10 Windows Phone Marketplace

We now know enough to make complete applications and games that will work correctly within the Phone environment and use the built-in features of the phone system. We are now going to take a look at how we can make sure that our programs are ready for the marketplace, then we are going to look at the submission procedure and finally we are going to consider some things we can do to make our programs stand out, and hopefully increase sales.

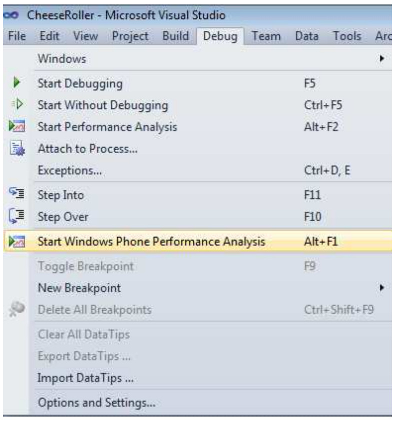
10.1 Preparing an Application for Sale

In this section we are going to focus on the things that we can do to prepare an application for sale.

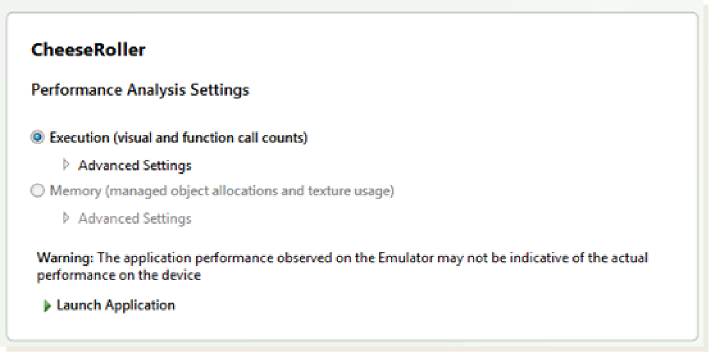
Performance Analysis

Modern computers are so powerful that they are often able to compensate for badly written software. It is frequently cost effective to buy a faster computer rather than optimise a program. However, when developing for Windows Phone we do have to worry about performance a lot more than we would for a desktop application. This is because the processor power is limited and the device itself has less memory than a PC.

The Windows Phone SDK provides a set of performance analysis tools that we can use to find out what our program is doing and how will it performs. They can give an idea of the loading our programs place on the processor, the screen update rate (how rapidly our program is updating the screen) and memory usage. We can even find out very low level detail about which particular methods are being called in our program. This is very useful as it means we can focus our optimisation on those parts of the program that are consuming most of the processor time.


CheeseRoller - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Arc
Windows
Start Debugging F5
Start Without Debugging Ctrl+F5
Start Performance Analysis Alt+F2
Attach to Process...
Exceptions... Ctrl+D, E
Step Into F11
Step Over F10
Start Windows Phone Performance Analysis Alt+F1
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Clear All DataTips
Export DataTips ...
Import DataTips ...
Options and Settings...

The analysis tools are started from the Debug menu in Visual Studio. When they are opened they display a configuration screen:



CheeseRoller

Performance Analysis Settings

● Execution (visual and function call counts)

▷ Advanced Settings

○ Memory (managed object allocations and texture usage)

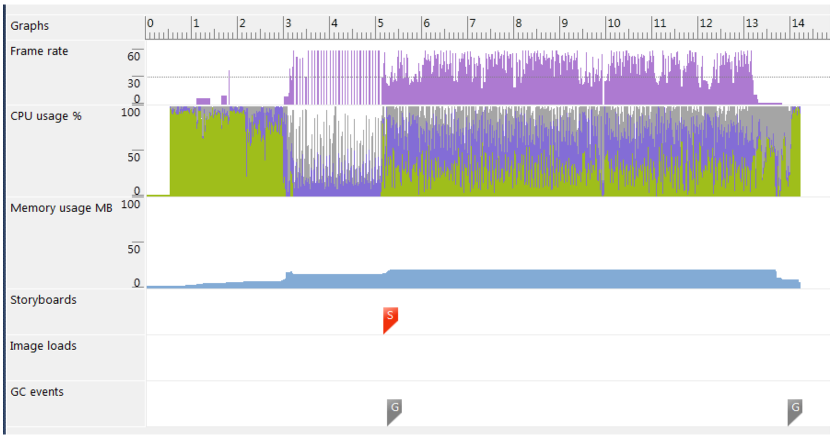
▷ Advanced Settings

Warning: The application performance observed on the Emulator may not be indicative of the actual performance on the device

▶ Launch Application

There are a number of different options we can select, depending on whether we are interested in performance or the memory usage. When we have selected our options the program will be run within the performance analysis too. The program can run inside the emulator or a hardware device. In either case the system produces a log file of the session which is stored as part of the Visual Studio Project. This makes it easy to look back at previous performance tests and determine the effect of changes that we make.

When the program is stopped at the end of a performance analysis session Visual Studio will write the log file and then allow us to explore the results.

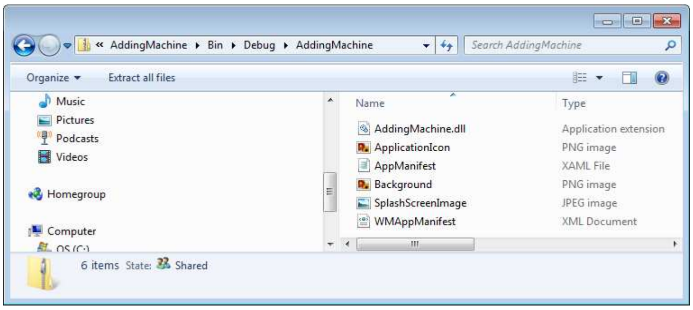


This is the log file for a Silverlight application that performs image processing. We can see that for the first few seconds the program is initialising the camera device. Then it starts to process the video stream and the CPU usage increases. The program also shows the frame rate that the application is achieving. The program itself does not use much memory, although it allocates some video buffers once the camera has initialised.

This is a very high level view of the information, it is possible to drill down and take a much more detailed view. Things to watch for are sudden jumps in CPU usage, a steady increase in memory use or a large number of Garbage Collection (GC) events. These might indicate that an application is not being very sensible with the way that it uses the phone resources.

Creating a XAP File for Application Distribution

We have already seen that an application is described by a Visual Studio solution. In Chapter 3 “Running Windows Phone Applications” we saw that an entire application is stored in a single XAP file. This is the file that is transferred into a Windows Phone device when the program is deployed. It is also the file that is submitted to the Windows Phone Marketplace when we want to sell our application.



Name	Type
AddingMachine.dll	Application extension
ApplicationIcon	PNG image
AppManifest	XAML File
Background	PNG image
SplashScreenImage	JPEG image
WMAppManifest	XML Document

6 items State: Shared

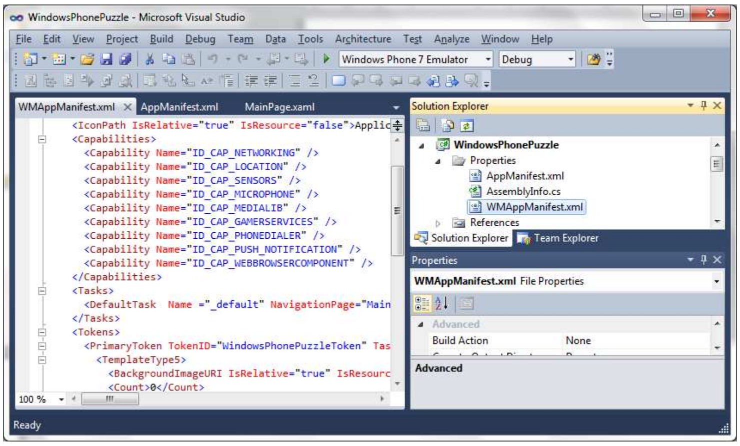
This file contains all the program assemblies and resources along with manifest files that describe the application. The `WMAppManifest.xml` file lists the capabilities of the phone that the application uses and also sets the genre of the application. This determines where on the phone the program is stored. Programs with the Game genre are stored in the game hub, applications are stored in the applications hub.

```
<?xml version="1.0" encoding="utf-8"?>

<Deployment xmlns="http://schemas.microsoft.com/windowsphone/2009/deployment"
AppPlatformVersion="7.0">
  <App xmlns="" ProductID="{b92ce770-1b22-4d52-998e-113c2784067a}" Title="PictureDisplay"
RuntimeType="Silverlight" Version="1.0.0.0" Genre="apps.normal"
Author="PictureDisplay author" Description="Sample description" Publisher="PictureDisplay">
    <IconPath IsRelative="true" IsResource="false">ApplicationIcon.png</IconPath>
    <Capabilities>
      <Capability Name="ID_CAP_GAMERSERVICES"/>
      <Capability Name="ID_CAP_IDENTITY_DEVICE"/>
      <Capability Name="ID_CAP_IDENTITY_USER"/>
      <Capability Name="ID_CAP_LOCATION"/>
      <Capability Name="ID_CAP_MEDIALIB"/>
      <Capability Name="ID_CAP_MICROPHONE"/>
      <Capability Name="ID_CAP_NETWORKING"/>
      <Capability Name="ID_CAP_PHONEDIALER"/>
      <Capability Name="ID_CAP_PUSH_NOTIFICATION"/>
      <Capability Name="ID_CAP_SENSORS"/>
      <Capability Name="ID_CAP_WEBBROWSERCOMPONENT"/>
    </Capabilities>
    <Tasks>
      <DefaultTask  Name ="_default" NavigationPage="MainPage.xaml"/>
    </Tasks>
    <Tokens>
      <PrimaryToken TokenID="PictureDisplayToken" TaskName="_default">
        <TemplateType5>
          <BackgroundImageURI IsRelative="true"
            IsResource="false">Background.png</BackgroundImageURI>
          <Count>0</Count>
          <Title>PictureDisplay</Title>
        </TemplateType5>
      </PrimaryToken>
    </Tokens>
  </App>
</Deployment>
```

Above you can see the default `WMAppManifest.xml` file that was created for the PictureDisplay application. This version of the file indicates that the application will use all of the phone capabilities. Actually the only feature that it uses is the `ID_CAP_MEDIALIB` one. When an application is submitted to the Marketplace the content of this file is checked against the calls that the application makes. If the application uses resources that are not specified in this file it will fail validation.

When a customer installs a program they are told the features of the phone that it will use and are asked to confirm this. The idea is to stop programs with unexpected behaviours being used by unsuspecting customers, for example a picture display program that also tracks the position of the user.

Before we submit this file we need to set the author, description and publisher fields.



```
    <IconPath IsRelative="true" IsResource="false">Applic
    <Capabilities>
      <Capability Name="ID_CAP_NETWORKING" />
      <Capability Name="ID_CAP_LOCATION" />
      <Capability Name="ID_CAP_SENSORS" />
      <Capability Name="ID_CAP_MICROPHONE" />
      <Capability Name="ID_CAP_MEDIALIB" />
      <Capability Name="ID_CAP_GAMERSERVICES" />
      <Capability Name="ID_CAP_PHONEDIALER" />
      <Capability Name="ID_CAP_PUSH_NOTIFICATION" />
      <Capability Name="ID_CAP_WEBBROWSERCOMPONENT" />
    </Capabilities>
    <Tasks>
      <DefaultTask  Name ="_default" NavigationPage="Main
    </Tasks>
    <Tokens>
      <PrimaryToken TokenID="WindowsPhonePuzzleToken" Tas
        <TemplateType5>
          <BackgroundImageURI IsRelative="true" IsResourc
          <Count>0</Count>
```

We can edit the manifest file from within Visual Studio. When we build the application it is incorporated into the XAP file that is produced. We can find this in the bin folder that Visual Studio creates as part of the application project.

Note that when we produce a version of the program for upload to the marketplace we need to make sure that it has been compiled in Release mode, not Debug.

Creating Application Tiles and Artwork

Before we can submit an application we also need to prepare the tiles that will denote the application on the device and we also have to prepare some artwork for display on the Windows Phone Marketplace. The icons must have particular sizes and be stored in Portable Network Graphics (PNG) files. The actual sizes that you need are:

- A small mobile app tile icon (required) which is used in the phone Windows Phone Marketplace, 99 x 99 pixels in size.
- A large mobile app tile icon (optional) which is used in the phone Windows Phone Marketplace, 173 x 173 pixels in size.
- A large PC app tile icon (required) which is used in the phone Windows Phone Marketplace, 200 x 200 pixels in size.
- Background art (optional)which is used in the Background panorama in the Marketplace entry for your application, 1000 x 800 pixels in size.

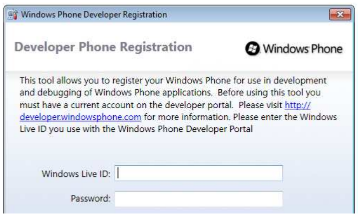
You must also create at least one screenshot which is 800x480 pixels in size. You can actually upload several screenshots, this is a very good idea as it gives potential purchasers a better idea of what your program is like.

It is a very good idea to get the help of a graphic designer to prepare the icons and background art for your program.

Testing Your Application

Unlocking Devices

A Windows Phone can normally only run programs that have been obtained via the Marketplace. However a marketplace member can unlock a phone so that it can be used to run programs that are loaded into it from Visual Studio. Paid Marketplace members can unlock up to three devices, whereas student members who joined via DreamSpark can only unlock one. The unlocking is performed using an application that is installed as part of the Windows Phone software development kit. This application can lock and unlock phones



Windows Phone Developer Registration

Developer Phone Registration

This tool allows you to register your Windows Phone for use in development and debugging of Windows Phone applications. Before using this tool you must have a current account on the developer portal. Please visit http://developer.windowsphone.com for more information. Please enter the Windows Live ID you use with the Windows Phone Developer Portal

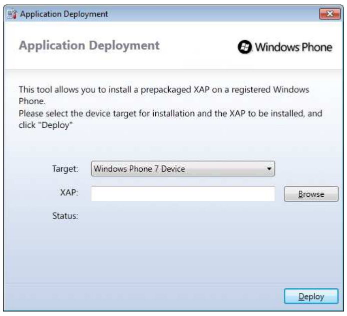
Windows Live ID:

Password:

Applications that are deployed to the phone will stay in the phone memory and can be run from the application or game hubs in the same way as ones obtained from the Marketplace. However, you can only have up to 10 of your own programs on a given device at any time.

Distributing XAP files

If you want to send a program to another developer you can send them the XAP file and they can use the Application Deployment program to send the XAP file to their unlocked device or the emulator on their Windows PC.



Application Deployment

Application Deployment

This tool allows you to install a prepackaged XAP on a registered Windows Phone.
Please select the device target for installation and the XAP to be installed, and click "Deploy"

Target: Windows Phone 7 Device

XAP:

Status:

This allows you to distribute applications for testing before you release the finished product.

Program Obfuscation

If you send someone a XAP file it is very easy for them to open this up and take a look at all things in it, including all the program code that you spend so long writing. Unfortunately it is a simple matter to open an assembly and take a look at how it works. We did this in chapter 3 using the ildasm program. This means that you should be careful when sending out programs that you don't give away all the hard work that you put into making the program. Whenever you send out a program you should take steps to make it more difficult for someone to unpick the contents.

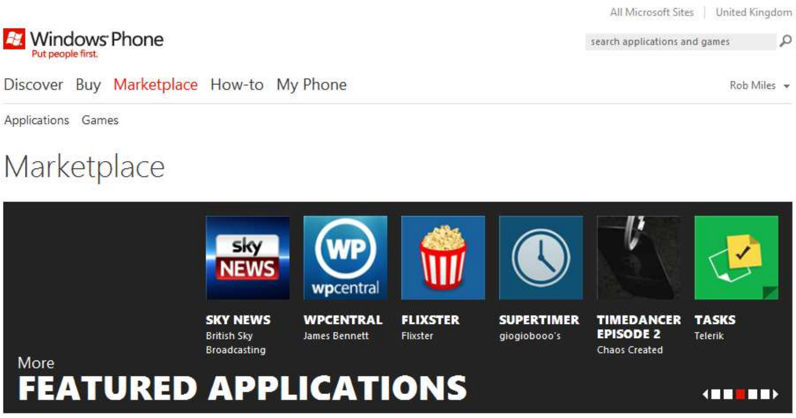
In software development this process is called *obfuscation*. It involves the use of a special tool that you use in the build process to make it very hard for someone looking at your program to deduce how it works and what it does. The tool changes variable names and adds spurious control flow to make a program very hard to understand.

There are a number of free obfuscation tools that you can use to make your program harder to understand. You could use these on your program code before you submit the XAP file. The Windows Phone developer programme also includes access to obfuscation tools from PreEmptive Solutions. These are being provided at very low cost (these systems are usually extremely expensive) and you should consider using them if you are concerned about this issue. These tools can also be used to instrument your code and allow you to find out what use is being made of the different parts of your program.

10.2 Distributing Windows Phone Applications and Games

The only way that a Windows Phone owner can get a program onto their device is by downloading it from the Windows Marketplace. Only programs that have been through the Marketplace validation process can be loaded onto a phone.

Obtaining Windows Phone Applications



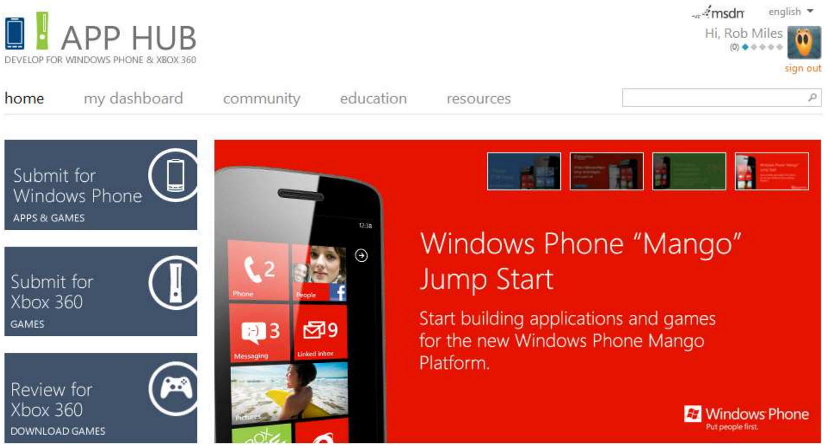
A Windows Phone owner can use their Windows Live ID to sign into their phone and gain access to services provided by Windows Live, Zune and Xbox Live along with the Marketplace.

Windows Phone owners can download applications via the Marketplace application on the device (via WIFI or the cellular network), the Zune software on the Windows PC or from the Windows Phone Marketplace tab on the Windows Phone website `www.windowsphone.com`. Very large applications cannot be downloaded via the cellular network and must be transferred using WIFI or the Zune program.

A given Windows Live ID can be used on up to five phone devices. If you remove an application from a device you can reload it later without buying it again as the Marketplace keeps a list of everything you have bought. The marketplace also provides a rating system which allows owners of applications to give ratings and feedback on them.

Creating Windows Phone Applications and Games

A registered Windows Phone developer can upload their applications and games to the Marketplace for distribution. They do this via the “App Hub” at `create.msdn.com`


msdn english
APP HUB
DEVELOP FOR WINDOWS PHONE & XBOX 360
Hi, Rob Miles
sign out
home my dashboard community education resources
Submit for Windows Phone
APPS & GAMES
Submit for Xbox 360
GAMES
Review for Xbox 360
DOWNLOAD GAMES
Windows Phone “Mango” Jump Start
Start building applications and games for the new Windows Phone Mango Platform.
Windows Phone
Put people first.

Marketplace Approval Testing

Before an application can be distributed by the Marketplace it must first go through an approval process. This makes sure that the application behaves correctly in respect of things like the Start and Back buttons and also that the program doesn’t do anything silly like grab all the memory or take twenty minutes to start running. This process also makes sure that programs that are offensive are not made generally available.

The testing is partly automatic and partly human. An engineer will run your program and ensure that it works correctly. The application will be tested with light and dark backgrounds on the phone to make sure that all the menus are visible.

If your application is not approved you will receive a written report that gives details of the parts of the program that need attention. When you resubmit the re-test will focus on the areas that were identified, not on the whole program again. Sometimes an application will be approved with some “warnings”. This means that issues have been identified that need to be addressed, but the situation is not so serious as to require that the application cannot be approved. In that situation you would be expected to release an upgraded version later.

Once an application has been approved it is listed in one of the application categories and is available for download. A developer can produce an upgraded version of an application which, once it has been through the approvals process, will be made available as an upgrade to all existing owners.

Some applications are free and others are paid. When a phone owner buys a paid application the cost can be added to their phone bill or they can register a credit card to be used to pay for purchases. Many applications have a “trial” mode which is free. This can later be upgraded to a fully featured version. When a program is running it can easily test whether it is being used in “full” or “trial” mode.

Registered developers are allowed 100 submissions of free applications per year. If each of these is approved this means that they can make 100 free applications available. If a developer wants to submit any more free applications each submission will cost them $20. A developer can submit an unlimited number of “paid” applications.

Applications can be loaded direction onto the phone “over the air” using the mobile phone operator or WiFi. They can also be loaded using the Zune software that also provides a media management for the phone as well. Marketplace membership

Before a developer can submit applications to the Marketplace they must first become a member. Membership is keyed to the Windows Live ID of the member. It costs $99 per year to be member of the marketplace. If you stop being a member of the marketplace all your applications are removed. Students who have access to the DreamSpark programme can get free membership of the Windows Phone marketplace.

Members of the marketplace have their identity validated when they join and the Marketplace gives them a unique key which is used to sign their applications. The Marketplace also retains their bank and tax details so that they can be paid for application sales.

Members of the Marketplace can set the selling price of their application and receive 70% of the cost paid by the customer. This is paid directly into their bank account once they have reached a threshold of $200 worth of sales.

Trial Mode

An application or game can be downloaded as a free demo. A program can easily determine whether or not it is operating in demo mode:

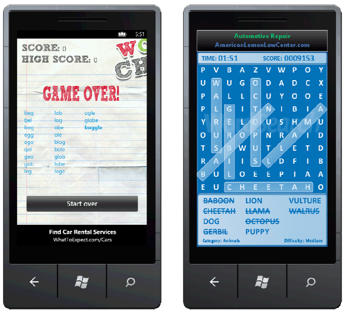
```
LicenseInformation info = new LicenseInformation();

if ( info.IsTrial() )
{
    // running in trial mode
}
```

The **`LicenseInformation`** class is in the **`Windows.Phone.Marketplace`** namespace. If the program is running in trial mode it can restrict the user actions, for example it could disable certain functions or stop the program running after a time limit. If the user wishes to upgrade the application it can use the Launcher mechanism to direct the user to the marketplace.

However, you should remember that a paid for program with a trial mode will still only be visible in the “paid” application part of Windows Phone Marketplace. This means that customers who only ever look for free applications will never see your program. Perhaps a more successful strategy might be to produce two versions of your program, a limited “free” one and a fully featured “paid” version. That way potential customers have more chance of finding and using your program.

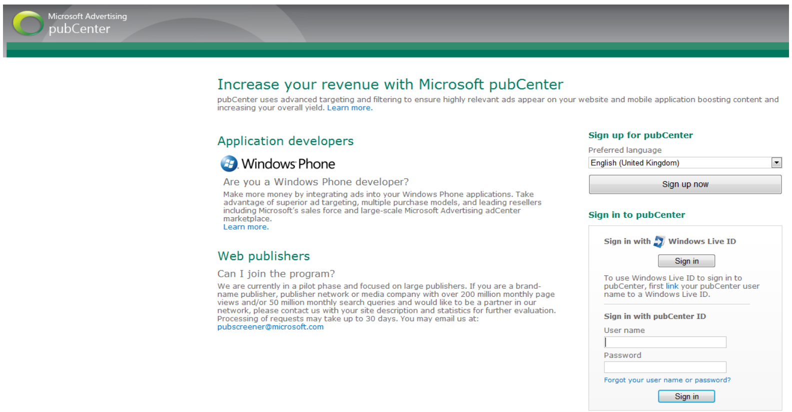
Adding Advertising

If you don’t want to sell your application, but you do want to make money from it, you can add in-application advertisements. These can be added to XNA or Silverlight programs and will display advertising content to users. The user can respond to the advertisement by linking through to the advertisers web site or placing a phone call to them. In either case you will receive 70% of the advertising revenue that is generated.



The advertisements are displayed in particular areas of the phone screen.

You can sign up for an account at the Microsoft PubCenter:



Increase your revenue with Microsoft pubCenter

pubCenter uses advanced targeting and filtering to ensure highly relevant ads appear on your website and mobile application boosting content and increasing your overall yield. Learn more.

Application developers

Windows Phone

Are you a Windows Phone developer?

Make more money by integrating ads into your Windows Phone applications. Take advantage of superior ad targeting, multiple purchase models, and leading resellers including Microsoft’s sales force and large-scale Microsoft Advertising adCenter marketplace.
Learn more.

Web publishers

Can I join the program?

We are currently in a pilot phase and focused on large publishers. If you are a brand-name publisher, publisher network or media company with over 200 million monthly page views and/or 50 million monthly search queries and would like to be a partner in our network, please contact us with your site description and statistics for further evaluation. Processing of requests may take up to 30 days. You may email us at:
pubscreener@microsoft.com

Sign up for pubCenter

Preferred language: English (United Kingdom)

Sign up now

Sign in to pubCenter

Sign in with Windows Live ID

Sign in

To use Windows Live ID to sign in to pubCenter, first link your pubCenter user name to a Windows Live ID.

Sign in with pubCenter ID

User name

Password

Forgot your user name or password?

Sign in

It is now possible to include advertising in applications in a large number of regions around the world. The Advertising SDK is part of the Windows Phone SDK and provides a set of test advert servers that you can use in XNA games and Silverlight applications.

The Submission and approval process

The submission process for programs is not hard to use. It is all managed from the developer site for Windows Phone and XNA Games:

```
http://create.msdn.com
```

Visitors to the site can download documentation and the development tools. Members of the Marketplace can use the dashboard pages to view the progress of submissions and submit new programs. There are a number of detailed walkthroughs that will take you through the membership and submission process, you should go through these to make sure you understand what is going on and what is required of you and any programs that you submit.

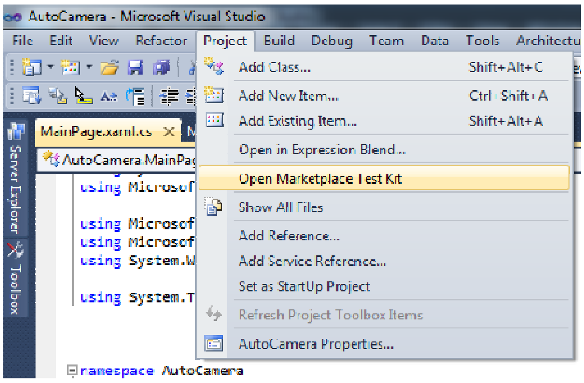
Windows Phone Application Certification Guidelines

The Windows Phone developer website provides access to a very detailed document that describes how to create applications. It is very important that you read the latest version of this document and follow the guidelines set out in it. It has been through numerous versions as the process has evolved, so you should make sure that you are using the most up to date version of the text. You can find the guidelines here:

`msdn.microsoft.com/en-us/library/hh184844.aspx`

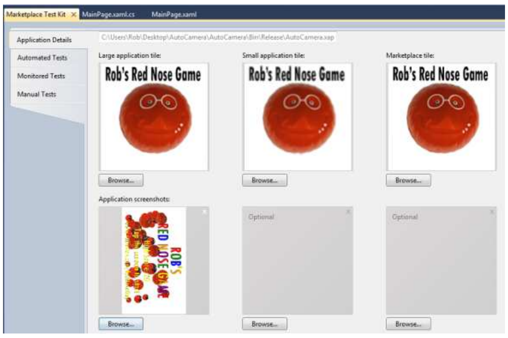
The Marketplace Test Kit

One way to make sure that an application passes the approvals process first time is to make use of the Marketplace Test Kit. This is built into the development tools and provides access to the same set of tools and procedures used by the testers. Visual Studio will check for changes to the procedures and update the Test Kit automatically so that it always reflects approvals policy.

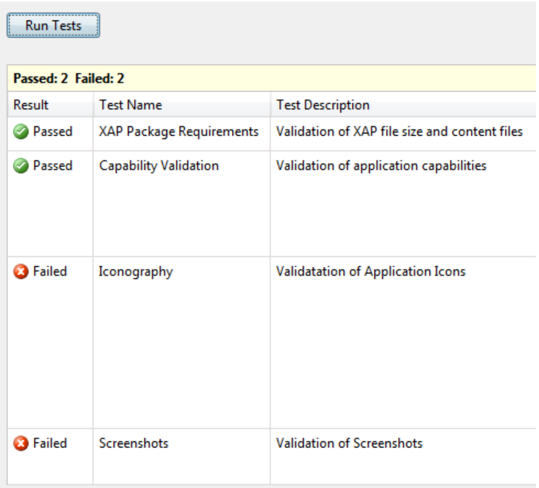


```
using Microsof
using Microsof
using Microsof
using System.W
using System.T
namespace AutoCamera
```

It can be found in the Project menu.



The tests even check for the correct file type and size of the application tiles. There are also some automated tests and the Test Kit also provides an itemised walkthrough of the tests performed by the testers so that you can perform these tests before you submit your application.



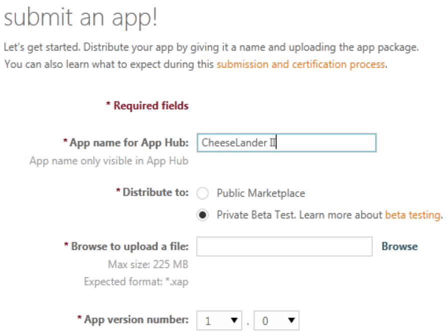
Passed: 2 Failed: 2

Result	Test Name	Test Description
✓ Passed	XAP Package Requirements	Validation of XAP file size and content files
✓ Passed	Capability Validation	Validation of application capabilities
✗ Failed	Iconography	Validatation of Application Icons
✗ Failed	Screenshots	Validation of Screenshots

This shows some tests having passed and others failed.

Private Beta Testing

If you want to get feedback from users before formally releasing your application you can create a "private" beta test release. Rather than making the application visible to all the customers on the marketplace, this form of release sends a weblink to up to 100 testers who can follow the link to a version of your program that can be downloaded onto a phone and used for up to 90 days, after which time the application is removed from their phone.



submit an app!

Let's get started. Distribute your app by giving it a name and uploading the app package.
You can also learn what to expect during this submission and certification process.

* Required fields

* App name for App Hub: CheeseLander II
App name only visible in App Hub

* Distribute to: ○ Public Marketplace
● Private Beta Test. Learn more about beta testing.

* Browse to upload a file: Browse
Max size: 225 MB
Expected format: *.xap

* App version number: 1 . 0

This option is set at the beginning of the submissions process.

10.3 Making your Application Stand Out

The Windows Phone Marketplace is now acquiring a fairly large number of applications and so if we want our programs to be the ones that are downloaded we have to do a bit of extra work to get noticed. Here are some tips that you might find useful.

Design to Sell

An attractive looking application will have a much greater chance of being noticed. The tiles, backgrounds and screenshots that are displayed on Marketplace should all be designed to help sell the application. You should also make sure that you provide lots of screenshots and also the large background graphic. Do not do the Marketplace description just before you publish the application. Make sure that you spend some time getting the words and the display just right. A really attractive application might even be picked as a specially selected one and highlighted on the Marketplace.

Target Different Localisations

While there might be more than one application like the one that you are selling, there might not be any in the German language. So, by producing a localised version of the program you could have that marketplace to yourself. There is very good localisation support available to Windows Phone developers, you can find out more here:

```
msdn.microsoft.com/en-us/hh336287
```

Use App Connect

The App Connect feature in Windows Phone makes it possible for customers to find your application even when they are searching for other things. For example a Bing search for “Knitting wool” could be linked through to your knitting pattern application. The process of linking an application in this way is free and all you have to do is add details of the categories under which your application can appear. You can find out more here:

```
msdn.microsoft.com/en-us/library/hh202969.aspx
```

Give your Program Away

If you provide a free version there is more chance that users will download it and use it. The free version could contain adverts or “nag” messages. Because it is easy to create a launcher that will take the user into the Windows Phone Marketplace you can provide a “buy upgrade” option that can target a paid version of the program or game.

Release Upgrades/Episodes

Rather than releasing a 15 level game or a fully featured program you could instead release a “starter” version and then provide upgrades at regular intervals. There are a number of reasons why this is a good idea.

- You will get to market earlier than your competition.
- Your application will appear regularly in the “new items” list on Marketplace. Since a lot of sales of an application are made in the few days after a new release, making more new releases will improve your sales.
- You can get the customers engaged in providing ideas for new content and features. This means that you can build a good relationship with them.

Change Categories

When you submit an application you will be asked to choose the category in which it is to be displayed in the Marketplace. It is often the case that one particular category doesn’t really work. A game could be thought of as a sports simulation, an arcade game or a puzzle. If you are not sure which is the best category, try them moving the game around and see how this affects sales. This will also get you exposure in other categories, which will also be beneficial.

Encourage Good Feedback

A good relationship with customers is something that you should work hard to cultivate. In the unlikely event of your application failing it should display a populated mail message and provide the user with an easy way of sending it. Then, if you receive such a message you should respond constructively. Building a relationship in this way makes customers into debuggers and advocates of your programs, which is really useful.

10.4 What To Do Next

At this point in the document you should have all the skills you need to make something that can be published on Windows Phone. Here is a list of things you can do to take the next step.

Register as a Developer

If you have not already done this you should head over to `create.msdn.com` and sign up. If you are a student you should stop off at `dreamspark.com` and get a free registration.

Get the Toolkit

The tools can be found at `create.msdn.com` and will provide you with everything you need in a single install. This includes Visual Studio, the emulator, Expression Blend, the Advertising SDK and programs to unlock your device.

Publish Something

Once you have got your development environment working you should put something out in Marketplace. It doesn't really matter whether it is something you are massively proud of, and you can always remove it later, but you never know, there might be someone out there desperate for the application that you make.

Make Suggestions

If you find something you don't like, or have an idea that could make things better, you can use the UserVoice forums to suggest things and vote on them. The Windows Phone team really do read and respond to these suggestions, and quite a few have found their way into the device. You can find the forums here: `wpdev.uservoice.com`

Resources

There are lots of good resources for phone developers.

Learning Windows Phone Development

Windows Phone Jump Start

`create.msdn.com/en-US/education/catalog/article/wp7_jump_start`

Windows Phone Code Samples

`msdn.microsoft.com/en-us/library/ff431744.aspx`

Application Features for Windows Phone

`msdn.microsoft.com/en-us/library/ff402551.aspx`

XNA

XNA Game Studio 4.0 on MSDN

`msdn.microsoft.com/en-us/library/bb200104.aspx`

XNA Game Development Resource Page

`create.msdn.com/en-us/education/gamedevelopment`

Farseer Physics Engine

`farseerphysics.codeplex.com`

Silverlight

Windows Phone Silverlight, Development Quickstarts

`create.msdn.com/en-us/education/quickstarts`

Royalty Free Icons

`thenounproject.com`

The Silverlight Toolkit

`silverlight.codeplex.com`

Design toolbox

`www.microsoft.com/design/toolbox`

Windows Phone Azure Toolkit

`watoolkitwp7.codeplex.com`

Program Ideas

We now know how to make programs for Windows Phone. We can create multi-page Silverlight interfaces which match those of the phone itself and we have also had a look at how to create games using XNA. We also know how to create connected applications that make use of data hosted on the internet. We have discovered some of the difficulties encountered by programmers when they write code for small, resource constrained, devices and we now also know how the Windows Phone operating system allows us to create useable applications in spite of these issues. Finally we know how to use the underlying Windows Phone system so that our applications can make use of features in the phone itself.

At this point we have a superb set of tools, next we have to find a problem to solve with them.

Application Ideas

Applications ideas do not always appear fully formed. Quite often you will start with a small idea for a solution and then add features or discover situations where the solution could be made even more useful. This is not a bad way to come up with something original, but you must beware of adding too many feature ideas before you have made something work, otherwise the whole thing might collapse under its own weight before you have built anything.

You can get good ideas for applications by talking to lots of people and trying to find out what would be useful to them. Not everyone is aware of just how much you can do with modern devices and so if you say that you can make something mobile that can track position, take pictures and make use of internet services they might have an idea of a situation where some of that power would be useful to them.

Game Ideas

In some ways game ideas are easier to come by than application ideas. Systems like XNA let you “doodle” with game code to find out what it does. I’ve already mentioned the importance of playtesting. This is where you give your game to other people to find out if the game is playable. Playtesting can also throw up ideas for game development. Often a playtester will suggest changes to the program that will make it more interesting and fun. Don’t be afraid to do silly things in a game (add 10 times as many aliens, invert gravity, make everything bigger/smaller etc etc) and see what happens to the play experience.

The great thing about a game is that rather than solving a specific problem, as with an application, a game just has to be fun to play.

Fun with Windows Phone

Windows Phone provides an amount of processing power that was unavailable in the world a few years ago, let alone in everyone’s pocket. It also provides an amazing level of connectivity, high performance graphics and devices such as cameras and location sensors that make it possible to build truly novel applications.

Personally I reckon that just being able to program for this device, let along sell the results, is inspiring enough. I hope you have learned enough from these notes to get yourself started on the fun you can have with this platform.

Rob Miles
October 2011

What We Have Learned

1. Windows Phone owners get their applications from the Windows Phone Marketplace.
2. Registered developers can submit applications for approval and distribution via the Marketplace.
3. Developers register and track the progress of their applications via the Windows Phone developer website.
4. Registering as a developer costs $99 a year, students can register for free via the Dreamspark programme.
5. Developers can produce free or paid applications, but are limited to submitting 100 free applications per year. Further submissions of free applications cost $20 each.
6. A registered developer can unlock phones so that they can run programs compiled in Visual Studio.
7. Applications are distributed as XAP files which contain all the resources and content along with a manifest file that describes the content and gives the phone setting information about the application.
8. XAP files can be loaded directly onto unlocked devices by developers.
9. Programmers should obfuscate their programs so that it is not easy for anyone obtaining a XAP file to find out how the code works. There is an obfuscation processor available via the Windows Phone developer web site.
10. A developer must read the Windows Phone Certification guidelines before submitting programs to the approval process.