Transform Scales – how the mapping is done

Bob Zawalich January 10, 2006
Updated July 11, 2021

(Danger! This is an extremely tedious document!)

In the Transform Scales plug-in, all scales are ultimately 12 pitches (the 12 chromatic pitches, which might be spelled C C# D D# E F F# G G# A A# B. Scales are defined in such as way that the spelling of enharmonic pitches can be specified, and a distinction is made between scale tones and non-scale tones. Scales can be edited, created, or deleted by the user.

When one scale is mapped to another, the plug-in attempts to make a mapping that will retain the character of the original scale. It tries, for one thing, to retain as many scale tones as possible.

Note that the examples below are all given in the key of C for simplicity, but they will work for any key. Essentially, the scale mapping is done on scale positions, independently of actual pitch, and the result is transposed as needed.

There are 3 different procedures used in mapping.

1. If the scale type of the 2 scales is the same, the scale is transposed. This is equivalent to writing out the 12 pitches of the 2 scales, starting from the root, and mapping the From scale to the To scale. So mapping C major to G major results in (scales tones are marked with *):

C major	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
G major	*G	Ab	*A	Bb	*B	*C	C#	*D	Eb	*E	F	*F#

and this is effectively the same as adding a fixed interval to each pitch. It is the same thing Sibelius does when transposing by non-diatonic intervals. Note that the scales tones line up and need no shifting.

2. If the scales types are different, but the scales have the same number of scale tones, a two-step process is done. First the scale tones of the To scale are written in order at the positions of the scale tones in the From scale. This can involve a bit of a shift. So for C major to C Dorian (C Db D Eb E F F# G Ab A Bb B), Eb and Bb are shifted, and we get:

Cmajor	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cdorian	*C	Db	*D	*Eb	E	*F	F#	*G	Ab	*A	*Bb	B
Mapping	*C		*D		*Eb	*F		*G		*A		*Bb

The non-scale tones have MIDI/chromatic values (0 – 11) that fill in the spaces, and their diatonic spelling is taken from the To scale. So in this case, between C and D we will get Db, etc.

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cdorian (To)	*C	Db	*D	*Eb	E	*F	F#	*G	Ab	*A	*Bb	B
Mapping	*C	Db	*D	Eb	*Eb	*F	F#	*G	Ab	*A	Bb	*Bb

Note that doing this caused both Eb and E to be mapped to Eb, and Bb and B to Bb. But all scale tones of the From scale are mapped to scale tones of the To scale.

3. If the scales types are different, but the scales have a different number of scale tones, things get more complex. For example, looking just at scale tones, Cmajor to Cminor pentatonic gives:

Cmajor (From)	*C		*D		*E	*F		*G		*A		*B
Cmin pentatonic (To)	*C			*Eb		*F		*G			*Bb	

If we just aligned the scale tones as above, we would have:

Cmajor (From)	*C		*D		*E	*F		*G		*A		*B
Cmin pentatonic (To)	*C			*Eb		*F		*G			*Bb	
Mapping	*C		*Eb		*F	*G		*Bb		?		?

Some programs I have looked at do this (and fill in the ? with the root pitch), but this seems to me that it would change the nature of chords written in the From scale. It did not seem like it captured any of the essence of the original scale. After looking at some other programs, it seemed they were doing something like this:

* Walk through the scale tones of the From scale, and look up that pitch in the To scale. If it is a scale tone in the To scale as well, add it to the map

Cmajor (From)	*C		*D		*E	*F		*G		*A		*B
Cmin pentatonic (To)	*C			*Eb		*F		*G			*Bb	
Mapping	*C					*F		*G				

For the remaining scale tones in the From scale (in this case D, E, A, and B), fill in the nearest scale tone from the To scale. In Cakewalk, if the distance between 2 scale tones is the same, they always choose the lowest tone. So E is equidistant between Eb and F, so they would choose Eb.

I added an extra differentiator: I choose the closest scale tone that has not been used yet, in an attempt to increase the variety.

Also: if do not map anything except the root to the root, so it maintains its unique role. I also ensure that every pitch is greater than or equal to the previous pitch, so the mapping never goes backwards.

So when I look at D, it is 2 semitones above C and 1 semitone below Eb, so I choose Eb. Now E is 1 semitone above Eb and 1 semitone below F. This might yield Eb, but that has already been used so I look to F. Since that is also used, I do pick up Eb, the previous pitch. (There is a fair amount of arbitrariness in this process.) In the same way, A gets Bb and B (equidistant between the used Bb and the used C, will pick the previous pitch, or Bb. So we get these scale tone mappings:

Cmajor (From)	*C		*D		*E	*F		*G		*A		*B
Cmin pentatonic (To)	*C			*Eb		*F		*G			*Bb	
Mapping	*C		*Eb		*Eb	*F		*G		*Bb		*Bb

Obviously, since there are 7 scales tones in C major and 5 in C minor pentatonic, we will have some duplications. This mapping at least keeps some of the same chords as the original, and perhaps more importantly, this is the only mapping I could come up with that seemed both reasonable and general enough to implement. There is a certain symmetry to the result that is somewhat pleasing as well.

The non scale tones are handled as in the previous case. Once again, we need to fill in the missing semitones, and we take the spelling of the tones from the To scale:

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cmin pentatonic (To)	*C	Db	D	*Eb	E	*F	F#	*G	Ab	A	*Bb	B
Mapping	*C	Db	*Eb	Eb	*Eb	*F	F#	*G	Ab	*Bb	Bb	*Bb

Note that there are the same number of scale tones in the From scale and the mapping.

This mechanism works in the reverse order as well. Taking C min pentatonic as From and C major as To:

Cmin pentatonic (From)	*C			*Eb		*F		*G			*Bb	
Cmajor (To)	*C		*D		*E	*F		*G		*A		*B
Mapping	*C					*F		*G				

gives the scales tones that are the same

Cmin pentatonic (From)	*C			*Eb		*F		*G			*Bb	
Cmajor (To)	*C		*D		*E	*F		*G		*A		*B
Mapping	*C			*D		*F		*G			*A	

fills in the missing scale tones

Cmin pentatonic (From)	*C	Db	D	*Eb	E	*F	F#	*G	Ab	A	*Bb	B
Cmajor (To)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Mapping	*C	Db	D	*D	E	*F	F#	*G	Ab	A	*A	B

fills in the non-scale tones.

It can handle mapping to a scale with more than 7 scale tones, like a half-whole diminished scale (*C *Db D *Eb *E F *F# *G Ab *A *Bb B):

Cmajor (From)	*C		*D		*E	*F		*G		*A		*B
C HW Diminished (To)	*C	*Db		*Eb	*E		*F#	*G		*A	*Bb	
Mapping	*C				*E	*F		*G		*A		

gives the scales tones that are the same

Cmajor (From)	*C		*D		*E	*F		*G		*A		*B
C HW Diminished (To)	*C	*Db		*Eb	*E		*F#	*G		*A	*Bb	
Mapping	*C		*Db		*E	*F		*G		*A		*Bb

fills in the missing scale tones

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
C HW Diminished (To)	*C	*Db	D	*Eb	*E	F	*F#	*G	Ab	*A	*Bb	B
Mapping	*C	Db	*Db	Eb	*E	*F	F#	*G	Ab	*A	Bb	*Bb

fills in the non-scale tones.

It actually works fine with the case 2 (same number of scale tones) mappings. For example for C major to C Dorian,

Cmajor	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cdorian	*C	Db	*D	*Eb	E	*F	F#	*G	Ab	*A	*Bb	B
Mapping	*C		*D		*Eb	*F		*G		*A		*Bb

Cmajor	*C		*D		*E	*F		*G		*A		*B
Cdorian	*C		*D	*Eb		*F		*G		*A	*Bb	
Mapping	*C		*D			*F		*G		*A		

maps the same tones

Cmajor	*C		*D		*E	*F		*G		*A		*B
Cdorian	*C		*D	*Eb		*F		*G		*A	*Bb	
Mapping	*C		*D		*Eb	*F		*G		*A		*Bb

fills in the scale tones . This happens to be identical to the result in method 2, and this has been true for all the cases I have tried. However, because method 2 is simpler and it is obvious what all the scale tones will be, whereas in this method, the filled in scale tones may wander a bit, I am keeping the 2nd method as well.

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cdorian (To)	*C	Db	*D	*Eb	E	*F	F#	*G	Ab	*A	*Bb	B
Mapping	*C	Db	*D	Eb	*Eb	*F	F#	*G	Ab	*A	Bb	*Bb

Constrain to scale. This forces non-scale tones in the From scale to map to the "nearest" scale tone in the To scale. Some programs just map every pitch to its nearest scale tone pitch, but I prefer to map the scale tones as above and then map the non-scale tones. So for C major to C Dorian, the scale tone mapping is as above:

Cmajor	*C		*D		*E	*F		*G		*A		*B
Cdorian	*C		*D	*Eb		*F		*G		*A	*Bb	
Mapping	*C		*D		*Eb	*F		*G		*A		*Bb

At this point, if the non-scale tone is equidistant between 2 scale tones, I am falling back on what Cakewalk seems to do, which is to choose the previous tone. It might be possible to improve things a bit by considering used/unused scale tones, but I am not bothering to do that.

One change I am also making is that I do not map anything but the root to the root. So in the example below, Db would normally map to C, but I force it up to D.

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cdorian (To)	*C	Db	*D	*Eb	E	*F	F#	*G	Ab	*A	*Bb	B
Mapping	*C	D	*D	Eb	*Eb	*F	F	*G	G	*A	Bb	*Bb

Reversibility

These mappings are in general non reversible – so you can take the output of a mapping and feed it back into another mapping and get back the original only in a few limited cases. These are:

- Transposition: if you map A to B and then B to A (e.g., C major to G major and then back) you will get the same you started with, because each value in the From scale is assigned to a unique value in the Toscale. So this is a reversible mapping
- Same number of scale tones. Any scale tones if the two scales will map reversibly, again because they are mapped to unique targets. Some non-scale tones will be reversible, but not all.

When mapping different numbers of scale tones, the results will generally be non-reversible. If two notes are mapped to the same pitch, there is no way to determine late

what the original pitches were, and so those pitches will be treated thereafter as if they were the same.

Conclusions and suitability to task

So there are at least 2 good obvious questions. The first is hard to answer: will music put through these transformations make any sense? I don't have enough data to answer that yet.

The second is: how do these transformations compare to those of other software products. nearly every sequencer in the world can do this (corrolary: so how hard can it be?) so how does this compare?

I have access to examples from Cakewalk Pro Audio 9 (I never upgraded to Sonar) and have screenshots from Digital performer. So here are some comparisons:

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cdorian (To)	*C	Db	*D	*Eb	E	*F	F#	*G	Ab	*A	*Bb	B
TS Mapping	*C	Db	*D	Eb	*Eb	*F	F#	*G	Ab	*A	Bb	*Bb
Cakewalk mapping	*C	C#	*D	D#	*D#	*F	F#	*G	G#	*A	A#	*A#

same, except for spelling

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cdorian (To)	*C	Db	*D	*Eb	E	*F	F#	*G	Ab	*A	*Bb	B
TS Mapping Snap	*C	D	*D	Eb	*Eb	*F	F	*G	G	*A	Bb	*Bb
Cakewalk mapping Snap	*C	C	*D	D#	*D#	*F	F	*G	G	*A	A#	*A#

same, except for spelling

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
C HW Diminished (To)	*C	*Db	D	*Eb	*E	F	*F#	*G	Ab	*A	*Bb	B
TS Mapping	*C	Db	*Db	Eb	*E	*F	F#	*G	Ab	*A	Bb	*Bb
Cakewalk mapping	*C	C#	D	D#	*D#	*F	F#	*F#	Ab	*A	Bb	*B

Many differences:

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cmin pentatonic (To)	*C	Db	D	*Eb	E	*F	F#	*G	Ab	A	*Bb	B
TS Mapping	*C	Db	*Eb	Eb	*Eb	*F	F#	*G	Ab	*Bb	Bb	*Bb
Cakewalk mapping	*C	C#	C#	E	*Eb	*F	F#	*G	G#	G#	B	*Bb

Many differences. I can't say I care much for their mapping, which in 2 places (E and B in From) goes in reverse.

Cmin pentatonic (From)	*C	Db	D	*Eb	E	*F	F#	*G	Ab	A	*Bb	B
Cmajor (To)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
TS Mapping	*C	Db	D	*D	E	*F	F#	*G	Ab	A	*A	B
Cakewalk mapping	*C	C#	C#	*D	F	*F	F#	*G	G#	G#	*A	C

Cmajor (From)	*C	Db	*D	Eb	*E	*F	F#	*G	Ab	*A	Bb	*B
Cmajor pentatonic (To)	*C	Db	D	*Eb	E	*F	F#	*G	Ab	A	*Bb	B
TS Mapping	*C	Db	*D	Eb	*E	*E	F#	*G	Ab	A	Bb	C
Cakewalk mapping	*C	C#	D	D#	E	E	G	*G	G#	A	A#	*A#
Digital Performer	*C	Db	*D	Eb	*E	*G	Ab	A	B	C	B	C

It looks to me as if DP is just taking the major pentatonic scale tones and lining them up in order with the C major: 1 2 3 4 5 -> 1 2 3 4 5 or C D E F G -> C D E G A , and then the remaining entries were filled with root C. Once the initial similarity ends, it kind of falls apart.