

Web Call Server 5.1 - EN

- [Release notes](#)
 - [Web SDK release notes](#)
 - [iOS SDK release notes](#)
 - [Android SDK release notes](#)
 - [Android SDK 1.1 release notes](#)
 - [WCS 5.0 release notes](#)
- [Quick deployment and testing of the server](#)
- [WCS 5.0 to WCS 5.1 update](#)
- [Installing and testing](#)
 - [Requirements and prerequisites](#)
 - [License types and server requirements](#)
 - [WCS installation and activation](#)
 - [Starting and stopping](#)
 - [Testing using the web interface](#)
 - [Installation troubleshooting](#)
 - [Checking for updates](#)
- [Working with the server](#)
 - [Architecture](#)
 - [Server core \(WCS Core\)](#)
 - [Manager module \(WCS Manager\)](#)
 - [Main commands](#)
 - [Core settings](#)
 - [Settings file flashphoner.properties](#)
 - [Settings file loadbalancing.xml](#)
 - [Settings file log4j.properties](#)
 - [Settings file watchdog.properties](#)
 - [Settings file watchdog.log4j.properties](#)
 - [Settings file rtsp.auth](#)
 - [Key store wss.jks](#)
 - [File flashphoner.serverid](#)
 - [Certificate myflashphoner-ca](#)
 - [WCS.version](#)
 - [Settings file wcs-core.properties](#)
 - [SDP settings files](#)
 - [Settings file database.yml](#)
 - [Manager settings](#)
 - [SSL certificates management](#)
 - [Websocket SSL](#)
 - [REST HTTPS](#)
 - [Receiving and importing Let's Encrypt SSL certificate](#)
 - [Logging](#)
 - [WCS Core logs](#)
 - [WCS Manager logs](#)
 - [Watchdog logs](#)
 - [Monitoring](#)
 - [Using SSH tunnel](#)
 - [Connecting from JConsole](#)
 - [Connecting from Visual VM](#)
 - [Connecting from Java Mission Control](#)
 - [Watchdog availability checking system](#)
 - [Load and resource usage information](#)
 - [Information about errors and configuration parameters](#)
 - [Streams monitoring with web interface](#)
 - [Calls monitoring with web interface](#)
 - [System status monitoring with web interface](#)
 - [Core command line interface](#)
 - [Using and accessing core command line](#)
 - [Complete list of core CLI commands](#)
 - [Public key authorization](#)
 - [Users management](#)
 - [Applications management](#)
 - [Server settings management](#)
 - [Manager command line interface](#)
 - [Using and accessing manager command line](#)
 - [Complete list of manager CLI commands](#)
 - [Server shutdown and startup](#)
 - [Network traffic analysis](#)
 - [Websocket](#)
 - [REST / HTTP](#)
 - [SIP](#)
 - [WebRTC](#)
 - [RTMFP](#)
 - [RTP](#)
 - [RTSP / RTP](#)
 - [RTMP](#)
 - [HLS](#)
 - [Inner RMI traffic](#)
 - [Diagnostics and troubleshooting](#)
 - [Preparing an error report](#)

- Possible problems and how to solve them
 - Accessory tools
 - How to get license number
 - CPU load investigation
- Server tuning recommendations
- Server functions
- Streaming video functions
 - Stream capturing and publishing to the server
 - From a web camera in a browser via WebRTC
 - From the computer screen (screen sharing) in a browser via WebRTC
 - From an HTML5 Canvas element (whiteboard) in a browser via WebRTC
 - By means of Flash Player via RTMP
 - Using RTMP encoder
 - Using Adobe FMLE
 - Using ffmpeg
 - Using OBS Studio
 - Using Wirecast
 - From an IP camera via RTSP
 - From another server via RTMP
 - RTMP stream capturing by re-publishing from another RTMP server
 - From another WCS server via WebRTC
 - From an Android mobile app via WebRTC
 - From an iOS mobile app via WebRTC
 - Capturing VOD from a file
 - Automatic streams capture on server start
 - Stream capture from dump file
 - Managing camera and microphone
 - Bitrate management when capturing WebRTC stream in browser
 - Captured stream management
 - Stream recording
 - Stopping the video stream on the server side
 - Taking a PNG snapshot of the stream
 - Stream mixer
 - Stream decoding
 - Stream transcoding
 - Stream watermarking
 - FPS filter
 - Using AAC codecs
 - Playing a video stream from the server
 - In a browser via WebRTC
 - In a browser using Flash Player via RTMP
 - In a browser via MSE
 - In a browser via Websocket + Canvas, WSPlayer
 - In a browser via HLS
 - In an Android mobile application via WebRTC
 - In an iOS mobile application via WebRTC
 - In a player via RTSP
 - In a player via RTMP
 - In a browser with Delight Player
 - Publishing and playing stream via WebRTC over TCP
 - Republishing a video stream
 - To another RTMP server
 - Republishing to Youtube
 - Republishing to Facebook
 - Republishing to Wowza
 - Republishing to WCS
 - Republishing to Azure Media Services
 - To another WCS server via WebRTC
- Working with chat rooms
- Streaming video CDN functions
 - CDN 1.0
 - CDN 2.0
 - CDN 2.1
- Streaming video and SIP integration functions
 - Stream capturing from a SIP call
 - Redirecting a SIP call to a stream (SIP as Stream function)
 - Republishing a SIP call to an RTMP stream to the given server (SIP as RTMP function)
 - Redirecting a stream to a SIP call using /call/inject_stream
 - Redirecting an audio file to a SIP call using /call/inject_sound
- WebRTC-SIP gateway functions
 - SIP calls in a WebRTC-compatible browser
 - SIP calls using Android SDK
 - SIP calls using iOS SDK
 - SIP complex functions
 - A call between two browsers made via the SIP server
 - A call to a SIP conference
 - A call to a mobile phone via the SIP server
 - Known issues
- Working through Firewall
 - HAProxy
 - TURN server
- Load testing
 - Server settings tests

- Load testing using another server
 - Load testing using WebRTC pulling
 - Stress test for SIP calls
 - Mixer load testing
- Load balancing
 - Balancer architecture
 - Configuring and starting the balancer
- Web SDK
- iOS SDK
- Android SDK
 - Android SDK 1.0
 - Android SDK 1.1
- Raw WebSocket API
- REST API
 - API overview
 - API methods
 - Returned objects
 - Examples
- REST Hooks
 - REST Methods
 - Invoking a REST method
 - Authorization on backend
 - Four types of REST methods
 - Type 1 - the connect method
 - Type 2 - the direct invoke
 - Type 3 - the event
 - Type 4 - the incoming call
 - The list of methods and their parameters
 - restClientConfig object description
 - Controlling REST methods
 - The match between client invocations and REST methods
 - REST methods object fields
 - Event statuses
 - Data exchange - OnDataEvent
 - Error handling
 - Using REST hook to authorize user by domain
- WCS in Amazon EC2
- Billing