DNN LC File Debugger

# C:\Users\ricar\AppData\Local\Microsoft\Windows\INetCache\Content.Word\lcfileslogview.png

# Introduction

The purpose of "DNN LC File Debug" is to display the files associated with a visualizer from a browser. When we define Script and Style in a visualizer, in the rendering of the page two files are downloaded: .js and .css. These files are downloaded from an unfriendly path that can generate confusion when debugging problems on the client side.

A Chrome extension has been created that displays a live log of the .js and .css files of the visualizer that loads the page.

# The Background

Working with Liquid Content and especially with the visualizer I have seen that it is difficult to perform a debugging on the client side. The visualizer downloads the code of scripts and styles in file that in the browser with an unfriendly url.

I've worked with Chrome extensions that make easy debugging on client site for example working on SAML or Knockout. Now the question is: Can the visualizers used in my page give me the information of these files?

# What do we want to achieve?

We want to develop a Chrome extension that displays a live log files associated with a visualizer. This extension will show the .js and .css files that are defined in the visualizer adding code in the script and styles tabs. It will also provide a quick access to the source code for the scripts and styles of a visualizer in the browser to debug them. And finally shows extra information for the downloads of the files associated with the visualizer.

Liquid Content API



Web Browser

Visualizers in page:

* Javascripts
* Css

Chrome extensions

# How to build it

To see how the application has been developed and all the information related to the development access here: [GitHub repository](https://github.com/dnnsoftware/Dnn.Evoq.LiquidContent.Samples.Public/tree/master/Integration.ChromeExtension).

# What would be the next steps?

In further developments, not only to show the source code of the files, but to be able to debug the javascript files in the browser devtools window. And finally, be able to display more data from the viewers on client side without using their related files, because they can exist visualizer without script or styles that we would like that they were shown in the log