Data Importer For SharePoint (DIFS) Documentation

Ref	DIFS
Date	03/02/2016
Version	2.8

Document Control Sheet

Revision History			
Date	Change	Version	
12/06/2012	First release	1.0	
03/10/2016	Updated for Office 365 & SharePoint 2016	2.0	
06/10/2016	Added support for reporting to console for Powershell scripting. Added support for managed meta data tagging using CSOM – 2013/2016/Online	2.2	
12/10/2016	Added support for multi-value managed meta data columns. Add example for document set creation. Added support for installation on 32bit Windows.	2.3	
28/10/2016	Added examples for Publishing Pages and Wiki Pages	2.4	
04/11/2016	Added support for ODBC data sources. Fixed check in issue when destination content type includes mandatory fields.	2.5	
14/11/2016	Added SourceDataSetType FileSystemFiles enabling simple folder > SharePoint imports. PerItemImportThrottle Support for multi value tax fields in 2010 (already supported for 2013+).	2.6	

Referenced Sources

Table of Contents

1 Introduction

1.1 What is DIFS?

DIFs is an application for Microsoft Windows which allows you to import data and documents into SharePoint lists and document libraries.

1.2 Why DIFS?

There are a lot of migration tools available that import data into SharePoint and so why develop DIFS?

Many of the available migration tools for SharePoint have a high cost, restrictive licencing model and can be quite limited in their flexibility / ability to address real life scenarios that arise in import / migration projects.

DIFS is free, flexible and leverages other available technologies. It can import from file systems and 100's of OleDB providers. You may use tools like Excel and SQL to cleanse data prior to import and by clever definition of your data sources, SQL statements and configuration files you can achieve complex migration scenarios.

1.3 How does DIFS work?

1. You install DIFS on your PC (See installation).
2. You look at the examples in this manual
3. You create a data source to import from such as;
    a. Such as an Excel spreadsheet, or
    b. SQL database table, or
    c. A simple file system folder.
4. You create an XML based configuration file that tells DIFS how to carry out the import (such as the destination library, credentials and data mappings).
5. You load the XML configuration file in the user interface.
6. You can change some settings in the user interface.
7. You start the import.

1.4 Used DIFS? Worked Great?

Please post your stories, experiences, on your blog or social media. Don’t forget to share the link with the project team.

Please let us know of any improvements.

2 Key Features

Key Features	
SharePoint Versions	SharePoint Server SharePoint Foundation 2010,2013 and 2016 SharePoint Online
Authentication	Supports forms based authentication
Meta data	Allows existing file meta data to be imported
Import Control	Imports can be paused, resumed and cancelled. Import progress is reported to the use interface.
Exception handling	Exceptions can be saved for correction and reprocessing.
Save settings	Import settings can be saved and retained for future use.
Uses Client Object Model	You may run the software on the server or a client PC.

3 Installation

3.1 DIFS

Run the setup.exe or MSI that you downloaded.

Install the 64bit version if you want to use 64bit data sources.

Install the 32bit version if you want to use 32bit data sources.

3.2 OleDB Data Source

You will need to ensure that you have the correct 32bit or 64bit OleDB provider for DIFS installed for the source that you wish to access.

3.2.1 Excel OleDB Data Source

Try

https://www.microsoft.com/en-gb/download/details.aspx?id=13255

or

https://www.google.co.uk/#q=excel+64+bit+oledb+

4 Examples

You will find the matching excel spreadsheets and XML configuration files for most of these examples in the directory into which you installed DIFS.

4.1 Import from a file share or local folder to a SharePoint Document Library

4.1.1 Overview

Take an existing folder and import the files and folders within in to SharePoint. For a more professional result (such as control over destination content types) please refer the other examples on file system migration.

4.1.2 The Source

A local path;

e.g. c:\Import

Or a unc path

e.g. \\myserver\mystuff

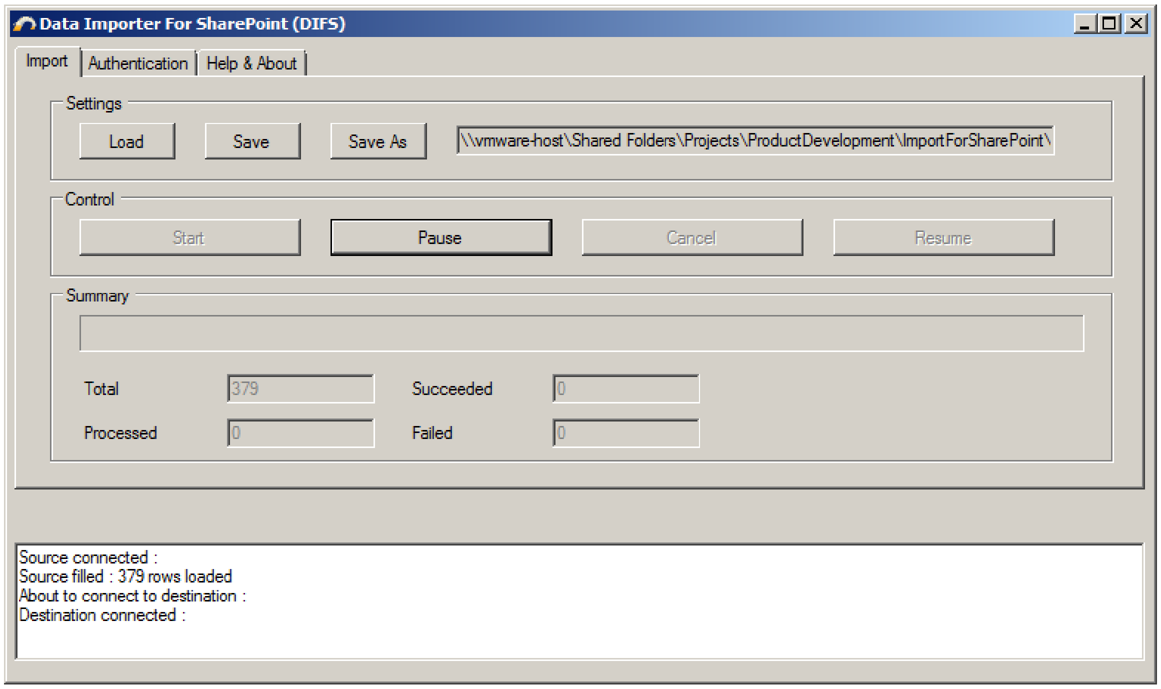
4.1.3 The Configuration

```xml
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <SourceDataSetType>FileSystemFiles</SourceDataSetType>
    <FileSystemFilesDataSetSettings>
      <Path>c:\Import</Path>
    </FileSystemFilesDataSetSettings>
</Source>
  <Destination>
    <AuthenticationSettings>
      <AuthenticationType>Current</AuthenticationType>
      <domain />
      <username />
    </AuthenticationSettings>
    <DestinationItemSettings>
      <DestinationItemType>Document</DestinationItemType>
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
    </DestinationItemSettings>
    <DestinationListSettings>
      <DestinationWebUrlRelative>/sites/SPImportHelper</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/sites/SPImportHelper/Shared
Documents</DestinationFolderUrlRelative>
      <DestinationServerUrl>http://productdev</DestinationServerUrl>
      <DestinationListName>Shared Documents</DestinationListName>
    </DestinationListSettings>
    <SourceColumns>
    </SourceColumns>
  </Destination>
</DataSetImportSettings>
```

4.1.4 The Destination

A document library

4.1.5 The Execution



\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\

Total	379	Succeeded	0
Processed	0	Failed	0

```
Source connected :
Source filled : 379 rows loaded
About to connect to destination :
Destination connected :
```

4.1.6 The Result

The files and folders underneath your source path will be imported to SharePoint.

4.2 *Import from an Excel Source into a SharePoint List*

4.2.1 Overview

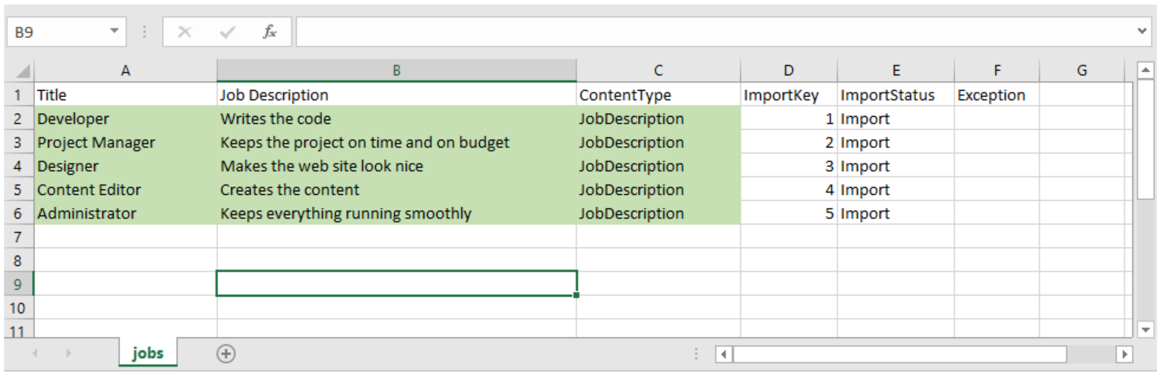
Import list items from an Excel spreadsheet source into a SharePoint Online (Office 365) list.

4.2.2 The Source

The source is an XLSX

Columns define the title, description, content type and control the import.

The worksheet is called "Jobs".

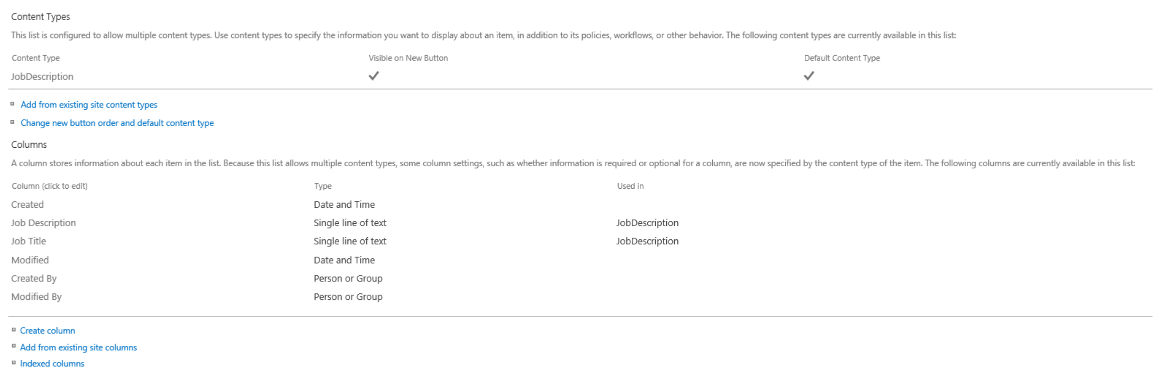


	A	B	C	D	E	F	G
1	Title	Job Description	ContentType	ImportKey	ImportStatus	Exception	
2	Developer	Writes the code	JobDescription	1	Import		
3	Project Manager	Keeps the project on time and on budget	JobDescription	2	Import		
4	Designer	Makes the web site look nice	JobDescription	3	Import		
5	Content Editor	Creates the content	JobDescription	4	Import		
6	Administrator	Keeps everything running smoothly	JobDescription	5	Import		
7							
8							
9							
10							
11							

4.2.3 The Configuration

```xml
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <!-- The source used is OLEDbSelect meaning that DIFS expected to run a SQL select statement against an OleDB data
source.  OleDB select will ignore columns like importstatus-->
    <SourceDataSetType>OLEDbSelect</SourceDataSetType>
    <!-- The Connection string uses a microsoft provider to access on Excel spreadsheet-->
    <OleDbSourceDataSetSettings>
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-host\Shared
Folders\Projects\ProductDevelopment\ImportForSharePoint\WorkingArea\Examples\JobDescriptions.xlsx;Extended Properties="Excel
12.0 Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbTableSourceDataSetSettings />
    <!-- The select statement will get all the jobs from the jobs worksheet -->
    <OleDbSelectSourceDataSetSettings>
      <SelectStatement>select * from [jobs$]</SelectStatement>
    </OleDbSelectSourceDataSetSettings>
  </Source>
  <Destination>
    <!-- The authentication type is Office365 i.e. online cloud sharepoint, you can load the configuration file to DIFS to
enter and save the credentials in encrypted format -->
    <AuthenticationSettings>
      <AuthenticationType>Office365</AuthenticationType>
      <domain />
      <username></username>
      <encryptedpassed></encryptedpassed>
    </AuthenticationSettings>
    <DestinationItemSettings>
      <!-- The destination you are tell DIFS to make is an item as opposed to a File or Folder -->
      <DestinationItemType>Item</DestinationItemType>
      <!-- If the item already exists then overwrite it -->
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
      <!-- Map data from your source into SharePoint fields-->
      <ImportMappings>
        <!-- Map title to title assuming that it is a string -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Title</DestinationField>
          <SourceColumn>Title</SourceColumn>
        </ImportMapping>
        <!-- Map job description to job description assuming that it is a string -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Job Description</DestinationField>
          <SourceColumn>Job Description</SourceColumn>
        </ImportMapping>
      </ImportMappings>
    </DestinationItemSettings>
    <!-- Tell DIFS exactly where the list is -->
    <DestinationListSettings>
      <DestinationWebUrlRelative>/sites/SPImportHelper</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/sites/SPImportHelper/Lists/Items</DestinationFolderUrlRelative>
      <DestinationServerUrl>https://company.sharepoint.com</DestinationServerUrl>
      <DestinationListName>Items</DestinationListName>
    </DestinationListSettings>
    <!-- Tell DIFS about the source you importing from-->
    <SourceColumns>
      <!-- The column in the source which contains the full path of the file being imported.  The value entered here is
ignored unless DestinationItemType is Document -->
      <SourceFileNameAndPath>FullName</SourceFileNameAndPath>
      <!-- The column in the source the value of which matches the content type in sharepoint to set on the item.  If you
are not using content types on the destination list you can enter an OOTB SharePoint content type such as Item, Document,
Folder -->
      <ContentType>ContentType</ContentType>
      <!-- The column in the source (if application) that contains the subfolder path to import to -->
      <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
      <!-- The column in the source which contains the destination file name. The value entered here is ignored unless
DestinationItemType is Document -->
      <DestinationFileName>DestinationFileName</DestinationFileName>
    </SourceColumns>
  </Destination>
</DataSetImportSettings>
```

4.2.4 The Destination



Content Types

This list is configured to allow multiple content types. Use content types to specify the information you want to display about an item, in addition to its policies, workflows, or other behavior. The following content types are currently available in this list:

Content Type	Visible on New Button	Default Content Type
JobDescription	✓	✓

- Add from existing site content types
- Change new button order and default content type

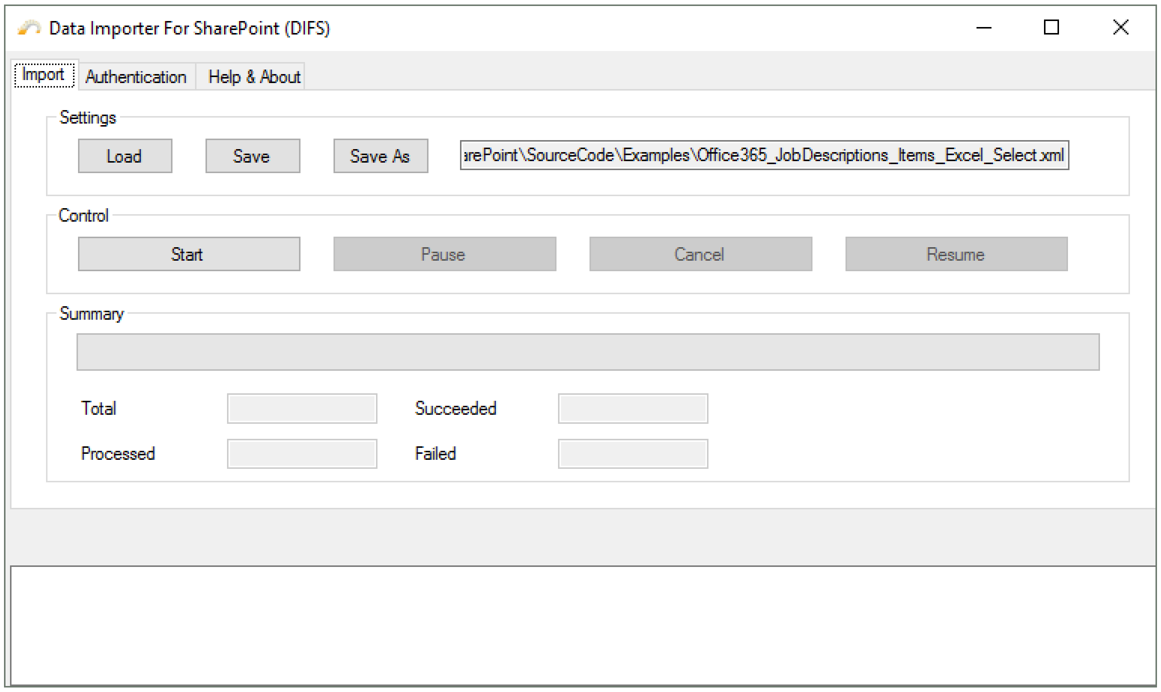
Columns

A column stores information about each item in the list. Because this list allows multiple content types, some column settings, such as whether information is required or optional for a column, are now specified by the content type of the item. The following columns are currently available in this list:

Column (click to edit)	Type	Used in
Created	Date and Time	
Job Description	Single line of text	JobDescription
Job Title	Single line of text	JobDescription
Modified	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

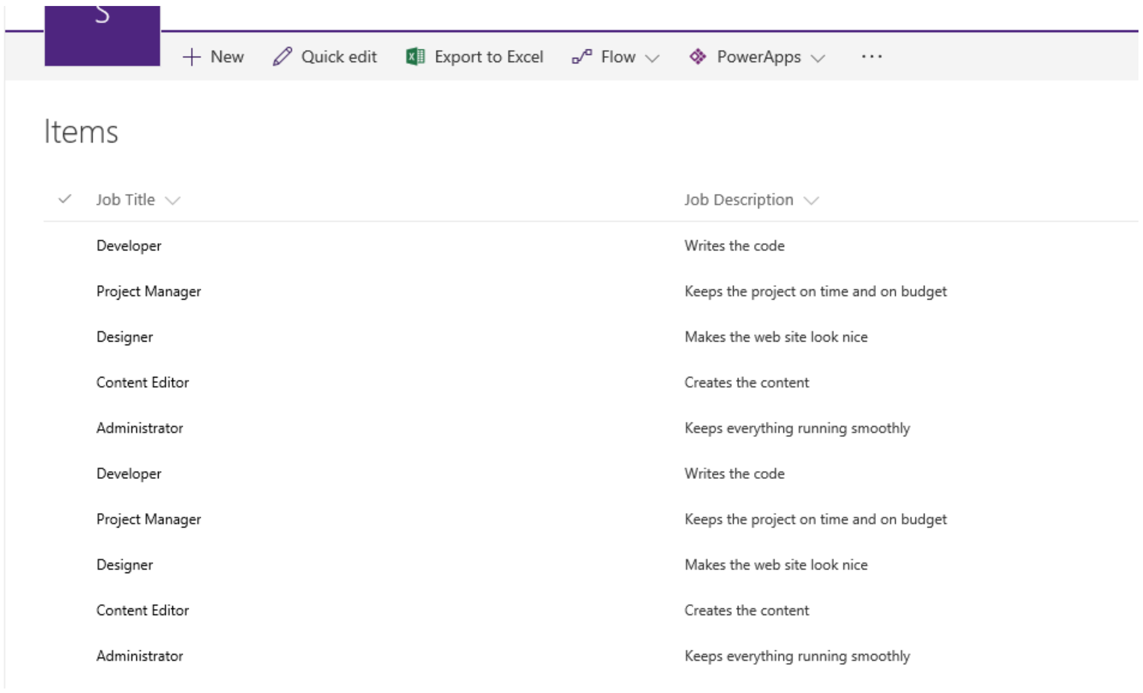
- Create column
- Add from existing site columns
- Indexed columns

4.2.5 The Execution



arePoint\SourceCode\Examples\Office365_JobDescriptions_Items_Excel_Select.xml

4.2.6 The Result



Items

Job Title	Job Description
Developer	Writes the code
Project Manager	Keeps the project on time and on budget
Designer	Makes the web site look nice
Content Editor	Creates the content
Administrator	Keeps everything running smoothly
Developer	Writes the code
Project Manager	Keeps the project on time and on budget
Designer	Makes the web site look nice
Content Editor	Creates the content
Administrator	Keeps everything running smoothly

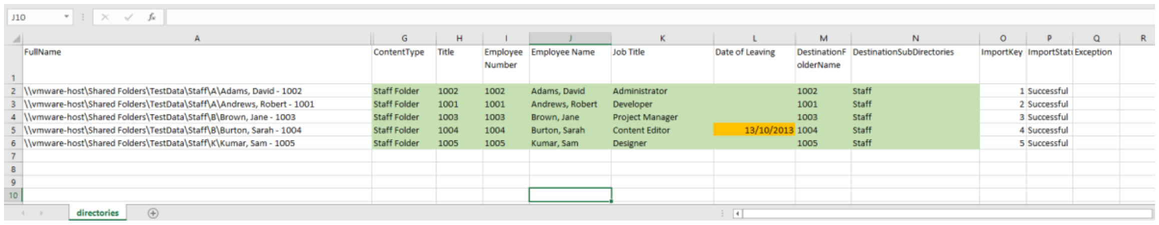
4.3 Create Folders from an Excel Source in a SharePoint Library adding meta data

4.3.1 Overview

Create a folder structure suitable for staff records.

Set meta title on the created folders including lookup fields and managed meta data.

4.3.2 The Source



J10

	A	G	H	I	J	K	L	M	N	O	P	Q	R
1	FullName	ContentType	Title	Employee Number	Employee Name	Job Title	Date of Leaving	DestinationFolderName	DestinationSubDirectories	ImportKey	ImportStatı	Exception	
2	\\vmware-host\Shared Folders\TestData\Staff\A\Adams, David - 1002	Staff Folder	1002	1002	Adams, David	Administrator		1002	Staff	1	Successful		
3	\\vmware-host\Shared Folders\TestData\Staff\A\Andrews, Robert - 1001	Staff Folder	1001	1001	Andrews, Robert	Developer		1001	Staff	2	Successful		
4	\\vmware-host\Shared Folders\TestData\Staff\B\Brown, Jane - 1003	Staff Folder	1003	1003	Brown, Jane	Project Manager		1003	Staff	3	Successful		
5	\\vmware-host\Shared Folders\TestData\Staff\B\Burton, Sarah - 1004	Staff Folder	1004	1004	Burton, Sarah	Content Editor	13/10/2013	1004	Staff	4	Successful		
6	\\vmware-host\Shared Folders\TestData\Staff\K\Kumar, Sam - 1005	Staff Folder	1005	1005	Kumar, Sam	Designer		1005	Staff	5	Successful		
7													
8													
9													
10													

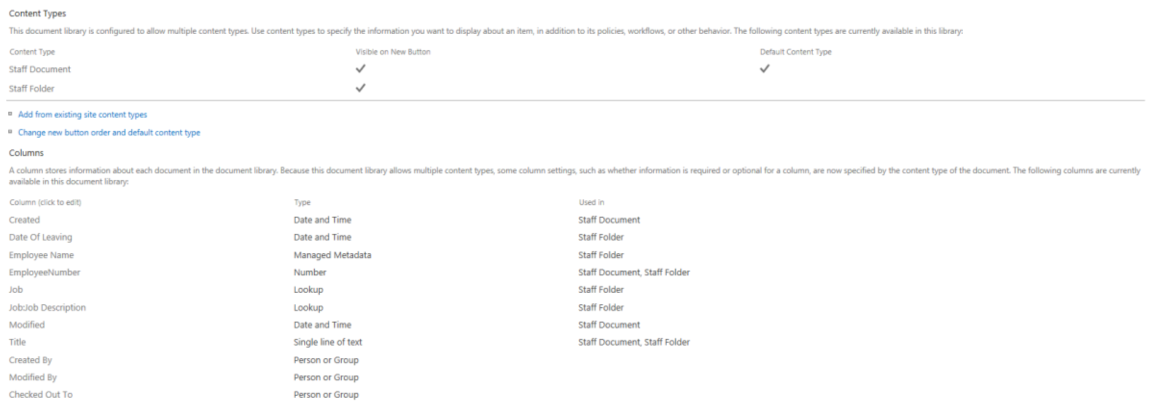
4.3.3 The Configuration

```
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <!-- The source used is OLEDbSelect meaning that DIFS expected to run a SQL select statement against an OleDB data
source.  OleDB select will ignore columns like importstatus-->
    <SourceDataSetType>OLEDbSelect</SourceDataSetType>
    <OleDbSourceDataSetSettings>
      <!-- The Connection string uses a microsoft provider to access on Excel spreadsheet-->
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-host\Shared
Folders\Projects\ProductDevelopment\ImportForSharePoint\WorkingArea\Examples\StaffFolders.xlsx;Extended Properties="Excel
12.0 Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbSelectSourceDataSetSettings>
      <!-- The select statement will get all the directories from the directories worksheet -->
      <SelectStatement>select * from [directories$]</SelectStatement>
    </OleDbSelectSourceDataSetSettings>
  </Source>
  <Destination>
    <!-- The authentication type is Office365 i.e. online cloud sharepoint, you can load the configuration file to DIFS to
enter and save the credentials in encrypted format -->
    <AuthenticationSettings>
      <AuthenticationType>Office365</AuthenticationType>
      <domain />
      <username></username>
      <encryptedpassed></encryptedpassed>
    </AuthenticationSettings>
    <DestinationItemSettings>
      <!-- The DestinationItemType you are telling DIFS to make is a Folder as opposed to an Item or Document -->
      <DestinationItemType>Folder</DestinationItemType>
      <!-- If the item already exists then overwrite it -->
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
      <!-- Map data from your source into SharePoint fields-->
      <ImportMappings>
```

```
        <!-- Map title to title assuming that it is a string -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Title</DestinationField>
          <SourceColumn>Title</SourceColumn>
        </ImportMapping>
        <!-- Map Employee Number to EmployeeNumber without conversion.  In this instance this works because both are numeric
fields-->
        <ImportMapping xsi:type="ImportMapping_Native">
          <DestinationField>EmployeeNumber</DestinationField>
          <SourceColumn>Employee Number</SourceColumn>
        </ImportMapping>
        <!-- Map Employee Name to Employee Name as ManagedMetaDataAutoAdd.  In this instance the value in the spreadsheet is
created as a managed meta data term and then the folder tagged with it -->
        <ImportMapping xsi:type="ImportMapping_ManagedMetaDataAutoAdd">
          <DestinationField>Employee Name</DestinationField>
          <SourceColumn>Employee Name</SourceColumn>
        </ImportMapping>
        <!-- Map Date of Leaving to DateOfLeaving converting from a string date.  In this instance the string is assumed to
be in UK date format-->
        <ImportMapping xsi:type="ImportMapping_DateTimeFromString">
          <DestinationField>DateOfLeaving</DestinationField>
          <SourceColumn>Date of Leaving</SourceColumn>
                    <!-- Refer to DateTime.ParseExact on MSDN for more information on ConversionMask (format) and culture
(iformat provider)-->
                    <ConversionMask>dd/MM/yyyy hh:mm:ss</ConversionMask>
                    <Culture>en-GB</Culture>
        </ImportMapping>
        <!-- Map Job Title to Job assuming that job is a look up field and the value of the column job title matches an
entry in it.-->
        <ImportMapping xsi:type="ImportMapping_Lookup">
                    <!-- This is the field of type lookup.-->
          <DestinationField>Job</DestinationField>
          <SourceColumn>Job Title</SourceColumn>
                    <!-- This is the list that contains the lookup values -->
                    <LookupListTitle>Items</LookupListTitle>
                    <!-- This is the field in the lookup list the value of which matches the value in the source column-->
                    <LookupFieldInternalName>Title</LookupFieldInternalName>
                    <!-- This is the CAML value type Text,Number,DateTime,Guid,MultiChoice,Lookup for the field on in the
lookup list that matches the value specified in the source column -->
                    <LookupFieldCAMLType>Text</LookupFieldCAMLType>
        </ImportMapping>
      </ImportMappings>
    </DestinationItemSettings>
    <!-- Tell DIFS exactly where the destination list or library is -->
    <DestinationListSettings>
      <DestinationWebUrlRelative>/sites/SPImportHelper</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/sites/SPImportHelper/Docs</DestinationFolderUrlRelative>
      <DestinationServerUrl>https://company.sharepoint.com</DestinationServerUrl>
      <DestinationListName>Docs</DestinationListName>
    </DestinationListSettings>
    <!-- Tell DIFS about the source you importing from-->
    <SourceColumns>
      <!-- The column in the source which contains the full path of the file being imported.  The value entered here is
ignored unless DestinationItemType is Document -->
      <SourceFileNameAndPath>FullName</SourceFileNameAndPath>
      <!-- The column in the source the value of which matches the content type in sharepoint to set on the item.  If you
are not using content types on the destination list you can enter an OOTB SharePoint content type such as Item, Document,
Folder -->
      <ContentType>ContentType</ContentType>
      <!-- The column in the source (if applicable) that contains the subfolder path (If any) to import to -->
      <!-- This could be, for example, "Staff" or "Staff/Managers" or "Staff/Managers/Retired".  The folder must already
exist and can be created using DIFS -->
      <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
      <!-- The column in the source which contains the destination name. The value entered here is ignored unless
DestinationItemType is Document or Folder -->
      <DestinationFileName>DestinationFolderName</DestinationFileName>
    </SourceColumns>
  </Destination>
</DataSetImportSettings>
```

4.3.4 The Destination

4.3.4.1 Library



Content Types

This document library is configured to allow multiple content types. Use content types to specify the information you want to display about an item, in addition to its policies, workflows, or other behavior. The following content types are currently available in this library:

Content Type	Visible on New Button	Default Content Type
Staff Document	✓	✓
Staff Folder	✓	

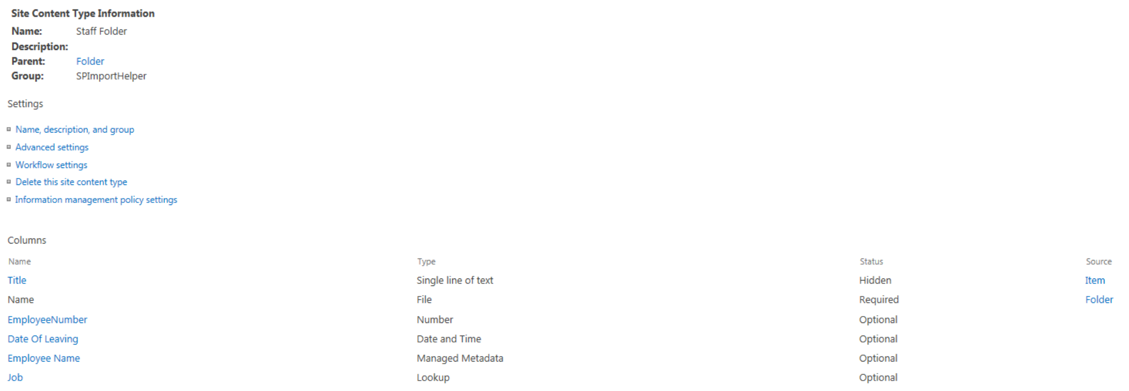
- Add from existing site content types
- Change new button order and default content type

Columns

A column stores information about each document in the document library. Because this document library allows multiple content types, some column settings, such as whether information is required or optional for a column, are now specified by the content type of the document. The following columns are currently available in this document library:

Column (click to edit)	Type	Used in
Created	Date and Time	Staff Document
Date Of Leaving	Date and Time	Staff Folder
Employee Name	Managed Metadata	Staff Folder
EmployeeNumber	Number	Staff Document, Staff Folder
Job	Lookup	Staff Folder
Job:Job Description	Lookup	Staff Folder
Modified	Date and Time	Staff Document
Title	Single line of text	Staff Document, Staff Folder
Created By	Person or Group	
Modified By	Person or Group	
Checked Out To	Person or Group	

4.3.4.2 Content Type



Site Content Type Information

Name:	Staff Folder
Description:	
Parent:	Folder
Group:	SPImportHelper

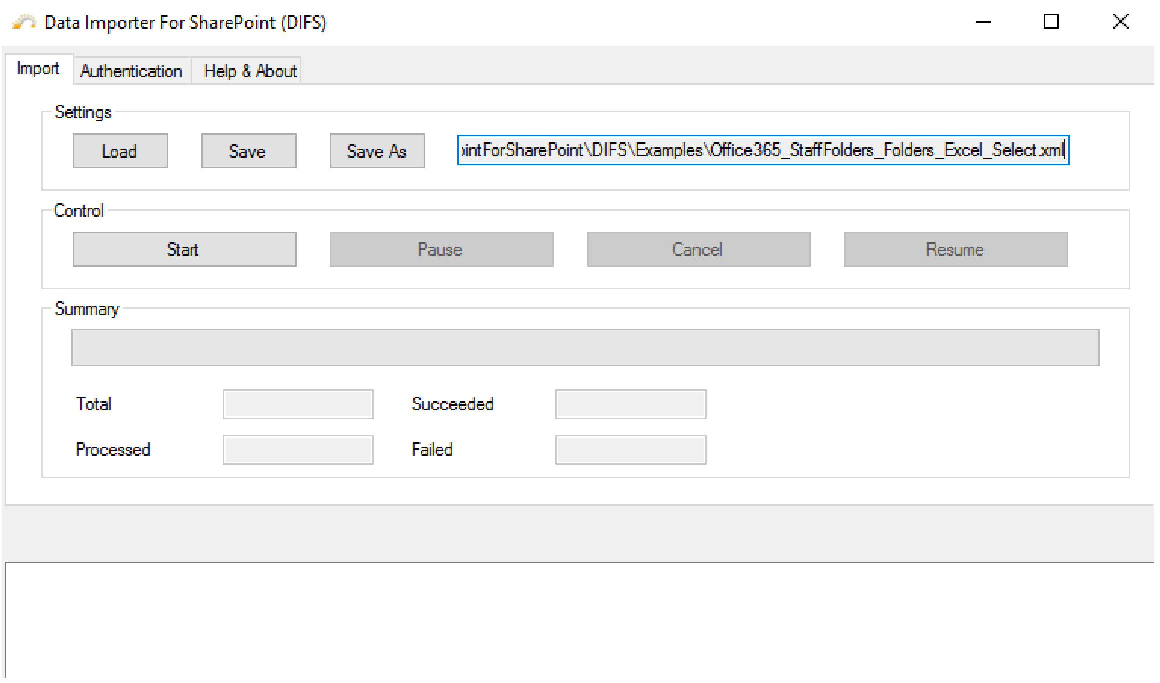
Settings

- Name, description, and group
- Advanced settings
- Workflow settings
- Delete this site content type
- Information management policy settings

Columns

Name	Type	Status	Source
Title	Single line of text	Hidden	Item
Name	File	Required	Folder
EmployeeNumber	Number	Optional	
Date Of Leaving	Date and Time	Optional	
Employee Name	Managed Metadata	Optional	
Job	Lookup	Optional	

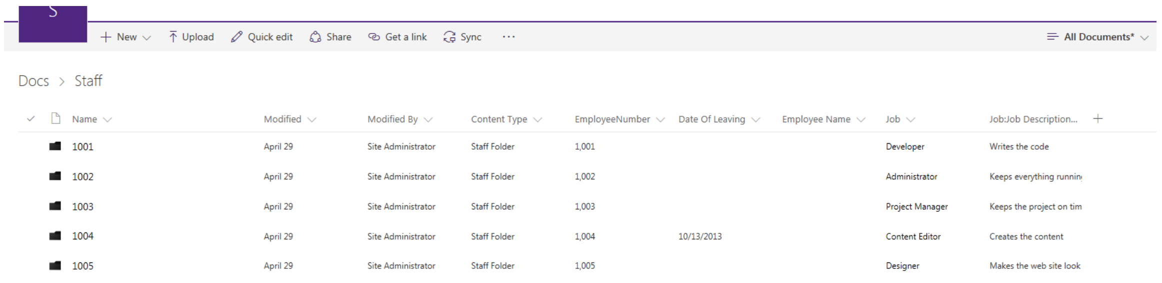
4.3.5 The Execution



ointForSharePoint\DIFS\Examples\Office365_StaffFolders_Folders_Excel_Select.xml

4.3.6 The Result

Folders have been created and meta data fields have been set.



Docs > Staff

Name	Modified	Modified By	Content Type	EmployeeNumber	Date Of Leaving	Employee Name	Job	Job:Job Description...
1001	April 29	Site Administrator	Staff Folder	1,001			Developer	Writes the code
1002	April 29	Site Administrator	Staff Folder	1,002			Administrator	Keeps everything runnin
1003	April 29	Site Administrator	Staff Folder	1,003			Project Manager	Keeps the project on tim
1004	April 29	Site Administrator	Staff Folder	1,004	10/13/2013		Content Editor	Creates the content
1005	April 29	Site Administrator	Staff Folder	1,005			Designer	Makes the web site look

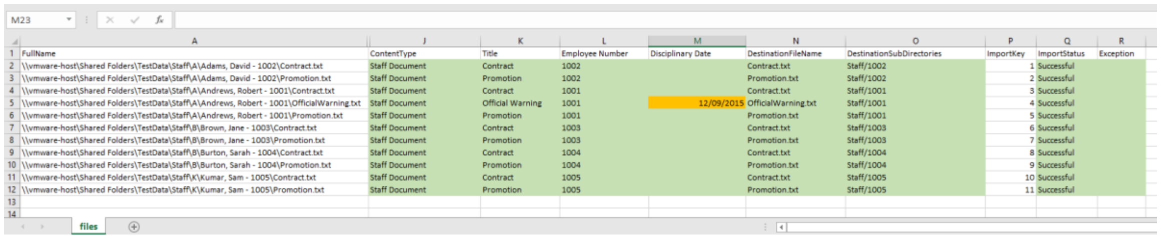
4.4 Import Documents from an Excel Source into a SharePoint Library

4.4.1 Overview

An Excel spreadsheet contains a list of documents, the path to each document and associated meta data.

DIFS imports the meta data from the spreadsheet and the associated document into a document library.

4.4.2 The Source



M23

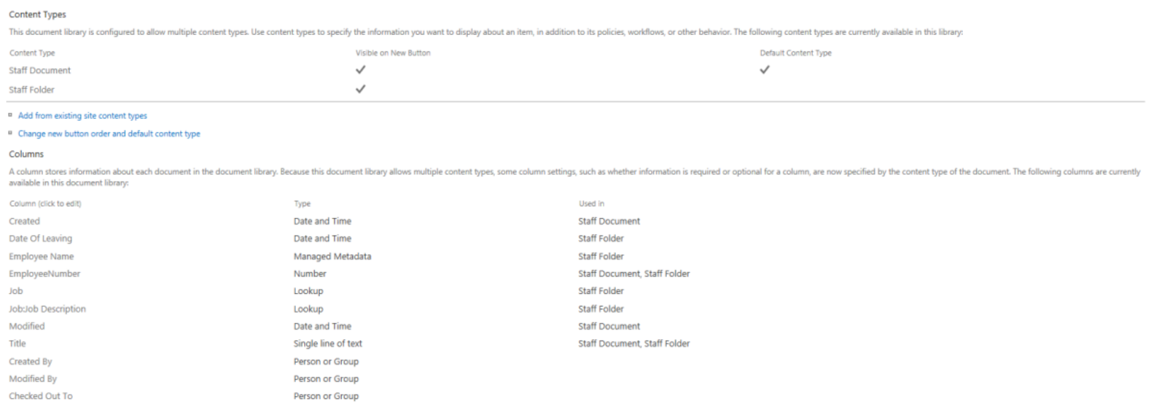
	A	J	K	L	M	N	O	P	Q	R
1	FullName	ContentType	Title	Employee Number	Disciplinary Date	DestinationFileName	DestinationSubDirectories	ImportKey	ImportStatus	Exception
2	\\vmware-host\Shared Folders\TestData\Staff\A\Adams, David - 1002\Contract.txt	Staff Document	Contract	1002		Contract.txt	Staff/1002	1	Successful	
3	\\vmware-host\Shared Folders\TestData\Staff\A\Adams, David - 1002\Promotion.txt	Staff Document	Promotion	1002		Promotion.txt	Staff/1002	2	Successful	
4	\\vmware-host\Shared Folders\TestData\Staff\A\Andrews, Robert - 1001\Contract.txt	Staff Document	Contract	1001		Contract.txt	Staff/1001	3	Successful	
5	\\vmware-host\Shared Folders\TestData\Staff\A\Andrews, Robert - 1001\OfficialWarning.txt	Staff Document	Official Warning	1001	12/09/2015	OfficialWarning.txt	Staff/1001	4	Successful	
6	\\vmware-host\Shared Folders\TestData\Staff\A\Andrews, Robert - 1001\Promotion.txt	Staff Document	Promotion	1001		Promotion.txt	Staff/1001	5	Successful	
7	\\vmware-host\Shared Folders\TestData\Staff\B\Brown, Jane - 1003\Contract.txt	Staff Document	Contract	1003		Contract.txt	Staff/1003	6	Successful	
8	\\vmware-host\Shared Folders\TestData\Staff\B\Brown, Jane - 1003\Promotion.txt	Staff Document	Promotion	1003		Promotion.txt	Staff/1003	7	Successful	
9	\\vmware-host\Shared Folders\TestData\Staff\B\Burton, Sarah - 1004\Contract.txt	Staff Document	Contract	1004		Contract.txt	Staff/1004	8	Successful	
10	\\vmware-host\Shared Folders\TestData\Staff\B\Burton, Sarah - 1004\Promotion.txt	Staff Document	Promotion	1004		Promotion.txt	Staff/1004	9	Successful	
11	\\vmware-host\Shared Folders\TestData\Staff\K\Kumar, Sam - 1005\Contract.txt	Staff Document	Contract	1005		Contract.txt	Staff/1005	10	Successful	
12	\\vmware-host\Shared Folders\TestData\Staff\K\Kumar, Sam - 1005\Promotion.txt	Staff Document	Promotion	1005		Promotion.txt	Staff/1005	11	Successful	
13										
14										

4.4.3 The Configuration

```
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <!-- The source used is OLEDbSelect meaning that DIFS expected to run a SQL select statement against an OleDB data
source.  OleDB select will ignore columns like importstatus-->
    <SourceDataSetType>OLEDbSelect</SourceDataSetType>
    <OleDbSourceDataSetSettings>
      <!-- The Connection string uses a microsoft provider to access an Excel spreadsheet-->
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-host\Shared
Folders\Projects\ProductDevelopment\ImportForSharePoint\WorkingArea\Examples\StaffDocuments.xlsx;Extended Properties="Excel
12.0 Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbSelectSourceDataSetSettings>
      <!-- The select statement will get all the documents from the files worksheet -->
      <SelectStatement>select * from [files$]</SelectStatement>
    </OleDbSelectSourceDataSetSettings>
  </Source>
  <Destination>
    <!-- The authentication type is Office365 i.e. online cloud sharepoint, you can load the configuration file to DIFS to
enter and save the credentials in encrypted format -->
    <AuthenticationSettings>
      <AuthenticationType>Office365</AuthenticationType>
      <domain />
      <username></username>
      <encryptedpassed></encryptedpassed>
    </AuthenticationSettings>
    <DestinationItemSettings>
      <!-- The DestinationItemType you are telling DIFS to make is a Document as opposed to an Item or Folder -->
      <DestinationItemType>Document</DestinationItemType>
      <!-- If the item already exists then overwrite it -->
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
      <!-- Map data from your source into SharePoint fields-->
      <ImportMappings>
```

```
        <!-- Map title to title assuming that it is a string -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Title</DestinationField>
          <SourceColumn>Title</SourceColumn>
        </ImportMapping>
        <!-- Map Employee Number to EmployeeNumber -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>EmployeeNumber</DestinationField>
          <SourceColumn>Employee Number</SourceColumn>
        </ImportMapping>
      </ImportMappings>
    </DestinationItemSettings>
    <!-- Tell DIFS exactly where the destination list or library is -->
    <DestinationListSettings>
      <DestinationWebUrlRelative>/sites/SPImportHelper</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/sites/SPImportHelper/Docs</DestinationFolderUrlRelative>
      <DestinationServerUrl>https://company.sharepoint.com</DestinationServerUrl>
      <DestinationListName>Docs</DestinationListName>
    </DestinationListSettings>
    <!-- Tell DIFS about the source you importing from-->
    <SourceColumns>
      <!-- The column in the source which contains the full path of the file being imported.  The value entered here is
ignored unless DestinationItemType is Document -->
      <SourceFileNameAndPath>FullName</SourceFileNameAndPath>
      <!-- The column in the source the value of which matches the content type in sharepoint to set on the item.  If you
are not using content types on the destination list you can enter an OOTB SharePoint content type such as Item, Document,
Folder -->
      <ContentType>ContentType</ContentType>
      <!-- The column in the source (if applicable) that contains the subfolder path (If any) to import to -->
      <!-- This could be, for example, "Staff" or "Staff/Bob Smith" or "Staff/1001".  The folder must already exist and can
be created using DIFS -->
      <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
      <!-- The column in the source which contains the destination name. The value entered here is ignored unless
DestinationItemType is Document or Folder -->
      <DestinationFileName>DestinationFileName</DestinationFileName>
    </SourceColumns>
  </Destination>
</DataSetImportSettings>
```

4.4.4 The Destination



Content Types

This document library is configured to allow multiple content types. Use content types to specify the information you want to display about an item, in addition to its policies, workflows, or other behavior. The following content types are currently available in this library:

Content Type	Visible on New Button	Default Content Type
Staff Document	✓	✓
Staff Folder	✓	

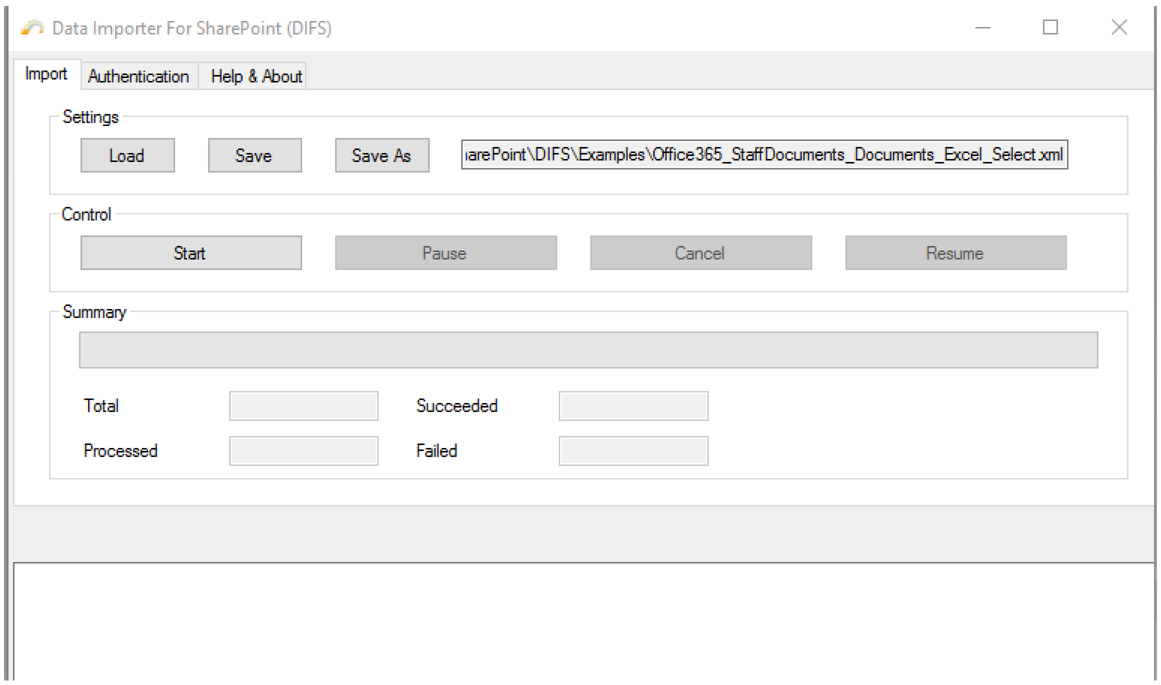
- Add from existing site content types
- Change new button order and default content type

Columns

A column stores information about each document in the document library. Because this document library allows multiple content types, some column settings, such as whether information is required or optional for a column, are now specified by the content type of the document. The following columns are currently available in this document library:

Column (click to edit)	Type	Used in
Created	Date and Time	Staff Document
Date Of Leaving	Date and Time	Staff Folder
Employee Name	Managed Metadata	Staff Folder
EmployeeNumber	Number	Staff Document, Staff Folder
Job	Lookup	Staff Folder
Job:Job Description	Lookup	Staff Folder
Modified	Date and Time	Staff Document
Title	Single line of text	Staff Document, Staff Folder
Created By	Person or Group	
Modified By	Person or Group	
Checked Out To	Person or Group	

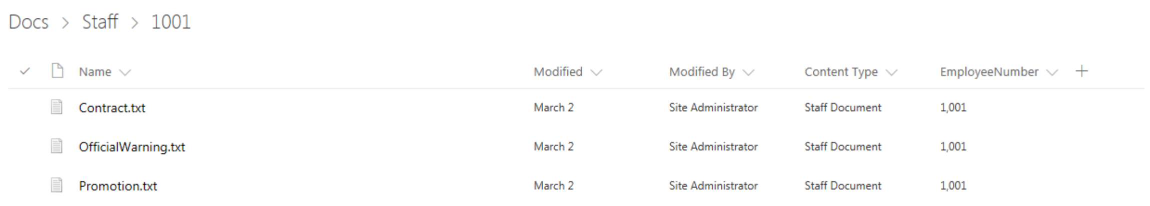
4.4.5 The Execution



iarePoint\DIFS\Examples\Office365_StaffDocuments_Documents_Excel_Select.xml

4.4.6 The Result

The staff documents are all in the correct folders.



Docs > Staff > 1001

Name	Modified	Modified By	Content Type	EmployeeNumber
Contract.txt	March 2	Site Administrator	Staff Document	1,001
OfficialWarning.txt	March 2	Site Administrator	Staff Document	1,001
Promotion.txt	March 2	Site Administrator	Staff Document	1,001

4.5 Create Document Sets from an Excel Source in SharePoint Library

4.5.1 Overview

Create a document set for each employee for their yearly review.

4.5.2 The Source

Content Type	Description	Title	Employee Number	Employee Name	DestinationFolder Name	DestinationSubDirectories
Staff Document Set	All documents for Adams, David Yearly Staff Appraisal	Adams, David Yearly Staff Appraisal	1002	Adams, David	1002	Appraisals
Staff Document Set	All documents for Andrews, Robert Yearly Staff Appraisal	Andrews, Robert Yearly Staff Appraisal	1001	Andrews, Robert	1001	Appraisals
Staff Document Set	All documents for Brown, Jane Yearly Staff Appraisal	Brown, Jane Yearly Staff Appraisal	1003	Brown, Jane	1003	Appraisals
Staff Document Set	All documents for Burton, Sarah Yearly Staff Appraisal	Burton, Sarah Yearly Staff Appraisal	1004	Burton, Sarah	1004	Appraisals
Staff Document Set	All documents for Kumar, Sam Yearly Staff Appraisal	Kumar, Sam Yearly Staff Appraisal	1005	Kumar, Sam	1005	Appraisals

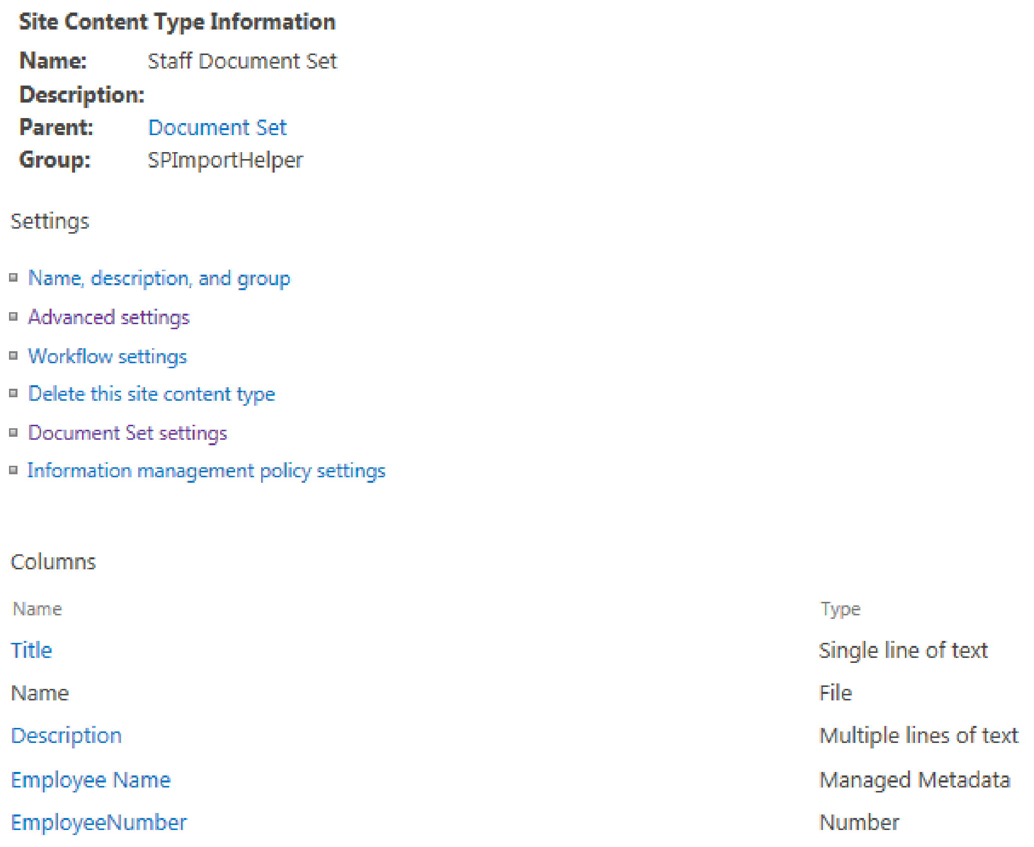
4.5.3 The Configuration

```xml
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <SourceDataSetType>OLEDbSelect</SourceDataSetType>
    <OleDbSourceDataSetSettings>
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-host\Shared
Folders\Projects\ProductDevelopment\ImportForSharePoint\WorkingArea\Examples\StaffDocumentSets.xlsx;Extended
Properties="Excel 12.0 Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbTableSourceDataSetSettings />
    <OleDbSelectSourceDataSetSettings>
      <SelectStatement>select * from [directories$]</SelectStatement>
    </OleDbSelectSourceDataSetSettings>
  </Source>
  <Destination>
    <AuthenticationSettings>
      <AuthenticationType>Office365</AuthenticationType>
      <domain />
      <username></username>
      <encryptedpassed></encryptedpassed>
    </AuthenticationSettings>
    <DestinationItemSettings>
      <!-- Document sets can be created in the same way as folders-->
      <DestinationItemType>Folder</DestinationItemType>
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
      <ImportMappings>
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Title</DestinationField>
          <SourceColumn>Title</SourceColumn>
        </ImportMapping>
        <!-- Each document set has a description field.  Since there are multiple description fields in SharePoint we will
use the internal name-->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>DocumentSetDescription</DestinationField>
          <SourceColumn>Description</SourceColumn>
        </ImportMapping>
        <ImportMapping xsi:type="ImportMapping_Native">
          <DestinationField>EmployeeNumber</DestinationField>
          <SourceColumn>Employee Number</SourceColumn>
        </ImportMapping>
        <ImportMapping xsi:type="ImportMapping_ManagedMetaDataCSOM">
          <DestinationField>Employee Name</DestinationField>
          <SourceColumn>Employee Name</SourceColumn>
        </ImportMapping>
      </ImportMappings>
    </DestinationItemSettings>
    <DestinationListSettings>
      <DestinationWebUrlRelative>/sites/SPImportHelper</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/sites/SPImportHelper/Docs</DestinationFolderUrlRelative>
      <DestinationServerUrl>https://company.sharepoint.com</DestinationServerUrl>
      <DestinationListName>Docs</DestinationListName>
    </DestinationListSettings>
    <SourceColumns>
      <SourceFileNameAndPath>FullName</SourceFileNameAndPath>
      <ContentType>ContentType</ContentType>
      <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
      <DestinationFileName>DestinationFolderName</DestinationFileName>
    </SourceColumns>
  </Destination>
</DataSetImportSettings>
```

4.5.4 The Destination

4.5.4.1 Content Type

A document set content type.



Site Content Type Information

Name:	Staff Document Set
Description:	
Parent:	Document Set
Group:	SPImportHelper

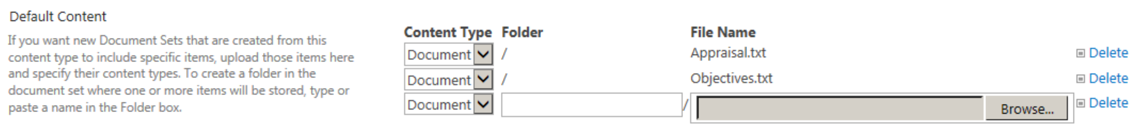
Settings

- Name, description, and group
- Advanced settings
- Workflow settings
- Delete this site content type
- Document Set settings
- Information management policy settings

Columns

Name	Type
Title	Single line of text
Name	File
Description	Multiple lines of text
Employee Name	Managed Metadata
EmployeeNumber	Number

Which deploys some standard documents.

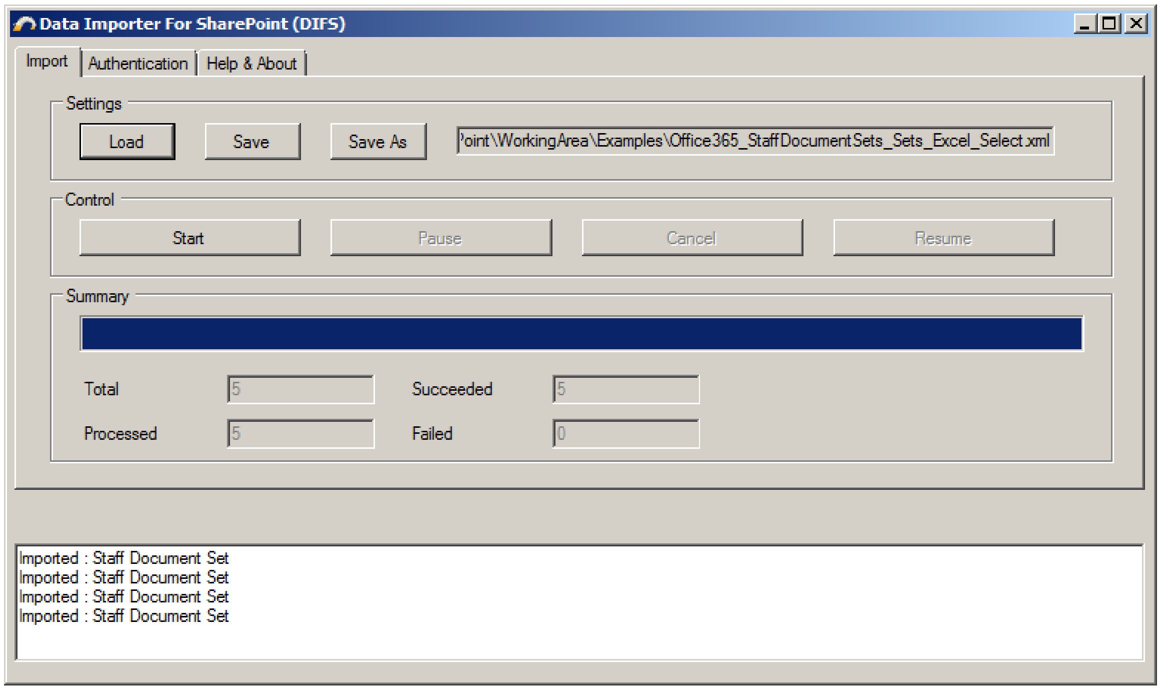


Default Content

If you want new Document Sets that are created from this content type to include specific items, upload those items here and specify their content types. To create a folder in the document set where one or more items will be stored, type or paste a name in the Folder box.

Content Type	Folder	File Name	
Document	/	Appraisal.txt	Delete
Document	/	Objectives.txt	Delete
Document	/		Delete

4.5.5 The Execution



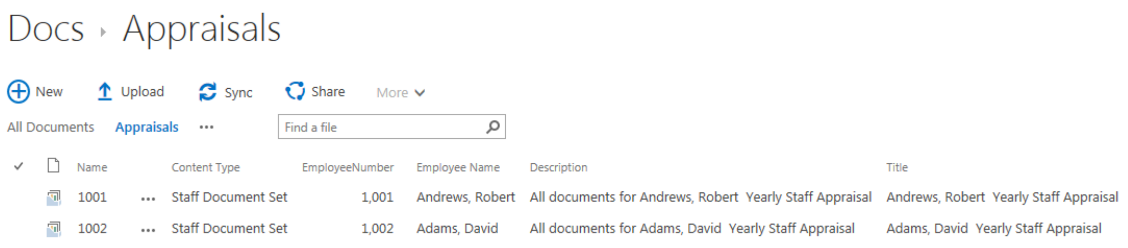
Point\WorkingArea\Examples\Office365_StaffDocumentSets_Sets_Excel_Select.xml

Total	5	Succeeded	5
Processed	5	Failed	0

```
Imported : Staff Document Set
Imported : Staff Document Set
Imported : Staff Document Set
Imported : Staff Document Set
```

4.5.6 The Result

The document sets are created.



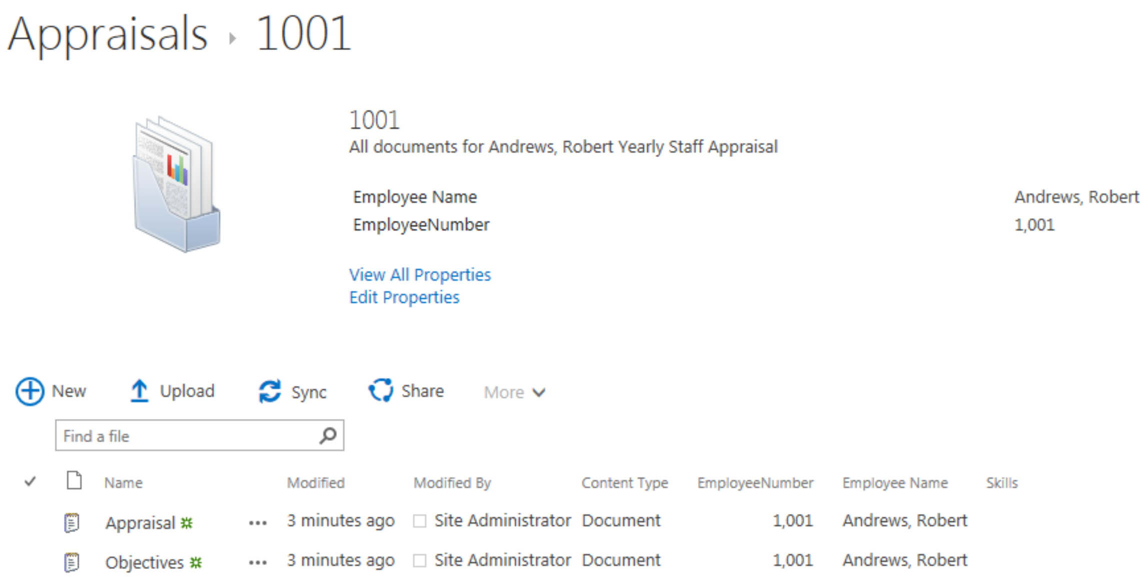
Docs › Appraisals

New Upload Sync Share More

All Documents Appraisals ··· Find a file

	Name		Content Type	EmployeeNumber	Employee Name	Description	Title
	1001	···	Staff Document Set	1,001	Andrews, Robert	All documents for Andrews, Robert Yearly Staff Appraisal	Andrews, Robert Yearly Staff Appraisal
	1002	···	Staff Document Set	1,002	Adams, David	All documents for Adams, David Yearly Staff Appraisal	Adams, David Yearly Staff Appraisal

Each document set auto-populates



Appraisals › 1001

1001
All documents for Andrews, Robert Yearly Staff Appraisal

Employee Name	Andrews, Robert
EmployeeNumber	1,001

View All Properties
Edit Properties

New Upload Sync Share More

Find a file

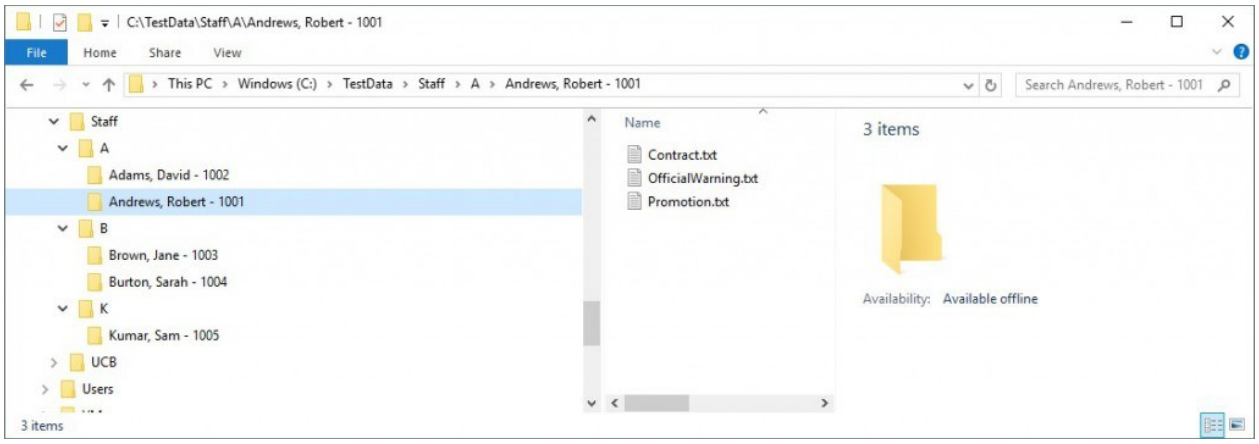
	Name		Modified	Modified By	Content Type	EmployeeNumber	Employee Name	Skills
	Appraisal	···	3 minutes ago	Site Administrator	Document	1,001	Andrews, Robert	
	Objectives	···	3 minutes ago	Site Administrator	Document	1,001	Andrews, Robert	

4.6 *File System Migration – Staff Records*

4.6.1 Scenario

Our theoretical scenario mirrors what is so often encountered when dealing with file shares.

We have a basic file structure. At the outer nodes the files are stored. The "meta data" is inferred by the location of the file in this structure as shown below.



C:\TestData\Staff\A\Andrews, Robert - 1001

This PC › Windows (C:) › TestData › Staff › A › Andrews, Robert - 1001

Search Andrews, Robert - 1001

- Staff
  - A
    - Adams, David - 1002
    - Andrews, Robert - 1001
  - B
    - Brown, Jane - 1003
    - Burton, Sarah - 1004
  - K
    - Kumar, Sam - 1005
- UCB
- Users

Contract.txt
OfficialWarning.txt
Promotion.txt

The file share is the only data source. There is no Staff Database. If there was then this might be handled differently with that database providing some of the data.

The example has you have now seen is for some Staff Records. Such records need to be stored in a compliant manner and retention scheduling is key to ensure that we retain only the correct records for each staff member. The requirement in this post is vastly simplified in comparison to most Staff Record scenarios but it serves to illustrate the concepts very well.

4.6.2 Why Migrate

Be clear on why you are migrating the data into SharePoint before you start the migration process. You may need to design the migration process to ensure that the desired benefits are achieved.

In our scenario the key drivers are;

- Compliance – Specifically data retention scheduling.
- Efficiency – Consolidation into SharePoint.
- Efficiency – Ease of use.
- Efficiency – Process automation.

4.6.3 Design and Implement the Destination

Before you execute migration you need to have a destination to migrate into.

This is key for two reasons;

- Migrating a live file share which is being updated is harder to manage.
- Until you have designed and implemented the destination you won't necessarily know how to define the import sources, in this example the two Excel spreadsheets.

For our scenario we have implemented;

- A record centre – /sites/Staff/Records.
- A record library – Records
- A top level folder "Staff"
- A content type for each staff folder "Staff Folder" which has a date of leaving field and retention action set from that date.
- A content type for each staff record "Staff Record" which has an employee number field.
- A content type for each disciplinary record "Staff Disciplinary Record" which has an Disciplinary Date and retention action set from that date because these records are kept for a shorter period of time.

It is sometimes useful to catalogue the file shares as part of this design process. This will give you an insight into the scenarios that your destination will need to cope with. The file shares can then be re-catalogued for migration at a later date.

4.6.4 Catalogue the File Share

To catalogue the file share we can use a PowerShell script.

You can run the script either from a PC as a user with access to the file share OR from the server hosting the file share. The script below will audit both the files in the file share and the directories, each to separate CSV files.

```
Function Audit-File($file)
{
   Write-Host "Auditing: " $file.FullName;
```

```
    $file | add-member -name "Owner" -membertype noteproperty -value (get-acl $_.fullname).owner;
    $file | Add-Member -Name "Action" -MemberType NoteProperty -Value "Copy";

    return $file
}

Function Audit-Files($source, $destination)
{

    Get-ChildItem -Recurse $source |  ?{-not $_.PSIsContainer} | ForEach-Object {Audit-File $_} | Sort-Object fullname | Select
FullName,CreationTime,LastWriteTime,Length,Owner,BaseName,Name,Extension,Action,ContentType,Title,Meta1,Meta2,De
stinationFileName,DestinationWebUrl,DestinationLibraryName,DestinationSubDirectories | Export-Csv -Force -
NoTypeInformation $destination


}

Function Audit-Directory($folder)
{
    Write-Host "Auditing: " $folder.FullName;

    $folder | add-member -name "Owner" -membertype noteproperty -value (get-acl $_.fullname).owner;
    $folder | Add-Member -Name "Files" -MemberType NoteProperty -Value ($_.GetFiles().Count).ToString();
    $folder | Add-Member -Name "Directories" -MemberType NoteProperty -Value ($_.GetDirectories().Count).ToString();
    $folder | Add-Member -Name "Action" -MemberType NoteProperty -Value "Create";

    return $folder;
}

Function Audit-Directories($source, $destination)
{

    Get-ChildItem -Recurse $source |  ?{$_.PSIsContainer} | ForEach-Object {Audit-Directory $_} | Sort-Object fullname |
Select
FullName,CreationTime,LastWriteTime,Owner,Files,Directories,Action,ContentType,Title,Meta1,Meta2,DestinationFileName,
DestinationWebUrl,DestinationLibraryName,DestinationSubDirectories | Export-Csv -Force -NoTypeInformation
$destination


}

Audit-Files "\\vmware-host\Shared Folders\TestData\Staff\" "\\vmware-host\Shared Folders\TestData\files.csv"
Audit-Directories "\\vmware-host\Shared Folders\TestData\Staff\" \\vmware-host\Shared Folders\TestData\directories.csv
```

The CSV files will contain the basic information that is available from the file system.

Tip: Try and use UNC paths instead of mapped drive letters when cataloguing file shares.

If you are unsure how to execute PowerShell scripts then pop “how to execute a powershell script” into your favourite search engine.

4.6.5 Prepare the Excel Spreadsheets

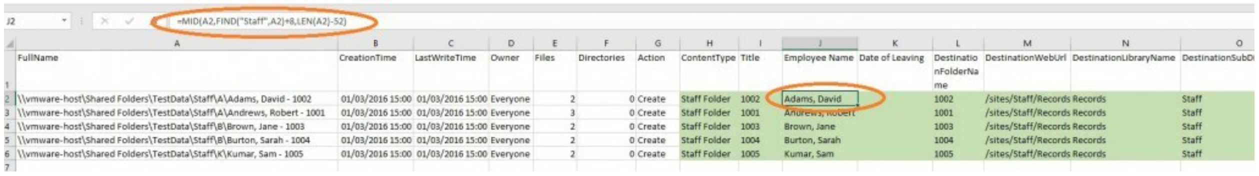
The CSV files can be imported to Excel and turned into a spreadsheet.

This will enable us to automate the population of meta data that is so important to the success of migration projects

4.6.5.1 Folders

In Excel we can easily populate some extra columns (shown here in green).

Here we are going to create an import source which will create a folder of content type "Staff Folder" for each staff member. This is going to have the Date of Leaving (which will drive retention schedules) and the staff name as meta data. The employee name and the employee number will be extracted from the folder name from the file system using an Excel formula.



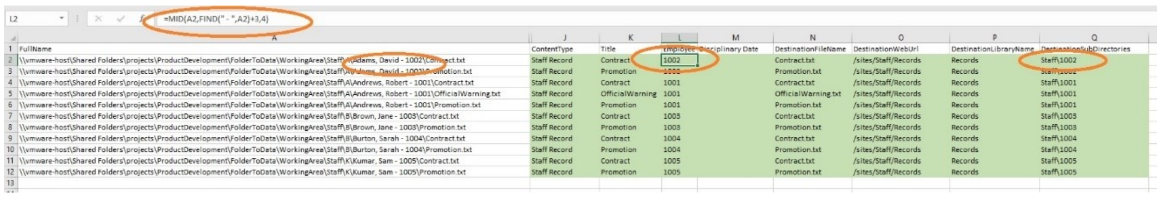
J2 =MID(A2,FIND("Staff",A2)+8,LEN(A2)-52)

FullName	CreationTime	LastWriteTime	Owner	Files	Directories	Action	ContentType	Title	Employee Name	Date of Leaving	DestinationFolderName	DestinationWebUrl	DestinationLibraryName	DestinationSubDi
\\vmware-host\Shared Folders\TestData\Staff\A\Adams, David - 1002	01/03/2016 15:00	01/03/2016 15:00	Everyone	2	0	Create	Staff Folder	1002	Adams, David		1002	/sites/Staff/Records	Records	Staff
\\vmware-host\Shared Folders\TestData\Staff\A\Andrews, Robert - 1001	01/03/2016 15:00	01/03/2016 15:00	Everyone	3	0	Create	Staff Folder	1001	Andrews, Robert		1001	/sites/Staff/Records	Records	Staff
\\vmware-host\Shared Folders\TestData\Staff\B\Brown, Jane - 1003	01/03/2016 15:00	01/03/2016 15:00	Everyone	2	0	Create	Staff Folder	1003	Brown, Jane		1003	/sites/Staff/Records	Records	Staff
\\vmware-host\Shared Folders\TestData\Staff\B\Burton, Sarah - 1004	01/03/2016 15:00	01/03/2016 15:00	Everyone	2	0	Create	Staff Folder	1004	Burton, Sarah		1004	/sites/Staff/Records	Records	Staff
\\vmware-host\Shared Folders\TestData\Staff\K\Kumar, Sam - 1005	01/03/2016 15:00	01/03/2016 15:00	Everyone	2	0	Create	Staff Folder	1005	Kumar, Sam		1005	/sites/Staff/Records	Records	Staff

4.6.5.2 Files

In Excel we can easily populate some extra columns shown in green.

Here we are going to create an import source which will import all of the files for the Staff. The employee number is extracted from the folder name which will tell us destination folder and the set some meta data against each document. This is extracted from the folder path.



L2 =MID(A2,FIND(" - ",A2)+3,4)

FullName	ContentType	Title	Employee	Disciplinary Date	DestinationFileName	DestinationWebUrl	DestinationLibraryName	DestinationSubDirectories
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\A\Adams, David - 1002\Contract.txt	Staff Record	Contract	1002		Contract.txt	/sites/Staff/Records	Records	Staff\1002
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\A\Adams, David - 1002\Promotion.txt	Staff Record	Promotion	1002		Promotion.txt	/sites/Staff/Records	Records	Staff\1002
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\A\Andrews, Robert - 1001\Contract.txt	Staff Record	Contract	1001		Contract.txt	/sites/Staff/Records	Records	Staff\1001
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\A\Andrews, Robert - 1001\OfficialWarning.txt	Staff Record	OfficialWarning	1001		OfficialWarning.txt	/sites/Staff/Records	Records	Staff\1001
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\A\Andrews, Robert - 1001\Promotion.txt	Staff Record	Promotion	1001		Promotion.txt	/sites/Staff/Records	Records	Staff\1001
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\B\Brown, Jane - 1003\Contract.txt	Staff Record	Contract	1003		Contract.txt	/sites/Staff/Records	Records	Staff\1003
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\B\Brown, Jane - 1003\Promotion.txt	Staff Record	Promotion	1003		Promotion.txt	/sites/Staff/Records	Records	Staff\1003
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\B\Burton, Sarah - 1004\Contract.txt	Staff Record	Contract	1004		Contract.txt	/sites/Staff/Records	Records	Staff\1004
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\B\Burton, Sarah - 1004\Promotion.txt	Staff Record	Promotion	1004		Promotion.txt	/sites/Staff/Records	Records	Staff\1004
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\K\Kumar, Sam - 1005\Contract.txt	Staff Record	Contract	1005		Contract.txt	/sites/Staff/Records	Records	Staff\1005
\\vmware-host\Shared Folders\projects\ProductDevelopment\FolderToData\WorkingArea\Staff\K\Kumar, Sam - 1005\Promotion.txt	Staff Record	Promotion	1005		Promotion.txt	/sites/Staff/Records	Records	Staff\1005

4.6.6 Get User Input

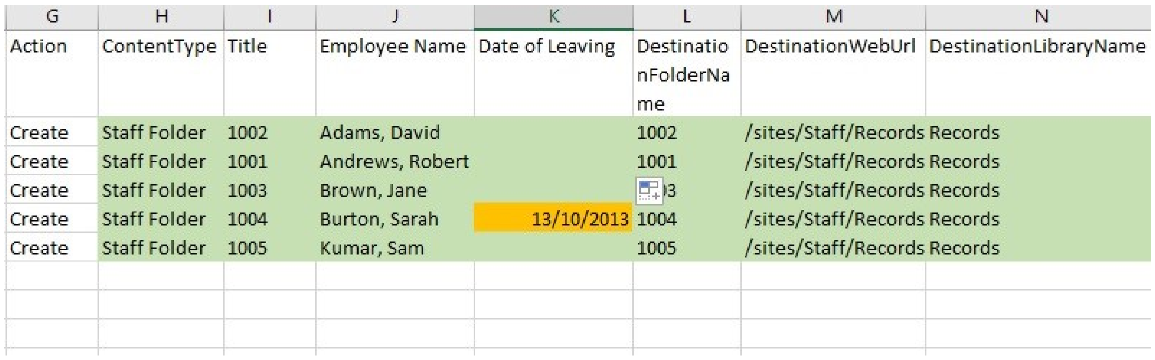
One of the strengths of Excel is that most users will have skills in using it.

What this enables us to do is use it to capture from the user base any additional data and permit the users to generally cleanse the data.

Tip: Give the users some guidance notes on how the Excel spreadsheets should be completed.

4.6.6.1 Folders

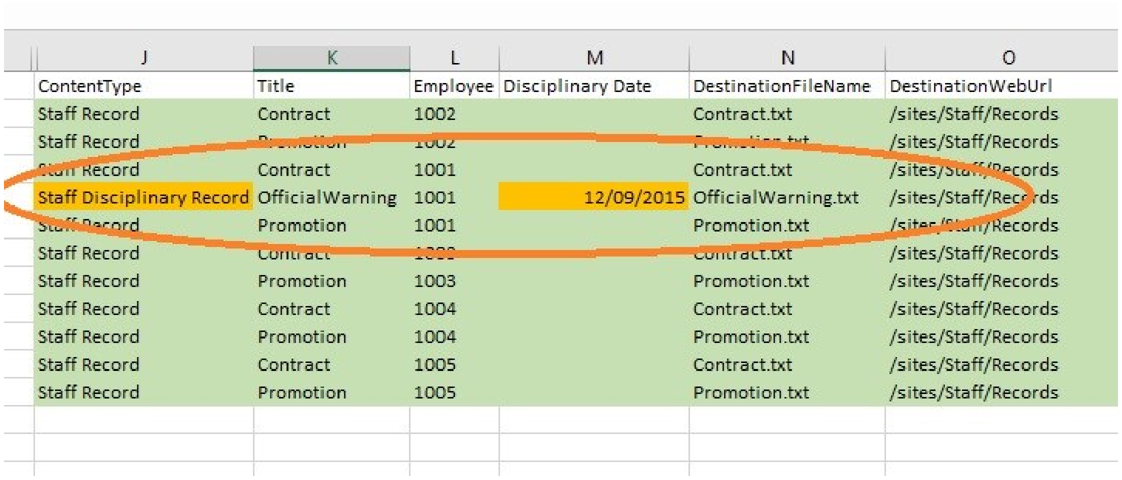
Here the user has completed the date of leaving field which in turn will allow SharePoint to manage the retention schedule accordingly.



G	H	I	J	K	L	M	N
Action	ContentType	Title	Employee Name	Date of Leaving	DestinationFolderName	DestinationWebUrl	DestinationLibraryName
Create	Staff Folder	1002	Adams, David		1002	/sites/Staff/Records	Records
Create	Staff Folder	1001	Andrews, Robert		1001	/sites/Staff/Records	Records
Create	Staff Folder	1003	Brown, Jane		[illegible]3	/sites/Staff/Records	Records
Create	Staff Folder	1004	Burton, Sarah	13/10/2013	1004	/sites/Staff/Records	Records
Create	Staff Folder	1005	Kumar, Sam		1005	/sites/Staff/Records	Records

4.6.6.2 Files

Here the user has spotted that one file is a disciplinary document. They have therefore changed the disciplinary date and content type accordingly.



J	K	L	M	N	O
ContentType	Title	Employee	Disciplinary Date	DestinationFileName	DestinationWebUrl
Staff Record	Contract	1002		Contract.txt	/sites/Staff/Records
Staff Record	Promotion	1002		Promotion.txt	/sites/Staff/Records
Staff Record	Contract	1001		Contract.txt	/sites/Staff/Records
Staff Disciplinary Record	OfficialWarning	1001	12/09/2015	OfficialWarning.txt	/sites/Staff/Records
Staff Record	Promotion	1001		Promotion.txt	/sites/Staff/Records
Staff Record	Contract	[illegible]		Contract.txt	/sites/Staff/Records
Staff Record	Promotion	1003		Promotion.txt	/sites/Staff/Records
Staff Record	Contract	1004		Contract.txt	/sites/Staff/Records
Staff Record	Promotion	1004		Promotion.txt	/sites/Staff/Records
Staff Record	Contract	1005		Contract.txt	/sites/Staff/Records
Staff Record	Promotion	1005		Promotion.txt	/sites/Staff/Records

4.6.7 Preparation Complete

Once preparation is complete you should have a set of Excel spreadsheets.

This should be double checked and quality controlled before you commence the migration process but the core work is done.

You can now create the folder structure in SharePoint - Create Folders from an Excel Source in a SharePoint Library adding meta data.

You can then import the documents into that folder structure - Import Documents from an Excel Source into a SharePoint Library,

4.7 Import from / Migrate Legacy Document Management Systems

4.7.1 The Source

4.7.1.1 Introduction

Most document management systems are built to the same basic architecture.

The meta data is stored in a database – typically SQL or Oracle.

The documents themselves are stored on file server(s).

With a bit of work to decode the schema it is frequently possible to perform a migration directly into SharePoint using DIFS.

This section works through an example for a document management system (DMS) with a very simple schema used for storing shipping contracts.

The same approach will work for more complex schemas as such as OpenText LiveLink.

Some DM systems require you to access the documents via an API. Talk to the DIFS developers in this instance for assistance.

4.7.1.2 Schema

The DMS has two tables “Documents” and “Versions”.

We can look at those tables.

```
use DMSystem
select top 10
ID,
TITLE,
field10,
field16
from Documents


select top 10
OBJECT_ID,
FilePath
from versions
```

ID	TITLE	field10	field16
1042	Contract Cost Agreement	Shipping	FALKLAND I
1043	Contract Cost Agreement	Shipping	ST HELENA
1045	Contract Cost Agreement	Shipping	ASCENSION
1047	Contract Cost Agreement	Shipping	BAHAMAS
1052	Contract Cost Agreement	Shipping	GHANA
1056	Contract Cost Agreement	Shipping	GAMBIA
1086	Contract Cost Agreement	Shipping	UGANDA
1088	Contract Cost Agreement	Shipping	UGANDA
1089	Contract Cost Agreement	Shipping	ZIMBABWE
1090	Contract Cost Agreement	Shipping	ZIMBABWE

OBJECT_ID	FilePath
124895	\\vmware-host\Shared

	Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
124896	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
124904	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
125003	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
125134	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
125135	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
125137	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
125138	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
125139	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf
125172	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf

Ok so this is an easy one to work out.

The ID relates the two tables.

The **Documents** table stores the meta data and the **Versions** table stores the pointers to the files.

4.7.1.3 Create an Import Source

Run a SQL command to create a single ImportSource table that contains all the information that DIFS needs to process the Import.

```
select TITLE,filename,field10,field16,FilePath,
'Contracts' as DestinationSubDirectories,
'Contract' as ContentType,
Versions.ID as ImportKey,
CAST('Import' as varchar(255)) as ImportStatus,
CAST(null as varchar(255)) as Exception
into ImportSource
from Documents,Versions
where documents.ID = versions.Object_ID
order by Versions.ID
```

We can have a look at our new table.

```
select top 10 * from ImportSource
```

TITLE	filename	field10	field16	FilePath	DestinationSubDirectories	ContentType	ImportKey	ImportStatus	Exception
Contract Cost Agreement	00000040.TIF	Shipping	FALKLAND I	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\Source Code\Examples\Test.pdf	Contracts	Contract	1033	Import	NULL
Contract Cost Agreement	00000041.TIF	Shipping	STHELENA	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\Source Code\Examples\Test.pdf	Contracts	Contract	1034	Import	NULL
Contract Cost Agreement	00000042.TIF	Shipping	ASCENSION	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\Source Code\Examples\Test.pdf	Contracts	Contract	1036	Import	NULL
Contract Cost Agreement	00000043.TIF	Shipping	BAHAMAS	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\Source Code\Examples\Test.pdf	Contracts	Contract	1038	Import	NULL
Contract Cost Agreement	0000003E.TIF	Shipping	GHANA	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\Source Code\Examples\Test.pdf	Contracts	Contract	1043	Import	NULL

Contract Cost Agreement	000000031.TIF	Shipping	GAMBIA	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf	Contracts	Contract	1046	Import	NULL
Contract Cost Agreement	000000052.TIF	Shipping	UGANDA	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf	Contracts	Contract	1073	Import	NULL
Contract Cost Agreement	000000053.TIF	Shipping	UGANDA	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf	Contracts	Contract	1075	Import	NULL
Contract Cost Agreement	000000054.TIF	Shipping	ZIMBABWE	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf	Contracts	Contract	1076	Import	NULL
Contract Cost Agreement	000000055.TIF	Shipping	ZIMBABWE	\\vmware-host\Shared Folders\Projects\ProductDevelopment\ImportForSharePoint\SourceCode\Examples\Test.pdf	Contracts	Contract	1077	Import	NULL

4.7.2 The Configuration

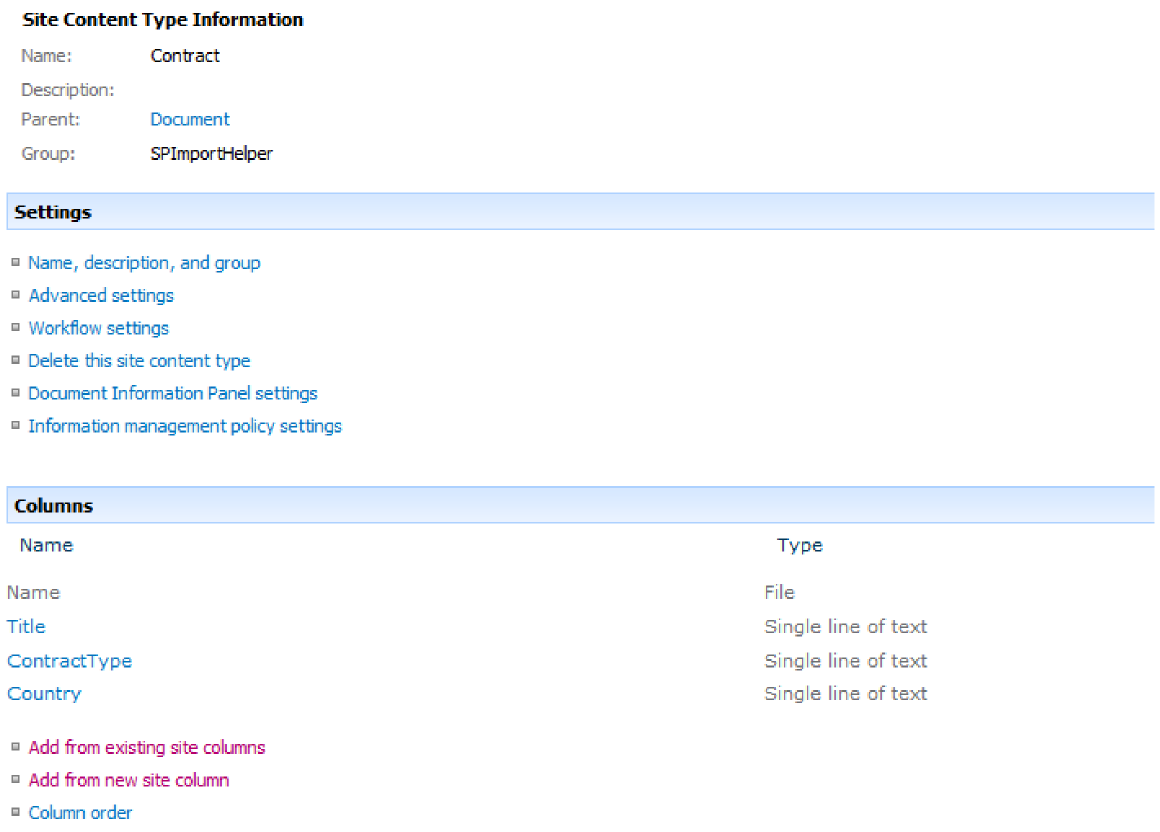
```
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <!-- The source used is OLEDbTable-->
    <SourceDataSetType>OLEDbTable</SourceDataSetType>
    <!-- The Connection string uses a SQL Server data set-->
    <OleDbSourceDataSetSettings>

<ConnectionString>Provider=SQLNCLI10;Server=127.0.0.1\SharePoint;Database=DMSystem;Trusted_Connection=yes;</ConnectionString
>
    </OleDbSourceDataSetSettings>
    <!-- We will use the ImportSource table-->
    <OleDbTableSourceDataSetSettings>
      <TableName>ImportSource</TableName>
    </OleDbTableSourceDataSetSettings>
  </Source>
  <Destination>
    <!-- The authentication type is current because we are using SharePoint Server On-Premises and running the import as a
user with sufficient permissions -->
    <AuthenticationSettings>
      <AuthenticationType>Current</AuthenticationType>
      <domain />
      <username></username>
      <encryptedpassed></encryptedpassed>
    </AuthenticationSettings>
    <DestinationItemSettings>
      <!-- The DestinationItemType you are telling DIFS to make is an item as opposed to a File or Folder -->
      <DestinationItemType>Document</DestinationItemType>
      <!-- If the item already exists then overwrite it -->
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
      <!-- Map data from your source into SharePoint fields-->
      <ImportMappings>
        <!-- Map title to title assuming that it is a string -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Title</DestinationField>
          <SourceColumn>Title</SourceColumn>
        </ImportMapping>
        <!-- Field10 is the contract type -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>ContractType</DestinationField>
          <SourceColumn>Field10</SourceColumn>
        </ImportMapping>
        <!-- Field16 is the country -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Country</DestinationField>
          <SourceColumn>Field16</SourceColumn>
        </ImportMapping>
      </ImportMappings>
    </DestinationItemSettings>
    <!-- Tell DIFS exactly where the destination list or library is -->
    <DestinationListSettings>
      <DestinationWebUrlRelative>/sites/SPImportHelper</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/sites/SPImportHelper/Docs</DestinationFolderUrlRelative>
      <DestinationServerUrl>http://productdev</DestinationServerUrl>
      <DestinationListName>Docs</DestinationListName>
    </DestinationListSettings>
    <!-- Tell DIFS about the source you importing from-->
    <SourceColumns>
      <!-- The column in the source which contains the full path of the file being imported.  The value entered here is 
ignored unless DestinationItemType is Document -->
      <SourceFileNameAndPath>filepath</SourceFileNameAndPath>
      <!-- The column in the source the value of which matches the content type in sharepoint to set on the item.  If you 
are not using content types on the destination list you can enter an OOTB SharePoint content type such as Item, Document, 
Folder -->
      <ContentType>ContentType</ContentType>
      <!-- The column in the source (if applicable) that contains the subfolder path (If any) to import to -->
      <!-- This could be, for example, "Staff" or "Staff/Managers" or "Staff/Managers/Retired".  The folder must already 
exist and can be created using DIFS -->
      <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
      <!-- The column in the source which contains the destination name. The value entered here is ignored unless 
DestinationItemType is Document or Folder -->
      <DestinationFileName>filename</DestinationFileName>
    </SourceColumns>
```

```
  </Destination>
</DataSetImportSettings>
```

4.7.4 The Destination

4.7.4.1 Content Type



Site Content Type Information

Name:	Contract
Description:	
Parent:	Document
Group:	SPImportHelper

Settings

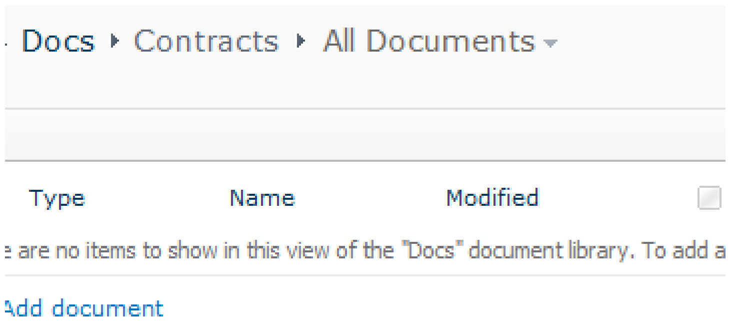
- Name, description, and group
- Advanced settings
- Workflow settings
- Delete this site content type
- Document Information Panel settings
- Information management policy settings

Columns

Name	Type
Name	File
Title	Single line of text
ContractType	Single line of text
Country	Single line of text

- Add from existing site columns
- Add from new site column
- Column order

4.7.4.2 Library / Folder



Docs ▸ Contracts ▸ All Documents

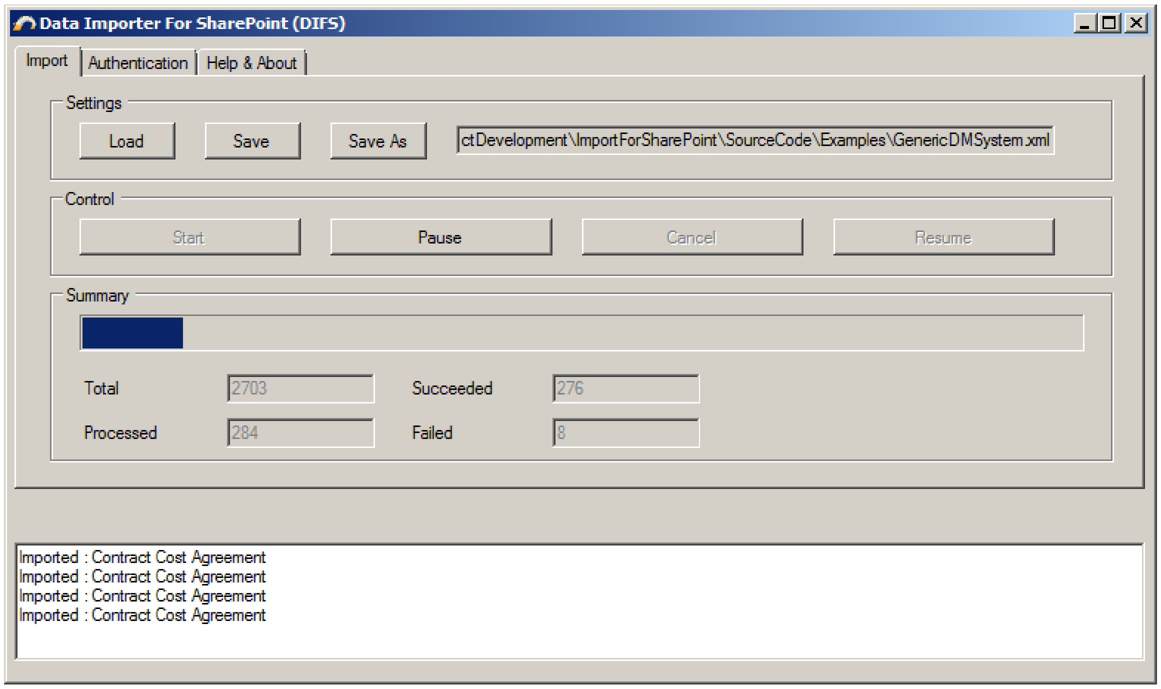
Type	Name	Modified

e are no items to show in this view of the "Docs" document library. To add a

Add document

4.7.5 The Execution

DIFS will show progress as files are imported.



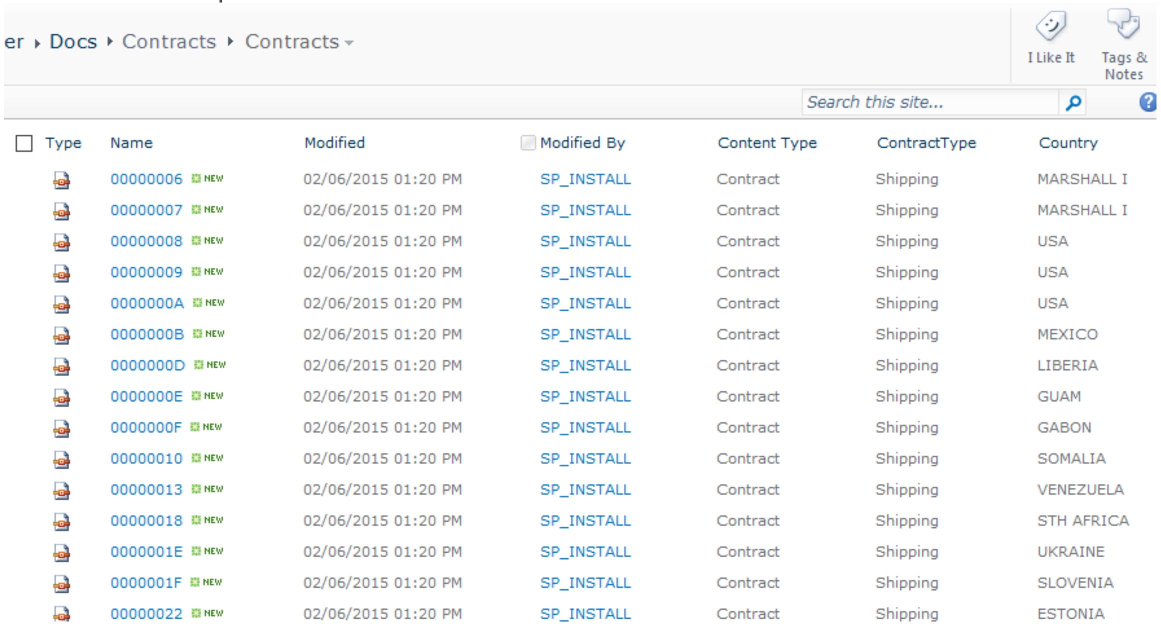
ctDevelopment\ImportForSharePoint\SourceCode\Examples\GenericDMSystem.xml

Total	2703	Succeeded	276
Processed	284	Failed	8

```
Imported : Contract Cost Agreement
Imported : Contract Cost Agreement
Imported : Contract Cost Agreement
Imported : Contract Cost Agreement
```

4.7.6 The Result

The files are imported into SharePoint / SharePoint online.



er ▸ Docs ▸ Contracts ▸ Contracts

Type	Name	Modified	Modified By	Content Type	ContractType	Country
	00000006 NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	MARSHALL I
	00000007 NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	MARSHALL I
	00000008 NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	USA
	00000009 NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	USA
	0000000A NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	USA
	0000000B NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	MEXICO
	0000000D NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	LIBERIA
	0000000E NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	GUAM
	0000000F NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	GABON
	00000010 NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	SOMALIA
	00000013 NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	VENEZUELA
	00000018 NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	STH AFRICA
	0000001E NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	UKRAINE
	0000001F NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	SLOVENIA
	00000022 NEW	02/06/2015 01:20 PM	SP_INSTALL	Contract	Shipping	ESTONIA

4.8 *Import Documents from an Excel Source into OneDrive for Business*

4.8.1 Overview

An Excel spreadsheet contains a list of documents, the path to each document and associated meta data.

DIFS imports the meta data from the spreadsheet and the associated document into OneDrive for Business.

4.8.2 The Source

FullName	ContentType	Title	DestinationFileName
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test1	Test1.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test2	Test2.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test3	Test3.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test4	Test4.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test5	Test5.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test6	Test6.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test7	Test7.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test8	Test8.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test9	Test9.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test10	Test10.pdf
\\vmware-host\Shared Folders\TestData\Generic\Test.pdf	Document	Test11	Test11.pdf

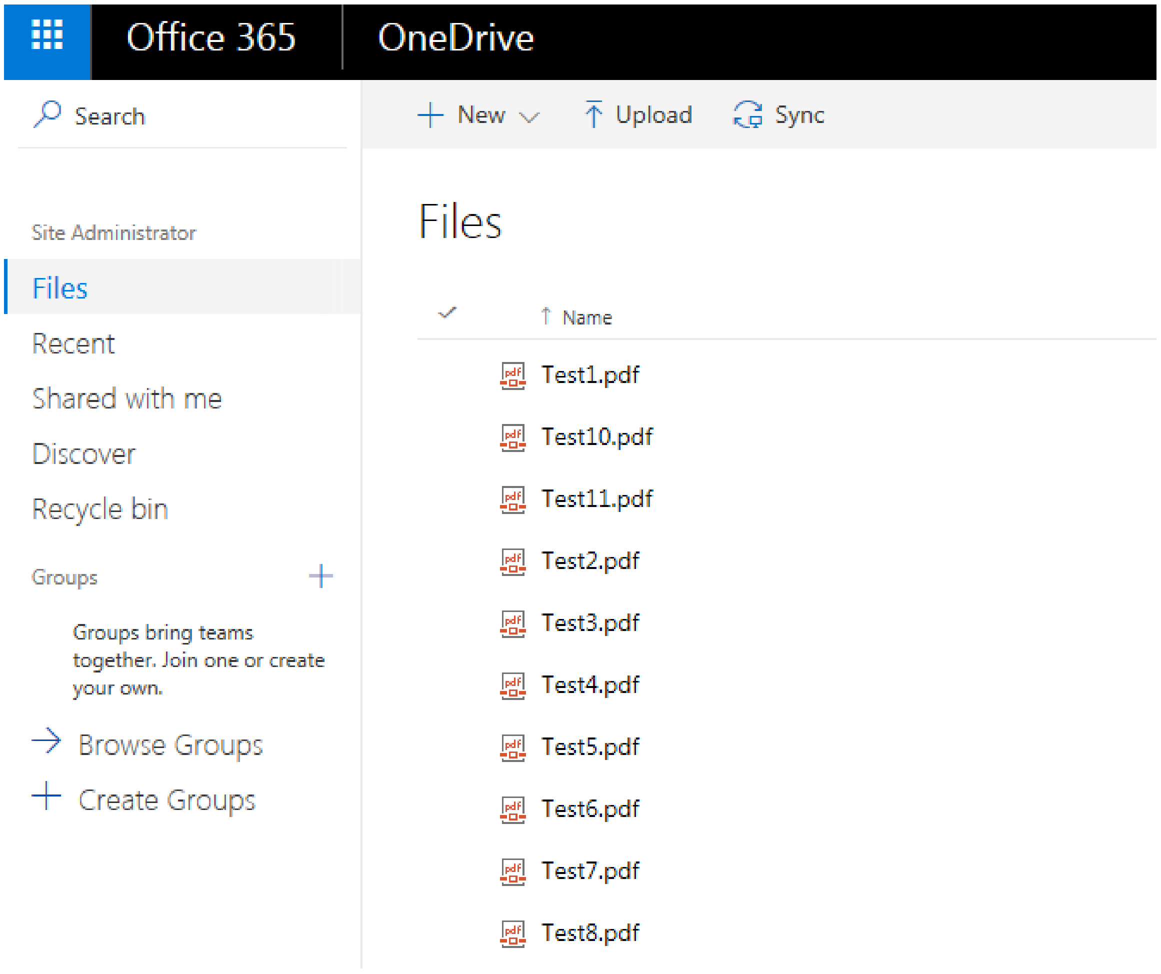
4.8.3 The Configuration

```
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <SourceDataSetType>OLEDbSelect</SourceDataSetType>
    <OleDbSourceDataSetSettings>
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-host\Shared 
Folders\Projects\ProductDevelopment\ImportForSharePoint\WorkingArea\Examples\OneDrive.xlsx;Extended Properties="Excel 12.0 
Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbTableSourceDataSetSettings />
    <OleDbSelectSourceDataSetSettings>
      <SelectStatement>select * from [files$]</SelectStatement>
    </OleDbSelectSourceDataSetSettings>
  </Source>
  <Destination>
    <AuthenticationSettings>
      <AuthenticationType>Office365</AuthenticationType>
      <domain />
      <username>me@company.onmicrosoft.com</username>
      <encryptedpassed> /encryptedpassed>
    </AuthenticationSettings>
    <DestinationItemSettings>
      <DestinationItemType>Document</DestinationItemType>
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
      <ImportMappings>
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Title</DestinationField>
          <SourceColumn>Title</SourceColumn>
        </ImportMapping>
      </ImportMappings>
    </DestinationItemSettings>
    <DestinationListSettings>
      <DestinationWebUrlRelative>/personal/me_company_onmicrosoft_com</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/personal/david_company_onmicrosoft_com/Documents</DestinationFolderUrlRelative>
      <DestinationServerUrl>https://company-my.sharepoint.com</DestinationServerUrl>
      <DestinationListName>Documents</DestinationListName>
    </DestinationListSettings>
    <SourceColumns>
      <SourceFileNameAndPath>FullName</SourceFileNameAndPath>
      <ContentType>ContentType</ContentType>
      <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
      <DestinationFileName>DestinationFileName</DestinationFileName>
      <Publish>Publish</Publish>
      <CheckInComment>CheckInComment</CheckInComment>
      <PublishComment>PublishComment</PublishComment>
    </SourceColumns>
  </Destination>
</DataSetImportSettings>
```

4.8.4 The Destination

User’s OneDrive for business “Files”.

4.8.5 The Result



Files

Groups bring teams together. Join one or create your own.

- Test1.pdf
- Test10.pdf
- Test11.pdf
- Test2.pdf
- Test3.pdf
- Test4.pdf
- Test5.pdf
- Test6.pdf
- Test7.pdf
- Test8.pdf

4.9 Import Publishing Pages into SharePoint

4.9.1 Overview

This process is most usually executed to;

- Bulk create publishing pages in SharePoint
- Or
- To migrate existing pages into SharePoint from other sources such as WordPress, Drupal or any other CMS.

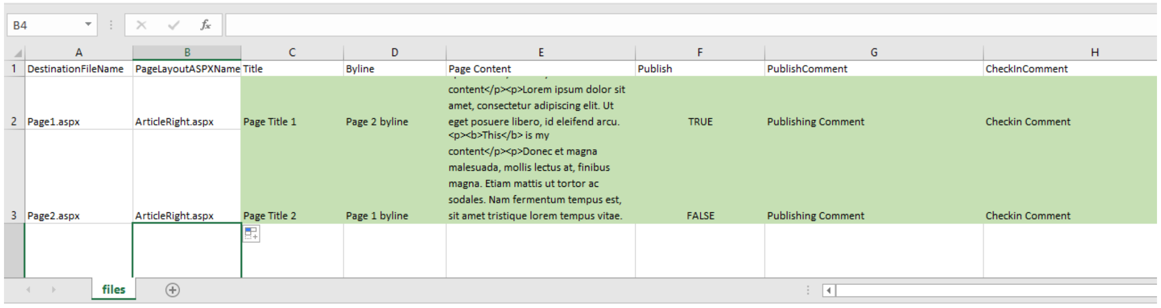
The process is demonstrated here from Excel but using an OleDB source to a CMS system database you can migrate directly. In such migrations, this is done to move the HTML content. You would define a separate import process for resources such as images / downloads etc. and also carry out any manipulation of the HTML before import.

Refer to the example on Legacy Document Management system migration to see how you can use the OleDBTable data source type to import from a SQL Database or similar, including, writing back migration status to the source data.

4.9.2 The Source

We are using the OOTB articleright.aspx page layout and please note the HTML content in the Page Content column.

Note that we have chosen not to publish Page2.aspx



B4

	A	B	C	D	E	F	G	H
1	DestinationFileName	PageLayoutASPXName	Title	Byline	Page Content	Publish	PublishComment	CheckInComment
2	Page1.aspx	ArticleRight.aspx	Page Title 1	Page 2 byline	content</p><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut eget posuere libero, id eleifend arcu.	TRUE	Publishing Comment	Checkin Comment
3	Page2.aspx	ArticleRight.aspx	Page Title 2	Page 1 byline	<p><b>This</b> is my content</p><p>Donec et magna malesuada, mollis lectus at, finibus magna. Etiam mattis ut tortor ac sodales. Nam fermentum tempus est, sit amet tristique lorem tempus vitae.	FALSE	Publishing Comment	Checkin Comment

files

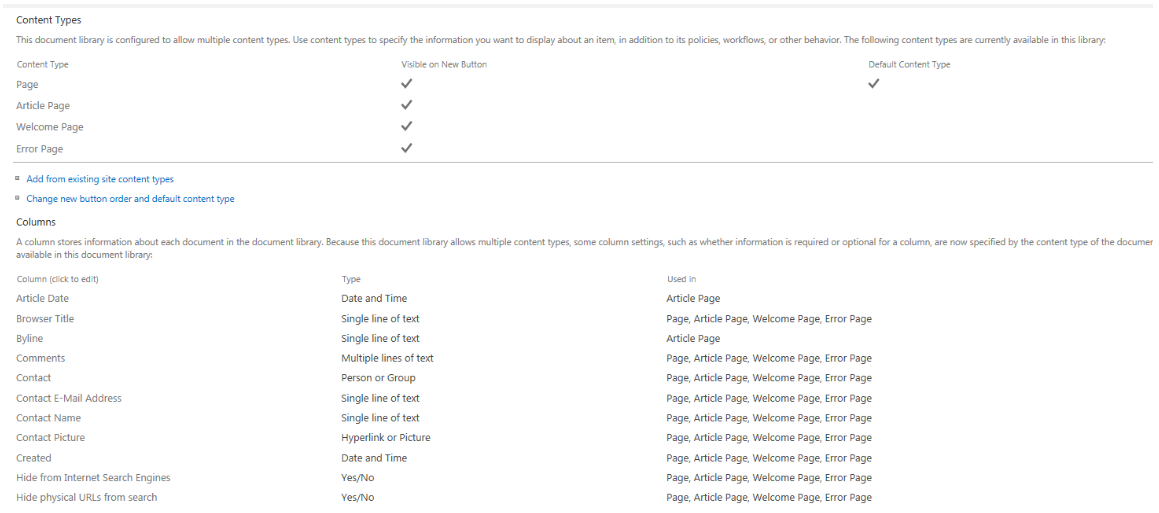
4.9.3 The Configuration

```xml
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <SourceDataSetType>OLEDbSelect</SourceDataSetType>
    <OleDbSourceDataSetSettings>
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-host\Shared
Folders\Projects\ProductDevelopment\ImportForSharePoint\WorkingArea\Examples\PublishingPage.xlsx;Extended Properties="Excel
12.0 Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbTableSourceDataSetSettings />
    <OleDbSelectSourceDataSetSettings>
      <SelectStatement>select * from [files$]</SelectStatement>
    </OleDbSelectSourceDataSetSettings>
  </Source>
  <Destination>
    <AuthenticationSettings>
      <AuthenticationType>Office365</AuthenticationType>
      <domain />
      <username>me@mycompany.onmicrosoft.com</username>
      <encryptedpassed></encryptedpassed>
    </AuthenticationSettings>
    <DestinationItemSettings>
      <!-- Destination item type PublishingPage tells the import to create publishing pages using the page layout name in
the column as set by PageLayoutASPXName  -->
      <DestinationItemType>PublishingPage</DestinationItemType>
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
      <ImportMappings>
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Title</DestinationField>
          <SourceColumn>Title</SourceColumn>
        </ImportMapping>
        <!-- We can map to byline or any field for that matter -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Byline</DestinationField>
          <SourceColumn>Byline</SourceColumn>
        </ImportMapping>
        <!-- The mapping to page content is one of the most import since it will map the HTML in that column into your new
page -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Page Content</DestinationField>
          <SourceColumn>Page Content</SourceColumn>
        </ImportMapping>
      </ImportMappings>
    </DestinationItemSettings>
    <DestinationListSettings>
      <DestinationWebUrlRelative>/sites/SPImportHelper</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/sites/SPImportHelper/Pages</DestinationFolderUrlRelative>
      <DestinationServerUrl>https://mycompany.sharepoint.com</DestinationServerUrl>
      <DestinationListName>Pages</DestinationListName>
    </DestinationListSettings>
    <SourceColumns>
      <SourceFileNameAndPath>FullName</SourceFileNameAndPath>
      <!-- Content Type is ignored for publishing pages since it is set by the page layout -->
      <ContentType>ContentType</ContentType>
      <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
      <DestinationFileName>DestinationFileName</DestinationFileName>
      <!-- If you want to publish your pages then create this column in your data source and set to something that equates
to boolean true for each page that you want to publish  -->
      <Publish>Publish</Publish>
      <!-- The name of the column in the source data set which contains a string to use as the check in comment for the page
-->
      <CheckInComment>CheckInComment</CheckInComment>
      <!-- The name of the column in the source data set which contains a string to use as the publish comment for the page
-->
      <PublishComment>PublishComment</PublishComment>
      <!-- The name of the column in the source data set which contains the name of the page layout to use e.g.
MyPageLayout.aspx -->
      <PageLayoutASPXName>PageLayoutASPXName</PageLayoutASPXName>
    </SourceColumns>
```

```
  </Destination>
</DataSetImportSettings>
```

4.9.4 The Destination

The destination is going to be the “Pages” library in the site created by the publishing feature.



Content Types

This document library is configured to allow multiple content types. Use content types to specify the information you want to display about an item, in addition to its policies, workflows, or other behavior. The following content types are currently available in this library:

Content Type	Visible on New Button	Default Content Type
Page	✓	✓
Article Page	✓	
Welcome Page	✓	
Error Page	✓	

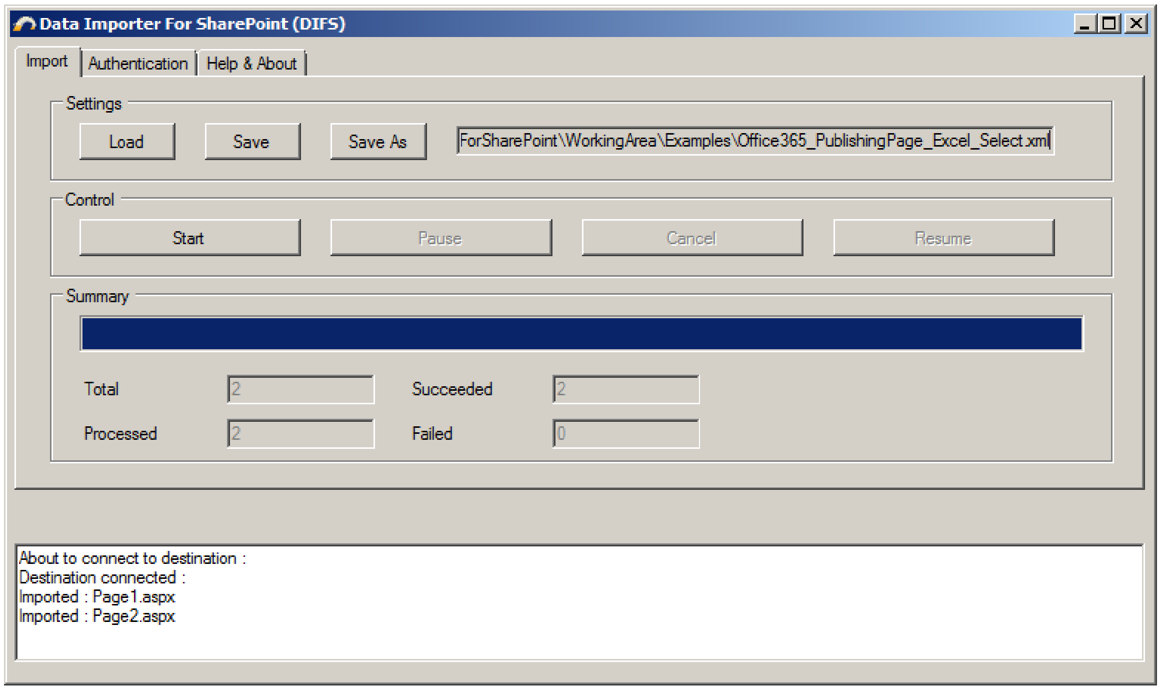
- Add from existing site content types
- Change new button order and default content type

Columns

A column stores information about each document in the document library. Because this document library allows multiple content types, some column settings, such as whether information is required or optional for a column, are now specified by the content type of the documer available in this document library:

Column (click to edit)	Type	Used in
Article Date	Date and Time	Article Page
Browser Title	Single line of text	Page, Article Page, Welcome Page, Error Page
Byline	Single line of text	Article Page
Comments	Multiple lines of text	Page, Article Page, Welcome Page, Error Page
Contact	Person or Group	Page, Article Page, Welcome Page, Error Page
Contact E-Mail Address	Single line of text	Page, Article Page, Welcome Page, Error Page
Contact Name	Single line of text	Page, Article Page, Welcome Page, Error Page
Contact Picture	Hyperlink or Picture	Page, Article Page, Welcome Page, Error Page
Created	Date and Time	Page, Article Page, Welcome Page, Error Page
Hide from Internet Search Engines	Yes/No	Page, Article Page, Welcome Page, Error Page
Hide physical URLs from search	Yes/No	Page, Article Page, Welcome Page, Error Page

4.9.5 The Execution



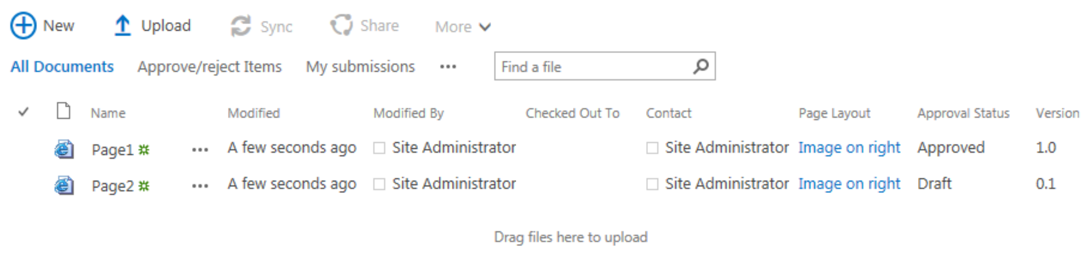
ForSharePoint\WorkingArea\Examples\Office365_PublishingPage_Excel_Select.xml

Total	2	Succeeded	2
Processed	2	Failed	0

```
About to connect to destination :
Destination connected :
Imported : Page1.aspx
Imported : Page2.aspx
```

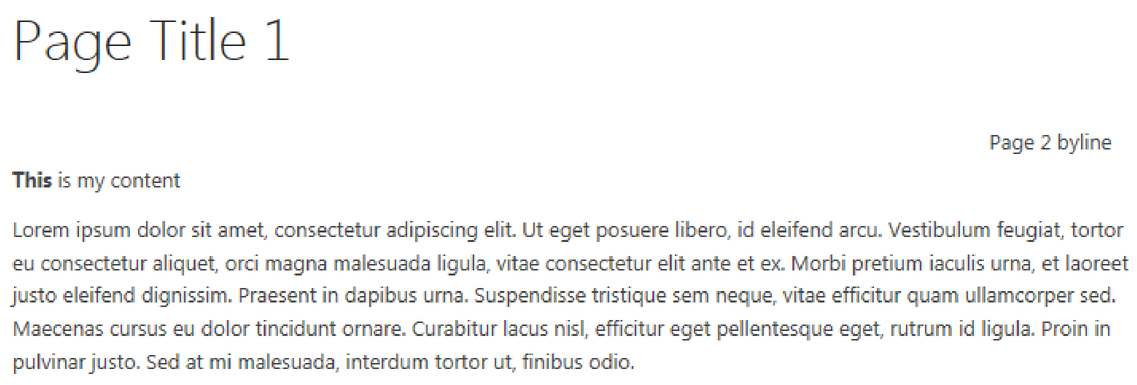
4.9.6 The Result

Two new pages have been created with the correct page layout and desired approval status.



	Name	Modified	Modified By	Checked Out To	Contact	Page Layout	Approval Status	Version
	Page1	A few seconds ago	Site Administrator		Site Administrator	Image on right	Approved	1.0
	Page2	A few seconds ago	Site Administrator		Site Administrator	Image on right	Draft	0.1

If we look at one of the pages it has the desired HTML content, Title and By Line.



Page Title 1

Page 2 byline

This is my content

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut eget posuere libero, id eleifend arcu. Vestibulum feugiat, tortor eu consectetur aliquet, orci magna malesuada ligula, vitae consectetur elit ante et ex. Morbi pretium iaculis urna, et laoreet justo eleifend dignissim. Praesent in dapibus urna. Suspendisse tristique sem neque, vitae efficitur quam ullamcorper sed. Maecenas cursus eu dolor tincidunt ornare. Curabitur lacus nisl, efficitur eget pellentesque eget, rutrum id ligula. Proin in pulvinar justo. Sed at mi malesuada, interdum tortor ut, finibus odio.

4.10 *Import Wiki (Site) Pages into SharePoint*

4.10.1 Overview

This process is most usually executed to;

- Bulk create Wiki (Site) pages in SharePoint
- Or
- To migrate existing pages into SharePoint from other sources such as WordPress, Drupal or any other CMS.

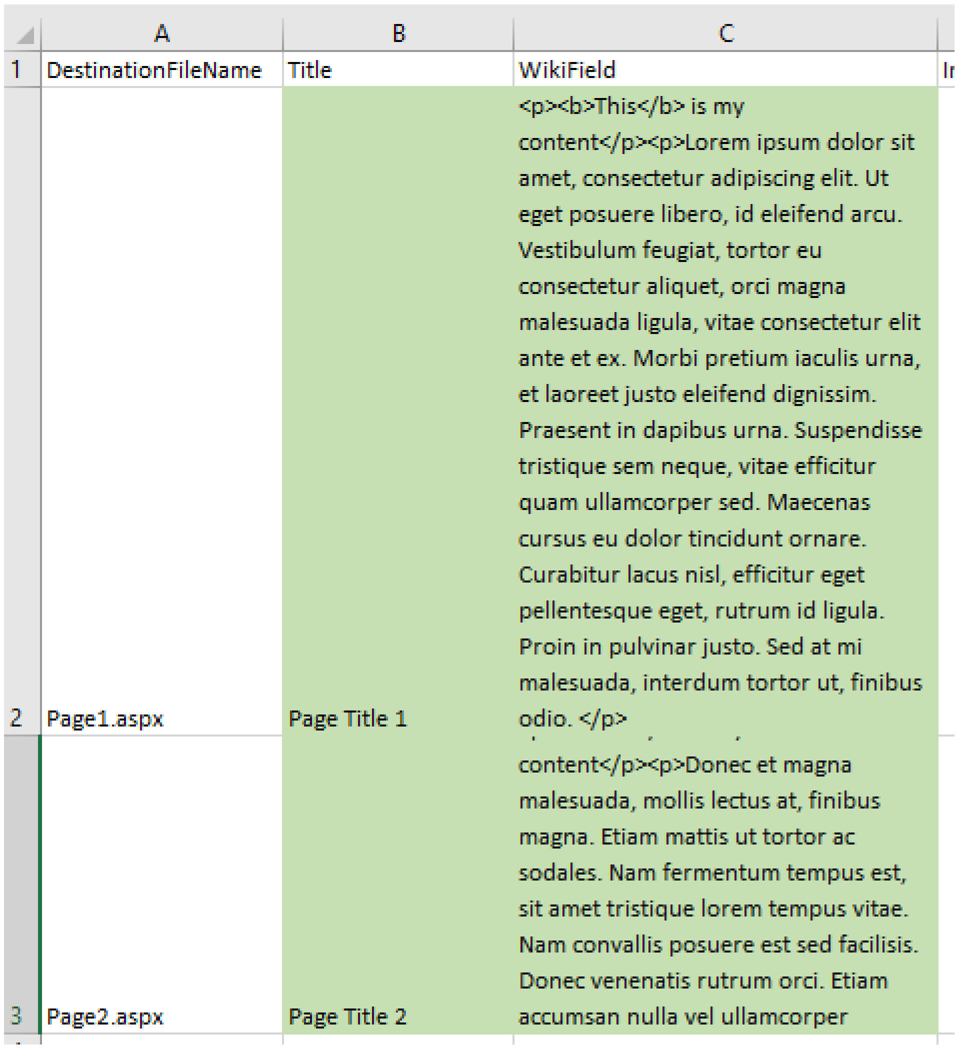
The process is demonstrated here from Excel but using an OleDB source to a CMS system database you can migrate directly. In such migrations, this is done to move the HTML content. You would define a separate import process for resources such as images / downloads etc. and also carry out any manipulation of the HTML before import.

Refer to the example on Legacy Document Management system migration to see how you can use the OleDBTable data source type to import from a SQL Database or similar, including, writing back migration status to the source data.

Please note that if you want to import pages into an Enterprise Wiki created using the site template of that name, then these are actually rather confusingly Publishing Pages and so please refer to the appropriate section.

4.10.2 The Source

Note that the WikiField column contains HTML for the page.



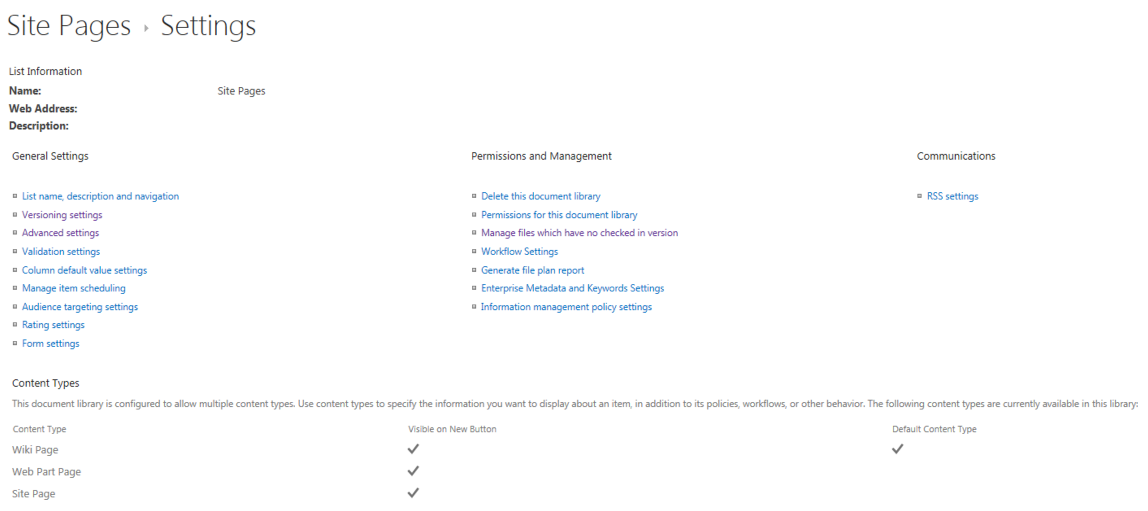
	A	B	C	
1	DestinationFileName	Title	WikiField	Ir
2	Page1.aspx	Page Title 1	<p><b>This</b> is my content</p><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut eget posuere libero, id eleifend arcu. Vestibulum feugiat, tortor eu consectetur aliquet, orci magna malesuada ligula, vitae consectetur elit ante et ex. Morbi pretium iaculis urna, et laoreet justo eleifend dignissim. Praesent in dapibus urna. Suspendisse tristique sem neque, vitae efficitur quam ullamcorper sed. Maecenas cursus eu dolor tincidunt ornare. Curabitur lacus nisl, efficitur eget pellentesque eget, rutrum id ligula. Proin in pulvinar justo. Sed at mi malesuada, interdum tortor ut, finibus odio. </p>	
3	Page2.aspx	Page Title 2	content</p><p>Donec et magna malesuada, mollis lectus at, finibus magna. Etiam mattis ut tortor ac sodales. Nam fermentum tempus est, sit amet tristique lorem tempus vitae. Nam convallis posuere est sed facilisis. Donec venenatis rutrum orci. Etiam accumsan nulla vel ullamcorper	

4.10.3 The Configuration

```
<?xml version="1.0" encoding="utf-8"?>
<DataSetImportSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Source>
    <SourceDataSetType>OLEDbSelect</SourceDataSetType>
    <OleDbSourceDataSetSettings>
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-host\Shared
Folders\Projects\ProductDevelopment\ImportForSharePoint\WorkingArea\Examples\WikiPage.xlsx;Extended Properties="Excel 12.0
Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbTableSourceDataSetSettings />
    <OleDbSelectSourceDataSetSettings>
      <SelectStatement>select * from [files$]</SelectStatement>
    </OleDbSelectSourceDataSetSettings>
  </Source>
  <Destination>
    <AuthenticationSettings>
      <AuthenticationType>Office365</AuthenticationType>
      <domain />
      <username>me@mycompany.onmicrosoft.com</username>
      <encryptedpassed></encryptedpassed>
    </AuthenticationSettings>
    <DestinationItemSettings>
      <DestinationItemType>WikiPage</DestinationItemType>
      <ItemExistsBehaviour>Overwrite</ItemExistsBehaviour>
      <ImportMappings>
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>Title</DestinationField>
          <SourceColumn>Title</SourceColumn>
        </ImportMapping>
        <!-- This is the important mapping - the source column WikiField provides the HTML content for the page -->
        <ImportMapping xsi:type="ImportMapping_String">
          <DestinationField>WikiField</DestinationField>
          <SourceColumn>WikiField</SourceColumn>
        </ImportMapping>
      </ImportMappings>
    </DestinationItemSettings>
    <DestinationListSettings>
      <DestinationWebUrlRelative>/sites/SPImportHelper</DestinationWebUrlRelative>
      <DestinationFolderUrlRelative>/sites/SPImportHelper/SitePages</DestinationFolderUrlRelative>
      <DestinationServerUrl>https://mycompany.sharepoint.com</DestinationServerUrl>
      <!-- Wiki pages normally go into the Site Pages library-->
      <DestinationListName>Site Pages</DestinationListName>
    </DestinationListSettings>
    <SourceColumns>
      <SourceFileNameAndPath>FullName</SourceFileNameAndPath>
      <!-- Ignored-->
      <ContentType>ContentType</ContentType>
      <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
      <!-- The name of the column in the source data set which contains the name of the page to create e.g. My Page.aspx-->
      <DestinationFileName>DestinationFileName</DestinationFileName>
      <Publish>Publish</Publish>
      <CheckInComment>CheckInComment</CheckInComment>
      <PublishComment>PublishComment</PublishComment>
    </SourceColumns>
  </Destination>
</DataSetImportSettings>
```

4.10.4 The Destination

The destination is going to be the “Site Pages” library.



Site Pages › Settings

List Information

Name:	Site Pages
Web Address:	
Description:	

General Settings

- List name, description and navigation
- Versioning settings
- Advanced settings
- Validation settings
- Column default value settings
- Manage item scheduling
- Audience targeting settings
- Rating settings
- Form settings

Permissions and Management

- Delete this document library
- Permissions for this document library
- Manage files which have no checked in version
- Workflow Settings
- Generate file plan report
- Enterprise Metadata and Keywords Settings
- Information management policy settings

Communications

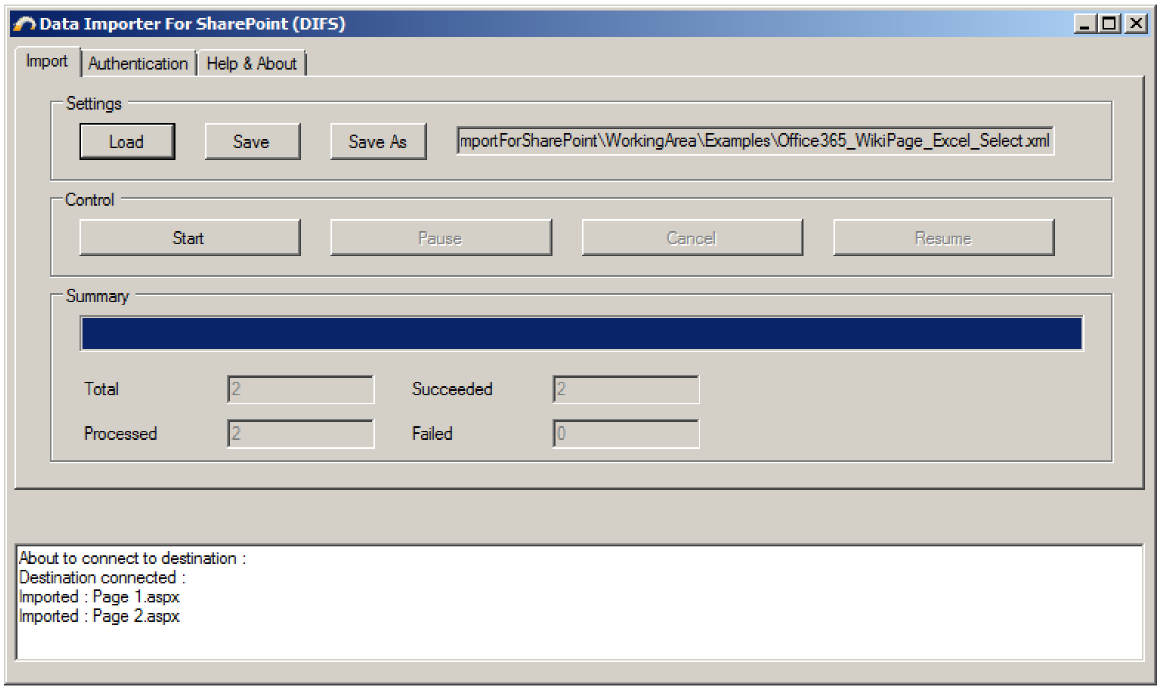
- RSS settings

Content Types

This document library is configured to allow multiple content types. Use content types to specify the information you want to display about an item, in addition to its policies, workflows, or other behavior. The following content types are currently available in this library:

Content Type	Visible on New Button	Default Content Type
Wiki Page	✓	✓
Web Part Page	✓	
Site Page	✓	

4.10.5 The Execution



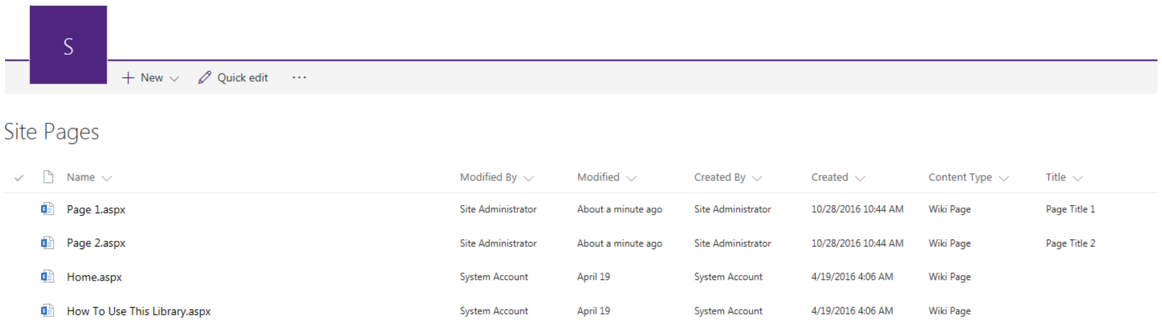
mportForSharePoint\WorkingArea\Examples\Office365_WikiPage_Excel_Select.xml

Total	2	Succeeded	2
Processed	2	Failed	0

```
About to connect to destination :
Destination connected :
Imported : Page 1.aspx
Imported : Page 2.aspx
```

4.10.6 The Result

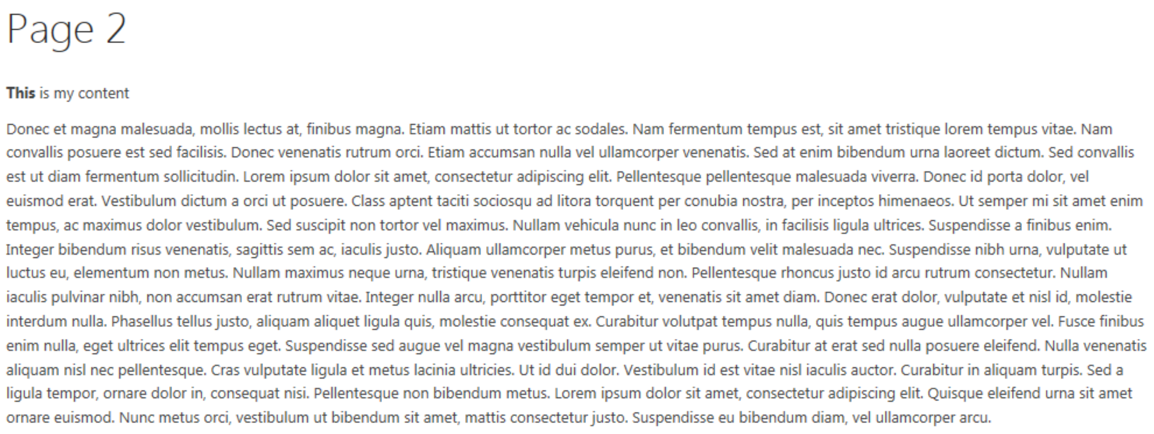
Two new pages have been created with the correct Title.



Site Pages

Name	Modified By	Modified	Created By	Created	Content Type	Title
Page 1.aspx	Site Administrator	About a minute ago	Site Administrator	10/28/2016 10:44 AM	Wiki Page	Page Title 1
Page 2.aspx	Site Administrator	About a minute ago	Site Administrator	10/28/2016 10:44 AM	Wiki Page	Page Title 2
Home.aspx	System Account	April 19	System Account	4/19/2016 4:06 AM	Wiki Page	
How To Use This Library.aspx	System Account	April 19	System Account	4/19/2016 4:06 AM	Wiki Page	

If we look at one of the pages it has the desired HTML content, you will note that the displayed title is the same as the destination file name and note the "Title" field which is normal for this type of page in SharePoint.



Page 2

This is my content

Donec et magna malesuada, mollis lectus at, finibus magna. Etiam mattis ut tortor ac sodales. Nam fermentum tempus est, sit amet tristique lorem tempus vitae. Nam convallis posuere est sed facilisis. Donec venenatis rutrum orci. Etiam accumsan nulla vel ullamcorper venenatis. Sed at enim bibendum urna laoreet dictum. Sed convallis est ut diam fermentum sollicitudin. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque pellentesque malesuada viverra. Donec id porta dolor, vel euismod erat. Vestibulum dictum a orci ut posuere. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut semper mi sit amet enim tempus, ac maximus dolor vestibulum. Sed suscipit non tortor vel maximus. Nullam vehicula nunc in leo convallis, in facilisis ligula ultrices. Suspendisse a finibus enim. Integer bibendum risus venenatis, sagittis sem ac, iaculis justo. Aliquam ullamcorper metus purus, et bibendum velit malesuada nec. Suspendisse nibh urna, vulputate ut luctus eu, elementum non metus. Nullam maximus neque urna, tristique venenatis turpis eleifend non. Pellentesque rhoncus justo id arcu rutrum consectetur. Nullam iaculis pulvinar nibh, non accumsan erat rutrum vitae. Integer nulla arcu, porttitor eget tempor et, venenatis sit amet diam. Donec erat dolor, vulputate et nisl id, molestie interdum nulla. Phasellus tellus justo, aliquam aliquet ligula quis, molestie consequat ex. Curabitur volutpat tempus nulla, quis tempus augue ullamcorper vel. Fusce finibus enim nulla, eget ultrices elit tempus eget. Suspendisse sed augue vel magna vestibulum semper ut vitae purus. Curabitur at erat sed nulla posuere eleifend. Nulla venenatis aliquam nisl nec pellentesque. Cras vulputate ligula et metus lacinia ultricies. Ut id dui dolor. Vestibulum id est vitae nisl iaculis auctor. Curabitur in aliquam turpis. Sed a ligula tempor, ornare dolor in, consequat nisi. Pellentesque non bibendum metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque eleifend urna sit amet ornare euismod. Nunc metus orci, vestibulum ut bibendum sit amet, mattis consectetur justo. Suspendisse eu bibendum diam, vel ullamcorper arcu.

5 Source Configuration

The Source section of the XML configuration file defines the source to Import from.

.

5.1 SourceDataSetType

Defines how to get the source data set.

5.1.1 OLEDbTable

You can define a data source as being a table using OLEDbTable. This allows each row to be updated after import as successful or otherwise along with anyException data. This is useful for large imports.

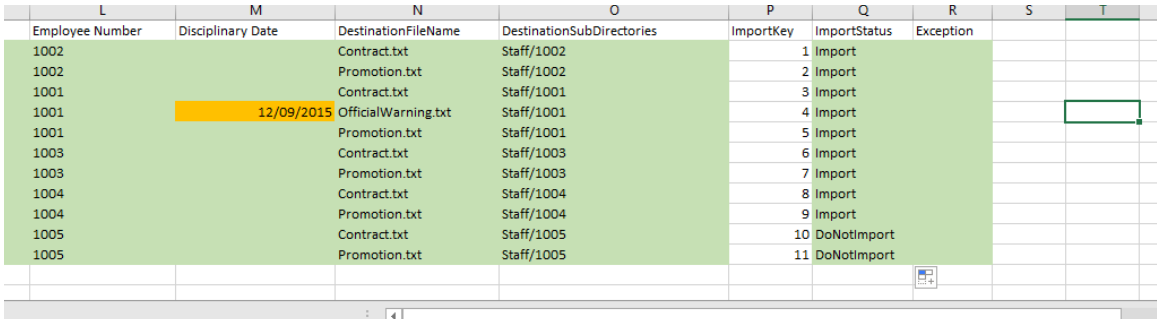
Below the table that we are telling DIFS to use is a worksheet “Clients” in an excel spreadsheet.

```
  <Source>
    <SourceDataSetType>OLEDbTable</SourceDataSetType>
    <OleDbSourceDataSetSettings>
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-
host\Shared Folders\projects\ClientFolders.xlsx;Extended Properties="Excel 12.0
Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbTableSourceDataSetSettings>
      <TableName>clients$</TableName>
    </OleDbTableSourceDataSetSettings>
  </Source>
```

The table must contain columns ImportKey, ImportStatus, Exception.

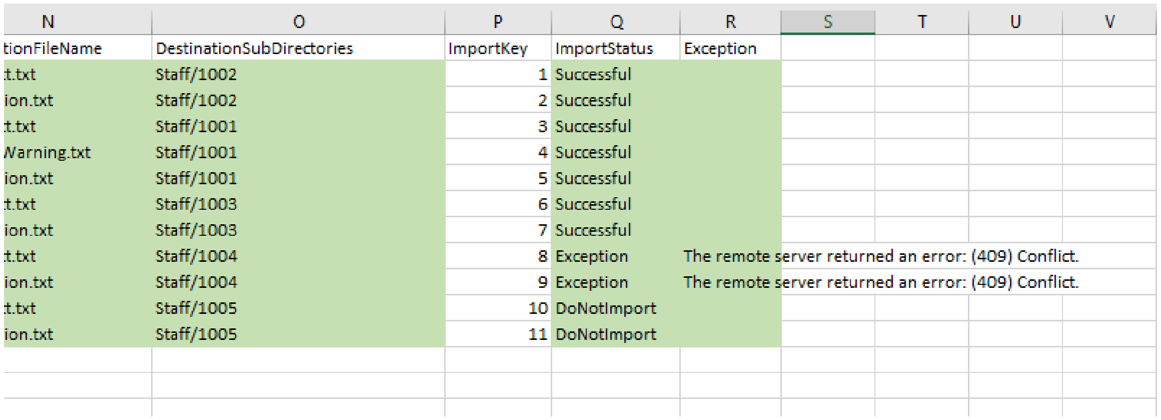
ImportKey must be unique.

In the example below only the first 10 rows will be considered for import since only they have the importstatus “Import”.



Employee Number	Disciplinary Date	DestinationFileName	DestinationSubDirectories	ImportKey	ImportStatus	Exception
1002		Contract.txt	Staff/1002	1	Import	
1002		Promotion.txt	Staff/1002	2	Import	
1001		Contract.txt	Staff/1001	3	Import	
1001	12/09/2015	OfficialWarning.txt	Staff/1001	4	Import	
1001		Promotion.txt	Staff/1001	5	Import	
1003		Contract.txt	Staff/1003	6	Import	
1003		Promotion.txt	Staff/1003	7	Import	
1004		Contract.txt	Staff/1004	8	Import	
1004		Promotion.txt	Staff/1004	9	Import	
1005		Contract.txt	Staff/1005	10	DoNotImport	
1005		Promotion.txt	Staff/1005	11	DoNotImport	

After you have run the import the spreadsheet will be updated.



tionFileName	DestinationSubDirectories	ImportKey	ImportStatus	Exception
t.txt	Staff/1002	1	Successful	
ion.txt	Staff/1002	2	Successful	
t.txt	Staff/1001	3	Successful	
Warning.txt	Staff/1001	4	Successful	
ion.txt	Staff/1001	5	Successful	
t.txt	Staff/1003	6	Successful	
ion.txt	Staff/1003	7	Successful	
t.txt	Staff/1004	8	Exception	The remote server returned an error: (409) Conflict.
ion.txt	Staff/1004	9	Exception	The remote server returned an error: (409) Conflict.
t.txt	Staff/1005	10	DoNotImport	
ion.txt	Staff/1005	11	DoNotImport	

You can then correct the causes of the Exception, change the ImportStatus back to Import and re-run.

5.1.2 OLEDbSelect

You can define a data source as simply the results of a select statement using OLEDbSelect.

```
  <Source>
    <!-- The source used is OLEDbSelect meaning that DIFS expected to run a SQL 
select statement against an OleDB data source.  OleDB select will ignore columns 
like importstatus-->
    <SourceDataSetType>OLEDbSelect</SourceDataSetType>
    <!-- The Connection string uses a microsoft provider to access an Excel 
spreadsheet-->
    <OleDbSourceDataSetSettings>
      <ConnectionString>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=\\vmware-
host\Shared 
Folders\Projects\ProductDevelopment\ImportForSharePoint\WorkingArea\Examples\JobDe
```

```
scriptions.xlsx;Extended Properties="Excel 12.0
Xml;HDR=YES;IMEX=0";</ConnectionString>
    </OleDbSourceDataSetSettings>
    <OleDbTableSourceDataSetSettings />
    <!-- The select statement will get all the jobs from the jobs worksheet -->
    <OleDbSelectSourceDataSetSettings>
      <SelectStatement>select * from [jobs$]</SelectStatement>
    </OleDbSelectSourceDataSetSettings>
  </Source>
```

5.1.3 ODBCSelect

You can define a data source as simply the results of a select statement using ODBCSelect.

```
  <Source>
    <!-- The source used is ODBCSelect meaning that DIFS expects to run a SQL
select statement against an ODBC data source.  This will ignore columns like
importstatus -->
    <SourceDataSetType>ODBCSelect</SourceDataSetType>
    <ODBCSourceDataSetSettings>
      <!-- So a system DSN has been created called WordPress, this has in fact
been pointed to a MySQL database hosting WordPress -->
      <ConnectionString>DSN=WordPress;Uid=root;Pwd=password;</ConnectionString>
    </ODBCSourceDataSetSettings>
    <ODBCTableSourceDataSetSettings />
    <ODBCSelectSourceDataSetSettings>
      <!-- The select statement to run to obtain the rows for import.  This one
actually gets all wordpress blog posts -->
      <SelectStatement>select *, concat(post_name,'.aspx') as DestinationFileName,
replace(post_content,'\n','&lt;br/&gt;') as PageContent, 'True' as Publish from
wp_posts where post_type = 'post' and post_status = 'publish'</SelectStatement>
    </ODBCSelectSourceDataSetSettings>
  </Source>
```

5.1.4 FileSystemFiles

You can define a data set as the contents of local directory or UNC path.

```
  <Source>
    <SourceDataSetType>FileSystemFiles</SourceDataSetType>
    <FileSystemFilesDataSetSettings>
      <Path>c:\Import</Path>
    </FileSystemFilesDataSetSettings>
</Source>
```

5.2 ConnectionString

This is any valid connection string.

A google of “connection string examples” will give us example connection strings for 100’s of different providers.

The connection string must match the source data type.   For example if using OleDB you need any OleDB connection string, similarly an ODBC connection string if you are using ODBC.

DIFS is most commonly used with Excel, Microsoft SQL Server and MySQL.

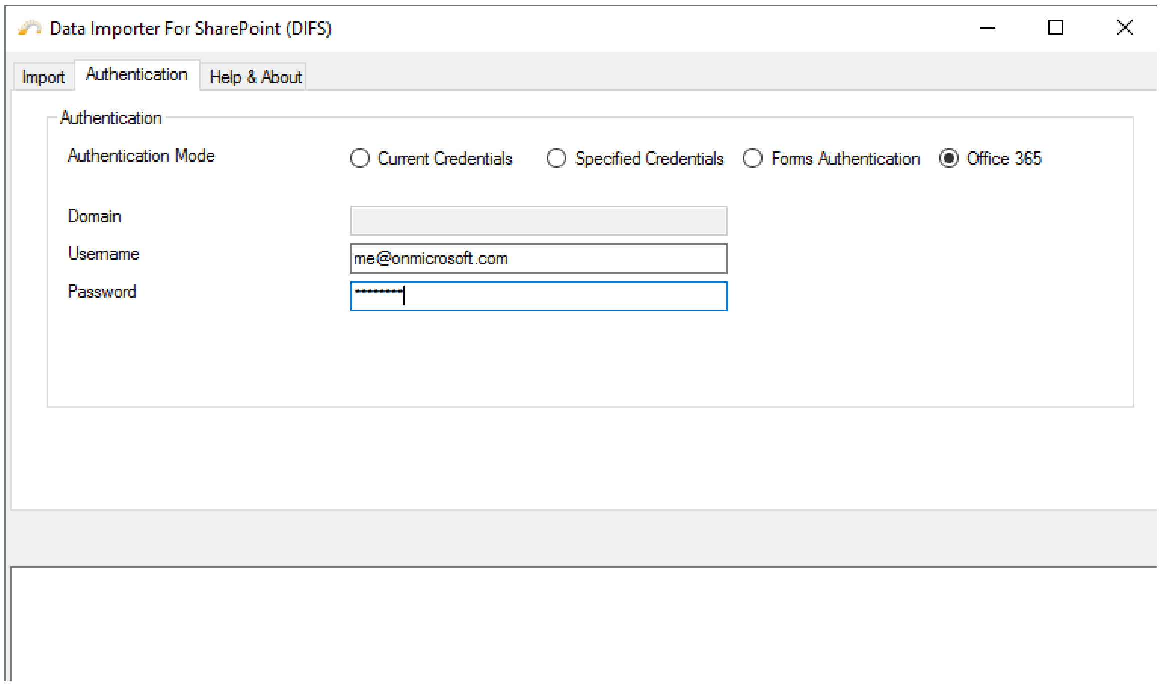
6 Destination Configuration

The Source section of the XML configuration file defines the destination location, authentication, data mapping etc.

6.1 *AuthenticationSettings*

This tells DIFS how to authenticate to SharePoint.

Since credentials are encrypted the easiest way to configure authentication is from the DIFS user interface – authentication tab.



Authentication

Authentication Mode: ( ) Current Credentials ( ) Specified Credentials ( ) Forms Authentication [✓ Office 365]

Domain:

Username: me@onmicrosoft.com

Password: ********

Once entered you can save the configuration file from the Import tab if required.

6.2 DestinationItemSettings

6.2.1 DestinationItemType

This tells DIFS the base type – Item, Document, Folder etc – so DIFS knows the basic action to execute.

The content type to set on the item is defined elsewhere.

6.2.2 ItemExistsBehaviour

This setting controls what happens when a file already exists.

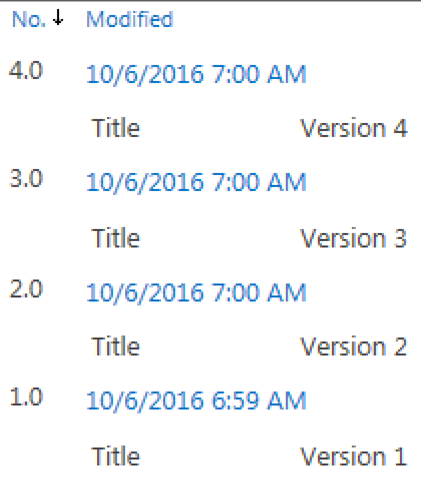
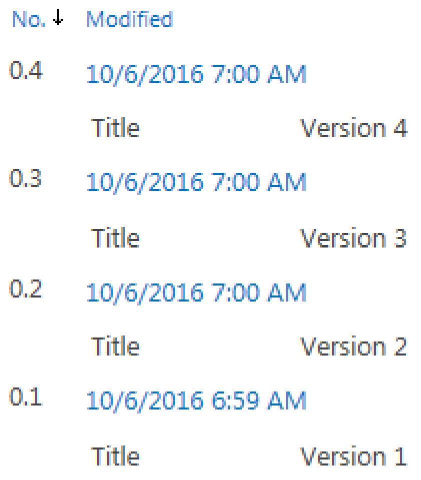
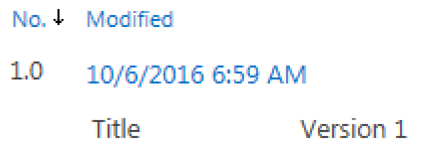
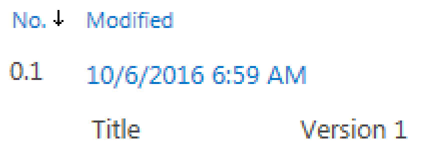
As such it also governs how you may import multiple versions. This can be useful when migrating from legacy systems which had version control and where you wish to re-produce that in SharePoint.

The effective that it has is determined by what versioning setting a SharePoint document library has.

The table below shows the files we are uploading – each source is targeted to the same destination TestA.txt.

FullName	ContentType	Title	DestinationFileName
Version1.txt	Document	Version 1	TestA.txt
Version2.txt	Document	Version 2	TestA.txt
Version3.txt	Document	Version 3	TestA.txt
Version4.txt	Document	Version 4	TestA.txt

The table below shows the SharePoint setting will determine the end result.

ItemExistsBehaviour	Library	Effect
Overwrite	Major Versions	 No. ↓ Modified 4.0 10/6/2016 7:00 AM Title Version 4 3.0 10/6/2016 7:00 AM Title Version 3 2.0 10/6/2016 7:00 AM Title Version 2 1.0 10/6/2016 6:59 AM Title Version 1
Overwrite	Minor Version	 No. ↓ Modified 0.4 10/6/2016 7:00 AM Title Version 4 0.3 10/6/2016 7:00 AM Title Version 3 0.2 10/6/2016 7:00 AM Title Version 2 0.1 10/6/2016 6:59 AM Title Version 1
Overwrite	No Versioning	Single version – Version 4
DoNotOverwrite	Major Versions	 No. ↓ Modified 1.0 10/6/2016 6:59 AM Title Version 1
DoNotOverwrite	Minor Version	 No. ↓ Modified 0.1 10/6/2016 6:59 AM Title Version 1
DoNotOverwrite	No Versioning	Single version – Version 1

6.2.3 ImportMappings

6.2.3.1 Default Mappings

When the destination item type is document the Created and Modified properties in SharePoint will be set to that of the source document. This can be overridden with a custom mapping to get the data from the data set if you require.

6.2.3.2 Custom Mappings

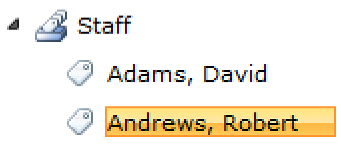
Using ImportMappings allows you to set column / field values on the destination items.

Mapping	Usage

String	Take the string equivalent of SourceColumn and set DestinationField to that value.
	
	`<ImportMapping xsi:type="ImportMapping_String">`
	`  <DestinationField>Title</DestinationField>`
	`  <SourceColumn>Title</SourceColumn>`
	`</ImportMapping>`

DateTimeFromString	Take the string value of SourceColumn, convert it to DateTime using the specified culture and format. Set DestinationField to that value.
	
	`<ImportMapping xsi:type="ImportMapping_DateTimeFromString">`
	`<DestinationField>DateOfLeaving</DestinationField>`
	`<SourceColumn>Date of Leaving</SourceColumn>`
	`<ConversionMask>dd/MM/yyyy hh:mm:ss</ConversionMask>`
	`<Culture>en-GB</Culture>`
	`</ImportMapping>`
	
	`<!-- Refer to DateTime.ParseExact on MSDN for more information on ConversionMask (format) and culture (iformat provider)-->`

Lookup	Take the value of SourceColumn. Find that value in the LookUpList. Set DestinationField to the id of that item.
	
	`<ImportMapping xsi:type="ImportMapping_Lookup">`
	`<!-- This is the field of type lookup.-->`
	`<DestinationField>Job</DestinationField>`
	`<SourceColumn>Job Title</SourceColumn>`
	`<!-- This is the list that contains the lookup values -->`
	`<LookupListTitle>Items</LookupListTitle>`

	<!-- This is the field in the lookup list the value of which matches the value in the source column--> <LookupFieldInternalName>Title</LookupFieldInternalName> <!-- This is the CAML value type Text,Number,DateTime,Guid,MultiChoice,Lookup for the field on in the lookup list that matches the value specified in the source column --> <LookupFieldCAMLType>Text</LookupFieldCAMLType> </ImportMapping>
ManagedMetaDataAutoAdd	Find the term set which DestinationField is linked to. Find a term in that set matching the value in SourceColumn. Set DestinationField to that term id. If the term does not exist DIFS will try and create it. “Andrews, Robert” would become.  Staff Adams, David Andrews, Robert If the term set is nested use, for example, a source value of “Managers\Andrews, Robert” or “UK\Staff\Managers\ Andrews, Robert” <ImportMapping xsi:type="ImportMapping_ManagedMetaDataAutoAdd"> <DestinationField>Employee Name</DestinationField> <SourceColumn>Employee Name</SourceColumn> </ImportMapping> Multi-value fields should be separated by a semi colon. E.g. “SQL; SharePoint” Would become; Skills SharePoint; SQL May not work for SharePoint Online.
ManagedMetaDataCSOM	Find the term set which DestinationField is linked to. Find a term in that set matching the value in SourceColumn. Set DestinationField to that term id. Supports Single and Multi-value

fields.

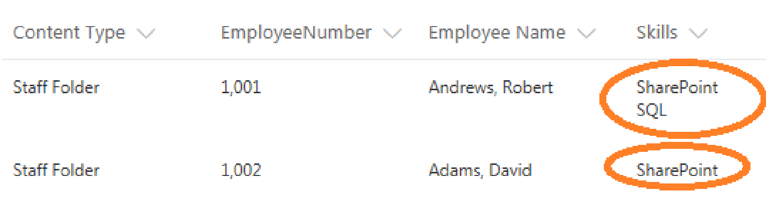
Multi-value fields should be separated by a semi colon.

The mapping will find the term (e.g. SQL) anywhere in a nested term set and will set the item to the value of the first matching term it finds.

Will not work on SharePoint 2010.

Title	Employee Number	Employee Name	Skills
1002	1002	Adams, David	SharePoint
1001	1001	Andrews, Robert	SharePoint;SQL

Would become;



Content Type	EmployeeNumber	Employee Name	Skills
Staff Folder	1,001	Andrews, Robert	SharePoint SQL
Staff Folder	1,002	Adams, David	SharePoint

```
<ImportMapping xsi:type="ImportMapping_ManagedMetaDataCSOM">
          <DestinationField>Skills</DestinationField>
          <SourceColumn>Skills</SourceColumn>
</ImportMapping>
```

Native

Get the value of SourceColumn and set destination with no conversion. This assumes that the source and destination have the same data type, e.g.both Numeric or both DateTime or that they are close enough that .Net can automatically cast the value, e.g. Integer to Numeric.

```
<ImportMapping xsi:type="ImportMapping_Native">
<DestinationField>EmployeeNumber</DestinationField>
<SourceColumn>Employee Number</SourceColumn>
</ImportMapping>
```

User

Get the value of SourceColumn, get the user object for that person in SharePoint. Set DestinationField (which must be a person field of course) to that value.

The value of Source could be “Bob Smith” or his email eg Bob.Smith@company.onmicrosoft.com should also work.

```
<ImportMapping xsi:type="ImportMapping_User">
```

	<DestinationField>Client Manager</DestinationField> <SourceColumn>ClientManager</SourceColumn> </ImportMapping>

6.3 DestinationExecutionSettings

6.3.1 PerItemImportThrottle

Sets a wait period in milliseconds that will be observed after the import of each item.

The default is 0.

This can be useful when working with older implementations to avoid triggering web request flood protection mechanisms for larger imports or where you are seeing "The underlying connection was closed" midway through the import.

For newer implementations returning the correct http status when overloaded the import will throttle and retry automatically.

The setting can also be useful if you want to reduce the overhead placed upon the farm as a result of an import being run.

The following example will wait 10 seconds after each item is imported.

```
<DestinationExecutionSettings>
  <PerItemImportThrottle>10000</PerItemImportThrottle>
</DestinationExecutionSettings>
```

6.4 SourceColumns

This section defines the columns that appear in the source data that tell DIFS how to create the item in SharePoint.

In all instances the value of the setting is the name of the column in the source data.

I.e.

```
<SourceFileNameAndPath>FullName</SourceFileNameAndPath>
```

Infers…

FullName	ContentType
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version1.txt	Document
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version2.txt	Document
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version3.txt	Document

The columns only need to be in your source data set if they are Mandatory. So for example SourceFileNameAndPath is mandatory for importing documents but not folders or items.

6.4.1 SourceFileNameAndPath

The name of the column in the source data which provides the full path to the file being imported.

Mandatory for documents.

6.4.2 ContentType

The name of the column in the source data which provides the name of the content type to set the item to.

Mandatory.

6.4.3 DestinationSubFolder

The name of the column in the source data which provides the sub folder into which to import the item.

6.4.4 DestinationFileName

The name of the column in the source data which provides the destination file name / folder name of the item to create in SharePoint. The values in that column cannot include any characters which are invalid for SharePoint such as ?!\ etc.

Mandatory for documents and folders.

6.4.5 Publish

The name of the column in the source data which contains the value "Yes" when the item is to be published. This column is particularly useful if you are trying to produce a specific version history in SharePoint.

To use the Publish column the destination library must have minor versions enabled.

The ItemExistsBehaviour must be set to overwrite.

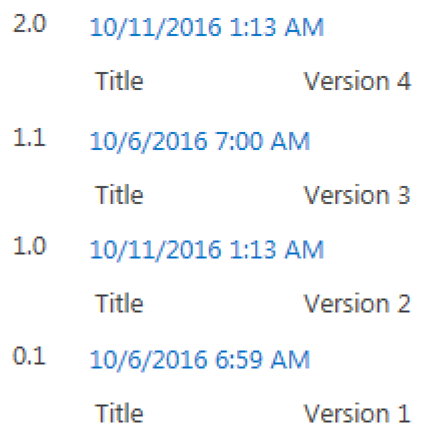
Working with this configuration:

```
<SourceColumns>
  <SourceFileNameAndPath>FullName</SourceFileNameAndPath>
  <ContentType>ContentType</ContentType>
  <DestinationSubFolder>DestinationSubDirectories</DestinationSubFolder>
  <DestinationFileName>DestinationFileName</DestinationFileName>
  <Publish>Publish</Publish>
</SourceColumns>
```

And with this source data:

FullName	Content Type	Title	DestinationFileName	Publish
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version1.txt	Document	Version 1	TestA.txt	FALSE
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version2.txt	Document	Version 2	TestA.txt	TRUE
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version3.txt	Document	Version 3	TestA.txt	FALSE
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version4.txt	Document	Version 4	TestA.txt	TRUE

You will end up with this version history:



2.0	10/11/2016 1:13 AM	
	Title	Version 4
1.1	10/6/2016 7:00 AM	
	Title	Version 3
1.0	10/11/2016 1:13 AM	
	Title	Version 2
0.1	10/6/2016 6:59 AM	
	Title	Version 1

Where the publish column is forcing the item to be published it becomes a major version.

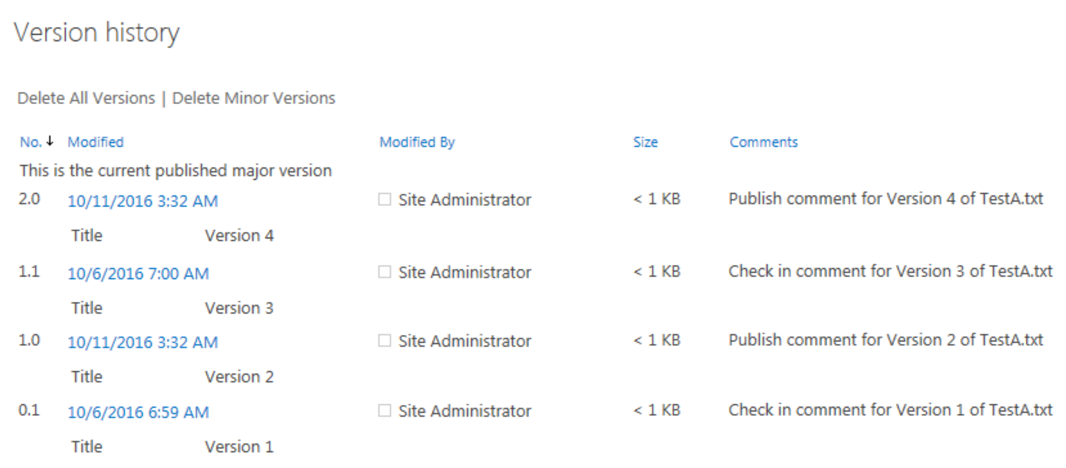
6.4.6 PublishComment

The name of the column in the source data which contains the value to use as the PublishComment.

The following source data:

FullName	ContentType	Title	DestinationFileName	Publish	PublishComment	CheckInComment
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version1.txt	Document	Version 1	TestA.txt	FALSE	Publish comment for Version 1 of TestA.txt	Check in comment for Version 1 of TestA.txt
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version2.txt	Document	Version 2	TestA.txt	TRUE	Publish comment for Version 2 of TestA.txt	Check in comment for Version 2 of TestA.txt
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version3.txt	Document	Version 3	TestA.txt	FALSE	Publish comment for Version 3 of TestA.txt	Check in comment for Version 3 of TestA.txt
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version4.txt	Document	Version 4	TestA.txt	TRUE	Publish comment for Version 4 of TestA.txt	Check in comment for Version 4 of TestA.txt

Will produce the following version history.



Version history

Delete All Versions | Delete Minor Versions

No. ↓	Modified	Modified By	Size	Comments
This is the current published major version				
2.0	10/11/2016 3:32 AM	☐ Site Administrator	< 1 KB	Publish comment for Version 4 of TestA.txt
	Title Version 4			
1.1	10/6/2016 7:00 AM	☐ Site Administrator	< 1 KB	Check in comment for Version 3 of TestA.txt
	Title Version 3			
1.0	10/11/2016 3:32 AM	☐ Site Administrator	< 1 KB	Publish comment for Version 2 of TestA.txt
	Title Version 2			
0.1	10/6/2016 6:59 AM	☐ Site Administrator	< 1 KB	Check in comment for Version 1 of TestA.txt
	Title Version 1			

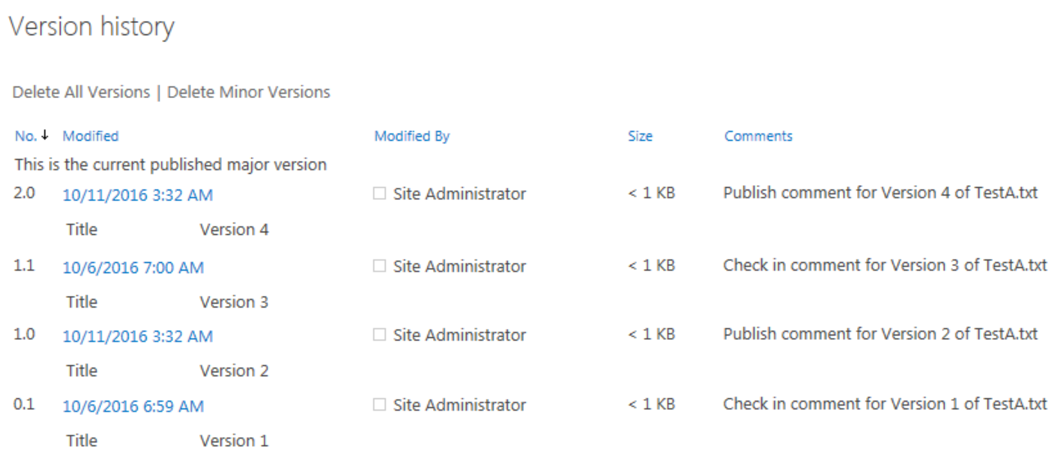
6.4.7 CheckInComment

The name of the column in the source data which contains the value to use as the Check InC omment.

The following source data:

FullName	ContentType	Title	DestinationFileName	Publish	PublishComment	CheckInComment
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version1.txt	Document	Version 1	TestA.txt	FALSE	Publish comment for Version 1 of TestA.txt	Check in comment for Version 1 of TestA.txt
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version2.txt	Document	Version 2	TestA.txt	TRUE	Publish comment for Version 2 of TestA.txt	Check in comment for Version 2 of TestA.txt
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version3.txt	Document	Version 3	TestA.txt	FALSE	Publish comment for Version 3 of TestA.txt	Check in comment for Version 3 of TestA.txt
\\vmware-host\Shared Folders\TestData\Generic\Versions\Version4.txt	Document	Version 4	TestA.txt	TRUE	Publish comment for Version 4 of TestA.txt	Check in comment for Version 4 of TestA.txt

Will produce the following version history.



Version history

Delete All Versions | Delete Minor Versions

No. ↓	Modified	Modified By	Size	Comments
This is the current published major version				
2.0	10/11/2016 3:32 AM	☐ Site Administrator	< 1 KB	Publish comment for Version 4 of TestA.txt
	Title Version 4			
1.1	10/6/2016 7:00 AM	☐ Site Administrator	< 1 KB	Check in comment for Version 3 of TestA.txt
	Title Version 3			
1.0	10/11/2016 3:32 AM	☐ Site Administrator	< 1 KB	Publish comment for Version 2 of TestA.txt
	Title Version 2			
0.1	10/6/2016 6:59 AM	☐ Site Administrator	< 1 KB	Check in comment for Version 1 of TestA.txt
	Title Version 1			

7 Troubleshooting

7.1 *Introduction*

This section provides some basic details on troubleshooting an import.

7.2 *Enable logging*

7.2.1 To user interface

Go to the logging tab.

7.2.2 To file / event log

You get enable logging to get more detailed information on your import.

This is particular useful when, for example, import mappings are not working because far more detail is logged here than is reported to the console.

Logging is performed by .Net trace listeners and these may be configured via the app configuration file e.g. difs.exe.config.

Section system diagnostics.

Below we have set autoflush to true and placed the log file in the temp directory to enable us to get some extra logging.

```
  <system.diagnostics>
    <!-- By setting autoflush false the listener will not be written to; this can
get BIG.  Set to true for troubleshooting -->
    <trace autoflush="true"></trace>
    <sources>
      <source name="SPImportHelper">
        <listeners>
          <remove name="Default"/>
          <!-- The log file specified below will contain full details the of
import.  The user running the import must have write permission and autoflush must
be set to true.  If left in the default program files directory you must Run as
Administrator -->
          <add name="eventlog"
              type="System.Diagnostics.TextWriterTraceListener"
              initializeData="c:\temp\SPImportHelper.log">
            <!--
            <filter type="System.Diagnostics.EventTypeFilter"
initializeData="Information"/>
            -->
          </add>
        </listeners>
      </source>
    </sources>
  </system.diagnostics>
```

Refer to the following articles for information on how you might use trace listeners.

https://msdn.microsoft.com/en-us/library/1txedc80(v=vs.110).aspx

https://msdn.microsoft.com/en-us/library/sk36c28t(v=vs.110).aspx