17 CAL 101

One advantage that SONAR has over any other music-sequencing product I've worked with is that it enables you to extend its functionality. If you find yourself in a situation in which you need to edit your MIDI or audio data in some way that is not possible with any of the current SONAR features (which is not a common occurrence, but it can happen), you can create a new editing function to take care of the task by using CAL. What is CAL, and how do you use it? Well, that's exactly what you'll learn in this chapter. This chapter will do the following:

- Define CAL.
- Show you how to run an existing CAL program.
- Explain prewritten CAL programs.
- Demonstrate how to view CAL programs.

What Is CAL?

CAL (*Cakewalk Application Language*) is a computer-programming language that exists within the SONAR environment. You can extend the functionality of SONAR by creating your own custom MIDI and audio data editing commands using CAL programs (also called s*cripts*). A CAL program is a set of instructions written in the Cakewalk Application Language that tells SONAR how to perform a certain task. For example, if you want to change the volume of every other MIDI note in track 1 to a certain value automatically, you can write a CAL program to do just that. And for future use, you can save CAL programs to disk as files with a .CAL extension.

Programming Languages A *programming language* is a set of commands, symbols, and rules that are used to "teach" a computer how to perform tasks. By combining these language elements in different ways, you can teach a computer to perform any number of tasks, such as recording and playing music. The combination of elements for a certain task or set of tasks is called a *computer program*. For example, SONAR is a computer program, albeit a very complex one.

A number of different kinds of programming languages exist, including BASIC, FORTRAN, C, LISP, and many others. Each has unique characteristics. If you are familiar with C and LISP, you'll feel right at home with CAL; it derives many of its characteristics from these two languages.

You might be saying to yourself, "Um, well, that's nice, but I know nothing about computer programming, so what good is CAL going to do for me?" Not to worry. Yes, CAL is a very complex feature of SONAR. If you really want to take full advantage of it, you have to learn how to use the language, but that doesn't mean CAL isn't accessible if you're a beginning user.

There are a number of prewritten CAL programs included with SONAR that you can use in your own projects. Cakewalk also provides a nice library of additional CAL programs on its Web site that you can download for free.

Running a CAL Program

Because all CAL programs are different, I can't explain how to use them in one all-encompassing way. When you run a CAL program, it usually asks you for some kind of input, depending on what the program is supposed to do and how it is supposed to manipulate your music data. But you can still follow this basic procedure to run a CAL program:

1. Select the track(s) (or data within the tracks) in the Track view that you want the CAL program to edit. This first step is not always necessary; it depends on the task the CAL program is supposed to perform. It also depends on whether the CAL program was written to process only selected data in a project or all the tracks in a project. The only way to determine the function of a CAL program is to view it with Windows Notepad, which you'll learn about later in this chapter.
2. Choose Process > Run CAL (or press Ctrl+F1) to display the Open dialog box.
3. Choose the CAL program you want to run and click Open.

That's all there is to it. Some CAL programs immediately carry out their tasks, whereas others first display additional dialog boxes if you need to input any values. The best way to begin using CAL (and to see how it works) is to try out some of the sample programs included with SONAR.

Run CAL Programs During Playback You can run CAL programs while a project is being played back. This means that you can hear the results of the CAL program on your data at the same time your music is being played. If you don't like what the CAL program does, just choose Edit > Undo (or press Ctrl+Z) to remove any changes the program makes to your data. If you then decide that you actually like the changes, instead of running the CAL program again, just choose Edit > Redo (or press Ctrl+Shift+Z) to put the changes back in place.

The CAL Files

To give you a better understanding of how CAL works and how you can benefit from it, I'll describe the prewritten CAL programs included with SONAR in the following sections. I'll give you a brief description of what each program does and how to use it.

Dominant 7th Chord.CAL

The Dominant 7th Chord.CAL program builds dominant seventh chords by adding three notes with the same time, velocity, and duration to each selected MIDI note in a track. In other words, if you select a note within a track and run Dominant 7th Chord.CAL, the program treats the selected note as the root of a dominant seventh chord and adds a minor third, a perfect fifth, and a minor seventh on top of it, thus creating a dominant seventh chord automatically.

Of course, if you know how to compose music, you probably won't get much use out of this CAL program. However, you might find it useful while working in the Staff view. While you're editing a MIDI data track in the Staff view, try highlighting a note and then running Dominant 7th Chord.CAL. It's cool to see those additional notes just appear as if by magic. This program can save you some time while you're inputting notes by hand, too.

Other Chord.CAL Programs

SONAR includes a number of other chord-building CAL programs that work the same way as Dominant 7th Chord.CAL, except they build different kinds of chords:

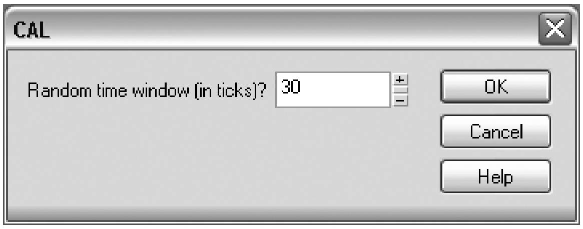
- **Major 7th Chord.CAL.** This builds major seventh chords by adding the major third, perfect fifth, and major seventh intervals to the selected root note or notes.
- **Major Chord.CAL.** This builds major chords by adding the major third and perfect fifth intervals to the selected root note or notes.
- **Minor 7th Chord.CAL.** This builds minor seventh chords by adding the minor third, perfect fifth, and minor seventh intervals to the selected root note or notes.
- **Minor Chord.CAL.** This builds minor chords by adding the minor third and perfect fifth intervals to the selected root note or notes.

Random Time.CAL

If you overindulge yourself while using SONAR's quantizing features, your music can sometimes sound like computer music with a robotic or machinelike feel to it. In some cases, this sound is desirable, but when you're working on a jazz or rhythm and blues piece, you don't want the drums (or any of the other instruments, for that matter) to sound like a robot played them. In this case, Random Time.CAL may be of some help.

This CAL program takes the start times of each selected event in a track and adds a random number of ticks to them. To give you some control over this randomization, the program first asks you for a number of ticks on which to base its changes. It then adds a random number to each event time that is between plus or minus one-half the number of ticks that you input. For instance, if you tell the program to use six ticks, each event time will have one of the following numbers (chosen at random) added to it: −3, −2, −1, 0, 1, 2, or 3. Using this program is a great way to add a little bit of "human" feel back into those robotic-sounding tracks. To use Random Time.CAL, just follow these steps:

1. Select the track(s) in the Track view that you want to process. Alternatively, you can select a single clip within a track or a specific range of events in one of the other views, such as the Piano Roll view or the Staff view.
2. Choose Process > Run CAL (or press Ctrl+F1) to display the Open dialog box.
3. Choose the Random Time.CAL file and click Open. The Random Time.CAL program will display a CAL dialog box (see Figure 17.1).



Random time window (in ticks)? 30

Figure 17.1 The Random Time.CAL program asks for the number of ticks upon which to base its event time processing.

4. Enter the number of ticks you want to use and click OK.

You'll probably need to experiment a little bit with the number of ticks that you use because a number that is too large can make your music sound sloppy or too far off the beat.

Scale Velocity.CAL

The Scale Velocity.CAL program is included with SONAR just to serve as a programming example; other than that, you don't really need it. SONAR already includes a Scale Velocity editing function, which provides even more features than Scale Velocity.CAL.

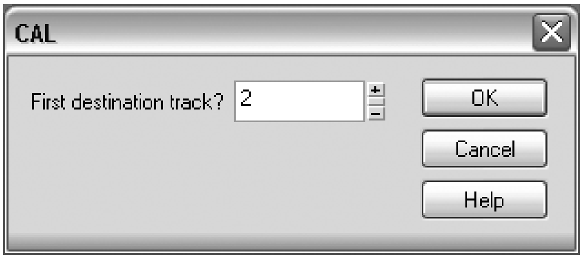
Split Channel to Tracks.CAL

If you ever need to share your music data with someone who owns a sequencing program other than SONAR, you can save your project as an .SMF (*Standard MIDI File*) file. Most computer music software products on the market support Standard MIDI Files; thus, they allow musicians to work together on the same song without having to own the same software.

However, not all Standard MIDI Files are created equally. Actually, several types of files are available; one in particular is called *Type 0.* A Type 0 MIDI file stores all its data—which is all the MIDI data from all 16 MIDI channels—on one track. Type 0 files are used often for video game composing but hardly ever used when composing for any other medium. Still, you might run across a Type 0 MIDI file, and if you open the file in SONAR, all the data shows up on one track in the Track view. Editing the data is rather difficult, so Split Channel to Tracks.CAL is a useful tool in this situation.

Split Channel to Tracks.CAL takes the selected track and separates the data from it by MIDI channel into 16 new tracks. For example, if the track contains data on MIDI channels 1, 4, 5, and 6, Split Channel to Tracks.CAL creates 16 new tracks (from the initial track), with the first track containing data from channel 1, the fourth track containing data from channel 4, and so on. The remaining tracks that don't have corresponding channel data are just blank. You use Split Channel to Tracks.CAL like this:

1. Select a track in the Track view.
2. If you want to split only a portion of the track, set the From and Thru markers to the appropriate time values.
3. Choose Process > Run CAL (or press Ctrl+F1) to open the Open dialog box.
4. Choose the Split Channel to Tracks.CAL file and click Open. The Split Channel to Tracks.CAL program will display a CAL dialog box (see Figure 17.2).



First destination track? 2

Figure 17.2 The Split Channel to Tracks.CAL program asks for the number of the track to start with when you're creating the new tracks.

5. Enter the number of the first track that you want Split Channel to Tracks.CAL to use when it creates the new tracks and click OK.

Overwriting Tracks Be sure to enter the number of the last track in your project plus one, so the newly created tracks don't overwrite any existing ones. For example, if the number of the last track in your project is 16, then enter 17.

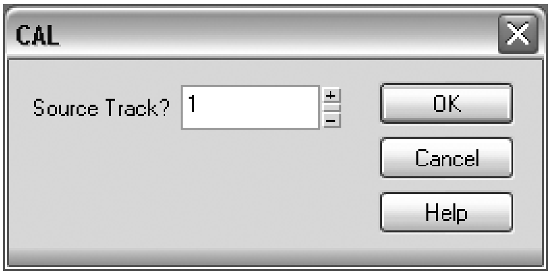
After it's finished processing the original track, Split Channel to Tracks.CAL will create 16 new tracks, starting with the track number you selected, each containing data from one of the 16 corresponding MIDI channels. Now you can access and edit the music data more easily.

Split Note to Tracks.CAL

The Split Note to Tracks.CAL program is similar to Split Channel to Tracks.CAL, except that instead of separating the MIDI data from a selected track by channel, it separates the data by note. For example, if you select a track that contains notes with values of C4, A2, and G3, Split Note to Tracks.CAL separates that track into three new tracks, each containing all notes with only one of the available note values. In this example, a new track containing only notes with a value of C4 would be created, another new track containing only A2 notes would be created, and another new track containing only G3 notes would be created.

This CAL program can be useful if you're working with a single drum track that contains the data for a number of different drum instruments. In MIDI, different drum instruments are represented by different note values because drums can't play melodies. So if you want to edit a single drum instrument at a time, having each instrument on its own track would be easier. In that case, Split Note to Tracks.CAL can be put to good use. To apply Split Note to Tracks.CAL to your music data, follow these steps:

1. Choose Process > Run CAL (or press Ctrl+F1) to open the Open dialog box.
2. Choose the Split Note to Tracks.CAL file and click Open.
3. The Split Note to Tracks.CAL program will ask for the number of your source track (see Figure 17.3). This is the track you want to split into new tracks. Enter a track number and click OK.

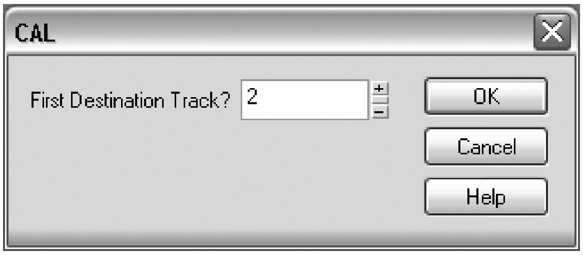


Source Track? 1

Figure 17.3 Here you can enter the source track for the Split Note to Tracks.CAL program.

4. The program will ask you for the number of the first destination track (see Figure 17.4). This is the number of the first new track that will be created. Enter a number and click OK.

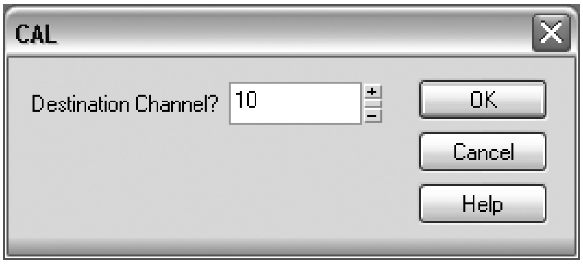
Overwriting Tracks Be sure to enter the number of the last track in your project plus one, so the newly created tracks don't overwrite any existing ones. For example, if the number of the last track in your project is 16, then enter 17.



First Destination Track? 2

Figure 17.4 Here you can enter the first destination track for the Split Note to Tracks.CAL program.

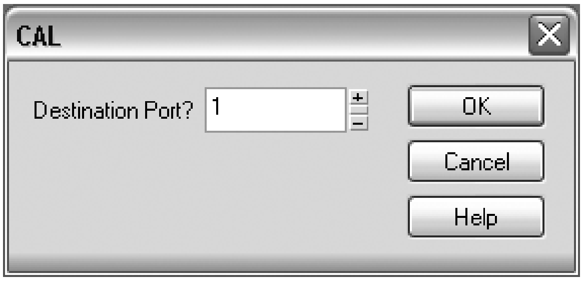
5. The program will ask you for the number of the destination channel (see Figure 17.5). This is the MIDI channel to which you want all the new tracks to be set. Unless you want to change the channel, you should simply select the same channel that the source track is using. Enter a number from 1 to 16 and click OK.



Destination Channel? 10

Figure 17.5 Here you can enter the destination channel for the Split Note to Tracks.CAL program.

6. Finally, the program will ask you for the number of the destination port (see Figure 17.6). This is the MIDI output to which you want all the new tracks to be set. Again, you should simply select the same output that the source track is using. Enter a number from 1 to 16 and click OK.



Destination Port? 1

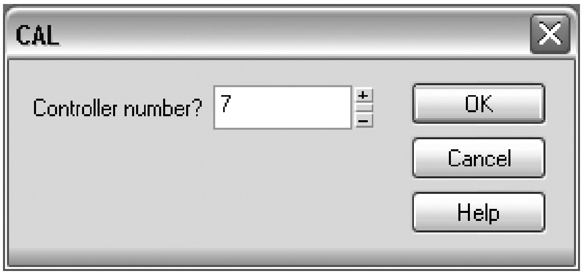
Figure 17.6 Here you can enter the destination port for the Split Note to Tracks.CAL program.

After you answer the last question, Split Note to Tracks.CAL will process the original track and create a number of new tracks (depending on how many different note values are present in the original track), each containing all the notes for each corresponding note value.

Thin Controller Data.CAL

You use MIDI controller data to add expressive qualities to your MIDI music tracks. For example, you can make a certain passage of music get gradually louder or softer (crescendo or decrescendo) by adding MIDI controller number 7 (Volume) to your MIDI tracks. Sometimes, though, an overabundance of MIDI data can overload your MIDI instruments and cause anomalies, such as stuck notes and delays in playback. If you have this problem, you can try thinning out the MIDI controller data in your tracks by using Thin Controller Data.CAL. This program decreases the amount of data being sent to your MIDI instruments by deleting only a select number of controller events—enough to reduce the amount of data without adversely affecting the music performance. It works like this:

1. Select the track(s) in the Track view that you want to process. You also can select a single clip within a track, or you can select a specific range of events within one of the other views, such as the Piano Roll view or the Staff view.
2. Choose Process > Run CAL (or press Ctrl+F1) to display the Open dialog box.
3. Choose the Thin Controller Data.CAL file and click Open.
4. The Thin Controller Data.CAL program will ask you for the number of the MIDI controller you want to process (see Figure 17.7). For example, if you want to remove some of the volume data from a track, use MIDI controller number 7. Enter a number from 0 to 127 and click OK.

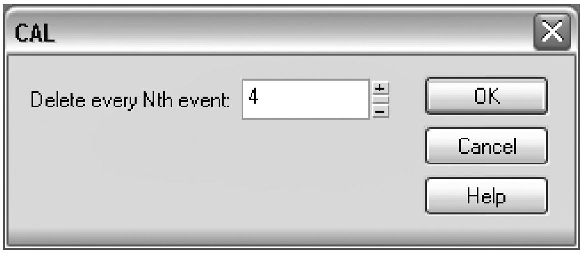


Controller number? 7

Figure 17.7 Here you can enter the controller number for the Thin Controller Data.CAL program.

5. The program will ask you for the thinning factor (see Figure 17.8). For example, if you enter a value of 4, the program will delete every fourth volume event it finds in the selected track or tracks. Enter a number from 1 to 100 and click OK.

After you answer the last question, Thin Controller Data.CAL will process the selected track or tracks and delete all the MIDI controller events that correspond to the MIDI controller number and the thinning factor you entered. If this procedure doesn't clear up your MIDI playback problems, you can try thinning the data some more, but be careful not to thin it too much;



Delete every Nth event: 4

Figure 17.8 Here you can enter the thinning factor for the Thin Controller Data.CAL program.

otherwise, your crescendos and decrescendos (or other controller-influenced music passages) will start to sound choppy rather than smooth.

Other Thin.CAL Programs

SONAR also includes two other controller-thinning CAL programs. These programs work almost the same way as Thin Controller Data.CAL, but each is targeted toward one specific type of controller. Thin Channel Aftertouch.CAL thins out channel aftertouch MIDI controller data, and Thin Pitch Wheel.CAL thins out pitch wheel (or pitch bend) MIDI controller data. To run these programs, you use the same procedure as you do with Thin Controller Data.CAL, but with one exception. The programs don't ask you to input the number of a MIDI controller because each of them is already targeted toward a specific controller. Other than that, they work in the same manner.

Viewing CAL Programs

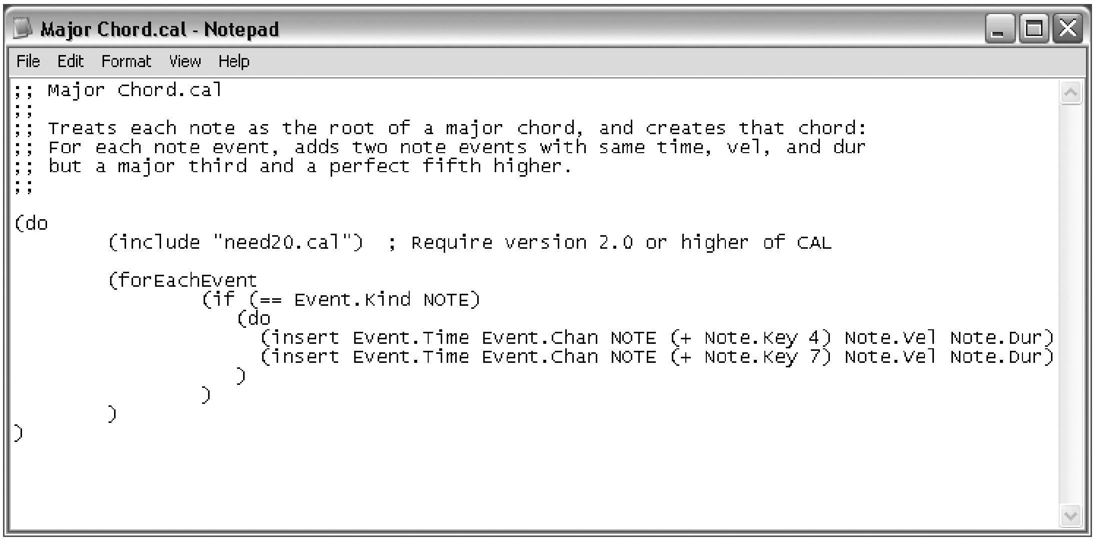
Unless a CAL program comes with some written instructions, you won't know what it is designed to do to the data in your project. This is especially true if you download CAL programs from the Internet. Many come with documentation, but many others don't. However, most programs do come with a brief description (as well as instructions for use) within their source code.

What Is Source Code? *Source code* (or *program code*) is the text of the programming language commands used for a particular program. You create a program by first writing its source code. Then a computer can run the program by reading the source code and executing the commands in the appropriate manner, thus carrying out the intended task.

To read the source code of a CAL program, you need to use Windows Notepad (or some other plain text editor). As an example, take a look at the source code for Major Chord.CAL:

1. In Windows XP or Vista, choose Start > All Programs > Accessories > Notepad to open Windows Notepad.

2. Choose File > Open and select the Major Chord.CAL file from the SONAR directory on your hard drive (or some other directory where your CAL files are stored). Click Open. Windows Notepad will open Major Chord.CAL and display its source code (see Figure 17.9).



```
;; Major Chord.cal
;;
;; Treats each note as the root of a major chord, and creates that chord:
;; For each note event, adds two note events with same time, vel, and dur
;; but a major third and a perfect fifth higher.
;;

(do
        (include "need20.cal")  ; Require version 2.0 or higher of CAL

        (forEachEvent
                (if (== Event.Kind NOTE)
                   (do
                     (insert Event.Time Event.Chan NOTE (+ Note.Key 4) Note.Vel Note.Dur)
                     (insert Event.Time Event.Chan NOTE (+ Note.Key 7) Note.Vel Note.Dur)
                   )
                )
        )
)
```

Figure 17.9 You can use Windows Notepad to examine and edit the source code of a CAL program.

As you can see in Figure 17.9, Windows Notepad allows you to see the source code of Major Chord.CAL and also to read the brief description included there. You can do the same thing with any other CAL program to find out how you can use it and what task it's supposed to perform. But that's not all! Using Windows Notepad, you can edit the source code for a CAL program, as well as create a CAL program from scratch. Because CAL programs are just plain text, you can use the same editing techniques you do with any other text, such as cut, copy, and paste text-editing procedures.