

The Lost Art Of Sampling: Part 3

By Steve Howell



Having introduced the concept of looping, velocity switching, and multisampling, it's time to actually make some samples. We give you a few hard-won tips that can make your life easier on the way...

I left you last time having introduced the concepts of multisampling and velocity switching, thus having possibly given you the idea that to accurately emulate any real-world instrument using samples, you need to sample it playing every note. For even more faithful results, you may feel you need to sample every note at a variety of loudnesses, so that when you're triggering your instrument samples over MIDI from a keyboard, MIDI notes hit hard (ie. with high MIDI velocity values) trigger samples of your chosen instrument being played loudly, while soft notes, appropriately, trigger softly played samples. However, before you swoon at the complexity of this task, it's not as black as it's painted.

As I explained in *Part 2*, you can usually get away with multisampling every few notes of an instrument in most cases, which consumes less time and sample memory. And as I hinted at the very end of the article, similar 'dodges' are possible with velocity switching. In fact, with some instruments, you can get away with taking just one sample of the instrument playing at its loudest, and use that as the basis for playback at a variety of different MIDI velocities.

Huh??

Confused? Didn't I spend valuable time in *Part 2* explaining how the tonal quality of (say) grand pianos or acoustic guitars varies tremendously over their dynamic range, and how samples of said pianos or guitars being played quietly *couldn't* therefore simply be played back more loudly and still sound like realistic recreations of the original instruments? Well... yes, I did. However, all modern samplers, whether in software or hardware, have such comprehensive sample-processing engines on board that it is easy to 'fake' the dynamic response of many instruments, provided you start with a loud, bright-sounding sample of the instrument being played near its maximum possible volume.

In the following example, an envelope is applied to the sample, and the attack of that envelope is mapped to trigger velocity, so that for lower trigger velocities, the attack time increases. The bottom three curves show the effect on the sound of combining the start time and the enveloping techniques.

Firstly, you need to map the MIDI velocity of the note that's triggering your sample to the sample's playback amplitude, so that the sample is played back more loudly as you play keys on your trigger keyboard harder. Then you map trigger velocity to the cutoff frequency of your sampler's built-in low-pass filter. In this way, as you play your one loud, bright instrument sample at lower MIDI velocities, the low-pass filter will take out more and more of the high frequencies, and make the sample less bright, and more muted and mellow in tone — in other words, more like the sound of the real instrument being played more softly. Of course, the precise details of how you achieve these controller routings will vary from sampler to sampler, so you'll have to check your documentation, but rest assured that all modern samplers are capable of this kind of routing.

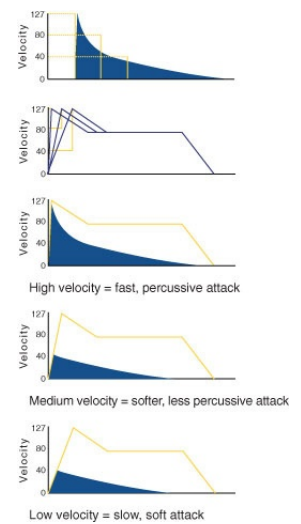
But didn't I explain in *Part 2* that most real instruments could sound *completely different* when played loudly and softly? Surely just routing trigger velocity to the filter and amplitude isn't going to deal with all those complex tonal changes? Well, you'd be surprised — these simple tricks can suffice for a fairly decent emulation of the dynamics of some instruments, but yes, with others, further refinement is required.

I explained you in *Part 2* how acoustic pianos have a highly percussive attack phase when played loudly, how stringed instruments like the cello and violin have a very obvious 'scrape' at the start of their notes when bowed aggressively, and how flutes have distinctive, noisy, 'chiffy' attacks when blown hard. These aspects are much less obvious or absent when the instruments are played more softly, so simply filtering samples of these instruments being played loudly isn't going to remove these qualities for realistic results at lower playback velocities. Something else is needed.

The required trick goes by different names in different samplers, but essentially, it involves also routing the sample trigger velocity to the point at which playback of the sample commences, as shown in the topmost diagram on the right. Basically, the lower the triggering MIDI velocity is, the later playback of the sample commences. When played hard, you hear the sample from its start, including its bright/aggressive-sounding attack; as you play more softly, more and more of this portion is omitted.

Of course, it's not as simple as that. Simply starting playback of a sample later could sound very strange, and would usually cause glitching as the sample playback level leaps from zero to a point some way through the instrument sample (such glitching is usually audible as a loud click — more on this later). Fortunately, the built-in synth engine found in all modern samplers can once again come to the rescue. All you have to do is apply an enveloped amplifier and/or filter to the sample, and map the sample trigger velocity to the attack time of the envelope! Simple, isn't it...? Seriously, though, although this sounds complex, the effect is easy to understand. Once set up in this way, the attack of the volume envelope applied to the sound will lengthen as trigger notes are played more softly. In other words, the sample will fade in more slowly the softer you play. If you've also applied a filter envelope to the sound and mapped it in the same way, the brightness of the sound will also fade up more slowly the more softly you play. Conversely, when the sample is played hard, the attack of envelope will be fast enough that you will hear the original sample from its start, aggressive attack portion and all. The second graph on the left may help to make this clearer.

With all of these techniques, you really can go some way to emulating an instrument's natural dynamic characteristics with just one sample. Played softly, it will be low in volume (thanks to the velocity-to-amplitude



Tricks to enable you to use one sample of an instrument played loudly to emulate that instrument being played over a wide dynamic range. In the top example, the trigger velocity is mapped to the start time of the sample, so the softer the incoming trigger note is, the later the sample starts, and the more the aggressive-sounding attack portion of the sample is missed out!



Applying velocity to sample start and envelope attack times in Native Instruments' Kontakt.

routing) and mellow in tone (because velocity is also mapped to the low-pass filter cutoff). If you've also mapped velocity to the sample start time and envelope attack, soft playing will also result in a sound that lacks perceptible 'attack' in a natural-sounding way. And, of course, if you play hard, you'll hear the original sample — a loud, bright sound with noticeable attack (see the final three graphs on the left). Of course, all of these settings need to be carefully tailored according to the instrument you're sampling. With a string

sound, the settings might be more extreme, so that light playing will result in a slow, languid attack, whilst playing hard will reveal the sound of the bow really 'digging in'. With more percussive sounds, the settings will have to be more subtle, so as not to give a perceptibly 'slow' attack, just a slightly 'softer', less percussive attack.

If this all sounds hopelessly complex, rest assured that sampled sounds can be made to sound more natural this way, and indeed have been for many years. Akai samplers from the S1000 onwards have had dedicated parameters for setting velocity to sample start and velocity to envelope attack (a screenshot shows the implementation in *Aksys*, Akai's S5000/S6000 editor). More modern software samplers such as Native Instruments' *Kontakt* (above) require you to specifically route incoming MIDI velocity to the sample start and envelope attack parameters, which is perhaps not as straightforward or intuitive, but potentially more flexible.



Applying velocity to sample start and envelope attack times in Akai's Aksys editor for S5000 and S6000 samplers.

Mono Or Stereo?

Whether you're using a software or hardware sampler, you still have to consider whether to take your samples in mono or stereo. This can be a surprisingly tricky choice. Of course, if you have a mono signal coming in through the L/R input, it's a no-brainer — sample in mono. There is absolutely no benefit in recording in stereo unnecessarily, as it's just a waste of sample RAM. However...

If your input source is *genuinely* stereo (if, for example, it has some lovely room ambience on it), which is best? Your first reaction might be to record in stereo, but that is often not ideal in practice. You see, many 'stereo' samples are actually mono sound sources panned centrally but with a stereo ambience spread across the stereo image. In isolation, they sound very impressive, and much more 'natural', but when used in the context of a mix, such samples can be more difficult to place in the stereo image. They can't just be panned around, and if you try to do this, the sample's natural stereo ambiances will be disturbed. Generally, my feeling is that it's better to sample in mono. Not only does this save on storage space, it gives you much more flexibility to place the instrument in your mix later on. Having said all that, there are occasions where sampling the instrument in stereo in its natural acoustic environment is apt (for example, when sampling a drum kit in stereo), as long as you don't need to move those instruments around in your mix.

And of course, technology has moved on. A few years ago, I might not have made the recommendation I have here, but these days, modern digital reverbs — and, of course, the new breed of convolving reverbs — allow you to create very natural acoustic spaces long after recording, and so the need to sample stereo ambience with the sound source has diminished.

Sampling Safari

Having now spent some time explaining the concepts that enable us to reproduce real instruments with samplers, it's time to look at the process of capturing samples, so that you can begin to think about putting these ideas into practice.

All modern samplers can import pre-recorded samples from CD and CD-ROM, but this series is designed to get you sampling yourself, and so it's important to explain here that hardware and software samplers differ slightly as regards creating your own samples.

As I mentioned briefly in the introduction to this series, one of the things software samplers can't usually do – rather confusingly, given that we still call them samplers – is sample. However, once you've captured some audio digitally by some other means (by using other recording software, for example, or by bringing pre-recorded audio into your host computer from somewhere else), you can use pretty much any software sampler to do everything with that audio that we've discussed until now, such as looping it, or creating velocity-switched multisamples. In contrast, nearly all hardware samplers are designed to capture samples as well as process and edit them. To this end, hardware samplers will typically have their own 'sample' or 'record' mode, accessed via separate pages in their operating system, with all the parameters required for effective sampling.

This said, whether you're capturing audio with sampling hardware or recording into your computer with the intention of importing the resulting audio into your software sampler later on, the principles are pretty much the same, and if you've been reading *SOS* for any length of time, there's a good chance they'll be familiar. As with any recording, what you'll hear at the output is only as good as what you record, so if you are sampling acoustic instruments and want them to sound authentic, you should always aim to use a decent microphone suited to the instrument you are sampling, placed appropriately in a suitable acoustic environment. Of course, choosing a mic, mic placement, and other traditional recording practices are beyond the scope of this series, but they are worth knowing, because no amount of post-processing is likely to make a poorly recorded sample sound good. However, if you're recording analogue sounds with a mic rather than capturing audio from an existing digital recording, one recording tip has to be mentioned here, and that's to capture your samples at as high a recording level as is possible without overloading your A-D converters, as this gives you a good signal-to-noise ratio (high signal/low noise). Hardware samplers all have input-level meters of one kind or another, and it's these you should watch, adjusting the recording level until you're as close as possible to the 0dB mark without overshooting it.



Typical sampling input parameters, from the display of an Akai hardware sampler.

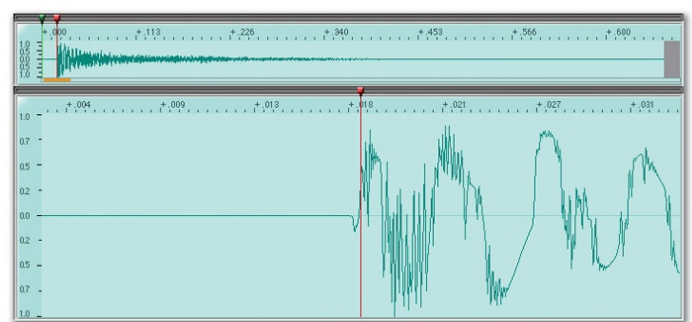
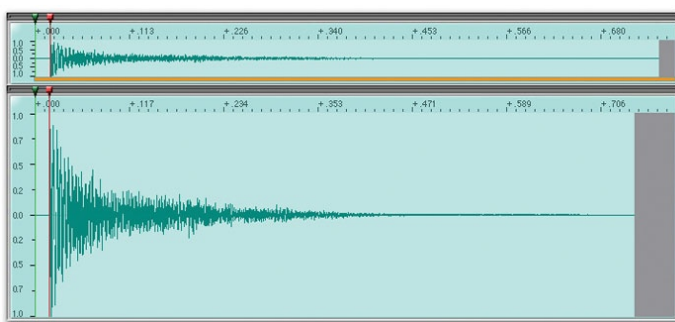
If you do 'overcook' the recording in this way, you will create samples full of digital distortion, which can be extremely unpleasant-sounding – and of course you'll hear the distortion every time you play back that sample. What's more, digital distortion can be even more obvious when the sample is transposed down or up even a few semitones – although this does mean that transposing samples is a good way to *check* for distortion. As with most recording problems, no amount of editing or processing afterwards is likely to get rid of distortion created at this stage, so if you *do* overload your converters, you're better off starting from scratch and re-recording your sample. Unless, of course, you think you can make use of the distorted sample in some creative context...

If you're making a recording for later import into a software sampler, and your recording levels are appropriately set, you're ready to go — although if you're sampling a synth sound, it is worth considering a few things at this stage which can make life easier later, when you come to edit your samples. If you're recording directly into a hardware sampler, these considerations are equally valid, but you'll probably also have a further layer of parameters relating to sampling which you may have to set up before you proceed. These will vary from sampler to sampler.

You will probably have to tell the device whether you're recording via its analogue or digital inputs (if it has the latter), and whether you're recording in mono or in stereo (see 'Sampling Synths' box later). If you're sampling a musical instrument, you'll probably have to tell the sampler which note you're about to feed it (this is sometimes known as the sample's Base Note). In addition, there may be curious-sounding parameters with names like Record Trigger and Record Length. The first of these enables you to put the sampler into Record mode, but to only begin sampling when a certain trigger condition is met, such as receiving a certain recording level or a particular MIDI note. The second determines how long the sampler will record for once it's been triggered. If your sampler has them, these options can be handy when you're sampling an instrument you're playing yourself, as you don't have to keep dashing from the sampler to the instrument and back again to capture your samples.

Editing Extravaganza

Once you've captured your audio, you will probably need to edit it in some way before it can be used. There's usually 'dead space' at the start of the sample — that is, an area of silence between the point where recording started and where the sound being sampled was played. If you leave half a second's silence at the start of a sample, there'll be half a second's delay before a note is heard after you trigger the sample (now that's what I *call* latency!), and more if the sample is transposed downwards. If you're using a hardware sampler, there will be a sample-editing mode for this purpose, and a Trim function to do the job. Some software samplers offer similar functions; where this is not the case, the sample will have to be trimmed in an external audio editor before it's imported into your sampling software.



The trimmed sample shown here looks reasonable when zoomed out...

Zooming in reveals that the sound's attack has been clipped. Playback is not starting from the zero axis, so the sample is now likely to click.

Wherever you do it, to edit the sample start point properly, you should zoom in quite dramatically so that you can place the start point at the very first sample of the recording. You *can* edit when zoomed out, but your edit will almost certainly not be accurate, and you might leave some delay before the note, or, worse, truncate the attack, which can introduce clicks and other side-effects.

It's easy to generate clicks when editing samples. If an edit results in a discontinuous portion of waveform, where the amplitude of the sample makes a sudden leap from one value before the edit point to a quite different one after it, you'll almost certainly hear an audible click. For this reason, experienced editors of

digital audio always try to make cuts at points where the waveform crosses the zero axis, and cut to another similar point, so that the amplitude values are at zero before and after the edit, and no click results. These points are known as 'zero crossings', and they can be particularly helpful when you're trying to make seamless sample loops, of which more in *Part 4*.

Of course, editing or trimming to zero crossings is not compulsory, and some samples can actually *benefit* from the added 'bite' of a click created from trimming the start of a sample to an 'off-axis' value. However, the same is not generally true of sample end points; very few samples sound good ending in a click. Nevertheless, samples should also be carefully trimmed at their ends. You might ask what the harm is in having some dead space at the *end* of a sample, where you'll never hear it; the answer is that it wastes sample memory and processing power.

If you have a 3.5-second drum sample with three seconds of silence at the end, you're almost doubling the sample RAM needed to play the sample, with no benefits whatsoever. In these days of plentiful sample RAM, that might seem pretty insignificant, but when you have hundreds of similarly wasteful samples loaded, you might be squandering hundreds of megabytes of memory —

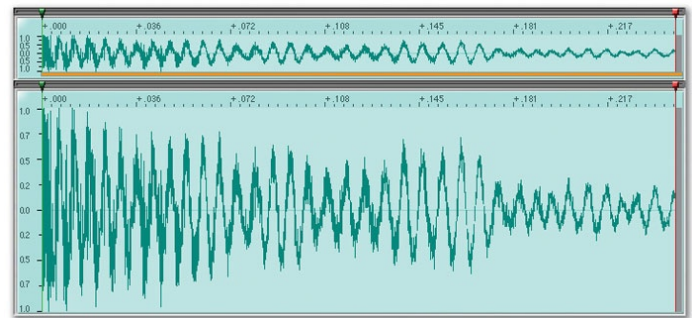
on silence! What's more, while that silence is being played, a voice on the sampler is still being used up, even though you can't hear anything, so your overall polyphony is affected by such samples. In software samplers that stream audio from their hosts' hard disks, the disk will likewise be working harder than it needs to, as will the host CPU, and thus maximum polyphony may be lower than it could be, or, in the worst case, notes may begin to drop out or stutter.

As I hope this makes clear, it's good practice to edit the end points of samples! This is best done in a similar way to trimming the start of samples, this time by zooming in as closely as possible and trying to place the end point at the very last sample in the waveform, and on a zero crossing if possible, so that the sample doesn't click at the end.

Sampling Synths

One popular use for samplers is to sample other synths. With old monosynths, this has many advantages and benefits. For a start, by definition, vintage monosynths can only play one note at a time — but in a sampler, they magically become polyphonic! However, there are other benefits. Most older analogue synths didn't have any way to store sounds, and so a sampler can be used as a way to create presets of your favourite synth sounds. Also, most old synths had no velocity sensitivity, but in a sampler, the sounds gain that capability.

When sampling a synth, there are a few things you can do to the source sound to make life easier for you when it comes to editing your samples. To begin with, set the synth's amplitude sustain to maximum — this will make the sound easier to loop later, and the sound's 'natural' envelope can always be restored in the sampler using its built-in amplitude envelope. Similarly, you should also disable all modulation (vibrato, panning, tremolo and so on). Basically, you should seek to make the source sound as static as possible — all the 'movement' and effects can be added later using the sampler's onboard facilities.



This noisy tom sound ends at around the 0.15 mark, and the waveform from that point onwards is just useless noise. Perhaps a fade can help?

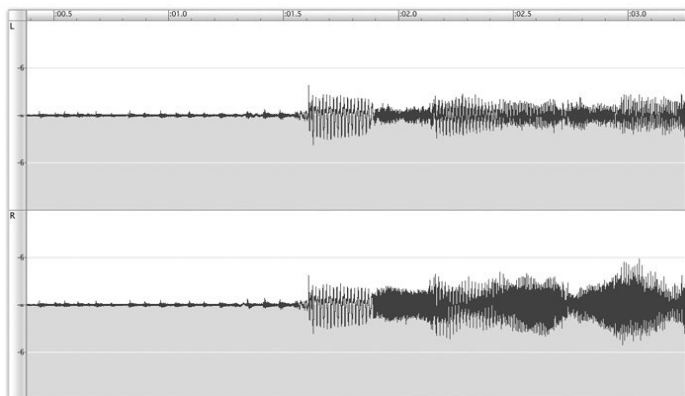
The filter, though, is a moot point — should you bypass that on the original? It depends. If you are sampling a classic synth with a distinctive filter (for example, a Minimoog, Korg MS20 or EDP Wasp), it's probably best to keep it in place even if it adds complications later, as you'll really want to keep the character of that old, quirky filter. But if the sound of the filter is not vital, bypass it and use the sampler's filters to recreate the sound later. Whilst the result may not be *totally* authentic, it will ultimately be more flexible and the audible difference may be negligible.

And should you leave the effects on? This is a similarly difficult question, and the answer depends on circumstances. In many modern synths, the effects are often the most important element of some sounds and if you bypass them, the sound can be extremely lacklustre! Mind you, many samplers these days are equipped with a similar complement of effects, so that sound can usually be recreated in the sampler. As with the issue of the filters, your sample will be generally more flexible if you can capture it 'dry' and add the effects later in your sampler.

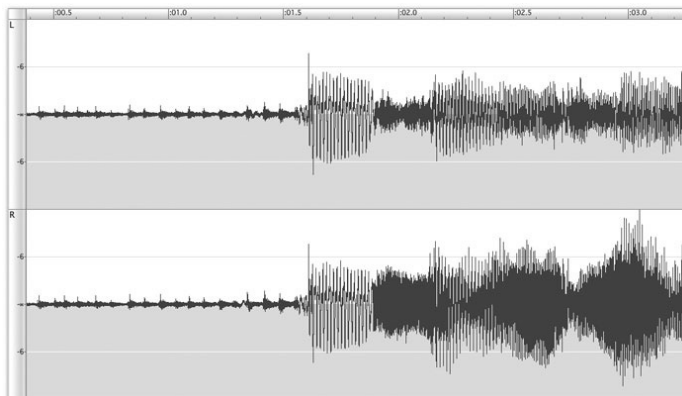
Having said all that, many sounds you hear in synths are created by using a unique combination of that synth's facilities. It might be through layering, filtering, modulation, effects and/or a combination of all of those — and so sometimes you just have to just sample the entire sound in all its glory in stereo, complete with all its effects *et al*.

Normalising & Fading

In my opinion, after trimming a newly captured sample, the next most important process is to normalise it. This process analyses the entire sample, searching for the highest level throughout, and then boosts the overall sample level so that the previously found maximum value comes to rest at 0dB, the maximum value for digital recording (see the screengrabs on page 64). By normalising your samples, you ensure that you're getting maximum signal level and dynamic range from your sampler. However, be aware that normalising raises the noise floor along with everything else in your sample, so if the sound is noisy to begin with, it may become unacceptably so after normalisation. As already stated, the best cure is to take good clean samples at the very start of the process!



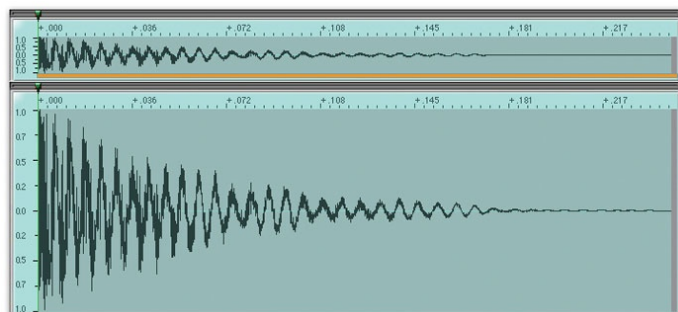
A section of stereo digital audio prior to normalisation. The loudest point is clearly visible in the Right channel at just past the three-second mark, but it's some way from being at maximum.



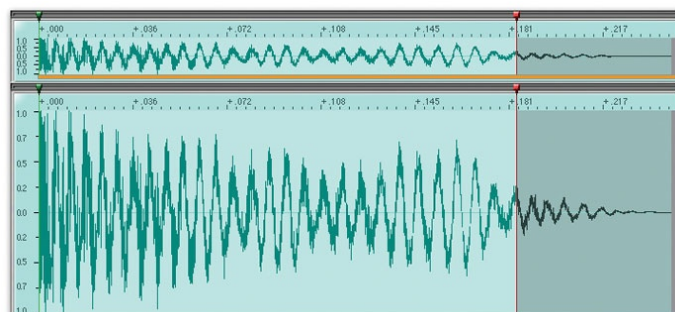
The same audio following normalisation. The volume throughout has been boosted such that the maximum value in the Right channel now sits at 0dB.

Samples of instruments that naturally decay slowly (piano, guitar) may well end noisily, particularly if they've been compressed when they were recorded, but this can be cleaned up if your sampler's editor offers a Fade function, allowing you to fade the sample out to clean silence by its end. If the noise is very evident throughout the sample, however, the sound of the noise fading out can itself be overly apparent when the sample is played back — a phenomenon much harder to convey in words than it is to recognise it when you hear it! Again, the best cure for this is to take a clean sample in the first place, but if this is not possible, you might be able to effect a rescue with one of the many denoising packages available these days for Macs and PCs (having said that, I really like the Denoise algorithm built into Adobe's *Audition*, although it is PC-only).

Like all sample-editing tools, Fade functions can be misused, as the examples shown on the page opposite demonstrate. The tom sound shown in the top screenshot ends abruptly, buried in noise. The simplest way to overcome this is to select the entire sample and apply a fade-out. Or is it? As the second example shows, although this resolves the problem of the sound's noisy ending, the body of the sound has also changed quite dramatically. Sometimes, this kind of fade can 'tighten up' a sound and make it 'punchier', but if it seems excessive, the alternative, as shown in the last waveform, is to leave the main body of the sample unfaded, and just fade its last part.

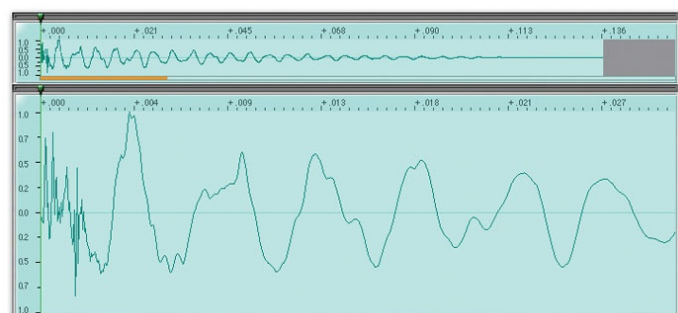


This over-zealous fade removes the noise, but also considerably changes the body of the sound.

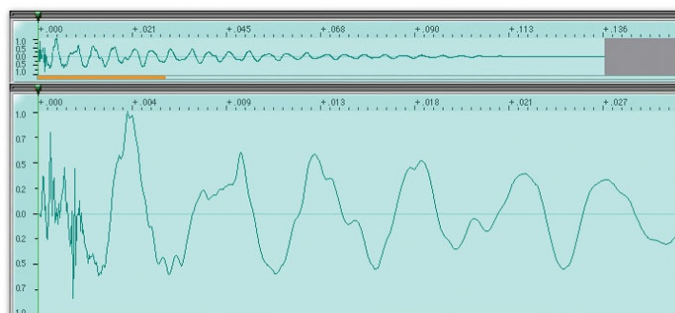


The alternative is to fade just the last portion, preserving the body of the tom sound and tidying up the tail.

Not all fades fade out, either. A quick fade-in can rescue a badly trimmed, 'clicky' sample start point, as the example waveforms shown above demonstrate. The sample clicks because it doesn't start on a zero crossing. However, with a short fade-in applied, this can be eliminated, or at least minimised. Of course, if it's too long, a fade-in will soften a sound's attack, so this approach should be used with care.



This badly trimmed waveform clicks audibly and unpleasantly, because playback commences some way into the sound.



The same waveform plays back smoothly when a quick fade-in is performed over the course of the first few hundred samples. Although the waveforms look very similar, look closely at the start section — it now begins on a zero crossing thanks to the fade-in, and sounds fine..

What's In A Name?

Much like getting the sound right at source, it is good practice to name your samples sensibly. You might think I'm just being pedantic, but it's amazing how hard it is to find a sound in a vast collection when you haven't named your samples according to some sort of system — even when you know it's there somewhere. If you don't believe me, heed the words of *SOS*'s samplist *extraordinaire* Dave Stewart, who went as far as to write a whole feature for *SOS* on the merits of keeping your sample library organised (*SOS May 1994*). And these aren't the words of an obsessive with too much time on his hands — Dave makes his living from knowing (amongst other things, obviously) where his sounds are, and how to get to them *quickly*.

I typically use three methods to name my samples. Firstly, I name according to the note being sampled (for example 'Synthbass C1', 'Strings D#3', and so on). At other times, I will just use my Akai S5000's auto-naming function and give the first sample a sensible 'root' name (eg. 'Synthbass 01') so that subsequent samples I record

up the keyboard range are auto-named accordingly ('Synthbass 02', 'Synthbass 03', and so on). My third method is a mixture of the previous two — I use an auto-numeric 'identifier' to get the samples in quickly, and I map those out in ascending order across the keyboard range. Once that's done, I rename the samples to use note names ('C1', 'C2', 'G1', and so on). How you work is entirely up to you, but it *is* worth naming your sample(s) sensibly — it will save you time in the long term.

There's one other thing about naming samples, which applies particularly to those using software samplers — avoid long names! These days, the OS in which the sampler will be used will almost certainly allow long names to be used — potentially up to 256 characters, which is great. Even if you're using a hardware sampler that can read or use WAV files made on a Mac or PC, long file names are possible, allowing us to use quite descriptive names for our samples.

However (there just *had* to be one, didn't there?) most samplers' user interfaces only allow a limited range of characters to be *displayed* — especially those of hardware samplers. So, tempting as it might be to name your sample 'Stereo Ambient Steinway Concert Grand Piano C1.wav', when this file is put into your sampler (or viewed in its browser), you might find that it only displays 'Stereo Ambient Stei~.wav' (or something even shorter and less comprehensible) with no indication of the sample's 'identifier'. This can make life very difficult later on. One way around this is to put the unique identifier (for example 'C4' or '01') at the *beginning* of the name. Alternatively, make sure that you keep your sample names short and succinct, for example: "St_Am_SteinC1.wav". This may initially seem cryptic and perhaps inelegant, but it can be far more useful in the long run.