

# XNA Tutorials

Utah State University  
Association for Computing Machinery  
XNA Special Interest Group  
RB Whitaker  
21 December 2007

## Console Windows

Supplementary Tutorial 8

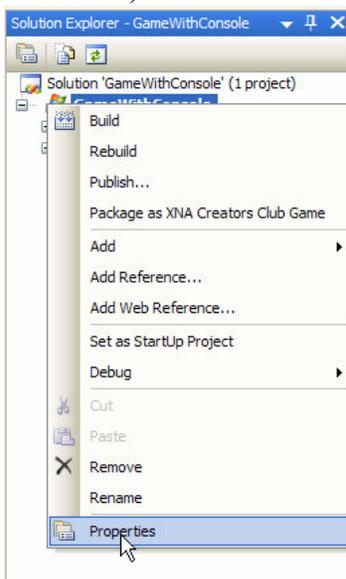
### Overview

The majority of beginning game programmers are already quite familiar with console windows. In fact, many of you probably have had *only* a console window for many of your programs before you started using XNA.

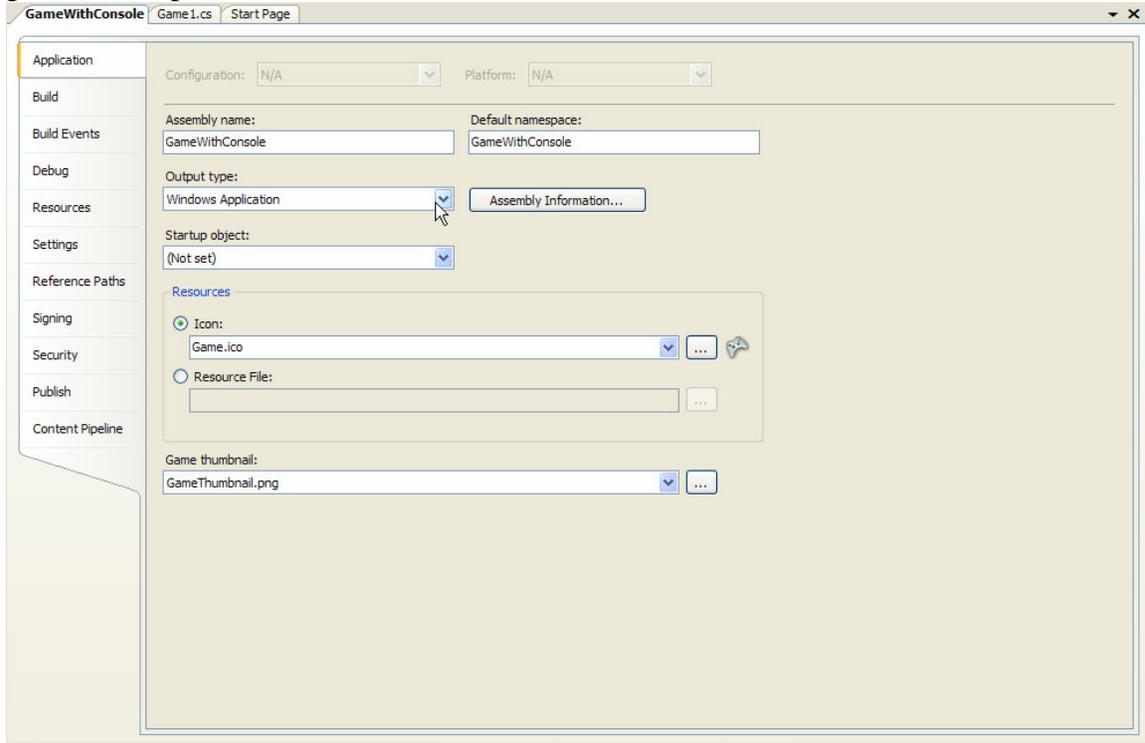
But now that we're using XNA, we don't have a console window any more. There will probably be times while you're programming that you will wish you had a simple console window to display some text for debugging purposes. This tutorial will show you how to get a console window to appear for the times when you need it.

### Console Windows

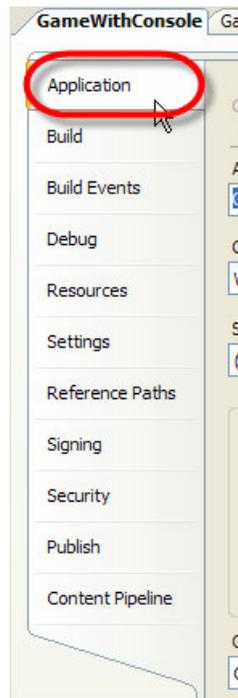
Getting a console to appear is fairly simple. The first thing we must do is open up the project's properties page. To do this, go to the Solution Explorer on the right side of the screen. Right click on your project, and the menu shown below will appear. Select **Properties** from the menu (at the bottom).



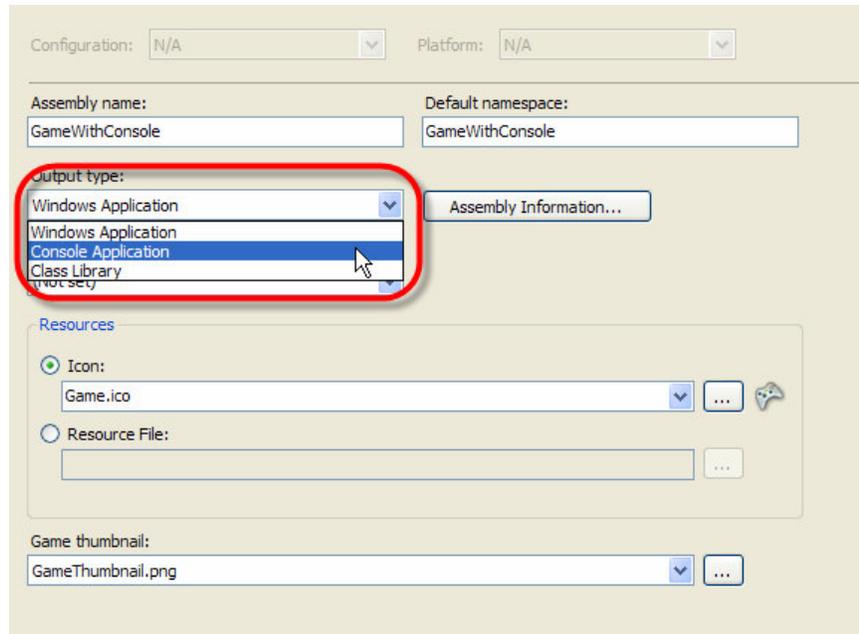
A new tab will appear that should look something like the image below. This is the properties page. From here there are many things you can do to configure how your game is set up.



Click on the **Application** button on the left side if it isn't already active.



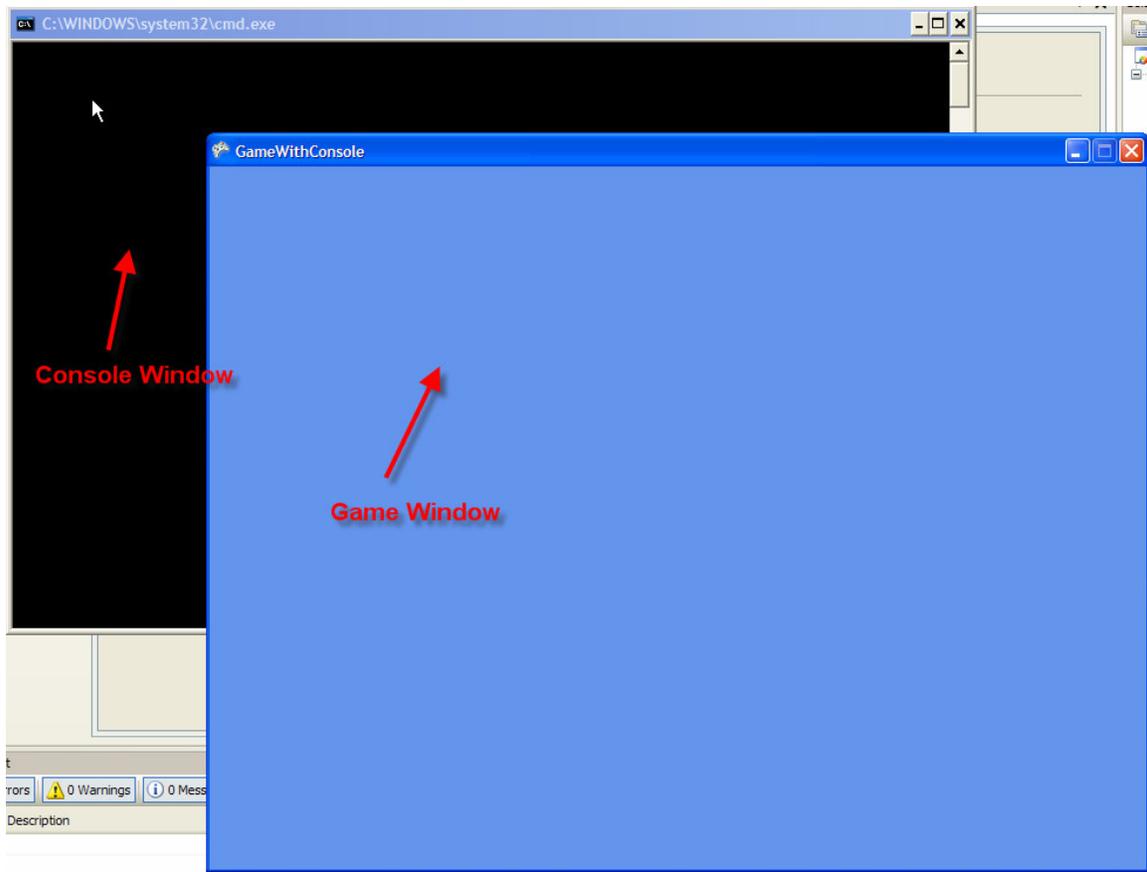
Find the box labeled **Output type**. It will most likely say “Windows Application” by default. Click on the down arrow button at the right end of this box to change the output type. To get a console window, we just want to change this from “Windows Application” to “Console Application”, so choose that from the list.



We should now be able to run our program again and get a console window in addition to our normal game window. (Shown in the image below.)

Console windows are actually pretty ugly, a little annoying to end users, and also a little confusing. You probably do not want to keep the console window in your game permanently, and so it can be turned back off, by repeating what we just did, but choosing “Windows Application” instead of “Console Application”. This will turn the console window back off.

Also note that when you close your game, the console window does not shut down automatically. It says “Press any key to continue . . .” Just do what it says and press a key while the console window is active, and the console window will close.



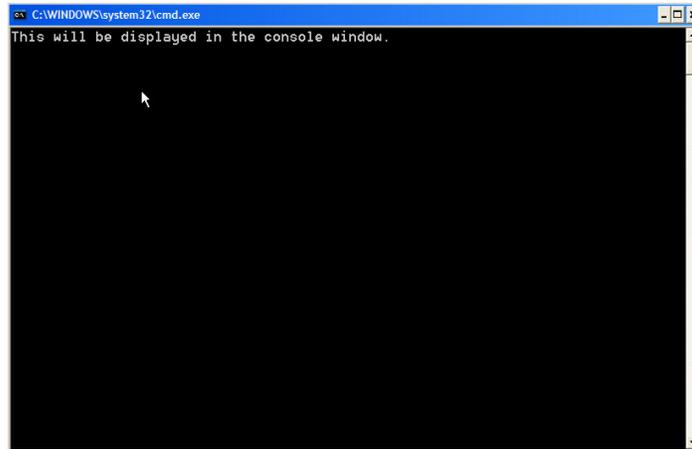
## Using the Console Window for Output

Many of you will already know how to use a console window, once you have it open. In that case, feel free to skip the rest of this tutorial. The rest of this tutorial is for the people who haven't really ever used a console window in C# before.

Now that we have a console window, let's discuss briefly, how to use it. The code for using a console window is quite simple. To write a line of text to the console window, try the following:

```
Console.WriteLine("This will be displayed in the console window.");
```

I put this code in my constructor, and so it will only be ran once, when the game is created. If you run this, you should see the following output in the console window:



If you put a `Console.WriteLine()` call in a method that gets called repeatedly, then the information will get printed out over and over again.

It might also be useful for us to print out the value of variables. This can be done the same way. The following code creates a new variable called **number** and gives it a value of 3. The code then prints out the value of **number**. (Note that it does not print out '3' necessarily, but the value of **number**. That is, when **number** changes, so will the value that is displayed.)

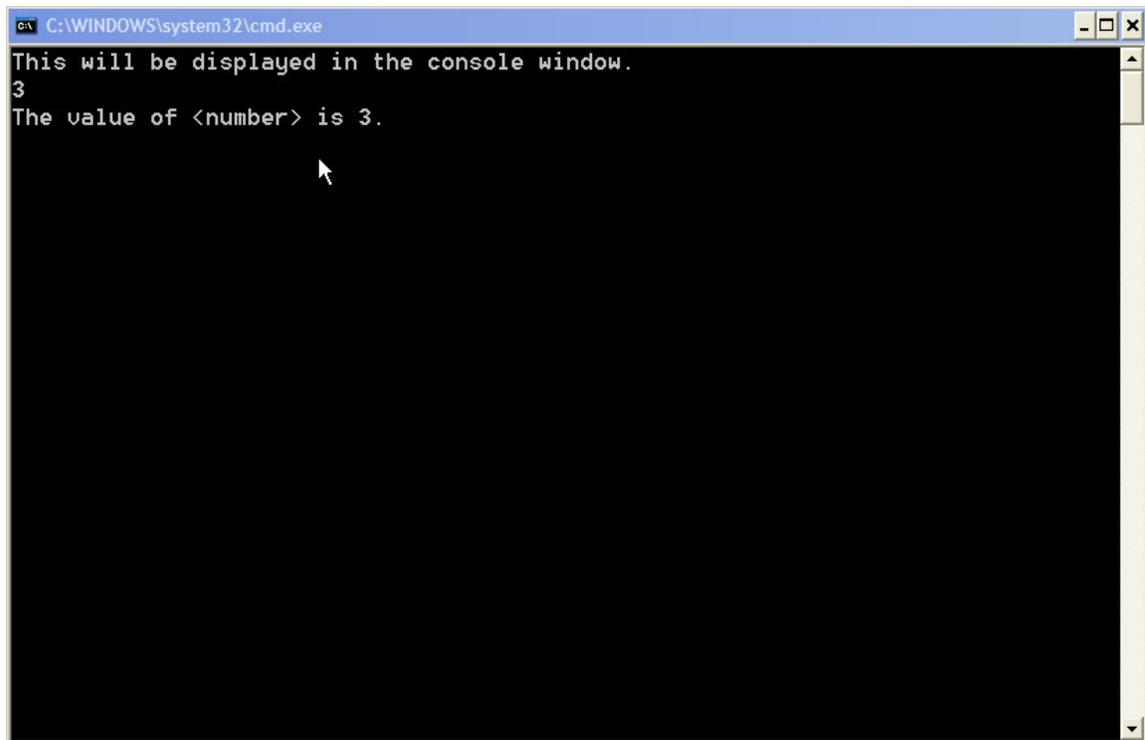
```
int number = 3;  
Console.WriteLine(number);
```

As a last trick, let's try a combination of printing out text and a value. This is fairly intuitive as well. To do this, use the following code:

```
Console.WriteLine("The value of <number> is " + number + ".");
```

This takes the string "The value of <number> is" and then appends (sticks on the end) the value that is in `number`. In this instance, the '+' is called the append operator. It combines two text strings together into one. In this case, it also converts the integer value of **number** into a text string as well (i.e. from the number 3 to the text string '3'), and so we don't have to worry about doing that ourselves.

The image below is what you should see if you run the program now:

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe". The window contains the following text:

```
This will be displayed in the console window.  
3  
The value of <number> is 3.
```

A mouse cursor is visible in the center of the window.

We can now go ahead and use this for all sorts of debugging. You can use it to print out the values of variables, print out text to make sure that certain methods are getting called, or just about anything you can think of. Console windows can be a powerful programming tool, just make sure to take them out (unless you have a really good reason) when you go to publish your game for end users.