Increase your revenue with In-App Purchase on Windows Phone 8

In-App Purchase is one of the key success to monetize from your application since you could distribute your product for free to get the number of users and then gain the money after that. This article will show you how to enable In-App Purchase on your Windows Phone 8 in few easy steps

Preparation





You need an active Windows Phone Dev Center Account before implementing In-App Purchase even for testing. Multiple ways of getting an account are outlined in this article: App deployment options for Windows Phone.



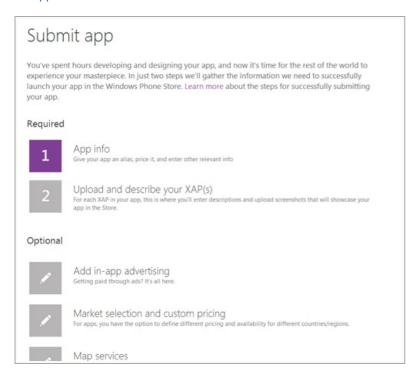
Warning: You will get an exception with the article test code if there is no complete Beta App submission for testing. Please

note that after creation of the submission it can take several hours before the required MetaData has been deployed to servers worldwide.

Submit the Windows Phone App as Beta version

First of all, you need to create new simple project for example Hello World. Compile the project to **make XAP file**. Place it somewhere on your computer.

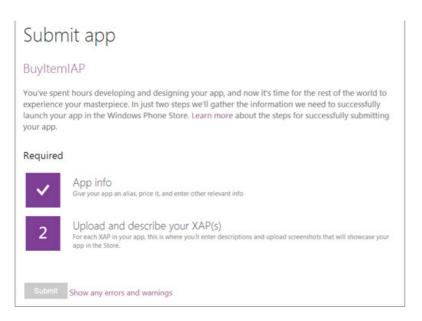
Now **go to Windows Phone Dev Center portal and Submit new App** at https://dev.windowsphone.com/en-us/AppSubmission/Hub **a**



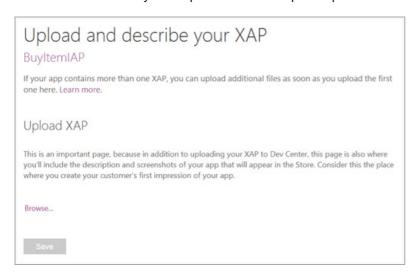
Click at **1** App Info and fill in Application's info. The important part of this page is you need to expand **More Options** and set **Distribution channels** to **Beta**. Don't forget to assign the Live email account of your beta tester in the textbox below as well. If you have more than one beta tester, you could separate each one by semi-colons.

App info BuyltemIAP The info on this page is used to refer to your app here in the Dev Center, and also controls how it appears in the Store. App info App alias* This name is used to refer to your app here on Dev Center. The name your customer sees is read directly from your XAP file. BuyltemIAP Category* games Subcategory educational Pricing Base price* Free or paid? If paid, how much? Learn how this affects pricing in different countries/regions 0.00 HKD You'll need to provide your tax or bank info (or both) if you want to submit paid apps. ☐ Offer free trials of this app. Before you select this option, make sure you've implemented a trial experience in your app. Learn more Market distribution O Distribute to all available markets at the base price tier Distribute to all markets except those with stricter content rules. Learn more. Continue distributing to current markets More options **★** Distribution channels Choosing Beta allows you to distribute your app to up to 10,000 people for testing. When you're ready to publish your app in the Public Store, you'll need to resubmit it as a new app. Enter Microsoft account email addresses for beta participants, separated by semi-colons.* Publish Automatically, as soon as the app is certified Manually publish your app at any time MPNS certificate Learn more about the benefits of authenticated push notifications and MPNS certificates. Save

Once done, click Submit and go to next step 2 Upload and describe you XAP(s)

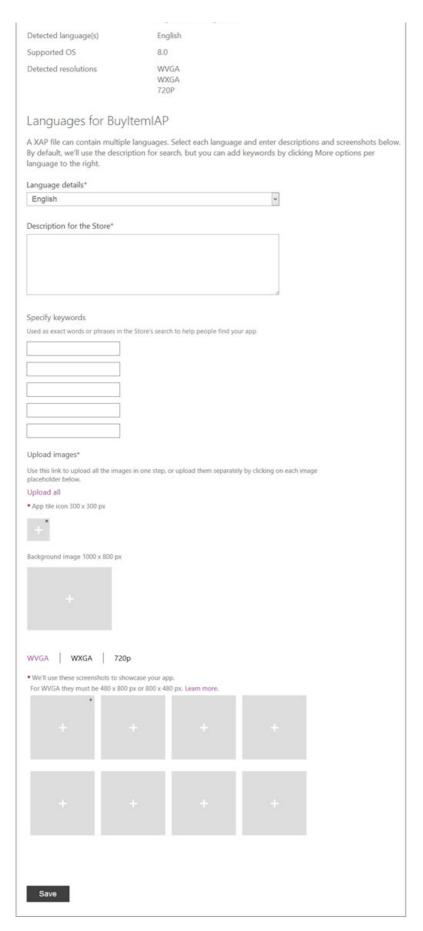


Browse for the XAP file you compiled in the first step and upload it to Store.



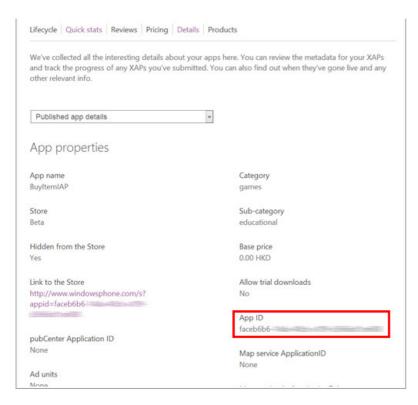
Fill in every single mandatory fill (which is not so many) and upload the App Icon (300x300 px), Background Image (1000x800px) and screenshots for every single supported screen resolution. Since this is a beta app submission, you could just submit the blank image with specific resolution. So don't worry about it.



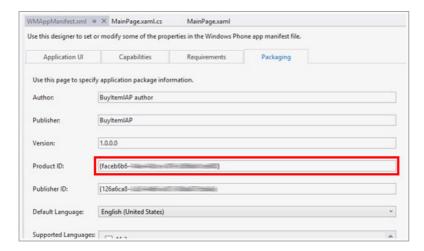


Save and Submit

Go to App Details page and note down the App ID.

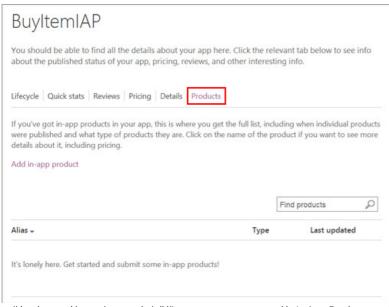


Change Product ID in WMAppManifest.xml file of your Windows Phone project to the App ID provided above.



Add In-App Product(s)

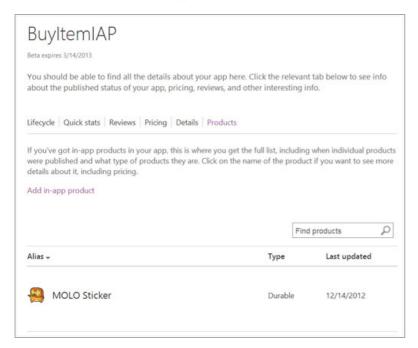
Now it's time to add In-App product associated with the application. To do that, click on **Products** link inside application info page.



Go on the instruction. Just fill in all of required fields. Please note that there are two types of In-App product, **Consumable** and **Durable**

Туре	Description
Consumable	A product that is purchased, used (consumed), and can be purchased again.
Durable	A product that is purchased and owned by the purchaser forever. It is not consumed.

Choose one suit your product best and ... done! =D Your In-App product is now ready. However please note that it might take up to 24 hours to make the item appear in the system.



Now move to coding part.

Coding Part 1: In-App product listing

Add a namespace using like this.

```
using Windows.ApplicationModel.Store;
using Store = Windows.ApplicationModel.Store;
```

To list the products associated with this application, just simply call the command.

```
ListingInformation li = await Store.CurrentApp.LoadListingInformationAsync();
foreach (string key in li.ProductListings.Keys)
{
         ProductListing pListing = li.ProductListings[key];
         System.Diagnostics.Debug.WriteLine(key);
}
```

You could get the product information from the variable pListing for example pListing. Name will present the product name entered in Windows Phone Dev Center portal, pListing. FormattedPrice present the price, etc.

To check that user has already bought the product or not yet, simply call:

```
Store.CurrentApp.LicenseInformation.ProductLicenses[key].IsActive
```

```
<ScrollViewer HorizontalAlignment="Left" Margin="12,0,12,0" Grid.Row="1">
            <ItemsControl x:Name="pics">
                <ItemsControl.ItemTemplate>
                    <DataTemplate>
                        <StackPanel>
                            <Grid>
                                <Grid.ColumnDefinitions>
                                     <ColumnDefinition Width="*"/>
                                     <ColumnDefinition Width="*"/>
                                </Grid.ColumnDefinitions>
                                <Image Margin="4" Source="{Binding imgLink}"/>
                                <StackPanel Grid.Column="1" Margin="0,30,0,0">
                                     <TextBlock Foreground="white" FontWeight="ExtraBold"
Text="{Binding Name}" />
                                    <TextBlock Foreground="white" FontWeight="Normal"
Text="{Binding Status}" />
                                    <Button Content="Buy Now" Visibility="{Binding
BuyNowButtonVisible}" Click="ButtonBuyNow_Clicked" Tag="{Binding key}" />
                                </StackPanel>
                            </Grid>
                        </StackPanel>
                    </DataTemplate>
                </ItemsControl.ItemTemplate>
            </ItemsControl>
        </ScrollViewer>
    public class ProductItem
    {
        public string imgLink { get; set; }
        public string Status { get; set; }
        public string Name { get; set; }
        public string key { get; set; }
        public System.Windows.Visibility BuyNowButtonVisible { get; set; }
    }
        public ObservableCollection<ProductItem> picItems = new
ObservableCollection<ProductItem>();
        protected override void
OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            RenderStoreItems();
            base.OnNavigatedTo(e);
        }
        private async void RenderStoreItems()
            picItems.Clear();
            try
            {
                //StoreManager mySM = new StoreManager();
                ListingInformation li = await
```

```
Store.CurrentApp.LoadListingInformationAsync();
                foreach (string key in li.ProductListings.Keys)
                {
                    ProductListing pListing = li.ProductListings[key];
                    System.Diagnostics.Debug.WriteLine(key);
                    string status =
Store.CurrentApp.LicenseInformation.ProductLicenses[key].IsActive ? "Purchased" :
pListing.FormattedPrice;
                    string imageLink = string.Empty;
                    picItems.Add(
                        new ProductItem {
                            imgLink = key.Equals("molostickerdummy") ? "/Res/41.png" :
"/Res/18.png",
                            Name = pListing.Name,
                            Status = status,
                            key = key,
                            BuyNowButtonVisible =
Store.CurrentApp.LicenseInformation.ProductLicenses[key].IsActive ?
System.Windows.Visibility.Collapsed : System.Windows.Visibility.Visible
                    );
                }
                pics.ItemsSource = picItems;
            catch (Exception e)
                System.Diagnostics.Debug.WriteLine(e.ToString());
            }
        }
```

And this is what you get as a result



Easy, right? Huh;)

Coding Part 2: Buy an item

To make a buy request, you could simply call this command

```
string receipt = await Store.CurrentApp.RequestProductPurchaseAsync(pID, false);
```

which pID represent the ProductId. And this is the complete way to call it.

```
private async void ButtonBuyNow_Clicked(object sender, RoutedEventArgs e)
{
    Button btn = sender as Button;

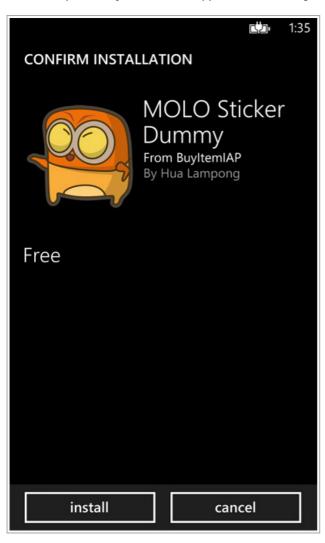
    string key = btn.Tag.ToString();

    if (!Store.CurrentApp.LicenseInformation.ProductLicenses[key].IsActive)
    {
        ListingInformation li = await
Store.CurrentApp.LoadListingInformationAsync();
        string pID = li.ProductListings[key].ProductId;

        string receipt = await Store.CurrentApp.RequestProductPurchaseAsync(pID, false);

        RenderStoreItems();
}
```

Once user press Buy Now button, application will navigate to Store like this.



Store will proceed the payment and redirect back to the application.



And as you can see, the item is now marked as Purchased!

Congratulations. Your application is now able to work with IAP ^ ^. You could apply these basic step for your product to increase the revenue gain from your Windows Phone 8 application. For more information about IAP API, please scroll down to Reference part. Valuable resources are right there.

Remark: Don't forget to change the Distribution Channels to Public Store once you finish the testing process to publish the application to Public.

Code Snippets

You can download the code for this example from File:BuyltemIAP.zip

Reference

For more information about IAP, you could find from links below

- In-App Purchase API overview for Windows Phone 8 🗗
- In-app purchase for Windows Phone 8 🗗
- In-app purchase testing for Windows Phone 8 🗗
- In-app purchase testing options for Windows Phone 8 🗗