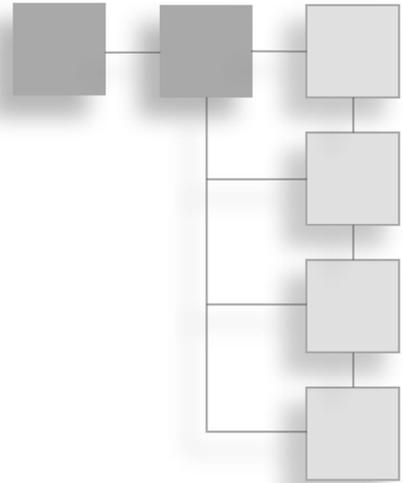# Chapter 17

Compiling Games

In this chapter, we will show you how to compile your games from the MFA format to a format that can be played on another computer, allowing other people to play the adventure games you create.

Note

You will need the full version of MMF2 to be able to compile games into a format that can be used by other people.

Exporting

When you create a game within MMF2, it will save it as a file with an .MFA extension, which means that only MMF2 can read the game. So if your friends are using MMF2, you can provide them with a copy of your game, and they will then be able to load it and run the application. If one of your friends tries to launch the MFA files but isn't running MMF2, he will get an error and will not be able to open or run the file.

The export option within MMF2 takes the MFA file format and produces a format that can be run on a particular piece of hardware; in some cases it might require a special plug-in or access, but on the whole, it will allow anyone with the right equipment to play the game.

To export a game, open or create a blank MMF2 application file and then click on the Application name in the Workspace toolbar to bring up the application properties.
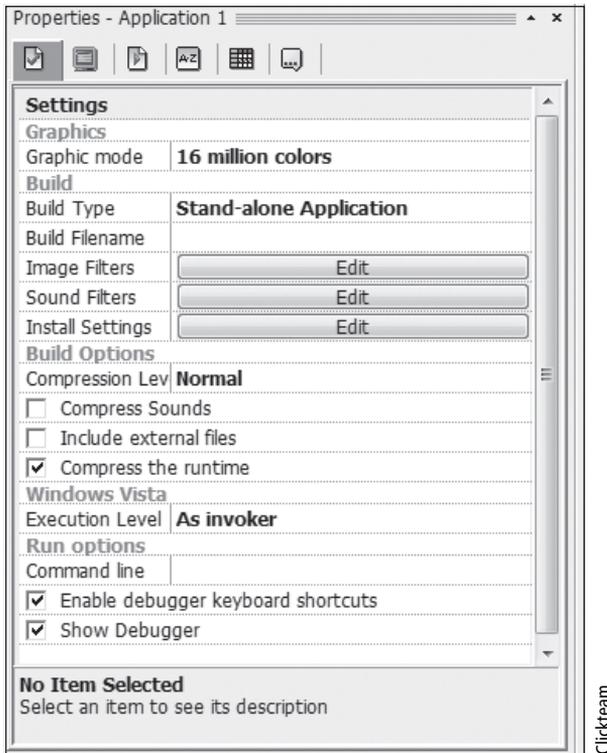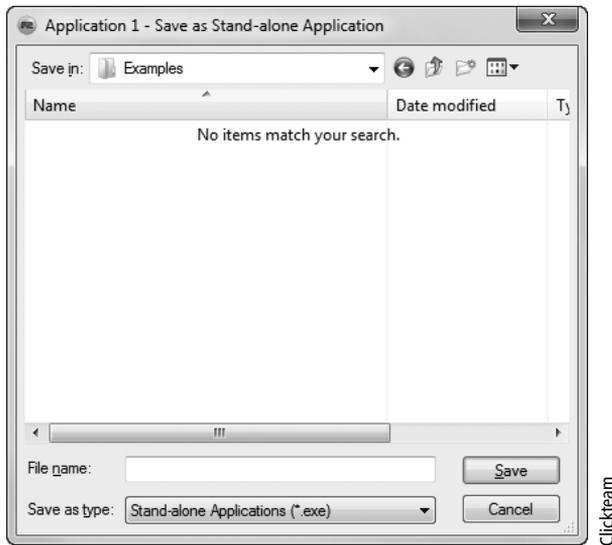
Figure 17.1
The Application Properties window and its default settings.

You can see the Settings tab properties window in Figure 17.1.

The default build type is Stand-alone Application, which will create a Windows executable. This format is compatible on any modern computer that runs Windows XP and Windows 7. The properties available in this tab and some of the others will depend on the build type that is selected; some features may be grayed out, which means they are not available for the currently selected platform.

It is really easy to export to a particular format. Here is an example of building a Windows executable file:

1. Select the build type from the drop-down box and then select File | Build Application from the File menu.

2. A dialog box appears in most cases asking where to save the exported file as shown in Figure 17.2.

3. After you type a filename and click on Save, MMF2 then begins to generate the required files for the selected build type.

Figure 17.2
The dialog box to export to a particular format.

Figure 17.3
The final exported file.

4. The file should then appear in the location that you selected. In this example, I typed "test" as the filename, and the icon/file appeared as shown in Figure 17.3.

Exporting Formats

Running the demo or full version of MMF2 will determine which options you have available in the build type. If you are running the demo version, you will not be able to export to any build type; if you have the full version, the list of available options will depend on which additional export plug-in you purchased.

Here is a list of available export options:

■ **Stand-alone Application:** This will create a Windows executable that can be run on any Windows PC running Windows XP and later.

■ **Screen Saver:** If you leave your PC logged in for a while but do not press any key on the keyboard or move the mouse, a screen saver may appear. A screen

saver was invented many years ago to help prevent monitor burn-in, which could occur if an image was left on-screen for a long time and then burned into the monitor, causing irreversible damage to its visual quality. As the quality of monitors has improved over the last 10 to 15 years, this problem has become a non-issue in most cases, but screen savers still generally exist. The screen saver format in MMF2 works for Windows operating systems only.

■ **Internet Application:** If you want to allow other people to play your games over the Internet, one way is to use the Internet Application export, which creates a special file that you can load into a web page. You will need a domain name and website where you can upload the file and HTML page. The Internet Application uses Clickteam's own web plug-in called Vitalize; unfortunately, users will need to download the Vitalize plug-in before they can play the game.

■ **Stand-alone Java Internet Applet:** Java is a format that has been used for a number of years and is able to be run on multiple operating systems, such as Windows, Linux, and Macintosh. The Java Applet is a web plug-in that can play within a web page, so as long as you have a domain and website to upload the file, many different people with different hardware will be able to play your games. The player accessing your game will need the Java plug-in installed on his computer before he can play, but Java is quite common and is already installed on many computers.

■ **Stand-alone Java Web Start:** The Web start version of Java allows you to put your game into an installer and enable the user to download it from the Internet and run it with a single click. Again, he will need Java installed to run this on his computer.

■ **SWF Flash Export:** One of the most common formats for creating rich and exciting games within a web browser is the Adobe Flash format (also known as SWF). Anyone who plays your game in this format will need to install the Flash plug-in, but this format is compatible with both Windows and Macintosh, and many computers will have it already installed.

■ **XNA and Windows Mobile:** A very exciting plug-in for MMF2 is the XNA format, which allows you to export your creations to the Xbox indie marketplace or Windows mobile formats. To export to the Xbox and Windows mobile formats, you will need the XNA exporter and need to purchase a yearly subscription to the XNA developer account. More information can be found on the Clickteam.com website.

- **iOS:** The iOS platform covers the Apple iPhone, iPad, and iPod devices and has been extremely popular over the last few years, and there is a growing number of adventure games for these products now being released. To use this exporter, you will need the iOS exporter, an Apple Mac computer with the latest Apple OS, and an Apple developer account. More information can be found on the Clickteam.com website. Even though you may not have all of these options available to you when using a particular version, it is good to know what you can export your games to if you want to. You may need to tailor the game mechanics or design of your game to accommodate the different playing styles of each platform, such as using a mouse for Windows computers and using touch for iPhone or iPad. Always try to remember the differences between the platforms if you intend to release a game on more than one platform.

Note

Java is not a particularly fast format, and other formats such as SWF and Vitalize will generally run faster. The key benefit is obviously the ability to play on multiple formats, so you will need to decide which is more important for your games—accessibility of platforms or performance. Of course, you can export to multiple formats and let your users decide which they prefer.
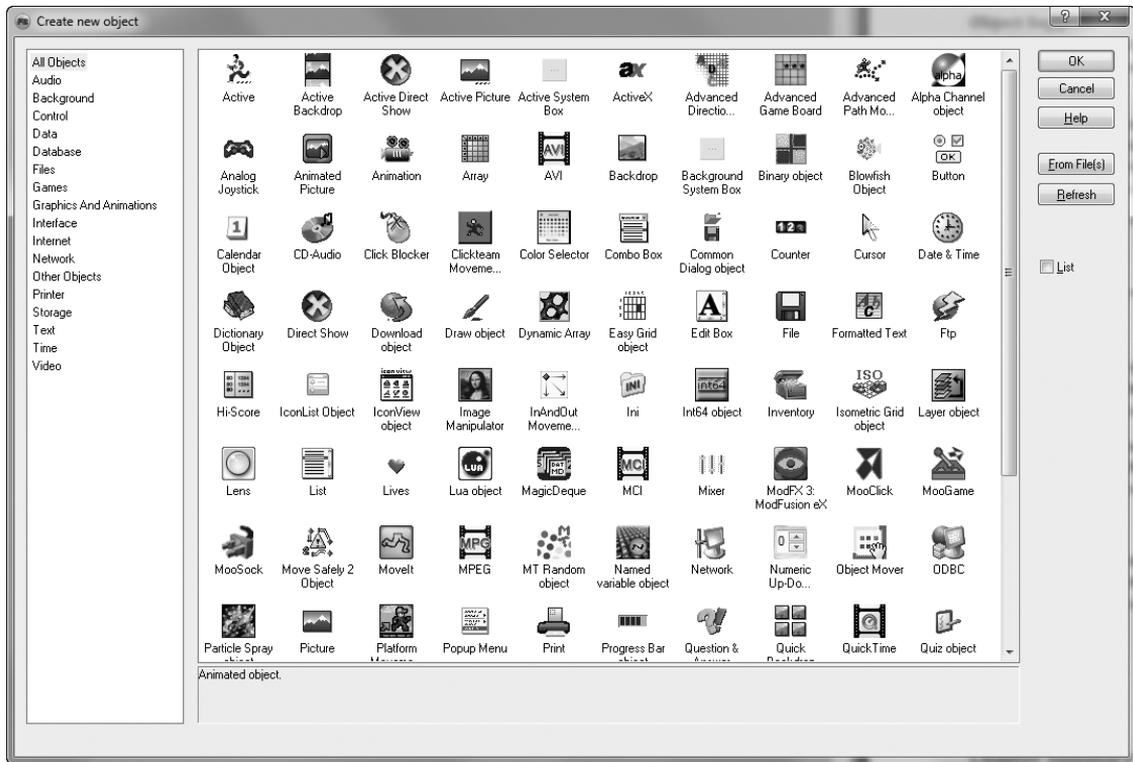
Note

When making a game for use on multiple platforms, you have to be aware of the different range of memory and performance requirements of each platform. For example, making a game for a PC will benefit from better memory and graphics capability than a mobile device, but you will also need to consider that, even on the same platform, there may be a range of technical specifications. For example, the PC platform can be sometimes considered a difficult platform to develop games for because you have to consider the different types of PC hardware and operating systems that can run on it. This is why PC game makers create minimum hardware specifications, so they can at least create a reasonable range of specifications to test on and ensure that the game works well on those machines.

Object Support

One of the main drawbacks of these different exporters is that not all objects are supported by every plug-in. For example, some Windows-based objects, such as pop-up menus, will not work on the iPhone.

When you select a build type, the number of objects you can select using the Insert Object option changes, so a Windows game generally allows the most types of objects that you can use, while mobile devices or SWF allow for less objects. You can see an
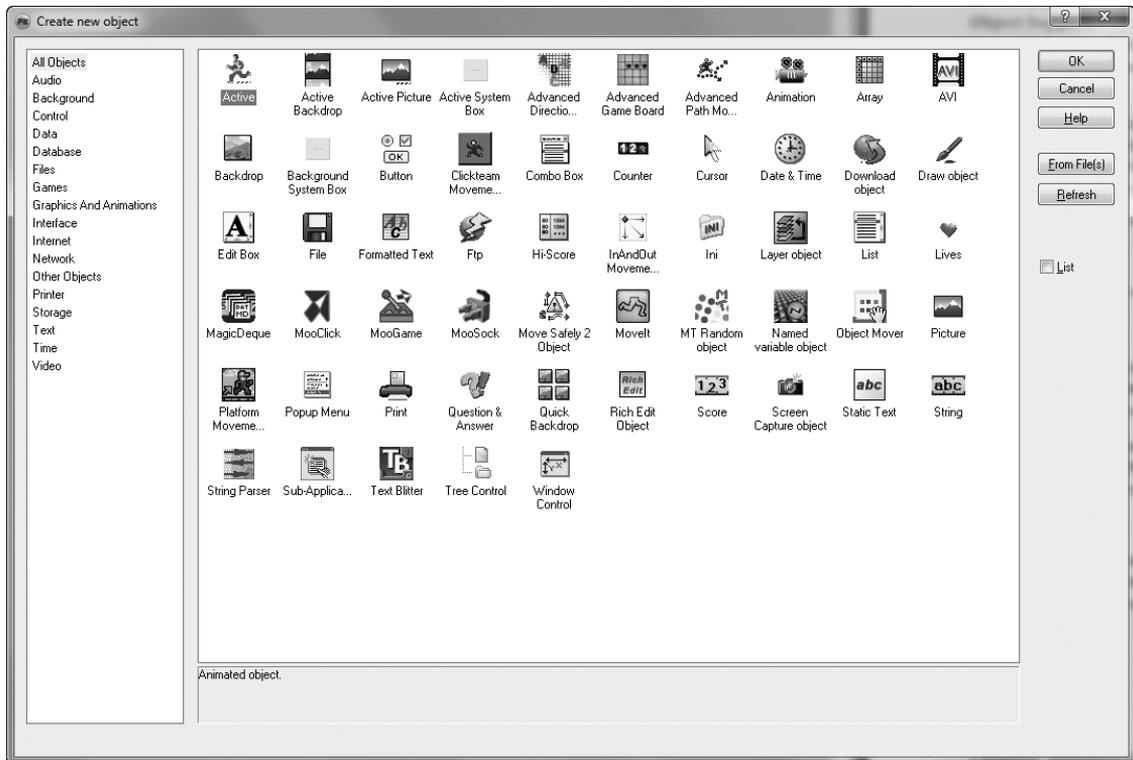
Figure 17.4
Windows-based objects.

example of the differences between the Windows build and Java build types in Figures 17.4 and 17.5.

It is really important that you make a note of the platforms that you intend to develop for before you begin writing any code, and make a note of the objects that you might need to work with and confirm that they are available on the exporters you intend to use.

In some cases, you can use alternative objects, which may require you to rework your code when moving onto another platform, so it is best to confirm what is available to you first and assess how much work will be involved in using another object.

Note

Be careful when using an alternative object to replace one from a particular platform, because some objects may not have all of the features of the object they are replacing. If they don't, then you won't be able to replicate the code you have already written.
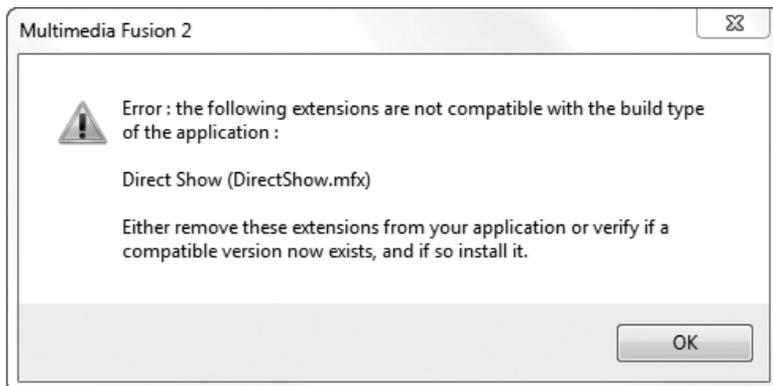
Figure 17.5
Java-based objects.

Exporting Errors

If you create a game in one format and then decide to export it to another format, you may receive a build error as shown in Figure 17.6. This message lets you know

Figure 17.6
The Export Error dialog box.

which object is causing an issue and will need to be removed before you can export to the selected format.

Chapter Summary

In this chapter, we looked at how you can create your games in a format that your friends and family can use on their own computers, Xbox consoles, Windows Mobile, iPhones, or over the Internet, allowing you to provide your games on as many different formats as possible.

In the next chapter, you will learn about the various resources you can access for additional help in creating your adventure games.