# Office AMS: Information Management

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This scenario shows how you can refresh the branding of existing sites collections and sites | * Office 365 Multi-Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Branding.Refresh, version 1.0 |
| Author: | Bert Jansen (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.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

# General comments

This scenario uses the Office AMS core library to iterate over existing sites and their sub sites with a purpose to verify and update the applied branding. The sample shows how to upgrade the site branding, but the same concept can be used to for example deploy a new library to a list of sites or to upgrade a custom action that was deployed at provisioning time, or…any operation that you need to move your existing sites to a newer version.

# Step1: Getting the sites you want to operate on

First you need to acquire a list of sites and/or sub sites that you want to perform changes for. The sample shows how to do this via search, but other options to fetch this list can be reading from a site directory or providing a management UI where admins can specify the list of sites to operate against.

## Using Search to get a list of sites

// Get a list of sites: search is one way to obtain this list, alternative can be a site directory

List<SiteEntity> sites = cc.Web.SiteSearchScopedByUrl("https://bertonline.sharepoint.com");

// Generic settings (apply changes on all webs or just root web

bool applyChangesToAllWebs = true;

// Optionally further refine the list of returned site collections

var filteredSites = from p in sites

where p.Url.Contains("13003")

select p;

List<SiteEntity> sitesAndSubSites = new List<SiteEntity>();

if (applyChangesToAllWebs)

{

// we want to update all webs, so the list of sites is extended with all sub sites

foreach (SiteEntity site in filteredSites)

{

sitesAndSubSites.Add(new SiteEntity() { Url = site.Url,

Title = site.Title,

Template = site.Template });

GetSubSites(cc, site.Url, ref sitesAndSubSites);

}

sites = sitesAndSubSites;

}

The call to GetSubSites is a recursive call so that the complete sub site tree is fetched.

Note:

Please be conscious of the amount of sites you’ve selected: verify you’ve selected the correct sites before continuing.

# STEP2: upgrade the branding of the existing sites

Once a site has been selected for processing you can leverage Office AMS core methods to easily manipulate the site. The sample shows how this is done for branding, but any type of change can be processed in this manner.

In order to speed up the code we’ve foreseen a pattern that leverages the web property bag to store information about the current settings. The code first reads these web property bag values and based on that actions are taken:

// Check if we've a property bag entry

string themeName = cc.Web.GetPropertyBagValueString(BRANDING\_THEME, "");

if (!String.IsNullOrEmpty(themeName))

{

// No theme property bag entry, assume no theme has been applied

if (themeName.Equals(currentThemeName, StringComparison.InvariantCultureIgnoreCase))

{

// the used theme matches to the theme we want to update

int? brandingVersion = cc.Web.GetPropertyBagValueInt(BRANDING\_VERSION, 0);

if (brandingVersion < currentBrandingVersion)

{

DeployTheme(cc, currentThemeName);

// Set the web propertybag entries

cc.Web.SetPropertyBagValue(BRANDING\_THEME, currentThemeName);

cc.Web.SetPropertyBagValue(BRANDING\_VERSION, currentBrandingVersion);

}

}

else

{

if (forceBranding)

{

DeployTheme(cc, currentThemeName);

// Set the web propertybag entries

cc.Web.SetPropertyBagValue(BRANDING\_THEME, currentThemeName);

cc.Web.SetPropertyBagValue(BRANDING\_VERSION, currentBrandingVersion);

}

}

}

The code actually update the theme is pretty straightforward and based on Office AMS Core methods:

string themeRoot = Path.Combine(AppRootPath, String.Format(@"Themes\{0}", themeName));

string spColorFile = Path.Combine(themeRoot, string.Format("{0}.spcolor", themeName));

string spFontFile = Path.Combine(themeRoot, string.Format("{0}.spfont", themeName));

string backgroundFile = Path.Combine(themeRoot, string.Format("{0}bg.jpg", themeName));

string logoFile = Path.Combine(themeRoot, string.Format("{0}logo.png", themeName));

if (IsThisASubSite(cc.Url))

{

// Retrieve the context of the root site of the site collection

using (ClientContext ccParent = new ClientContext(GetRootSite(cc.Url)))

{

ccParent.Credentials = cc.Credentials;

cc.Web.DeployThemeToSubWeb(ccParent.Web, themeName, spColorFile, spFontFile, backgroundFile, "");

cc.Web.SetThemeToSubWeb(ccParent.Web, themeName);

}

}

else

{

cc.Web.DeployThemeToWeb(themeName, spColorFile, spFontFile, backgroundFile, "");

cc.Web.SetThemeToWeb(themeName);

}