

## Description - Genserver process systemd configuration

GenServer server node: amsdc2-n-sv0023

### Script path

```
[root@amsdc2-n-sv0023 ~]# ls -l /glb/eu/epe/data/genrem/genserver/production/server/app/genserver.sh
-rwxrwxr-x 1 s_genrem g_ortc00 1600 Aug  3 10:10
/glb/eu/epe/data/genrem/genserver/production/server/app/genserver.sh
[root@amsdc2-n-sv0023 ~]#
```

### Check on open port

```
[root@amsdc2-n-sv0023 ~]# netstat -an |grep <port #>
tcp        0      0 0.0.0.0:<port #>      0.0.0.0:*          LISTEN

[root@amsdc2-n-sv0023 ~]# netstat -an |grep "3001"    GenServer production port, should always listen
[root@amsdc2-n-sv0023 ~]# netstat -an |grep "3000"    GenServer development port, listens during development
[root@amsdc2-n-sv0023 ~]# netstat -an |grep "8081"    WebSocket production port, should always listen
[root@amsdc2-n-sv0023 ~]# netstat -an |grep "8080"    WebSocket development port, listens during development
```

```
[root@amsdc2-n-sv0023 ~]# cat /lib/systemd/system/genserver.service
```

[Unit]

Description=description: Startup/Shutdown genserver process

After=network.target syslog.target

[Service]

User=s\_genrem

Group=g\_ortc00

Type=forking

RemainAfterExit=yes

ExecStop=/glb/eu/epe/data/genrem/genserver/production/server/app/genserver.sh stop

ExecStart=/glb/eu/epe/data/genrem/genserver/production/server/app/genserver.sh start

Restart=always

[Install]

WantedBy=default.target

```
[root@amsdc2-n-sv0023 ~]#
```

```
systemctl daemon-reload
```

```
systemctl status genserver.service
```

```
systemctl start genserver.service
```

```
systemctl status genserver.service
```

```
systemctl enable genserver.service
```

ORTEC Perforce specification:

//Logistiek/NAM/GENREM/Release/GenServer/Server/systemd start-stop script/genserver.sh#2

```
[root@amsdc2-n-sv0023 ~]# cat /glb/eu/epe/data/genrem/genserver/production/server/app/genserver.sh
#!/usr/bin/bash
#
# Purpose :- Starting/Stopping the GenServer process through systemd service
#           Systemd file - /lib/systemd/system/genserver.service

task="$1"

# logger function
LOG_FACILITY="local3.info"
logmaster ()
{
    logger -t genserver -p $LOG_FACILITY $1
}

# stop the genserver process
function genserver_stop()
{
    # checking the genserver process status.
    ps -ef | egrep -w 'node|main.js' | grep -v grep
    if [ "$?" -eq 0 ]
    then
        # killing the genserver process if it is running
        PIDOF=$(ps -ef | egrep -w 'node|main.js' | grep -v grep | awk '{print $2}')
        kill -9 "$PIDOF"

        # checking the genserver process status
        ps -ef | egrep -w 'node|main.js' | grep -v grep
        if [ "$?" -eq 1 ]
        then
            logmaster "INFO: genserver process has been stopped"
        else
            logmaster "INFO: genserver process could not stop"
        fi
    fi
}

# start the genserver process
function genserver_start()
{
    # checking the genserver process status.
    ps -ef | egrep -w 'node|main.js' | grep -v grep
    if [ "$?" -eq 0 ]
    then
        logmaster "genserver process is already running in the server"
    else
        cd /glb/eu/epe/data/genrem/genserver/production/server/app
        source genserver.environment
        #su s_genrem -c "/glb/eu/epe/data/genrem/genserver/node-distribution/bin/node main.js >/dev/null 2>&1"
        nohup /glb/eu/epe/data/genrem/genserver/node-distribution/bin/node main.js > /dev/null 2>&1 &
    fi
}

# main
if [ "$task" == "start" ]
then
    genserver_start
elif [ "$task" == "stop" ]
then
    genserver_stop
fi
##
```