# OFFICE AMS: Theme Management

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This sample shows how to assign, upload and change the used theme on the host web. | * Office 365 Multi Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Branding.DeployCustomThemeWeb, version 1.1 |
| Author: | Vesa Juvonen (Microsoft) |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // THIS CODE IS PROVIDED \*AS IS\* WITHOUT WARRANTY OF  // ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY  // IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR  // PURPOSE, MERCHANTABILITY, OR NON-INFRINGEMENT.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

# CHANGES

21-5-2013 Fixed Namespace bug in

# Scenario: Set theme to host web

This provider-hosted sample application for SharePoint demonstrates using the Client-Side Object Model (CSOM) make changes to a design theme or Composed Look by uploading a custom color palette and using APIs to apply them. This pattern can be used to brand remote-provisioned sites of all types.

## Deploying the ThemE

Before you can apply the theme to a web, you must first upload the necessary files. Assets files used with the theme are uploaded to the host web by using CSOM and use the FileCreationInformation object. Files are uploaded to the theme gallery in the default location (*\_catalogs/theme/15/*).

You can get instance to the catalog by using catalog ID 123 and then enumerate the files looking for the “15” folder.

// Get the path to the file which we are about to deploy

string file = sourceAddress;

List themesList = web.GetCatalog(123);

// get the theme list

web.Context.Load(themesList);

web.Context.ExecuteQuery();

Folder rootfolder = themesList.RootFolder;

web.Context.Load(rootfolder);

web.Context.Load(rootfolder.Folders);

web.Context.ExecuteQuery();

Folder folder15 = rootfolder;

foreach (Folder folder in rootfolder.Folders)

{

if (folder.Name == "15")

{

folder15 = folder;

break;

}

}

// Use CSOM to upload the file to the web

FileCreationInformation newFile = new FileCreationInformation();

newFile.Content = System.IO.File.ReadAllBytes(file);

newFile.Url = folder15.ServerRelativeUrl + "/" + System.IO.Path.GetFileName(sourceAddress);

newFile.Overwrite = true;

Microsoft.SharePoint.Client.File uploadFile = folder15.Files.Add(newFile);

web.Context.Load(uploadFile);

web.Context.ExecuteQuery();

## Creating new Composed look

New theme option to the Change look and feel UI can be provided by creating new entry to the Composite looks list located at the /\_catalogs/design URL. This list has the configured URLs for color, font and background files for each theme, including the used master page.

You can get reference to this list by using catalog ID 124 as follows:

// Let's get instance to the composite look gallery

List themesOverviewList = web.GetCatalog(124);

web.Context.Load(themesOverviewList);

web.Context.ExecuteQuery();

// Let's get instance to the composite look gallery

List themesOverviewList = web.GetCatalog(124);

web.Context.Load(themesOverviewList);

web.Context.ExecuteQuery();

// Do not add duplicate, if the theme is already there

if (!ThemeEntryExists(web, themesOverviewList, themeName))

{

// if web information is not available, load it

if (!web.IsObjectPropertyInstantiated("ServerRelativeUrl"))

{

web.Context.Load(web);

web.Context.ExecuteQuery();

}

// Let's create new theme entry. Notice that theme selection is not available from UI in personal sites, so this is just for consistency sake

ListItemCreationInformation itemInfo = new ListItemCreationInformation();

Microsoft.SharePoint.Client.ListItem item = themesOverviewList.AddItem(itemInfo);

item["Name"] = themeName;

item["Title"] = themeName;

if (!string.IsNullOrEmpty(colorFilePath))

{

item["ThemeUrl"] = URLCombine(web.ServerRelativeUrl, string.Format("/\_catalogs/theme/15/{0}", System.IO.Path.GetFileName(colorFilePath)));

}

if (!string.IsNullOrEmpty(fontFilePath))

{

item["FontSchemeUrl"] = URLCombine(web.ServerRelativeUrl, string.Format("/\_catalogs/theme/15/{0}", System.IO.Path.GetFileName(fontFilePath)));

}

if (!string.IsNullOrEmpty(backGroundPath))

{

item["ImageUrl"] = URLCombine(web.ServerRelativeUrl, string.Format("/\_catalogs/theme/15/{0}", System.IO.Path.GetFileName(backGroundPath)));

}

// we use seattle master if anythign else is not set

if (string.IsNullOrEmpty(masterPageName))

{

item["MasterPageUrl"] = URLCombine(web.ServerRelativeUrl, "/\_catalogs/masterpage/seattle.master");

}

else

{

item["MasterPageUrl"] = URLCombine(web.ServerRelativeUrl, string.Format("/\_catalogs/masterpage/{0}", Path.GetFileName(masterPageName)));

}

item["DisplayOrder"] = 11;

item.Update();

web.Context.ExecuteQuery();

## Applying the theme to host

Now that we have uploaded our theme and create the composite look, we are now able to apply the theme to the host web.

// Let's get instance to the composite look gallery

List themeList = web.GetCatalog(124);

web.Context.Load(themeList);

web.Context.ExecuteQuery();

// We are assuming that the theme exists

CamlQuery query = new CamlQuery();

string camlString = @"

<View>

<Query>

<Where>

<Eq>

<FieldRef Name='Name' />

<Value Type='Text'>{0}</Value>

</Eq>

</Where>

</Query>

</View>";

// Let's update the theme name accordingly

camlString = string.Format(camlString, themeName);

query.ViewXml = camlString;

var found = themeList.GetItems(query);

web.Context.Load(found);

web.Context.ExecuteQuery();

if (found.Count > 0)

{

Microsoft.SharePoint.Client.ListItem themeEntry = found[0];

//Set the properties for applying custom theme which was jus uplaoded

string spColorURL = null;

if (themeEntry["ThemeUrl"] != null && themeEntry["ThemeUrl"].ToString().Length > 0)

{

spColorURL = MakeAsRelativeUrl((themeEntry["ThemeUrl"] as FieldUrlValue).Url);

}

string spFontURL = null;

if (themeEntry["FontSchemeUrl"] != null && themeEntry["FontSchemeUrl"].ToString().Length > 0)

{

spFontURL = MakeAsRelativeUrl((themeEntry["FontSchemeUrl"] as FieldUrlValue).Url);

}

string backGroundImage = null;

if (themeEntry["ImageUrl"] != null && themeEntry["ImageUrl"].ToString().Length > 0)

{

backGroundImage = MakeAsRelativeUrl((themeEntry["ImageUrl"] as FieldUrlValue).Url);

}

// Set theme to host web

web.ApplyTheme(spColorURL,

spFontURL,

backGroundImage,

false);

// Let's also update master page, if needed

if (themeEntry["MasterPageUrl"] != null && themeEntry["MasterPageUrl"].ToString().Length > 0)

{

web.MasterUrl = MakeAsRelativeUrl((themeEntry["MasterPageUrl"] as FieldUrlValue).Url);

}

// Execute the needed code

web.Context.ExecuteQuery();