

Dependency Manager

Overview

The Dependency Manager is a feature that manages linking among internal files. It is used to help preserve the integrity of links leading to content maintained in OU Campus, including binary files if [Binary Management](#) is enabled. Dependency Manager maintains the correct link association when files or directories are moved or renamed. All the files that link to a moved or renamed file or directory are automatically republished to maintain the correct link. It also tracks the deletion of content and creates a notification in the Broken Pages list view.

Dependency Manager tracks links within a site and among sites within an account. This includes any file on any site within that account, but not sites that are contained with other accounts. The Dependency Manager automatically inserts dependency tags after it has been activated by an administrator and runs scans. For example, if Page A exists in OU Campus, and there are 100 other pages in OU Campus with links to Page A, these links are automatically updated if Page A is renamed or moved. If Page A is deleted, the 100 pages are also listed on the Broken Pages report found in **Reports > Required Actions > Broken Pages**. Level 9 and 10 administrators have complete access to the report. Users with an authority level of 8 or lower can view and perform actions only

on the content to which they have access.

Dependency Manager is an optional feature which must be activated at the account level before it can be enabled for each individual site.

Dependency Scans

When Dependency Manager is first activated for a site, all existing links in the site must be converted from their normal URL state (absolute or root-relative) into dependency tags. A Level 10 administrator must run an initial Dependency Scan on each site in order to replace any paths in `<a>` and `<link>` tags for the `href` attribute. This is achieved in **Setup > Sites**, by hovering over the site in question and clicking on **Scan > Dependency Scan**. For more information, visit the [Site Actions](#) page.

After the initial Dependency Scan has been run and Dependency Manager is activated for the site, the system will automatically include the dependency tag for all new internal links, provided that users browse for the file to be linked (as opposed to copying and pasting the URL into the field).

It is recommended that any Dependency Manager scan or revert be run during off-peak hours.

Prior to activating Dependency Manager and scanning the sites, it is advised that each site be exported. This will allow for a complete revert, using Zip Import, back to the site's original state and URL configuration for links, should it be decided that utilizing Dependency Manager is not appropriate for the site. It is important to note that this will revert the site to the state it was at the time of export, and if any changes have been made since then, these will be lost.

Scanning Multiple Sites in an Account

If there are multiple sites within an account that need to be scanned, either for the first time or after all DM tags on the site have been reverted back to normal URLs, each site needs to have Dependency Scan run on it twice, with the exception of the last site in the list. These scans need to be run in two passes.

The first Dependency Scan pass should scan each site one after another, and the second pass should start from the first site scanned and scan each one again with the exception of the last site. For example, if an account includes 10 sites, 19 total scans would need to be performed in order to ensure that all links among sites include the correct dependency tags.

The following example can be used as a reference:

1. There are three sites in an OU Campus account; all three sites have Dependency Manager turned off and all links on the pages are standard URLs.
2. On the first site, turn on Dependency Manager (DM) and perform a dependency scan to get all the page paths into the database. As part of this scan, all the links to pages that are in the same site are replaced with dependency tags (e.g. `{{f:####}}`).
3. The DM database now knows about all the pages in site one, but knows nothing about the pages in sites two or three, and therefore treats them as external links. Any links on site one that point to pages on site two or three are still standard URLs, and all links on sites two and three are still standard URLs.
4. Now, activate DM on site two and run a dependency scan. As part of this scan, all the links to pages in site two are replaced with DM tags, as well as links to pages in site one, as the DM database already knows about site one. Now, all links on site two that point to site one or two are DM tags, but links on site two that point to site three are still standard URLs. All links on site 3 are still standard URLs.
5. Finally, activate DM on site three and perform a dependency scan. As part of this scan, all the links to pages in site three are replaced with DM tags, as well as links to pages in site one and two, as the DM database already knows about site one and site two. Site three is now completely up to date, and all internal links on the site are dependency tags.
6. Now that the DM database knows that links to sites two and three are indeed internal links, perform a dependency scan on site one again to get the links in site one that point to site two and three to be updated with dependency tags.
7. Perform a dependency scan on site two again, to get the links in site two that point to site one and three to be updated with dependency tags.
8. Now all internal links on all sites in the account have dependency tags instead of standard URLs.

Adding Links when Using Dependency Manager

When using Dependency Manager, links are inserted in a fashion similar to when Dependency Manager is disabled. Links can be added to the markup for a page in many ways, but only content that is browsed to with a file chooser has the tag automatically inserted; otherwise, an additional scan can be run. The ways of adding links are:

- Using Insert/Edit Link via the WYSIWYG
- Using the Source Editor

- Using the HTML Source Editor (not automatic)
- When Binary Management is enabled with Insert/Edit Image via the WYSIWYG
- Integrated into template design

When the link is inserted, the Link URL is shown as a dependency tag, which starts with a “d” or an “f”. These are similar in syntax to those tags used for designating assets.

When using a file chooser to select the file, once the appropriate page is selected, a dependency tag is shown in both the modal and the code view of a page. The page path is shown below the field in the Insert/Edit Link modal.

Links can also be added without browsing (for example, in the HTML Source Editor) and then a directory scan can be performed to convert the paths to dependency tags.

Example syntax for the tags are shown as follows:

- `{{d:####}}` — Directory tag
- `{{f:####}}` — File/Page tag
- `{{a:####}}` — Asset tag
- `{{s:####}}` — S-tag: Used with XML/XSL templating to pull content from a PCF file on staging

Within Gadgets

The Gadgets sidebar includes the Dependency Tag Info gadget, which provides the search functionality to find a dependency manager tag and also features the reporting information about the tag, such as the type of tag it is, the associated product name, the path to the product, the site under management, and a linked list of subscribers.

For more information, visit the [Dependency Tag Info Gadget](#) page.

With Assets

The Dependency Manager supports linking from within an asset; dependency tags can be added when creating and editing assets, and when the asset is inserted on the subscribing page the dependency tag is included in the content.

With Multi-Target Publish

Dependency links can be selected across publish targets if Multi-Target Publish is configured and in use for the account. Dependency Manager and dependency tags only pertain to pages maintained within OU Campus.

With Binary Management

With the advent of Binary Management, binary files, such as images, PDFs, and other documents, are managed by Dependency Manager when Binary Management is enabled for a site and binary files are uploaded to the staging server in OU Campus v10. Dependency tags are used to track and manage binary files on staging and production in a fashion similar to other linked-to files. For example, when file linking is managed by the Dependency Manager, if a binary file is renamed, moved, or published, then pages with links to that dependency are updated.

Unpublished dependencies included images can be included with a page publish by selecting to include unpublished dependencies.

When Dependency Manager is used with Binary Management, binary files are treated like all other files managed by OU Campus, offering many advantages to both users and administrators, including reporting.

Example of a Dependency Tag in WYSIWYG

Insert Link

Basic | Advanced

URL **{{f:7322496}}**
/academics/library.html

Text to display Library

Title

Target None

Class (not set)

Cancel OK

Examples of Dependency Tags in Source Editor

```
Save Save As Undo Redo Find Link Refresh Syntax XML Theme Default A A
<h3>Unlimited Resources</h3>
<p>The GU Library boasts over 1.6 million books, articles, maps, and media covering all disciplines of study. In addition, the online catalog provides a plethora of documents that are easily accessible anywhere in the world. Private and public tours of this architectural landmark are available on a daily basis.</p>
76 <p><a href="{{f:7322496}}">Library</a></p>
77 <p><a href="{{f:7360321}}">Example</a></p></ouc:div>
78
79
```

More About Links and Linking

If a site has Dependency Manager turned on, then a Dependency Manager tag is created when linking to any file on any site within that account, as long as the target file is under management of OU Campus. This is irrespective of the target site's Dependency Manager setting. The only exception to this is if the file on the target site has never been recorded by the Dependency Manager in the database. Two common causes of this are:

1. The target file existed on the site before Dependency Manager came into being and that file has not been edited in a way that would cause this tag to be properly updated.
2. The target file was uploaded to staging via SFTP or WebDAV and also has not been touched in a way to cause the database to assign a Dependency Manager tag to the file.

In these cases, when linking, the appropriate text-type URL to the file is returned.

The other class of selectable files that only return text URLs are files on a target production server that are not under OU Campus management.

Among Sites

If one site has Dependency Manager enabled and another does not, the linking behavior is determined by the site record of the site doing the linking. So, for example, if Dependency Manager is on for Site A within an account, the link to Site B with the Dependency Manager off is still tracked with a dependency tag. However, if Site B links to Site A, there is not the ability to add a dependency tag in Site B.

When Publishing

When pages are published that contain dependency tag links, the dependency tag is replaced with the URL to the target page. The Dependency Manager supports both root relative URL links (e.g., /folder/folder/file.ext) as well as absolute URL links (e.g., http://www.college.edu/folder/folder/file.ext). The Dependency Manager does not create page relative links (although it can read and convert them into dependency tags when the site is scanned). The Dependency Manager will create either root relative or absolute URL links when pages are published based upon what the current access setting for URLs is for each page.

Publish Threshold

The Publish Threshold setting defines the maximum number of pages to be auto-published when an asset is republished or a page is moved or renamed when Dependency Manager is being utilized. If the threshold is set to 0, there is no limit on the number of pages that can be auto-published. If a different limit is set, and publishing the asset, or renaming or moving a directory or page will require more pages to be published than allowed, an administrator will have to publish the asset or make the change to the directory or page.

About Unpublished Dependencies

The unpublished dependencies publish option is available with a site, directory, or page publish, and lets a user choose whether or not to include in the publish any additional files that the page is dependent upon for content. This applies if the additional content has not yet been published to the target server. For example, if Page A has been created but not yet published, and Page B, which is being published, links to it, selecting to publish the unpublished dependencies will ensure that Page A is also published. This also extends to any binary files that have been inserted on a page, but not yet published to the production server.

This action prevents broken links from existing on the current page being published. This checkbox at publish time is only shown if the Dependency Manager is being used on the site, dependency tags are used in the page being published, and one or more of those dependency tags refer to content that has not yet been published to the target server.

S-Tag and Triggered Publish

The S-tag uses the dependency tag syntax, is preceded with s: instead of d: or f:, and is accessible through file choosers, which are commonly used in the parameters of a page. When using the tag on a page, that page executes a triggered publish when the source file is republished. The XSL can use the path generated by the tag to display content being maintained on a different page. This is similar in concept to an include file or an asset. When the original file is updated, the page containing the content is also published automatically. One example of how this is utilized is when a mobile page uses content from the main web site. When the content on the main web site is updated and published, the mobile page is also published with the updated content.

Renaming or Moving a Directory or File

Dependency Manager maintains the correct link association when files or directories are moved or renamed. All the files that link to a moved or renamed file or directory are automatically republished to maintain the correct link.

When a file or directory is either moved or renamed, a prompt that all dependent pages will be republished will appear. This will republish the page based on the current version of the page on the live production server with the new dependency links. If changes have been made but not published to the live site, those changes will not be published to the live production server, but will be available in OU Campus and still saved on the staging server. This includes backed-up versions.

About HTTP Root

If HTTP Root is changed in the site settings, the entire site should be republished to ensure the links are updated. The HTTP Root setting is used to indicate the location of the site on the web server as accessed through HTTP and should usually correlate to FTP Home or FTP Root.

File Types

Before a file is scanned, OU Campus determines if a file contains HTML according to file extension. The Dependency Manager scanner utilizes this information to determine if a file contains HTML. The list of extensions that determine if a file contains HTML is as follows:

- .html
- .htm
- .xhtml
- .shtml
- .shtm
- .jsp
- .php
- .php3
- .phtml
- .asp
- .chtml
- .cshtml
- .dna
- .cfm

- .inc
- .aspx
- .mht
- .ssi

Dependencies Reporting

When utilizing Custom Reports to produce a Products or Assets report, the report lists the dependency tag for the content. The reports also include a list of subscribers to the dependency.

Broken Links

If content is deleted, a broken link is created. The system has various indications for broken links. Dependency Manager provides informative reporting for broken links and broken assets. When a dependency is lost, such as a linked-to page or a subscribed-to asset is deleted, the dependency tag, path, and file name are shown within asterisks, e.g. ***** Broken f:1234 /training/about/filename.html*****.

Broken links are reported in the following locations:

- WYSIWYG Editor
- Page-based Link Check
- Publish-based Link Check
- Site Check
- Custom Reports
- Broken Pages

With Recycle

If a recycle bin has been set up for the site, when recycling a file, a prompt indicates how many links will be broken if the page is recycled. Keep in mind that the warning will not appear prior to deleting a directory or file if the recycle bin has not been configured. After the file is recycled, the system displays the pages have broken links as a result.

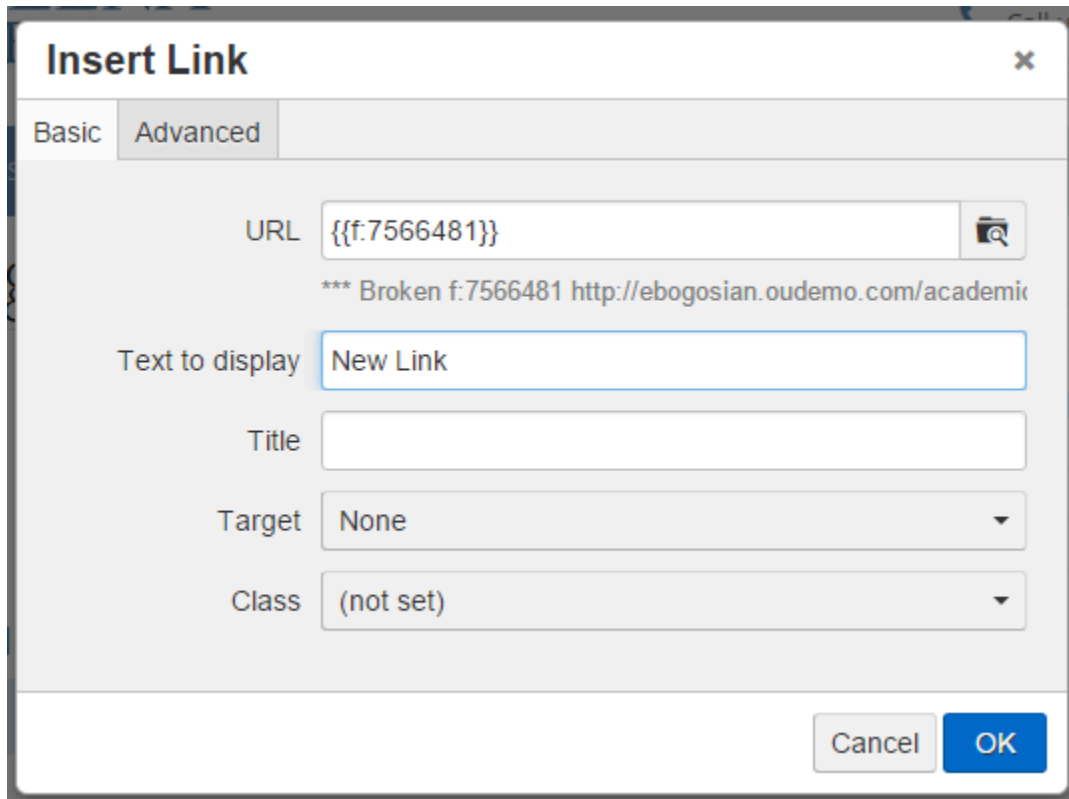
With Source Editor

Source Editor reports any broken links within the source code by highlighting the relevant dependency tag red.

```
csspath="/_resources/ou/editor/content.css" cssmenu="/_resourc
class="maincontent"/><p> What a piece of work is a man! How no
express and admirable! In action how like an angel, in apprehe
animals. And yet, to me, what is this quintessence of dust? Ma
72 <p><a href="**{f:7566481}**">Broken Link</a></p></ouc:div>
```

Within the WYSIWYG Editor

OU Campus provides a visual indication that a link is broken within the WYSIWYG Editor as well. It includes the note that the link is broken, the actual tag, and the file name with path that is being linked to. To view this, position the cursor on the link or select the link, and click Insert/Edit Link. The notification is displayed below the Link URL field.



Insert Link [X]

Basic | **Advanced**

URL: [Magnifying Glass]

*** Broken f:7566481 http://ebogosian.oudemo.com/academic

Text to display:

Title:

Target: [v]

Class: [v]

Cancel OK

Additional Scenarios and Considerations

When a directory with dependency tags pointing to files within the same directory is copied, the links in the files of the copied directory either have to be manually updated or OmniUpdate Customer Support can be contacted to help with this scenario.

This may include:

1. Copying a directory to create a new directory within the same site with the same starting content.
2. Copying a directory's files to convert it into a new site.

To take a directory from an existing site that is using dependency tags and make it into a new site of its own (e.g., <http://www.college.edu/athletics/> is to become <http://athletics.college.edu/>), specific steps must be taken to obtain the desired outcome. It is important to note that most of the following steps are necessary for this process with sites that do not use dependency tags. The instructions below add the steps needed when Dependency Manager is used on the source site.

1. Perform the process during a downtime (evening or weekend).
2. In the **Setup > Sites** view, select the **Revert** tool to remove all dependency tags from the source site.
3. Use the **Zip Export** tool to download a copy of the site from OU Campus.
4. Create the new site record the same way sites are normally created in OU Campus.
5. On the workstation being used, open the exported zip file and extract the folder desired to be the source of the new site.
6. Create a new zip file of just the directory that is the origin of the new site.

7. In the new site in OU Campus, from the top **Content > Pages** view, use the **Upload** tool to import the zip file just created (select the **Ignore containing folder** check box of the **Zip Import**, if there is one).
8. Once the files are uploaded, on the new site use the **Find and Replace** from the **Content** menu to search for the paths that have the containing folder, which was ignored upon import, and remove the now incorrect folder name (e.g., Find: /athletics; Replace with: /).
9. Use the **Find and Replace** to correct the links from the new site to the source site, usually found in header and footer navigation and links (e.g., any paths that are not on the new site, such as /admissions... would become http://www.college.edu/admissions...).
10. On the source site, use the **Find and Replace** from the **Content** menu to search for the paths that have the now removed folder and change them to the new site URL (e.g., Find: /athletics...; Replace with: http://athletics.college.edu/...).
11. Use the scan tool from **Setup > Sites** view to scan the new and old sites to create new dependency tags for each site.
12. On the source site, delete the now obsolete /athletics... directory.
13. If any URL paths were missed in step 10, any pages on the source site with links to the now deleted directory are shown via **Reports > Required Actions > Broken Pages** list view.
14. Fix the pages listed in the **Broken Pages** report; the two sites will now have correct dependency tags for each site and for sites within the account.
15. Populate the new site's **Production Server** with any needed binaries to support the new site.
16. Continue to setup and configure the new site as would be done for any other site.

For questions regarding this process, or to have only the individual directory being copied reverse scanned (Revert), contact Support.

Dependency Manager and RFC2396

The Dependency Manager scan and revert scan is RFC 2396 compliant. In accordance with the specification, the Dependency Manager can convert URLs with very complex % encoded paths. URLs with unencoded spaces are not accepted.

Use Case

The Dependency Manager supports the use of %20 instead of a space in a file path works. A file can be uploaded via FTP that has spaces in the name. Insert the image on to the page and note that the inserted image has %20 in the file path. Enable both the Dependency Manager and Binary Management, and run a scan. The image path will be converted into a dependency tag. Run a revert scan and the link is converted back to %20. The %20 is the escaped encoding for the US-ASCII space character.

Unreserved/Encoded

The following are recommended characters, as using them in file names will translate into easy to read URLs. They are unreserved and never encoded.

- Uppercase letters
- Lowercase letters
- Numbers

- The following special characters: - _ . ! ~ * ' ()

Reserved/Encoded

The following list of reserved characters are encoded by the Dependency Manager, but will result in URLs that are difficult to read. They are safe to use, but are not recommended:

- The space character
- The following special characters: " < > # % ; = > ? [\] ^ ` { | }

Reserved/Unencoded

The following reserved characters are not encoded and are safe to use in paths as well:

- / : @ & + \$,

All other UTF-8 characters may or may not be encoded depending on whether the file that contains the URL is UTF-8 or ASCII.

Important Information and Reminders

1. Dependency Manager tracks and updates links within the following file types and extensions:
 - PCF: .pcf
 - XML: .xml
 - XSL: .xsl, .xslt
 - HTML: .html, .htm, .shtml, .shtm, .xhtml, .jsp, .php, .php3, .phtml, .asp, .chtml, .dna, .cfm, .inc, .aspx,
2. Dependency Manager does not track binary files, such as images (with the exception of those that are products of a PCF), unless Binary Management is enabled. Without Binary Manager, any changes such as file moves or renaming to binary files, with the exception of those that are products of a PCF, will require that any links be updated manually. Otherwise, Dependency Manager works with anything editable from within OU Campus or generated by an editable file from within OU Campus.
3. Dependency Manager does not track links to pages or sites not under management by OU Campus. This also applies to sites that are not in the same account as the current site under OU Campus management.
4. The Dependency Manager is account-bound and can only track and manage URL changes to pages or directories that are within sites in the same OU Campus account.
5. To utilize dependency tags when using the file choosers, `dependency="yes"` needs to be declared as an attribute in a `<variable>` or `<parameter>` node with `type="filechooser"`.
Example:

```
<variable name="leftnav_include" prompt="Left Nav" alt="Choose your left nav" type="filechooser" path="/includes/navs" dependency="yes">/includes/navs/default.html</variable>
```
6. As a best practice, it is suggested that a complete site export be completed on each site prior to activating Dependency Manager to allow for an immediate and complete reversal via Zip Import if the end result is not desired.
7. If a directory or file is uploaded or deleted via FTP, run the scanner on the affected directories or entire site to avoid publishing issues.

8. The following regex can be used to see which links have been replaced with Dependency tags: `\{\{\{ [fd] : \d+ \}\}\}`
9. The Dependency Manager supports linking from within assets; dependency tags can be added when creating and editing assets, and when inserted on a page the tag is honored.
10. Dependency Manager links are case sensitive. As such, it is advised that paths for links to pages managed in OU Campus are always inserted using the browse functionality.
11. Dependency Manager inserts dependency tags into `<a>` and `<link>` tags for the `href` attribute automatically, both when the Dependency Manager scanner is run and when new links are created. However, tags can be manually included elsewhere as desired.
12. Dependency Manager will not automatically add dependency tags to any other types of tags, including PHP, ASP, and JavaScript.

Important Exception: Dependency tags cannot be placed within processing instructions or the prologue, the instructions prior to the opening XML comment. Entering in dependency tags within these will cause an error as the page will not be able to be rendered.

13. Dependency Manager renders root relative links (i.e., `/directory/page`) and absolute links (i.e., `http:www.college.edu/directory/page`), depending on the site settings. The rendered URL to the production site cannot and will not be page relative (i.e., `../page`).
14. If there is production side processing that requires certain formatting of paths, be aware of how the paths will render and either avoid using Dependency Manager where it might be adversely affected or change the production side processing.

If page relative URLs currently exist on any of the pages and it is required that they remain page relative, the Dependency Manager scanner should be run on a per directory basis, not a site-wide basis.

If page relative URLs are required going forward, these will need to be added manually without using the WYSIWYG Editor and Source Editor browse functions or the file chooser.

If the HTTP Root setting is changed in the site settings, the entire site should be republished.

15. Run the Dependency Manager scanner if a page to which others are dependent is restored from the Recycle Bin, or if a directory or file is uploaded or deleted via FTP/SFTP/WebDAV. The scan can be run on only the affected directory or directories.
16. The Revert scan is not reverting the site to its state prior to Dependency Manager being utilized. It is replacing the dependency tag with the root relative or absolute URL, dependent on the site settings.
17. Dependency Manager will not scan or update files within the following folders:
 - OMNI_RESOURCES at root
 - OMNI_ASSETS at root
 - Recycle Bin (if defined) at root
 - OMNI_INF at all levels
18. OU Campus does not contain any tracking of which sites have been scanned. If this is desired, it is recommended that it be completed internally.
19. Reporting is available with details on which dependency tags refer to which pages, and which pages are currently subscribing to those pages.
20. Dependency tags will not be referenced in the Custom Reports if the page or directory was deleted.

21 Dependency tags are supported across publish targets, if Multi-Target Publish is being utilized.

Final Notes

Please be sure to read the [Dependency Manager White Paper](#) prior to contacting Support. It contains information on the benefits, limitations, and considerations.