# Office AMS: User Profile updater

|  |  |
| --- | --- |
| Summary: | Applies to: |
| This sample shows how to read user profile properties and how to update them. This code works for all users and all properties including simple type properties as complex types | * Office 365 Multi Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Core.UserProfilePropertyUpdater, version 2.0 |
| Author: | Kimmo Forss, Vesa Juvonen, 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

Updating and reading of user profile properties happens via the user profile web service (userprofileservice.asmx). If you’re using Office 365 Multi-Tenant you can launch this service from the my site host (e.g. <https://bertonline-my.sharepoint.com/_vti_bin/userprofileservice.asmx>) or from tenant administration (e.g. <https://bertonline-admin.sharepoint.com/_vti_bin/userprofileservice.asmx>). It’s important that if you want to update **all** user profile properties for **all** users that you use the user profile web service in combination with the tenant administration site. For Office 365 Dedicated or for on-premises you should provide the my site host as there’s no tenant administration sites by default. The sample is implemented in this way.

Typical use cases for this are:

* Filling custom SharePoint user profile properties with data coming from other sources (e.g. on-premises system, provisioning systems like FIM,…)
* Updating standard SharePoint user profile properties such as the SPS-MUILanguages property which determines in which language the users see their my site and other sites (assuming the correct languages have been set for the respective sites)

Important:

Use this capability with care as this allows you to override “system properties” such as workemail which contain data coming from Exchange Online in a typical Office 365 MT or D deployment.

# Scenario 1: Read and Write simple type user profile properties

This scenario shows how to read and write simple type user profile properties. To make this work you need to specify tenant administration credentials: if you specify regular user credentials you’ll not be able to update the user profile properties of other users.

## Read a simple user profile property

First you need to setup the authentication context and create an instance of the UserProfileManager

UserProfileManager upm = new UserProfileManager();

// Office 365 Multi-tenant sample

upm.User = "bert.jansen@bertonline.onmicrosoft.com";

upm.Password = GetPassWord();

upm.TenantAdminUrl = "https://bertonline-admin.sharepoint.com";

For Office 365 Multi-Tenant this sample works with the SpoCredentialAuthentication provider which uses the SharePointOnlineCredential object to authenticate against SharePoint Online. Once the UserProfileManager is instantiated reading a simple property is very simple:

string userLoginName = "i:0#.f|membership|kevinc@set1.bertonline.info";

upm.GetPropertyForUser<String>("AboutMe", userLoginName);

When you’re testing against SharePoint 2013 on-premises or Office 365 Dedicated you’ll need to use the below setup:

UserProfileManager upm = new UserProfileManager();

string userLoginName = @"SET1\KevinC";

upm.User = "administrator";

upm.Password = GetPassWord();

upm.Domain = "SET1";

upm.MySiteHost = "https://sp2013-my.set1.bertonline.info";

Note:

The user needs to be specified with his accountname

## Write a simple user profile property

Updating a property is as simple as reading as one can see in below sample:

string userLoginName = "i:0#.f|membership|kevinc@set1.bertonline.info";

upm.SetPropertyForUser<String>("AboutMe", "I love using Office AMS!", userLoginName);

# Scenario 2: Read and Write complex user profile properties

This scenario shows how to read and write complex user profile properties. To make this work you need to specify tenant administration credentials: if you specify regular user credentials you’ll not be able to update the user profile properties of other users.

## Read a complex user profile property

The user profile web service returns a PropertyData object that is used to hold simple and complex types. In case you cannot get the data via the above simple type methods you can always resort returning the PropertyData type:

string userLoginName = "i:0#.f|membership|kevinc@set1.bertonline.info";

UserProfileASMX.PropertyData p = upm.GetPropertyForUser("SPS-MUILanguages", userLoginName);

Console.WriteLine(p.Values[0].Value.ToString());

## Write a complex user profile property

The same PropertyData object can be used to also write data back as shown in below sample:

string userLoginName = "i:0#.f|membership|kevinc@set1.bertonline.info";

UserProfileASMX.PropertyData[] pMui = new UserProfileASMX.PropertyData[1];

pMui[0] = new UserProfileASMX.PropertyData();

pMui[0].Name = "SPS-MUILanguages";

pMui[0].Values = new UserProfileASMX.ValueData[1];

pMui[0].Values[0] = new UserProfileASMX.ValueData();

pMui[0].Values[0].Value = "nl-BE,en-US";

pMui[0].IsValueChanged = true;

upm.SetPropertyForUser("SPS-MUILanguages", pMui, userLoginName);