uSync 1.6 for Umbraco 6.1.6 - Release Notes

uSync 1.6 is a significant refactoring of uSync for umbraco 6.1.6 a number of new features have been introduced and a lot of fixes have been applied .

Mapping Values in data types

uSync can now map umbraco values inside datatypes. where a macro stores an ID of another element in umbraco (such as a content node) usync can map these between installations. Currently usync can map :

- Content Nodes,
- Media Nodes,
- Stylesheets
- Tab Ids

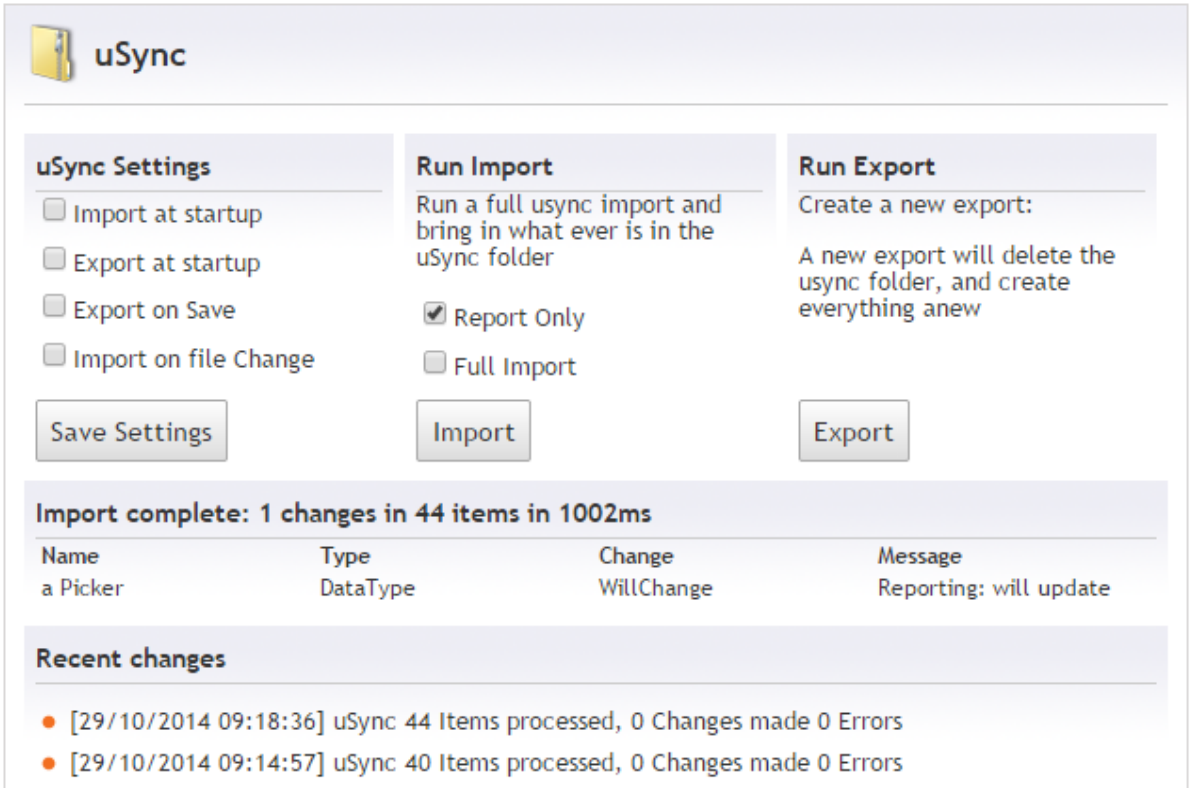
This means uSync work for MultiNodeTree Pickers, Stylesheets in the Rich Text Ediror and uComponents Sub Tabs - it can work for any arbitrary datatype you just need to update the mapping in the usync config file.

```
<MappedDataTypes>
        <!-- MultiNodeTreePicker -->
        <add key="7e062c13-7c41-4ad9-b389-41d88aeef87c" value="content" />
        <!-- uComponents: Legacy MNTP -->
        <add key="c2d6894b-e788-4425-bcf2-308568e3d38b" value="content" />
        <!-- RTE Content -->
        <add key="5e9b75ae-face-41c8-b47e-5f4b0fd82f83" value="stylesheet"
           preValueType="text" propertySplitChar="|" propertyPosition="6" />
        <!-- uComponents SubTabs -->
        <add key="2c787731-cd81-48cd-a94f-4930185bdb58" value="tab"
           preValueType="json" propertyName="TabIds" idRegEx="\d{1,9}"/>
        <!-- ultimate picker -->
        <add key="cdbf0b5d-5cb2-445f-bc12-fcaaec07cf2c" value="content"
           preValueType="text" propertySplitChar="|" propertyPosition="2" />
</MappedDataTypes>
```

The main data types that contain IDs are already configured in the uSyncSettings.config file, if you want to map a new datatype you can add it to the mappedDataTypes section of the config file. (See the Science Bit at the end of this document)

Reporting Dashboard

uSync has a reporting dashboard, that allows you to run a report against the import to see what will be changed before running the change.



The dashboard will also report the status of any import, and you can email the reports to a specified email address by using the settings in the config file

Rollback

uSync can be configured to perform rollback should an import fail, after an item is imported uSync compared the database with the import and if they don't match it can attempt a rollback to the settings before the import started.

Rollback can be configured to either roll back individual imports or rollback the whole import. It is controlled via the **itemRestore** and **fullRestore** settings in the config file. if either is on then snapshot backups will be taken before an item is imported and used for rollback if needed.

> **Caution:** Rollback will attempt to change settings back even for the slightest of reasons, if any values, properties or descriptions do not match or are not in the same order as on the original import then uSync will mark the import as mismatched, and if rollback is set - roll back the import to the pre Import value. unless you have real critical reasons to do so, i would hold back on setting up rollback.

How Rollback Works

With rollback turned on uSync does the following during an import

1. The import is compared to the database, if there is a difference then an import is started
2. A backup of the database object is taken and stored in a snapshot folder (by default in app_data/temp/usync/backups/)
3. The import is performed
4. the database is once again compared to the import, if there are no differences then the import is marked as successful
5. if the db doesn't match a rollback can be performed
    5.1. For Item Restore the backup file is appled back to the database
    5.2. for a Full Restore, all files in the backup folder are re-applied to the database

Renames

uSync now copes with renaming and deleting various types of object, so if you delete a document type on one site, and import into another will see it deleted there too.

Caution: you need to exercise caution with deletes - because if you delete the wrong document type, you will lose a lot of content.

Renames and deletes are recorded in fileops.config in the root of the uSync folder - you must ensure this file is transferred to any new umbraco site if you want to capture renames and deletes.

uSync 1.6 Improvements and Fixes

The Following improvements and fixes have been made to this version of uSync:

All objects

- Renaming and Deleting
- Improved importing, only importing changes (just like v2.2 but better :))
- Reporting
- Post Import checking and rollback

Document Types and Media Types:

- Renaming / Deleting Tabs
- Changing data types for property types
- Re-ordering of properties

Data Types

- Better support for changing the underlying type of data type
- Content and Media ID Mapping
- Stylesheet Mapping
- Tab ID Mapping
- First time save of properties (removing the need for double saving)

Languages and Dictionaries

- Support for changing language types

Macros

- Support for renaming of Macro Properties

Stylesheets

- Capture of stylesheet properties, saving properties now saves the stylesheet as well

Templates

- Support for changing structure of templates

Known Issues

A. **Renaming document type properties can cause content loss:**

If you rename a generic property in a document type, then during the import uSync will create a new property with that name and delete the existing one, this will cause any content associated with the existing property type to be lost.

Workaround: Prevent orphan property types from being deleted during import

In uSyncSettings.Config - setting *DeletePropertyValues="False"* will stop old property values from being deleted.

```
<DocumentTypes DeletePropertyValues="false" />
```

Note: *this setting is incompatible with Rollback, as the resulting document will never match the import - and hence will always be rolled back.*

The Science Bit : ID Mapping Config

uSync looks for numbers inside datatypes that it thinks might be IDs, without the settings above it would just search every value and every string looking for an ID and then attempting to map it. While it's not very likely it is possible that when searching for Ids uSync could find something that looks like an ID and even mapps to something in umbraco but isn't an ID.

To help prevent this uSync uses the mappings below to narrow down where the IDs are stored.

Identifying the data type.

uSync uses the DataTypeDefinition ID to find an ID within the a usync file this is stored in the Definition value, inside umbraco this is identified as the Property Guid of a Datatype.

Type of IDs stored

uSync supports, mapping content (both content and media are checked), Stylesheets, and Tab Ids

propertyValueType

Within a config file the IDs may be stored in a number of ways depending on the datatype,

text: the ID is stored within a string which may or may not contain other numbers.

propertySplitChar: the string value that splits up the text string

propertyPosistion: the position within the split string that contains the ID

Example the Rich Text Editor control stores all of it's settings, in a single comma delimited value:

```
",code,undo,redo,cut,copy,mcepasteword,stylepicker,bold,italic,bullist,numlist,outdent
,indent,mcelink,unlink,mceinsertanchor,mceimage,umbracomacro,mceinserttable,umbracoemb
ed,mcecharmap,|1|1,2,3,|0|500,400|1046,|True|500|"
```

Within this value is the stylesheet ID(s) for when adding a stylesheet to the Editor. looking at the string above the ID 1046 is the stylesheet ID. for this string we see: propertySplitChar: '|' and propertyPosition: 6

with these settings uSync will search for ids in the string “1046,” (which sometimes can contain multiple ids) and map them between installations.

number: the ID is stored as a number - there is no special mapping for this, uSyns will just search all the preValues for what looks like an ID.

json: Increasingly datatypes are storing information within a JSON string - while JSON settings can be treated as a text string, we can reduce the chance of false mappings by telling uSync exactly where the IDs are mapped inside the JSON.

propertyName: The Name of the Property inside the JSON string

Example: the uComponents: SubTabs datatype stores values in a JSON string
{“TabIds”:[7,11],”SubTabType”:1,”ShowLabel”:false}

In this string 7,11 are tab IDs. we could just search the text- but that increases the chances of us finding another number and thinking that is an ID (for example the 1 might be a valid tab id)
for this type we set propertyName = TabIds and uSync will look for IDs in the string “[7,11]“

What does an ID lookalike?

when searching for IDs uSync looks for numbers that might be valid IDs, most IDs inside umbraco are above 1000 so by default uSync looks for a number between 4 and 9 digits long. (RegEX “\d{4,9}”)

However there are some occasions where the ID might not look like this - most notably when looking for Tab IDs which start at 1 .

you can tell uSync to search for different IDs by overriding it’s regular expression for your datatype setting the idRegEx value will override the search criteria uSync uses for IDs.

for example: idRegEx=”\d{1,9}” looks for an number between 1 and 9 digits long