Global vs collection-level context

Last updated: Wed, 27 Jan 2021 19:05:43 GMT

Find information about the difference between global and collection-level context within the CONTENTdm website. There are two contexts defined for all pages within the CONTENTdm website, global and collection. The global context applies to the home page, the about page, and the search page when more than one collection is being browsed or searched. The collection landing pages, item pages, and a search page for a single collection have a collection-level context. This context allows you to define customizations that apply only to specific collections. The most common application of this context is defining a custom page header for a collection, but this context can be utilized by any UI customization.

This global vs. collection-level context is sometimes explicit and sometimes implicit. For example, in the Website Configuration Tool there is a separate tab for collections where you can define the header and other settings for each collection. You can also specify whether a specific collection should have a unique identity or setting, or whether it should inherit whatever settings have been specified at the global level.

Most advanced customization options are available at both the global and collection levels in the Website Configuration Tool, with one notable exception (JavaScript). For example, if you want to use Custom CSS files on your site, you can upload a .css file at the global level in the Website Configuration Tool (Global Settings > Custom > Custom CSS) and/or at the collection level (Collection-level Settings > Custom > Custom CSS). These CSS files will automatically be loaded by the CONTENTdm website as the page context changes.

The reason that the custom JavaScript option does not follow this paradigm is that loading separate JavaScript code after the page context has been determined is slow and would lead to a noticeable delay for end users. The React routing system will instead tell your custom JavaScript code when to fire based on which events your custom code has subscribed to. See the JavaScript Customizations section for more information on how this works.