INFORMATION ARCHITECTURE USER GUIDE

Information Architecture User Guide

Introduction

This guide is for the Technical Writing team responsible for maintaining the JustFood help.

The JustFood help is integrated with Microsoft Business Central help, and multiple versions of the help are maintained. Because of this setup, specific steps must be followed to ensure updates are applied correctly across versions.

JustFood source files are stored in DevOps, and Microsoft source files are stored in GitHub. Source files are edited in Visual Studio Code, and DocFx is used to build the help output.

Set up DocFx

DocFx is used to generate the online help files.

Important: If you decide to upgrade your DocFx version, create a backup of your existing version in case the new version doesn't work.

Install DocFx

1. Download DocFx from https://github.com/dotnet/docfx/releases.
2. Unzip the files to *c:\docfx*.

Add DocFx to Path

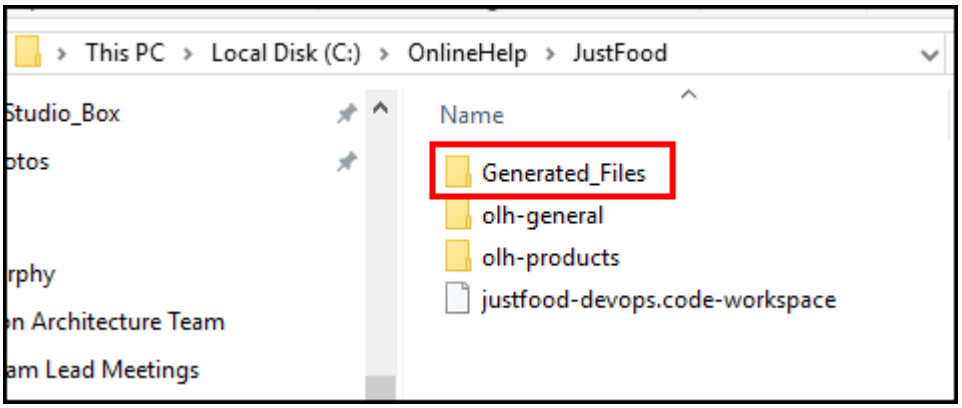
These steps allow you to run the *docfx.exe* from anywhere.

1. Click the Windows icon.
2. Enter *Advanced System Settings*.
3. Select *View Advanced System Settings*.
4. Click **Environment Variables**.
5. Under System Variables, select *Path* (if Path doesn't exist, click New).
6. Click **Edit**.
7. Click New.
8. Add *C:\docfx* (or whatever folder the docfx.exe file is located).
9. Click **OK**.
10. Click **OK**.

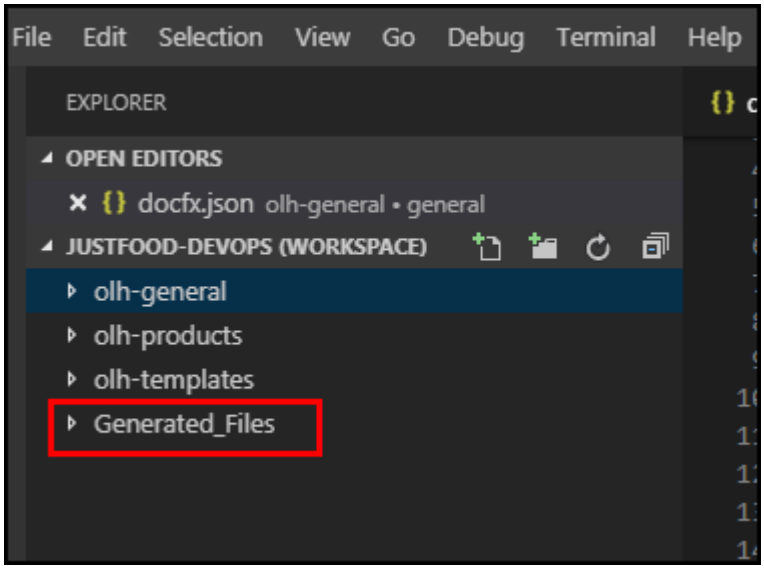
Create a Folder for the Generated Files

When the *docfx.json* file is run for a help system, DocFx creates a folder that contains the generated files. This step creates the folder that will hold the generated files.

1. Create the *Generated_Files* folder in *C:\OnlineHelp\JustFood*.



2. Add the *Generated_Files* folder to the workspace that contains the cloned repositories.
   a. Open your repository workspace in Visual Studio Code.
   b. Choose **File>Add Folder to Workspace**.
   c. Navigate to C:\OnlineHelp\JustFood.
   d. Select *Generated_Files*.
   e. Click **Add**.



Troubleshoot DocFx

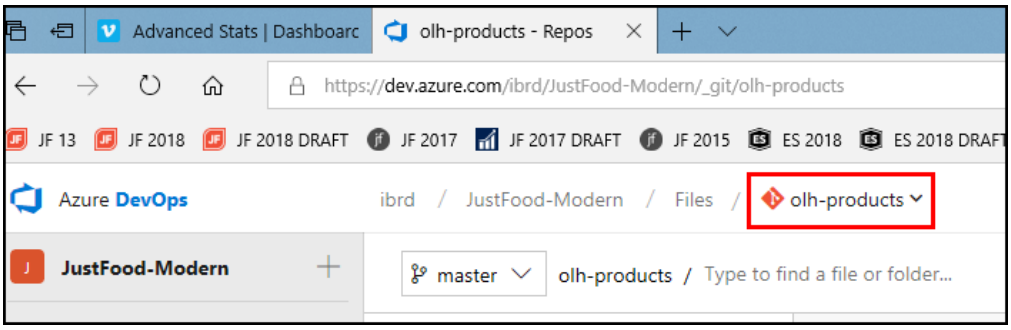
If you have DocFx questions, try the following resources:

- Search in GitHub. https://github.com/dotnet/docfx/issues
- Ask a question on the DocFx Getting Started page.

Clone a JustFood Repository

This step creates a folder on your machine, to which you can add JustFood folders and files. Those folders and files can then be uploaded to DevOps.

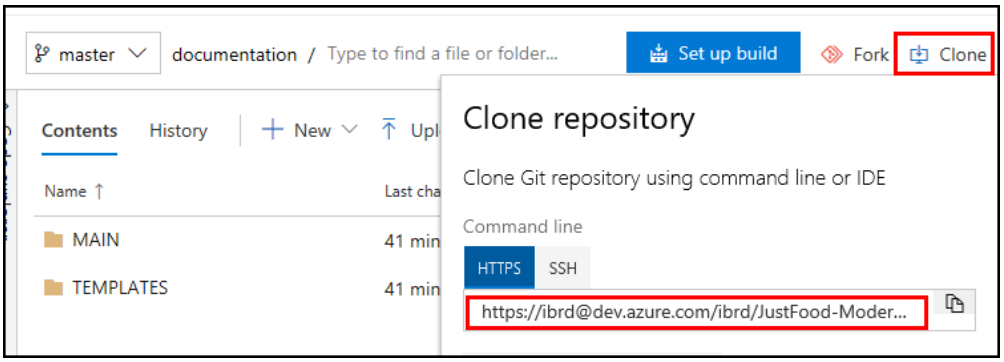
1. Open DevOps (https://dev.azure.com/ibrd).
2. Select the appropriate project.
3. Select **Repos**.
4. Select the repo that you want to clone.



For JustFood, you need to clone the following repositories:

- *olh-general*: Contains files that are not version dependent such as the video files
- *olh-products*: Contains all of the files and templates used to generate the different product helps.
- *olh-templates*: Contains the default templates used to generate the various helps.

5. Click **Clone** and copy the URL.



6. Open Visual Studio Code.
7. Go to **View>Command Palette** (**Ctrl+Shift+P**).
8. Select **Git:Clone**.
9. Paste the URL that you copied, and press **<Enter>**.
10. Select the folder where you want the clone to be created.

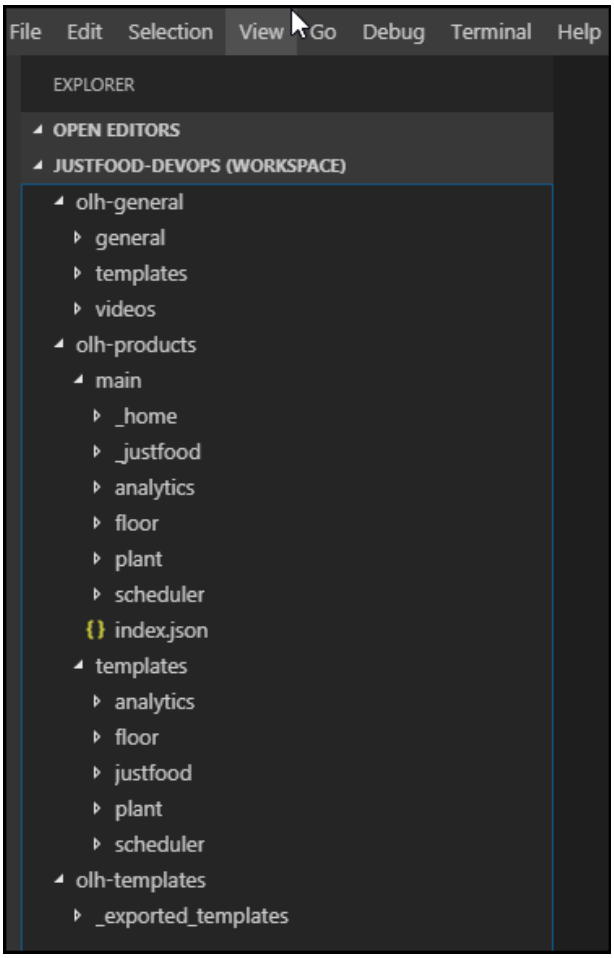
    For JustFood, place the clones in the following folders:

    - *olh-general* to *C:\OnlineHelp\JustFood*
    - *olh-products* to *C:\OnlineHelp\JustFood*
    - *olh-templates* to *C:\OnlineHelp\Templates*

11. Click **Select Repository Location**.
12. Enter your credentials if prompted.
13. Click **Open Repository**.

The repository folder was added to the selected location on your local machine.

14. Repeat steps 4-13 for the other repositories, and then choose **Add to Workspace**.



15. Save the workspace to *C:\OnlineHelp\JustFood*.

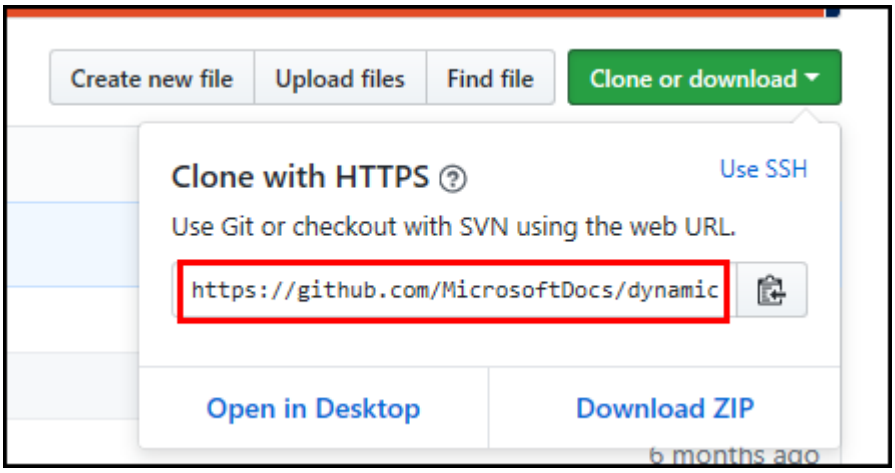
    You can now make changes to the help files, and generate the help as necessary. When you make changes to the files, or add files, you need to then commit your changes and synchronize with DevOps.

Clone Microsoft Source Files

You need to create a clone of the Microsoft help files to add the most up-to-date files to the JustFood help system.

1. Open the MS source files on GitHub. https://github.com/MicrosoftDocs/dynamics365smb-docs/tree/FromPrivateLive-sync-working

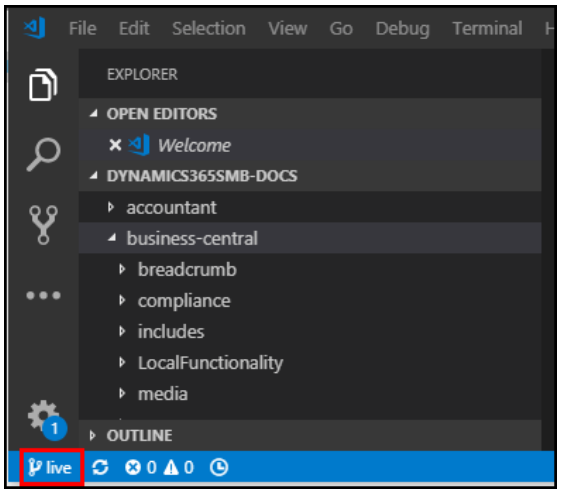
2. Click **Clone or download**, and copy the URL.



3. Open Visual Studio Code.
4. Go to **View>Command Palette** (**Ctrl+Shift+P**).
5. Select **Git:Clone**.
6. Paste the URL that you copied, and press **<Enter>**.
7. Select the following folder for the location of the clone: *C:\OnlineHelp\Microsoft_Source_Files*.
8. Click **Select Repository Location**.
9. Click **Open Repository**.

   The repository folder was added to the selected location on your local machine.

   Note: The Microsoft clone contains all of the Dynamics 365 files but you only need the business-central folder.
10. Switch to the *FromPrivateLive-sync-working* branch.
    a. Click *live* in the bottom-left corner.



    b. Select origin/FromPrivateLive-sync.

Create the Help

Create a Version Folder in the Generated_Files Folder

This step adds a version folder to the *Generated_Files* folder so that you can find the files for a certain version.

1. In Explorer, navigate to *C:\OnlineHelp\JustFood\Generated_Files*.
2. In the *Generated_Files* folder, create a folder that represents the product's version. For example, *Spring_2019*.

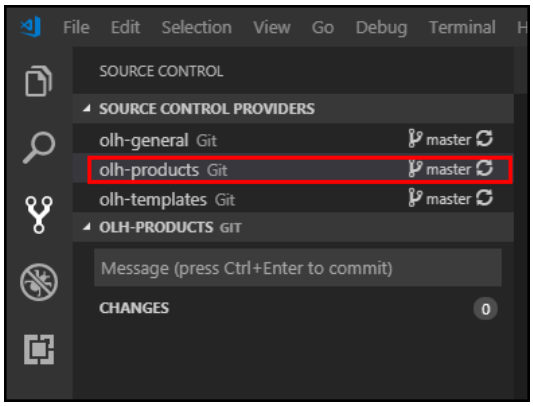
Select the Master Repo

This step is to verify that the *master* repo is selected when you're doing your work. The master repo always contains the newest version of the help.

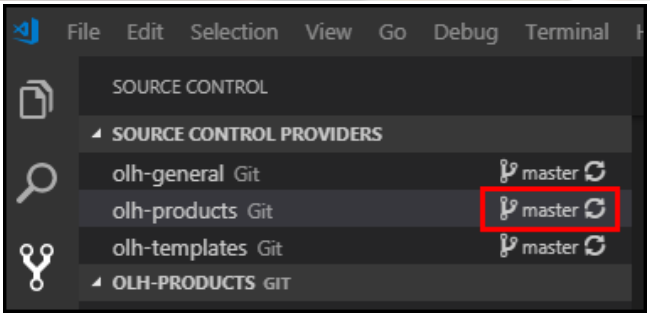
Important: It's important that you pay attention to which branch you are in when making changes, and when publishing your changes. If you make changes in the master, but do NOT publish the changes to DevOps, and then switch to another branch, those changes will then apply to the other branch.

1. In Visual Studio Code, click the Source Control icon.

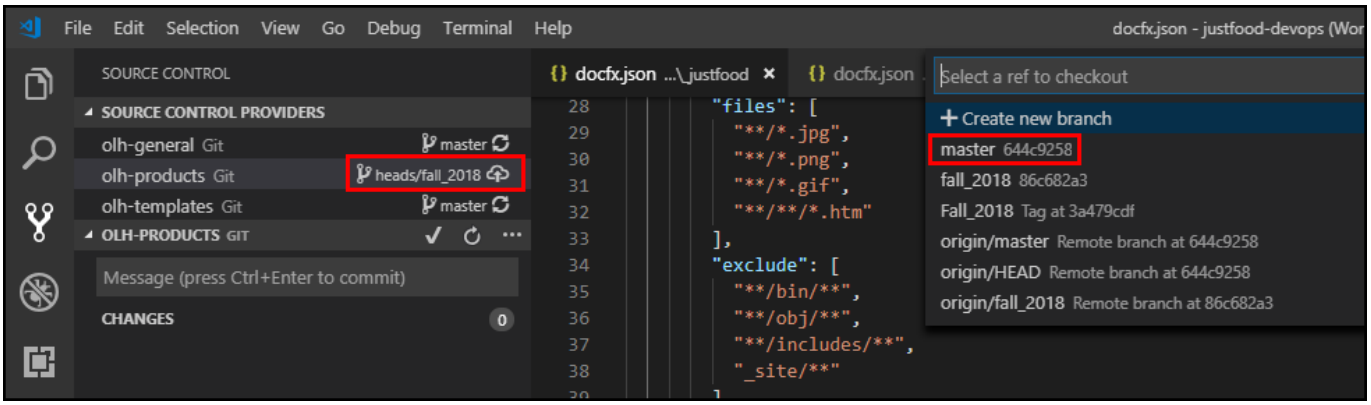
2. Under Source Control Providers, select *olh-products*.



3. If *master* is already displayed, then you do not have to do anything.



4. If *master* isn't displayed, click the *<version>* branch, and select *master*.



Update the Help Files

Update the docfx.json File

This step is to update the folder where the generated files are stored.

1. Open the workspace in Visual Studio Code.
2. Navigate to *olh-products/main/<product>/docfx.json*.

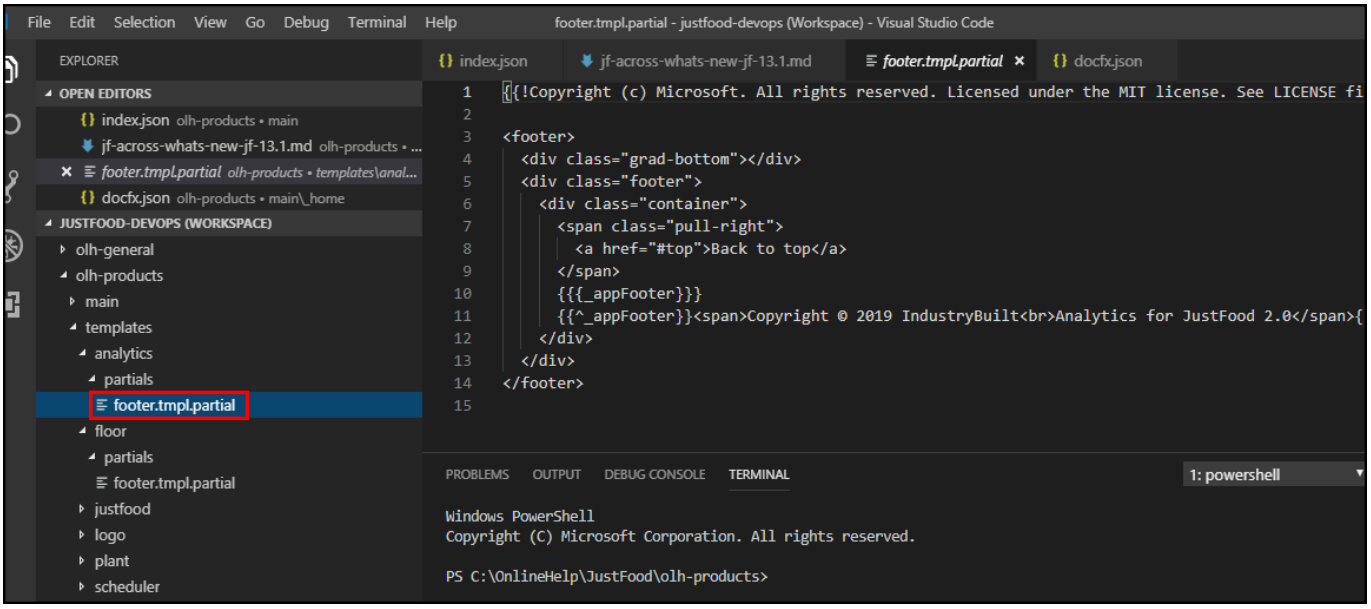
3. Change the version to the new version.
   This must be the same folder as defined in Create a Version Folder in the Generated_Files Folder.
4. Repeat for the remaining products:
   - _home
   - _justfood
   - analytics

- floor
- plant
- scheduler

Update the Footer Information

These steps update the product version numbers, and copyright year if required.

1. Open the workspace in Visual Studio Code.
2. Navigate to *olh-products/templates/<product>/partials/footer.tmpl.partial*.

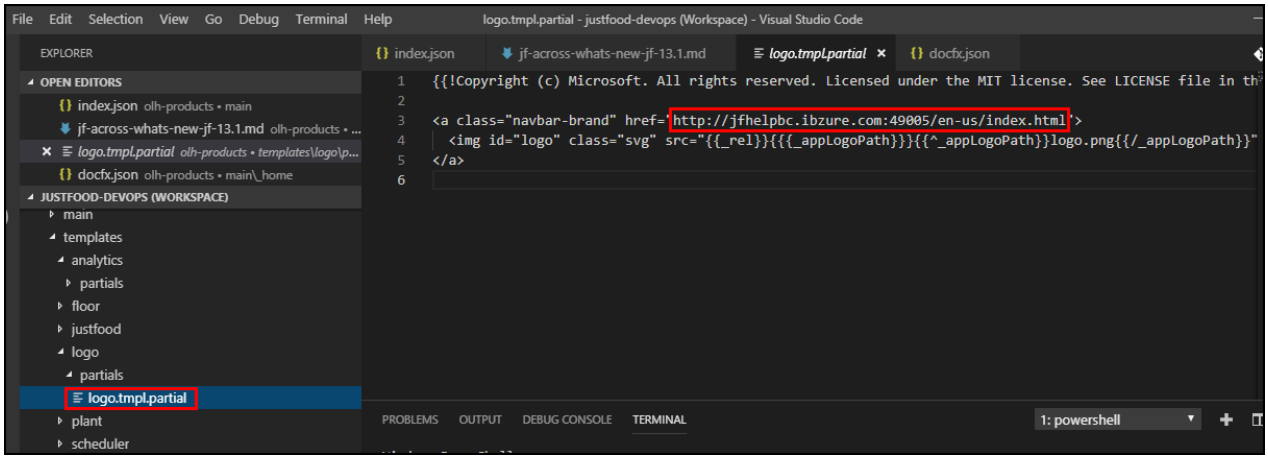


3. Change the product version number.
4. Change the copyright year if required.
5. Repeat for the remaining products.

Update the Logo Link

This updates the link for the logo in the header.

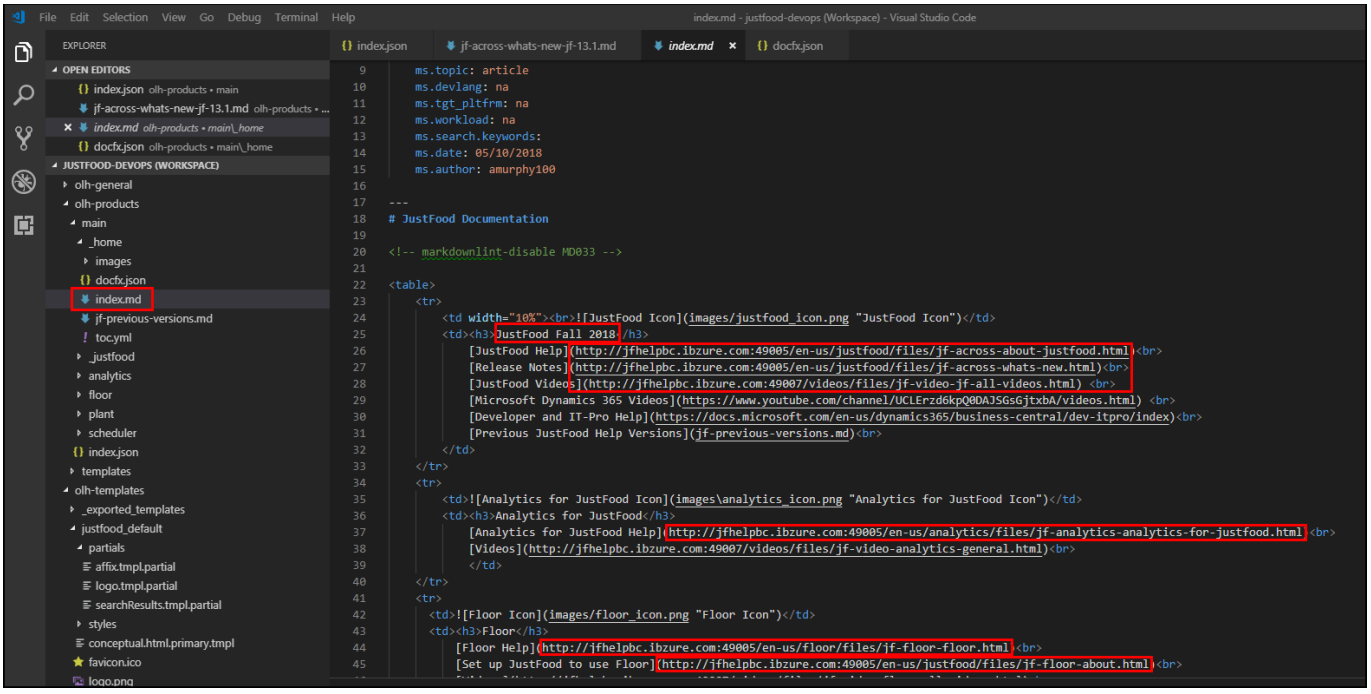
1. Update the logo file that is used by the various product helps.
   a. Navigate to *olh-products/templates/logo/partials/logo.tmpl.partial*.
   b. Update the URL to point to the new URL for the main Help page. For example, http://jfhelpbc.ibzure.com:49007/spring_2019/en-us/index.html.



Update the Main Home Pages

These steps update the information on the main Home page.

1. Navigate to *olh-products/main/_home*.
2. Open *index.md*, and update the links and text as required.



3. Open *jf-previous-versions.md*, and add links to the previous versions of the help.

Update the JustFood Files

These steps detail some of the JustFood files that need to be created or updated for the new release.

1. Navigate to *olh-products/_justfood/files*.
2. Create new versions of the release notes. For example,
   - *jf-across-whats-new-jf-14-applications.md*
   - *jf-across-whats-new-jf-14.md*
3. Update the following files to point to the latest release notes.
   - *jf-across-whats-new.md*
   - *jf-across-whats-new-applications.md*
4. Update the links to the application help URLs in the following files:
   - *jf-powerbi-powerbi-for-justfood.md*
   - *jf-floor-about.md*
   - *jf-plant-about.md*
   - *jf-scheduler-about.md*
5. Update the *toc.md* file.
   - Under *Working with JustFood>Release Notes*, add link to latest release notes.
   - Under *Working with JustFood>Applications>Application Release Notes*, add link to latest application release notes.

Update the JustFood Release Notes File

1. Navigate to *olh-products/_justfood/files*.
2. Update the JF release notes file (*jf-across-whats-new-jf-<version#>.md*) as required.
   - Update metadata and heading
   - Update link to video file when known.
   - Update the cumulative update version that JF was tested on (when known)
   - Update link to application release notes.
   - Add Microsoft changes as appropriate. Depending on size of MS release, might not be possible to include all changes.
   - Add JF changes.

Update the Application Release Notes File

1. Navigate to *olh-products/_justfood/files*.
2. Update JF application release notes file (*jf-across-whats-new-jf-<version#>-applications.md*).
   - Update metadata and headings
   - Add application changes.

Update the Application Files

1. Navigate to *olh-products/<application>/files*.
2. Update the JustFood requirement and the version number
3. Update any links that go to the JustFood online help to point to the new help URL.
4. Add new information as required.

Update the Video Files

1. Navigate to *olh-general/videos/files*.
2. Add the latest release notes video to *jf-video-jf-release-notes.md*.

Build the Final Online Help Files

Once you're finished with the help systems, and need to publish them to the Help Server, you can generate the help systems.

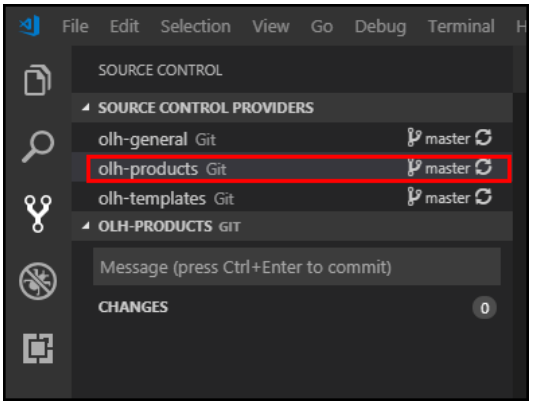
Select the Version Branch

This step is to ensure that you're in the correct branch when building the files.

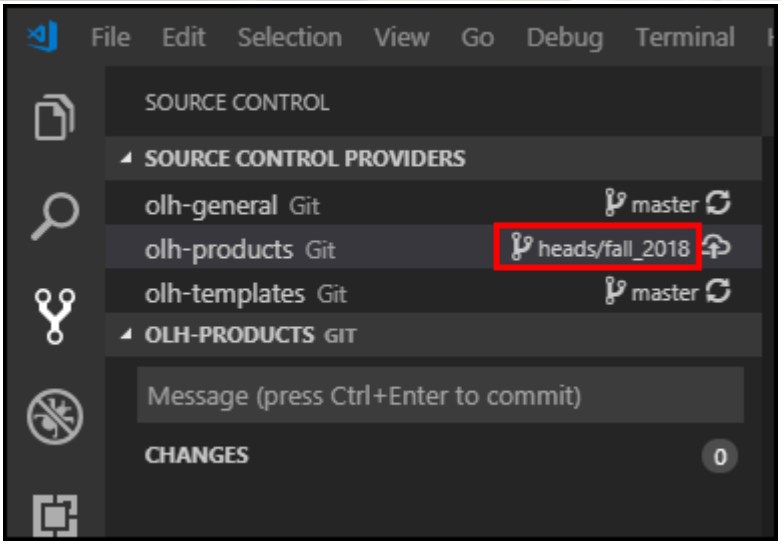
Important: It's important that you pay attention to which branch you are in when making changes, and when publishing your changes. If you make changes in the master, but do NOT publish the changes to DevOps, and then switch to another branch, those changes will then apply to the other branch.

1. In Visual Studio Code, click the Source Control icon.

2. Under Source Control Providers, select *olh-products*.



3. Select the branch that you want to generate the help for.



Update the General Logo Link

You need to update the logo file that is used by the files and help contained in *olh-general*.

Important: The general logo link should point to the newest LIVE version of the help.

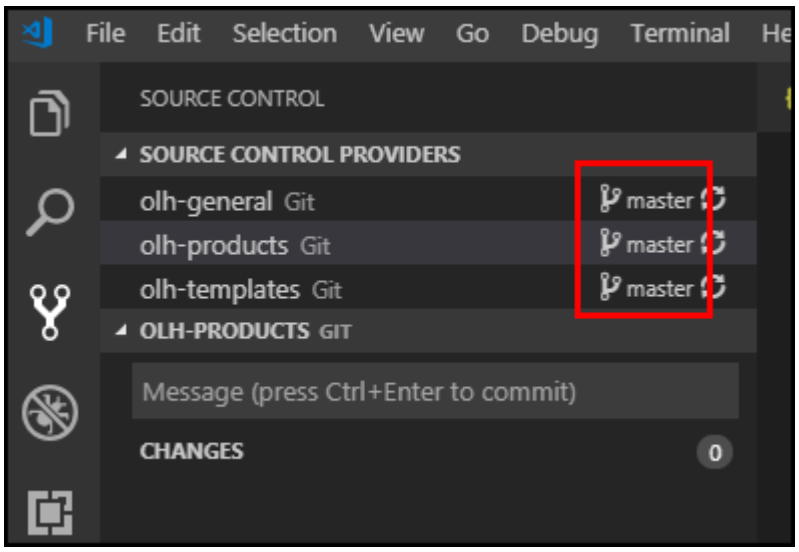
1. Navigate to *olh-templates/justfood_default/partials/logo.tmpl.partial*.
2. Update the URL to point to the latest URL for the main Help page. For example, if the Spring 2019 help is about to go live, then point to http://jfhelpbc.ibzure.com:49007/spring_2019/en-us/index.html.

Generate the Help Systems

Use DocFx to generate a help system using the *docfx.json* file.

Note: See DocFx for getting DocFx set up.

Important: You need to be aware of which branch you are in. When generating files for the master repo, your source control providers should look like the following:



1. Delete all of the files in *C:\OnlineHelp\JustFood\Generated_Files* that you'll be generating new files for. This removes any obsolete files.
2. In Visual Studio Code, right-click the **docfx.json** file for the help that you want to generate.
3. Choose **Open in Terminal**.
4. In the Terminal window, enter

   ```
   docfx docfx.json
   ```

   Files are created in *C:\OnlineHelp\JustFood\Generated_Files\<help folder>*.
5. If you want to host the help locally, enter

   ```
   docfx serve <location of generated files> -p <port number>
   ```

   For example,

   ```
   docfx serve C:\OnlineHelp\JustFood\Generated_Files\Spring_2019\justfood
   -p 9090
   ```

   To open a locally hosted site, enter the following URL into a browser:

   http://localhost:<port number>/files/<file name.html>

   For example,

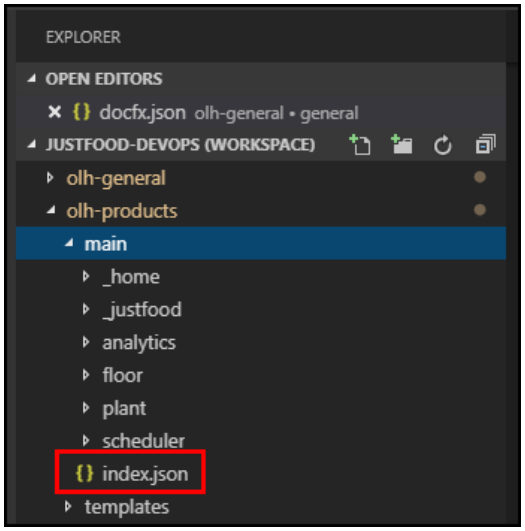
   http://localhost:9090/files/index.html
6. Repeat the steps for the remaining help systems.
   - olh-general
     - general
     - videos
   - olh-products>main
     - _home
     - _justfood
     - analytics
     - floor
     - plant
     - scheduler

Build the Main Search File

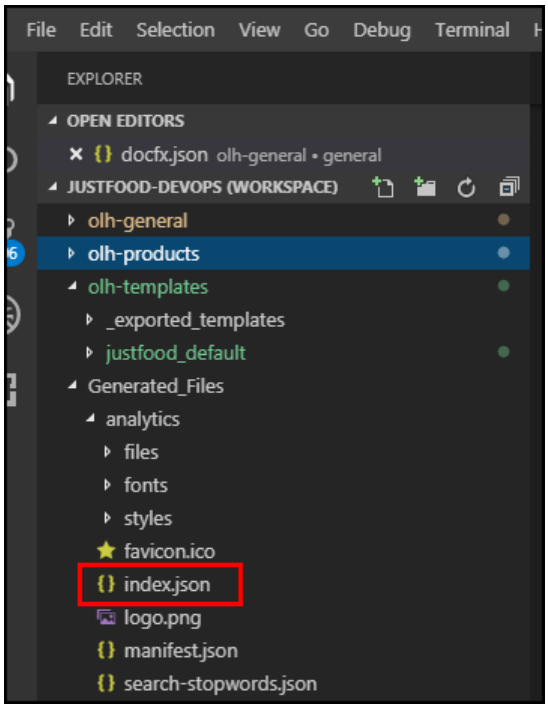
You need to build the file that allows all of the help systems to be searched from the main Home page.

Note: Do not add information from the *olh-general>general* folder.

1. In Visual Studio Code, open the *index.json* file located in *olh-products>main*.

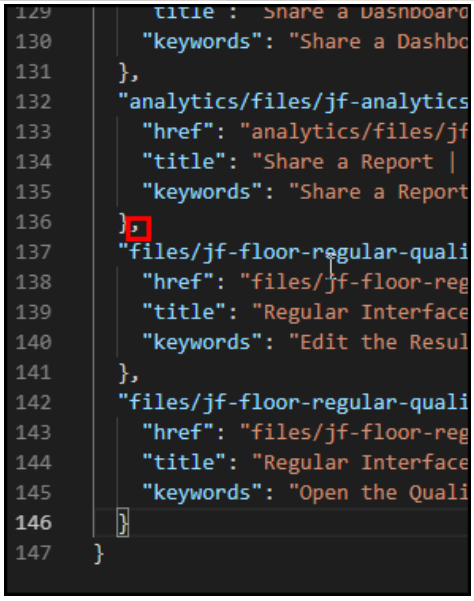


2. Delete the information in the main *index.json* file, except for the first and last brackets { }.
3. Open a product *index.json* file from *Generated_Files*.

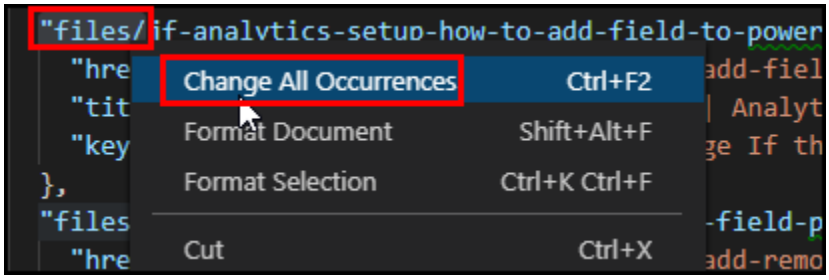


4. From the product *index.json* file, copy the information starting at line 2 to the second last line. You don't need to copy the first and last brackets { }.
5. Paste the information into the main *index.json* file.

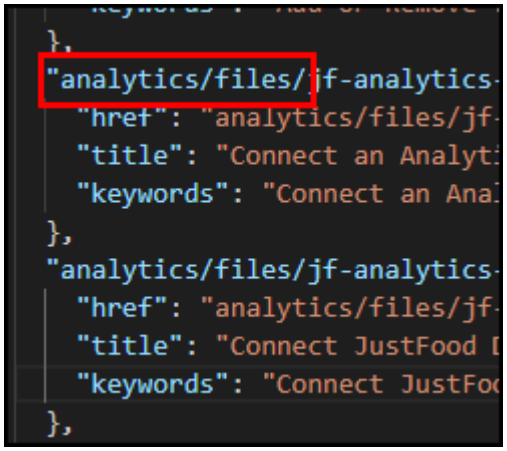
   Note: If pasting after other entries, you need to add a comma after the bracket.



6. Select *"files/*, then right-click and choose **Change All Occurrences**.

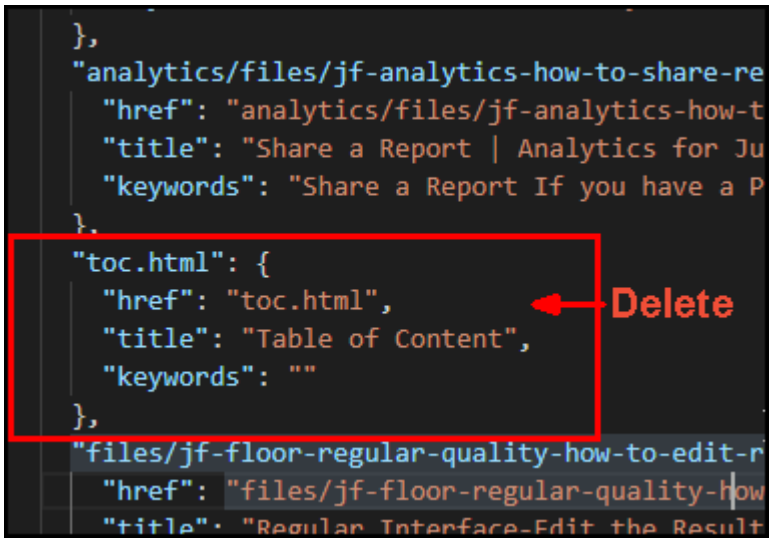


7. Press **<Delete>**, and then replace with product information as required:
   - Home_page: Nothing needs to be replaced
   - JustFood: replace *"files/* with *"justfood/files/*
   - Floor: Replace *"files/* with *"floor/files/*
   - Plant*:* Replace *"files/* with *"plant/files/*
   - Scheduler*:* Replace *"files/* with *"scheduler/files/*
   - Analytics*:* Replace *"files/* with *"analytics/files/*
   - Videos: Replace *"files/* with *"http://jfhelpbc.ibzure.com:49007/videos/files/*



8. Repeat steps 3. to 7. for the remaining help systems.

9. Search for *toc.html*, and delete the toc entries.

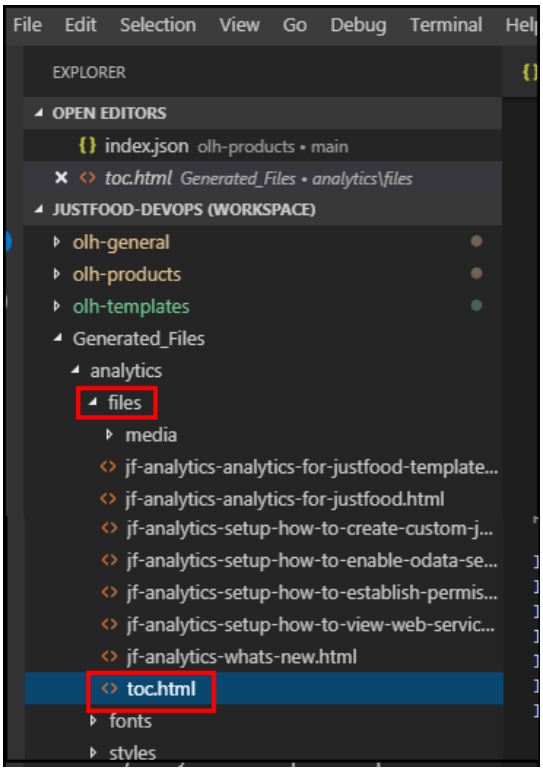


Update the TOC Files

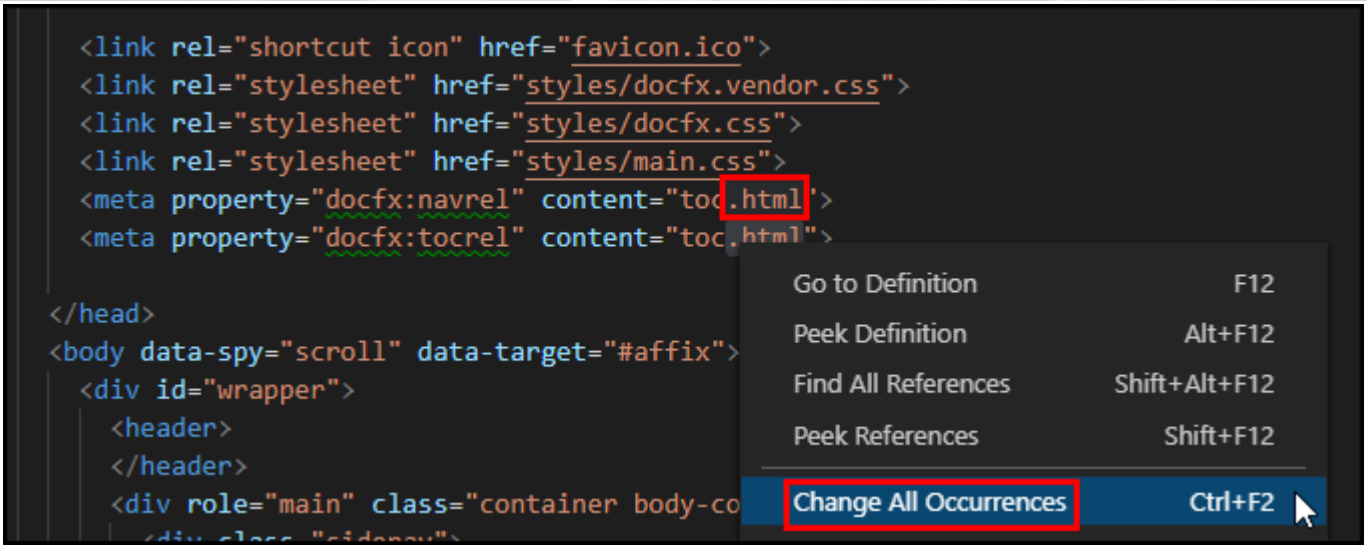
This removes the .html from the TOC files, and allows the files to appear in the help's breadcrumb.

Note: The steps do not need to be performed on the *toc.html* files in the *home_page* or *olh-general>general* folders.

1. In Visual Studio Code, open a product *toc.html* file located in *Generated_Files/<product>/files*.

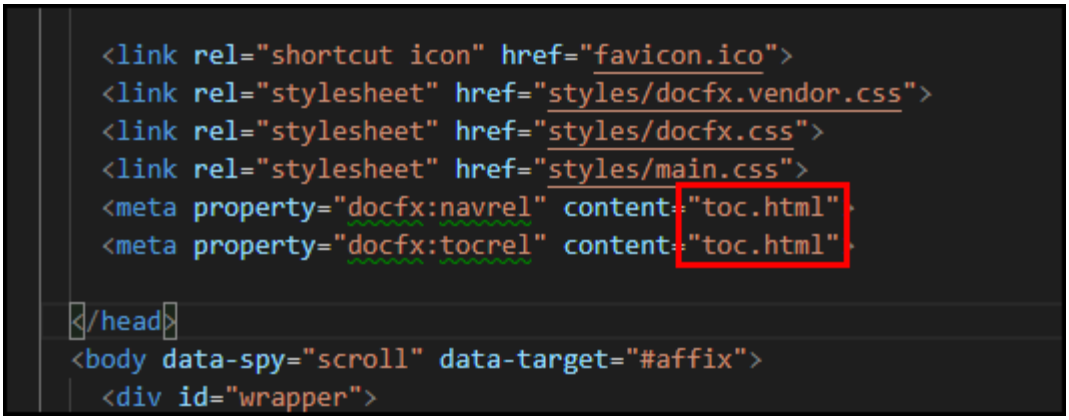


2. Select *.html,* then right-click and choose **Change All Occurrences**.



3. Press **<Delete>** to delete all of the .html entries.
4. Add *.html* back to the two *toc* entries.

   Note: Not sure if this step is actually needed.



5. Repeat the steps for the remaining product help systems.

   Note: Do not update the *toc.html* file in the *home_page* or *general* folders.
6. Publish the online help files.

Publish the Online Help Files

Copy the Files to the Help Server

Once a help system has been generated, and the files have been updated as required, you need to publish the files to the help server.

1. Connect to the Help Server from a remote desktop. (ibhelpeast.ibzure.com)
2. Create a version folder in *C:\inetpub\wwwroot\JustFood_BusinessCentral_Help*. For example, Spring_2019 or Fall_2019.
3. Copy the required files from *C:\OnlineHelp\JustFood\Generated_Files*.
4. Paste the files to the **Downloads** folder on the Help Server.
5. Copy the files from the **Downloads** folder to the appropriate folder in *C:\inetpub\wwwroot*.
6. Copy the contents of the *en-us* folder to the following folders:
   - *en-ca*
   - *fr-ca*

   This allows English-Canada and French-Canada versions of the product to open JustFood help even if the help isn't language specific.

Update the Context Sensitive Help

These steps explain how to populate the necessary table to make the JustFood context-sensitive help work.

These steps do not work for report objects.

Add Links to the Page help

This is so a page object opens the correct Help page. This does not work for report objects.

You need to know the page object number, and the name of the help page that will be opened from the page.

Note: These steps will change for extensions.

1. Open Dynamics NAV Development Environment.
2. Choose **Tools>Object Designer**.
3. Select *Codeunit 23019999 Initialize Page Documentation*.
4. Click **Design**.
5. Add `InsertPageDocumentation(23019000,'jf-skype-about');`

   where

   `23019000` is the page object number

   and

   `jf-skype-about` is the name of the help page
6. Choose **File>Save**.

7. Populate the Page Documentation table 2000000198.

   This populates the table and allows the JustFood context-sensitive help to work.

   a. In JustFood, open the JustFood Setup page

   b. Choose the **Initialize Page Documentation** action.

Send Populated Codeunit to Developer

After the *Codeunit 23019999 Initialize Page Documentation* has been populated, you need to send the codeunit to a developer.

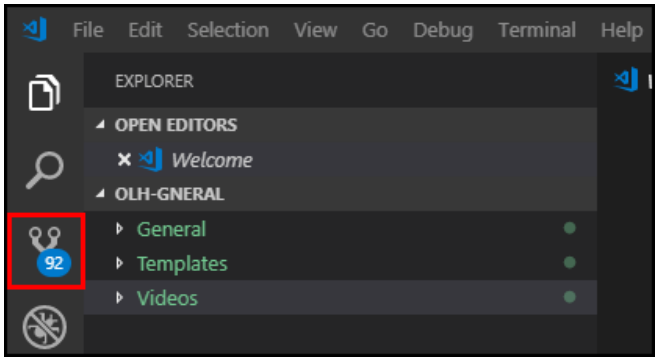
1. Open Dynamics NAV Development Environment.
2. Choose **Tools>Object Designer**.
3. Select *Codeunit 23019999 Initialize Page Documentation*.
4. Choose **File>Export**.
5. Save the COD23019999 as a .fob.
6. Send the fob to the developer assigned to context sensitive help.

Synchronize your Changes with DevOps

Commit your Changes

Once you have completed your changes to the folders and files, you need to commit the changes, and then synchronize with DevOps, which adds your updated files to DevOps.

In Visual Studio Code, if there are changed files, the Source Control icon will show a number.

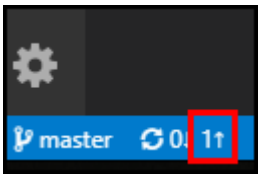


1. In Visual Studio Code, click the **Source Con**trol icon.
2. In the Message box, enter the reason for the changes.
3. Click the **Commit** (Checkmark) icon.

   This checks in your changes to your local copy of the repository.

4. In the bottom left corner of Visual Studio Code, click **Synchronize Changes**.

   The number shows how many changes there are to send to DevOps

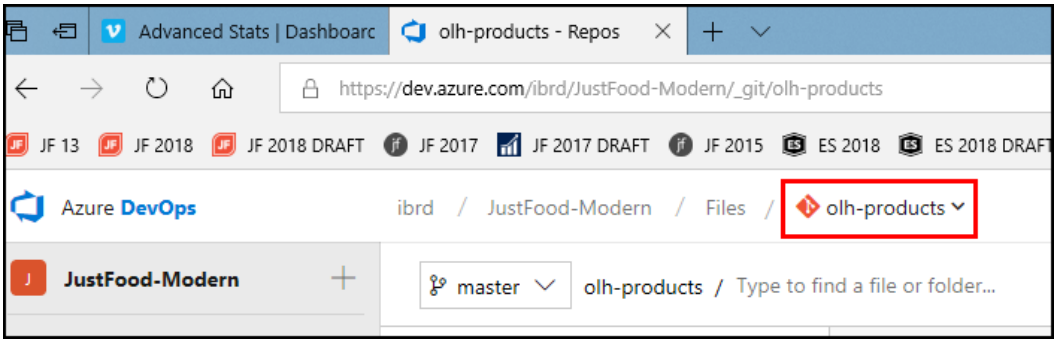


   When adding files to a new repository, click **Publish Changes**.

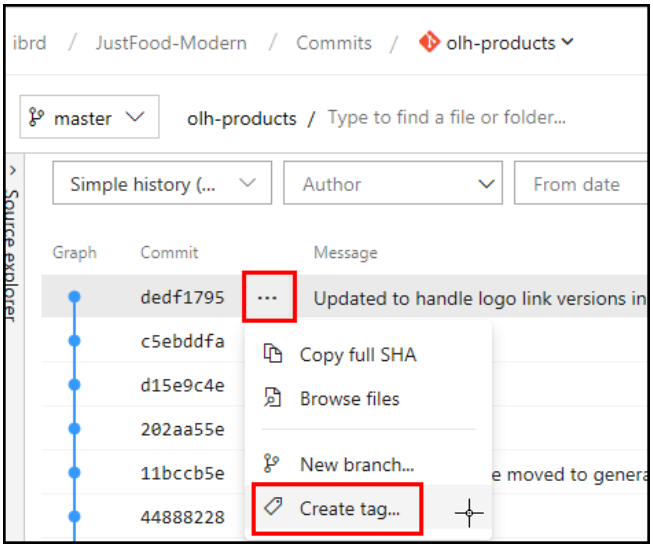
Tag the Files

When the documentation is finished for a release, and before you create a branch, you can tag the files with the release number so that you can find the files related to a specific release.

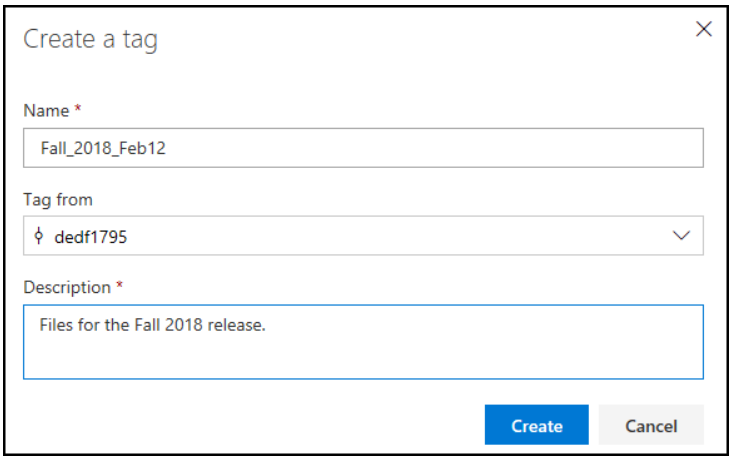
1. Open DevOps (https://dev.azure.com/ibrd).
2. Select the appropriate project.
3. Click **Repos**.
4. Select the repo that you want to tag.



5. Under **Repos**, click **Commits**.
6. For the commit that represents the latest changes, click **More Actions** and then select **Create tag**.



7. Enter a tag that explains what the tag is for, and enter a description.



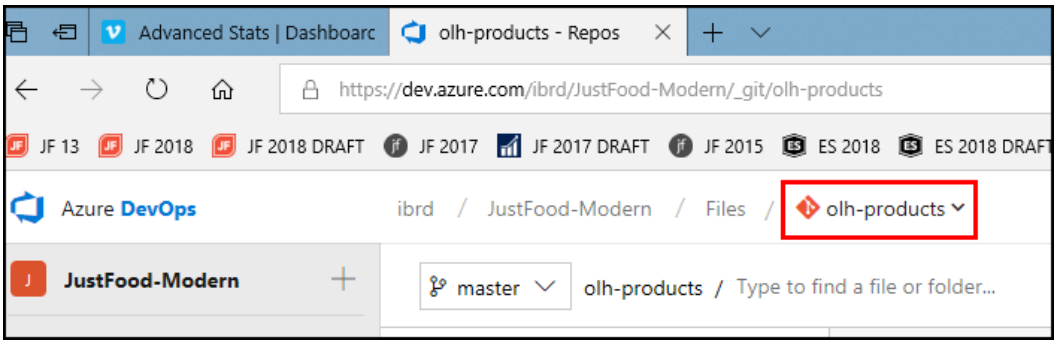
8. Click **Create**.

Create a Branch

Create a branch of the release that was created. This allows you to update and publish previous help versions if needed.

> **Important**: The master is always the latest version of the help that you're working in. The branch is the older version of the help.

1. Open DevOps (https://dev.azure.com/ibrd).
2. Select the appropriate project.
3. Click **Repos**.
4. Select the *olh-products* repo.



5. Under **Repos**, click **Branches**.
6. Click **New branch**.
7. Enter the branch name. The name should be for the older version of the help. For example, *fall_2018*.
8. Select *master* in the **Based on** field.
9. Click **Create branch**.