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

Overview

Supported platforms and browsers

	Chrome	Firefox	Safari
Windows			×
Linux			×
Mac OS			
Android	×	×	×
iOS	×	×	×

Operation flowchart



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

Quick manual on testing

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-s	haring.html		
	S	Screen Sharing	
Chrome e	extension Install Now		
FPS	Width H	Height	
30	640 480	80	
W/o extension?	Use Mic?	Mic	
Ø		Default - Microphone	
		→	
	My Screen		Denview
wss://demo.flashphone	er.com:8443/3a437ca9	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**:

	Two-way S	treaming	
	Local		Player
5c6f	Publish	3a437ca9	Stop Available
			PLAYING
	wss://demo.flashphoner.com:8443	D	isconnect
	ESTABLIS	HED	

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



2. Establishing a connection to the server



3. Receiving from the server an event confirming successful connection SESSION_STATUS.ESTABLISHED code

```
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UNPUBLISHED, function(){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

STREAM_STATUS.PUBLISHING code

```
session.createStream({
 name: streamName,
  display: localVideo,
  constraints: constraints
}).on(STREAM_STATUS.PUBLISHING, function(publishStream){
  setStatus(STREAM_STATUS.PUBLISHING);
  session.createStream({
     name: streamName,
     display: remoteVideo
  }).on(STREAM_STATUS.PLAYING, function(previewStream){
document.getElementById(previewStream.id()).addEventListener('resize',
function(event){
          resizeVideo(event.target);
      });
     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();
   };
   ...
```



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UNPUBLISHED code

```
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.

The most of moredn browsers do not need an extension, but WCS Javascript API may be used with a Chrome extension for screen sharing for backward compatibility. The example how to deploy a custom screen sharing extension follows.

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

G Sign in - Go	oogle Accounts 🗴 🔚
← ⇒ C	Attps://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.☆ =
	Google
	One account. All of Google.
	Sign in with your Google Account
	Enter your email Next Need help?
	Create account
	One Google Account for everything Google G M ♥ • 6

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

C enttps://chrome.google.com/we	ebstore/developer/dashboard?hl=en&pli=1	5
chrome web store	Search the store	C
Developer Dashboard		
Your Listings		
To publish a new app, extension or theme, click "Add	new item". Learn more	
Chrome Web Store The Chrome Web Store is an open marketplace for web apps.	. Start uploading your apps now!	Add new iten
A one-time developer registration fee of US\$5	5.00 is required to verify your account and publish items. Pay this fee now - Learn	more
A one-time developer registration fee of US\$5	5.00 is required to verify your account and publish items. Pay this fee now - Learn	n more
A one-time developer registration fee of US\$5	5.00 is required to verify your account and publish items. Pay t	ihis fee now <mark>- Learn</mark>

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)

Pack the extension

Pack files from chrome-extension folder into ZIP archive.

Publish the extension

1. Sign in to Chrome Developer Dashboard

2. In the dashboard, click Add new item button

-	Search the store	C

3. Accept the developer agreement



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

훝 Upload - Developer Dashi 🗙 🦲		<u> </u>	×
← → C 🔒 https://chrome.google.com/webstore/develope	r/update?hl=en&authuser=1	&publisherId=g1111631047 న	2 ≡
Chrome web store Developer Dashboard > Add new item		Search the store	٩
Upload an extension or app (.zip file) chrome-extension.zip 0 B / 1.38 KB Remove Choose file Upload			
Uploading an item: Upload a ZIP file of your item directory, not a packaged CRX file. Include a well-designed product icon in your manifest (more info). Read the documentation about creating and packaging apps. Need more help? Check out the Chrome Web Store developer documentation.			

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

-> C 🔒	https://cnrome.google.com/webstore/developer/	a alth /lattla a fla a alt -la thela t	h d a fluor a grad (2 k l
		edit/hjjhafbcdlcbmjkbjogd	hdofkpnggmj?hl=en 🖓
chrome	web store d > Test Screen Sharing	÷	Search the store C
Edit Item	In-app Products		
Upload	Test Screen Sharing Short name: Not specified in manifest Version 1.4 by Upload Updated Package This Chrome extension is developed for <your domain=""> to ena</your>	ble screen capture.	
Detailed description		Focus or users sh Characte	n explaining what the item does and why ould install it. ers remaining: 16000
Icon	\$		
Test Screen Shari	Upload new icon		=
Test Screen Shari	Upload new icon ng - Edit ×	it/mclajopigkalbkpcdemdgd	ຢ □ cdbmlibpch?hl=en&authuser=ີຜ
Test Screen Shari C A h Inline install	Upload new icon ng - Edit ×	it/mclajopigkalbkpcdemdgd Select thi: your item More info	cdbmlibpch?hl=en&authuser= $\%$ s checkbox to allow inline install for (note that this may delay publishing).
Test Screen Shari C An Inline install Analytics	Upload new Icon mg - Edit × ttps://chrome.google.com/webstore/developer/ed This item uses inline install. Google Analytics ID UA- (example: 1)	it/mclajopigkalbkpcdemdgd Select this your item More info 234567-1)	cdbmlibpch?hl=en&authuser=5
Test Screen Shari C Inline install Analytics Requirements	Upload new icon ng - Edit × ttps://chrome.google.com/webstore/developer/ed This item uses inline install. Google Analytics ID UA- (example: 1 No requirements	it/mclajopigkalbkpcdemdgd Select this your item More info	dbmlibpch?hl=en&authuser=☆ cdbmlibpch?hl=en&authuser=☆ checkbox to allow inline install for (note that this may delay publishing).
Test Screen Shari	Upload new icon Ing - Edit × This item uses inline install. Google Analytics ID UA- (example: 1 No requirements Public Everyone can see it Unlisted Only people with the link can see it. Ohy people with the link can see it. Ohy provide testers from your developer dashboard can see it. You can also include members of a Google Group that you own or man	it/mclajopigkalbkpcdemdgd Select this your item More info 234567-1)	cdbmlibpch?hl=en&authuser=☆ checkbox to allow inline install for (note that this may delay publishing).

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

<u>-</u>	ogle.com/webs	tore/developer	/dashboard/	g15527730490	755806720?hl=en	삷
n chrome web store				-	Search the store	Q
eveloper Dashboard						
Check out new Chrome Web Store Mu target system based on its NaCl archite	Itiple Platform feature acture.	which reduces the	download size of	your application by c	distributing only relevant files	to the
Your Listings (1 - 1 of 1)	Created	Last published	Weekly users	Status		
- Test Screen Sharing						
Version 1.4	4/6/16		0	Draft		Publish Edit
						intere inte
***** (0)						

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

C https://chrome.goo	gle.com/webst	tore/develope	r/dashboard	?hl=en-US&auth	nuser=1	₽ ☆
n chrome web store					Search the store	. C
eveloper Dashboard						
our Listings (1 - 1 of 1)	Created	Last published	Weekly users	Status		
Flashphoner Screen Sharin Version 1.4 (0)	g 2/23/16	2/25/16	12	Published		Stats Unpublish Edit More info
	and a second					

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)	
Requirements	No requirements	
Visibility options	Public Everyone can see it	
	Unlisted Only people with the link can see it.	
	Private Only trusted testers from your developer dashboard can see it. You can also include members of a Google Group that you own or manage.	

- 2. Verify and add website with your domain to the extension
 - Click Add a New Site



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

→ C 🔒 https://v	www.google.com/webmasters/tools/home?hl=en			ŵ
G <mark>oogle</mark>			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 G apps. To get started, just add your site or app now. https://m.flashphoner.com/	Google-friendly websites an ADD PROPERTY	nd mobi	le

• 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.google.com/webma	sters/verification/verification?hl=en&authuser=0&the $\frac{1}{\sqrt{2}}$				
Google					
Search Console	Help ~				
Verify your ownership of https://m.flashphoner.com/. Lean	n more.				
Recommended method Alternate methods					
Recommended: HTML file upload Upload an HTML file to your site.					
1. Download this HTML verification file. [google4b93d2d2	3a1de58ff.html]				
2. Upload the file to https://m.flashphoner.com/					
Confirm successful upload by visiting https://m.flashphoner.com/google4b93d2d3a1de58ff.html in your browser.					
To stay verified, don't remove the HTML file, even after ve	rification succeeds.				
VERIFY Not now					

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



• 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/hjjhafbcdlcbn	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 your select that website from the list. You can			
	https://m.flashphoner.com/	register as the owner for a site using Google			
	Add a new site Refresh list	Webmaster Tools. More info			
	General 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			

CONFIGURING THE CLIENT

- Edit Screen-sharing.html and Screen-sharing.js
 - In <u>Screen-sharing.html</u> the <u>chrome-webstore-item</u> parameter should point to your extension in the Chrome Store

• In Screen-sharing.js replace the value of the chromeScreenSharingExtensionId parameter by the ID of your extension

To get the ID of the extension, click More info of this extension in Chrome Developer Dashboard



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 (for example 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 <u>Share audio</u> option in Chrome dialog window while choosing streaming source window or browser tab:

cation Window Chrome Ta
and a first state of

Chrome extension code:

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

Capture source (screen or window) management in Firefox browser

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

code:

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

Source selection interface example:

FPS	Width	Height	Source		
30	640	480	Screen 🗸		
	Use Mic?	M default: Microp	Screen Window horne (recant		
My Screen					
wss://test2.flas	hphoner.com:8443/f5b7	8c77	Start		

Attention

In modern Firefox versions, a media source may be chosen only in browser sharing dialog window.

Program window capture example

		Screen	Sharing			
FPS 30	Width 640 Use Mic?	Height 480 Mic default: Microph	Source Window >		C: Users's solar threads the solar s	алык (Сном Ратэ 109–70 109–351 109–351 109–454 109–65 109–65 109–23 109–24 109–23 109–24 100–24 100
				→	where (<) in provide size account, size where (<) in provide size (<) is a where (<) in provide size (<) in provide size (<) in where (<) in provide size (<) in provide size (<) in where (<) in provide size (<) in provide size (<) in where (<) in provide size (<) in provide size (<) in the (<) in provide size (<) in provide size (<) in the (<) in provide size (<) in provide size (<) in the (<) in provide size (<) in provide size (<) in the (<) in provide size (<) in provide size (<) in the (<) in provide size (<) in provide s	ing 80% ing 778 ing 133 ing 477 ing 133 ing 477 ing 255 ing 265 ing 265 ing 129 ing
wss://test2.flas	My s hphoner.com:8443/f5b7	Screen 8c77	Stop		Felest[b]3334 Preview	
	PUBL	ISHING				

Screen capture example

		Screen	Sharing			
FPS 30 Use I	Width 640 Mic?	Height 480	Source Screen ~		O Sceen Sharing ← → C @	
	Control and a control and			÷	4	30 640 Use Mc?
wss://test2.flashphor	my scre	7	Stop			Preview
	PUBLISH	ING				

Screen sharing without extension

Firefox browser

Firefox browser does not use extension to share screen

Chromium based browsers

Since Chrome 73 and Flashphoner WebSDK 0.5.28.2753.86 screen sharing is possible without extension. To do this constraints.video.withoutExtension parameter should be passed when stream is created

code

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

Safari browser in MacOS

Since Safari 13 and Flashphoner WebSDK 0.5.28.2753.152 screen 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 since Flashphoner WebSDK build 0.5.28.2753.152
- 2. System sound capture only works starting from Chrome 74

Known issues

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



2. Bitrate problems are possible when publishing a stream under Windows 10 or Windows 8 with hardware acceleration enabled in a browser

6 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 or use the VP8 codec to publish

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:



4. Chrome browser stops sending video traffic when application window captured is minimized

