
Load multiple JavaScript files

Last updated: Mon, 06 Dec 2021 16:05:22 GMT

Link: [See a live example.](#)

View: [View source code in CONTENTdm Cookbook Demo Site.](#)

Allow your CONTENTdm website to use multiple Javascript files.

CONTENTdm's Website Configuration Tool has a single upload dialog for uploading JavaScript (**Global settings** > **Custom** > **Custom Scripts**). This is convenient if you have only a single custom function that you want to apply to your site, but it can become difficult to maintain if a monolithic JavaScript file needs to do several different things on different pages and in different contexts. The solution is to create a loader script that can link in separate, self-contained JavaScript files.

The idea here is a single JavaScript file that does nothing more than retrieves and loads additional JavaScript files. Those separate JavaScript files will still need to adhere to the [CONTENTdm event lifecycle](#) and execute at the appropriate time. The advantage of this approach is that each of these separate JavaScript files is self-contained and portable. There are no interdependencies among the script files that are being loaded.

Because the CONTENTdm Website Configuration Tool supports uploading only a single JavaScript file on the **Custom Scripts** dialog, the additional JavaScript files can be stored in another location using the file manager under the **Custom Pages** dialog. For more information about the file manager see [Working with the file manager](#).

This example loader script uses JavaScript, promises to read the external JavaScript files and insert them into the DOM. This loader script is designed to be self-executing and is not triggered by an event in the page lifecycle. The external scripts will need to adhere to the lifecycle to make sure they execute at the correct time and on the correct pages.

The bottom section of the script is where you configure the file path and names of the external JavaScript files being loaded. In this example, the **Custom Pages** file manager was used to create a folder called "js" and the additional JavaScript files were uploaded to that folder. In the bottom section:

- Replace `/customization/global/pages/js/` with your JavaScript folder in
`// path corresponding to directory where JS scrips are uploaded`
`const fileDirectory = '/customization/global/pages/js/';`
- Update `'example1.js'`, `'example2.js'`, and `'example3.js'` with your script files in
`// array containing file names of each JS file`
`const scriptFilesToLoad = [
 'example1.js',
 'example2.js',
 'example3.js'
];`

