# From the computer screen (screen sharing) in a browser via **WebRTC**

- Overview
  - Supported platforms and browsers
  - Operation flowchart
- Quick manual on testing
  - · Capturing a video stream from the computer screen and preparing for publishing
- Call flow • To developer
  - Extension for Google Chrome with publication in Chrome Store
    - Create a Google account
    - Register as Chrome Web Store Developer
    - Customization for your domain
    - Package the extension
    - Publish the extension
    - Extension Inline Installation
  - Configuring the client • Media source parameters
    - Parameter list
  - System sound capture in Chrome browser
  - Capture source (screen or window) management in Firefox browser
     Screen sharing without extension
    - - Firefox browser
      - Chromium based browsers
      - Safari browser in MacOS
      - Known limits
- Known issues

# **Overview**

## Supported platforms and browsers

	Chrome	Firefox
Windows	+	+
Mac OS	+	+

**Operation flowchart** 

#### Browser 1 - Publisher



Browser 2 - Player

- 1. The browser establishes a connection to the server via the Websocket protocol and sends the publish command.
- 2. The browser captures the screen and sends a WebRTC stream to the server.
- 3. The second browser establishes a connection also via Websokcet and sends the play command.
- 4. The second browser receives the WebRTC stream and plays the stream on the page.

# Quick manual on testing

## Capturing a video stream from the computer screen and preparing for publishing

1. For the test we use the demo server at demo.flashphoner.com and the Screen Sharing web application in the Chrome browser

https://demo.flashphoner.com/client2/examples/demo/streaming/screen-sharing/screen-sharing.html

			Scree	en Sharing		
	Chrome e	extension Insta	II Now			
F	PS	Width	Height			
30		640	480			
W/	o extension?	Use Mic	?	Mic		
	4		Defa	ault - Microphone 🔹		
		My S	creen		<b>→</b>	
wss://der	no.flashphone	r.com:8443/3a43	7ca9	Start	_	Preview

2. Click the "Start" button. The browser asks for permission to access the screen, and screen capturing starts, then the stream is publishing:



#### 3. Make sure the stream is sent to the server and the system operates normally in chrome://webrtc-internals



4. Open Two Way Streaming in a new window, click Connect and specify the stream id, then click Play.



5. Playback diagrams in chrome://webrtc-internals



# Call flow

Below is the call flow when using the Screen Sharing example

screen-sharing.html

screen-sharing.js



#### 1. Checking if the extension install is required

Browser.isFirefox(), Browser.isChrome();code

```
if (Browser.isFirefox()) {
   $("#installExtensionButton").show();
    . . .
} else if (Browser.isChrome()) {
   $('#mediaSourceForm').hide();
   interval = setInterval(function() {
       chrome.runtime.sendMessage(extensionId, {type: "isInstalled"}, function (response) {
            if (response) {
                $("#extension").hide();
                clearInterval(interval);
                onExtensionAvailable();
            } else {
                (inIframe()) ? $("#installFromMarket").show() : $("#installExtensionButton").show();
            }
       });
   }, 500);
} else {
   $("#notify").modal('show');
   return false;
}
```

#### 2. Establishing a connection to the server.

Flashphoner.createSession();code

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function(session){
    //session connected, start streaming
    startStreaming(session);
}).on(SESSION_STATUS.DISCONNECTED, function(){
    setStatus(SESSION_STATUS.DISCONNECTED);
    onStopped();
}).on(SESSION_STATUS.FAILED, function(){
    setStatus(SESSION_STATUS.FAILED);
    onStopped();
});
```

3. Receiving from the server an event confirming successful connection.

#### ConnectionStatusEvent ESTABLISHEDcode

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function(session){
    //session connected, start streaming
    startStreaming(session);
    ...
});
```

#### 4. Publishing the stream.

#### stream.publish();code

```
session.createStream({
    name: streamName,
    display: localVideo,
    constraints: constraints
}).on(STREAM_STATUS.PUBLISHING, function(publishStream){
    ...
}).on(STREAM_STATUS.UNPUBLISHED, function(){
    ...
}).on(STREAM_STATUS.FAILED, function(stream){
    ...
}).publish();
```

5. Receiving from the server an event confirming successful publishing.

StreamStatusEvent, статус PUBLISHINGcode

```
session.createStream({
   name: streamName,
   display: localVideo,
   constraints: constraints
}).on(STREAM_STATUS.PUBLISHING, function(publishStream){
    . . .
   setStatus(STREAM_STATUS.PUBLISHING);
   //play preview
   session.createStream({
       name: streamName,
       display: remoteVideo
   }).on(STREAM_STATUS.PLAYING, function(previewStream){
       document.getElementById(previewStream.id()).addEventListener('resize', function(event){
           resizeVideo(event.target);
       });
        //enable stop button
       onStarted(publishStream, previewStream);
    }).on(STREAM_STATUS.STOPPED, function(){
       publishStream.stop();
    }).on(STREAM_STATUS.FAILED, function(stream){
       //preview failed, stop publishStream
       if (publishStream.status() == STREAM_STATUS.PUBLISHING) {
            setStatus(STREAM_STATUS.FAILED, stream);
            publishStream.stop();
       }
    }).play();
}).on(STREAM_STATUS.UNPUBLISHED, function(){
    . . .
}).on(STREAM_STATUS.FAILED, function(stream){
   . . .
}).publish();
```

#### 6. Sending the audio-video stream via WebRTC

7. Stopping publishing the stream.

stream.stop();code

```
session.createStream({
   name: streamName,
   display: localVideo,
   constraints: constraints
}).on(STREAM_STATUS.PUBLISHING, function(publishStream){
   /*
     * User can stop sharing screen capture using Chrome "stop" button.
     * Catch onended video track event and stop publishing.
     * /
   document.getElementById(publishStream.id()).srcObject.getVideoTracks()[0].onended = function (e) {
        publishStream.stop();
    };
    . . .
   setStatus(STREAM_STATUS.PUBLISHING);
   //play preview
   session.createStream({
       name: streamName,
        display: remoteVideo
   }).on(STREAM_STATUS.PLAYING, function(previewStream){
    }).on(STREAM_STATUS.STOPPED, function(){
        publishStream.stop();
    }).on(STREAM_STATUS.FAILED, function(stream){
        //preview failed, stop publishStream
        if (publishStream.status() == STREAM_STATUS.PUBLISHING) {
           setStatus(STREAM_STATUS.FAILED, stream);
           publishStream.stop();
        }
    }).play();
}).publish();
```

8. Receiving from the server an event confirming unpublishing of the stream.

StreamStatusEvent, ctatyc UNPUBLISHEDcode

```
session.createStream({
    name: streamName,
    display: localVideo,
    constraints: constraints
}).on(STREAM_STATUS.PUBLISHING, function(publishStream){
    ...
}).on(STREAM_STATUS.UNPUBLISHED, function(){
    setStatus(STREAM_STATUS.UNPUBLISHED);
    //enable start button
    onStopped();
}).on(STREAM_STATUS.FAILED, function(stream){
    ...
}).publish();
```

# To developer

The screen sharing function can be used to publish a video stream that demonstrates the desktop or an application window. WCS API uses a Chrome extensions for screen sharing. Firefox browser since version 52 does not require an extension.

## Extension for Google Chrome with publication in Chrome Store

Source code for the extension is available by the following link:

**Chrome Screen Sharing Extension** 

#### Create a Google account

1. Go to google.com and click 'Sign in' button



2. On the 'Sign in' page, click 'Create account' link



3. 'Create your Google Account' page will be opened Fill the required fields and click 'Next step' button to create the account.

## Register as Chrome Web Store Developer

1. Sign in to Chrome Developer Dashboard with the created Google account 2. Pay a one-time \$5 developer signup fee



## Customization for your domain

Follow the procedures described below to use the extensions with your domain. Edit manifest file manifest.json of the Chrome extension.

Change:

- name
- author
- description
- homepage\_url
- under "externally\_connectable":"matches" change flashphoner.com to your domain

Save your icons for the extension to chrome-extension directory and edit the file names in "icons" and "web\_accessible\_resources". (For more information, see Manifest - Icons and Supplying Images.)

## Package the extension

Package files from chrome-extension folder into ZIP archive.

### Publish the extension

- Sign in to Chrome Developer Dashboard
   In the dashboard, click 'Add new item' button



#### 3. Accept the developer agreement

C  https://chrome.google.com/webstor	e/developer/terms?hl=en&continue=ht	tps://	/chrome.google.	com/wit
chrome web store	·	Sea	rch the store	(
oogle Chrome Web Store Developer Ag	reement	*		
finitions				
ogle: Google Inc., a Delaware corporation with principal place of t w, CA 94043, United States.	business at 1600 Amphitheatre Parkway, Mountain			
veloper or You: Any person or company who is registered to deve b Store in accordance with the terms of this Agreement.	lop Products for publication and distribution on the			
ab Store: The Google Chrome Web Store site operated by Google tribution to users of Google Chrome.	, where Developers can publish Products for direct			
yment Processor(s): Any party authorized by Google to provide p h optional Payment Accounts to charge Google Chrome users for	ayment processing services that enable Developers Products distributed via the Web Store.	-		
Referently and the				

4. On the 'Upload' page, choose the chrome-extension.zip file and click 'Upload' button

				4	_ 🗆 🗙
🚖 Upload - Developer Dashk	×				
← → C 🔒 https://c	hrome.google.com/w	/ebstore/developer/u	pdate?hl=en&authuser=1	&publisherId=g1111	631047☆ 🔳
n chrome web s	tore			Search the store	٩.
Upload an extension chrome-extension.zip Choose file Upload	or app (.zip file) 0 B / 1.38 KB	Remove			
Upload a ZIP file of your ite     Include a well-designed pro     Read the documentation at     Need more help? Check ou	m directory, not a packaged of iduct icon in your manifest (m yout creating and packaging a tt the Chrome Web Store devi	CRX file. lore info). apps. eloper documentation.			

5. When the extension is uploaded, the page for editing the extension draft will be opened Edit the extension as required and click 'Save draft and return to dashboard' button at the bottom of the page.

🚡 Test Screen Sh	aring - Edit X	▲ - □	×
← → C 🔒	https://chrome.google.com/websi	tore/developer/edit/hjjhafbcdlcbmjkbjogdlhdofkpnggmj?hl=en $\stackrel{\wedge}{\sim}$	Ξ
Chrome	e Web store ard > Test Screen Sharing	Search the store Q	))
Edit Item	In-app Products		
Upload Detailed description	Test Screen Sharing Short name: Not specified in m Version 1.4 by Upload Updated Package This Chrome extension is developed for <y< th=""><th>anifest our domain&gt; to enable screen capture. Focus on explaining what the item does and why users should install it. Characters remaining: 16000</th><th></th></y<>	anifest our domain> to enable screen capture. Focus on explaining what the item does and why users should install it. Characters remaining: 16000	
Icon	Upload new icon		

Inline install	☑ This item uses inline install.	Select this checkbox to allow inline install for your item (note that this may delay publishing). More info
Analytics	Google Analytics ID UA (example: 1234567-1)	
lequirements	No requirements	
Visibility options	Public     Everyone can see It	
	Only people with the link can see it.	
	<ul> <li>Private</li> <li>Only trusted testers from your developer dashboard can see it.</li> <li>You can also include members of a Google Group that you own or manage.</li> </ul>	

6. The extension will appear in the developer dashboard Click 'Publish' link to publish the extension.

The mups.//chrome.go	ogle.com/webs	tore/developer	/dashboard/	rg15527750490	/55606/20(ni=en	22
chrome web store				-	Search the store	Q
eveloper Dashboard						
Check out new Chrome Web Store Mu target system based on its NaCI archite	Itiple Platform feature ecture.	which reduces the	download size of	f your application by o	distributing only relevant files	to the
our Listings (1 - 1 of 1)	Created	Last published	Weekly users	Status		
our Listings (1 - 1 of 1)	Created	Last published	Weekly users	Status	,	Publish Edit
Test Screen Sharing Version 1.4	Created 4/6/16	Last published	Weekly users	Status Draft		Publish Edit More info

A published extension will have status 'Published' as on the image below.

🚡 Developer Dashboard - 🔿 🗙 🦲						<b>▲</b> _ □ ×
← → C 🔒 https://chrome.google.	<b>com</b> /webst	tore/develope	r/dashboard	?hl=en-US&autł	nuser=1	¶☆≣
chrome web store Developer Dashboard				Ţ	Search the store	٩
Your Listings (1 - 1 of 1)	Created	Last published	Weekly users	Status		
♥ Flashphoner Screen Sharing Version 1.4 ★★★★★ (0)	2/23/16	2/25/16	12	Published	Sta	ats   Unpublish   Edit More info
rarger users in other languages. More into	*					Add new item

For more information, see Chrome Web Store Publishing Tutorial.

### **Extension Inline Installation**

Extension installation can be initiated by clicking a link on the Screen Sharing client page. Note that the extension to be installed should be customized, published and approved before inline installation can be used.

Follow the procedures described below to use the client with your extensions.

1. When publishing, select 'Inline Install' option

Inline install	☑ This item uses inline install.	Select this checkbox to allow inline install for your item (note that this may delay publishing). More info
Analytics	Google Analytics ID UA (example: 1234567-1)	
equirements	No requirements	
Visibility options	Public     Everyone can see it	
	Unlisted     Only people with the link can see it.	
	<ul> <li>Private</li> <li>Only trusted testers from your developer dashboard can see it.</li> <li>You can also include members of a Google Group that you own or manage.</li> </ul>	

2. Verify and add website with your domain to the extension

- Click 'Add a New Site'

-> C 🔒 ht	tps://chrome.google.com/webstore/developer/edit/hjjhafbcdlcbmjk	bjogdlhdofkpnggmj?hl=en&authuser=0 🖓
Websites	Verify that this is an official item for a website you own: None Add a new site Refresh list Link to website for your item (optional)	If your item is associated with a website that you own, select that website from the list. You can register as the owner for a site using Google Webmaster Tools. More info Providing the URLs for description and support pages can improve your users' experience and
	② Link to support & FAQ for your item (optional)	help make this item's ratings and comments more meaningful. More info

# - Google Search Console page will open in a new tab Enter URL with your domain.

🖉 Developer Dashboard - 🔾	× 🖶 Search Console - Home 🛛 ×	± _	
← → C 🔒 https://v	ww.google.com/webmasters/tools/home?hl=en		☆ =
Google		0	S
Search Console	Help ~	\$	•
Home All Messages Other Resources	Welcome to Search Console         Get the data, tools, and diagnostics needed to create and maintain Google-friendly websites and apps.         To get started, just add your site or app now.         Image: Im	l mobile	e

- A page with the instruction for the site verification will open Follow the procedure steps and click 'Verify' button.

🖉 🚡 Developer Dashboard - 🗆 🗙	Search Console ×
← → C 🔒 https://www.ge	$\operatorname{cogle.com}/\operatorname{webmasters}/\operatorname{verification}/\operatorname{verification}?hl=en&authuser=0&the \mathfrak{k} = 1$
Google	
Search Console	Help ~
Verify your ownership of https://m.	flashphoner.com/. Learn more.
Recommended method	Alternate methods
Recommended: HTML file of Upload an HTML file to your site.	upload
1. Download this HTML verificat	tion file. [google4b93d2d3a1de58ff.html]
2. Upload the file to https://m.fla	shphoner.com/
<ol><li>Confirm successful upload by</li></ol>	visiting https://m.flashphoner.com/google4b93d2d3a1de58ff.html in your browser.
<ol> <li>Click Verify below.</li> <li>To stay verified, don't remove the</li> </ol>	e HTML file, even after verification succeeds.
VERIFY Not now	

## - If the verification is passed, a page confirming successful verification will be opened

🖉 Developer	er Dashboard - 🗆 🗙 💼 Search Console 🛛 🗙 💽		÷ -		×
← ⇒ C	https://www.google.com/webmasters/verification/verification-suc	cess?hl=en	&sitel	Jrl=☆	≡
Goog	j <mark>le 🕷</mark>		0	s	*
Search	h Console		Help +		
<b>~</b>	Congratulations, you have successfully verified your ownership of https://m.flashphoner.com/. Continue	ŝ.			Ŧ

- The website will appear in the list in the extension options and the extension can be associated with the site

C 🧯	https://chrome.google.com/webstore/developer/edit/hjjhafbcdlcbm	njkbjogdlhdofkpnggmj?hl=en&authus		
Websites	Verify that this is an official item for a website you own:	If your item is associated with a website that yo		
	https://m.flashphoner.com/	register as the owner for a site using Google Webmaster Tools. More info		
	Add a new site Refresh list			
	Link to website for your item (optional)	Providing the URLs for description and suppor		
		pages can improve your users' experience and help make this item's ratings and comments		
	Dipk to support 8 EAO for your item (optional)	more meaningful. More info		

#### Configuring the client

Edit Screen-sharing.html and Screen-sharing.js

- In Screen-sharing.html the chrome-webstore-item parameter should point to your extension in the Chrome Store
- In Screen-sharing is replace the value of the chromeScreenSharingExtensionId parameter to the ID of your extension

To get the ID of the extension, click 'More info' of this extension inChrome Developer Dashboard.

🚊 Developer Dashboard - Cl 🗙 🦲					<b>•</b> -		×
← → C 🔒 https://chrome.go	ogle.com/web:	store/develo	per/dashb	oard?hl=en		ŵ	=
Flashphoner Screen Sharin Version 1.4 ***** (0)	g 2/23/16	2/25/16	14	Published	Stats   Unpublish   Edit More info		
Target users in other languages. More	info 🔻				Item ID: nibaajpipmleofphigmgaifhoikjmbkg		

## Media source parameters

To configure screen media source parameters, use parameters of the Configuration object passed to the init() method upon initializing of the Flashphoner API instance.

```
var f = Flashphoner.getInstance();
var configuration = new Configuration();
....
configuration.screenSharingVideoWidth = 1920;
configuration.screenSharingVideoHeight = 1080;
configuration.screenSharingVideoFps = 10;
f.init(configuration);
```

#### Parameter list

Parameter	Description
screenSharingVideoWidth	Screen media source width
screenSharingVideoHeight	Screen media source height
screenSharingVideoFps	Screen media source framerate

These parameters set marginal values of resolution and framerate (FPS). For instance, screenSharingVideoWidth = 1080 means the width of the source video cannot be more than 1080 pixels, but can be less. (i.e. when sending a stream of an app window that has the width of 720 pixels).

## System sound capture in Chrome browser

In Chrome browser, there is ability to translate audio stream from system sound source while capturing a screen. The feature is useful in screencasting, for example. To capture system sound, set "Share audio" option in Chrome extension dialog window while choosing streaming source window or browser tab:

Your Entire Screen	Application Window	Chrome Tab
antes - <sup>18</sup> 00-anteses at 200 A 21 de una aço a calemanda en conserva por el co	ano-aragon • 4 (r 1] (l ) 12 <b>2 5 8 8</b>	
State of Shaning 200 State of Shaning 201 State of Shaning 202 State of Shaning	N-16000 - 1000	
	in and	
E Altar	105   100   pape	
	ian ta an ta ta ta san san	
	CHARLES MADE AND ADDRESS	

```
Chrome extension code:
```

callback({sourceId: sourceId, systemSoundAccess: opts.canRequestAudioTrack});

## Capture source (screen or window) management in Firefox browser

In Firefox browser, all the screen or some program window can be chosen as video stream source with constraints.video.mediaSource parameter

code:

<pre>constraints.video.type = "screen";</pre>
if (Browser.isFirefox()){
<pre>constraints.video.mediaSource = \$('#mediaSource').val();</pre>
}
<pre>session.createStream({</pre>
name: streamName,
display: localVideo,
constraints: constraints
})

Source selection interface example:



Program window capture



Screen capture

		Screen	Sharing			
F 30	EPS Width 640	Height 480	Source Screen		C Screen Sharing ← → C @	€× ÷
	Use Mic?	Mi default: Micropt	c none (Realte V			International States St
				→		
	No field barriers on this field of the set o	An an anna an an Anna Anna Anna Anna An				My Sen wis Jiesi2 Stellphoner cont 64,51/5676/7 PUDUSH
	My So	reen			-	e 9 9 9 e
wss://test	t2.flashphoner.com:8443/f5b78	c77	Stop			Preview
	PUBLI	SHING				

Screen sharing without extension

### **Firefox browser**

Firefox browser does not use extension to share screen

### Chromium based browsers

Since Chrome 73 and Flashphoner WebSDK0.5.28.2753.86screen sharing is possible without extension. To do this constraints.video.withoutExt ension parameter should be passed when stream is created

#### code

```
if ($("#woChromeExtension").prop('checked')) {
    constraints.video.withoutExtension = true;
}
```

### Safari browser in MacOS

Since Safari 13and Flashphoner WebSDK0.5.28.2753.152screen sharing is possible without extension. To do this constraints.video.withoutExtension parameter should be passed when stream is created

#### code

```
if ($("#woChromeExtension").prop('checked') || Browser.isSafari()) {
    constraints.video.withoutExtension = true;
}
```

#### **Known limits**

1. In Chrome browser, picture resolution and FPS are set by source dimensions (screen, windows or browser tab) and by real picture updating speed, not by constraints/ This issue is fixed sinceFlashphoner WebSDK build0.5.28.2753.152

2. System sound capture only works starting from Crome 74

## Known issues

1.If the web application is inside an iframe element, publishing of the video stream may fail.

Symptoms: IceServer errors in the browser console. Solution: put the app out of iframe to an individual page.

2. If publishing of the stream goes under Windows 10 or Windows 8 and hardware acceleration is enabled in the Google Chrome browser, bitrate problems are possible.

Symptoms: low quality of the video, muddy picture, bitrate shown in chrome://webrtc-internals is less than 100 kbps. Solution: turn off hardware acceleration in the browser, switch the browser of the server to use the VP8 codec.

3. All streams captured from the screen stop if any one of them stops

Symptoms: while multiple streams are captured from the screen on one tab in the Chrome browser, if one stream is stopped, all streams stop. Solution: cache tracks by the source of the video and stop them along with the last stream that uses that source, for example:

```
var handleUnpublished = function(stream) {
   console.log("Stream unpublished with status " + stream.status());
    //get track label
    var video = document.getElementById(stream.id() + LOCAL_CACHED_VIDEO);
    var track = video.srcObject.getVideoTracks()[0];
   var label = track.label;
    //see if someone using this source
    if (countDisplaysWithVideoLabel(label) > 1) {
        //remove srcObject but don't stop tracks
       pushTrack(track);
        video.srcObject = null;
    } else {
       var tracks = popTracks(track);
       for (var i = 0; i < tracks.length; i++) {</pre>
           tracks[i].stop();
        }
    }
    //release resources
    Flashphoner.releaseLocalMedia(streamVideoDisplay);
    //remove stream display
    display.removeChild(streamDisplay);
    session.disconnect();
};
```