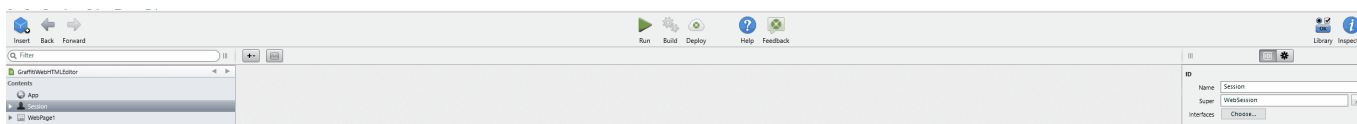


Session Attributes

Do not use the demo values of `cdn.graffitisuite.com` in your projects. This will break eventually.

GraffitiSuite allows you to offload the associated JavaScript, CSS, and Font files required for the suite to function properly. This is accomplished by first uploading the scripts directory to the server of your choice, then creating and uploading the HTACCESS file (see “HTACCESS for File Hosting” note, Apache HTACCESS now included in scripts directories by default), and finally adding the attributes to the Session object as follows:

| AttributeName | AttributeValue |
|---------------|---|
| CDNSecure | https://yourhosting.com/graffitisuite/scripts |
| CDNUnsecure | http://yourhosting.com/graffitisuite/scripts |
| DebugRemote | Boolean |
| DeployRemote | Boolean |



Depending upon protocol for your app (HTTP/HTTPS), GraffitiSuite will automatically choose the appropriate attribute value. If you do not provide these attributes in the Session object, GraffitiSuite will continue to serve its local version of the files with protocol selection handled automatically by Xojo based on the connection type.

If DebugRemote is set to False and running your project in debug mode, the resources from the scripts folder will be served by your application. When set to True and running your application in Debug, resources will be loaded from the URLs supplied by your CDNSecure and CDNUnsecure attributes.

If DeployRemote is set to False and your project is compiled, the resources from the scripts folder will be served by your application. When set to True and your project is compiled, resources will be loaded from the URLs supplied by your CDNSecure and CDNUnsecure attributes.

File Hosting Setup

In order to serve the JavaScript, CSS, and Font files from an alternate location, you must add an HTACCESS file (“`.htaccess`”) to the web directory where you will be storing the scripts. This file is now included by default in the Scripts directory.

The contents below ensure that your files will not be blocked by Cross-Origin policies.

Apache

HTACCESS files for Apache are now included by default. If you receive 500 Internal Server Errors, see below or [Common Hosting Errors](#) for more information.

```
Options All -Indexes

<IfModule mod_headers.c>
  <FilesMatch "\.(ttf|ttc|otf|eot|woff|woff2|font.css|css|js)$">
    Header set Access-Control-Allow-Origin "*"
    Header set Access-Control-Allow-Headers "origin"
    Header set Access-Control-Allow-Methods "GET"
  </FilesMatch>
</IfModule>

<IfModule mod_expires.c>
  # Fonts
  AddType application/vnd.ms-fontobject .eot
  AddType application/x-font-ttf .ttf
  AddType application/x-font-opentype .otf
  AddType application/x-font-woff .woff
  AddType image/svg+xml .svg
  # Turn it on
  ExpiresActive On

  # Images
  ExpiresByType image/jpeg "access plus 1 year"
  ExpiresByType image/gif "access plus 1 year"
  ExpiresByType image/png "access plus 1 year"
  ExpiresByType image/webp "access plus 1 year"
  ExpiresByType image/svg+xml "access plus 1 year"
  ExpiresByType image/x-icon "access plus 1 year"

  # Video
  ExpiresByType video/mp4 "access plus 1 year"
  ExpiresByType video/mpeg "access plus 1 year"

  # CSS, JavaScript
  ExpiresByType text/css "access plus 1 month"
  ExpiresByType text/javascript "access plus 1 month"
  ExpiresByType application/javascript "access plus 1 month"

  # Fonts
  ExpiresByType application/vnd.ms-fontobject "access plus 1 year"
  ExpiresByType application/x-font-ttf "access plus 1 year"
  ExpiresByType application/x-font-opentype "access plus 1 year"
  ExpiresByType application/x-font-woff "access plus 1 year"
  ExpiresByType image/svg+xml "access plus 1 year"

  # Others
```

```
ExpiresByType application/pdf "access plus 1 month"  
ExpiresByType application/x-shockwave-flash "access plus 1 month"  
</IfModule>
```

Some users have reported needing the following section in addition to the code above in your .htaccess file:

```
<Directory /path/to/scripts>  
    AllowOverride FileInfo Options Indexes  
</Directory>
```

As of R28

The .htaccess for the scripts directory will also include file caching directives to reduce the load on your server and load your web application more quickly. Once you have updated your scripts folder, users may need to clear their cache either browser-wide or site-wide (using key combinations such as CTRL+SHIFT+R in Chrome for Windows) before accessing your application. This functionality requires that your server have mod_expires installed and activated.

NGINX

By default nginx has no default mime types for fonts, and an incorrect mime type for .eot files. Go to folder with configs and find where mime types are defined. Usually, that's in the mimes.types file.

Search for .eot and add the strings below:

```
application/x-font-ttf          ttc ttf;  
application/x-font-otf         otf;  
application/font-woff          woff;  
application/font-woff2         woff2;  
application/vnd.ms-fontobject eot;
```

For CORS headers, add something like this to your vhost config:

```
location ~* \.(eot|otf|ttf|woff|woff2)$ {  
    add_header Access-Control-Allow-Origin *;  
    add_header Access-Control-Allow-Header "origin";  
    add_header Access-Control-Allow-Methods "GET";  
}
```

IIS

Add the following MIME type declarations via IIS Manager (HTTP Headers tab of website properties):

```
.eot application/vnd.ms-fontobject  
.ttf application/octet-stream  
.svg image/svg+xml
```

.woff application/x-woff

If you do not have access to the IIS Manager, you can add these declarations in your Web.config file, in the <system.webServer> section.

```
<system.webServer>
  <staticContent>
    <mimeMap fileExtension=".eot" mimeType="application/vnd.ms-fontobject" />
    <mimeMap fileExtension=".ttf" mimeType="application/octet-stream" />
    <mimeMap fileExtension=".svg" mimeType="image/svg+xml" />
    <mimeMap fileExtension=".woff" mimeType="application/x-woff" />
  </staticContent>
</system.webServer>
```

For CORS, first, configure the OPTIONSVerbHandler to execute before .Net handlers.

In IIS console, select "Handler Mappings" (either on server level or site level; beware that on site level it will redefine all the handlers for your site and ignore any change done on server level after that; and of course on server level, this could break other sites if they need their own handling of options verb).

In Action pane, select "View ordered list..." Seek OPTIONSVerbHandler, and move it up (lots of clicks...). You can also do this in web.config by redefining all handlers under <system.webServer><handlers> (<clear> then <add ...> them back, this is what does the IIS console for you) (By the way, there is no need to ask for "read" permission on this handler.)

Second, configure custom http headers for your cors needs, such as:

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name="Access-Control-Allow-Origin" value="*" />
      <add name="Access-Control-Allow-Methods"
value="GET,POST,OPTIONS" />
      <add name="Access-Control-Allow-Headers" value="Content-Type,
soapaction" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

From:
<https://graffitisuite.com/wiki/> - **GraffitiSuite Documentation**

Permanent link:
<https://graffitisuite.com/wiki/doku.php?id=web:cdnhosting>

Last update: **2020/09/17 00:22**

