

Chapter 7: Sound Supplement

Copyright © 2013, 2015 by Charles Kelly

7.12 Audio Compression Using XACT

WAV files are not compressed and can be quite large, especially if we want to use one or more music tracks in our game. XACT has the ability to compress the game audio. This can result in significantly smaller file sizes for our finished game.

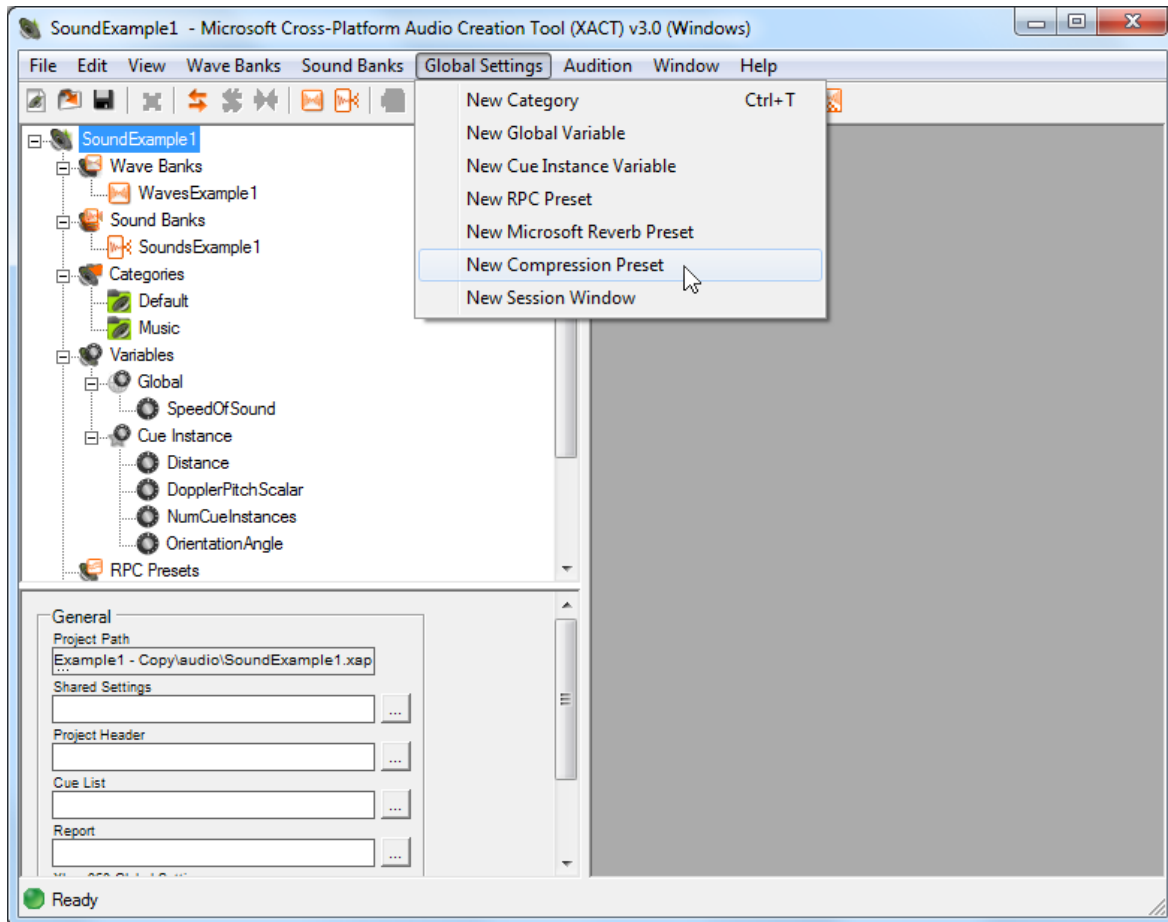


Figure 7.12. Creating a New Compression Preset in XACT.

To compress our game audio we create a Compression Preset by selecting “New Compression Preset” in the “Global Settings” menu (Figure 7.12). The compression preset may be renamed if desired. We will use the default name “Compression Preset”. A compression preset may also be created by right-clicking on the Compression Presets item in the XACT explorer (left window). Compression Presets may be applied to an entire Wave Bank or individual sounds within a Wave Bank.

To apply the Compression Preset to a Wave Bank, select the desired Wave Bank and select the Compression Preset from the drop-down menu in the properties window (Figure 7.13). To apply a Compression Preset to individual files, select the file in the Wave Bank window and select the Compression Preset from the drop-down menu in the properties window.

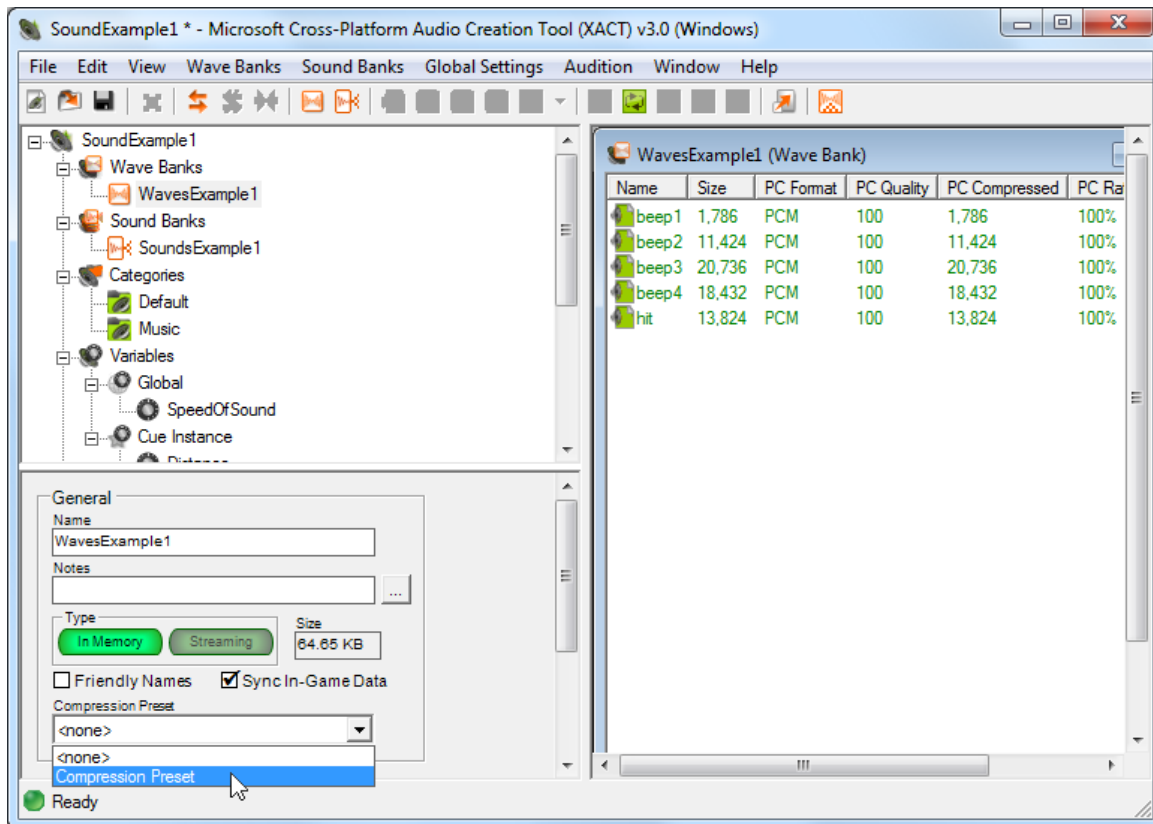


Figure 7.13. Selecting a Compression Preset

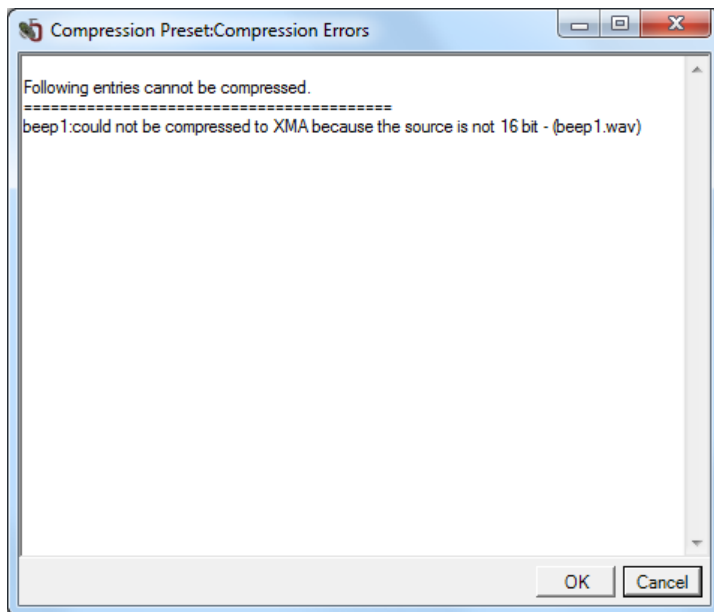


Figure 7.14. Compression Preset Errors

One or more error messages may be displayed if the format of the wav files is not compatible with the desired compression format (Figure 7.14). The errors may be corrected by changing the compression settings or converting the wav file. To convert the wav file, open the file in Audacity, select File/Export... and save as type WAV (Microsoft) signed 16 bit PCM. After converting the wav file, right-click the Wave Bank and select “Rescan Wave Bank Entries”.

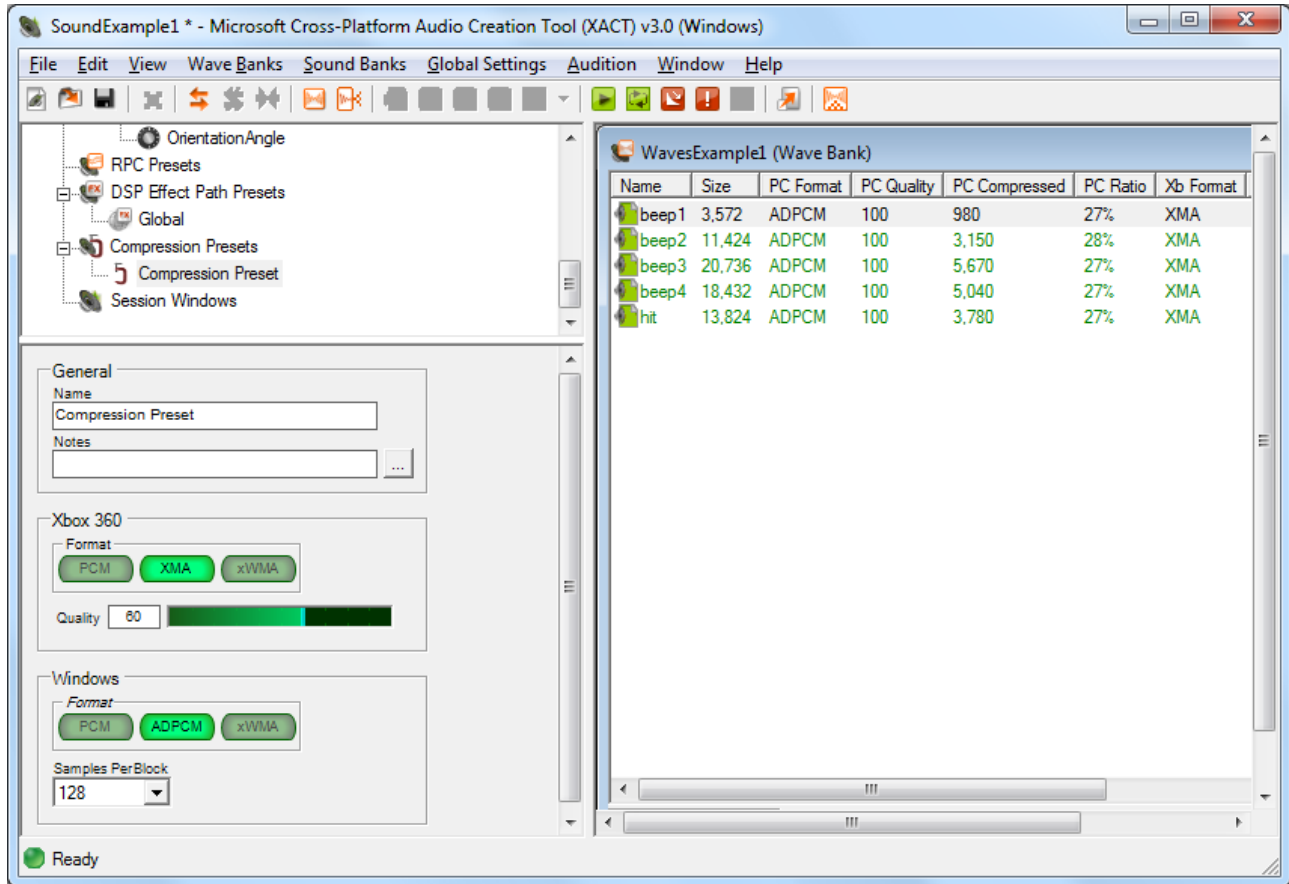


Figure 7.15. Compression Preset Settings

To change the compression settings, select the Compression Preset in the upper-left window. The compression settings for the Xbox 360 and Windows are displayed in the lower-left window. The possible formats for Windows are: PCM, ADPCM and xWMA. Pulse-Code Modulation (PCM) is a lossless uncompressed format usually stored in a .wav file. This is the native format for .wav files and will result in no compression. Adaptive Differential Pulse Code Modulation (ADPCM) is a lossy compression format. Lossy compression formats can alter the audio slightly in order to achieve a higher compression. ADPCM can achieve compression ratios of up to 4:1. Windows Media Audio format (xWMA) is a lossy compression format. The Quality setting affects the amount of compression. The amount of compression achieved can vary greatly from one wav file to the next. Compression ratios of 30:1 or higher are possible with xWMA. It is also possible for xWMA to create a larger file than the original. A little experimentation will be necessary to achieve the best results. In Figure 7.16 xWMA is applied to the actionTheme wave file. The original file was 8,617,984 bytes. The compressed size is only 301,050 bytes.

The format of the compression used for each file is displayed in the PC Format column of the (Wave Bank) window (Figure 7.15). To achieve maximum compression it may be necessary to create

multiple compression presets and assign them to individual files.

7.13 Pausing Sounds by Category

There are times when we may wish to pause the playback of sounds. Normally this is only required if the sound is set to loop or is very long, such as music. The XACT audio tool and engine supports pausing the playback of sounds by category.

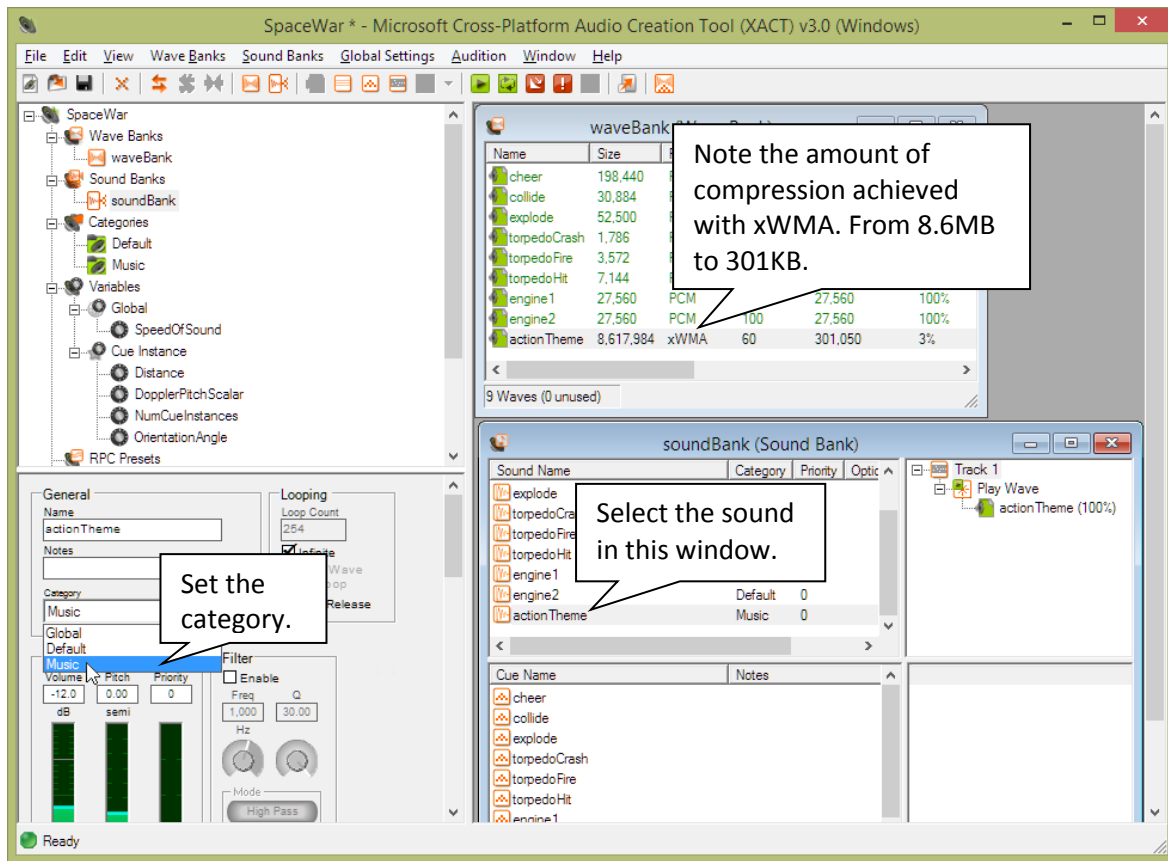


Figure 7.16. Setting the Category of a Sound in XACT.

To set the category of a sound select the sound in the soundBank window and then select the desired category from the Category drop-down. In Figure 7.16 we have set the category of actionTheme to Music.

Sounds are paused by creating an XACTCATEGORY variable that contains the category to pause and passing it as a parameter to the xactEngine Pause function. Resuming a sound is similar. We will add two new functions to our Audio class for pausing and resuming sounds.

```

//=====
// Pause sound specified by category.
// If the category does not exist no error occurs.
//=====
void Audio::pauseCategory(const char category[])
{
    if (soundBank == NULL)
        return;
    XACTCATEGORY iCategory = xactEngine->GetCategory(category);
    xactEngine->Pause(iCategory,true);
}

//=====
// Resume playback of paused sound specified by category.
// If cue does not exist no error occurs.
//=====
void Audio::resumeCategory(const char category[])
{
    if (soundBank == NULL)
        return;
    XACTCATEGORY iCategory = xactEngine->GetCategory(category);
    xactEngine->Pause(iCategory,false);
}

```

Listing 7.22. Pausing and resuming sounds by category.

Our Spacewar game calls these functions from inside the consoleCommand function. We chose to place the function calls here because the consoleCommand function is always called by the game loop even if the game is paused. Notice in Listing 7.23 that we have also added console commands to turn the game musing on or off.

```

//=====
// Process console commands
//=====
void Spacewar::consoleCommand()
{
    // The music control is located here because consoleCommand is always called,
    // even if the game is paused.
    if(paused || musicOff)
        audio->pauseCategory("Music");
    else
        audio->resumeCategory("Music");

    command = console->getCommand();    // get command from console
    if(command == "")                  // if no command
        return;

    if (command == "help")             // if "help" command
    {
        console->print("Console Commands:");
        console->print("fps - toggle display of frames per second");
        console->print("gravity off - turns off planet gravity");
        console->print("gravity on - turns on planet gravity");
        console->print("planet off - disables planet");
        console->print("planet on - enables planet");
        console->print("music on - plays music");
        console->print("music off - no music");
    }
}

```

```

    return;
}
if (command == "fps")
{
    fpsOn = !fpsOn;           // toggle display of fps
    if(fpsOn)
        console->print("fps On");
    else
        console->print("fps Off");
}

if (command == "gravity off")
{
    planet.setMass(0);
    console->print("Gravity Off");
}
else if (command == "gravity on")
{
    planet.setMass(planetNS::MASS);
    console->print("Gravity On");
}
else if (command == "planet off")
{
    planet.disable();
    console->print("Planet Off");
}
else if (command == "planet on")
{
    planet.enable();
    console->print("Planet On");
}
else if (command == "music off")
{
    musicOff = true;
    console->print("Music Off");
}
else if (command == "music on")
{
    musicOff = false;
    console->print("Music On");
}
}

```

Listing 7.23. Adding calls to pause and resume sound categories.

All of the code shown here is available in the Spacewar game on the Chapter 11 page at <http://www.programming2dgames.com>.