

WebViewer Export Configuration

The Onyx 3D WebViewer allows to create a web based representation of the dataset for viewing in web browsers, including on mobile devices. The WebViewer can be integrated into websites or web portals of service providers, but does not itself provide any functions that go beyond mere presentation. See also entry [Web-based communication](#).

Usage

Export a web based 3D dataset in the fullscreen view of the respective finding using the object list context menu entry "Web Export". It is also available through the context menu entry of the thumbnail in tab 3D. Enter a name under which the link is stored in Onyx and fill out the following dialog:

Entry	Description	Standard
Show Visible Objects Only	If checked, no invisible objects are exported to reduce the file size.	on
Animation (source)	Include the animation which can be generated when saving the finding from the modules V.T.O. 3D, Aligner 3D or Sim 3D	on
Comment	Display the comment of the finding as static text in the webviewer	off
Object List	Show an Onyx-like list for modifying objects	off
Tools	Show a toolbar for showing/hiding attachments, malocclusion, teeth colors and other details that have been exported with the finding. Especially on mobile devices this option should be preferred to the Object List	off
Colors	Use individualized colors in the webviewer	off
Password	Always choose a new password for data protection (or leave empty for random password)	
Embed Password into Link	on: The password is written as part of the link (lower security), off: The password must be communicated separately and must be entered on starting the viewer (higher security)	off
Patient Data	Select the data to display on the information popup	all off
Link	Result which can be given to the user or can be embedded into a patient-specific web page	
Password	Only shown when not included in the link	

Configuration and Individualization

The upload has to be configured in [System Options](#) on tab [Upload Data](#) > 3D-Web-Viewer before it becomes available as menu item.

For individualizing the webviewer with a custom logo, the site linked as "Viewer Homepage" in the settings must embed the webviewer as shown in the following example and must also contain the logo as a transparent image element.

[Example code for a page with individualized viewer](#)

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head> <title>My 3D Viewer</title> <meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />
```

```
<script>
  function addIFrame() { // Frame is embedded using this script function
    var viewerLink = '//onyx.orthorobot.com/webviewer/main.html'; // Change this
link to the Viewer subfolder that is uploaded next to your data. In the above
example, this would be '???..com/OnyxWebviewerUploads/Viewer/main.html'
    var viewerData = window.location.search; // If the data
is loaded into a lab portal, this link can also be created dynamically. In this
example, the parameters are taken from the URL.
    var iframe = document.createElement('iframe'); // Optionally
add the iframe as a child to any other element of your website. In this simple
example, it is appended to the end of the document.
    iframe.setAttribute('name', 'OnyxCephWebGL');
    iframe.setAttribute('style', 'position:fixed; top:0px; left:0px; bottom:0px;
right:0px; width:100%; height:100%; border:none; margin:0; padding:0;
overflow:hidden; z-index:1;');
    iframe.setAttribute('src', viewerLink+viewerData);
    document.body.appendChild(iframe);
  }
</script>
</head>
<body onload="addIFrame()">
  
  <!-- the iframe is inserted here -->
</body>
</html>
```

Troubleshooting

First steps to configure your own server and viewer website

1. Fill out the configuration as described above.
2. Upload a dataset and check that the link that is behind the `mLink=` parameter can also be downloaded in the browser separately.
3. Copy the example code from above to a html page on your own server and change the variable `viewerLink` to the viewer on your own server as described in the comment.
4. Add a logo which is ideally transparent and scaling well with the site also for mobile devices.
5. Link your own viewer site in the OnyxCeph configuration and test the upload.

The upload in OnyxCeph does not work

- Check the upload-settings and verify that you can connect with your settings using an auxiliary tool like TotalCommander or FileZilla
- If it is working with the same account settings on such other programs, test the other protocols (FTP, FTPES, FTPS, SFTS).

The viewer page reports "File not found - Please check link"

- Check the data address in the configuration: can you access the `iiwgl` file linked in the URL after the `mLink=` parameter? If not use an FTP tool to see where your data is uploaded to and how you can access those files over http or https. Possibly, the file type must be declared explicitly as "allowed to download" in the server configuration.

- Notice that the host directory is relative on the server and will be different from the data address which is needed for HTTP(S) access
- Open the development tools in your browser using F12 and look for any red error message in the console:
 - If the error contains something about "CORS", it is not loading because of a security mechanism which must be configured correctly at the server.
 - If the error complains about http and https: verify your configuration: a viewer at https cannot access data using http

The viewer page reports "Wrong password or corrupt file"

- The file returned by the link in the `mLink=` parameter is not the correct data, but in such cases often an error page from the server like "404-File not found".
- Check again by entering this link alone that the correct file is delivered by the server.

Extended Interface

The webviewer supports a wide range of additional options for individualizing appearance and functionality. For instance, the icons can be exchanged and the font size can be controlled for "responsive design", i.e. when the viewer is embedded in a more complex page. These options are accessible with the extended interface which is building on the simple example above.

Example code for the extended interface

```
function addIFrame() {

    // add event handler to catch the loading finished event from the viewer:
    window.addEventListener("message", eventFromWebviewer);

    var viewerLink = '//onyx.orthorobot.com/webviewer/main.html'; // Change this
    link to the Viewer subfolder that is uploaded next to your data. In the above
    example, this would be '???/OnyxWebviewerUploads/Viewer/main.html'
    var viewerData = window.location.search; // If the data is
    loaded into a lab portal, this link can also be created dynamically. In this
    example, the parameters are taken from the URL.
    var iframe = document.createElement('iframe'); // Optionally add
    the iframe as a child to any other element of your website. In this simple example,
    it is appended to the end of the document.

    iframe.setAttribute('name', 'OnyxCephWebGL');
    iframe.setAttribute('style', 'position:fixed; top:50px; left:0px; bottom:0px;
    right:0px; width:100%; height:80%; border:none; margin:0; padding:0;
    overflow:hidden; z-index:1;');
    iframe.setAttribute('src', viewerLink+viewerData);
    document.body.appendChild(iframe);
}

// handle events from the viewer
function eventFromWebviewer(event) {

    var eventObj = JSON.parse(event.data);
    if(eventObj !== null && eventObj.type === "LOADING" && eventObj.value ===
    "finished") {
```




```

    // Do any initial setup here if required, like:
    /* functions for responsive design scaling and replacing icons
    window.frames.OnyxCephWebGL.postMessage({'type':"SCALE_UI","percent":"50%"},"*'); // scale down the ui for better reactive sizes when the viewer is not
fullscreen
window.frames.OnyxCephWebGL.postMessage({'type':"REPLACE_ICON","id":"viewUpperBtn"
,"part":"path","value":"M ... Z"},"*"); // see below
    // id: possible values are:
    //     viewUpperBtn, viewFrontBtn, viewLowerBtn, viewRotateBtn, infoBtn,
    //     aniPlayBtn, aniPauseBtn, aniStepBtn, aniRepeatBtn, aniNoRepeatBtn,
    //     aniSpeed0Btn, aniSpeed1Btn, aniSpeed2Btn,
    //     aniMarkerBtn0, aniMarkerBtn1, ... // replace all with a loop up to
aniMarkerBtn300 (current maximum)
    //     toggleClippingBtn, toggleRulerBtn, toggleCommentBtn,
toggleMalocclusionBtn, toggleAttachmentsBtn
    // part: "path" or "viewBox", see below. Note that it is not possible to
change the style or add other svg elements.
    // value: d attribute of the svg path element for "part":"path" or viewBox
attribute of the svg element for "part":"viewBox"
    // Example for changing all the step icons to smaller icons (i.e. smaller
numbers)
    for(var i=0; i<300; i++) // Tiny numbers (bottom aligned)
window.frames.OnyxCephWebGL.postMessage({'type':"REPLACE_ICON","id":"aniMarkerBtn"
+ i + "", "part":"viewBox", "value":"-20 -30 64 64"},"*");
    for(var i=0; i<300; i+=5) // Scale medium for 0,5,10,...
window.frames.OnyxCephWebGL.postMessage({'type':"REPLACE_ICON","id":"aniMarkerBtn"
+ i + "", "part":"viewBox", "value":"-5 -5 34 34"},"*");
    // Example for changing all the icons back to the pin-icons:
    for(var i=0; i<300; i++) {
window.frames.OnyxCephWebGL.postMessage({'type':"REPLACE_ICON","id":"aniMarkerBtn"
+ i + "", "part":"viewBox", "value":"0 0 24 24"},"*");
window.frames.OnyxCephWebGL.postMessage({'type':"REPLACE_ICON","id":"aniMarkerBtn"
+ i + "", "part":"path", "value":"M12,11.5A2.5,2.5 0 0,1 9.5,9A2.5,2.5 0 0,1
12,6.5A2.5,2.5 0 0,1 14.5,9A2.5,2.5 0 0,1 12,11.5M12,2A7,7 0 0,0 5,9C5,14.25 12,22
12,22C12,22 19,14.25 19,9A7,7 0 0,0 12,2Z"},"*");
    }
    */
    /* functions for controlling the timeline
    window.frames.OnyxCephWebGL.postMessage({'type':"PLAY_T0","key":"0"},"*");
// Plan to specific key "0", "1", ... or "first" or "last"
    */
    /* possible functions for automatic control of the "tools"
    window.frames.OnyxCephWebGL.postMessage({'type':"ATTACHMENT_STYLE",
"value":"ghost"},"*"); // "normal","hidden","ghost" (show Attm. as in
animation/not at all/as ghosts only)
window.frames.OnyxCephWebGL.postMessage({'type':"MALOCCLUSION_STYLE","value":"ghos
t"},"*"); // "normal","hidden","ghost","full" (overlay M0 as in animation/not/as
ghost/very strong)
    window.frames.OnyxCephWebGL.postMessage({'type':"TEETHCOLOR_STYLE",
"value":"hidden"},"*"); // "normal","hidden" (show/hide colors exported with the
aligner animation like occlusal contacts)
    window.frames.OnyxCephWebGL.postMessage({'type':"COMMENT_STYLE",
"value":"short"},"*"); // "none","short","full" (comment off/step comment
only/step + finding comment)
    window.frames.OnyxCephWebGL.postMessage({'type':"RULER_STYLE",
"value":"none"},"*"); // "none","ruler" (ruler on/off)

```

```
    */
    /* functions for control of the "navigation panel" on the left
    window.frames.OnyxCephWebGL.postMessage({'type':"CAMERA",      "from"
:"left"}', '*'); // "left", "right", "top", "bottom", "front", "back"
    window.frames.OnyxCephWebGL.postMessage({'type':"CAMERA",      "zoom"
:"reset"}', '*'); // reset zoom
    window.frames.OnyxCephWebGL.postMessage({'type':"CAMERA",
"transl":"reset"}', '*'); // reset translation
    window.frames.OnyxCephWebGL.postMessage({'type':"HIDE_JAW",    "jaw"
:"upper"}', '*'); // "upper", "lower", "none"
    window.frames.OnyxCephWebGL.postMessage({'type':"CLIPPING",    "side"
:"left"}', '*'); // "left", "right", "none", "switch" (switch toggles through the
modes)
    */
  }
}
```

Links

	Web-based communication
	WebView Upload Settings
	WebViewer Controls

From:

<http://onyxwiki.net/> - [OnyxCeph³™ Wiki]

Permanent link:

http://onyxwiki.net/doku.php?id=en:function_webexport&rev=1697458936

Last update:

2023/10/16 14:22

