

VIEWER, ORBITTOUR, AND 360° PRESENTATIONS



Table of Contents

Introduction.....	3
What is ORBITVU VIEWER.....	4
What is Orbitvu Presentation (360°/3D).....	5
Software download.....	6
Licensing.....	9
Integration with websites	11
Embed 360° Presentation using ORBITVU SUN embed codes.....	12
Embed ORBITTOUR using ORBITVU SUN embed code	15
Hosting 360° Presentation on your own server	18
Flash only (eBay etc.)	22
Server side CORS headers.....	24
Server side configuration for Flash	25
Offline usage on mobile devices.....	26
Offline use with Android device	27
Offline use with iPad device.....	29
Integration examples	36
Embedding with ORBITVU SUN embed codes	37
Parameters reference	47
How to customize ORBITVU VIEWER with parameters.....	48
ORBITVU VIEWER parameters reference.....	50
Customizing ORBITVU SUN embed codes	62
Customizing ORBITTOUR embed code.....	64
ORBITVU VIEWER API	68
ORBITTOUR API	75

Introduction

What is ORBITVU VIEWER

ORBITVU VIEWER is an interactive player for [Orbitvu presentations](#) (360°/3D) created with ALPHASHOT EDITOR or ORBITVU SUN.

Technically ORBITVU VIEWER is Javascript/Flash™ application distributed as an orbitvu12/ folder. It automatically detects Flash™ plugin presence and uses either Flash or HTML5 technology in the browser to play 360°/3D presentations.

Below you can see ORBITVU VIEWER playing 360° [Orbitvu presentation](#) of Canon camera:

ORBITVU VIEWER is included for free with presentations hosted at ORBITVU SUN. Orbitvu presentations that are hosted on other servers need their own copy of ORBITVU VIEWER (separate license).

ORBITVU VIEWER is an integral part of ORBITTOUR that enhances 360°/3D Orbitvu presentations with interactive views, hotspots, images, videos, and descriptions. ORBITTOUR is available with ORBITVU SUN only.

- i** ORBITVU VIEWER works in Flash mode or in HTML5 mode. The default mode is Flash mode.

Above means that if there is no Flash support in the browser (which is the case on mobile devices) then HTML5 mode is used automatically. It is possible to force using HTML5 with `force_html5=yes` parameter.

Due to a big amount of Flash plugin vulnerabilities, Flash is now often blocked by Web Browsers so it is recommended to use HTML5 version with `force_html5=yes`

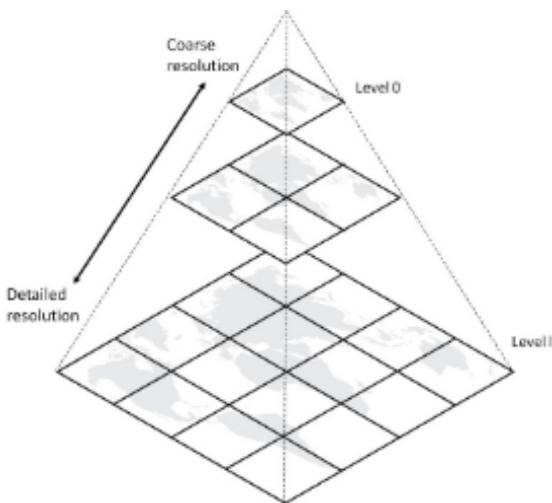
What is Orbitvu Presentation (360°/3D)

Technically, Orbitvu presentation is a folder that contains image files (for 2D / 360° / 3D content) and some XML configuration files. Its structure is like:

```
presentationX/
  images/
    a_0_0_1000_0_0.jpg
    a_0_0_1000_0_1.jpg
    a_0_0_1000_1_0.jpg
    a_0_0_1000_1_1.jpg
    (...)

  images2d/
    some_image.jpeg
  config.xml
  content2.xml
  meta.xml
```

Orbitvu presentation structure is optimised for the Web Browser environment. Specifically, image files for 360° presentation are cut into small tiles in order to make Orbitvu presentation loading as efficient as possible.



Presentation of this type can be played by [ORBITVU VIEWER](#).

To create Orbitvu presentation you can use ALPHASHOT EDITOR software or [ORBITVU SUN](#).

Software download

Download ORBITVU VIEWER from [Orbitvu SUN](#). Log in using credentials that you've received in a separate e-mail.

- 💡 You don't have to download ORBITVU VIEWER if you're using ORBITVU SUN to host Orbitvu presentations (fully featured ORBITVU VIEWER version is automatically used in this case).

The screenshot shows a web browser window for the Orbitvu SUN server. The address bar indicates a secure connection to <https://orbitvu.co/account/login>. A green banner at the top states "You have been successfully logged out". The main content area features the Orbitvu SUN logo and the text "ORBITVU SUN server - digital asset management and distribution platform" followed by "Manage, reuse and redistribute your 360° presentations effortlessly". Below this is a "learn more" button. The page is divided into two main sections: "Log in" on the left and "Register free" on the right. The "Log in" section contains fields for "E-mail" (with value "@orbitvu.com") and "Password" (with value "*****"). The "Register free" section contains fields for "Company", "E-mail address", "Country" (with dropdown menu "Choose a country"), "Password", "Confirm password", and checkboxes for "Accept terms and conditions" and "Send me newsletter". Three numbered circles (1, 2, 3) are overlaid on the "Log in" section, pointing to the E-mail field, the Password field, and the "Log in" button respectively.

Go to My Products section and choose SOFTWARE, then scroll down to VIEWER section.

ORBITVU SUN Subscription plans

Subscription plan lets you store and manage your packshots and 360° presentations within ORBITVU SUN. All plans are valid for **one year** and include: ORBITTOUR CDN

	No. of presentations	Storage Limit	View limit per month	Statistics	Ecommerce integrations	
TRIAL	Free 3 months	20	300.0 MB	1 000	yes	no
your plan	Customers forever	20	300.0 MB	1 000	no	no
				<small>views</small>		<small>Default if you've purchased Viewer/Editor license</small>

Select ORBITVU VIEWER version to download and click Download link

ORBITVU VIEWER

Documentation (including ORBITVU SUN embed codes)

[Online documentation](#) (opens in new window)

Latest releases

ORBITVU VIEWER: **3.0.21** (2017-01-26) [releases history](#)

Licenses

License type	License owner	Locked domains	Support expires at	Available version	Download link
BASIC	ORBITVU Sp. z o.o.			3.0.21	Download
DEMO	ORBITVU Sp. z o.o.			3.0.21	Download
ORBITVU VIEWER ROYALTY FREE	Orbitvu Sp. z o.o.		2017-08-05	3.0.21	Download
ORBITVU VIEWER ROYALTY FREE	Orbitvu Sp. z o.o.		2016-12-21 <small>Renew maintenance</small>	3.0.0 <small>Version 3.0.21 is out! Click to upgrade</small>	Download

Proceed by clicking Download button

ORBITVU VIEWER ROYALTY FREE

[download](#)

Version: 3.0.21

Product license: [view license](#)

[Download now](#)

Older releases

- [3.0.20 - download](#)
- [3.0.0 - download](#)
- [2.3.6 - download](#)
- [2.3.0 - download](#)
- [2.2.3 - download](#)

Licensing

ORBITVU VIEWER has a number of different licensing options. Detailed feature comparison is available at Orbitvu website: <https://orbitvu.com/orbitvu-viewer-feature-comparison>.

Detailed license is available at [https://orbitvu.co/media/documents/
LICENCE_ORBITVU_VIEWER.pdf](https://orbitvu.co/media/documents/LICENCE_ORBITVU_VIEWER.pdf)

Types of ORBITVU VIEWER licenses:

- **DEMO**

The freely available fully featured version of ORBITVU VIEWER. License restricts its use to test purposes only. Demo version always shows Info button that opens a popup with Orbitvu information - this cannot be disabled.

- **BASIC**

Freely available version of ORBITVU VIEWER but with limited functionality:

- max zoom is limited to 1024px x 1024px
- it is not possible to change colours of the buttons
- no branding button
- no social button
- no logo button
- no API (Application Programming Interface) so it can't be used with ORBITTOURs downloaded from ORBITVU SUN

- **STANDARD**

The commercial version that makes it possible for clients to use ORBITVU VIEWER on their own servers.

This version is branded with client's company name (shown in Info popup (in Flash version it is also visible in context menu visible after right mouse button click).

Restrictions:

- ORBITVU VIEWER STANDARD can be only used for a single domain name that has to be provided by client during purchase
- no API (Application Programming Interface) so it can't be used with ORBITTOURs downloaded from ORBITVU SUN

- **ROYALTY FREE**

The fully featured commercial version of ORBITVU VIEWER that lets clients use ORBITVU VIEWER on their own servers and makes it possible to transfer ORBITVU VIEWER with presentations to their own customers. Features:

- Fully featured - API (Application Programming Interface) included - can be used with ORBITTOURs
- No forced domain restrictions
- Can be transferred with presentations to clients of ORBITVU clients.

- **SUN**

The special version of ORBITVU VIEWER used for all the presentations hosted with ORBITVU SUN. It **can't be used** with presentations hosted elsewhere. This is fully featured and always up to date version of ORBITVU VIEWER.

Clients using ORBITVU SUN to host their presentations don't have to buy separate ORBITVU VIEWER license - they are always provided with the newest version of the VIEWER.

Integration with websites

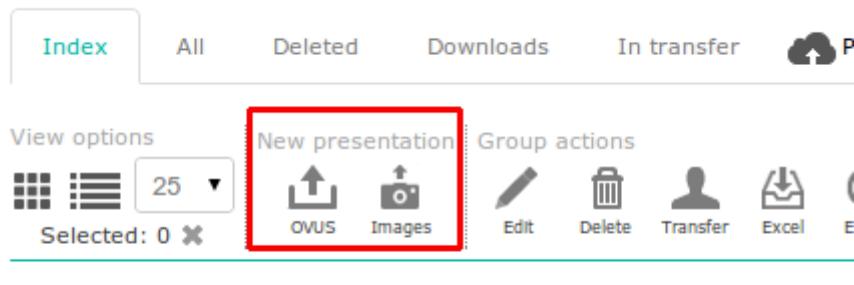
Embed 360° Presentation using ORBITVU SUN embed codes

The most effective and straightforward way to have presentations at your website is to use [ORBITVU SUN](#) hosting platform. This way you don't need technical knowledge and don't have to deal with technical issues like how to make presentations work with Facebook or Twitter. Only a few simple steps are necessary for this:

- Log in to ORBITVU SUN

The screenshot shows the ORBITVU SUN website interface. On the left, there is a 'Login' form with fields for 'E-mail *' and 'Password *', and buttons for 'Reset password' and 'Login'. A yellow box highlights this form. On the right, there is a 'Register free' form with fields for 'Company *', 'E-mail address *' (with a note about business email), 'Password *', 'Confirm password *', and checkboxes for 'Accept terms and conditions *' and 'Send me newsletter'. A 'Register' button is at the bottom. The ORBITVU SUN logo is visible at the top left.

- Upload 360° presentations
You can upload presentations directly from ORBITVU EDITOR or from ORBITVU SUN management panel



- Configure presentation
Adjust parameters using ORBITVU SUN's presentation editor

Movie

Author: John Smith | SKU: - | In portfolio no | 20.2 MB | © 2016-08-29 | edit
+ add a tag

Assets ORBITTOUR Transfer Download Delete presentation

Basic settings

- Teaser: Zoom in
- GUI buttons style: Square scroll
- Color theme: Custom
- First frame: 1
- Extra buttons:
 - Social:
 - Buy: a.g.: http://orbitvu.co
 - Branding text:
 - Branding link: a.g.: http://orbitvu.com

Save

The right side shows a 3D rendering of a white device with a blue circular logo and the word "ORBITVU".

- Get embed codes
Copy Embed codes from presentation's Embed tab:

← Presentations

Egg-Shell Prestige Woodwind

Author: encore7.com | SKU: - | Audio Tube Amplifier | In portfolio no | 71.9 MB | © 2013-08-09 | License | edit
+ add a tag

Assets ORBITTOUR Statistics Transfer Download Delete presentation

Standard

- Standard
- Alternate
- Ebay.com
- Allegro.pl
- CSS only
- Direct link
- QRcode
- Facebook
- Twitter
- Image
- GIF
- IFRAME

Embed

```
<script type="text/javascript">
src="//orbitvu.co/001/pcFGPQAZSh3mJWtPGezuBP/ov360f13/script?
width=auto&content2=yes&height=auto&force_html5=yes"</script>
<div class="orbitvu-viewer" style="width: 600px; height: 600px;">
<div id="ovContent-pcFGPQAZSh3mJWtPGezuBP" style="background:
url(//d12rnwmnj7urumm7.cloudfront.net/001/pcFGPQAZSh3mJWtPGezuBP/proc/...)">
```

Auto size:
Width: **600px**
Height: **600px**

You can use other dimensions or set different settings by using VIEWER parameters. Check out the [VIEWER documentation](#) for details.

The right side shows a 3D rendering of a white audio tube amplifier with a screen displaying its internal components.

- Paste the code into your page

```
<script type="text/javascript" src="//orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/ov3601/3/
script?width=auto&content2=yes&height=auto"></script>
<div class="orbitvu-viewer" style="background: url("//static.orbitvu.co/001/
pcFGPQAZSh3mJWrPGezuBP/iprof/600/600/img01.png) no-repeat center center
#ffffffffff; width: 600px; height: 600px">
    <div id="ovContent-pcFGPQAZSh3mJWrPGezuBP" class="orbitvu-container"></div>
</div>
```

- 💡 Additionally, you can fine-tune embedded presentation by using [extra parameters](#). For example to force VIEWER always run in HTML5 mode one can use `force_html5=yes` parameter:

```
<script type="text/javascript" src="//orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/ov3601/3/
script?width=auto&content2=yes&force_html5=yes&height=auto"></script>
```

- 💡 It is possible to change some details of generated embed codes, e.g. id of the presentation container element. Check [embed codes reference](#) for details.

Embed ORBITTOUR using ORBITVU SUN embed code

- Log in to [ORBITVU SUN](#)

The screenshot shows two side-by-side forms. On the left is the 'Login' form, which includes fields for 'E-mail *' and 'Password *', and buttons for 'Reset password' and 'Login'. On the right is the 'Register free' form, which includes fields for 'Company *', 'E-mail address *', 'Password *', and 'Confirm password *'. It also features checkboxes for 'Accept terms and conditions *' (with a link to 'Download terms and conditions') and 'Send mc newsletter' (with a note about unsubscribing). A 'Register' button is at the bottom.

- Upload presentations
You can upload presentations directly from ORBITVU EDITOR or from [ORBITVU SUN](#) management panel.

The screenshot shows the 'Index' tab selected in the navigation bar. Below it, there are filters for 'View options' (grid view, 25 items), 'Selected: 0', and a search bar containing 'ORBITTOUR'. In the center, there's a 'New presentation' button with two sub-options: 'OVUS' and 'Images', both of which are highlighted with a red box. To the right of these are 'Group actions' buttons for 'Edit', 'Delete', 'Transfer', 'Excel', and a dropdown menu. At the bottom, there are sorting options for 'Name' and 'Category'.

- Go to ORBITTOUR Editor page

Egg-SHELL Prestige Woodwind

Author: encore7.com | SKU: - | : Audio Tube Amplifier | : Home Music | In portfolio: no | 71.9 MB | 2013-08-09 | License | edit
dokumentacja x + add a tag

Assets **ORBITTOUR** Statistics Transfer Download Delete presentation

Drop image or video URL or click

Basic settings Advanced settings Embed codes

Standard <script type="text/javascript" src="//orbitvu.co/001/nCFD0A7Sh3mIWxDGozuRD/av26013/script2

- Create ORBITTOUR
Use online editor to add Hotspots and Views

Views

Drag thumbnail here to create view

Back

Hotspots

Defaults SEE MORE

Right side Fuses Glass front panel

ORBITTOUR

Click thumbnail to change view

Teaser message:

Tour: Play type: none Transition time: 2.5 s Autoplay

Message style: Position top Text Background Size 10 Time 5 s

Other settings: Style Default style Node HTML5 Switch Views on hover: no Default: HTML5

Templates

High heels

Created at: 2016-10-10 10:17:56 Type: 360° (36)

Apply Remove

Enter template name Add

Create new template from current ORBITTOUR

- Copy embed code and paste into your site

High heels

Created at: 2016-10-10 10:17:56 Type: 360° (36)

Apply **Remove**

Enter template name **Add**

Create new template from current ORBITTOUR

Add play button: **none** **2.5 s** autoplay

Message style: **(reset)** Position: **top** Text: **Text** Background: **Background** Size: **10** Time: **5 s**

Other settings: Style: **Default style** Mode: **HTML5** Switch Views on hover: **no** Default: **HTML5** Background color: **Background color**

Save **Clear**

Translations

Translate ORBITTOUR to: **British English (en-gb)** **Translate!**

Language will be automatically selected according to the user's browser language preferences. You can also use preferred_language parameter within embed code to force usage of specific language.

Polish (pl)

Spanish (es)

Direct link

<https://orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/2/orbittour/479/view>

Open in new tab

Embed code

```
<div id="orbittour-pcFGPQAZSh3mJWrPGezuBP" class="orbittour-container"></div>
<script type="text/javascript" src="/orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/2/orbittour/479/script"></script>
```

Preview



- ⚠** When embedding ORBITTOUR at your site note that ORBITTOUR will try to fit its size to the container element (this lets you create responsive layouts). Because of that, you should specify some width and height for the element that ORBITTOUR will appear in. If you embed ORBITTOUR into an element (eg. div) with the height of 0px then ORBITTOUR will be shown in 16:9 proportions (width:height).

Hosting 360° Presentation on your own server

To host presentations from your own server you need:

- access to your server (to copy files and configure HTTP server)
- your own copy of ORBITVU VIEWER (you have to buy the license or use the ORBITVU VIEWER BASIC)
- 360° presentations
- some technical knowledge on HTML and server administration
- you have to understand what is ORBITVU PRESENTATION and what is [ORBITVU VIEWER](#).

Deployment

- VIEWER
 - Copy ORBITVU VIEWER to your server (copy orbitvu12/ folder)
 - Configure your web server to serve files from orbitvu12/ directory (make sure these are publicly accessible under some URL, e.g.: <http://yourdomain.com/orbitvu12/>)
- Presentations
 - Copy Presentations to your server (usually you'll use some FTP client)
 - Configure your web server to serve presentation files under some publicly accessible URLs, e.g.: <http://yourdomain.com/presentations/>
- Server configuration
 - Add Access-Control-Allow-Origin *; header to both VIEWER and Presentations URLs to avoid possible CORS (Cross-origin resource sharing) issues
 - Add crossdomain.xml file to the domain used to serve presentation files in order to avoid Flash security issues, e.g. <https://mydomain.com/crossdomain.xml>
- Coding
 - Create web pages that are using your ORBITVU VIEWER and your presentations URLs (see below for details)

i You might also want to get and configure SSL certificate for your domain which is required e.g. if you'll want to add Facebook integration.

i To identify the CORS issues you should use HTML5 version of the viewer. Disable flash in your browser or set force_html5 VIEWER's parameter to yes.

Web page coding

- Load ORBITVU VIEWER helper scripts
You can add scripts to the <head> section of the page.

```
<script type="text/javascript" src="http://yourdomain.com/orbitvu12/swfobject.js"></script>
<script type="text/javascript" src="http://yourdomain.com/orbitvu12/orbitvu.js"></script>
```

- i** If you're going to always use HTML5 version of the ORBITVU VIEWER (using force_html5: 'yes') then you don't have to load swfobject.js file.

- Create container element for the presentation
Create a container element that your presentation will be inserted to. It is necessary to set the unique id for this container.

```
<div id="presentation1-container"></div>
```

- i** If you want to have multiple presentations at your page then just create container with unique id for each of these

- Initialize presentation
The last step is to call JavaScript function inject_orbitvu that will inject the presentation into the container.

```
1  <script type="text/javascript">
2      inject_orbitvu(
3          'presentation1-container',
4          'http://yourdomain.com/orbitvu12/orbitvuer12.swf',
5          'http://yourdomain.com/orbitvu12/expressInstall.swf',
6          {
7              ovus_folder: "http://yourdomain.com/presentations/presentation1/",
8              content2: 'yes',
9              width: "500",
10             height: "400"
11         });
12     </script>
```

Line 3:

id of the container element

Lines 4-5:

URLs to ORBITVU VIEWER files (hosted at your server)

Line 7:

URL to the presentation directory (hosted at your server)

Line 8:

Set to yes if there is content2.xml file in your presentation's directory, otherwise set to no or remove

Line 9-10:

Other ORBITVU VIEWER parameters. For more see [parameters reference](#).

⚠️ Helper scripts have to be loaded BEFORE you call inject_orbitvu

ℹ️ If you have multiple presentations at your page then you need to call *inject_orbitvu* for each of them

ℹ️ You might want to call *inject_orbitvu* inside *onready* event handler

Complete example

Example uses jQuery to defer JavaScript execution on page load

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />

    <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
    <script type="text/javascript" src="http://yourdomain.com/orbitvu12/swfobject.
```

```

js"></script>
<script type="text/javascript" src="http://yourdomain.com/orbitvu12/orbitvu.
js"></script>

<script type="text/javascript">
$(function() {
    inject_orbitvu(
        'presentation1-container',
        'http://yourdomain.com/orbitvu12/orbitvuer12.swf',
        'http://yourdomain.com/orbitvu12/expressInstall.swf',
        {
            ovus_folder: "http://yourdomain.com/presentations/presentation1",
            content2: "yes",
            width: "500",
            height: "400"
        });
});
</script>
</head>
<body>
<h1>Some text</h1>

<div id="presentation1-container"></div>
</body>
</html>

```

- i** Take care of setting proper DOCTYPE for your page especially for Internet Explorer and HTML5 version of ORBITVU VIEWER

Flash only (eBay etc.)

For some sites, e.g. eBay, due to their policies, it might be only possible to use Flash version of ORBITVU VIEWER. Below we will show how to get Flash only embed code.

-  There is also an Only CSS embed code on ORBITVU SUN that is supposed to work with sites (e.g. eBay) that restrict JavaScript use.

ORBITVU SUN embed code

Using ORBITVU SUN you can just copy the eBay embed code from the Embed tab in presentation details and paste it to your auction code.



Your own server

Typical Flash embed code (with some extra parameters) is used to embed presentations at eBay. Parameters that you have to specify are:

- URL of ORBITVU VIEWER SWF file
- URL of the presentation

ovus_folder parameter is used to specify presentation URL. Note that it should be in encoded form. For example, if your URL is:

```
http://mydomain.com/path/to/presentation/
```

then after URL encoding, it will be:

```
http%3A%2F%2Fmydomain.com%2Fpath%2Fto%2Fpresentation%2F
```

You can use site like <http://meyerweb.com/eric/tools/dencoder/> to have your address encoded

The full example of the embed code:

```
<object id="unique-id-DFSD83jD777s" width="100%" height="100%">
    <param name="movie" value="http://mydomain.com/orbitvu12/orbitvuer12.swf" />
    <param name="allowfullscreen" value="true" />
    <param name="allowscriptaccess" value="always" />
    <param name="wmode" value="default" />
    <param name="flashvars"
        value="ovus_folder=http%3A%2F%2Fmydomain.com%2Fpath%2Fto%2Fpresentation%2F"
    />
    <embed width="100%" height="100%" wmode="default" allowscriptaccess="always"
        allowfullscreen="true" type="application/x-shockwave-flash"
        src="http://mydomain.com/orbitvu12/orbitvuer12.swf"
        flashvars="ovus_folder=http%3A%2F%2Fmydomain.
com%2Fpath%2Fto%2Fpresentation%2F">
    </embed>
</object>
```



Note that you have to use unique id for the <object> element

Server side CORS headers

If you're hosting presentation files from the different domain than the rest of your site then you'll have to configure CORS (Cross-Origin Request Sharing) headers. You need to verify that you have `Access-Control-Allow-Origin *` header properly set (use debug toolbar in your browser to do it).

If your web server is Apache, then you can set `Access-Control-Allow-Origin *` using the following `.htaccess` file (located in the directory that contains presentation files):

```
<IfModule mod_headers.c>
    Header set Access-Control-Allow-Origin "*"
</IfModule>
```



⚠ Make sure you also [configured your server](#) properly for Flash security

Server side configuration for Flash

If your presentation files are served from a different domain than the rest of your site then it is possible that Flash security settings will restrict access to these files. This might cause that 360 presentation doesn't work with Flash.

In this case, you'll have to create the special file called `crossdomain.xml` and make it accessible from the the domain hosting your presentations, e.g. <http://yourdomain.com/crossdomain.xml>. The content of `crossdomain.xml` file should be as follows:

```
<cross-domain-policy>
    <allow-access-from domain="*" secure="false"/>
    <site-control permitted-cross-domain-policies="all"/>
    <allow-http-request-headers-from domain="*" headers="*"/>
</cross-domain-policy>
```

 Make sure you have [configured your server](#) properly for HTML5

Offline usage on mobile devices

Offline use with Android device

To host presentations directly from your Android device (offline) you need:

- HTTP server running on your device
- your own ORBITVU VIEWER (you have to buy the license or use a free version)
- [Orbitvu presentation](#)

 Due to the browsers restrictions (Same origin policy in particular), Android browsers do not work properly with Orbitvu presentations that are opened directly from the filesystem. To overcome this limitation it is required to start HTTP server on your device.

Deployment

- prepare your [Orbitvu presentation](#) and ORBITVU VIEWER files (usually, you'll want to use files generated by ORBITVU EDITOR software). You need:
 - orbitvu12/ - directory containing ORBITVU VIEWER
 - presentation_name/ - directory containing [Orbitvu presentation](#) files
 - presentation_name.html - HTML file that embeds presentation
- Install HTTP server software on your Android device. We've confirmed that [KSWEB](#) works properly but you can use HTTP server of your choice (following steps assume you're using KSWEB).
- Copy Orbitvu presentation files (specified above) to your Android device

You can use FTP server built in into KSWEB. To do it:

- start **KSWEB** on your device
- navigate to FTP tab
- click **Add user**
- enter a username, password and the following path: /mnt/sdcard/htdocs (this is the default path used by KSWEB)
- write down **IP address** and **port** shown by KSWEB
- upload presentation files from your computer to Android device using FTP client like Filezilla (use credentials you've specified while adding FTP user)
You can also use Airdroid to copy files to /mnt/sdcard/htdocs
- Assuming you've copied presentation files directly to /mnt/sdcard/htdocs you can now visit:
http://localhost:8080/presentation_name.html in your Android browser.



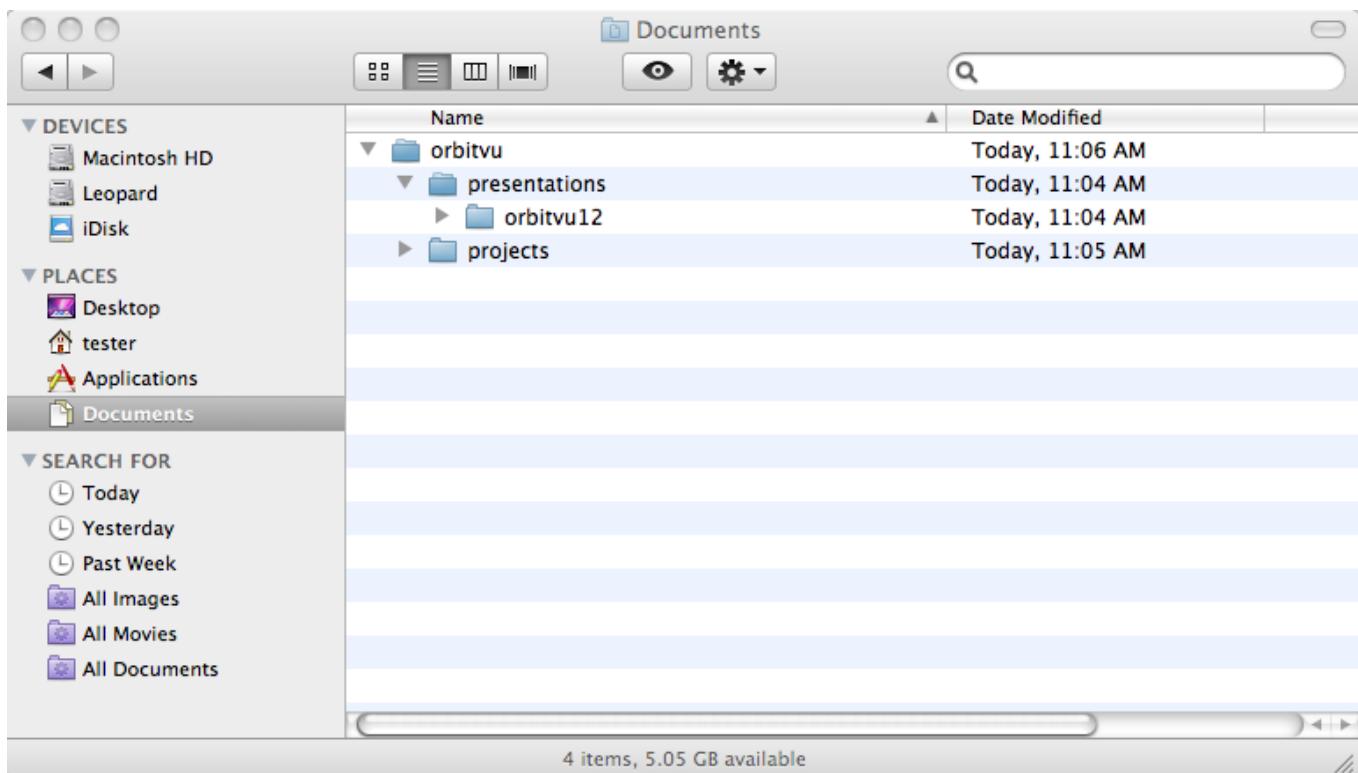
You should secure your installation of KSWEB. Read more at KSWEB website.

Offline use with iPad device

In order to have [Orbitvu presentation](#) (360°/3D) installed on iPad to be used offline, we will use Goodreader application, which can be downloaded directly from App Store.

- Locate [Orbitvu presentations](#) on your computer

ORBITVU EDITOR creates all Orbitvu presentations inside a folder defined in Preferences. Usually, it is: My Documents/orbitvu/presentations.



Creating a new project with name e.g. ananas will add a new subfolder in the projects/ directory.

Generating Presentation will create a folder with the same name in presentations/ subfolder. In addition, you will also see `ananas.html` file, which is a sample file used to preview 360° Orbitvu presentation inside a web browser.

Inside presentations/ folder, you will also find `orbitvu12/` folder, which includes DEMO version of ORBITVU VIEWER – a Flash/HTML5 player which basically displays 360° view on a website.

After creating the `ananas` project and creating the presentation, you should have following file structure:

```
Documents/  
    orbitvu/
```

```
projects/
    ananas/
presentations/
    orbitvu12/
    ananas/
    ananas.html
```

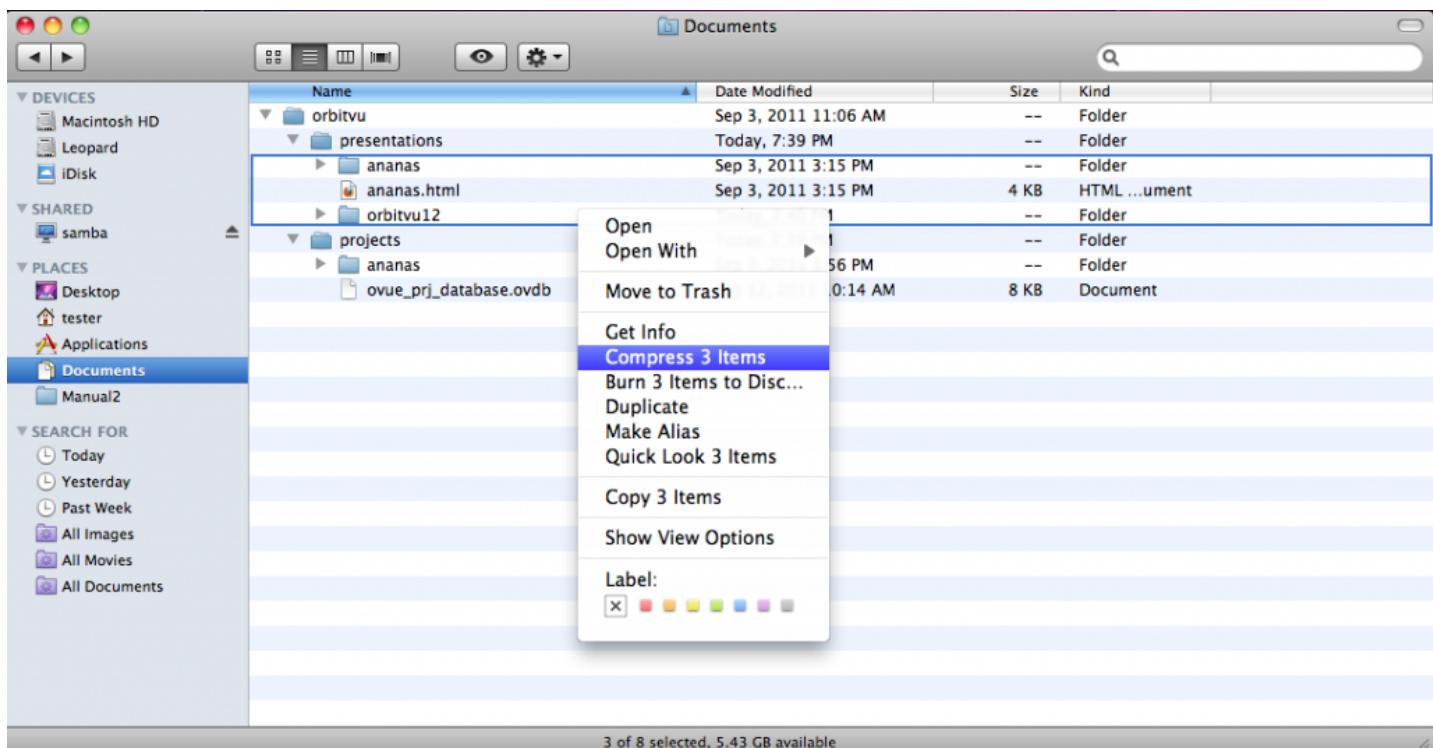
- Create zip file with presentation and the viewer

To view 360° Orbitvu presentations on iPad you need to upload the ORBITVU VIEWER, too. Select:

- presentation folder (ananas/),
- sample HTML file (ananas.html) ,
- ORBITVU VIEWER folder (orbitvu12/)

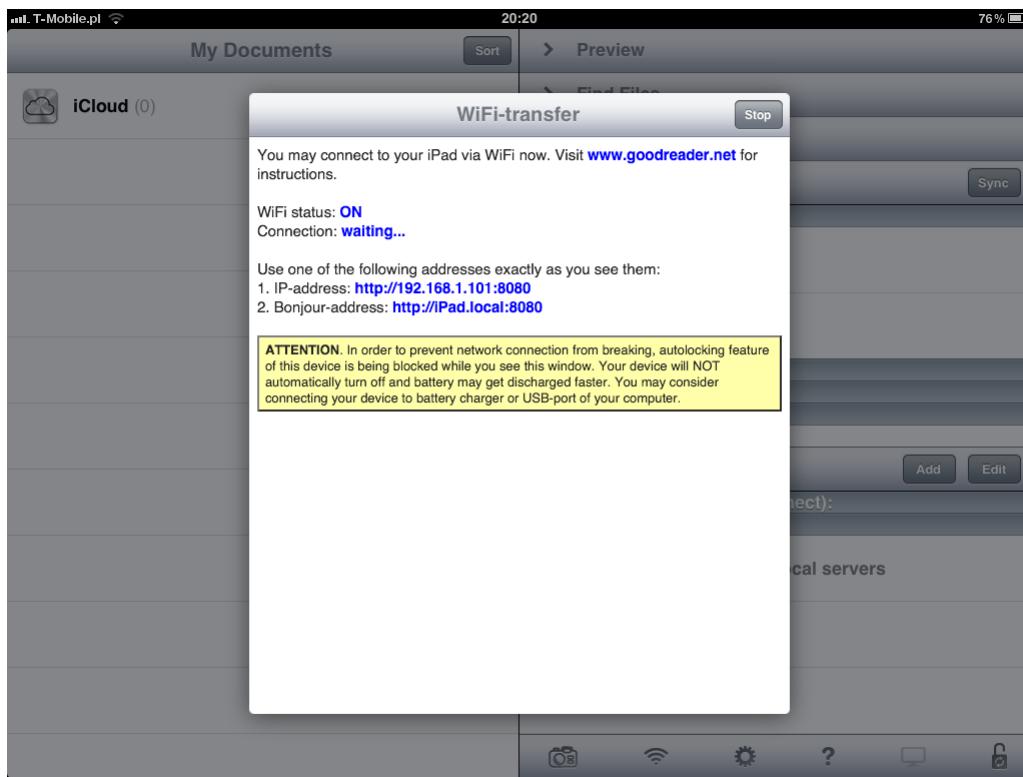
and compress these into a single zip file.

You can, of course, add more presentations to zip file at once.

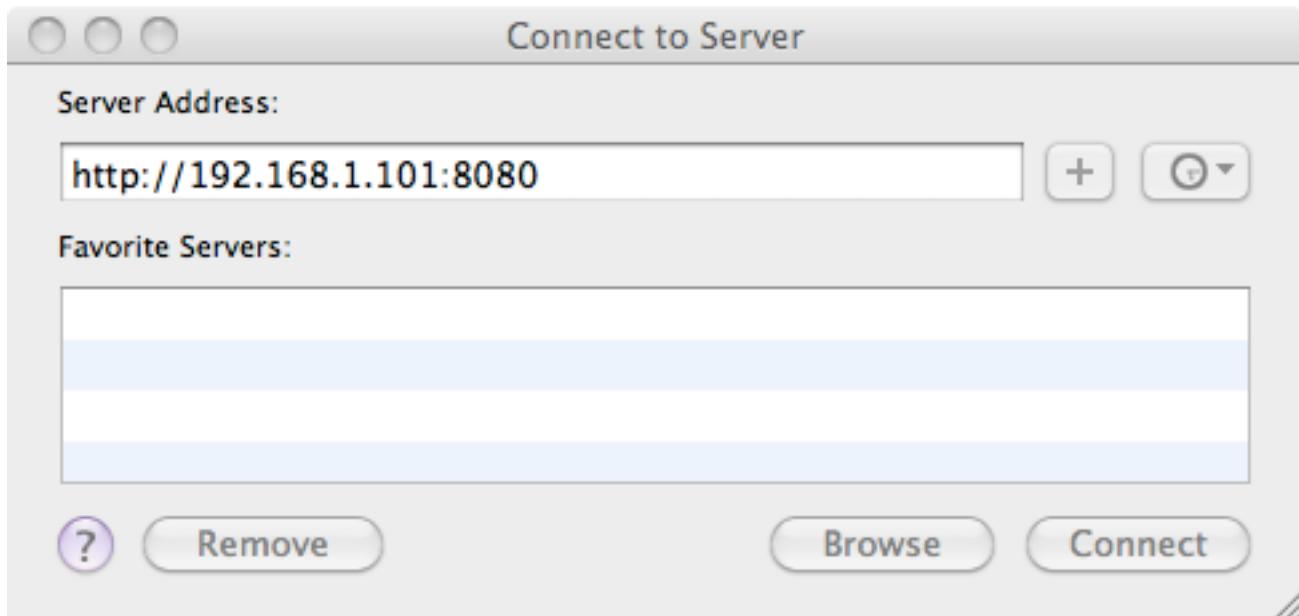


- Upload the zip file into iPad with Goodreader

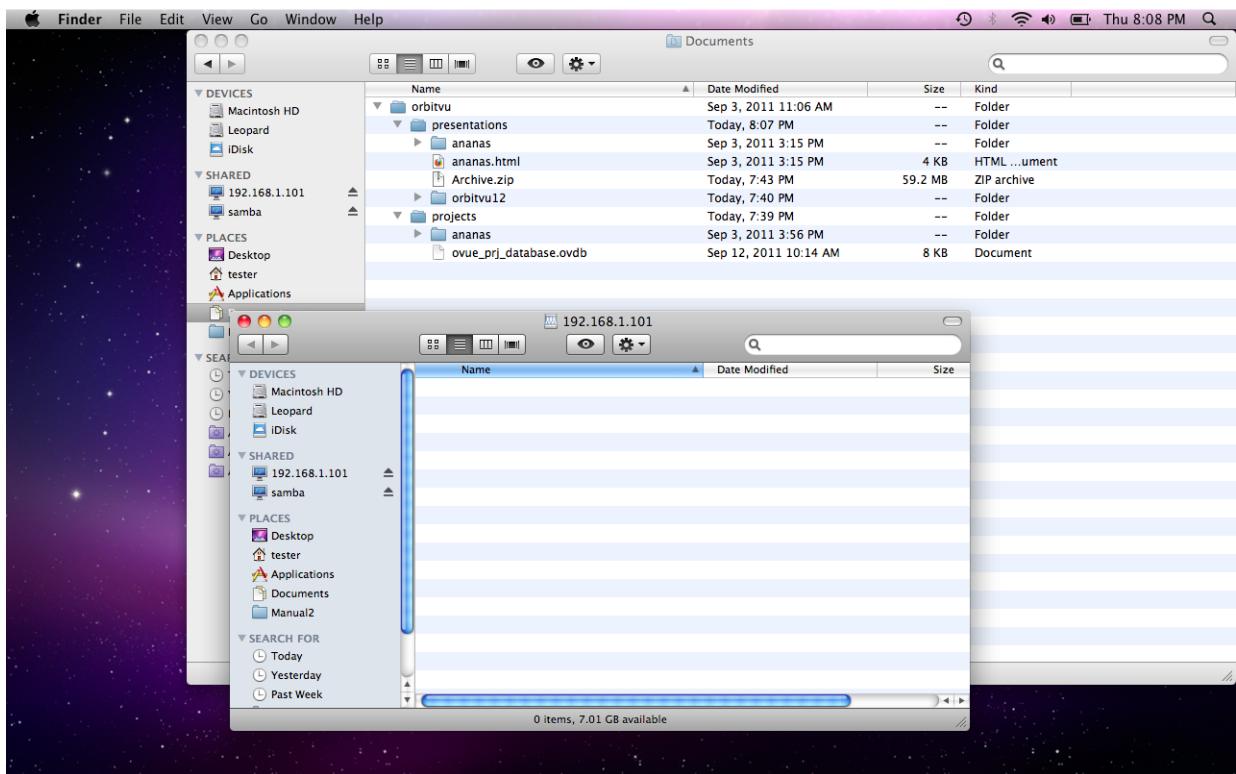
Goodreader offers several ways to upload files. One of the most convenient is to use a WiFi connection. Open Goodreader on iPad and tap on WiFi icon on the bottom bar.



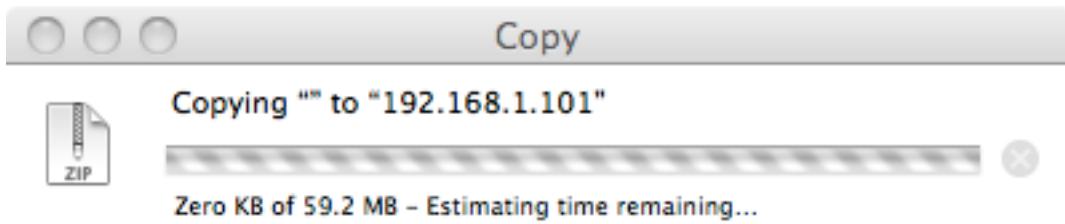
In the popup window, you can see IP-address – here <http://192.168.1.101:8080>. Go to Mac Finder -> Go -> Connect to server... In the popup window enter the above address.



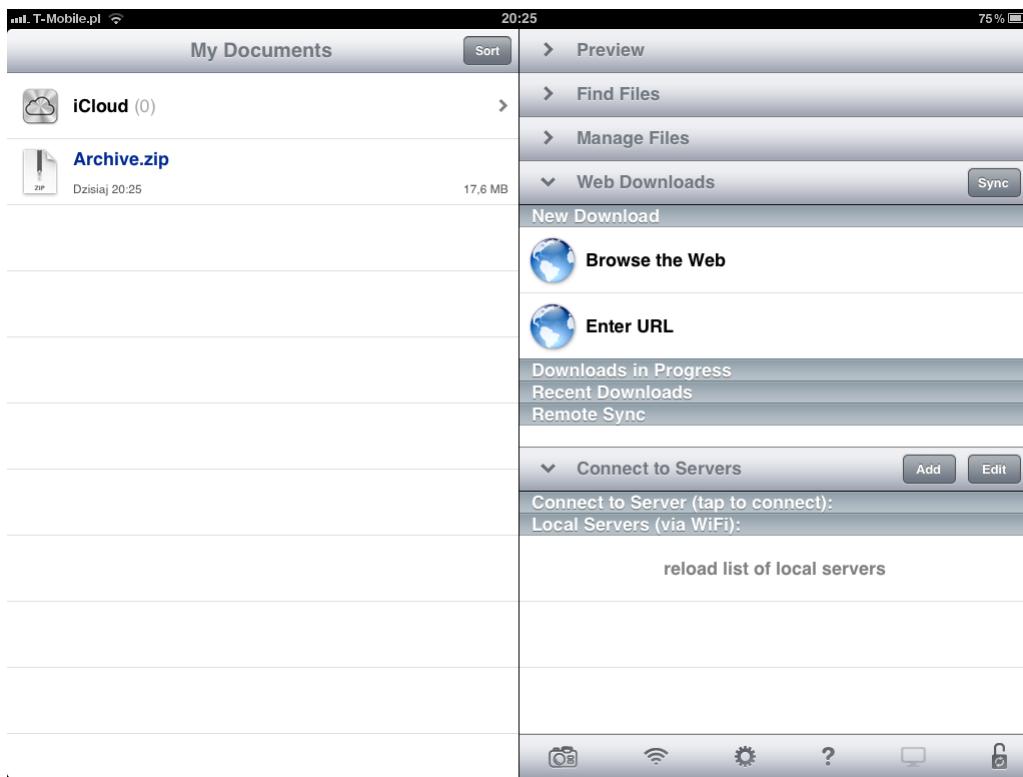
A new Finder window will pop-up. You can now drag Archive.zip file into the new window. It will copy your zip file into Goodreader.



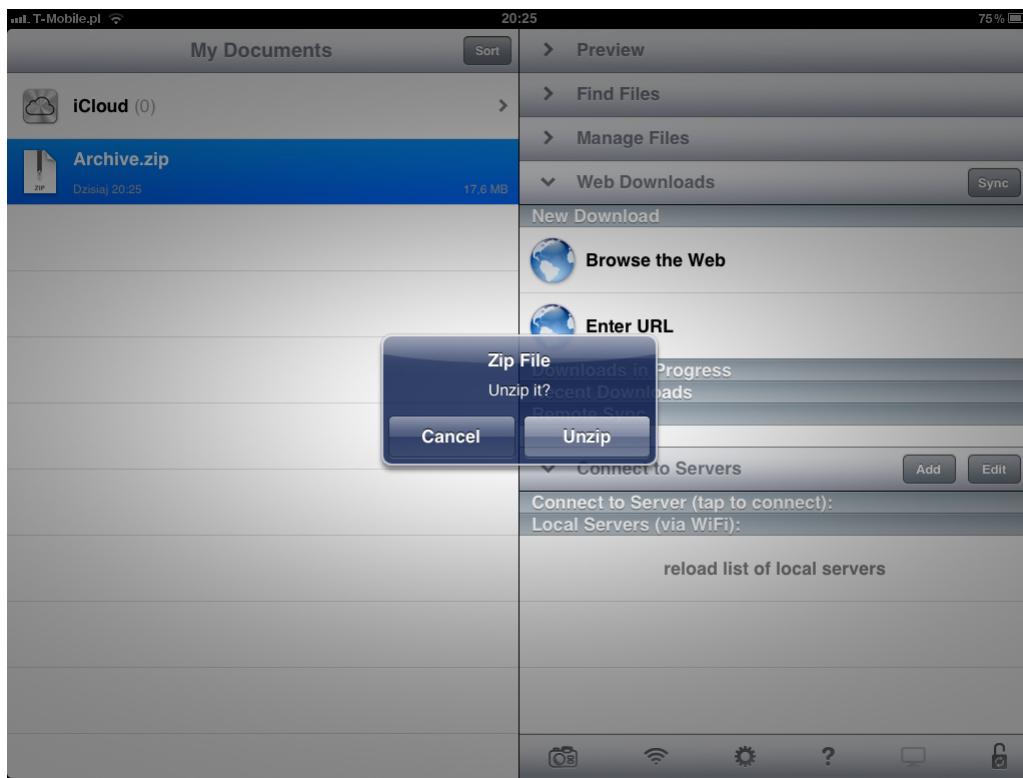
Copying window should pop-up.



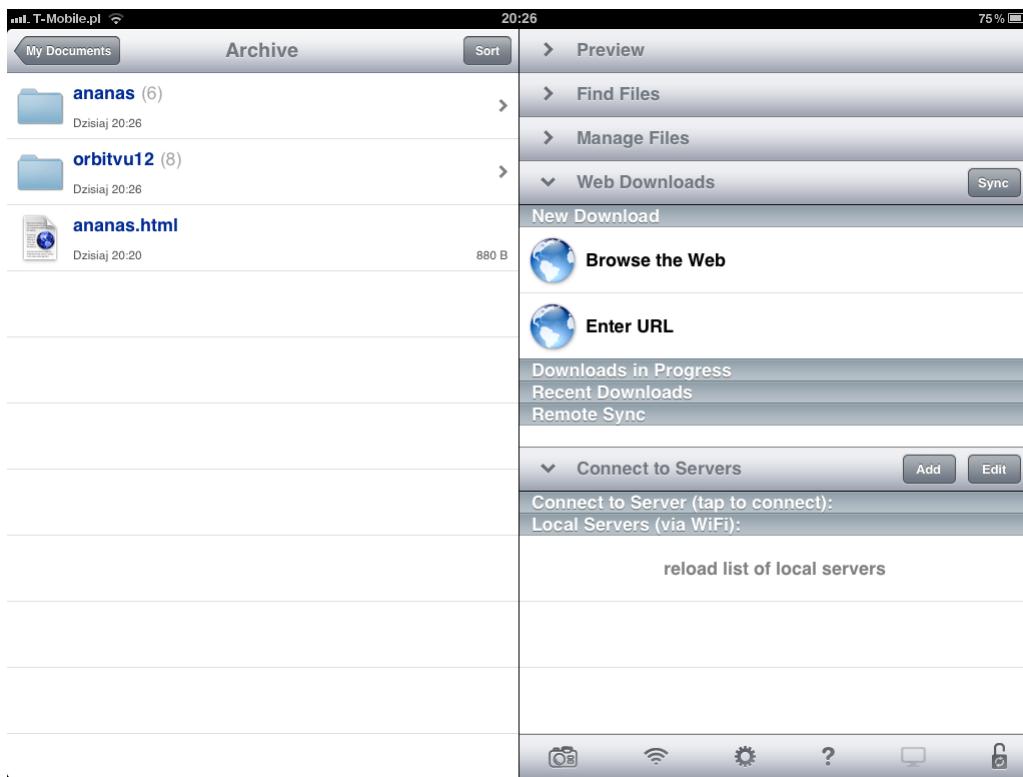
As soon as copying is done go back to iPad and close wifi connection (STOP button in upper right corner). You should see the Archive.zip file in Goodreader now.



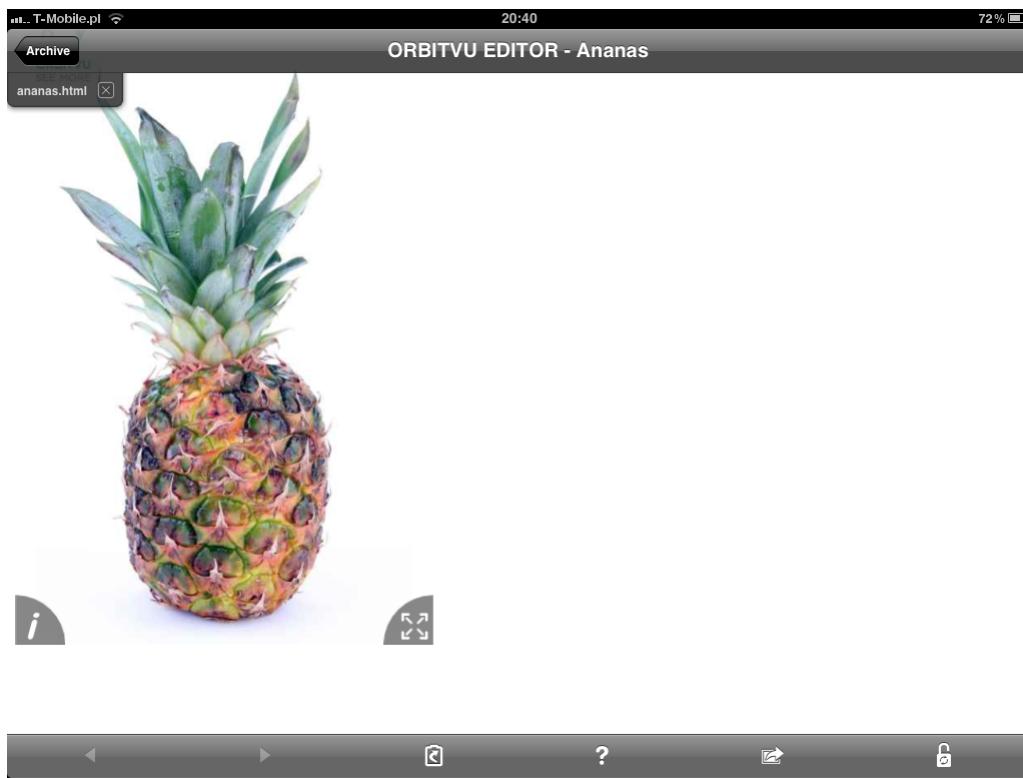
Tap on the file to unzip it.



You should see new Archive folder inside Goodreader. Tap the folder name and you should see your ananas presentation.



Tap on **ananas.html** file. It should open the presentation in the new preview window!



 You can use other ways to transfer the zip file to iPad. One of the most convenient

ways is to use SUN server and download as HTML action. Just log in to SUN server, click on the link and download it directly into Goodreader.

Integration examples

Embedding with ORBITVU SUN embed codes

ORBITVU SUN is a platform that lets you manage, reuse and redistribute your ORBITVU Presentations effortlessly. You can set necessary parameters from its interface and just copy & paste desired embed code into your site.

If you need further adjusting it is also possible by tweaking [parameter values](#). Below you'll find some typical embed codes generated for you by ORBITVU SUN

Standard embed code

```
<!DOCTYPE html>
<html>
<body>
    // YOUR SITE CODE HERE
    <script type="text/javascript" src="//orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHp/ov3601/3/script?width=auto&content2=yes&height=auto"></script>
    <div class="orbitvu-viewer" style="width: 250px; height: 250px">
        <div id="ovContent-dSz54kMJ7rHKtC6E2YBHp" class="orbitvu-container" style="background: url("//static.orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHp/iprocs/250/250/img01.png) no-repeat center center #ffffff;"></div>
    </div>
    // YOUR SITE CODE HERE
</body>
</html>
```

QR Code

```
<a href="https://orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHp/ov3601/3/view" target="_blank">
    
</a>
```



Image (rotated)

```

```



Max scale

Here we add `max_scale` to limit maximum possible zoom and `viewer_uid` to make it possible to use the same presentation multiple times on one page. Compare how much you can zoom here with the [first example](#) at this page.

```
<!DOCTYPE html>
<html>
<body>
    // YOUR SITE CODE HERE
    <script type="text/javascript" src="//orbitvu.co/001/dSz54kJ7rHKtC6E2YBHpf/ov3601/3/script?width=auto&content2=yes&height=auto&max_scale=0.15&viewer_uid=xl2&force_html5=yes"></script>
    <div class="orbitvu-viewer" style="border:1px solid #444444; width: 100%; height: 250px">
        <div id="ovContent-xl2" class="orbitvu-container" style="background: url("//static.orbitvu.co/001/dSz54kJ7rHKtC6E2YBHpf/iprof/250/250/img01.png) no-repeat center center #ffffff;"></div>
    </div>
    // YOUR SITE CODE HERE
</body>
</html>
```

HTML5/Flash mode

Using [force_html5](#) to enable/disable Flash.

```
<!DOCTYPE html>
<html>
<body>
    // YOUR SITE CODE HERE
    <script type="text/javascript" src="//orbitvu.co/001/dSz54kJ7rHKtC6E2YBHpf/ov3601/
3/script?width=auto&content2=yes&height=auto&force_html5=yes&viewer_uid=x13"></script>
    <div class="orbitvu-viewer" style="border: 1px solid #444444; width: 100%; height:
250px">
        <div id="ovContent-x13" class="orbitvu-container" style="background:
url("//static.orbitvu.co/001/dSz54kJ7rHKtC6E2YBHpf/proc/250/250/img01.png) no-repeat
center center #ffffff;"></div>
    </div>
    // YOUR SITE CODE HERE
</body>
</html>
```

HTML5 initial info

Using [force_html5](#) to disable Flash and [html5_initial_info](#) to show info window.

```
<!DOCTYPE html>
<html>
<body>
    // YOUR SITE CODE HERE
    <script type="text/javascript" src="//orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHpf/ov3601/
3/script?width=auto&content2=yes&height=auto&force_html5=yes&viewer_uid=x13_2&
html5_initial_info=all"></script>
    <div class="orbitvu-viewer" style="border: 1px solid #444444; width: 100%; height:
250px">
        <div id="ovContent-x13_2" class="orbitvu-container" style="background:
url("//static.orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHpf/proc/250/250/img01.png) no-repeat
center center #ffffff;"></div>
    </div>
    // YOUR SITE CODE HERE
</body>
</html>
```

HTML5 hint

Using [force_html5](#) to disable Flash and [html5_hint](#) to show infographic.

```
<!DOCTYPE html>
<html>
<body>
    // YOUR SITE CODE HERE
    <script type="text/javascript" src="//orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHpf/ov3601/
3/script?width=auto&content2=yes&height=auto&force_html5=yes&viewer_uid=x13_3&
html5_hint=yes"></script>
    <div class="orbitvu-viewer" style="border: 1px solid #444444; width: 100%; height:
250px">
```

```

<div id="ovContent-x13_3" class="orbitvu-container" style="background:
url(//static.orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHpf/proc/250/250/img01.png) no-repeat
center center #ffffff;"></div>
</div>
// YOUR SITE CODE HERE
</body>
</html>

```

Delayed execution

You can defer injection of the presentation into the page using [viewer_delayed](#) parameter. For example, you might want to show presentation on button click (note that this requires some programming knowledge).

```

<!DOCTYPE html>
<html>
<body>
// YOUR SITE CODE HERE
<script type="text/javascript" src="//orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHpf/ov3601/
3/script?width=auto&content2=yes&height=auto&viewer_delayed=yes&
viewer_uid=x14"></script>
<div class="orbitvu-viewer" style="border: 1px solid #444444; width: 100%; height:
250px">
<div id="ovContent-x14" class="orbitvu-container"></div>
</div>

<button onclick="inject_viewer_x14()">Inject presentation</button>
// YOUR SITE CODE HERE
</body>
</html>

```

Inject presentation

Responsive VIEWER within hidden tab

Sometimes you might want to place ORBITVU VIEWER into the separate tab that is initially hidden and you still want ORBITVU VIEWER to scale responsively. Starting from version 3.0.32 of ORBITVU VIEWER such situation will be handled automatically and VIEWER will delay it's initialization until it is shown.

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="//code.jquery.com/ui/1.11.4/themes/smoothness/jquery-
ui.css">
    <style type="text/css">
        #tabs-container {
            height: 300px;
        }

        /* make tabs width responsive */
        #tabs {
            width: 200px;
            height: 100%;
            box-sizing: border-box;
        }

        @media (min-width:800px) {
            #tabs {
                width: 300px;
            }
        }

        @media (min-width:1000px) {
            #tabs {
                width: 350px;
            }
        }
    </style>
</head>
<body>
    <div id="tabs-container">
        <ul>
            <li><a href="#tab1">Tab 1</a></li>
            <li><a href="#tab2">Tab 2</a></li>
            <li><a href="#tab3">Tab 3</a></li>
        </ul>
        <div id="tab1">Content for Tab 1</div>
        <div id="tab2">Content for Tab 2</div>
        <div id="tab3">Content for Tab 3</div>
    </div>
</body>
</html>
```

```

}

@media (min-width:1200px) {
    #tabs {
        width: 500px;
    }
}

</style>
</head>
<body>

// YOUR SITE CODE HERE

<div id="tabs-container" > <!-- use container to set tabs height -->
    <div id="tabs">
        <ul>
            <li><a href="#tabs-1">First tab</a></li>
            <li><a href="#tabs-2">360&deg;</a></li>
        </ul>
        <div id="tabs-1">
            <p>Switch to second tab to see 360&deg; view</p>
        </div>
        <div id="tabs-2">
            <div id="ovContent-x17" class="orbitvu-container"></div>
        </div>
    </div>
</div>
<!-- Use jQuery UI tabs plugin -->
<script type="text/javascript" src="https://code.jquery.com/jquery-1.12.1.min.js"></script>
<script type="text/javascript" src="https://code.jquery.com/ui/1.11.4/jquery-ui.min.js"></script>
<script type="text/javascript">
    // jQuery tabs
    $("#tabs").tabs({
        heightStyle: "fill"
    });
</script>
<script type="text/javascript" src="//orbitvu.co/001/dSz54kJ7rHKtC6E2YBHp/ov3601/3/script?width=auto&content2=yes&height=auto&force_html5=yes"></script>
// YOUR SITE CODE HERE
</body>
</html>

```

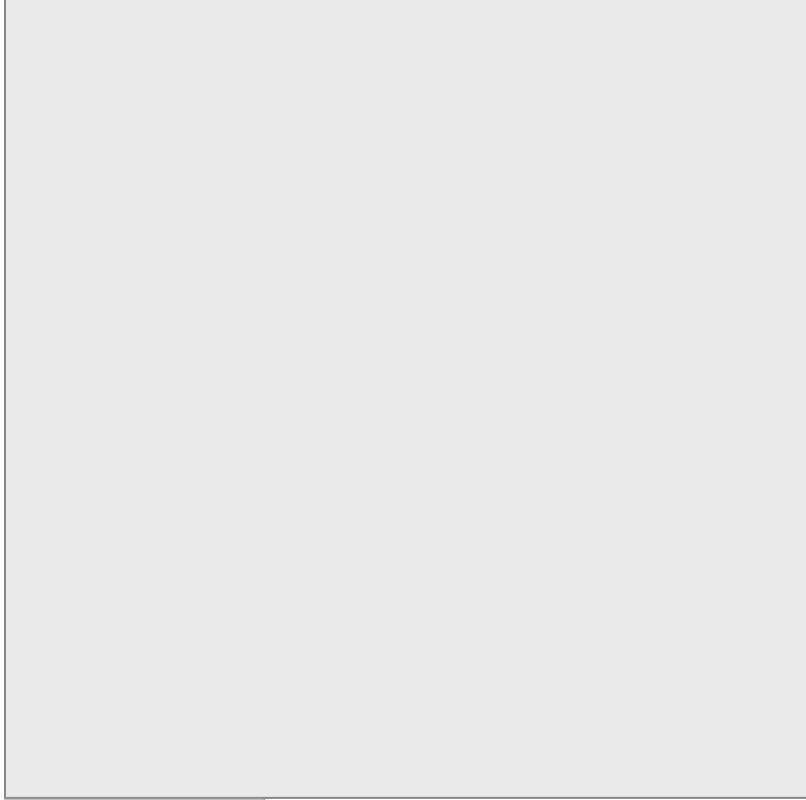
- [First tab](#)
- [360°](#)

Switch to second tab to
see 360° view

ORBITTOUR in a hidden container

If you need to initialize ORBITTOUR in a hidden container that will be shown dynamically later (e.g. on button click), then you do it as usual. ORBITTOUR will detect dynamically when it is shown and will adjust its size properly. Another option is to use [orbittour_width](#) and [orbittour_height](#) parameters that will force ORBITTOUR dimensions.

```
<!DOCTYPE html>
<html>
<body>
    // YOUR SITE CODE HERE
    <div class="orbitwrapper" style="width: 400px; height: 400px; background-color: #eaeaea; border: 1px solid #888888;">
        <div id="hidden-orbittour-wrapper" style="display: none; height: 100%;">
            <div id="orbittour-dSz54kJ7rHKtC6E2YBHp" class="orbittour-container"></div>
            <script type="text/javascript" src="//orbitvu.co/001/dSz54kJ7rHKtC6E2YBHp/2/orbittour/468/script?force_html5=yes"></script>
        </div>
    </div>
    <button onclick="document.getElementById('hidden-orbittour-wrapper').style.display='block';">Show orbittour</button>
    // YOUR SITE CODE HERE
</body>
</html>
```



Show orbittour

Responsive layout

It is possible to use ORBITVU VIEWER in responsive layouts using [width](#) and [height](#) set to [auto](#). ORBITVU VIEWER will fit itself into container element.

Container with height of 0px. ORBITVU VIEWER will be shown using 16:9 proportion:

```
<!DOCTYPE html>
<html>
<body>
    // YOUR SITE CODE HERE
    <script type="text/javascript" src="//orbitvu.co/001/dSz54kJ7rHKtC6E2YBpf/ov3601/
3/script?width=auto&content2=yes&height=auto&viewer_uid=x15&force_html5=yes"></script>
    <div class="orbitvu-viewer" style="width: 100%; height: 100%">
        <div id="ovContent-x15" class="orbitvu-container"></div>
    </div>
    // YOUR SITE CODE HERE
</body>
</html>
```

Container element with height set using media queries.

```
<!DOCTYPE html>
<html>
<body>
```

```

// YOUR SITE CODE HERE
<style type="text/css">
.media-query-viewer-wrapper {
    height: 100px;
}

@media (min-height:200px) {
.media-query-viewer-wrapper {
    height: 200px;
}
}

@media (min-height:300px) {
.media-query-viewer-wrapper {
    height: 300px;
}
}

</style>

<div class="media-query-viewer-wrapper" style="width: 50%;">
<script type="text/javascript" src="//orbitvu.co/001/dSz54kMJ7rHktC6E2YBpf/
ov3601/3/script?width=auto&content2=yes&height=auto&viewer_uid=x16&
force_html5=yes"></script>
<div class="orbitvu-viewer" style="width: 100%; height: 100%">
<div id="ovContent-x16" class="orbitvu-container"></div>
</div>
</div>
// YOUR SITE CODE HERE
</body>
</html>

```

Parameters reference

How to customize ORBITVU VIEWER with parameters

ORBITVU VIEWER behaviour can be customised in many ways using [parameters](#). Parameters can be applied to JavaScript code or embed codes/preview URLs from ORBITVU SUN.

Below you can find examples of adding force_html5=yes parameter:

ORBITVU SUN presentation preview URL

```
http://orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/ov3601/2/view?force_html5=yes
```

ORBITVU SUN Presentation embed code

```
<div class="orbitvu-viewer" style="background: url(/static.orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/600/600/img01.png) no-repeat center center #ffffff; width: 600px; height: 600px">
  <script type="text/javascript" src="//orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/ov3601/3/script?width=600&content2=yes&height=600&force_html5=yes"></script>
</div>
```

- i** Some additional parameters that can be seen in the embed code, e.g. content2 are automatically generated [ORBITVU SUN](#) parameters, not ORBITVU VIEWER ones.

ORBITTOUR embed code

```
<div id="orbittour-eVejoY36KsRzEWVXoqmPcG" class="orbittour-container"></div>
<script type="text/javascript" src="//orbitvu.co/001/eVejoY36KsRzEWVXoqmPcG/2/orbittour/689/script?force_html5=yes"></script>
```

JavaScript code

```
inject_orbitvu(
  'presentation1-container',
  'http://yourdomain.com/orbitvu12/orbitvuer12.swf',
  'http://yourdomain.com/orbitvu12/expressInstall.swf',
  {
    ovus_folder: "http://yourdomain.com/presentations/presentation1/",
    width: "500",
    height: "400",
```

```
        force_html5: 'yes'  
    }) ;
```

ORBITVU VIEWER parameters reference

auto_drag_switch

Defines if ORBITVU VIEWER should switch automatically to drag mode on zoom. If set to no then ORBITVU VIEWER operates in rotate mode.

Version: 2.3.32+

Values: yes | no

Default: yes

auto_rotate_dir

Autorotation direction

Values: left | right

Default: rotate_dir

background_color

Background color

Default: 0xffffffff (white)

border

Viewer window border

Values: yes | no

Default: yes

border_color

Border color

Default: 0xffffffff (white)

branding

Text that will be shown in a top right corner inside VIEWER area. This is related to branding_url parameter

Value: any text

branding_url

URL of the page that user will be redirected to when clicking branding button (related to branding parameter)

buttons_type

Type of buttons shown in the Viewer. There are two types of buttons: desktop buttons and mobile buttons. Mobile ones are designed for use with mobile devices so there is no zoom in/zoom out buttons (as you're supposed to use pinch gesture to zoom) and no drag/rotate switch as you can drag with two fingers and rotate with single finger, etc. Also The layout of mobile buttons is different. The value of auto will automatically select mobile buttons for mobile devices.

Values: auto | desktop | mobile

Default: auto

Version: 3.1.1 +

config_url

URL of config.xml file

Default: <ovus_folder>/config.xml

content2

Defines if content.xml or content2.xml configuration file should be used. Unless you have very old presentations you should always set it to yes.

Values: yes | no

Default: no

 If you're not sure if you should set yes or no then check your presentation directory to see if it contains content2.xml file. If yes, then use yes.

continuous

Continuous rotation – useful for half rotations or animations

Values: yes | no

Default: yes

disable_nosize

If ORBITVU VIEWER is initialized without specific height and containing element has a height of 0px then by default, ORBITVU VIEWER will use a height of 56.25% (16:9 proportion). You can disable this behaviour by setting `disable_nosize` to yes.

Values: yes | no

Default: no

doubletap_mode

When you double tap then ORBITVU VIEWER can:

fullscreen - switch to fullscreen

zoom - zoom in/zoom out, depending on current zoom level

disabled - do nothing

Values: fullscreen | zoom | disabled

Default: fullscreen

VIEWER mode: HTML5

external_access_init

This parameter is DEPRECATED - use [viewer_api_init](#) instead!

Name of JavaScript callback (function) that will be called after initialization of ORBITVU VIEWER.

API object will be passed to the callback.

Default: not set

Release: Royalty Free, Corporate, DEMO

i The value of this parameter should be a String, not a function itself.

i In Royalty Free, Corporate and DEMO versions of ORBITVU VIEWER it is possible to get programming access to its methods. For more details check out [API documentation](#).

⚠ USE [viewer_api_init](#) instead of external_access_init for VIEWER version 2.3.33+

first_frame

First frame to display

Values: 1 - no. of frames

Default: 1

first_row

First row to display in 3D presentation

Version: 2.3.31+

Values: 1 - no. of rows

Default: 1

force_html5

Force ORBITVU VIEWER to work in HTML5 mode

Values: yes | no

Default: no

i ORBITVU VIEWER automatically detects if Flash is installed and enabled. If not, then HTML5 version is used.

frame_rate

Auto rotation speed [frames/sec]

Default: 12

fullscreen_enabled

Disable fullscreen mode. This will also cause that fullscreen button is not shown

Values: yes | no

Default: yes

gui_background

The background colour of GUI elements like buttons or loader.

Example: 0x00ff00

Value is in a form of 0x<hex> where <hex> is a colour in hexadecimal notation.

gui_foreground

Foreground colour of GUI elements like buttons or loader

Example: 0x00ff00

Value is in a form of 0x<hex> where <hex> is a colour in hexadecimal notation.

gui_foreground_active

Foreground colour of active (eg. being clicked) GUI elements like buttons

Example: 0x00ff00

height

Height of the presentation

Values: integer value or 'auto'

Default: 300

 The integer value is in pixels; auto means 100%

 If auto is used and container element, that VIEWER is inserted into, has height of 0, then height will be set in 16 : 9 proportion to VIEWER's width

html5_hint

Viewer in HTML5 mode can display small infographic that illustrates the possibility of rotation in one (360°) or two (3D) axes. Infographic is hidden during any VIEWER interaction. It shows up again with 5 seconds delay. See [example](#)

Version: 2.3.21+

Values: yes | no

Default: no

VIEWER mode: HTML5

html5_initial_info

For HTML5 mode you can force VIEWER to show info window on start. Info window presents simple infographics illustrating possible touch/mouse actions to manipulate VIEWER. It is possible to have this window shown only on touch devices (value of **touch**) or desktop devices (value of **desktop**) or both, desktop and touch devices (value of **all**). See [example](#)

Version: 2.3.18+

Values: no | desktop | touch | all

Default: no

VIEWER mode: HTML5

image_folder

URL of images folder for 360 degree presentation

Default: <ovus_folder>/images/

lang

Tooltip language. The value of '_' means that no tooltip will be shown over buttons.

Values: cs | de | dk | en | es | fi | fr | fy | hu | it | lt | no | pl | pt | ro | ru | se | sk | sl | sr | _ | auto

Default: en

 **auto** is only available with SUN Server

loader

Show or hide loader visible on VIEWER initialization

VIEWER mode: HTML5

Version: 2.3.33+

Values: yes | no

logo_src

Link to the image that will be shown in top left corner of the VIEWER area, eg. Buy button. This is related to [logo_url](#) parameter

logo_url

URL of the page that user will be redirected to when clicking Logo image (specified with [logo_src](#) parameter)

max_scale

Maximum possible zoom scale of presentation.

Values: 0 - 1

Example: 0.28

Default: not set

How to calculate scale:

If your presentation maximum size is: 5448px x 3607px then get larger value (5448) and desired maximum value (eg. 100px). Then divide it to get max_scale: $100/5448=0.0184$ $100/3607=0.0277$

 The value of 0 is a special one. It will cause that initial scale (one that fits best presentation container) will be used. The result is that presentation is shown as usual but it is not possible to zoom.

mousewheel

Enable (yes) or disable (no) mouse wheel zoom

Values: yes | no

Default: yes

Version: 2.3.24+

- i** There are API methods that let you switch this parameter dynamically: [enable_mousewheel](#) and [disable_mousewheel](#).

ovskin

Show ORBITVU identity (within info popup and flash context menu)

Values: yes | no

Default: no

ovus_folder

URL of ORBITVU Presentation folder (containing unpacked .OVUS file)

Default: BASE folder - default url for the HTML page that might be specified using <base> tag

partial_load

When enabled Orbitvu VIEWER will be ready for a user interaction as soon as 4 initial frames are loaded. Then it will load rest of the frames in the background.

Default: no

Version: 3.0.30+

Values: yes | no

preload_width

Force initial image preload width. By default initially loaded images fit container size and while zooming in larger images are loaded. With this option, you can load bigger images immediately.

Default: 0

preload_height

Force initial image preload height

Default: 0

report_bugs

Debugging mode – report exceptions in VIEWER window

Values: yes | no

Default: no

VIEWER mode: Flash

rotate_dir

Rotation direction

Values: right | left

Default: right

social

Defines if social (facebook, twitter, e-mail) buttons are shown.

Values: yes | no

Default: no

social_url

URL of the page that will be published to social networks

Default: At SUN Server it is a direct link to presentation

style

User interface (buttons) style.

0 – default round style

1 – square style

2 – round style with rotation scroll

3 – square style with rotation scroll

4 – no buttons at all

Values: 0 | 1 | 2 | 3 | 4

Default: 1

teaser

To encourage end users to interact with 360° presentations you can set the initial behavior of the viewer:

- **zoomin** – presentation is preloaded and slightly zooms in when mouse over viewer window (Flash version only)
- **autorotate** – presentation is preloaded and starts with auto-rotation
- **play** – presentation is not preloaded starts with play button, after pressing play button it is preloaded and starts to auto rotate
- **playonerotation** – 2.3.17 – presentation is not preloaded starts with play button, after pressing play button it is preloaded and does one rotation
- **playonerotationslow** – 2.3.17 – presentation is not preloaded starts with play button, after pressing play button it is preloaded and does one slow rotation
- **static** – presentation is preloaded but nothing happens
- **onerotation** - presentation is preloaded and just after loading it does quick single rotation
- **onerotationslow** - presentation is preloaded and just after loading it does single rotation (slower than onerotation)

Values: autorotate | zoomin | play | playonerotation | playonerotationslow | static | onerotation | onerotationslow

Default: static

teaserplay_overlay_color

The color of an overlay of teaser with value: play | playonerotation | playonerotationslow

Example: 0xFFFFF8C

Value is in a form of 0x<hex> where <hex> is a colour in hexadecimal notation.

FFFFFF = white color; The last two positions define opacity of an overlay [8C = 55% opacity];

Default: 0x00000099 (black with 60% opacity)

Version: 3.4.0+

transition

Between frame transition filter

Values: alpha | blur | no

Default: no

VIEWER mode: Flash

width

Width of the presentation

Values: integer value or 'auto'

Default: 400

 The integer value is in pixels; auto means 100%

wmode

Adobe Flash wmode option: [see more](#)

Default: not set

VIEWER mode: Flash

xml_url

URL of content2.xml file

Default: <ovus_folder>/content2.xml

 For older presentations, it is possible that filename will be content.xml. This is determined by the content2 parameter.

valign

Presentation vertical alignment

Values: top | center | bottom

Default: center

VIEWER mode: Flash

viewer_api_init

Name of JavaScript callback (function) that will be passed API object before initialization of ORBITVU VIEWER

This callback is called as soon as ORBITVU VIEWER starts loading so that it is possible to listen to loading events (as opposed to external_access_init)

Default: not set

Version: 2.3.33+

Release: Royalty Free, Corporate, DEMO

i In Royalty Free, Corporate and DEMO versions of ORBITVU VIEWER it is possible to get programming access to its methods. For more details check out [API documentation](#).

i The value of this parameter should be a String, not the function itself.

vmove

Enable vertical movement during manual rotation

Values: yes | no

Default: yes

vmove_dir

Vertical rotation direction for 3D presentations

Values: down | up

Default: down

Version: 2.3.18+

Customizing ORBITVU SUN embed codes

When using ORBITVU SUN embed codes it is possible to fine tune generated code to suit your needs. This might be necessary in some cases, e.g. when trying to embed the same presentation multiple times on a single page or if you want to dynamically load/inject presentation using JavaScript.

There are two main things to consider:

- container element of the presentation and it's id

This is the element that presentation will be inserted into. Usually, it is a **div** element that you have to place somewhere on your page. Default id of container element is:

ovContent-<UID>

like:

```
<div id="ovContent-myuid" class="orbitvu-container"></div>
```

You might want to use your own id instead of the default one.

- JavaScript function used to inject presentation

The default name of inject function is:

inject_viewer_-<UID>

If using the same presentation multiple times on single page you must have this name unique for each presentation. You might also want to have the inject function called on demand, not immediately on page load.

ORBITVU VIEWER parameters can be applied to JavaScript code or to embed code/preview code URLs from ORBITVU SUN.

An Example of adding viewer_uid=myuid parameter:

```
<script type="text/javascript" src="//orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/ov3601/3/script?width=auto&content2=yes&height=auto&viewer_uid=myuid"></script>
<div class="orbitvu-viewer" style="background: url("//static.orbitvu.co/001/pcFGPQAZSh3mJWrPGezuBP/iprof/600/600/img01.png) no-repeat center center #ffffff; width: 600px; height: 600px">
    <div id="ovContent-myuid" class="orbitvu-container"></div>
</div>
```

Below you'll find a list of available parameters

viewer_uid

You can change default value of container id and inject function name (JavaScript) by setting new UID

Default: <UID> - uid of ORBITVU SUN presentation

Value: string that is both valid JavaScript function name and HTML id attribute

- i** If you set this parameter to myuid the name of the inject function will be: `inject_viewer_myuid` and id of the referenced container element will be `ovContent-myuid`. You have to change id of the container element in your HTML code.

viewer_delayed

If you want to delay the moment presentation is injected into the page then you can set viewer_delayed to yes and then call inject function by yourself when you need. The default name of the inject function is: `inject_viewer_<UID>`, but can be changed with viewer_uid parameter.

Values: yes | no

Default: no

custom_viewer

If you have your own VIEWER license, e.g. for a customised VIEWER look, then you can use it with Orbitvu SUN. You have to visit your Orbitvu SUN account preferences and upload your custom VIEWER (zip archive). Since then you can add `custom_viewer=yes` to embed codes to use your viewer.

Values: yes | no

Default: no

Customizing ORBITTOUR embed code

When using ORBITTOUR embed codes it is possible to customise generated code. This might be necessary in some cases, e.g. when trying to embed the same ORBITTOUR multiple times on the single page or if you want to dynamically load/inject ORBITTOUR using JavaScript.

With ORBITTOUR you can also use most of ORBITVU VIEWER [parameters](#).

ORBITTOUR parameters can be applied to embed code generated by ORBITVU SUN.

An example of adding the [force_html5](#) parameter:

```
<div id="orbittour-hhyoAwhnuh2NQLgNfpdRYk" class="orbittour-container"></div>
<script type="text/javascript" src="//orbitvu.co/001/hhyoAwhnuh2NQLgNfpdRYk/2/orbittour/
523/script?force_html5=yes"></script>
```

background_color

Set the background colour of ORBITTOUR. The value of this parameter should be defined as a hexadecimal colour number (same as used in HTML) + hexadecimal opacity value, eg:

- 000000ff
color code: 000000 (black) and opacity value: ff (opaque)
- 00000000 (transparent)
color code: 000000 and opacity value: 00 (transparent)

orbittour_uid

Set it to unique value if you'd like to have the same ORBITTOUR embedded multiple times on the single page. This will cause that both ORBITTOUR container element id and ORBITTOUR inject function (JavaScript) will have unique names across the page.

Default: unique ORBITVU SUN presentation uid

Example: myuid_1

orbittour_container_id

You can explicitly set **id** of the ORBITTOUR container element

Default: orbittour-<orbittour_uid>

preferred_language

You can force ORBITTOUR to use specific language (if only ORBITTOUR has translations for this language)

Default: not set - language will be automatically chosen according to browser language preference

Value: language code from the list of available ORBITTOUR translations, e.g. en, de

orbittour_width

ORBITTOUR tries to automatically calculate its size according to the container dimensions but in some cases, it might be required to explicitly set **width** and **height**. For example, if ORBITTOUR is initialized inside a hidden container (with **display: none**) then it will not be possible to calculate proper dimensions.

Default: not set - calculated automatically

orbittour_height

ORBITTOUR tries to automatically calculate its size according to the container dimensions but in some cases, it might be required to explicitly set **width** and **height**. For example, if ORBITTOUR is initialized inside a hidden container (with **display: none**) then it will not be possible to calculate proper dimensions.

Default: not set - calculated automatically

orbittour_delayed

By default, ORBITTOUR will be immediately injected into your web page, but if you want to inject it later, then you can block immediate injection using [orbittour_delayed](#) parameter and then call inject function programmatically when needed. The function that injects ORBITTOUR is:

```
inject_orbittour_(<orbittour_id>)
```

Values: yes | no

Default: no

orbittour_style

Change style used by ORBITTOUR:

Values: default | right_views | noviews

Default: uses value set for ORBITTOUR at ORBITVU SUN

custom_viewer

If you have your own VIEWER license, e.g. for a customised VIEWER look, then you can use it with ORBITVU SUN. You have to visit your Orbitvu SUN account preferences and upload your

custom VIEWER (zip archive). From then you can add `custom_viewer=yes` to embed codes to use your copy of ORBITVU VIEWER.

Values: yes | no

Default: no

external_init

Name of the globally available JavaScript function that will be called after ORBITTOUR is initialized. The function will be passed API object that provides some ORBITTOUR manipulation methods, e.g.:

```
<script type="text/javascript">
  var orbittour_api = null;
  window['my_orbit_init'] = function(orbit_api){
    orbittour_api = orbit_api;
  }

  // ...
  function call_on_dynamic_resize(){
    if (orbittour_api){
      orbittour_api.handle_resize();
    }
  }
</script>
```

The method name is passed like:

```
<div id="orbittour-hhyoAwhnuh2NQLgNfpdRYk" class="orbittour-container"></div>
<script type="text/javascript" src="//orbitvu.co/001/hhyoAwhnuh2NQLgNfpdRYk/2/orbittour/
523/script?external_init=my_orbit_init"></script>
```

play_type

If set to the value other than `none` will cause play/pause button to be shown. When pressed, ORBITTOUR will start animating between subsequent views or hotspots (depending on [play_type](#) value). Note that if there are no views or hotspots then play button will not be shown.

Values: none | views | hotspots

Default: no

play_time

How long to wait between moving to next view or hotspot (in milliseconds) when animation is played ([play_type](#) is set to views or hotspots)

Values: 2500 | 3000 | 3500 | 4000 | 4500 | 5000 | 7500 | 10000

Default: 2500

play_auto

Enable autoplay of ORBITTOUR (if [play_type](#) is not set to none). Playing will start 1 second after presentation is loaded

Values: yes | no

Default: no

ORBITVU VIEWER API

In Royalty Free, Corporate and DEMO versions of ORBITVU VIEWER it is possible to control ORBITVU VIEWER programmatically, using JavaScript.

To get access to ORBITVU VIEWER API object you have to do the following:

- define a global callback function (JavaScript)
- set viewer_api_init parameter to the name of this function

Sample code:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />

        <script type="text/javascript">
            // Global variable to store VIEWER API object reference
            var viewer_api_obj = null;

            // Callback called while VIEWER is loading
            window['my_loading_callback'] = function(progress){
                console.log('progress: ' + progress);
            }

            // Callback called when viewer is initialized (all required frames are loaded)
            // From this point it is possible to manipulate it, eg. rotate/zoom
            window['my_viewer_initialized_callback'] = function(){
                // as soon as viewer is initialized rotate to 10-th frame
                viewer_api_obj.setScene({hangle: 10});
            }

            // callback called on API initialization used to set another callbacks and store API object reference
            window['my_api_callback_name'] = function(api_obj) {
                viewer_api_obj = api_obj;
                api_obj.addCallback('my_viewer_initialized_callback',
                    'viewer_initialized');
                api_obj.addCallback('my_loading_callback', 'loading');
            };
        </script>
    </head>
    <body>
        // YOUR SITE CODE HERE
    </body>

```

```

<div class="orbitvu-viewer" style="background: url(/static.orbitvu.co/001/CmWVYLSjZz3nh3FvjzepBQ/300/350/img01.png) no-repeat center center #ffffff; width: 350px; height: 300px">
    <script type="text/javascript" src="//orbitvu.co/001/CmWVYLSjZz3nh3FvjzepBQ/ov3601/2/script?newembed=yes&width=350&content2=yes&report_bugs=no&height=300&fromscript=1&viewer_api_init=my_api_callback_name">
        </script>

    </div>
    // YOUR SITE CODE HERE
</body>
</html>

```

It is also possible to use [external_access_init](#) parameter but it is deprecated since VIEWER 2.3.33 in favour of [viewer_api_init](#). Sample code for [external_access_init](#):

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />

        <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
        <script type="text/javascript">
            var viewer_api_obj = null;
            window['my_api_callback_name'] = function(local_viewer) {
                /* STARTING from ORBITVU VIEWER 2.3.3 it is not required to use code below */
                /* start: viewer <= 2.3.2*/
                if (!local_viewer) {
                    // specify id of VIEWER Flash object
                    local_viewer = document.getElementById('ovContent-dSz54kMJ7rHKtC6E2YBHpf-obj');
                }
                /* end: viewer <= 2.3.2*/

                viewer_api_obj = local_viewer;
                local_viewer.setScene({hangle: 10});
            };
        </script>
    </head>
    <body>
        // YOUR SITE CODE HERE
        <div class="orbitvu-viewer" style="background: url(/static.orbitvu.co/001/CmWVYLSjZz3nh3FvjzepBQ/300/350/img01.png) no-repeat center center #ffffff; width: 350px; height: 300px">
            <script type="text/javascript" src="//orbitvu.co/001/CmWVYLSjZz3nh3FvjzepBQ/ov3601/2/script?newembed=yes&width=350&content2=yes&report_bugs=no&height=300&fromscript=1&viewer_api_init=my_api_callback_name">
                </script>
        </div>
    </body>

```

```

ov3601/2/script?newembed=yes
    &width=350&content2=yes&report_bugs=no&height=300&fromscript=1
    &external_access_init=my_api_callback_name"
    ></script>

</div>
// YOUR SITE CODE HERE
</body>
</html>

```

getScene

Returns information about currently displayed scene

```
viewer_api_obj.getScene();
```

Returns the dictionary object with the following information:

- **scale** - current scale
- **centerx** - current X coordinate of viewport centre (in pixels - relative to presentation's maximum resolution)
- **centery** - current Y coordinate of viewport centre (in pixels - relative to presentation's maximum resolution)
- **centerX** - same as centerx
- **centerY** - same as centery
- **hangle** - the number of currently shown horizontal frame
- **vangle** - the number of currently shown vertical frame

setScene

Main function used to manipulate presentation:

```
viewer_api_obj.setScene({<param_name>: <param_value>});
```

Parameters:

- **hangle**
Rotate to horizontal frame
Values: 0 - <number of frames>
Example: viewer_api_obj.setScene({hangle: 10});
- **vangle**
Set vertical angle for 3D presentations
- **hangleInc**
Change horizontal angle by specified number of frames, eg. 2 or -2

- **vangleInc**
Change vertical angle by specified number of lines, eg. 1 or -1
- **panX**
Move current view along horizontal axis, eg.: 100 (move 100px to the right), -10% (move 10% of the current view to the left), -30px (move 30px to the left).
- **panY**
Move current view along vertical axis, eg.: 100 (move 100px to the bottom), -10% (move 10% of the current view to the top), -30px (move 30px to the top).
- **mode**
Mode in which ORBITVU VIEWER operates
Values: rotate | move | onerotation | onerotationslow
Default: rotate
- **autorotate**
Start/stop autorotation
Values: yes | no
Default: no
- **autorotate_dir**
Autorotation direction
Values: left | right
- **zoomMul**
Zoom by specified factor
- **scaleUp**
Zoom in
- **scaleDown**
Zoom out

 You should not mix different parameters together in a single call

addCallback

Adds a callback to handle ORBITVU VIEWER events:

```
viewer_api_obj.addCallback('<callback_function_name>', '<event_name>');
```



You have to pass name of the globally accessible callback function, not the function object itself

Available events:

- **activity_start**

Available since 3.3.0

Called when any activity occurs, eg. when VIEWER is interacted with or when API calls cause some actions to happen

- **activity_end**

Available since 3.3.0

Called when no activity is happening in VIEWER, eg. when autorotation is stopped

- **autorotate_start**

- **autorotate_stop**

Called when autorotation is stopped, e.g. user clicks the presentation

- **center_change**

Called when view centre is changed

- **fullscreen_off**

- **fullscreen_on**

- **interaction_start**

Available since 2.3.27

Called on any user interaction with VIEWER, e.g. mouse click, button click, zoom in/out with mouse wheel

- **loading**

Available since 2.3.33

Called when ORBITVU VIEWER is loading frames. The callback is passed numeric value representing a percentage of already loaded frames.

When [partial_load](#) is set to yes then you might be also interested in **partial_loading** event defined below as in this case **loading** event will only apply to 4 first frames loaded.

- **mode_changed**

Available since 2.3.33

Called when VIEWER mode is changed eg. from rotate to drag

Values: rotate | drag | onerotation | onerotationslow

- **partial_loading**

Available since 3.0.26

Called when VIEWER is loading frames in the background (when [partial_load](#) is set to yes). The callback is passed numeric value representing a percentage of already loaded frames.

- **partially_initialized**
Called when [partial_load](#) is set to yes and when the first set of 4 frames has been loaded (VIEWER is able to interact with a user).
- **rotate**
Called while rotating
- **set_scene**
Triggered when setting the scene starts.
- **set_scene_complete**
Triggered when setting the scene is completed
- **viewer_initialized**
Available since 2.3.33
Called when all required frames are loaded and it is possible to manipulate/interact with VIEWER.
- **zoom_in**
- **zoom_out**

calculate_position

Available since 2.3.15

Converts x, y coordinates relative to presentation's **max_width** and **max_height** to current viewport coordinates (scaled).

For example, if presentation's **max_width/max_height** is 4422px x 3073px and it is displayed in a viewport of size 368px x 400px then coordinates of x: 1000 y: 500 will be converted to coordinates relative to viewport depending on current scale and center position:

scale: 0.175, centerX: 2623, centerY: 1930

converted:

x: -99, y: -50

This is useful if you want to display something in the specific position over the presentation.

Example usage:

```
viewer.calculate_position({x: 1000, y: 500})
```

where a **viewer** is an object returned by the [viewer_api_init](#) (or [external_access_init](#)) callback.

Parameters:

- **coords**
dictionary with x, y values, eg.: {x: 1000, y: 500}

get_presentation_info

Available since 2.3.15

Returns the dictionary containing the following information:

- `width` - current viewport width
- `height` - current viewport height
- `max_width` - max presentation width
- `max_height` - max presentation height
- `handles_no` - number of horizontal frames
- `vangles_no` - number of vertical frames
- `min_scale` - minimum zoom scale value
- `max_scale` - maximum zoom scale value

handle_resize

Manually inform ORBITVU VIEWER that its container size has changed:

```
viewer_api_obj.handle_resize();
```

This is called automatically on window resize but if you intend to e.g. interactively change dimensions of the VIEWER's container element, then you can call `handle_resize` to make sure ORBITVU VIEWER is shown correctly.

enable_mousewheel

Enable zoom on the mouse wheel. This method switches [mousewheel](#) parameter to 'yes':

```
viewer_api_obj.enable_mousewheel();
```

disable_mousewheel

Disable zoom on the mouse wheel. This method switches [mousewheel](#) parameter to 'no':

```
viewer_api_obj.disable_mousewheel();
```

set_forced_mode

Available since 2.3.33

Switch to drag or rotate mode of the ORBITVU VIEWER

Values: drag | rotate

ORBITTOUR API

ORBITTOUR embedded at web page exposes API object that provides some useful methods to customise ORBITTOUR behaviour.

To get access to ORBITTOUR API object you have to define global JavaScript function and pass its name to ORBITTOUR's embed script:

```
<script type="text/javascript">
  window['my_orbit_init'] = function(orbittour_api){
    // do something with API object
  }
</script>
<script type="text/javascript" src="//orbitvu.co/001/dSz54kMJ7rHKtC6E2YBHpf/2/orbittour/
468/script?external_init=my_orbit_init"></script>
```

The function will be called after ORBITTOUR is initialized.

Available API methods are:

handle_resize

Manually inform ORBITTOUR that it should recalculate its dimensions. Useful when you change the size of container element with some other script. Sample usage:

```
$('#orbit-container').css('width', 500);
orbittour_api.handle_resize();
```

 This method is called automatically when browser window is resized

enable_mousewheel

Proxy method to ORBITVU VIEWER API. This will enable VIEWER zoom on a mouse wheel. This method switches [mousewheel](#) parameter to 'yes':

```
orbittour_api.enable_mousewheel();
```

disable_mousewheel

Proxy method to ORBITVU VIEWER API. This will disable VIEWER zoom on a mouse wheel. This method switches [mousewheel](#) parameter to 'no':

```
orbittour_api.disable_mousewheel();
```

setScene

Proxy method to ORBITVU VIEWER API [setScene](#) method:

```
orbittour_api.setScene({scale: 0.5});
```

getScene

Proxy method to ORBITVU VIEWER API [getScene](#) method:

```
let scene = orbittour_api.getScene();
```