Permission issue: $PIDFILE" && exit 1

chown $USER: $PIDFILE

fi


# Function that starts the daemon/service

do_start() {

echo $1

check_status

procs=$?

if [ $procs -eq 0 ] ; then

start-stop-daemon --start --quiet --pidfile ${PIDFILE} \

--chuid ${USER} --background --make-pidfile \

--exec ${DAEMON} -- ${DAEMON_ARGS}

return $?

else

detailed_info "${DESC} is already Running !!!" $procs

exit 1

fi

```

```
}
```

```
# Function that stops the daemon/service
```

```
do_stop() {
```

```
echo $1
```

```
check_status
```

```
if [ $? -ne 0 ] ; then
```

```
start-stop-daemon --stop --quiet --pidfile ${PIDFILE}
```

```
return $?
```

```
else
```

```
display 0 "${DESC} is already Stopped. You may try: $0 force-restart"
```

```
exit 1
```

```
fi
```

```
}
```

```
get_pids(){
```

```
pids=$(ps -Ao pid,cmd | grep $DAEMON | grep -v grep | awk '{print $1}')
```

```
return $pids
```

```
}
```

```
# Function that checks the status of daemon/service
```

```
check_status() {
```

```
echo $1
```

```
# start-stop-daemon --status --pidfile ${PIDFILE}
```

```
status=$(ps -Ao pid,cmd | grep $DAEMON | grep -v grep | awk '{print $1}' | wc -l)
```

```
return $status
```

```
}
```

```
# Function that forcibly-stops all running daemon/service
```

```
force_stop() {
```

```
echo $1
```

```
pids=$(ps -Ao pid,cmd | grep $DAEMON | grep -v grep | awk '{print $1}')
```

```
if [ ! -z "$pids" ] ; then
```

```
kill -9 $pids
```

```
fi
```

```
return $?
```

```
}
```

```
detailed_info() {
```

```
procs=$2
```

```
if [ $procs -eq 1 ] ; then
```

```
display 0 "$1"
```

```
echo "FINE, ${procs} ${DESC} is Running."
```

```
echo "Details :"
```

```
pid=`cat $PIDFILE`
```

```
echo "Start Time : $(ps -p $pid -wo lstart=)"
```

```
echo "Total UpTime: $(ps -p $pid -wo etime=)"
```

```
echo "Process ID : ${pid}"
```

```

echo ""

else

display 1 "WARNING !!!"

display 1 "${procs} ${DESC}s are Running !!!"

pids=$(ps -Ao pid,cmd | grep $DAEMON | grep -v grep | awk '{print $1}')

echo "Details :"

echo -n "Process IDs : "

echo $pids

# echo $pids | tr ' ' ,

echo "In order to fix, Hit command: $0 force-restart"

echo ""

fi

}

case "$1" in

start)

do_start "Starting ${DESC} "

display $?

;;

stop)

do_stop "Stopping ${DESC} "

display $?

;;

status)

check_status "Current Status of ${DESC}:"

procs=$?

```

```
if [ $procs -eq 1 ] ; then
detailed_info "RUNNING" $procs
elif [ $procs -eq 0 ] ; then
display 1 "STOPPED"
else
detailed_info "" $procs
fi
;;

restart|reload)
do_stop "Stopping ${DESC} "
display $?
sleep 1
do_start "Starting ${DESC} "
display $?
;;

force-restart)
force_stop "Forcely Restarting ${DESC} "
sleep 1
do_start "Starting ${DESC} "
display $?
;;

force-stop)
force_stop "Forcely Stopping all running ${DESC} "
display $?
;;
```

```
cs)
```

```
ps -Ao pid,cmd | grep $DAEMON | grep -v grep | awk '{print $1}' | wc -l
```

```
;;
```

```
*)
```

```
display 1 "Usage: $0
```

```
{start|stop|restart/reload|status|force-restart|force-stop}"
```

```
exit 1
```

```
;;
```

```
esac
```

```
exit 0
```