

Microsoft DirectMusic Producer: Game Development Tutorial

Legal Notice

This is a preliminary document and may be changed substantially prior to final commercial release. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user. Unless otherwise noted, the example companies, organizations, products, people and events depicted herein are fictitious and no association with any real company, organization, product, person or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unpublished work. © 2001 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Windows, Windows Media, Windows NT, ActiveSync, ActiveX, Direct3D, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, FrontPage, JScript, Microsoft Press, MSN, NetShow, Outlook, PowerPoint, SQL Server, Visual Basic, Visual C++, Visual InterDev, Visual J++, Visual Studio, WebTV, Win32, and Win32s are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

GM/GS® Sound Set © Copyright 1996 Roland Corporation U.S.

Roland is a registered trademark of the Roland Corporation.

Note The GM.dls collection supplied with DirectMusic uses the GM/GS® Sound Set, which is trademarked by the Roland Corporation and is the legal property of the Roland Corporation. The GM/GS Sound Set must not be modified and may only be redistributed in its entirety, without alteration. The component sounds and waveforms in the Roland GM.dls collection must not be used as source material to create other DLS sounds or collections.

Some components supplied with DirectMusic Producer can be modified and used as source material for creating your own components. However, some components have specific copyright restrictions and must not be modified or used as source material for creating your own components. This is true of any DirectMusic Producer and interactive music components that you download from the Microsoft Network (MSN) or other commercial Web sites.

You must conform to the legal restrictions for any component you use and read any copyright information contained in the Info tab of the individual components you use.

Table of Contents

Microsoft DirectMusic Producer: Game Development Tutorial.....	1
Installation and System Requirements.....	1
About the DirectMusic Farm Demo.....	2
Scoring an Interactive Application.....	3
What is DirectMusic?.....	3
How DirectMusic and DirectMusic Producer Work Together.....	4
Working with DirectMusic Components.....	4
Styles.....	5
Synthesis and Downloadable Sounds.....	5
Segments.....	5
Putting it all together.....	6
Tutorial Steps.....	7
Lesson 1: Getting Started.....	7
Lesson 2: Create a band.....	9
Lesson 3: Create patterns.....	12
Lesson 4: Create primary segments.....	22
Lesson 5: Create a secondary segment.....	29
Lesson 6: Create a DLS instrument.....	32
Lesson 7: Test playback.....	37
Lesson 8: Make wave segments.....	40
Lesson 9: Create an Audiopath.....	41
Lesson 10: Create a script file.....	43
Lesson 11: Export runtime files and test the game.....	47

Microsoft DirectMusic Producer: Game Development Tutorial

This step-by-step tutorial introduces the Microsoft® DirectMusic® Producer project workspace and guides you through the process of creating a DirectMusic-based score for a game. To complete the tutorial, you do not have to provide music or a game application. A sample game audio player application and all the elements needed for creating the music are included with the installation of DirectMusic Producer. You will copy and paste elements into DirectMusic components, and then configure the components and create run time files. You will then open the audio player and hear how the music you create works.

In addition to DirectMusic Producer, you will use the following elements, which is installed by default in C:\My Documents\DMUSProducer\Tutorial:

- **MIDI files.** The music that will be used to create the interactive music components is contained in several MIDI files. The music was composed using a third-party MIDI sequencer program, and then exported as standard MIDI files with a .mid file name extension.
- **Sample and sound effects files.** Several sound files with a .wav file name extension will be imported into DirectMusic components. One of the files is an instrument sample that will be used to create a downloadable sounds (DLS) file. The other files are farm sound effects that will be imported into DirectMusic components.
- **Game audio player and source.** After finishing the tutorial, you can run the audio player, which uses the DirectMusic components that you create. Source files for the application are also included.
- **Script file.** The audio player calls routines that you write in a DirectMusic script file. Script files give you more control over how your music plays in an application. You can type the script into the Script Designer or copy it from a text file.
- **Finished components.** If you run into difficulties or prefer not to perform the steps in the tutorial, the finished DirectMusic components are provided.

The tutorial does not require that you know how to compose music; however, the concepts will be easier to grasp if you have a general knowledge of music construction, sound design, and composition with a MIDI sequencer. A basic knowledge of programming is also helpful, but is not required for understanding the concepts behind script files.

For more information about DirectMusic Producer, see the online Help documentation that is installed with the product. If you are a developer and would like more information about how to implement DirectMusic in your application, download the DirectX 8 Software Development Kit (SDK) from the Microsoft MSDN™ Online Downloads Web site at <http://msdn.microsoft.com/downloads/default.asp>.

Installation and System Requirements

To follow the tutorial, you need DirectMusic Producer and the tutorial content files, which are included with the installation of DirectMusic Producer. By default, the tutorial files are located in C:\My Documents\DMUSProducer\Tutorial.

If you do not already have DirectMusic Producer installed on your computer, you can download it from the MSDN Download Web site at <http://msdn.microsoft.com/downloads/default.asp>.

Before installing DirectMusic Producer, you need to install the DirectX 8 runtime. The runtimes for Microsoft Windows® 95, Windows 98, Windows 98 Second Edition, Windows Millennium Edition, and Windows 2000 can be downloaded from the DirectX Web site at <http://www.microsoft.com/directx/homeuser/downloads/default.asp>.

You can also install the complete Microsoft DirectX SDK from the MSDN Online Web site. The SDK includes the DirectX 8 runtime, and complete information for developing applications using any of the DirectX technologies.

System Requirements

- Microsoft Windows 95, Windows 98, Windows 98 Second Edition, Windows Millennium Edition, and Windows 2000.
- The full DirectX installation process requires approximately 50 megabytes (MB) of free space on your hard drive. After installation, the DirectX download takes approximately 15 MB of hard drive space.
- DirectMusic Producer installed separately from the full DirectX installation requires approximately 8 MB of hard drive space.
- If you have an earlier version of DirectX installed on your system, you will see little difference in used space on your hard drive following installation of DirectX 8.0. DirectX 8.0 will overwrite the earlier versions.

Note After installation, the DirectX 8.0 runtime cannot be uninstalled. The installation process changes core components and makes numerous registry changes within your operating system. Microsoft does not support uninstallation.

About the DirectMusic Farm Demo



You will play the DirectMusic components that you create in the tutorial with the DirectMusic Farm Demo. The demo is an audio player application that was built for a fictitious game. An application such as this one can be created for a real game to enable composers and sound designers to test DirectMusic components in an application outside the game environment.

In addition to demonstrating how DirectMusic can be used in an application, the audio player demonstrates variability, how user input can affect the music (interactivity), and how DirectMusic can be used to add sound effects. The audio player also gives you an opportunity to learn how to prepare the music and sound you create in DirectMusic Producer for use in an application.

The following types of sound and music are used in the game:

- **Sound effects.** User action in the game triggers sound effects to play.
- **Backgrounds.** The game has three levels that correspond to a time of day, and music backgrounds that play throughout each level. The fictitious game user starts with the Night level and progresses to Dawn. As the user does so, the music changes and becomes fuller. When the user stops the game, the music transitions to the end piece.
- **Secondary Segment.** This DirectMusic component contains music that plays over another sound. During game play, user action triggers a bird to fly across the screen. As the bird flies, music plays over the background.

In this tutorial, you will create all the music and sound effects components for the game. When you finish, you can open the DirectMusic Farm Demo audio player, and test the components. Because a script file is used, you can experiment with changing the music and settings without having to recompile the player.

Before beginning the tutorial, you should read the following short sections so that you have an understanding of basic DirectMusic concepts.

Scoring an Interactive Application

Games and other interactive applications are controlled by user input so they are designed with variability and flexibility in mind. However, the music used in many of these applications is linear and very inflexible. Synchronizing linear music with user interaction often results in a musical score that sounds mechanical or jarring. DirectMusic Producer enables you to enhance your interactive application with music that is non-linear so that users experience a continuous musical score that is actually being composed in real time.

Most linear music has a predetermined beginning, middle, and end, like a story. When writing linear music, a composer starts from the beginning and works linearly to the end. The end user must play it back the same way, from beginning to end, or the experience is incomplete.

Non-linear music is written to synchronize with events generated by users in interactive applications. In essence, the user composes a linear score in real time while playing the game or interactive application. The score is created from musical elements supplied by the composer. To make music non-linear, a composer thinks in terms of patterns, like pattern-based MIDI sequencing. The composer creates these patterns so that they can be combined in a number of ways to create a variety of musical results. The score seems linear and continuous to the end user, but is actually being assembled from patterns in real time.

In addition to arranging patterns of music into unique compositions, a DirectMusic score employs several other methods to ensure that transitions between patterns sound smooth. The composer also has many options for introducing variability so the music sounds fresh and new. As you go through the tutorial, you will see how these methods work.

What is DirectMusic?

Microsoft DirectMusic is a part of Microsoft DirectX®, a set of components installed in the Microsoft Windows operating system to provide fast and full-featured support for graphics, input, network play, and sound. In combination with Microsoft DirectSound®, DirectMusic provides a complete solution for playing music and sound effects in games and other multimedia applications.

All of the data for reproducing interactive music and sound—the musical sequences, sampled sounds, and configuration information—is contained in DirectMusic components. To compose DirectMusic components and create files from the components that can be used in an application, you use DirectMusic Producer. Any application using the files created with DirectMusic Producer uses the DirectMusic application programming interface (API) to load and play those files.

When creating a DirectMusic score for an application, the composer or sound designer must work closely with the application developers. DirectMusic and DirectMusic Producer are designed to work together, and they share some functionality; that is, some behaviors of the sound files can be controlled either at composition time in DirectMusic Producer, or at run time by the application.

How DirectMusic and DirectMusic Producer Work Together

Applications that use the DirectMusic API are not restricted to playing content from DirectMusic Producer. They can play wave and MIDI files, and they can construct music and sound effects. However, the full power of DirectMusic can be unleashed only with files authored in DirectMusic Producer.

Using DirectMusic Producer content, an application can do the following:

- Play music or sound effects with automatic variations, ranging from subtle changes in the timing to different patterns of notes.
- Change the choice of patterns in the soundtrack by setting a value called the groove level.
- Synchronize different pieces of music on rhythmic boundaries.
- Synchronize other events in the application with the rhythm; for example, display graphics in time with music.
- Compose a musical piece at run time, based on note and chord patterns authored in DirectMusic Producer.
- Compose smooth transitions between pieces of music at run time, even when the current piece is interrupted.
- Play different musical parts on different audiopaths, so that different effects and 3-D positioning can be applied to different band instruments or other sounds within a single file.
- Use scripts to give the sound designer more direct control over the soundtrack.

You create the DirectMusic score and sound for an interactive application by building and integrating a number of DirectMusic components, including styles, DLS sound sets, and segments.

Working with DirectMusic Components

You produce a DirectMusic score by first creating DirectMusic components with DirectMusic Producer, and then saving the components in a number of working files. When you are ready to hand-off the components to the application developer, you convert the working files into run time files. The developer then writes the code that incorporates the files and plays the components contained in them.

The main DirectMusic components are styles, DLS collections, and segments.

- Styles contain basic musical sequences or patterns.
- DLS collections contain wave samples and other information used by the synthesizer to create instrument timbres and other sounds.
- Segments contain an assemblage of components, and they are used by an application to play a DirectMusic score.

The following sections describe the components in more detail. You will learn more about the components as you work with them in the tutorial. The tutorial does not cover every feature of the components, but the basic concepts you learn will help you expand your knowledge with the online Help and with experimentation after you finish the tutorial and move ahead with DirectMusic Producer.

Styles

Styles are building blocks containing basic musical sequences and information that can be used to create many different pieces of music.

A style contains the following elements:

- **Patterns.** A pattern consists of musical phrases—sequences of MIDI notes and controller data—with one or more instrument parts. A style normally has multiple patterns that are each labeled to serve a different role in an interactive score.
- **Motifs.** If you want to play a musical phrase with or on top of a pattern, you create a motif. For example, you could create a short French horn motif that synchronized with a game event and played on top of a background pattern. The construction of a motif is very similar to that of a pattern.
- **Bands.** A band is a set of instruments, along with their assignments and properties. The instruments are assigned to performance channels (PChannels), which are similar in concept to MIDI channels. They are also given volume and stereo pan positions. A band can contain any available instrument, including the General MIDI (GM/GS) set that is provided with DirectMusic and instruments collections that you create.
- **Global style settings.** These include time signature and tempo.

Synthesis and Downloadable Sounds

Most consumer computers play MIDI sequences through synthesizers contained in sound cards or other dedicated hardware modules. Though these synthesizers are based on the General MIDI (GM) sound standard, the sounds vary significantly from model to model. For example, a high-end sound card may produce a very realistic violin sound, where a low-end card may produce a sound that is nothing like a violin.

On the other hand, DirectMusic Producer itself, and most applications that use DirectMusic to play content, use the Microsoft Synthesizer, which synthesizes sounds with software. The main advantage of software synthesis is that the sound of a MIDI score is not dependent on the MIDI sounds included in a user's sound card. A violin sound, for example, sounds the same regardless of the quality of the computer or sound card.

The synthesizer uses sound samples that conform to the DLS Level 2 standard, created by the MIDI Manufacturers Association. A sample might represent the timbre of a particular musical instrument at a given pitch or within a range of pitches; it might represent a non-musical sound effect; or it might even consist of a complete musical phrase.

The synthesizer can modify samples according to supplied parameters, which include pitch and articulation settings. Even sound effects included as audio files, also known as wave files or waves, can be modified in this way. You could create a segment based on a wave sample of an airplane, for example. The application using the sound could apply pitch bend at run time to reflect the changing speed of the airplane.

Segments

Segments are the basic components played by an interactive application. You can think of a segment as a workspace where you assemble components and instructions on a timeline to compose your non-linear score. A segment consists of tracks into which you place elements such as DirectMusic components (including styles and bands), imported MIDI sequences and sound files, and time-based instructions (such as chord and style changes). For example, you can use a style track to play patterns in a style, a chord track to change the chord and scale of the music over time, and a band track to associate a set of instruments with a segment.

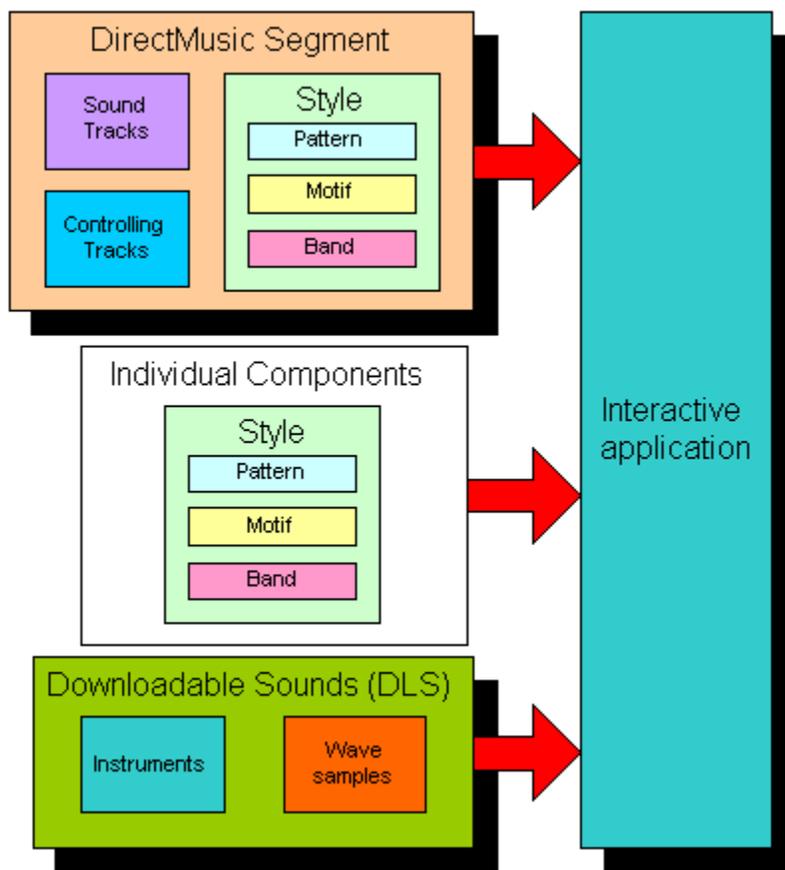
A segment contains sequential data and can be played linearly from beginning to end. However, the same data is not necessarily played each time the segment is played, because a segment can contain multiple variations. In addition, transitions between segments can be initiated at any time by an application, and you can design how the transitions are handled so the music sounds continuous.

There are two types of segments: primary and secondary. One segment is played as the primary segment, and any number of segments can be played along with it as secondary segments. One segment is designated as the controlling segment, and it controls global parameters such as tempo and chords. Usually, this is the primary segment.

When a segment contains one or more styles, the labels that you apply to the patterns in the style are used to create a continuous piece of music based on data, settings, and rules that you apply. For example, a pattern might be designated as an introduction or ending, or it might be assigned a certain level of complexity or groove range. In addition, pattern parts can have variations that are selected randomly during playback. For example, you can create a violin part in a pattern that has up to 32 variations.

Putting it all together

The following figure shows the main components of a DirectMusic score and how they are packaged for use in an interactive application.



There are two methods of implementing DirectMusic in an application; an application can play one or more DirectMusic segments directly (the top block in the figure), or it can compose segments in real time based on individual styles that you provide (the middle block). The method you choose depends on how much control over the music an application is to provide. A complete score can use either method or a combination.

In addition to music components, a DirectMusic composition requires at least one DLS collection. You can import samples into collections or use samples from the default GM/GS sound set that is installed with DirectMusic. Not pictured is an audiopath component, which you can add to define the output path of the Microsoft Synthesizer, and a script file.

Tutorial Steps

In this tutorial, you will create segments that are played directly by the application. The following steps describe a typical process of creating a DirectMusic score, which is roughly the process you will follow in this tutorial.

1. Create one or more style components.
2. Create DLS instruments or select instruments from an existing DLS sound set, such as the General MIDI set included with DirectMusic Producer.
3. Create one or more segments.
4. Create an audiopath component.
5. Create a script file. A script file makes it easier for a developer to use DirectMusic in an application, and gives you more control over how the music plays.
6. Create DirectMusic run time files from the DirectMusic Producer working files, and deliver them to the application developer.

In each DirectMusic component, you have the opportunity to add variation. You decide the make-up of these elements and how they will be combined.

In the next section, you begin the tutorial. In the first lesson, you will set up a new DirectMusic Producer project.

Lesson 1: Getting Started

In this lesson, you will start DirectMusic Producer, create a new project, and set up the project folder. In practice, you can add new components at any time, but it is helpful to begin with a basic framework.

We recommend that you follow the steps exactly, and then after you have successfully completed the tutorial, experiment with the settings and processes. For example, you can replace the tutorial music with your own or add more variations. As you explore DirectMusic Producer, you will find that there may be many ways to accomplish the same tasks. This tutorial describes one way. As you become more familiar with DirectMusic Producer, you will perhaps discover more efficient methods of working, and a system that works better with your style of composition and sound design. The tutorial is designed to introduce you to some of the most important components of the application without explaining every aspect of their use.

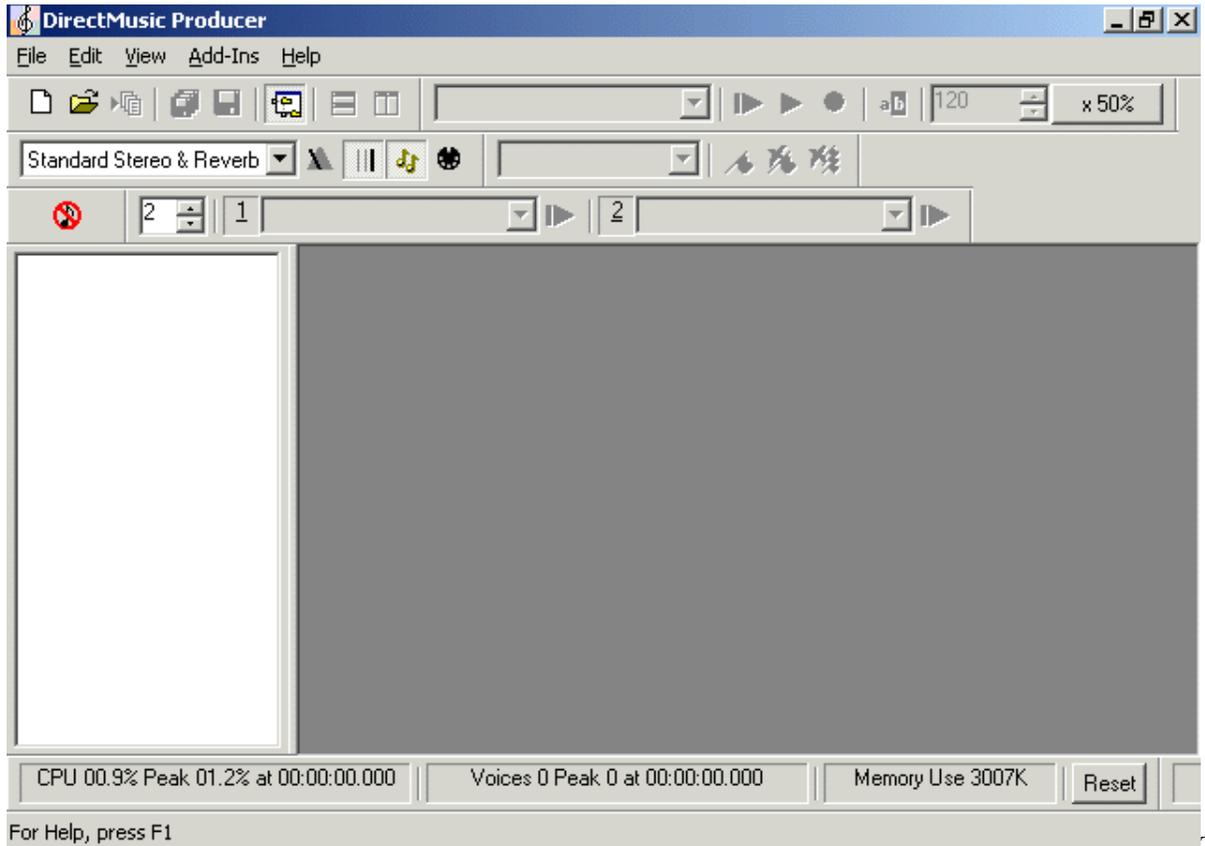
Each lesson includes suggested topics in the online Help documentation included with DirectMusic Producer, which describe the user interface in more detail.

For more information, see the following topics in the online Help

- Project Workspace
- Project Tree

Start DirectMusic Producer

1. On the **Start** menu, point to **Programs**, point to **Microsoft DirectMusic**, and then click **Microsoft DirectMusic Producer**.
2. If one or more projects are open in the project tree, click each project in turn, and then on the **File** menu, click **Close Project**.
3. Your empty workspace should resemble the following figure:



The

workspace is divided into three sections:

- **Toolbars and menu.** You control and configure DirectMusic components with the toolbar buttons and menu items at the top.
- **Project tree.** Files associated with a DirectMusic Producer project and the components contained in the files are displayed in the project tree.
- **Editing window.** You create and edit DirectMusic components in editor and designer windows that open in the editing window.

For more information, see the following topic in the online Help

- Main Window

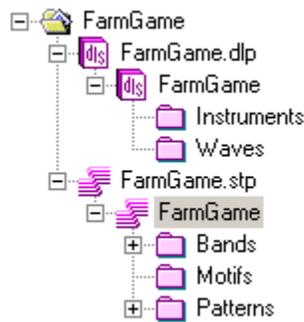
Create a new project

1. On the **File** menu, click **New**.
2. Click **Project**, press the CTRL key, and then click **DLS Collection** and **Style**.
3. Clear the **Use Default Names** check box, type **FarmGame**, and then click **Ok**.
4. Accept the default location or enter a new location for the project.

A folder is created at the location you specify that contains the following files:

File name	Description
FarmGame.dlp	DLS collection file
FarmGame.pro	Project file
FarmGame.stp	Style file

The files are also reflected in the project tree, as are the DirectMusic components contained in the files.



Each component contains subsections into which you will import files and enter settings. In the DLS collection file, you will add an instrument sample and performance settings. In the style file, you will import a number of MIDI sequences and enter instructions that affect how the sequences are to be played. The patterns, bands, and motifs contained in a style are the building blocks from which you build your DirectMusic segments.

It is recommended that you save the project every time you complete the steps in a topic.

- To save the project, on the **File** menu, click **Save Project FarmGame**.

In the next lesson, you will assign instruments to the band that plays the DirectMusic score.

Lesson 2: Create a band

Bands contain collections of instruments, along with their volume and pan settings. For example, you can create one band that contains an accordion and clarinet for playing polka music and another that contains guitar, bass, and a drum kit for playing rock.

You add an instrument to a band by mapping it to a performance channel, or PChannel. PChannels are similar in concept to MIDI channels. However, unlike MIDI channels, you can create up to 999 PChannels. You can map a saxophone, for example, to PChannel 3, and then any music parts assigned to PChannel 3 will play the saxophone. The FarmGame score is primarily orchestral, so the band contains instruments such as strings and a cello. In this lesson, you will assign instruments from the default GM sound set.

For more information, see the following topics in the online Help

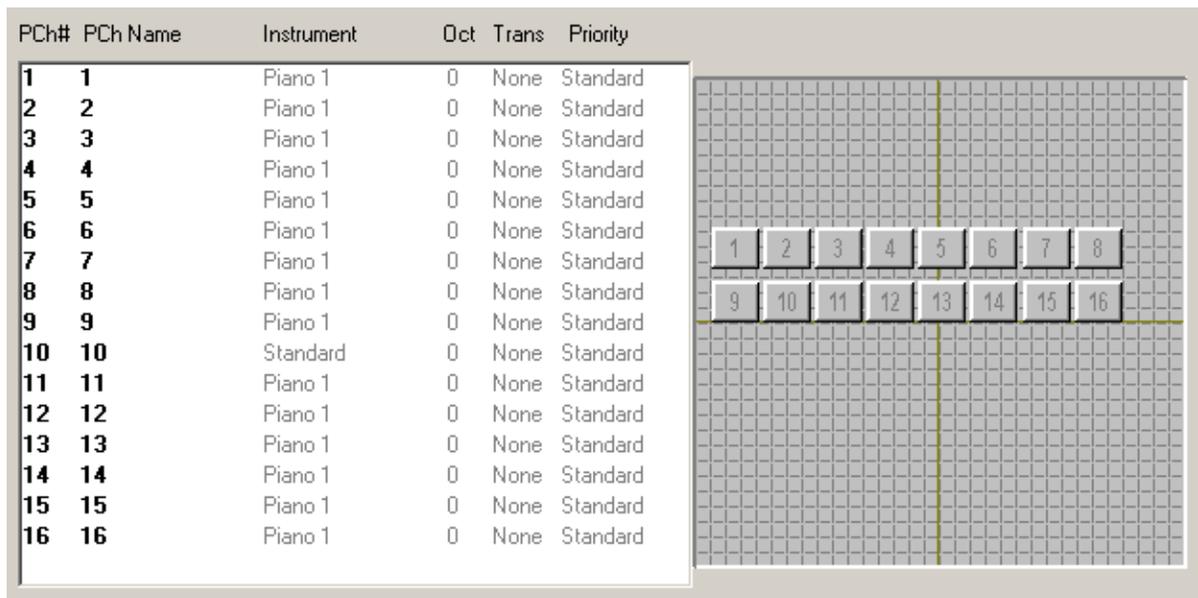
- About Bands
- Creating a Band

Create a new band and assign instruments

1. In the project tree, expand the FarmGame style (FarmGame.stp) if necessary, and then expand the Bands folder.

The Bands folder contains one band, Band1, which is the default band that is created with a new style.

2. Double-click Band1, and the **Band Editor** opens.



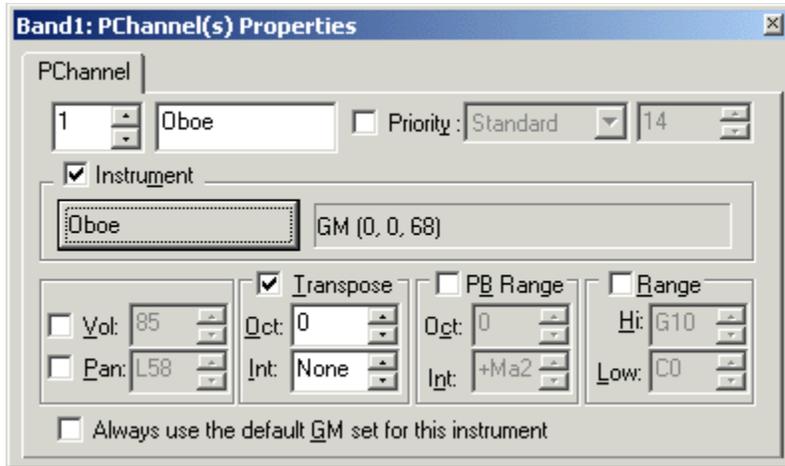
The area on the left is the **Performance Channel** list where you assign instruments to PChannels. The area on the right is the **Volume and Pan Graphic Interface** where you mix the volume levels and set the stereo pan position of the PChannels.

3. In the PChannel list, right-click PChannel 1 (listed as 1 in the PCh# column), and then click **Properties**. The **PChannel Properties** window opens.
4. Click the **Instrument** check box to select it. The **Instrument patch assignment** button is initially set to Piano 1.
5. Click the **Instrument patch assignment** button and the **Instrument Family** menu opens.
6. On the **Instrument Family** menu, point to **Reed**, and then click **Oboe**.

- In the name box next to the PChannel number, type **Oboe**.

Note You can give the PChannel any name you choose. However, for this tutorial, type the name of the instrument as described so that identification of the PChannel is easier.

The property window should resemble the following figure.



- Without closing the **Band Properties** window, click the next empty PChannel in the **Performance Channel** list.
- Repeat steps 4 through 8 to assign instruments to PChannels 2 through 9 according to the following table:

PChannel	Instrument family	Instrument
2	Ensemble	Strings
3	Orchestral	PizzicatoStr
4	Synth Pad	Fantasia
5	Orchestral	Cello
6	Brass	French Horns
7	Orchestral	Harp
8	Orchestral	Timpani
9	Synth Pad	Sweep Pad

Note You can quickly jump between property windows in DirectMusic Producer without closing a window first. To move from one property window to the next, simply click the next item for which you want to view properties. The properties window will change to display the settings for the selected item.

To finish the setup of the band, you will set initial volume levels and stereo pan positions.

For more information, see the following topics in the online Help

- Band Editor Window
- PChannel Properties

Set initial volume levels and pan positions

1. Click PChannel 1 in the list. The PChannel 1 mixing button is highlighted red in the **Volume and Pan Graphic Interface**.
2. Drag the PChannel 1 mixing button to roughly the center of the interface.

Moving a button up or down increases or decreases the volume of a PChannel; moving it left or right pans the instrument accordingly.

3. Repeat steps 1 and 2 for each PChannel.

PCh#	PCh Name	Instrument	Oct	Trans	Priority
1	Oboe	Oboe	0	None	Standard
2	Strings	Strings	0	None	Standard
3	Pizz	PizzicatoStr	0	None	Standard
4	Fantasia	Fantasia	0	None	Standard
5	Cello	Cello	0	None	Standard
6	FrHrn	French Horns	0	None	Standard
7	Harp	Harp	0	None	Standard
8	Timp	Timpani	0	None	Standard
9	Sweep	Sweep Pad	0	None	Standard
10	10	Standard	0	None	Standard
11	11	Piano 1	0	None	Standard
12	12	Piano 1	0	None	Standard
13	13	Piano 1	0	None	Standard
14	14	Piano 1	0	None	Standard
15	15	Piano 1	0	None	Standard
16	16	Piano 1	0	None	Standard
17	17	Piano 1	0	None	Standard

Each instrument is set initially to half volume and panned to center. In the next lesson, you will create patterns containing MIDI sequences that use the instruments in Band1. At the end of the lesson, you can reopen Band1 and perform a final mix of the instruments as you play back the patterns.

For more information, see the following topics in the online Help

- Volume and Pan Graphical Interface
- PChannel Volume and Pan

Lesson 3: Create patterns

In this lesson, you will import MIDI files into five patterns.

Patterns make up the main body of a style, containing the core musical sequences from which a non-linear score is created. If a MIDI keyboard is connected to your computer, you can record sequences directly into patterns. For the tutorial, however, you will import MIDI files that were created with a third-party sequencer into new patterns.

Each instrument part in a pattern can contain up to 32 different sequences called variations. A pattern can play a different variation each time it plays, depending on how the pattern is configured. This is one of the primary ways that DirectMusic enables you to add variability to non-linear music. Each variation can contain an entirely different set of notes or you can use the variations to simply introduce subtle variability in a rhythmic pattern or a performance. In the tutorial, you will add only one variation to each pattern to keep it simple.

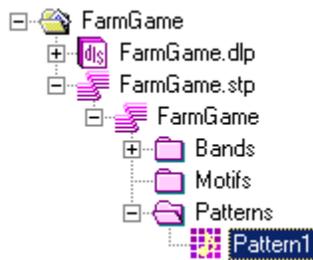
Each pattern serves a different purpose in the interactive score. As you continue with the tutorial, you will see how all the pieces fit together.

For more information, see the following topics in the online Help

- Style Designer
- Creating Patterns and Motifs in the Project Tree

Import MIDI files into patterns

1. In the project tree, expand the FarmGame style, and then expand the Patterns folder.



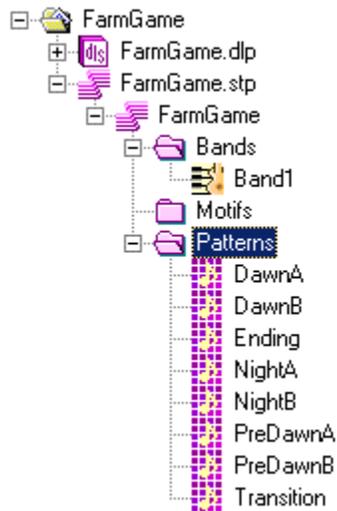
By default, one empty pattern (Pattern1) is created with a new style. You will not use this pattern. Instead, new patterns will be created as you import MIDI files.

2. Right-click the Patterns folder, and then click **Import MIDI File as Pattern**.
3. Locate the game tutorial MIDI files. By default, the files are located in c:\My Documents\DMUSProducer\Tutorial\RawElements.
4. Select all the files in the Farm folder that have a .mid file name extension, except SSBird.mid, and then click **Open**. To select all files, click the first file in the list, press the SHIFT key, and then click the last file. Then, press the CTRL key and click SSBird.mid to clear the selection.

A new pattern and band is created for each MIDI file. Because the same instrumentation (Band1) will be used for all patterns, you can delete all the bands that were added with the new patterns.

5. Click DawnA in the Bands folder, press the DELETE key, and then press ENTER or click **OK** to confirm the deletion.

- Continue to delete all the bands except Band1, and then delete Pattern1. The Patterns folder should resemble the following figure.

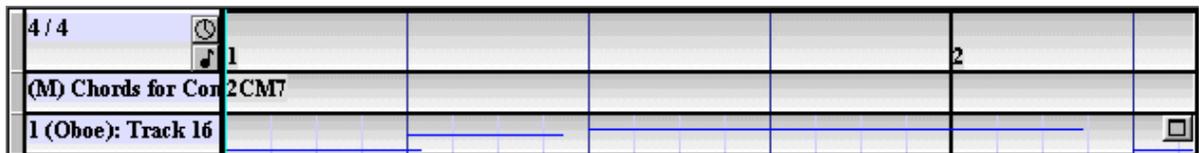


In the next topic, you will configure the patterns and add variations. You will also play the patterns to hear how the variations work.

Copy sequences to create new variations

The three patterns DawnB, NightB, and PreDawnB contain sequences that you will copy into variations in DawnA, NightA, and PreDawnA. After you have created the variations in the *A* patterns, you will delete the *B* patterns.

- Double-click NightB. The **Pattern Editor** opens.



Notice that the part in the MIDI file is mapped to PChannel 1, which is assigned to Oboe. You can play the pattern by clicking the **Play from Start** button.

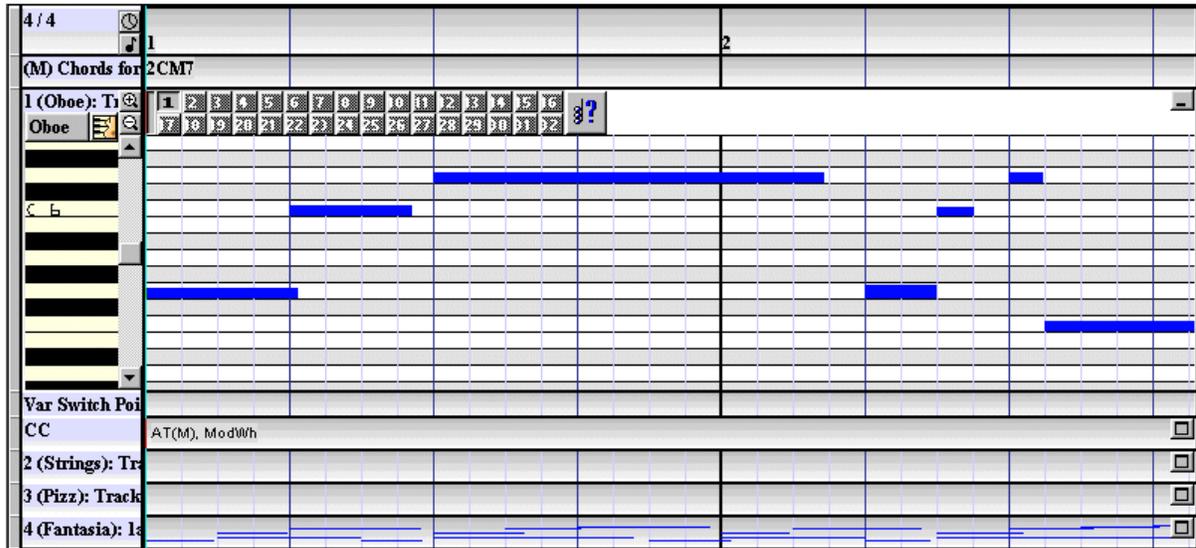


- Click the **Selection** button for Part 1. **Selection** buttons run along the left edge of the tracks and parts, and identify which ones are to be included in an edit operation.



- To select the whole part, on the **Edit** menu, click **Select All**.
- On the **Edit** menu, click **Copy**. The part is copied to the clipboard.
- Double-click NightA.

- Click the **Maximize** button  for part 1 (Oboe).



The **Sequence Editor** opens for part 1, displaying a MIDI sequence. The **Variation** strip above the sequence shows that variation 1 is currently enabled, which means that it can be played in the pattern. It is also active, which means the sequence is displayed. The other 31 variations are disabled and inactive.

- Click variation 1 to make it inactive—no sequences displayed.



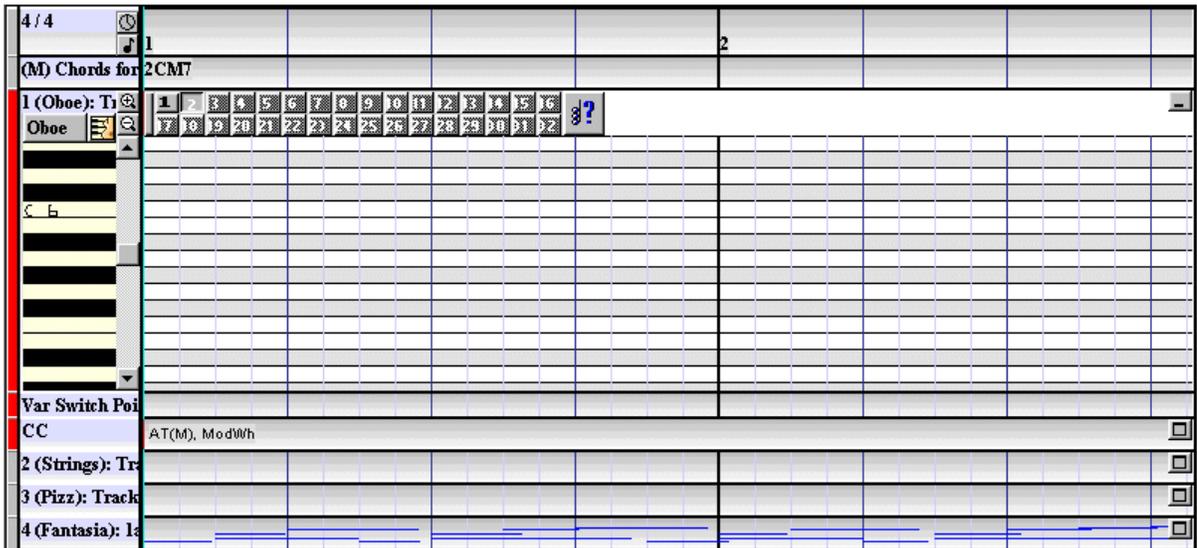
- Click variation 2 to make it active. The empty variation 2 sequence is displayed.



- Right-click variation 2, and then click **Enable** in the shortcut menu. Variation 2 is enabled and active.



- Click the **Selection** button for Part 1, and then press the HOME key. The play cursor, the red vertical line, which shows the current play position, moves to the beginning of the pattern.



Note that data is pasted starting at the position of the play cursor. If the cursor is not at the beginning of the pattern, the sequence will be offset.

- In the **Edit** menu, point to **Paste**, and then click **Merge**.

The oboe part is copied into variation 2. The pattern now has two variations. You can see the sequence in each variation by activating either variation button. You can also hear each by activating a variation and then clicking the **Play from start** button.

- Right-click the part name (1 Oboe), and click **Properties**. The **Part Properties** window opens.
- In the **Play** list, click **by shuffling**, and then close the window. The order of the variations will be shuffled, and a variation will not play again until all of the variations have played.

To hear how this sounds, click the **Selection** button to deselect the part, and then click anywhere in the timeline on the **Pattern Editor**. Then, click the **Play from start** button. After finishing the tutorial, you can experiment with variability by adding more variations and adjusting parameters in the **Variation Choices Window**.

Note There are two ways to select a part: by using the **Selection** button and by clicking the part name or sequence. Each method is used for a different purpose.

- Select a part with the **Selection** button if you want to copy, cut, or paste data in the part.
- Select a part by clicking the part name or sequence if you want to play only the variations that you have made active with the variation strip. If you want to hear playback of the variations controlled by DirectMusic, such as randomly or by shuffling, click in an area outside the part: in the timeline or another part.

- Repeat steps 1 through 13 to create variations in PreDawnA and DawnA. Copy parts as described in the following table:

Copy from:	Paste to:
PreDawnB Cello part	PreDawnA Cello part variation 2
DawnB Oboe part	DawnA Oboe part variation 2

15. Click the NightB pattern in the project tree, press the DELETE key, and then press ENTER or click **OK** to confirm the deletion.
16. Repeat step 15 to delete PreDawnB and DawnB.

The patterns now contain the proper sequences and variations. In the next topic, you adjust the properties of the style and patterns.

For more information, see the following topics in the online Help

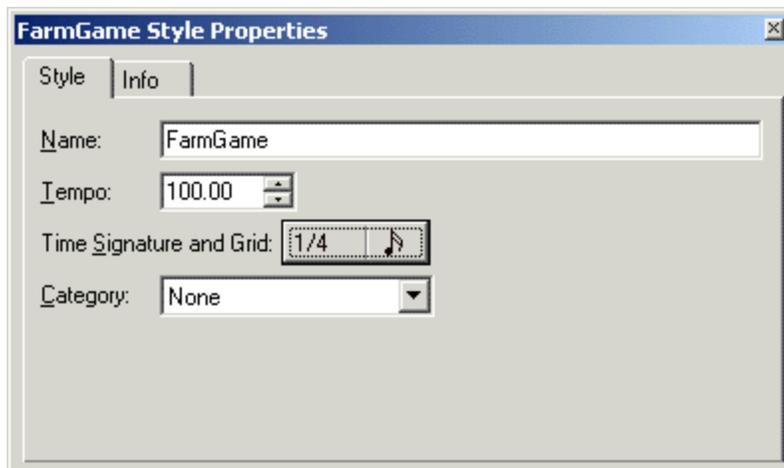
- Part Variations
- Variation Choices

Change pattern properties

In this section, you will make changes to the style and pattern properties so the music plays back properly.

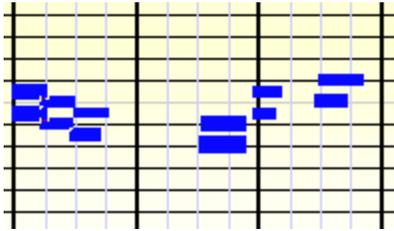
1. Right-click the Patterns folder in the project tree, and then click **Properties**. The **Style Properties** window opens.
The settings you make in the **Style Properties** window affect all the components of the style.
2. On the **Style** tab, enter a tempo of 100. You can type **100** or use the UP ARROW and DOWN ARROW buttons to change the numbers in steps.
3. Click **Time Signature and Grid**.
4. In the **Set Style Time Signature and Grid** window, enter a beats per measure setting of 1. Leave the beats setting at 4, and **Divide Beat** on the sixteenth note .

The following figure shows the **Style Properties** window properly configured.



The time signature of the music used in the tutorial varies, so to simplify notation the time signature has been set overall to one beat per measure. You can experiment with **Time Signature** tracks in a segment if you choose after the tutorial.

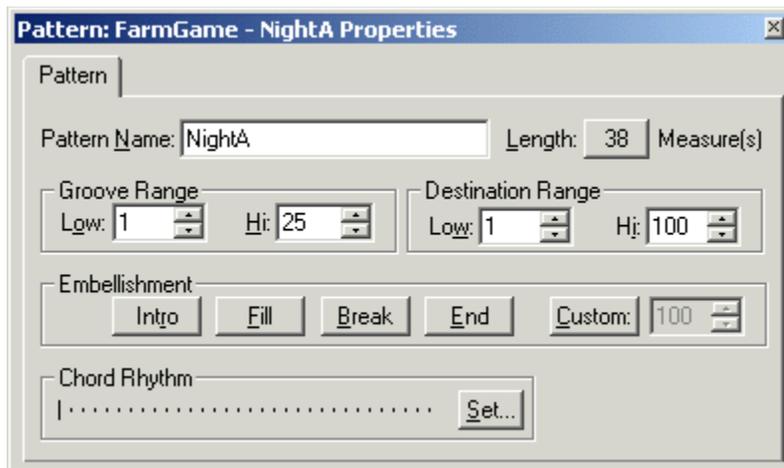
The **Grid** is set to sixteenth notes. When you view sequences in a pattern, you will see that each beat is divided into four sections—sixteenth notes. When you add or edit notes, you will use this grid to align notes to the correct sixteenth note.



5. With the **Style Properties** window open, click the NightA pattern, and the **Pattern Properties** window opens.

The settings you make in the **Pattern Properties** window affect only the selected pattern.

6. Click **Length**, and then enter **38**.
7. In the **Groove Range** area, enter **1** in the **Low** box and **25** in the **Hi** box.



Settings in the **Groove Range** and **Embellishment** areas establish how the pattern will be used. Clicking the **End** button, for example, labels the pattern as an ending. The groove range you set also gives the pattern a label. A low groove range can be used to label a pattern with a sparse arrangement or one that plays with a low intensity. You can set patterns with fuller arrangements or ones that play with higher intensities to higher groove ranges.

As you progress through the tutorial, you will see how these labels are used to build the DirectMusic components.

8. With the **Pattern Properties** window open, in the project tree, click PreDawnA. The window displays the PreDawnA properties.

9. Repeat steps 6 through 8 to set properties for the remaining patterns according to the following table:

Pattern	Measures	Groove Range	Embellishment
PreDawnA	38	26-50	None
DawnA	38	51-75	None
Ending	12	1-100	End
Transition	6	1-100	Intro

In the next topic, you add and modify individual notes in a pattern.

For more information, see the following topics in the online Help

- Pattern Properties
- Style Properties in the Project Tree

Add and edit notes in a pattern

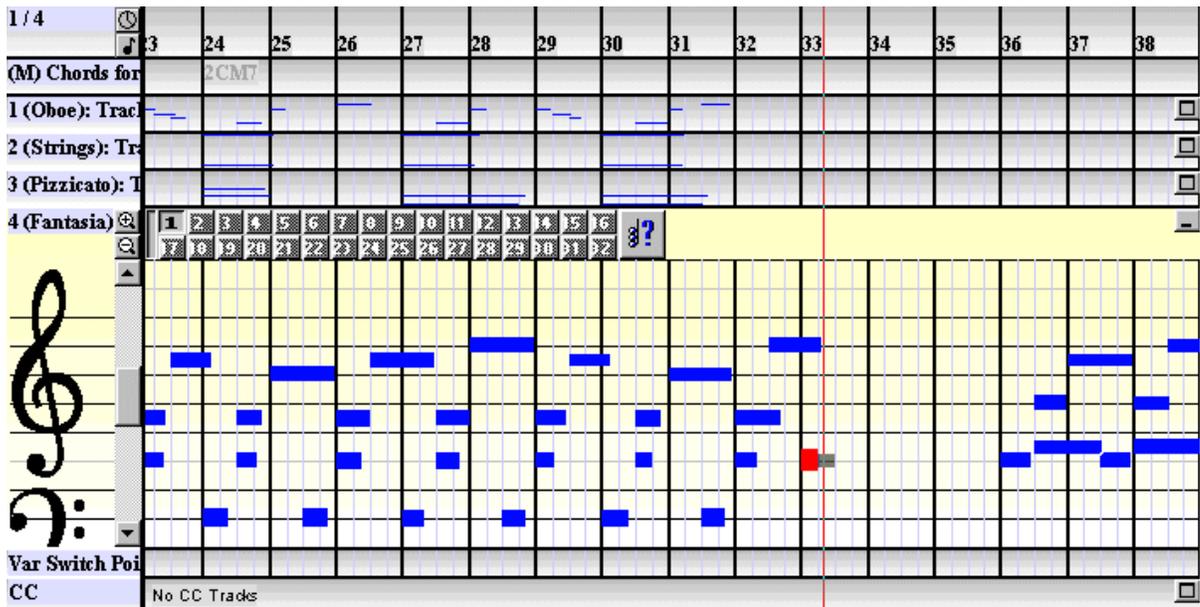
There are four ways you can add notes to a variation in a pattern. You can import or copy a MIDI sequence as you did previously, record a sequence while you play it in real-time with a MIDI keyboard, or insert notes manually with a mouse or the keyboard. After a sequence has been added to a part, you can use your mouse to move notes, change note length, and adjust velocity.

In this topic, you add notes to a section of a variation, and then edit the length and velocity of the notes.

1. Open the NightA pattern in the **Pattern Editor**, and click the **Maximize** button for part 4 (Fantasia).
2. Right-click the part name, point to **Views**, and then click **Hybrid Notation**. You can also press the H key to toggle between the **Hybrid Notation** and **Piano Roll** views.
3. Center the treble staff, if necessary, to view the sequence.
4. Use the horizontal scroll bar to scroll the sequence to beat 33.

Note that beats 33 through 35 are empty. In the following steps, you will insert the same notes that are in beats 36 through 38 into the empty beats. For example, the first note in beat 36 is a C5, so you will insert a C5 in beat 33. You do not need to know the names of the notes, because you can insert and align the notes visually.

- In the first quarter of beat 33, click C5 on the staff, and then press the INSERT key. The following figure shows C5 added to beat 33:



If you want to find out what a note is, you can right-click the note and view its properties. Use the zoom buttons  to expand the view if necessary.

- Repeat step 5 to add the remaining 7 notes to beats 33 through 35.

If you insert a note in the wrong position, you can drag it to a new position or select it and press the DELETE key.

- Click beat 32 in the timeline, and the play cursor moves to that position.
- Click the **Play** button, and the pattern plays from beat 32.



You should hear the notes in beats 33 through 35 repeated in beats 36 through 38. However, you may notice that the notes are of different lengths and velocities.

- To change the length of a note, move the mouse over the right edge of the note until the mouse pointer changes to horizontal arrows.
- Click and drag the edge to match the length of the corresponding note in beats 36 through 38.
- To change the velocity of a note, move the mouse over the note until the mouse pointer changes to vertical arrows.
- Click and drag the top edge up or down to change velocity accordingly to match the corresponding note in beats 36 through 38.
- Repeat steps 9 through 12 for the other notes. Note that it is not necessary for the lengths and velocities to match exactly.

Play the pattern from beat 32 to make sure the notes are the same, and the two sections sound natural when played together.

In the next topic, you will play back the patterns and finish setting up the style.

For more information, see the following topic in the online Help

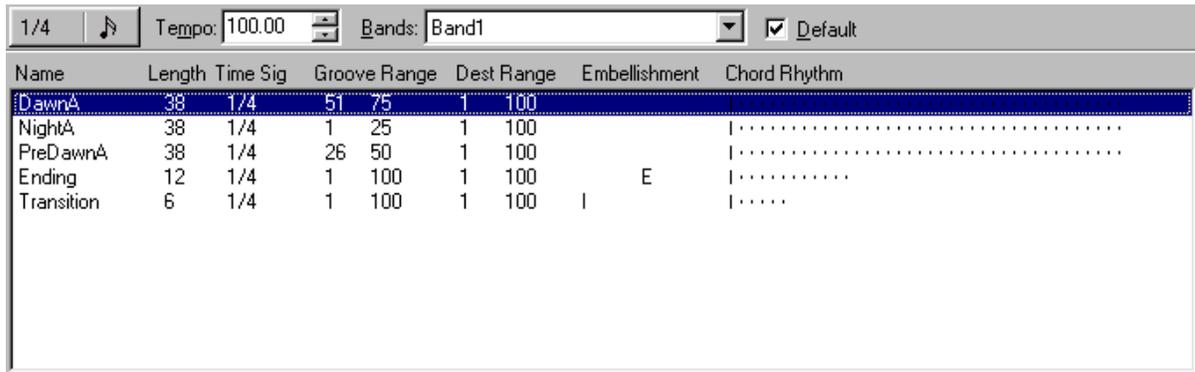
- Note Entry and Editing

Finish setting up the style

You can now play any of the patterns by opening one in the **Pattern Editor** and clicking the **Play from start** button. Notice that patterns are not necessarily musically complete. You can think of patterns as the fundamental building blocks from which a complete non-linear score is created. Patterns are components of styles, and styles are built into segments. Segments are the components that are played by an application.

- Double-click the FarmGame style component icon , which is directly under the FarmGame style file icon, and the **Style Designer** window opens.

Note There are two identical style icons: one representing the style file (FarmGame.stp), and one directly under it representing the style component that is contained in the file (FarmGame). You will also see two identical icons in other components, such as DLS collections, and segments. In most cases, you will view and change properties of the component. You will use a **File Properties** window to view and change properties relating to the file.



Name	Length	Time Sig	Groove Range	Dest Range	Embellishment	Chord Rhythm
DawnA	38	1/4	51 75	1 100	
NightA	38	1/4	1 25	1 100	
PreDawnA	38	1/4	26 50	1 100	
Ending	12	1/4	1 100	1 100	E
Transition	6	1/4	1 100	1 100	I

This window lists all the components of the style and their settings, as well as global style properties. You can double-click a component to open it in an editor window, or right-click a component to open a shortcut menu.

If you want, you can open Band1 in the **Band Editor** and mix the instruments. First, open a pattern in the **Pattern Editor**, and click the **Play from start** button. Then, open Band1 from the project tree and change the positions of the PChannel mixing buttons.

In a game or application, a DirectMusic score is created by the user. The job of the composer or sound designer is to supply the building blocks—the DirectMusic components. If you are successful in composing the variations and designing the components, the music will seem to play as one continuous score.

All the musical building blocks are in place: the instruments are assigned to the band, and the patterns are configured to play variable sequences. In the next lesson, you will assemble the building blocks into the segments that will be used in the game.

For more information, see the following topics in the online Help

- Style Designer Window
- Volume and Pan Graphical Interface

Lesson 4: Create primary segments

An application does not play components of a style directly. Styles must first be assembled into one or more segments. You can create segment files using DirectMusic Producer, or an application can be written to create segments from styles programmatically. In the tutorial, you will use the first method.

From the style components, you will create the following primary segments:

Name	Purpose
BGNight	Background music that plays during the night level.
BGPredawn	Background music that plays during the predawn level.
BGDawn	Background music that plays during the dawn level.

Primary segments are designed to play as stand-alone components. If you want to play two or more segments concurrently, you create one primary segment and multiple secondary segments. In lesson 5, you will create the following secondary segment that plays on top of the backgrounds:

Name	Purpose
SSBird	The secondary musical segment that plays over a background segment when a user initiates a bird flyby.

In Lesson 8, you will build secondary segments that contain wave files:

Name	Purpose
SfxAlarm	Plays over the background when user action starts the alarm clock.
SfxCougar	Plays over the background when a user clicks a cougar.
SfxCow	Plays over the background when a user clicks a cow.
SfxRooster	Plays over the background when a user clicks the rooster.
SfxSheep	Plays over the background when a user clicks a sheep.
SfxWolf	Plays over the background when a user clicks a wolf.

Note A common prefix naming convention is helpful for keeping like components together in the project tree, and when working with game developers and others on a project team.

For more information, see the following topics in the online Help

- Segment Designer
- Segment Tracks

Create empty music segments

1. To create a new segment, right-click the project name (FarmGame) in the project tree, and then click **New**.
2. In the **Create New Files** window, click **Segment**, clear the **Use Default Names** check box, and then type **BGNight**. The new segment is added to the project and the **Segment Length** window opens.
3. Click **Number Measures**, and then enter **38**. The **Add New Tracks** window opens with Band track selected.

- Press the CTRL key, and then click the following tracks to add them to the segment with the Band track: Chord, Groove, Marker, Style, and Tempo. Note that you can change any property, including the number of measures, at any time.

	1	2	3	4	5	6
1-32: Tempo	120.00					
1: Time Sig						
1: Chords (Key)						
1: Grooves						
1: Styles						
1: Bands						
1: Marker						

The new segment opens in the editing pane with seven tracks: the six you added and the **Time Signature** track, which is added by default.

- Repeat steps 1 through 4 to create the remaining two new primary segments: BGDawn and BGPredawn.

The FarmGame project tree should now contain a style, a DLS collection, and three segments:



Add style and groove data to the segments

To configure segments to play music or sounds, you add data to the tracks. You add it to the beginning of a track, and at other points in a track if you want properties of the music to change over time. For example, you can start with a tempo of 120, and then slow the music to 100 at bar 23 by inserting a new tempo in the Tempo track at that point.

To add data to a track, right-click in the track at the point where you want the data added, and then click **Insert**. To change data, right-click the data, and then click **Properties**. In both cases, a properties window opens into which you add or change settings. You can also copy or delete data, or drag the data icon to a new position in a track.

You will first build the BGPredawn segment, and then with information provided in a table, build the other segments following the same steps.

- Double-click the BGPredawn segment in the project tree. The empty segment opens in the editing pane.
- Right-click the first measure of the Styles track, and then click **Insert**.
- In the **Style Properties** window, select the FarmGame style.

Notice that the style's tempo, time signature, and band are added to the appropriate tracks. Band data is copied to a segment, so you can change band settings in a segment without affecting the original band.

- With **Style Properties** open, click the first measure of the **Grooves** track. The properties window changes to display Groove Level Properties.

- Enter a **Groove Level** of **26**. Make sure **None** is selected in the **Embellishment** list. Close the window.

	1	2	3	4	5	6
1-32: Tempo	100.00					
1: Time Sig	1 / 4					
1: Chords (Key)						
1: Grooves	(26)					
1: Styles	FarmGame					
1: Bands	Band1					
1: Marker						

Click the **Play from start** button. The segment plays a pattern in the FarmGame style with a groove range that covers a groove level of 26. In this case, it is the PredawnA pattern. At the end of the segment the music stops.

Note In the score, you chose to have each segment play only one groove level. However, a groove track can contain multiple groove and embellishment changes. For example, you could assemble one long segment that played a complete piece with an intro, multiple patterns, and an ending by simply entering groove data at the appropriate points in the groove track.

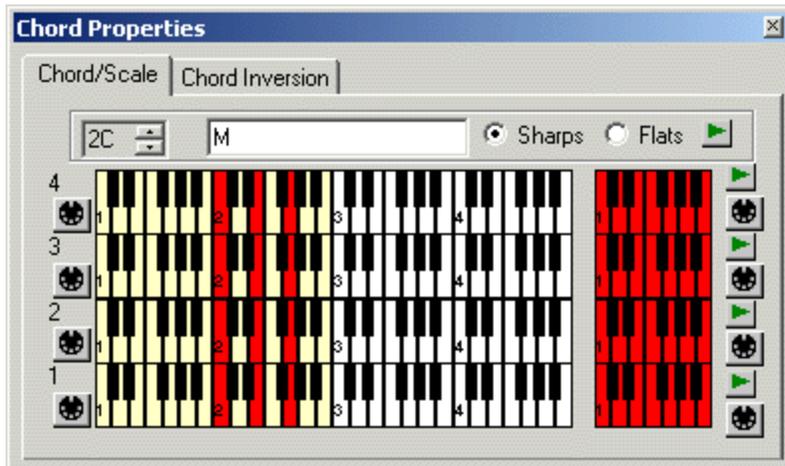
Add chord data to the segments

In the game, the music must modulate up when transitioning to higher levels. To accomplish that, you will transpose segments by adding chord data to the Chords track.

- With the BGPredawn segment open in the editing pane, right-click the first measure of the Chords track, and click **Insert**. The **Chord Properties** window opens.

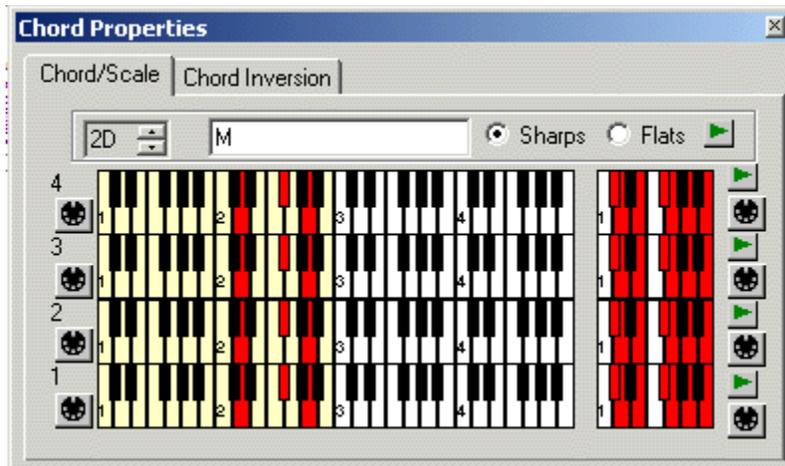
DirectMusic Producer imports sequences as C major, regardless of the actual key that a piece of music is written in. If you want a sequence to play in its original key, do not insert any chord data, or if you do, set the chord properties to C major.

When you insert chord data, the properties window opens for the first time with the default chord and C major scale.



The BGPredawn segment must play in the same scale transposed up one step from the BGNight segment. To accomplish that, you will leave BGNight in the default C major key and insert chord data in BGPredawn that transposes it to D major.

2. On the **Chord/Scale** tab, right-click anywhere in the keyboard, and then clear **Lock All Scale Levels**.
3. Click the UP ARROW button in the selection box to change the chord root name to 2D, and then close the window.



Click the **Play from start** button. The music plays one step higher. In the next topic, you will set segment properties.

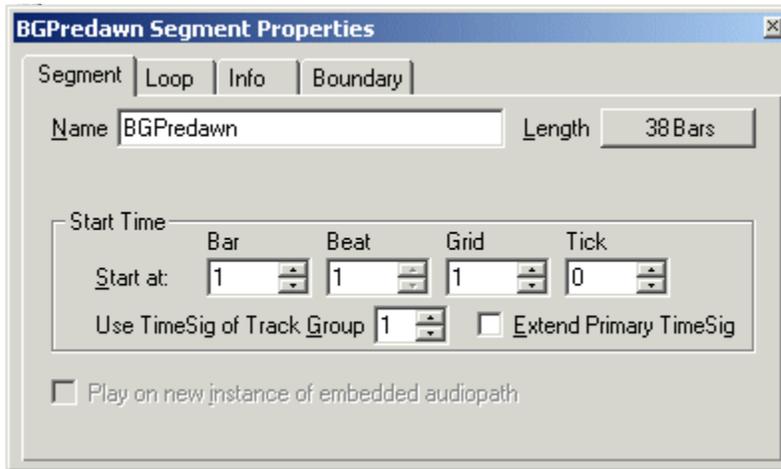
For more information, see the following topics in the online Help

- Chord Track
- Chord/Scale Tab

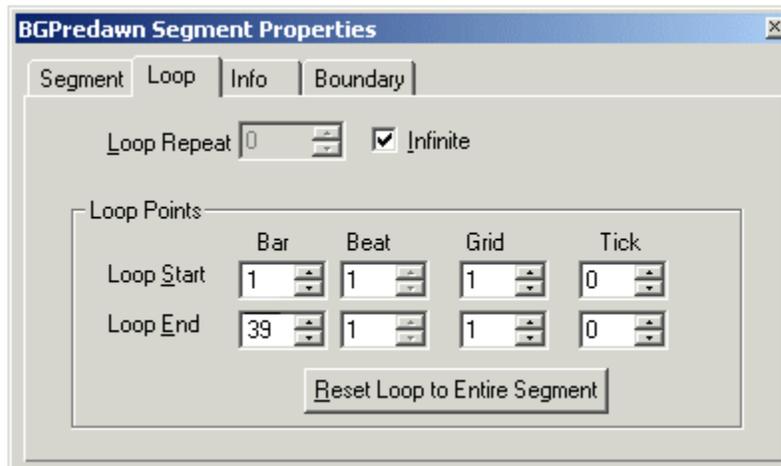
Set segment properties

Segment properties define global parameters for a segment. Properties affect the timing of a segment and how it works with other segments.

1. In the project tree, right-click the BGPredawn segment, and then click **Properties**. The **BGPredawn Segment Properties** window opens.



2. Click the **Loop** tab, and then select the **Loop Repeat Infinite** check box. When the segment plays, it will loop until it is stopped.



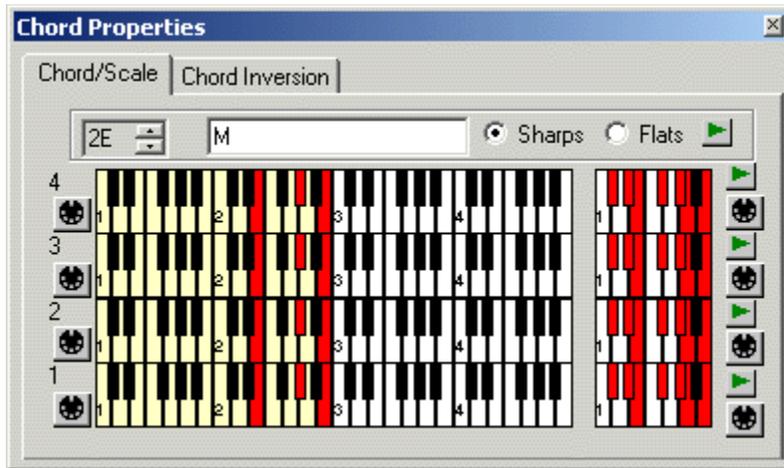
When the segment plays, it will start at bar 1, play up to bar 39, and then loop back to bar 1.

Click the **Play from start** button. The segment loops the Predawn pattern in D major and with all the correct parameters. If playback sounds wrong, go through the steps again and check all the settings.

3. If the segment plays as expected, repeat the steps in this section and the previous two sections (**Add style and groove data to the segments**, and **Add chord data to the segments**) for BGNight and BGDawn. All properties are the same except for the following settings:

Segment	Groove bar 1	Chord bar 1
BGNight	1	None, use default
BGDawn	51	2EM

The BGDawn segment uses an E major chord and scale. Remember to first right-click anywhere in the keyboard, and then clear **Lock All Scale Levels** before changing the chord to 2E. The Chord Properties window should resemble the following figure:



In the next topic, you will finish configuring the segments by adding markers to the marker tracks.

For more information, see the following topics in the online Help

- Segment Properties
- Loop Tab

Add markers to the Marker tracks

You add settings to a segment's properties that define when transitions from one segment to another can take place. Having the ability to define transition points is one very important tool for creating a smooth interactive score.

When a transition request is received from an application, such as when an event is triggered by some user action, DirectMusic waits for the next defined transition point in the segment before starting the transition. For example, you can set the segment to transition only on the beginning of a measure. Then, when a user triggers a transition interactively, DirectMusic will perform the transition at the start of the next measure. You can also add transition points with markers in a marker track. Markers enable you to manually denote specific times when you allow transitions to occur.

In creating the BGNight, BGPredawn, and BGDawn sequences, a time signature of 1 quarter note per measure was used rather than a variable time signature. This simplifies musical notation, but makes it more difficult to define transition points because transitions cannot happen naturally on any beat or measure. In order for transitions to sound natural, you will add the transition points manually using markers.

Data in a marker track are of two types: markers, which determine transition points out of segment, and enter switch points, which determine transition points into a segment. In the tutorial, we will use markers only.

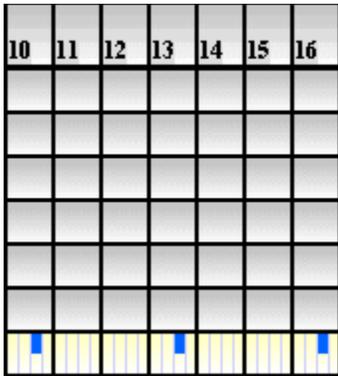
1. With the BGPredawn segment open in the editing pane, click in the top half of the first sixteenth note of beat one in the Marker track, and then press the INSERT key. Notice that the beats in the track are divided into four sections by a grid, the same as the beats in the pattern tracks.



The marker is added and the **Marker/Enter Switch Point Properties** window opens. Make sure to add the marker to the top half of the track. Data on the bottom half are enter switch points.

2. Without closing the **Marker/Enter Switch Point Properties** window, click in the top half of the first sixteenth note of beat 4, and then press the INSERT key.
3. Repeat step 2 to insert markers at the following positions: 7, 10.5, 13.5, 16.5, 21, 24, 27, 30, 33, and 36.

Notice that three of the markers occur between beats. Place these markers in the third sixteenth note of the beats, as shown in the following figure:



If you insert a marker in the wrong position, you can drag it to a new position or select it, and then press the DELETE key.

For more information, see the following topics in the online Help

- Marker Track
- Transition Options: Performance Boundary

Copy the Marker track

Rather than repeat the task of entering marker data in the BGNight and BGDawn segments, you can copy and paste the marker track from BGPredawn. You can use the same method to copy any track.

1. Click the **Selection** button for the Marker track, and then on the **Edit** menu, click **Select All**.
2. On the **Edit** menu, click **Copy**.
3. Open the BGNight segment in the editing pane.
4. Click the **Selection** button on the Marker track.
5. Press the HOME key to move the play cursor to the beginning of the segment.

Note that data is pasted starting at the position of the play cursor. If the cursor is not at the beginning of the pattern, the markers will be offset.

6. On the **Edit** menu, point to **Paste**, and then click **Merge**.
7. Repeat steps 3 through 6 for the BGDawn segment.

You will use the same copy procedure to create the SSBird secondary segment in the next lesson. In Lesson 7, you will learn how to test the transitions between segments.

For more information, see the following topics in the online Help

- Selecting Regions in Segments and Patterns
- Selected and Active Tracks

Lesson 5: Create a secondary segment

When you initiate a transition from one segment to another, the outgoing segment stops when the incoming segment starts. If you want to play two or more segments at the same time, you assign one of them as the primary segment and the others as secondary segments. When you want to play a secondary segment, you do not initiate a transition. The secondary segment plays with the primary segment, and the sounds of the two segments are mixed together.

In this section of the tutorial, you will create a secondary segment by importing a MIDI file directly into sequence tracks of the segment, and then cutting and pasting the sequences into variations in a pattern track. Pattern tracks are similar to the pattern components you created in the FarmGame style. Both provide a way for you to add variation, and both have their advantages. In general, you use patterns in styles. However, pattern tracks provide a simple way to add variation without having to create a style.

For more information, see the following topics in the online Help

- Primary and Secondary Segments

Create and configure the segment

1. Right-click the project name in the project tree, point to **Import**, and then click **MIDI File as Segment**.
2. Locate and open SSBird.mid.

A new segment is created containing a short sequence track with four parts. A part is an individual MIDI sequence that is assigned to an instrument. In this case, all four parts are assigned to PChannel 11.

	1	2	3	4	5	6
1-32: Tempo	100.00					
1: Time Sig	1 / 4					
1: Chords (Key=C/0#'s)	1CM7					
1: 11 (11): Sequence	[MIDI notes]					<input type="checkbox"/>
1: 11 (11): Track 12	[MIDI notes]					<input type="checkbox"/>
1: 11 (11): Track 12	[MIDI notes]					<input type="checkbox"/>
1: 11 (11): Track 12	[MIDI notes]					<input type="checkbox"/>

When you play the segment, you will notice that a default instrument plays. This occurs because there is no band assigned to the segment. When SSBird is played with one of the background segments, it will use that segment's band, style, tempo, groove level, chord, and any other property assigned to it.

When you play the segment, you will also notice that many of the notes collide. The music was composed using a third-party MIDI sequencer program that was not capable of creating pattern variations. Therefore, as a workaround, the variations were placed in individual parts of a MIDI sequence. In the following steps, you will add a pattern track, and then cut and paste the sequence tracks into the first four variations of the pattern.

3. Right-click in an open area of the editing pane, and then click **Add Tracks**.
4. Add a Band Track and Pattern Track.

- In the new pattern track, click the **Maximize** button. To view the complete Variation strip, click the **Zoom**  button in the lower right corner of the editing pane several times.

All the variations are active and enabled.



- Click variations 1 through 4 to deactivate these variations.



- Right-click anywhere in the variation strip, and then click **Disable**. All active variations (5-32) are disabled and deactivated.



In the next topic, you will move the parts from the sequence track to variations in the pattern track.

For more information, see the following topics in the online Help

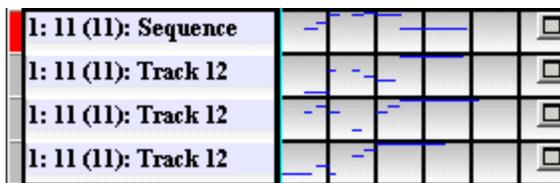
- Pattern Track
- Variation Buttons

Move sequence parts to the pattern track

- Click variation 1 to make it active. It should be the only active variation.



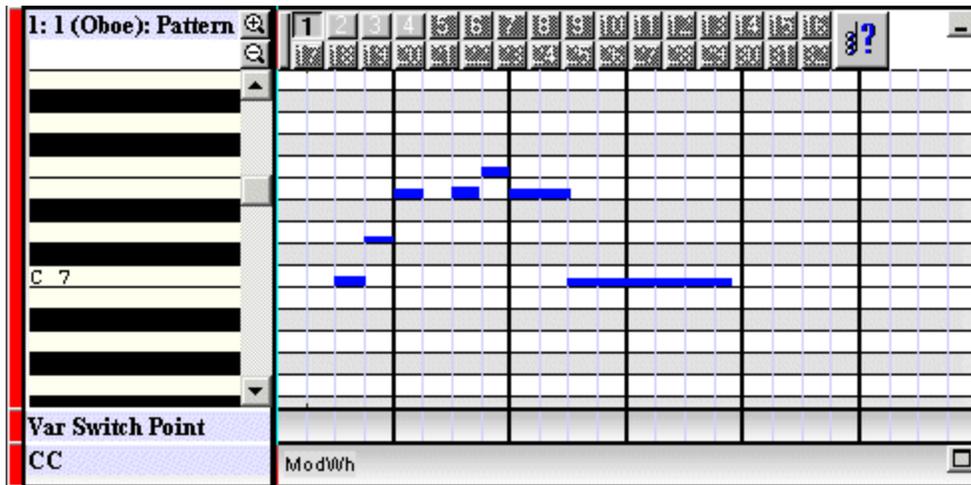
- Click the **Selection** button of the first part in the sequence track.



- On the **Edit** menu, click **Select All**, and then on the **Edit** menu, click **Cut**.
- Click the **Selection** button on part 1 to deselect it, and then click the **Selection** button on the pattern track.
- Press the HOME key to move the play cursor to the beginning of the pattern.

Note that data is pasted starting at the position of the play cursor. If the cursor is not at the beginning of the pattern, the inserted sequence will be offset and not start at the proper time.

- On the **Edit** menu, point to **Paste**, and then click **Merge**.



The sequence from part 1 is pasted into variation 1 of the pattern. If you do not see the notes, scroll the variation to display the C7 note in the Pitch index.

- Deactivate variation 1, activate variation 2, and then click the **Selection** button to deselect the pattern.
- Repeat steps 2 through 7 for the remaining three parts.
- Deselect the pattern track, and select the four parts of the sequence track.
- Right-click one of the selected parts, and click **Delete Tracks**.

In the next topic, you will finish configuring the segment.

Configure the properties

- Right-click the pattern track name, which is currently 1(Oboe):Pattern, and click **Properties**.
- In the **Pattern Track Properties** window, on the **Part** tab, **PChannel 11**.
- In the **Play** list, click **by shuffling**.
- To hear the variations played by DirectMusic, click the timeline, and then click the **Play from start** button.

The SSBird segment plays on PChannel 11, which has not been assigned to an instrument. In the next Lesson, you will create an instrument for the segment from a sample that has been saved as an audio file with a .wav file name extension. The other instruments in the band are from the default GM DLS collection. The new instrument you create will be added to the band as a DLS instrument.

For more information, see the following topics in the online Help

- Sequence Track
- Pattern Track

Lesson 6: Create a DLS instrument

All DirectMusic instrument sounds are samples. When DirectMusic plays a note, it is actually playing a small wave sample, which in many cases is a recording of an acoustical instrument. A complete GM/GS collection of instruments is installed on a user's computer with the installation of DirectMusic. You can use this default collection with any DirectMusic Producer score.

DirectMusic Producer also enables you to import your own instrument samples or for that matter any file with a .wav file name extension. For example, you could create a custom collection that included instruments not found in the GM collection, or import a sound effect, or even an entire musical riff. You can also use the **DLS Designer Window** to fine tune how the samples play.

In the tutorial, you will import an ocarina sample, adjust parameters with the **DLS Designer Window**, and then add the instrument to Band1 in the FarmGame style.

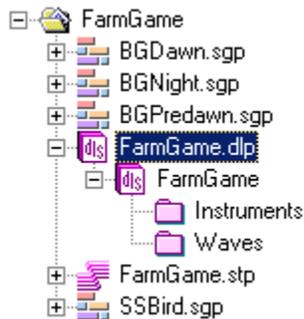
For more information, see the following topics in the online Help

- Synthesis and Downloadable Sounds
- Managing DLS Collections

Import a wave file into the FarmGame DLS

When you initially set up the FarmGame project, you created a new DLS collection.

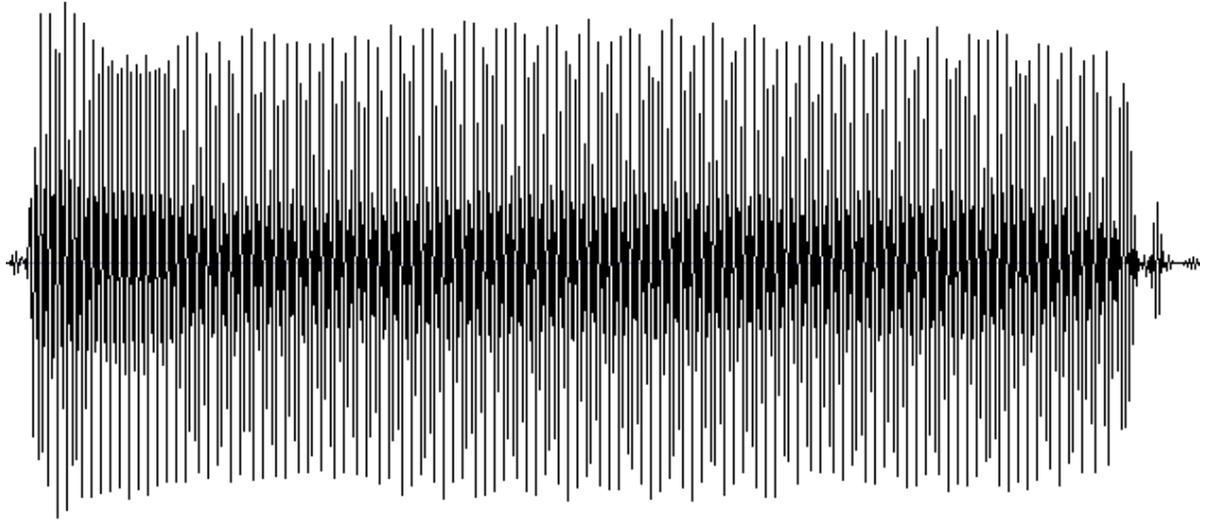
1. Expand the FarmGame.dlp folder in the project tree.



Inside the DLS folder are two folders: Waves and Instruments. Waves holds the sound files with .wav file name extensions, and Instruments holds the instruments that you create from the sound files and their settings.

2. Right-click the Waves folder, and then click **Insert Wave**.
3. Locate the folder containing the tutorial audio files, and then open Ocarina.wav. By default, the files are located in C:\My Documents\DMUSProducer\Tutorial\RawElements.
4. Expand the Waves folder in the project tree, right-click the wave name, and then click **Rename**. Rename the wave Ocarina.

5. Double-click Ocarina. The file opens and displays the waveform of the sound in the **Wave Editor**.



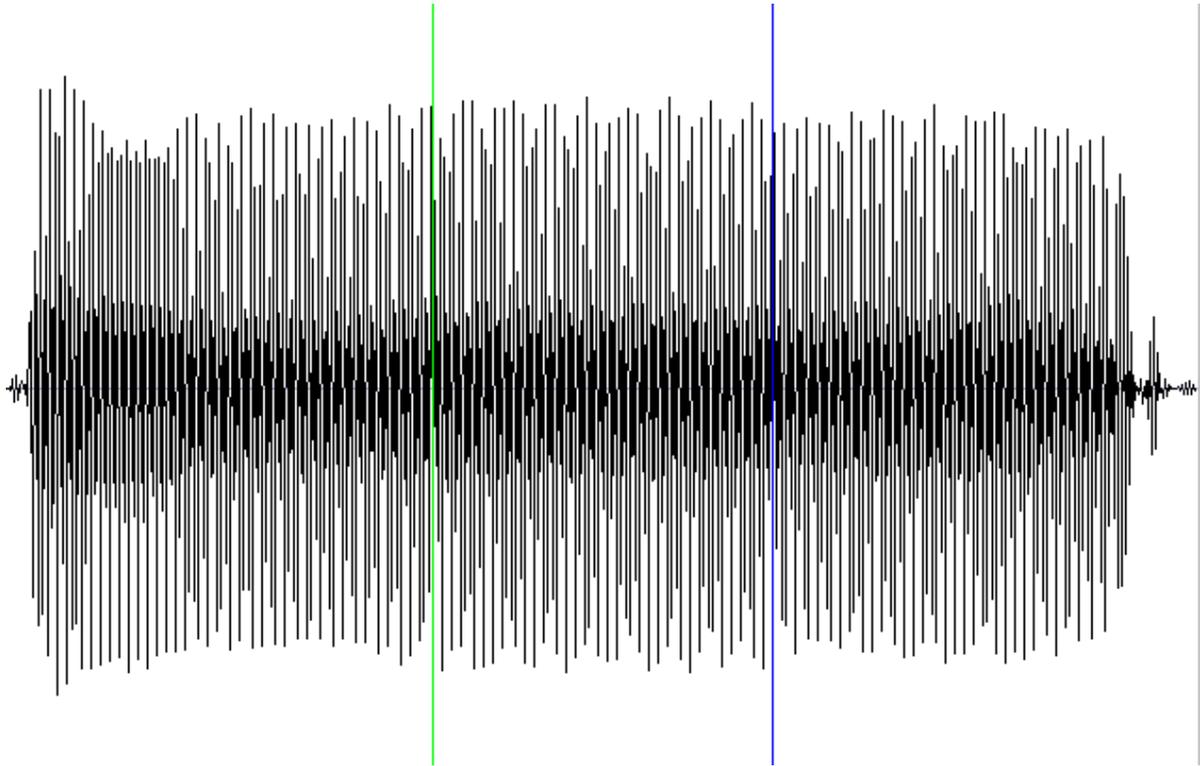
6. Click the **Play from start** button to play the sample.

For the ocarina to work in the tutorial, the sample must be configured to loop. Rather than spend time locating loop points in the tutorial, you can enter them manually in the following steps.

7. In the project tree, right-click Ocarina, and then click **Properties**.
8. On the **Wave** tab, click the **Loop** check box to select it, and then enter the following settings:

	Start	End
	29198	52373
Selection	0	0

Notice that green and blue loop point lines are added to the wave display.



Click the **Play from start** button. The wave plays from the start of the sample to the blue line (loop end), moves back to the green line (loop start), and then loops continuously between these two points until you click the stop button.

In the next topic, you will add the wave to the DLS collection as an instrument.

For more information, see the following topics in the online Help

- Wave Editor
- Wave Component Properties

Add a new instrument

1. In the FarmGame DLS folder, right-click the Instruments folder, and then click **Insert Instrument**. An empty instrument is added to the folder.
2. Expand the Instruments folder, if necessary. Right-click the instrument icon , and then click **Properties**.
3. In the **Instrument Properties** window, type the new name, **Ocarina**. Note that other text information can be added to the instrument.
4. Double-click the Ocarina instrument, and the **DLS Designer Window** opens.

In this window, you will add the ocarina sound to the ocarina instrument and configure performance settings.

Configure the instrument

Many settings can be made in the **DLS Designer Window** to fine-tune how an instrument plays. However, for the tutorial, you will only enter the most essential settings.

1. Ocarina should be selected by default in the **Wave** list. The **Wave** list is used to assign a wave file to a note range in an instrument.
2. Change **Root** to C#7.

The ocarina was recorded playing a C. However, the tuning is approximately one-half step sharp. To compensate, the root is entered as C#7. After the tutorial, you can further adjust the tuning in the **Wave Properties** window.

3. Enter a **Note Range** of C6 to D#9. You can also set the note range by dragging the ends of the **Region Bar**, which is located above the keyboard.

Setting a note range restricts the range of notes over which a particular wave sample plays. An instrument can be configured to use multiple wave samples that are each assigned to different note ranges.

The screenshot displays the DLS Designer Window for an instrument named 'Ocarina'. At the top, a keyboard graphic shows a yellow 'Region Bar' spanning from C6 to D#9. Below the keyboard, the 'Instrument' section includes controls for MSB (0), LSB (1), Patch (0), Drums (unchecked), and Max Level (00.0 dB). The 'Audition Options' section has Click Velocity (100) and Solo Layer (unchecked), Multiple Layers (checked). The 'Region' section shows Wave (Ocarina), Root (C#7), Condition (<None>), Note Range (C6 to D#9), Velocity Range (0 to 127), Note Overlap Allowed (checked), Use Instrument Articulation (checked), and Key Group (No Group). The 'Instrument Articulation' section features buttons for Volume, Pitch, Mod LFO, Vib LFO, and Filter, with DLS1 checked. A graph shows envelope curves for Delay (00.001), Attack (00.001), Hold (00.001), Decay (00.001 to 00.000), Sustain (100.0%), and Release (00.001 to 00.001). On the right, sliders control Shutdown Time, Vel to Attack, Key to Hold, Key to Decay, and Pan (Mid).

To play the ocarina instrument, click any note on the keyboard within the note range. The note plays as long as you hold the mouse button down. You can also play the notes with a MIDI keyboard connected to your computer.

For more information, see the following topics in the online Help

- DLS Designer Window

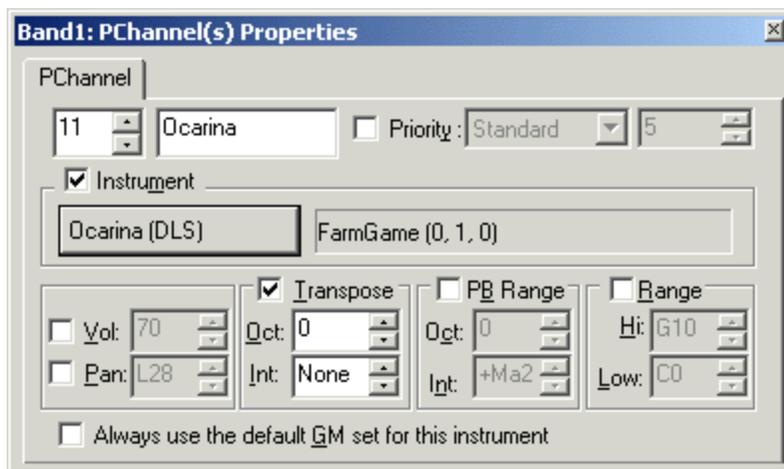
Add the instrument to the band

DLS collections contain the instruments and their sounds. They are separate from the components that contain the music: the styles and segments. With the two separate, you can create a DLS for one project, for example, and then later import the DLS into another project.

You make DLS instruments available to a style or segment by adding them to a band.

1. In the FarmGame style Bands folder, double-click Band1, and the **Band Editor** opens.
2. In the **Performance Channel** list, right-click PChannel 11, and then click **Properties**.
3. In the **Band Properties** window, change the PChannel name to Ocarina.
4. Select the **Instrument** check box, and then click the **Instrument patch assignment** button.
5. On the Instrument Family menu, click **Other DLS**.
6. In the **Choose DLS Instrument** window, click **Ocarina**.

If the project contained more DLS collections or a collection contained more instruments, they would appear in the lists. The PChannel properties window should resemble the following figure:



7. In the **Volume and Pan Graphic Interface**, drag the PChannel 11 icon to the center with the other icons.

You can now add Band1 to a style or segment. When the SSBird segment plays, it will inherit Band1 from the primary segment and play the ocarina.

In a previous lesson, you added the FarmGame style to each segment. When you did so, certain style attributes were added to the segment and the band was copied to the band tracks. Because each segment contains a copy of the original Band1, whenever you make a change to a band in one place, you must copy the updated band data to all the segments that use the band otherwise the changes will not be heard.

8. Open the BGPredawn segment.
5. Click Band1 in the FarmGame style Bands folder, drag it to the BGPredawn segment, and then drop it on the Band1 data in the first measure of the Bands Track.
6. Repeat steps 8 and 9 for all the other segments, including SSBird. Even though SSBird uses the band it inherits from the primary segment, you can add a band to the segment to help you test playback.

Note If you do not hear the ocarina instrument when SSBird plays over a primary segment, try updating Band1 in the primary segment.

All of the musical components have been created. In the next lesson, you test the segments to hear how the transitions and secondary segments work.

For more information, see the following topics in the online Help

- Band Editor Window
- PChannel Properties Window

Lesson 7: Test playback

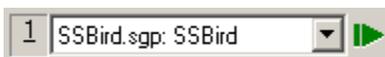
When you create a linear music score, you compose one continuous piece. With non-linear music, on the other hand, you must not only compose multiple pieces, you must consider how all the pieces work together in a number of ways. With linear music, there is only one playback path: from the beginning of the music to the end. With non-linear music, there can be many paths—even a near infinite number. To produce non-linear music that plays correctly no matter which paths a user takes, you need to test a number of combinations.

For the tutorial, and for most of the music you create, you will test two types of interactions between components:

- **Transitions.** How the transitions work from one primary segment to another. For example, if you have set a segment to transition on the beat, you need to be able to listen to all of those transitions to determine whether they work. If a user can transition to a number of segments, you need to be able to test each one. To test transitions within DirectMusic Producer, you can use the **Transition** button.



- **Secondary segments.** How secondary segments work as layers over the primary segments. In the game, you need to know how the SSBird segment sounds as it is played with each background segment. To test secondary segments, you can use the **Secondary Segment Toolbar**.



For more information, see the following topics in the online Help

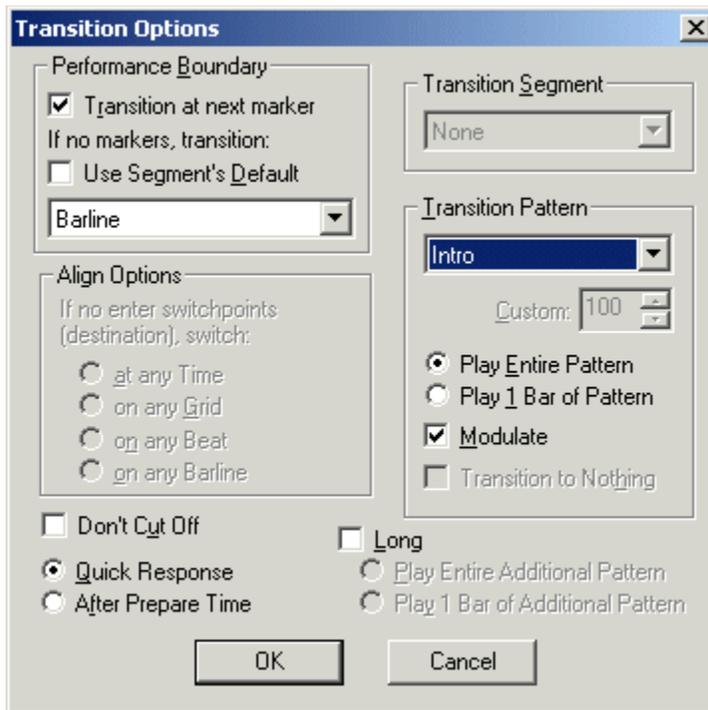
- Transitions
- Secondary Segment Toolbar

Test primary segment transitions

1. Open the BGNight segment in the editing pane.
2. Right-click the **Transition** button. The **Transition Options** window opens. Note that the **Transition** button is not enabled unless a segment is playing, but you can still right-click the button to open the window.
3. Select the **Transition at next marker** check box, if necessary.

Settings in the **Performance Boundary** area define when a segment transitions from another segment. With **Transition at next marker** checked, the segment waits for the next marker in the segment that is currently playing before making a transition.

4. In the **Transition Segment** list, click **None**, if necessary.
5. In the **Transition Pattern** list, click **Intro**, and then click **OK**.



6. Expand the BGDawn, BGNight, and BGPredawn folders in the project tree.
7. Click the BGNight component under BGNight.sgp, and then click the **Play from start** button.
8. While BGNight plays, click the BGPredawn component in the project tree.
9. At any point, click the **Transition** button.

At the next marker, the music transitions to the transition pattern (Intro), which in this case is the pattern named Transition. After the intro, the BGPredawn segment plays. Stop the music and test the transitions at various points in the BGNight segment.

Note that the intro uses the correct chord and scale. This occurs because patterns labeled as Intro use the chord and scale of the segment to which they are transitioning, or their destination segment.

10. Repeat steps 7 through 9, and test the transition from BGPredawn to BGDawn.
11. Open the **Transition Options** window. Then, change the **Transition Pattern** to End, and select the **Transition to Nothing** check box.
12. Test the End pattern with each of the background segments.

Note that like Intro, the End pattern uses the correct chord and scale. This occurs because End patterns use the chord and scale of the segment from which they are transitioning, or their source segment.

If transitions do not work as expected, make sure that markers are properly placed, and **Transition at next marker** is selected in the **Transition Options** window.

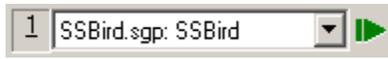
After you are satisfied with the transitions, you can test the secondary segment.

For more information, see the following topics in the online Help

- Transition Options Dialog Box

Test the secondary segment

1. Open the BGPredawn segment.
2. In one of the **Secondary Transport Control** lists in the **Secondary Segment Toolbar**, click the SSBird segment.



3. Right-click the **Play from start** button next to the secondary segment list. The **Secondary Segment Audition Options** window opens.
4. In the **Performance Boundary** list, make sure **Beat** is selected.



Play the BGPredawn segment. At any point, play SSBird. Notice that SSBird uses the primary segment’s style, chord, band, and tempo. Open BGDawn and play SSBird. Notice that SSBird modulates up to match the BGDawn’s E major chord and scale. Change the **Performance Boundary** setting to play on markers, and notice how the music works.

In the next lesson, you can turn to the simpler task of adding the sound effects to the game.

For more information, see the following topics in the online Help

- Secondary Segment Audition Options

Lesson 8: Make wave segments

The last DirectMusic components you will create for the game are the sound effects. You could import the sound files into the FarmGame DLS, as with the ocarina, and play them in patterns or motifs. If you did that, you would have control over many attributes of the sounds, such as the pitch and envelope. However, in this case, you do not need that much control.

The easiest way to deal with sound effects is to import them as wave tracks in segments. Then they can be played as secondary segments, the same as SSBird.

For more information, see the following topics in the online Help

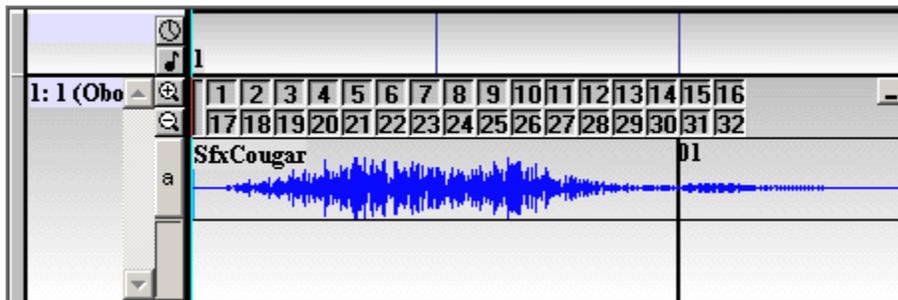
- Wave Track
- Wave Track Properties

Import Wave files and test the segments

1. Right-click the project name in the project tree, point to **Import**, and then click **Wave File as Segment**.
2. Locate SfxAlarm.wav. Click it, then press the SHIFT key and click SfxWolf.wav. All six sound effects files are selected.

The .wav files are imported and six wave segments are created.

3. Open the SfxCougar segment.



Notice that the wave has been pasted into 32 variations. You can work with wave tracks the same as pattern tracks. For example, you can add variability to the cougar sound by placing different cougar growls into several variations.

4. Right-click the part name **1 (Oboe)**, and then click **Properties**.
5. On the **Wave Part** tab, enter **PChannel 12**, and then type **SFX** as the name for the PChannel.
6. Repeat steps 3 through 5 for each wave segment.

By assigning the waves a unique PChannel, you can change PChannel properties, such as volume and pan, which affect the wave sounds only. You can also create a unique audiopath for the waves, which you will do in the next lesson.

To test the sounds with a primary segment, select one of the wave segments in a **Secondary Segment** toolbar, and then open the **Secondary Segment Audition Options** window and click **Immediate**.

Open and start playing BGPredawn, and then click the **Play from start** button in the **Secondary Segment** toolbar.

You may have noticed that the farm animal sounds seem unnatural because they have the same reverb effect as the music. In the next lesson, you create an audiopath in the project that makes sound effects play back dry (without the reverb effects). You will also add an echo effect to SSBird.

For more information, see the following topics in the online Help

- Wave Variations
- Secondary Segment Toolbar

Lesson 9: Create an Audiopath

The Microsoft Synthesizer creates and mixes the sounds of the instruments and other waves into one stereo output signal. The signal by itself is complete, but it is dry, meaning that it contains no effects or processing. You can add effects by routing all the sound through one of the default audiopaths, such as **Standard Stereo and Reverb**. However, if you want to process sounds individually, and customize the effects and configuration, you can create a custom audiopath.

In this lesson, you will build and configure a custom audiopath for the FarmGame project that patches the ocarina through an echo effect, and removes any effects from the sound effects. The other instruments will be routed through **Standard Stereo and Reverb** by default.

Set up the Mix Groups

1. Right-click the FarmGame project in the project tree, and then click **New**.
2. Click **Audiopath**, clear the **Use Default Names** check box, and then type **APFarm**.
3. Click **Create an empty audiopath**. The **Add/Remove PChannels** window opens.
4. Enter **PChannel 11**, click **Add**, and then click **OK**. The **Audiopath Designer** window opens with one Mix Group assigned to PChannel 11: Ocarina.
5. Right-click in an empty area of the window, and click **Insert Mix Group**.
6. In the **Add/Remove PChannels** window, add PChannel 12, and then click **OK**. A second Mix Group assigned to the SFX PChannel is added to the audiopath

In the next topic, you will add and configure an effect.

Add and configure effects in the Mix Groups

The **Effect Palette** contains the effects that you can add to a Mix Group. After adding one or more effects, you can open property windows and configure the effects.

1. Drag Echo to the Effects List column of 11: Mix Group. An echo object is created with the name Echo1.
2. Right-click Echo1, and then click **Properties**. The **Echo1 Effect Properties** window opens.
3. Click **Edit Effect Properties**.

- In the edit window, enter the following settings, and then close both windows:

Setting	Value
Wet Dry Mix	30
Feedback	0
Left Delay	300
Right Delay	600

The Mix Group assigned to PChannel 11 patches the ocarina through the Echo1 effect; and the PChannel 12 Mix Group patches the wave sounds through no effects, so the sound is dry.

In the next topic, you will add the audiopath to the appropriate segments.

Add the audiopath to the segments

There are several ways to apply an audiopath to a DirectMusic score. You can apply one or more audiopaths programmatically through the game or interactive application, or with a script file. You can also apply an audiopath by selecting it in the **Transport Options** toolbar, but this method only applies the audiopath while you are working with the segments in DirectMusic Producer.



The method you will use inserts an audiopath directly into the segments. By building an audiopath into a segment, the correct effects will be used no matter how or in what environment the segment is played.

- Right-click the SSBird component, point to **New**, and then click **Audiopath**.
- Click **Create a new audiopath based on an existing audiopath**, and then click APFarm.aup: APFarm. APFarm is added to the segment.
- Open the properties window for SSBird, and click the **Segment** tab. The **Play on new instance of embedded audiopath** check box is selected when the audiopath is added. This selection associates the audiopath with the segment.
- Repeat steps 1 and 2 for each of the sound effects wave segments: SfxAlarm, SfxCougar, SfxCow, SfxRooster, SfxSheep, and SfxWolf.

Finally, add the **Standard Stereo & Reverb** audiopath to the background segments.

- Right-click the BGDawn component, point to **New**, and then click **Audiopath**.
- Click **Create a new audiopath based on an existing audiopath**, and then click **Standard Stereo & Reverb**.
- Repeat steps 5 and 6 for BGPredawn and BGNight.

Play the segments to hear how the effects work. After the tutorial, you can experiment with adding more effects and changing the configuration. With audiopaths, you can easily create a complex and rich soundscape.

With all the musical and sound components finished, you begin to package the components to play in the game application by creating a script file.

For more information, see the following topics in the online Help

- Audiopath Designer
- Creating an Audiopath

Lesson 10: Create a script file

After you create DirectMusic components, you must communicate with the application developer your vision of how the pieces should work in the application. This is not always easy to do. There may be hundreds of segments, motifs, and chord, band, and groove level changes that must be carefully interwoven into a game. To help the application developer and give you more control over how your DirectMusic components play, you can create a script file. Instead of writing a complicated chord change into the game code, for example, you can write a routine in a script file that the game developer calls with one simple line of code.

In the tutorial, you will use a script file as a simple interface for the music and sound effects.

For more information, see the following topics in the online Help

- Script Designer
- AudioVBScript Language

Create a new script file and add references

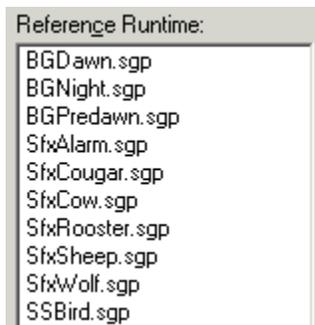
1. Right-click the project name in the project tree, and click **New**.
2. Click **Script**, clear the **Use Default Name** check box, and then type the name **FarmMusic**. The **Script Designer** opens with an empty script.

Before entering the script, you will add references that link the script to the DirectMusic components. When a method is called in a script routine, the script engine uses the reference to locate the component.

3. Expand the FarmMusic folder in the project tree, and then expand the Script Content folder.

You can embed the components in the script file or add references to the components. In the tutorial, you will add references

4. In the project tree, right-click Reference Runtime, and then click **Add/Remove Files**.
5. Select the following files in the **Available Files in Project** list, and then under **Reference:**, click **Add>>** to add them to the **Reference Runtime** list. To select multiple files simultaneously, press the CTRL key and click the files.



Add the script

Scripts can be written using any Microsoft ActiveX® Scripting language, but DirectMusic Producer enables scripting in a special language called AudioVBScript. AudioVBScript is a subset of Microsoft Visual Basic® Scripting Edition (VBScript), therefore any script in AudioVBScript can be run in the VBScript environment as well.

In the tutorial script, you will use the general Visual Basic Scripting Edition language instead because the script uses a statement block that is not supported with AudioVBScript.

The FarmMusic script contains three types of statements: routines, declarations, and comments.

- **Routines.** Blocks of script that start with **sub** [*name of routine*], and end with **end sub**. Routines contain instructions that perform tasks. In the FarmMusic script, the **Play** method is used in most of the routines to play segments.
- **Declarations.** A statement, often beginning with the keyword **dim**, that introduces and defines new names for the script runtime program. In the FarmMusic script you will use the variable *PlayFlag* to keep track of which background segment is currently playing.
- **Comments.** One or more words on a line following an apostrophe. You can add comments to a script that serve as reminders and to make the script easier to read.

For more information, see the documentation for Visual Basic Scripting Edition.

1. In the **Source** pane, type the following script or copy the script from the text file FarmMusic.txt, which is located in the RawElements folder.

```
' This script creates an interface for the Farm Game score.

' These three routines play the Intro at the next marker, and
' then the background segment. Any notes already playing will
' not cut off. The PlayFlag variable keeps track of which segment
' is playing so it can be stopped.

dim PlayFlag

sub dmBGNight
    BGNight.play(PlayIntro)
    PlayFlag = 1
end sub

sub dmBGPredawn
    BGPredawn.Play(PlayIntro+AtMarker+NoCutOff)
    PlayFlag = 2
end sub

sub dmBGDawn
    BGDawn.Play(PlayIntro+AtMarker+NoCutOff)
    PlayFlag = 3
end sub

' Stops background segments

sub dmAllStop
    Select Case PlayFlag
        Case 1
            BGNight.stop
        Case 2
            BGPredawn.stop
        Case 3
            BGDawn.stop
    End Select
end sub

' Plays the end pattern, then stops

sub dmEnding
    BGNight.Play(PlayEnd+AtMarker+NoCutOff)
```

```

    BGNight.stop(AtImmediate)
end sub

' Plays SSBird as a secondary segment and starts on the beat

sub dmSSBird
    SSBird.Play(IsSecondary+AtBeat)
end sub

' These routines play the sound effects as secondary segments

sub dmSfxCow
    SfxCow.Play(IsSecondary)
end sub

sub dmSfxSheep
    SfxSheep.Play(IsSecondary)
end sub

sub dmSfxCougar
    SfxCougar.Play(IsSecondary)
end sub

sub dmSfxWolf
    SfxWolf.Play(IsSecondary)
end sub

sub dmSfxAlarm
    SfxAlarm.Play(IsSecondary)
end sub

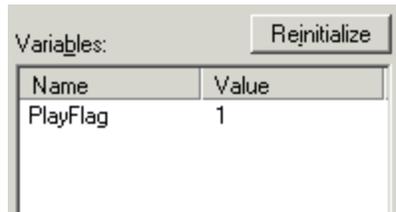
Sub dmSfxRooster
    SfxRooster.Play(IsSecondary)
end sub

```

2. Right-click the FarmMusic script component in the project tree, and then click **Properties**.
3. In the **Language** list, click **VB Script**, and then close **Properties**. In most cases, you can use the AudioVBScript language. However, in this case, you are using the **Select Case** script statement block, which is not supported with AudioVBScript.
4. Click **Refresh**, if necessary and the routines appear in the **Routines** pane.



If the script uses variables, the **Variables** pane displays the value of each one as the script runs. In the following figure, the dmBGNight routine was run and *PlayFlag* was set to 1.



If you change part of the script, you may have to click the **Refresh** button to update the routines. The **Reinitialize** button sets all variables to <empty>.

Routines and variables do not appear in the panes if there is an error in the script, such as a misspelling or a reference made to a non-existent component. To locate errors, you can use the **Message Window**.

For more information, see the following topics in the online Help

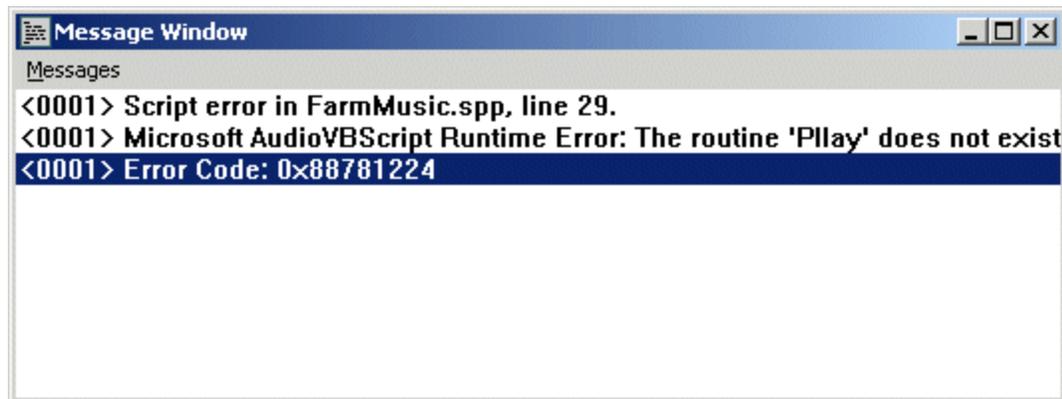
- DirectMusic Scripting Reference
- Script Designer

Open the Message Window and test playback

The **Message Window** displays errors to help you debug a script.

- On the **Add-Ins** menu, click **Message Window**.

A message is generated each time you attempt to run a routine that contains an error.



Double-click dmBGNight in the **Routines** pane. The BGNight segment plays. You can double-click the other routines to test transitions between primary segments and to hear how the secondary segments work. When the game runs, the program will call routines in the script. By testing playback in the **Script Designer**, you hear exactly how the music will sound in the game.

Note that you can run the dmEnding routine from any background segment and the script transitions to the end pattern in the BGNight segment with the correct chord and scale. This works because the ending pattern is part of the style that is assigned to all three segments. The chord and scale are correct because a pattern labeled as an End embellishment uses the chord and scale of the source segment.

Double-click dmAllStop and the background segment that is currently playing stops.

With a script file, the game developer does not need to understand the technicalities of music or possess a musical ear. The developer simply calls the routines in the game application, and the script handles playback. The last step in creating DirectMusic components is exporting the runtime files that are used by the game application.

For more information, see the following topics in the online Help

- Message Window
- Performance.Trace

Lesson 11: Export runtime files and test the game

Before the DirectMusic components can be used by an application, they must be converted into a form that can be read by the DirectMusic runtime engine, and then packaged into files. This process is performed automatically by DirectMusic Producer.

1. Click the FarmGame project folder in the project tree.
2. On the **File** menu, click **Runtime Save All Files: Project FarmGame**.
In the FarmGame folder on your hard disk drive, a RuntimeFiles folder is created containing all the runtime files.
3. Use Windows Explorer to open the Farm game player application resources folder, by default C:\My Documents\DMUSProducer\FarmGameApp\Res, and then delete all the files except for Ui.bmp.
4. Locate the RuntimeFiles folder, which is located in your DirectMusic Producer project folder, and then copy all the files to the Res folder.

For the game application to locate the runtime files, you must not move or rename the Res folder. The Res folder must be in the same folder as the game executable file Farm.exe.

For more information, see the following topics in the online Help

- Delivering Content
- Containers

Test the music in the game

The DirectMusic Farm Demo audio player is designed to test your DirectMusic score in the context of a simple application. The audio player can be found in the FarmGameApp folder. Double-click Farm.exe or click **Start**, click **Run**, and then enter the path to Farm.exe to open the player.



You can play the sound effects and secondary segment at any time. After you start a background, you can play the end. Click the **All Stop** button if you want the current background segment to stop immediately.

You are finished with the tutorial after you have successfully tested playback of the DirectMusic components in the game audio player.

The next step in learning how to use DirectMusic Producer is to go back through the tutorial at various points and experiment with new settings and configurations. For example, you can try importing your own music into patterns, adding variations, adjusting the envelope and pitch of the ocarina, and trying different transition boundaries.

If you are a developer, we suggest reading the DirectX Audio section of the DirectX 8 SDK to learn how to add DirectMusic to an application.

If you are a composer or sound designer, we suggest you look through the rest of the Help documentation to get ideas for moving forward with DirectMusic. In addition, go through the Demo8 project (located in the DMUSProducer folder) to see and hear demonstrations of the most important features of DirectMusic. Then create a new project using your music and samples.

For more information, see the following topics in the online Help

- Documentation Roadmap
- Scenarios