

# SharePoint 2010 Multiple File Upload Script

Version 1.0

<http://get-spscripts.com>

<http://www.codeplex.com/spfileupload>

[@PhillipChilds](#)

## Description

The “SharePoint 2010 Multiple File Upload Script” is a PowerShell script which uploads multiple files from a local folder or network share into SharePoint 2010 document libraries. It also provides the option to apply metadata to the files during upload using an XML manifest file.

It includes the following features:

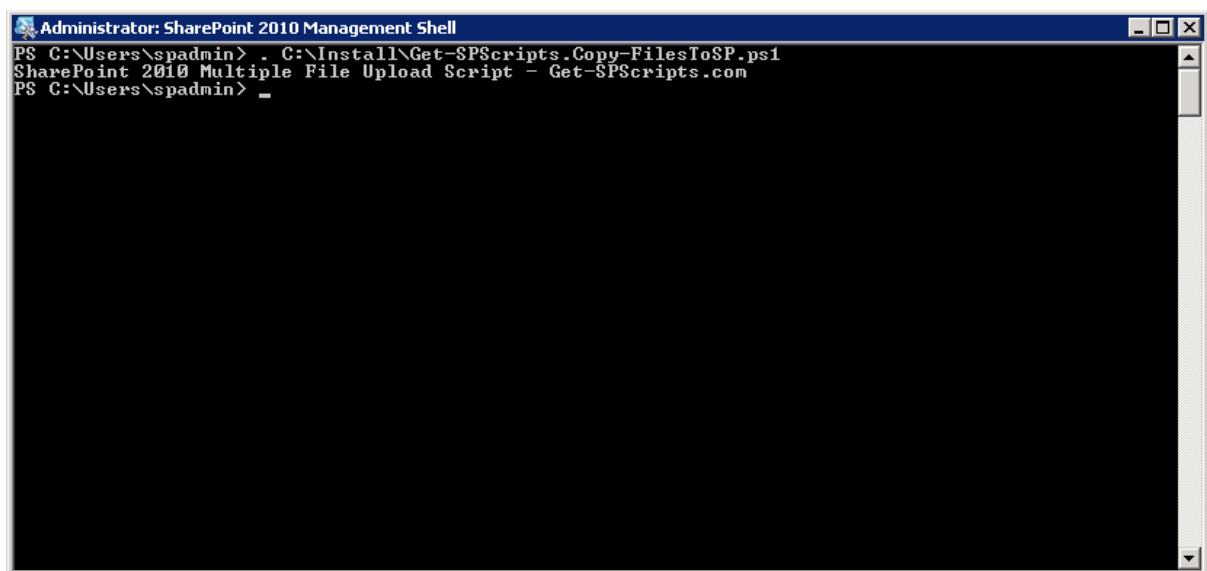
- Upload files from a single folder or an entire folder hierarchy
- Retain the folder hierarchy from the source path into the destination document library
- Choose the root or a specific destination folder in the document library to copy the files into
- Support for adding values to the following column types in the manifest file:
  - Single line of text
  - Multiple lines of text
  - Choice – Single value (drop-down/radio button)
  - Choice – Multiple value (checkboxes)
  - Currency
  - Number
  - Yes/No (checkbox)
  - Person or Group – Single value
  - Person or Group – Multiple value
  - Hyperlink or Picture
  - Managed Metadata – Single value
  - Managed Metadata – Multiple value
- Automatic file check in after upload, if required
- Automatic file approval after upload, if required
- Option to overwrite existing files in the destination document library
- Option to flatten the structure when copying from a multi-level folder hierarchy to a single folder in a document library (i.e., merging sub-folders to a single location)

One of the advantages of using PowerShell for this tool is the ability for you to customise the script should you wish to add any custom features. Any contributions to the Codeplex site would be welcomed ☺

Whilst I would expect the script to work with any compatible version of SharePoint 2010 and PowerShell 2.0 console application, it has only been tested using the SharePoint 2010 Management Shell and Windows PowerShell ISE on Windows Server 2008 R2 with SharePoint Server 2010 Enterprise Edition. This guide assumes that you already have the appropriate pre-requisites and permissions applied to be able to administer SharePoint 2010 with PowerShell.

## Installation

1. Go to <http://www.codeplex.com/spfileupload> and download the **Get-SPScripts.Copy-FilesToSP.ps1** script to a location on your hard disk.
2. Start the **SharePoint 2010 Management Shell** or **Windows PowerShell ISE** from the start menu.
3. Type “. ” (to clarify, that is [dot][space]) followed by the full path to the downloaded script file from step 1 – e.g., . C:\Install\Get-SPScripts.Copy-FilesToSP.ps1 – and hit RETURN. You should receive the message “SharePoint 2010 Multiple File Upload Script - Get-SPScripts.com”.



```
Administrator: SharePoint 2010 Management Shell
PS C:\Users\spadmin> . C:\Install\Get-SPScripts.Copy-FilesToSP.ps1
SharePoint 2010 Multiple File Upload Script - Get-SPScripts.com
PS C:\Users\spadmin> _
```

4. You are now ready to start uploading files using the script.

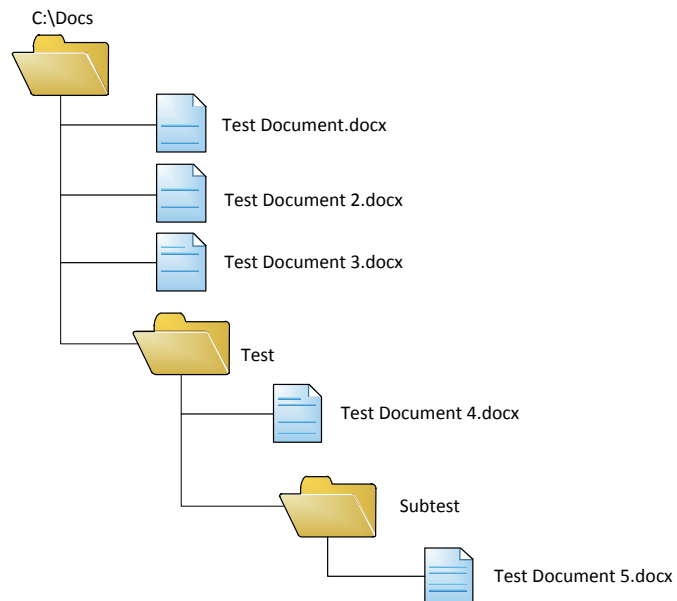
## User Guide

You use the SharePoint 2010 Multiple File Upload Script by typing the *Copy-FilesToSP* command in the PowerShell console interface. This command includes a number of parameters and switches to provide the various features available from the script. The table below lists each parameter available to you:

Parameter	Description	Example
<b>LocalPath</b> (mandatory)	The local or network (UNC) path of the file or files to be uploaded to SharePoint	"C:\Docs"
<b>SiteUrl</b> (mandatory)	The URL of the site in SharePoint where the files will be uploaded	"http://portal/team"
<b>Library</b> (mandatory)	The display name of the document library in SharePoint where the files will be uploaded	"Shared Documents"
<b>LibraryStartFolder</b> (optional)	The document library folder into which the files will be uploaded. The root folder will be used if the LibraryStartFolder parameter is not specified	"/Folder/Subfolder"
<b>ManifestFilePath</b> (optional)	The path and file name of the manifest file being used to categorise the uploaded files with metadata	"C:\Install\DocManifest.xml"
<b>IncludeSubFolders</b> (optional)	Copy files from the root folder and all subfolders in the local or network path specified. If this parameter is not specified, files will be copied from the root folder only.	N/A
<b>Approve</b> (optional)	Automatically approve uploaded files, if required. If this parameter is not specified, the script will not approve files by default.	N/A
<b>CheckIn</b> (optional)	Automatically check in uploaded files, if required. If this parameter is not specified, the script will not check in files by default.	N/A
<b>Overwrite</b> (optional)	Overwrite any files found in the document library with the same name. If this parameter is not specified, the script will not overwrite files by default.	N/A
<b>FlattenStructure</b> (optional)	Merge sub-folders from the local or network path specified to a single location in the document library. If this parameter is not specified, the script will replicate the folder structure from the source folder into the document library.	N/A

## Examples:

For these examples we are using a local folder with the following hierarchy



### Scenario 1 – Replicate local folder structure to document library and copy all files

The following command will copy all files from “C:\Docs” and all subfolders to the document library “Shared Documents” in the site “http://portal/team”, overwriting any existing documents found with the same name. The command will also tag these documents with a metadata structure defined in the “DocManifest.xml” file:

**Copy-FiletoSP -LocalPath "C:\Docs" -SiteUrl "http://portal/team" -Library "Shared Documents" -ManifestFilePath "C:\Install\DocManifest.xml" -IncludeSubFolders -Overwrite**

As a result of this command, the document library will contain an exact replica of the folder structure and files contained in the C:\Docs structure defined above.

### Scenario 2 – Copy files from one folder to another in the document library

The following command will copy all files from the “C:\Docs\Test” folder to the “Policies” folder in the document library Shared Documents. As the document library in this example has tight versioning and approval controls, the command will also check in and approve the files once uploaded:

**Copy-FiletoSP -LocalPath "C:\Docs\Test" -SiteUrl "http://portal/team" -Library "Shared Documents" -LibraryStartFolder "/Policies" -Approve -CheckIn**

Based on the C:\Docs structure defined above, the Policies folder in the document library will contain just “Test Document 4.docx”.

### Scenario 3 – Flatten the original file structure and copy all files into the root of the document library

The following command will copy all files from “C:\Docs” and all subfolders to the root folder of the “Shared Documents” document library. The folder structure will not be replicated:

**Copy-FiletoSP -LocalPath "C:\Docs" -SiteUrl "http://portal/team" -Library "Shared Documents" -IncludeSubFolders -FlattenStructure**

Based on the C:\Docs structure defined above, the document library root will contain all five documents.

## Metadata Manifest XML File

An optional XML-based manifest file can be used with the Copy-FilestoSP command to apply column values (metadata) to files during upload. The format of the XML file is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<Columns>
  <Column name="ColumnName" type="ColumnType">
    <Values>
      <Value>ColumnValue</Value>
    </Values>
  </Column>
</Columns>
```

The following table lists the column types supported and how to reference them in the XML file. **Please note: All columns must be created and assigned to the document library *before* using the XML manifest file – running the Copy-FilestoSP command will *not* create the columns for you:**

Friendly name	Column type in XML file	Further information
Single line of text	Text	
Multiple lines of text	Note	
Choice – Single value	Choice	e.g., Drop-Down Menu or Radio Buttons
Choice – Multiple value	MultiChoice	e.g., Checkboxes
Number	Number	
Currency	Currency	
Yes/No	Boolean	
Person or Group – Single value	User	
Person or Group – Multiple value	UserMulti	
Hyperlink or Picture	URL	Requires a <Description> and <Value> key in the XML file to define the display name and URL of the link respectively
Managed Metadata – Single value	TaxonomyFieldType	The script will attempt to find the term specified in the XML file from the term set associated with the Managed Metadata column. If it fails to find a term with the name specified, it will create a new term for you in the term set.
Managed Metadata – Multiple value	TaxonomyFieldTypeMulti	The script will attempt to find the terms specified in the XML file from the term set associated with the Managed Metadata column. If it fails to find a term with the name specified, it will create a new term for you in the term set.

## Example:

The XML file used in this example will set values for every column type listed in the table above. When used with the Copy-FilestoSP command, the values specified in the manifest XML file will be assigned to every document uploaded to the document library as part of that session. If you require different values to be set on different documents, you will need to run a Copy-FilestoSP command with a separate manifest XML file for each set of documents that you wish to tag differently.

```
<?xml version="1.0" encoding="utf-8"?>
<Columns>
  <Column name="Test MMSingle" type="TaxonomyFieldType">
    <Values>
      <Value>SharePoint</Value>
    </Values>
  </Column>
  <Column name="Test MMMulti" type="TaxonomyFieldTypeMulti">
    <Values>
      <Value>SharePoint</Value>
      <Value>SharePoint Consultant</Value>
      <Value>SharePoint Designer</Value>
    </Values>
  </Column>
  <Column name="Test SingleText" type="Text">
    <Values>
      <Value>Line of text</Value>
    </Values>
  </Column>
  <Column name="Test MultiText" type="Note">
    <Values>
      <Value>First line of text</Value>
      <Value></Value>
      <Value>Second line of text</Value>
    </Values>
  </Column>
  <Column name="Test YesNo" type="Boolean">
    <Values>
      <Value>True</Value>
    </Values>
  </Column>
  <Column name="Test Choice" type="Choice">
    <Values>
      <Value>Choice A</Value>
    </Values>
  </Column>
  <Column name="Test MultiChoice" type="MultiChoice">
    <Values>
      <Value>Choice A</Value>
      <Value>Choice C</Value>
    </Values>
  </Column>
  <Column name="Test Number" type="Number">
    <Values>
      <Value>1.00</Value>
    </Values>
  </Column>
  <Column name="Test Currency" type="Currency">
    <Values>
      <Value>1.00</Value>
    </Values>
  </Column>
</Columns>
```

```

</Column>
<Column name="Test Hyperlink" type="URL">
  <Values>
    <Description>Get-SPScripts</Description>
    <Value>http://get-spscripts.com</Value>
  </Values>
</Column>
<Column name="Test PictureLink" type="URL">
  <Values>
    <Description>Phil Childs</Description>
    <Value>http://4.bp.blogspot.com/_3ZvpJCdZvQk/TCTDY7m2tzi/AAAAAAAAAKo/PRrCxyIVszc/S220/PhilChilds.jpg</Value>
  </Values>
</Column>
<Column name="Test PersonSingle" type="User">
  <Values>
    <Value>PACDOMAIN2\shaun.young</Value>
  </Values>
</Column>
<Column name="Test PersonMulti" type="UserMulti">
  <Values>
    <Value>PACDOMAIN2\spadmin</Value>
    <Value>PACDOMAIN2\phil.childs</Value>
    <Value>Approvers</Value>
  </Values>
</Column>
</Columns>

```


The screenshot below shows the View Properties page of a document in SharePoint, which has had its metadata set by the XML file specified in the example above:

Shared Documents - Test Document.docx

View

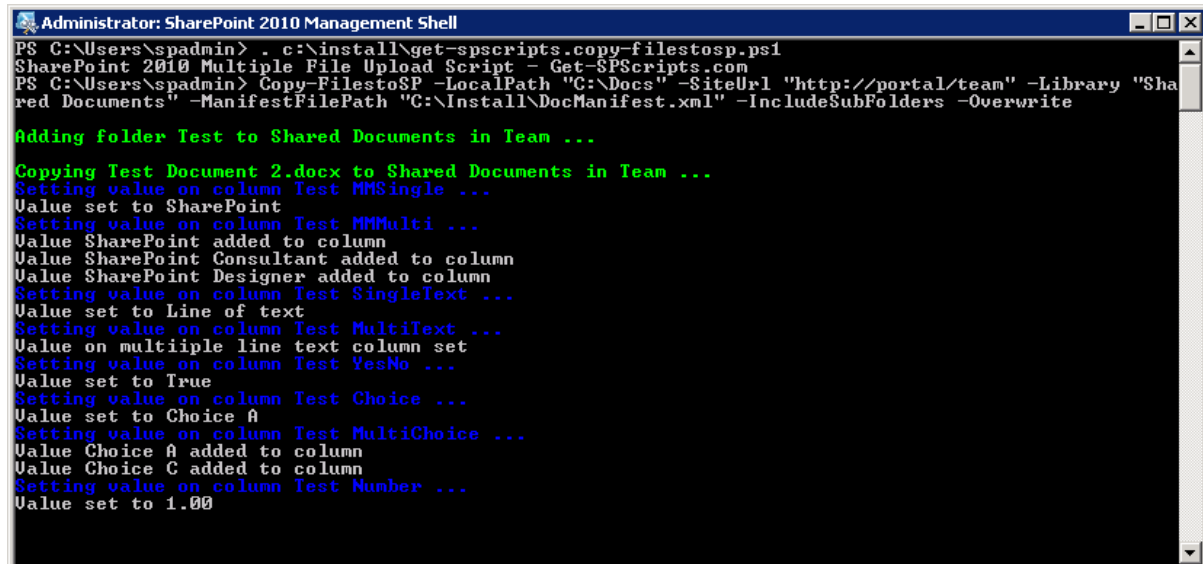
[Edit Item](#)
[Version History](#)
[Manage Permissions](#)
[Check Out](#)
[Alert Me](#)
[Manage Copies](#)
[Delete Item](#)

Manage Actions

Name	Test Document
Title	
Test MMSingle	SharePoint
Test MMMulti	SharePoint; SharePoint Consultant; SharePoint Designer
Test SingleText	Line of text
Test MultiText	First line of text Second line of text
Test YesNo	Yes
Test Choice	Choice A
Test MultiChoice	Choice A; Choice C
Test Number	1
Test Currency	£1.00
Test Hyperlink	<a href="#">Get-SPScripts</a>
Test PictureLink	
Test PersonSingle	<a href="#">Shaun Young</a>
Test PersonMulti	<a href="#">pacdomain2\spadmin</a> <a href="#">Phil Childs</a> <a href="#">Approvers</a>

Once you have defined the XML file, it can be referenced in the Copy-FilestoSP command by using the **ManifestFilePath** parameter, for example:

**Copy-FilestoSP -LocalPath "C:\Docs" -SiteUrl "http://portal/team" -Library "Shared Documents" -ManifestFilePath "C:\Install\DocManifest.xml" -IncludeSubFolders -Overwrite**



```
Administrator: SharePoint 2010 Management Shell
PS C:\Users\spadmin> . c:\install\get-spscripts.copy-filestosp.ps1
SharePoint 2010 Multiple File Upload Script - Get-SPScripts.com
PS C:\Users\spadmin> Copy-FilestoSP -LocalPath "C:\Docs" -SiteUrl "http://portal/team" -Library "Shared Documents" -ManifestFilePath "C:\Install\DocManifest.xml" -IncludeSubFolders -Overwrite

Adding folder Test to Shared Documents in Team ...

Copying Test Document 2.docx to Shared Documents in Team ...
Setting value on column Test MMSingle ...
Value set to SharePoint
Setting value on column Test MMMulti ...
Value SharePoint added to column
Value SharePoint Consultant added to column
Value SharePoint Designer added to column
Setting value on column Test SingleText ...
Value set to Line of text
Setting value on column Test MultiText ...
Value on multiple line text column set
Setting value on column Test YesNo ...
Value set to True
Setting value on column Test Choice ...
Value set to Choice A
Setting value on column Test MultiChoice ...
Value Choice A added to column
Value Choice C added to column
Setting value on column Test Number ...
Value set to 1.00
```