# Office AMS: Site Permissions and external users

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This sample shows an implementation of a SharePoint Taxonomy Picker control that can be used on provider hosted SharePoint apps. | * Office 365 Multi-Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Core.Permissions, version 1.0 |
| Author: | Frank Marasco, 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 sample focuses on two scenarios:

* Showing how you can easily manipulate site collection administrators using CSOM code
* Showing how to get the external sharing status and external users of a site collection or tenant

# Scenario 1: working with site collection administrators

This scenario is only using regular CSOM API’s and thus having site collection permissions on the site (e.g. your account is a site collection admin already) is needed in order to update the site collection administrators of that site. First step in the sample is creating a ClientContext object by a user with the proper permissions:

ClientContext cc = new AuthenticationManager().GetSharePointOnlineAuthenticatedContextTenant(String.Format("https://{0}.sharepoint.com/sites/{1}", tenantName, siteName), String.Format("{0}@{1}.onmicrosoft.com", userName, tenantName), password);

Using that ClientContext object you can then get a list of the current site collection administrators or update the site collection administrators as is shown in below code snippets:

List<UserEntity> admins = cc.Web.GetAdministrators();

List<UserEntity> adminsToAdd = new List<UserEntity>();

adminsToAdd.Add(new UserEntity() { LoginName = "i:0#.f|membership|user@domain" });

cc.Web.AddAdministrators(adminsToAdd);

UserEntity adminToRemove = new UserEntity() { LoginName = "i:0#.f|membership|user@domain" };

cc.Web.RemoveAdministrator(adminToRemove);

If you to set the site collection administrators for site collections where you’re not already a site collection administrator then this is possible by creating a clientcontext object using a registered app with tenant level permissions as explained below:

## ClientContext based on an OAuth token with tenant level permissions

// Use (Get-MsolCompanyInformation).ObjectID to obtain Target/Tenant realm: <guid>

//

// Manually register an app via the appregnew.aspx page and generate an App ID and

// App Secret. The App title and App domain can be a simple string like "MyApp"

//

// Update the AppID in your worker role settings

//

// Add the AppSecret in your worker role settings

//

// Manually set the permission XML for you app via the appinv.aspx page:

// 1/ Lookup your app via its AppID

// 2/ Paste the permission XML and click on create

//

// Sample permission XML:

// <AppPermissionRequests AllowAppOnlyPolicy="true">

// <AppPermissionRequest Scope="http://sharepoint/content/tenant" Right="FullControl" />

// </AppPermissionRequests>

//

// As you're granting tenant wide full control to an app the appsecret is as important

// as the password from your SharePoint administration account!

Once you’ve done that you can use below code to obtain a clientcontext object for this app:

ClientContext cc = new AuthenticationManager().GetAppOnlyAuthenticatedContext("https://tenantname-my.sharepoint.com/personal/user2", "<your tenant realm>", "<appID>", "<appsecret>");

The provided URL is for the site collection for which you do want to change the site collection administrators for. Once you’ve this client context you can use the same code as shown above, but now you can change the site collection administrators for any site collection, including the OneDrive for Business site collections that each of the Office 365 users have.

# Scenario 2: working with external sharing (Office 365 MT only)

This scenario shows how to deal with external sharing: get the external sharing status of a site collection and get a list of external users for a site collection or for the complete tenant. Since these functionalities require the tenant CSOM libraries you need to create a ClientContext against the tenant admin site collection as shown below. The user account used here needs to be a tenant administrator account.

ClientContext ccTenant = new AuthenticationManager().GetSharePointOnlineAuthenticatedContextTenant(String.Format("https://{0}-admin.sharepoint.com/", tenantName), String.Format("{0}@{1}.onmicrosoft.com", userName, tenantName), password);

Once the clientcontext is ready you can use below code to get the external sharing status and to get a list of external users.

ccTenant.Web.GetSharingCapabilitiesTenant(new Uri(String.Format("https://{0}.sharepoint.com/sites/{1}", tenantName, siteName)))

List<ExternalUserEntity> externalUsers = ccTenant.Web.GetExternalUsersForSiteTenant(new Uri(String.Format("https://{0}.sharepoint.com/sites/{1}", tenantName, siteName)));

List<ExternalUserEntity> externalUsers = ccTenant.Web.GetExternalUsersTenant();