SkyDrive - How to upload content on Windows Phone

This article explains how to upload files to SkyDrive cloud storage from Windows Phone, using Live Connect

Introduction

SkyDrive is a Microsoft cloud service that allows you to access your files and documents on all your phones and computers, and to share them easily with your friends and colleagues. The service is available as an extension to the native file browser on many platforms (e.g. Windows, Mac, iOS, Android) and is also accessible from most web browsers. In addition, it is possible to access the service directly in your apps using the set of controls and APIs in the Live Connect SDK (these allow developers to integrate with SkyDrive, Hotmail, and Messenger).

This article explains how to obtain the LiveConnect SDK, add the APIs to your project, login to the service, and upload a file. The code works on Windows Phone 8. To use it on Windows Phone 7.x you need some extra jobs described here.

Prerequisites

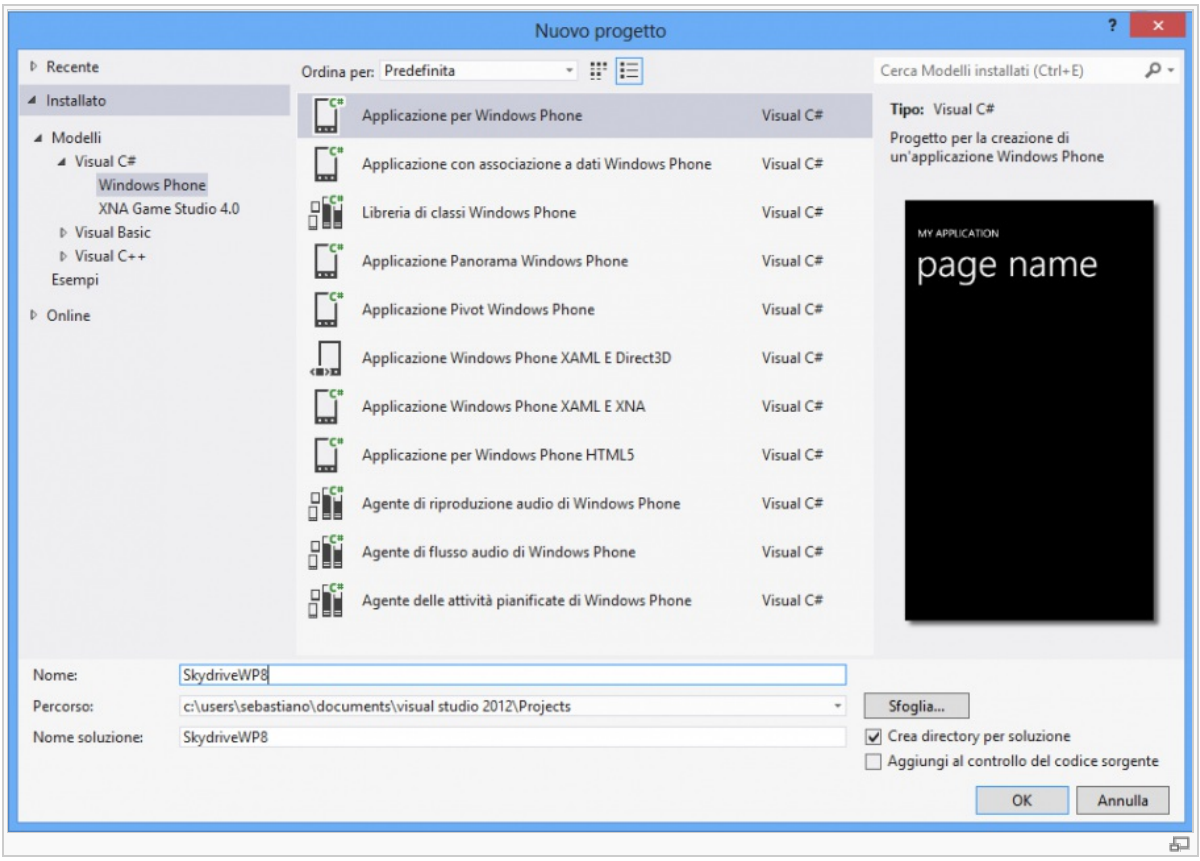
At the time of writing this article, the LiveConnect APIs are not included by default on the Windows Phone SDK.

Before you start developing:

- Download lastest Windows Phone SDK
- Download Live Connect SDK

Creating the application

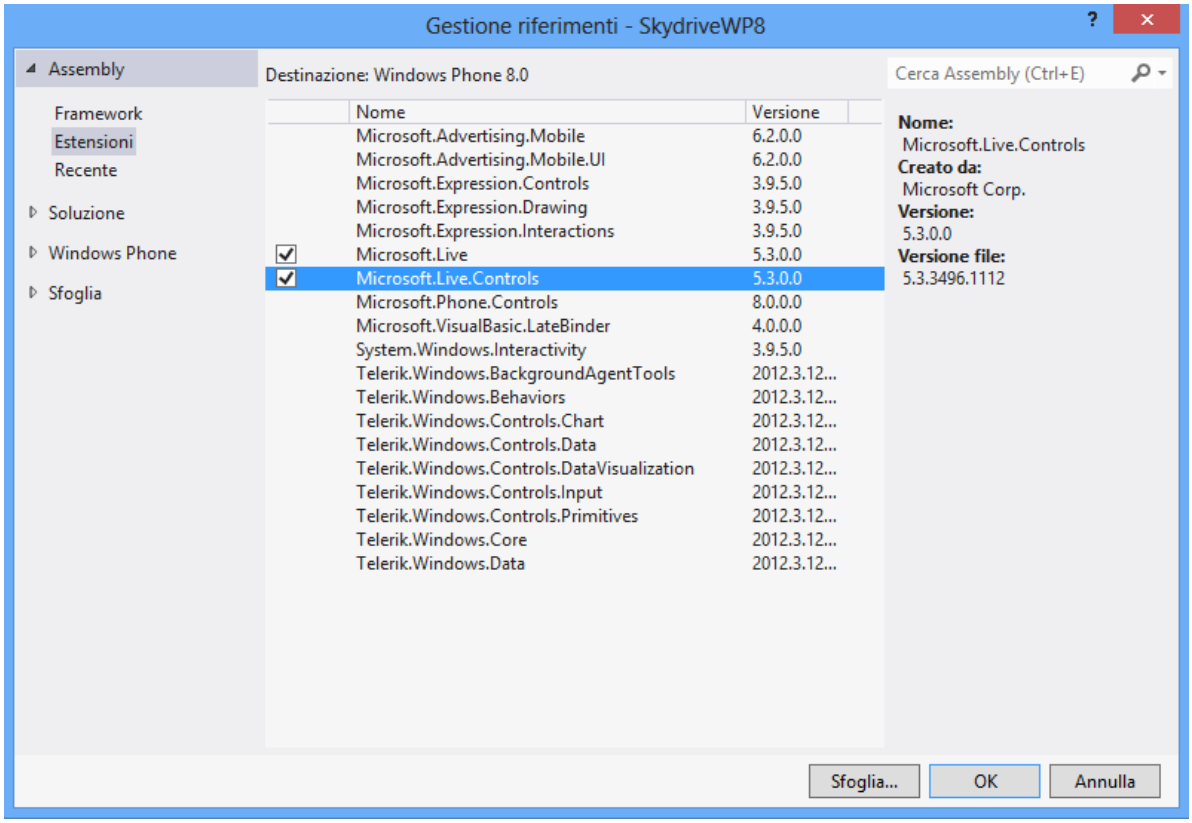
Create a new HelloWorldPhone Windows Phone application in Microsoft Visual Studio 2010 Express for Windows Phone using the **Visual C# -> Silverlight for Windows Phone -> Windows Phone Application project** template and name it *SkyDrive*:



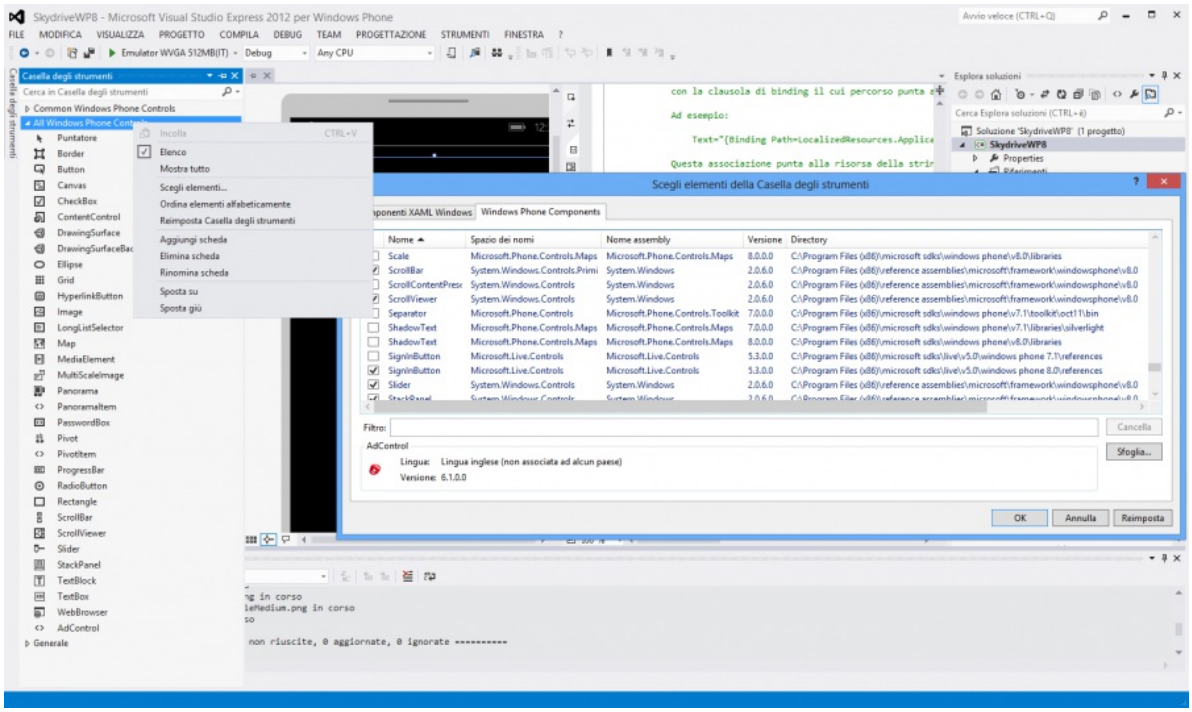
Choose last OS version

(C) Copyright Nokia 2014. All rights reserved.

Bring up the **Add Reference** dialog and go to the .Net tab. Scroll down and add the `Microsoft.Live` and `Microsoft.Live.Controls` assemblies to your project.

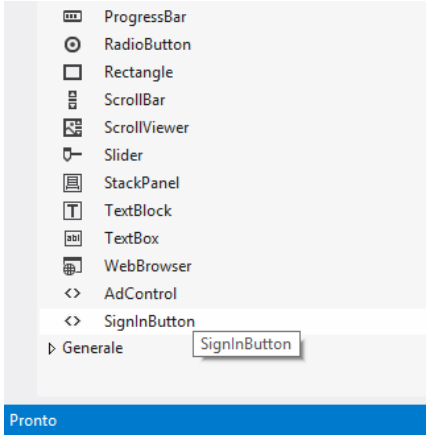


Open **MainPage.xaml** in the designer. Bring the Toolbox in view and right-click to select Choose Items… Under Windows Phone Components tab, find the SignInButton control in the Microsoft.Live.Control namespace and add to your toolbox (You only need to do this once).



If everything is well installed you should see the `SignInButton` control in your toolbox. Drag it onto your designer surface.

(C) Copyright Nokia 2014. All rights reserved.



Scopes and permissions

Before your app makes requests of the Live Connect APIs to work with Live Connect info, in most cases you must get permission from the user to access that info or to create new objects on behalf of the user. In the Live Connect APIs, this permission is called a scope. Each scope grants a different permission level.

There are three types of scopes:

- Core scopes are central to the Live Connect APIs and involve users' core profile and contact data.
- Extended scopes allow you to work with users' extended profile and contact data.
- Developer scopes allow you to work with developers' client IDs.

For our sample project we need the following scopes

Core scopes

Scope	Enables
wl.basic	Read access to a user's basic profile info. Also enables read access to a user's list of contacts.
wl.offline_access	The ability of an app to read and update a user's info at any time. Without this scope, an app can access the user's info only while the user is signed in to Live Connect and is using your app.
wl.signin	Single sign-in behavior. With single sign-in, users who are already signed in to Live Connect are also signed in to your website.

Extended scopes

Scope	Enables
wl.skydrive_update	Read and write access to a user's files stored in SkyDrive.

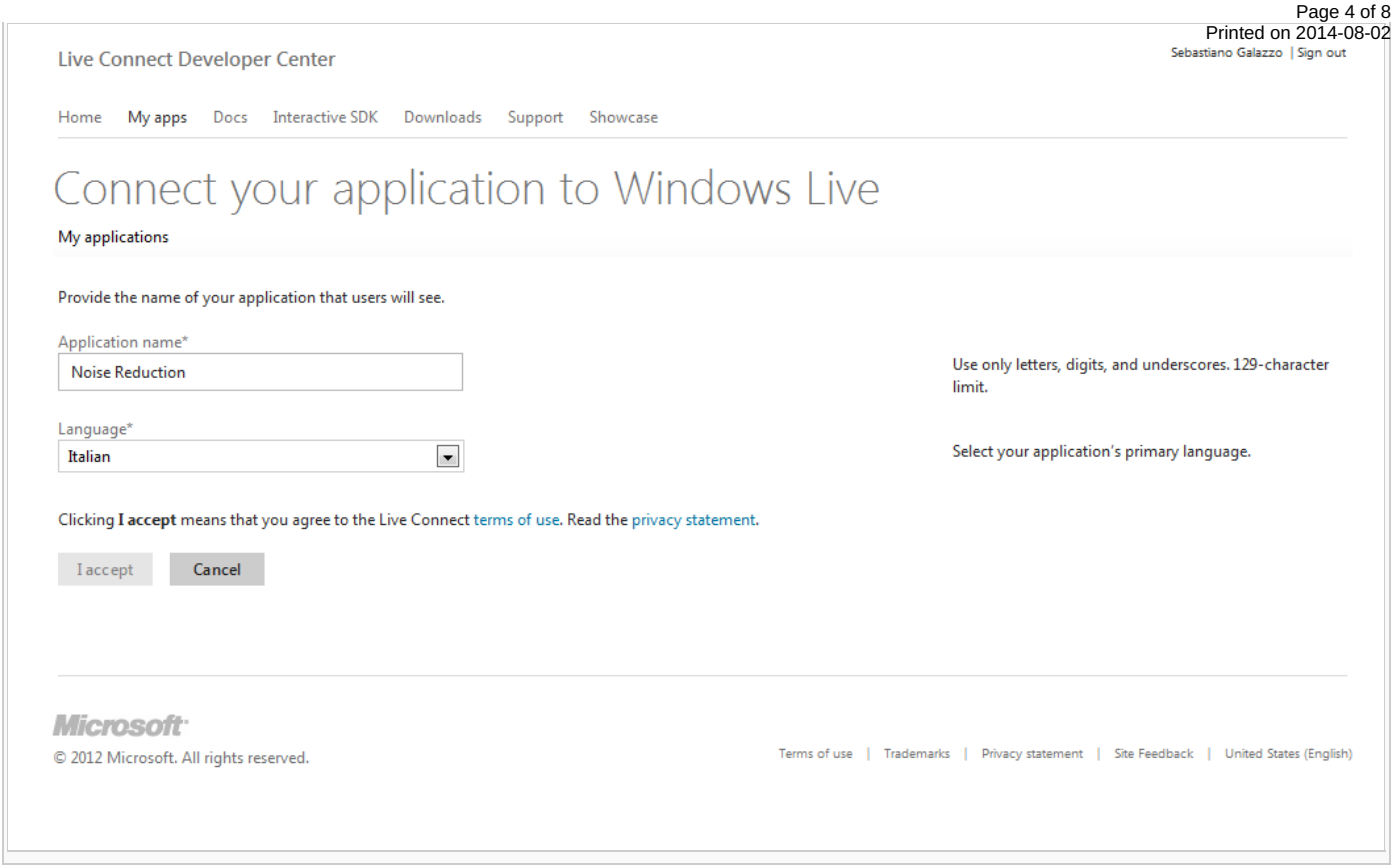
Into.xaml file your code should look like that:

```
<my:SignInButton Content="Button" Name="skydrive" Scopes="wl.basic wl.signin
wl.offline_access wl.skydrive_update" SessionChanged="skydrive_SessionChanged"  />
```

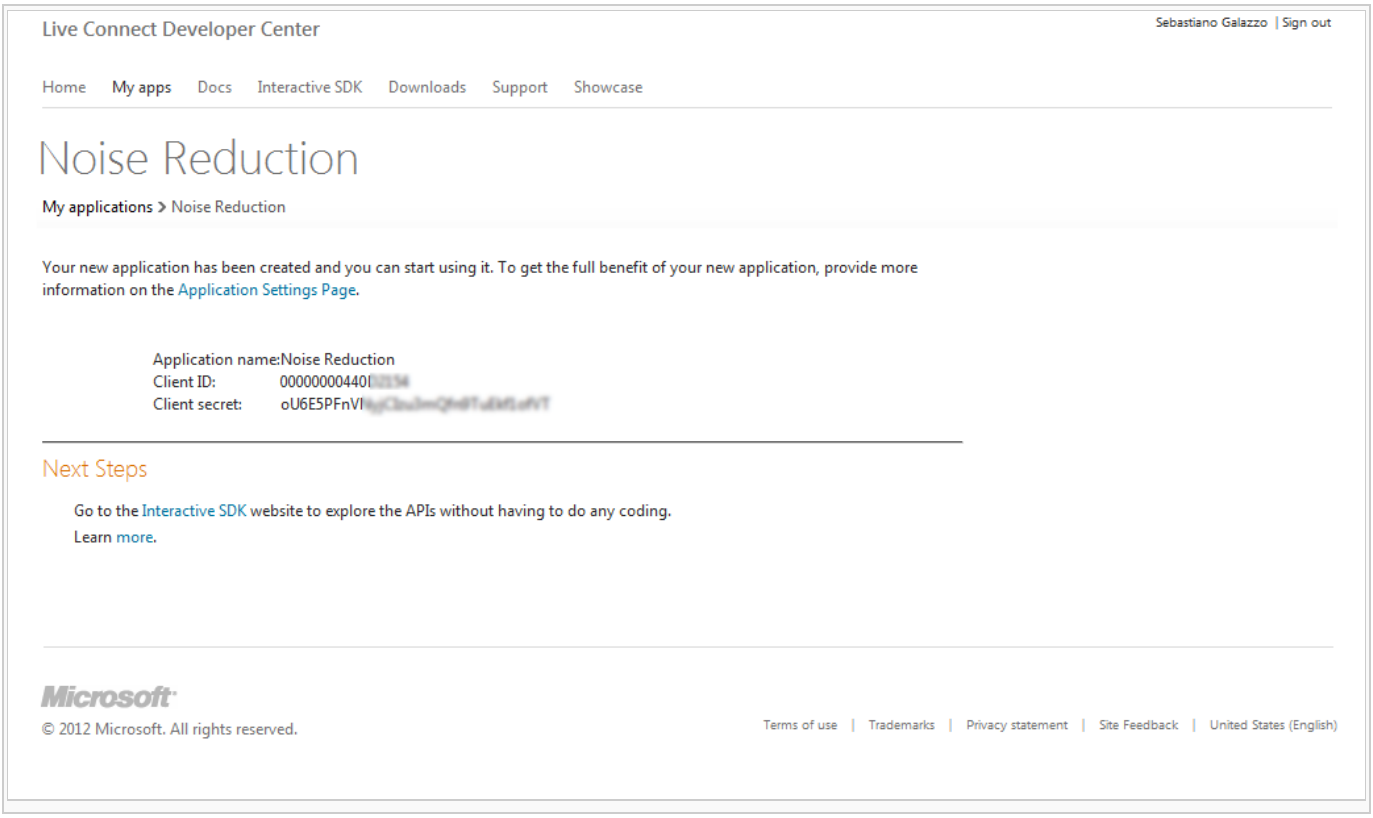
ClientId

Now we need to connect our application to Live Connect. Go to the application management site and click the Create Application. Create an application and provide a name for it.

(C) Copyright Nokia 2014. All rights reserved.



If everything goes well you will be provided with a `ClientID` and a Client secret code:



Now let's add the `ClientID` to our `SignInButton` code:

```
<my:SignInButton Content="Button" Name="skydrive" ClientId="00000000440xxxxx"
Scopes="wl.basic wl.signin wl.offline_access wl.skydrive_update" Branding="Skydrive"
TextType="SignIn" SessionChanged="skydrive_SessionChanged"  />
```

Using SkyDrive APIs

```
using Microsoft.Live;
using Microsoft.Live.Controls;
```


(C) Copyright Nokia 2014. All rights reserved.

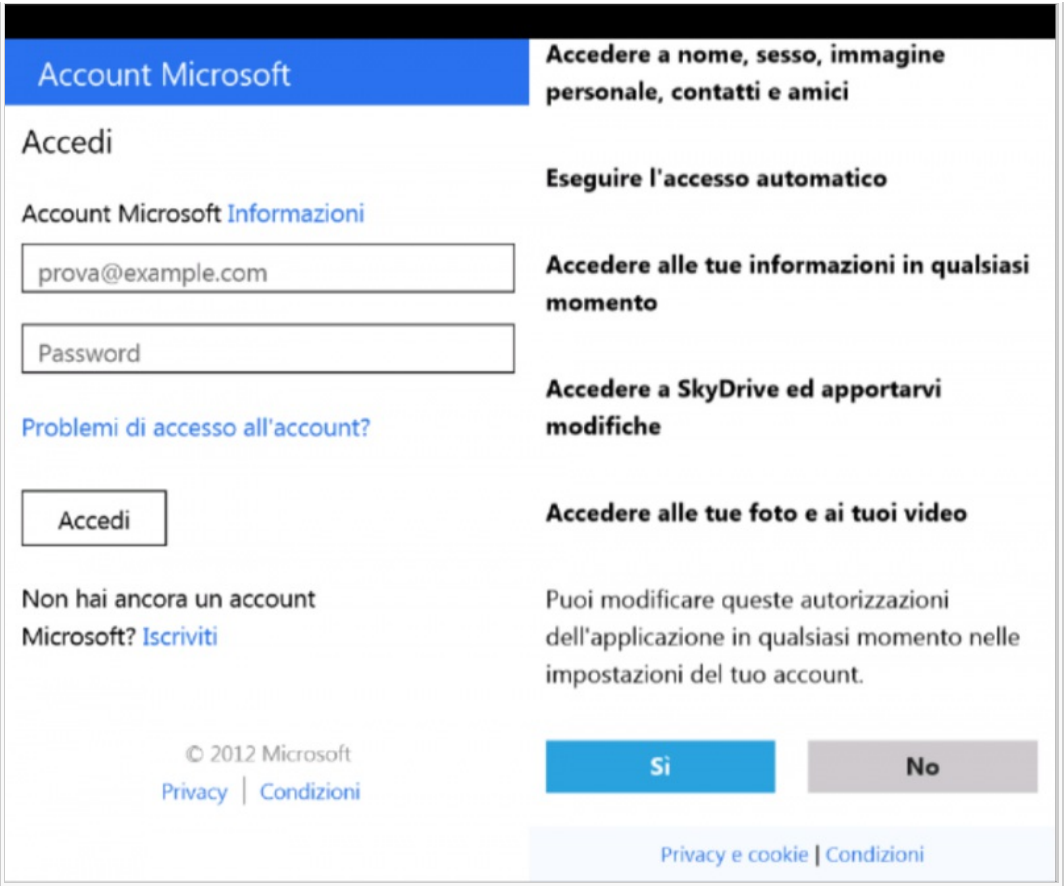
```
// SkyDrive session
private LiveConnectClient client;
```

SkyDrive Login session management

```
private void skydrive_SessionChanged(object sender, LiveConnectSessionChangedEventArgs
e)
{

            if (e != null && e.Status == LiveConnectSessionStatus.Connected)
            {
                this.client = new LiveConnectClient(e.Session);
                this.GetAccountInformations();
            }
            else
            {
                this.client = null;
                InfoText.Text = e.Error != null ? e.Error.ToString() : string.Empty;
            }

}

 private async void GetAccountInformations()
 {
            try
            {
                LiveOperationResult operationResult = await this.client.GetAsync("me");
                var jsonResult = operationResult.Result as dynamic;
                string firstName = jsonResult.first_name ?? string.Empty;
                string lastName = jsonResult.last_name ?? string.Empty;
                InfoText.Text = "Welcome " + firstName + " " + lastName;
            }
            catch (Exception e)
            {
                InfoText.Text = e.ToString();
            }
 }
```

Here the login screenshot

(C) Copyright Nokia 2014. All rights reserved.



Login screenshot

Uploading a file

First we create a Demo file to upload. Here an example of a bynary file:

```
private string fileName = "sample.dat";
private IsolatedStorageFile isf = IsolatedStorageFile.GetUserStoreForApplication();

private void CreateFileIntoIsolatedStorage()
{
            if (isf.FileExists(fileName))
            {
                isf.DeleteFile(fileName);
            }

            IsolatedStorageFileStream isfStream =   new
IsolatedStorageFileStream(fileName,  FileMode.Create,
IsolatedStorageFile.GetUserStoreForApplication());

            byte[] output = new byte[25];

            for (int i = 0; i < 25; i++)
            {
                output[i] = (byte)(i);
            }

            isfStream.Write(output, 0, output.Length);

            isfStream.Close();
}
```

In this example we call the method from constructor, but of course is possible to create the file when and where needed.

```
public MainPage()
{
            InitializeComponent();
           CreateFileIntoIsolatedStorage();
```


(C) Copyright Nokia 2014. All rights reserved.

```
 }
```

Now we can upload the file

```
private async void btnUpload_Click(object sender, RoutedEventArgs e)
{
       using (IsolatedStorageFile store =
IsolatedStorageFile.GetUserStoreForApplication())
      {
             using (var fileStream = store.OpenFile(strSaveName, FileMode.Open,
FileAccess.Read, FileShare.Read))
            {
                    try
                    {
                        LiveOperationResult res = await
client.BackgroundUploadAsync("me/skydrive",
                                                                                new
Uri("/shared/transfers/"+fileName, UriKind.Relative),

OverwriteOption.Overwrite
                                                                                );
                       InfoText.Text = "File "+fileName+" uploaded";
                    }   catch (Exception ex)  {

                    }
             }
       }
 }
```

BackgroundUploadAsync

Parameters

- path
  Type: `String`

The folder ID of the folder in SkyDrive to upload the file to, for example "folder.8c8ce076ca27823f.8C8CE076CA27823F!142"

- uploadLocation
  Type: `Uri`

The unique isolated storage path to where the file is to be uploaded from, for example "/shared/transfers/myFile2.txt"

- option
  Type: `OverwriteOption`

Specifies whether the file that's being uploaded to SkyDrive should overwrite an existing file with the same file name in that location.

Code sample

- Noise Reduction

Links

- limitations

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.