BRANDING TOOL

By Don Kirkham, MCPD

Twitter: @DonKirkham

This tool consists of two console applications, used to deploy and apply branding to SharePoint Online, and SharePoint 2013 On-Premise. Each application uses a custom XML file which determines where to execute the actions in the file and a list of actions to execute. These actions normally consists of uploading all of the branding assets into a site collection, and then applying a Composed Look for the site collection and sub-sites.

This document explains how to use the tool, which actions are available and how to use them.

Table of Contents

[Overall Guidance 2](#_Toc409446809)

[Executable Programs 2](#_Toc409446810)

[Custom XML File / Default XML File 3](#_Toc409446811)

[Action Nodes 5](#_Toc409446812)

[Upload Master Page 6](#_Toc409446813)

[Upload Page Layout 7](#_Toc409446814)

[Upload File 9](#_Toc409446815)

[Upload Theme 10](#_Toc409446816)

[Create Theme 11](#_Toc409446817)

[Apply Theme 12](#_Toc409446818)

# Overall Guidance

## Executable Programs

The two application executable files are named BrandingTool-Cloud.exe and BrandingTool\_OnPrem.exe by default, but may be renamed, as needed. The only differences between the applications is the method used to authenticate the user and the version of Microsoft SharePoint Client libraries used.

#### Terms

For this document, the term “App” is used to reference the version of the program you are executing.

In the Attributes sections for the node, the term **[Required]** means that the attribute must be present and cannot be blank(=""). **[Optional]** means that blank is a valid value. In most cases, removing the Optional attribute completely will result in the same performance as blank, but it is recommended to leave all of the attributes to facilitate the building of future XML files.

**Url**: Any of the attributes that contain the term Url (i.e. colorFileUrl, subWebUrl) are intended to contain a relative Url from the rootUrl contained in the site node. The rootUrl and the attribute Url will be combined to get an absolute Url to the file or web.

**Path**: Any of the attributes that contain the term Path (i.e., filePath, colorFilePath, masterFilePath) are designating a path on the local machine (where the App is being run from) to a path in the file structure where a specific file (or files, in the case of a wildcard) are located. The paths may be Absolute, UNC, or Relative.

* Absolute: Full path to a drive mapped on local machine (e.g. C:\BrandingFiles\file.txt, G:\Contoso\Branding\file.txt)
* UNC: Full UNC path to a network shared resource (e.g., [\\netshare\Branding\file.txt](file:///\\netshare\Branding\file.txt))
* Relative: Relative paths use a combination of the path to the XML file being used, the rootPath attribute in the site node, and the path listed in the attribute. The rootPath attribute may be Absolute, UNC, or Relative. If it is Absolute or UNC, the location of the XML is no longer used in the full path. If it is relative, it is appended to the local path of the XML file.

If an Absolute path is used in the attributePath, the location of the XML file and the footpath are ignored, and the path used is what is listed in the attributePath. If an Absolute path is used in the rootPath, the location of the XML file is ignored, and the path used is a combination of the rootPath and attributePath (if the attributePath is Relative.)

Examples: XML file location: C:\Branding Files

rootPath=“*site1\files*” and filePath=“*file.txt*”: Path used is *C:\Branding Files\site1\files\file.txt*

rootPath=“*G:\ContosoProject*” and filePath=“*css\file.css*”: Path used is *G:\ContosoProject\css\file.css*

rootPath=“*\\shared\ContosoProject*” and filePath=“*.\css\file.css*”: Path used is *G:\ContosoProject\css\file.css*

rootPath=“*\\shared\ContosoProject*” and filePath=“C:\css\file.css”: Path used is *C:\css\file.css*

rootPath=“” and filePath=“*file.css*”: Path used is *C:\Branding Files\file.css*

rootPath=“*..\AnotherFolder*” and filePath=“*file.css*”: Path used is *C:\AnotherFolder\file.css*

#### Program Requirements

The libraries Microsoft.SharePoint.Client.dll and Microsoft.SharePoint.Client.Runtime.dll must either be installed on the local machine (in the Global Assembly) or present in the same directory as the App.

#### App Execution

The App may be executed in several ways:

1. Double-clicking the executable file (.exe) will start the App and it will use the default XML file (see below)
2. An XML file can be dropped on the executable and the App will start, using the dropped XML file.
3. The App can be run from the command line (cmd.exe) with or without parameters. Before running the App, the directory at the command prompt will need to be changed to the directory containing the App. Running the App without parameters will use the default XML file or single parameter can be used to specify an XML file to use. The path to the XML file may be absolute of relative to the location of the App.  
    example: BrandingTool\_Cloud.exe “..\Branding Files\CloudDeployment.xml”
4. A batch file or PowerShell can be used to execute the App, using the same steps as running it from the command line.

## Custom XML File / Default XML File

The App uses a custom XML file to determine which actions are to be performed. The name of the XML file can be anything, but any name other than the default will require the XML file to be specified by either dropping the XML file on the App, or specifying its name at the command line.

The default XML file is located in the same folder as the App and uses the same name as the App, with an xml extension. For example, if the Cloud App name is not changed (BrandingTool\_Cloud.exe), then the default XML file is BrandingTool\_Cloud.xml. If the App name has been changed to ClientBrandingDeployment.exe, then the default XML file is Client BrandingDeployment.xml.

The App will show an error if no XML file is specified and it can’t find the default. It will also display what the default xml file is needed for proper execution.

The XML file must follow standard XML guidelines or the App will give an error. The XML structure needed by the App is as follows:

<?xml version="1.0" encoding="utf-8" ?>

<branding>

<credentials username="" password="" />

<site rootUrl="" rootPath="">

ACTIONS NODES GO HERE

</site>

</branding>

#### Credentials Node

Format: <credentials username="" password="" />

The username and password can be specified in the XML file. If they are omitted or left blank, the App will prompt the user for their values. These credentials will be used to authenticate to Sharepoint, so the user must have permission to perform all of the actions listed in the XML file.

Attributes

|  |  |
| --- | --- |
| Key Name | Instructions |
| username | [Optional] If omitted, App will prompt user for value. Cloud credentials must use the username@domain format. OnPrem credentials may use the username@domain or domain\username format. E.g., bob@Contoso.sharepoint.com, bob@Contoso, Contoso\bob |
| password | [Optional] If omitted, App will prompt user for value. In the XML file the value is stored as clear text, so safeguard as needed. |

#### Site Node

Format: <site rootUrl="" rootPath="" /></site>

The site node designate the root site where the actions will be applied. Credentials provided via the credentials node or when prompted by the App will need permission on the site to perform all actions in the site node.

Multiple site nodes may be used in the XML file, using standard XML syntax. For Example:

<?xml version="1.0" encoding="utf-8" ?>

<branding>

<credentials username="" password="" />

<site rootUrl="Site1" rootPath="">

ACTIONS NODES GO HERE

</site>

<site url="Site2" rootPath="">

ACTIONS NODES GO HERE

</site>

<site url="Site3" rootPath="">

ACTIONS NODES GO HERE

</site>

</branding>

Attributes

|  |  |
| --- | --- |
| Key Name | Instructions |
| rootUrl | [Optional] If omitted, the App will prompt user for value. |
| rootPath | [Optional] Use to set a default root path for all Path attributes listed in the action nodes. See Terms section above for full explanation of how Path attributes are used. |

# Action Nodes

Action nodes contain all of the deployment and application actions for a specific site (and possibly sub-sites.) The action nodes are executed in order, so any references to a file in the Sharepoint infrastructure requires that the referenced file is already present in SharePoint, whether that is an out-of-the-box (OOTB) element or part of a custom branding solution.

As needs develop, more action node types will be added to the project.

## Upload Master Page

The Upload Master Page action is used to deploy Master pages and Master page preview files. The action can handle legacy .master files, as well as HTML master page files. Once deployed, the master page is assigned the appropriate Content Type, which in the case of the HTML master page, creates the linked .master files. All files in the node are deployed to the Master Page Gallery.

Syntax

<uploadMasterPage folder="" masterFilePath="" previewFilePath="" title="" description="" />

Attributes

|  |  |
| --- | --- |
| Key Name | Instructions |
| folder | [Optional] If left blank, the master page and/or preview file are deployed to the root folder of the Master Page Gallery. If this attribute has a value, files are deployed to a subfolder named for the value. |
| masterFilePath | [Required] A path to the master page file to be uploaded (see rootPath discussion above for explanation of how paths are combined to designate the actual file.) This path should end with the master page filename and either a .master extension or .html exptension. (e.g., Contoso.master, ..\master\Contoso.html, C:\files\Contoso.html, etc.) |
| previewFilePath | [Optional] A path to the master page preview file to be uploaded (see rootPath discussion above for explanation of how paths are combined to designate the actual file.) |
| title | [Optional] The SharePoint Title of the master page. If left blank, the filename will be used as the Title. |
| description | [Optional] The SharePoint description of the master page. |

Example

<uploadMasterPage folder="**Contoso**" masterFilePath="**.\masterpage\Contoso\Contoso.html**" previewFilePath="**.\masterpage\Contoso\Contoso.preview**" title="**Contoso.com Master page**" description="**Master Page used for Contoso**" />

## Upload Page Layout

The Upload Page Layout action is used to deploy Page Layouts to the Master Page gallery. Once deployed, the page layout is assigned the appropriate Content Type, which in the case of the HTML master page, creates the linked .master files. All files in the node are deployed to the Master Page Gallery.

Syntax

<uploadPageLayout folder="" filePath="" title= " description="" associatedContentTypeID="" />

Attributes

|  |  |
| --- | --- |
| Key Name | Instructions |
| folder | [Optional] If left blank, the page layout file is deployed to the root folder of the Master Page Gallery. If this attribute has a value, the file is deployed to a subfolder named for the value. |
| filePath | [Required] A path to the page layout file to be uploaded (see rootPath discussion above for explanation of how paths are combined to designate the actual file.) This path should end with the page layout filename and a .aspx extension. (e.g., ContosoPL.aspx, ..\master\ContosoPL.aspx, etc.) |
| title | [Optional] The SharePoint Title of the page layout. If left blank, the filename will be used as the Title. |
| description | [Optional] The SharePoint description of the page layout. |
| associatedContentType | [Required] The content type ID that is associated with this page layout. See below for instructions on getting the Content Type ID. |

Associated Content Type ID

The associated content type ID can be pulled from Sharepoint by going to the “\_layouts/15/ mngctype.aspx” page and clicking on the target Content Type. The Content Type ID can be pulled directly from the ctype property of the url (see highlighted section below)

http://<...>/\_layouts/15/ManageContentType.aspx?ctype=0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF39004C1F8B46085B4D22B1CDC3DE08CFFB9C0055EF50AAFF2E4BADA437E4BAE09A30F8&Source=<...>aspx

**Common Page Layout Content Types**

|  |  |
| --- | --- |
| Name | Content Type ID |
| Article Page | 0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF3900242457EFB8B24247815D688C526CD44D |
| Welcome Page | 0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF390064DEA0F50FC8C147B0B6EA0636C4A7D4 |
| Catalog Item – Reuse | 0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF3900B46186789C3140CC85BE610336E86BBB |
| Enterprise Wiki Page | 0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF39004C1F8B46085B4D22B1CDC3DE08CFFB9C |
| Error Page | 0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF3900796F542FC5E446758C697981E370458C |
| Project Page | 0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF39004C1F8B46085B4D22B1CDC3DE08CFFB9C0055EF50AAFF2E4BADA437E4BAE09A30F8 |
| Redirect Page | 0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF3900FD0E870BA06948879DBD5F9813CD8799 |

Example

<uploadPageLayout folder="**Contoso**" filePath="**.\masterpage\Contoso\Contoso\_PL1.aspx**" title="**Contoso Layout 1**" description="**Contoso layout for news pages**" associatedContentTypeID="**0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007948130EC3DB064584E219954237AF390064DEA0F50FC8C147B0B6EA0636C4A7D4**" />

# Upload File

The Upload File action allows files to be uploaded into specific locations in SharePoint. By using wildcards, an entire folder structure can be uploaded at once.

Syntax

<uploadFile filePath="**.\masterpage\Ancestry\\***" library="**[MasterPage]**" folder="**Ancestry**" />

Attributes

|  |  |
| --- | --- |
| Key Name | Instructions |
| filePath | [Required] A path to the file(s) to be uploaded (see rootPath discussion above for explanation of how paths are combined to designate the actual file(s).) For a single file, the path should end in the filename and extension of the file. For multiple files, see the Wildcard section below. |
| library | [Required] Can contain a relative Url from the site or a designated gallery. See Library section below for further explanation. |
| folder | [Optional] If blank, the file will be uploaded into the root of the gallery designated. If this attribute has a value, the file is deployed to a subfolder named for the value. |

**Wildcards**

Wildcard can be used to simplify uploading files by allowing all files in a folder to be uploaded with one action node. Below is an explanation of the options for wildcards:

\*.\* All files in the folder will be uploaded. If there are subfolders present, corresponding subfolders will be created in the destination library and all files from those subfolders will be uploaded.

\* The files in the designated path will NOT be uploaded, but all subfolders will be created and files from those folders will be uploaded.

Patterns Standard DOS file patterns can be used (i.e., \*.css, t\*.\*), but if used, only the files in the designated path that match the pattern will be uploaded, NOT subfolders.

**Libraries**

By using a site relative Url, a library can be targeted for uploaded files. By using the following designations, special galleries can be targets.

[MasterPage] Targets the Master Page Gallery / Catalog  
 [Theme] Targets the Themes Gallery / Catalog

Example

<uploadFile filePath="**.\masterpage\Contoso\\***" library="**[MasterPage]**" folder="**Contoso**" />

# Upload Theme

The Upload Theme action deploys files for color theme, background image, and fonts. It then uses those files, along with a master page that has already been uploaded to create a Composed Look. All of the files uploaded in this action will be placed in the root folder of the Themes gallery (\_catalogs/theme/15). If other locations for the theme files are desired, use the Upload Files action, and then the Create Theme action, pointing the Urls to the locations used in the Upload File action.

Syntax

<uploadTheme themeName="" masterPageName="" colorFilePath="" backgroundImagePath="" fontFilePath="" />

Attributes

|  |  |
| --- | --- |
| Key Name | Instructions |
| themeName | [Required] Used as the name of the Composed Look |
| masterPageName | [Required] The filename (including .master extension) of an existing master page in the master page gallery. If the master page is located in a subfolder of the gallery, include that in the name (i.e., Contoso/Contoso.master) |
| colorFilePath | [Optional] A path to the color theme (.spcolor) file to be uploaded (see rootPath discussion above for explanation of how paths are combined to designate the actual file(s).) |
| backgroundImagePath | [Optional] A path to the background image file to be uploaded (see rootPath discussion above for explanation of how paths are combined to designate the actual file(s).) |
| fontFilePath | [Optional] A path to the font (.spfont) file to be uploaded (see rootPath discussion above for explanation of how paths are combined to designate the actual file(s).) |

Example

<uploadTheme themeName="**Contoso**" masterPageName="**Contoso/Contoso.master**" colorFilePath="**.\theme\Contoso.spcolor**" backgroundImagePath="" fontFilePath="" />

# Create Theme

The Create Theme action creates a new theme using Urls to existing files in the SharePoint site. (In contrast, the Upload Theme action uploads the files and then creates the theme.) While it is recommended that .spcolor and .spfont files always be placed in the Themes Gallery, the background image can be placed anywhere and this action allows you to use any image, including OOTB images in the \_layouts/15/images folder.

Syntax

<createTheme themeName="" masterPageName="" colorFileUrl="" fontFileUrl="" backgroundImageUrl="" />

Attributes

|  |  |
| --- | --- |
| Key Name | Instructions |
| themeName | [Required] Used as the name of the Composed Look |
| masterPageName | [Required] The filename (including .master extension) of an existing master page in the master page gallery. If the master page is located in a subfolder of the gallery, include that in the name (i.e., Contoso/Contoso.master) |
| colorFileUrl | [Optional] Site relative Url to a .spcolor file. If just the filename is included (i.e. Contoso.spcolor), the Themes gallery will be used. |
| fontFileUrl | [Optional] Site relative Url to a .spfont file. If just the filename is included (i.e. Contoso.spfont), the Themes gallery will be used. |
| backgroundImageUrl | [Optional] Site relative Url to any valid image file. If just the filename is included (i.e. Contoso.png), the Themes gallery will be used. |

Example

<createTheme themeName="**Contoso2**" masterPageName="**Contoso/Contoso.master**" colorFileUrl="**Contoso.spcolor**" fontFileUrl="**Contoso.spfont** " backgroundImageUrl="**/\_layouts/15/images/image\_bg005.jpg**" />

# Apply Theme

The Apply Theme action takes the named Composed Look and applies it to the target site or sub-site. It is also possible to traverse all sub-sites, applying the Composed Look.

Syntax

<applyTheme themeName="" subWebUrl="" applyToSubWebs="" />

Attributes

|  |  |
| --- | --- |
| Key Name | Instructions |
| themeName | [Required] The name of the Composed Look to be applied. |
| subWebUrl | [Optional] If left blank, the root site from the site node is used. By placing a relative Url to a sub-site, the Composed look will be applied to that target site. |
| applyToSubWebs | [Optional] If blank or false, only the target site will have the Composed Look applied. If true, all sub-sites will be also set to the designated Composed Look. See Composed Look Inheritance section below for more details. |

**Composed Look Inheritance**

By default, sites created from the Publishing template inherit the theme elements from their parent site, but may have that inheritance overridden. The Apply Theme action does not change the inheritance property, so even though the ApplyToSubWebs attribute is set to false, sub-sites with inheritance enabled will still inherit the theme and sub-site with inheritance will NOT inherit the theme.

If a sub-site is created without using the Publishing template (i.e. Team, Project, etc.), then there is no option to inherit the theme. If ApplyToSubWebs is set to true, any non-publishing sites will have the named Composed Look copied from the site collection root Composed Look Catalog (\_catalogs/design) and then applied to that sub-web.

Example

<applyTheme themeName="**Ancestry**" subWebUrl="" applyToSubWebs="**true**" />