

Запуск и остановка

- [Способы запуска](#)
 - [Запуск с выводом в stdout](#)
 - [Параметры JVM](#)

После [активации лицензии](#) запустите WCS командой

```
service webcallserver start
```

Остановка сервера выполняется командой

```
service webcallserver stop
```

ВНИМАНИЕ, запуск сервера может занимать около 1 минуты.

Есть несколько способов убедиться в том, что сервер стартовал и готов к работе:

1. Убедитесь, что серверный процесс работает.

```
ps aux | grep WebCallServer
```

В консоли должно отобразиться два процесса: WCS Core (20850 на примере ниже) и WCS Manager (20806 на примере ниже):

```
[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. Убедитесь, что серверный процесс слушает основные порты.

```
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
```

Если вы использовали стандартный набор портов, в выводе слушаемых портов, командой netstat, должны появиться порты 8080 (Websockets) и 1935 (RTMP) или другие порты, которые вы сконфигурировали для WCS5 сервера.

3. Убедитесь, что WCS5 сервер пишет основной серверный лог Ядра.

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

В логе должна появиться информация о настройках, с которыми стартовал сервер.

Пример:

```
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
```

Логи должны реагировать на соединения web-клиентов. Если [при тестировании](#) этого не происходит, убедитесь что серверный процесс работает и web-клиент корректно настроен на соединение именно с этим сервером. См. раздел [Возможные неисправности и способы их устранения](#) для получения дополнительной информации.

4. Убедитесь что WCS пишет основной серверный лог Административного модуля

```
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)
```

Если серверный процесс запущен и в логах нет ошибок, это означает что WCS сервер готов к работе и можно начинать тестирование.

Способы запуска

Запуск сервера осуществляется командой:

```
service webcallserver start
```

Кроме этого запуск может быть осуществлен командой:

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

Подготовка запуска и переменных окружения происходит в скрипте `setenv.sh`. В этом скрипте могут быть заданы дополнительные параметры для WCS Core и WCS Manager. Кроме того, здесь же задается параметр, предотвращающий утечки памяти на многопроцессорных системах:

```
MALLOC_ARENA_MAX=4
```

Технически запуск происходит следующим образом: сначала запускается процесс WCS Manager, потом этот процесс запускает дочерний процесс WCS Core.

Запуск с выводом в stdout

В некоторых случаях, например когда сервер не стартует и не выдает никаких ошибок, требуется запуск с прямым логгированием в консоль 'stdout'. Прямой вывод в stdout не может использоваться в production т.к. при закрытии консоли или потере SSH-соединения сервер будет остановлен. Поэтому рекомендуется использовать вывод stdout только в целях отладки.

Для того чтобы запустить сервер в данном режиме, необходимо выполнить команду:

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

В этом случае логи корневого процесса WCS Manager будут выводиться прямо в консоль и при нажатии Ctrl+C сервер будет остановлен. Для того чтобы настроить вывод логов дочернего процесса WCS Core, воспользуйтесь [настройкой node.enable_stdout=true](#).

Параметры JVM

Параметры задаются в файле `setenv.sh`.

Здесь вы можете добавить любые опции запуска в следующие переменные:

WCS_JAVA_OPTS- список опций для WCS Core

WCS_MANAGER_OPTS- список опций для WCS Manager

Несмотря на то что один процесс является родителем другого, процессы полностью изолированы и опции, заданные для родителя никак не влияют на дочерний процесс. Например, если для родителя и для дочернего процесса установлена опция `-Xmx512M`, каждый из процессов будет выполняться изолировано и иметь 512 мегабайт heap-памяти.