Starting and stopping

After you activate the license, start WCS using the command:

```
service webcallserver start
```

Stopping the server is done with the command

```
service webcallserver stop
```

ATTENTION, server start can take up to 1 minute.

Here are a few ways to make sure the server has started and is ready to work:

Make sure the server process is running.

```
ps aux | grep WebCallServer
```

The console should display two processes: WCS Core (20850 on the example below) and WCS Manager (20806 on the example below):

```
[root@localhost tmp]# ps aux | grep WebCallServer
root     20806  0.4 45.1 522148 236512 pts/0   Sl   10:56   0:59 java -Dloader.path=/usr/local
/FlashphonerWebCallServer-5.0.2993/lib/tbs-commons.jar,/usr/local/FlashphonerWebCallServer-5.0.2993/lib
/wcs_manager-1.0.jar -Dcom.flashphoner.fms.AppHome=/usr/local/FlashphonerWebCallServer -jar /usr/local
/FlashphonerWebCallServer-5.0.2993/lib/wcs_manager-1.0.jar -Xmx1200M -Djava.net.preferIPv4Stack=true -Dcom.sun.
management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.
port=50999 -XX:ErrorFile=/usr/local/FlashphonerWebCallServer/logs/error%p.log -Dcom.flashphoner.fms.AppHome=/usr
/local/FlashphonerWebCallServer -Djava.library.path=/usr/local/FlashphonerWebCallServer/lib/so:/usr/local
/FlashphonerWebCallServer/lib -cp /usr/local/FlashphonerWebCallServer/lib/* com.flashphoner.server.Server
root     20850  0.0 16.5 1567800 86636 pts/0   Sl   10:57   0:06 java -Xmx1200M -Djava.net.preferIPv4Stack=true
-Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.
jmxremote.port=50999 -XX:ErrorFile=/usr/local/FlashphonerWebCallServer/logs/error%p.log -Dcom.flashphoner.fms.
AppHome=/usr/local/FlashphonerWebCallServer -Djava.library.path=/usr/local/FlashphonerWebCallServer/lib/so:/usr
/local/FlashphonerWebCallServer/lib -cp /usr/local/FlashphonerWebCallServer/lib/* com.flashphoner.server.Server
root     22053  0.0  0.1  61152   732 pts/0    R+   14:51   0:00 grep WebCallServer
[root@localhost tmp]#
```

2. Make sure the server process listens the main ports.

```
netstat -nlp | grep java

[root@localhost tmp]# netstat -nlp | grep java
tcp        0      0 0.0.0.0:1098  0.0.0.0:* LISTEN 20850/java
tcp        0      0 0.0.0.0:8080  0.0.0.0:* LISTEN 20850/java
tcp        0      0 0.0.0.0:8081  0.0.0.0:* LISTEN 20850/java
tcp        0      0 0.0.0.0:50999 0.0.0.0:* LISTEN 20850/java
tcp        0      0 0.0.0.0:8443  0.0.0.0:* LISTEN 20850/java
tcp        0      0 :::9091       :::*      LISTEN 20806/java
tcp        0      0 :::1099       :::*      LISTEN 20806/java
tcp        0      0 :::2000       :::*      LISTEN 20806/java
udp        0      0 0.0.0.0:1935  0.0.0.0:*        20850/java
```

If you used a standard number of ports, you should see ports 8080 (Websockets) and 1935 (RTMP) as well as other port you configured for the WCS5 server in the netstat listened ports output.

3. Make sure the WCS5 server writes the main server log of the Core.

```
tail -f /usr/local/FlashphonerWebCallServer/logs/server_logs/flashphoner.log
```

The log should display information about settings the server started with.

For example:

```
15:59:37,378 INFO        Config - main LOAD_BALANCING_SERVERS: null
15:59:37,378 INFO        Config - main STREAM_MODE_UDP: true
15:59:37,379 INFO        Config - main LOAD_TOOL_ENABLED: false
15:59:37,379 INFO        Config - main CLI_ENABLED: false
15:59:37,379 INFO        Config - main RMI_PORT: 1098
15:59:37,379 INFO        Server - main Starting server...
15:59:37,492 INFO        KeepAliveManager - KeepAliveManager Start keepAlive thread KeepAliveManager
15:59:37,614 INFO        Server - main Listening RTMFP on 1935 port, bufferSize: 64000
15:59:37,750 INFO        Server - main Listening WebSocket on 8080 port, bufferSize: 64000
15:59:37,759 INFO        Server - main Listening WebSocket Ssl on 8443 port, bufferSize: 64000
```

Logs should react on connection of web clients. If that does not happen during testing, make sure the server process is running and the web client is configured properly to connect this particular server. See the Troubleshooting section for additional information.

4. Make sure WCS does write the main server log of the Manager module

```
tail -f /usr/local/FlashphonerWebCallServer/logs/flashphoner_manager.log

15:59:36,277 INFO  gerCommandLineRunner - main Starting server node
15:59:36,278 INFO  ServerProcess - main Starting server node
15:59:36,293 INFO  ServerProcess - main Arg: java
15:59:36,294 INFO  ServerProcess - main Arg: -Xmx1200M
15:59:36,294 INFO  ServerProcess - main Arg: -Djava.net.preferIPv4Stack=true
15:59:36,294 INFO  ServerProcess - main Arg: -Dcom.sun.management.jmxremote.ssl=false
15:59:36,294 INFO  ServerProcess - main Arg: -Dcom.sun.management.jmxremote.authenticate=false
15:59:36,294 INFO  ServerProcess - main Arg: -Dcom.sun.management.jmxremote.port=50999
15:59:36,294 INFO  ServerProcess - main Arg: -XX:ErrorFile=/usr/local/FlashphonerWebCallServer/logs/error%p.log
15:59:36,294 INFO  ServerProcess - main Arg: -Dcom.flashphoner.fms.AppHome=/usr/local/FlashphonerWebCallServer
15:59:36,294 INFO  ServerProcess - main Arg: -Djava.library.path=/usr/local/FlashphonerWebCallServer/lib/so:/usr
/local/FlashphonerWebCallServer/lib
15:59:36,294 INFO  ServerProcess - main Arg: -cp
15:59:36,295 INFO  ServerProcess - main Arg: /usr/local/FlashphonerWebCallServer/lib/*
15:59:36,295 INFO  ServerProcess - main Arg: com.flashphoner.server.Server
15:59:36,313 INFO  Manager - main Started Manager in 49.416 seconds (JVM running for 51.706)
```

If the server process is running and logs have no error, this means the WCS server is ready to work and you can start testing it.

Two ways to start WCS

Starting the server is performed with this command:

```
service webcallserver start
```

Besides, you can start the server using:

```
cd /usr/local/FlashphonerWebCallServer/bin
./webcallserver start
```

Environment variables and parameters of the start are set in the setenv.sh script. In this script you can se additional parameters for WCS Core and WCS Manager. Also, here you can set the parameter that prevent memory leaks on multi-CPU systems:

```
MALLOC_ARENA_MAX=4
```

Technically, starting goes as follows: first, the WCS Manager process starts, then that process launches the child process, WCS Core.

Starting with stdout output

In some cases, for example, if the server won't start and does not produce any errors, you may need to start the server with direct logging to the 'stdout' console. Direct output to stdout cannot be used in production, because the server will be stopped if the console is closed or the SSH connection is lost. That is why we recommend using stdout output only for debug purposes.

To start the server in this mode, use the following command:

```
cd /usr/local/FlashphonerWebCallServer/bin
./webcallserver start standalone
```

In this case, logs of the root process, WCS Manager, are printed directly to the console, and the server will be stopped if you press Ctrl+C.
To configure output of logs of the child process, WCS Core, use thenode.enable_stdout=true setting.

JVM parameters

Parameters are set in the setenv.sh file.

Here yoiu can add any specific startparameters using the following varaibles:

WCS_JAVA_OPTS- the list of options for WCS Core

WCS_MANAGER_OPTS- the list of options for WCS Manager

Even though, one process is a parent of the other one, these processes are completely independent, and options set for the parent do not influence the child process. For example, if both for the parent and the child processes the -Xmx512M option is set, each of the processes will be executed isolated and have 512 megabytes of heap memory.