
USING THE JS LINK UPDATE UTILITY

OVERVIEW

UpdateJsLinks is a utility that allows you to update, in bulk, the JSLink setting of libraries that share the same name across multiple SharePoint sites. This is useful when you want to use the same FormsViewer form in several sites, for example to split your form data between different clients or departments.

This document details the steps for adding the UpdateJsLinks page to your SharePoint environment (which you generally only need to do once), and the steps for using the utility.

PREREQUISITES

Before using the UpdateJsLinks utility, you should have installed FormsViewer to an ancestor (parent, etc.) site of the sites that you plan to configure and prepared an FVLinks file for your form. You can download the base FVLinks file and instructions on setting it up from the following location:

http://www.infopathdev.com/files/folders/formsviewer_v11/entry107623.aspx

You can make sure that your FVLinks file is configured correctly by following the steps to add it to one of the target libraries and then using the UpdateJsLinks utility to propagate the setting across all of the sites.

Since multiple sites will be sharing the same FVLinks file, you should add it to an ancestor site of the target sites and use a site collection-relative JSLink setting, e.g. ~sitecollection/SiteAssets/FVLinks-Expense.js



SETTING UP THE UPDATEJSLINKS UTILITY PAGE

1. Navigate to a SharePoint site that is an ancestor (parent, etc.) of the sites you want to update.
2. Upload the UpdateJsLinks.html file to any library in the site from step 1. Typically a good place for it is a Site Assets library.
3. Create a page in the site from step 1 where you will place the UpdateJsLinks.html utility. This can be essentially any type of page – an empty list, a library, or a page within a SitePages library.
4. Navigate to the page from step 3.
5. Click the gear icon in the top right, and click Edit Page.
6. Click the Insert tab on the ribbon and click the Web Part icon.
7. From the Media and Content category, select Content Editor, and click Add.
8. Hover your mouse over the top right corner of the Content Editor web part, click the small downward triangle, and click Edit Web Part.
9. In the Content Link Field, enter the full URL of the UpdateJsLinks.html file you uploaded in step 2. This would typically be the full URL of the library where you uploaded it (not including /Forms/ or anything after that), plus UpdateJsLinks.html, for example:
<https://mycompany.sharepoint.com/sites/Customers/SiteAssets/UpdateJsLinks.html>
10. Click OK in the web part settings pane. Verify that several fields and labels are added to the page. If not, try adjusting the URL in step 9.
11. Click Save or Stop Editing in the ribbon to save the page.



USING THE UPDATEJSLINKS UTILITY PAGE

The UpdateJsLinks utility is currently designed to crawl all descendant sites of a particular site and modify the JSLink setting in a particular view of libraries with a particular name. For example, it can crawl subsites looking for libraries with the name **Expense Reports** and set the JSLink setting on views with the name **PendingItems.aspx**. To update several different views, you can run the utility more than once.

1. Navigate to the page where you added the UpdateJsLinks utility.
2. Provide values for the four fields at the top of the utility (you do not need to modify any that are pre-populated with the desired value):
 - a. Base Site Url – URL of the site where the utility should start its crawl to look for sub-sites. This defaults to the URL of the current site.
 - b. Library Name – The name of the libraries that you want to update.
 - c. View Name – Name of the view that you want to update in each library. This defaults to AllItems.aspx.
 - d. JSLink URL – The new value you want to use for the JSLink URL in each library. This will typically be something like *~sitecollection/SiteAssets/FVLinks-Expenses.js*
3. Click the Run button. The operation should begin and start outputting a log to the large box under the Run button. There are likely to be some errors when the utility tries to crawl the FormsViewer app and possibly other apps (apps show up as sites when the utility queries sites), but as long as it reports updating the libraries and views it is looking for, this generally means it was successful.

