

Web Call Server 5.2 - EN

- Quick deployment and testing of the server
- WCS update to 5.2
- 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 and updating
- Removing
- Working with the server
 - Architecture
 - Server core (WCS Core)
 - 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
 - SSL certificates management
 - Websocket SSL
 - HTTPS
 - Receiving and importing Let's Encrypt SSL certificate
 - Logging
 - WCS Core logs
 - Monitoring
 - Using SSH tunnel
 - Connecting from JConsole
 - Connecting from Visual VM
 - Connecting from Java Mission Control
 - WCS integration to Prometheus
 - WCS to Zabbix integration
 - Load and resource usage information
 - Information about errors and configuration parameters
 - Stream parameters monitoring with REST API
 - Server health check
 - Centralized stream statistics and CDN events collection to MySQL DB
 - Centralized server data logging to ClickHouse DB
 - Command line interface
 - Using and accessing command line
 - Access restriction from certain interfaces
 - Complete list of CLI commands
 - Public key authorization
 - Users management
 - Applications management
 - Server settings management
 - CLI v 2
 - Using and accessing CLI v 2
 - Access restriction to CLI v2 from certain interfaces
 - Complete list of CLI v 2 commands
 - Streams management
 - Connections management
 - CDN node management
 - Network traffic analysis
 - Websocket
 - REST / HTTP
 - SIP
 - WebRTC
 - RTMFP
 - RTP
 - RTSP / RTP
 - RTMP
 - HLS
 - Server domain name support as external address
 - HTTP interfaces access restriction
 - Websocket connections restriction by domain
 - Stream publishing and playback restriction by name
 - Websocket client URI configuration
 - Media ports management

- Server ports configuration to accept client connections
- Thread pools tuning
- Diagnostics and troubleshooting
 - Preparing an error report
 - Possible problems and how to solve them
 - Accessory tools
 - How to get license number
 - CPU load investigation
 - Stack trace logging
 - DNS testing and caching
- Server tuning recommendations
- Memory management in Java
- Server performance testing
- Before moving to production
- 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
 - RTP stream publishing via RTSP
 - MPEG-TS RTP stream publishing
 - WebRTC publishing via WHIP
 - Automatic streams capture on server start
 - RTSP-interleaved stream capture from dump file
 - Managing camera and microphone
 - Bitrate management when capturing WebRTC stream in browser
 - Key frames management while capturing WebRTC in browser
 - Published stream normalizing
 - Jitter buffer and frames collection in stream published
 - Captured stream management
 - Stream recording
 - Stopping the video stream on the server side
 - Taking a PNG snapshot of the stream
 - Stream decoding
 - Stream transcoding
 - Stream watermarking
 - FPS filter
 - Using AAC codecs
 - WebRTC stream picture rotation
 - Minimal publishing bitrate control
 - Decoded frames interception and handling
 - Decoded frames interception and handling with OpenCV
 - Server audio processing
 - Injecting one stream into another
 - 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
 - In a browser via WebRTC ABR
 - Stream availability for playback
- Publishing and playing stream via WebRTC over TCP
- Publisher and player channel quality control
- WebRTC traffic encryption hardware acceleration
- DTLS support for WebRTC streaming
- WebRTC RTP bundle support
- IPv6 support for WebRTC
- Websocket traffic proxying for WebRTC publishing/playing
- H264 encoding profiles management
- Stream event passing to subscribers
- 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
 - Republishing to AWS MediaLive
 - Republishing to Periscope, Twitch, Telegram etc
 - To another WCS server via WebRTC
- Working with chat rooms
- Stream mixer functions
 - Stream mixer
 - Real-time stream mixer with MCU functions
 - Custom mixer layout configuration with a special markup language
- SFU functions with Simulcast
- Streaming video CDN functions
 - CDN 1.0
 - CDN 2.0
 - CDN 2.1
 - CDN 2.2
 - CDN 2.3
 - CDN 2.4
 - CDN 2.5
 - CDN 2.6
 - Choosing a protocol to transfer media data within CDN
 - DTLS support for WebRTC between CDN servers
 - Network traffic flow between CDN servers behind NAT
 - Stream publishing to CDN servers restriction
 - CDN nodes name resolution
- 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)
 - Republishing incoming SIP call to a stream
 - Redirecting a stream to a SIP call using /call/inject_stream
 - Redirecting an audio file to a SIP call using /call/inject_sound
 - SIP call stream raw audio recording
- WebRTC-SIP gateway functions
 - DTMF support
 - 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/RTMP pulling
 - Stress test for SIP calls
 - Mixer load testing
 - Scripts to test a maximum number of WebRTC publishers/subscribers
- Load balancing
 - Load balancer setup based on HAProxy
 - Obsolete internal balancer functions
 - Balancer architecture
 - Configuring and starting the balancer
- Web SDK
- SFU 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
- Sending custom error message to a client
- Using REST hook to authorize user by domain
- WCS in Amazon EC2
 - AWS load balancer with auto scale quick setup
 - coturn setup in AWS EC2 instance
 - Deploying WCS with CloudFormation
- WCS on Digital Ocean
- WCS on Google Cloud Platform
 - GCP load balancer with autoscale quick setup
- WCS in Yandex.Cloud
- WCS in Docker
- WCS in Equinix Metal (ex Packet.Net)
- WCS in WSL 2
- Billing
- Technical support