uSync. Content Edition

How it works

uSync.ContentEdition is primarily a DLL that attaches to the ApplicationStarted Event inside Umbraco, that is it runs every time the application starts/restarts.

On application start it does the following (depending on configuration)

Import

1. It looks for the uSync\Content folder and imports any .content files
2. It looks for the uSync\MediaItems folder and imports and .media files
3. It goes through what it's imported and trys to fix ids/

Export (first time and off by default)

1. uSync walks the content tree and creates a .content file for every bit of content
2. uSync walks the media tree and creates a .media file for every media item and copies the media item to the uSync/Files/ folder

Events

1. uSync attaches to the Save and Delete (actually trashing) events for content and media . so it can capture when changes are made
2. Whenever you save or delete and content or media uSync does an export on that specific item. Handling things like renames and moves on the way.

ID Mapping

on it's own exporting and importing the content works a bit, but it does leave a few issues, mainly the fact that Umbraco uses internal IDs to link items and when you move content those IDs don't remain the same. uSync does some basic id mapping to overcome this

Export mapping.

When content are exported usync searches for IDs within the properties of the content, when it finds these it's replaces them with the unique key for a content item (GUID)

Import mapping

This means when content is imported at the other end uSync has to search through all the properties and attempt to turn those GUIDs back to IDs. It does the following

1. When it creates a piece of content the GUID of this item is written to an import file (/app_data/temp/_usyncimport.xml) alongside the GUID of the content item from the import file.

2. Once all content is imported ID mapping searches the new content items for GUIDs when it finds one it attempts to find a match in the import file.
    a. If one exists then the GUID of the content on this installation can be used to map back to the id.
    b. If one doesn't exist it attempts to use the GUID to find the ID on the local install (because this installation may have created the content not the source)

Slightly more complex

That's the simple view, it is in fact slightly more complex than that, because in-order to keep some sort of consistency, uSync will always attempt to get the Master GUID (the GUID of the content on the system it was first created) – so even when exporting it attempts to go from the ID of the content to its GUID to the master GUID in the import file (if one exists).

This means content can move between many installations, as long as they all agree on the master the GUID should always be the same.

Renaming and Moving

One of the principles of uSync is that it leaves vaugly human readable trace on the installation, that means the name of the folders and content on disk are the names of the content. not the IDs this makes the folders within the uSync folder quite nice to read, and understand, it makes renaming and moving content a pain.

If the ID was used then renaming content wouldn't matter because the ID would stay the same, but our folder structure would just be numbers. So uSync jumps through a hoop or two.

Source File

As well as an import file usync has a source file (app_data/temp/_usyncsource.xml) the source file has one main purpose that is to track the name of content so when it's saved we can tell if it's been renamed.

On every save, uSync checks the name of the content vs the one it has in the source file. If it's changed then it's a rename, so it renames the .content / .media file and it moves folders around.