

The basics.....

Sampling is a process where an audio signal is recorded digitally after which it can be played from a musical keyboard and/or triggered from pads, etc.. These recordings (largely known as 'samples') form the basis of a sound and can pretty much be regarded as the 'oscillators' such as you might find in an analog synthesiser - in fact, many modern products use samples rather than electronic oscillators as the basis for their sounds.

Typically, such samples then pass through filters and amplifiers which are controlled by envelopes, LFOs and velocity, etc., much like a standard analog synthesiser. In products that rely on sampled waveforms as the basis of their sounds, this is known as 'S+S' or 'sampling and synthesis'.

We will look at the details later in this tutorial.

The history.....

Musicians and composers have been messing around with recorded sound for many decades. As long ago as the 1940s, composers were recording acoustic sounds and speeding them up and slowing them down and reversing them, looping them and splicing them all together to create collages of manipulated sounds which became known as *musique concrete*.

The grand-daddy of sampling keyboards is not, contrary to popular opinion, the Mellotron but the instrument on which it was based - the Chamberlin. Invented by one Harry Chamberlin in the late 40s, it is said that Chamberlin (a keen home organist and tape recording hobbyist) had the idea that if you could somehow play tape recordings of 'real' instruments from some form of musical keyboard, you could play *any* instrument. Chamberlin managed to make his instrument (in his garage!) and sold quite a few during the '50s but being individually hand-built in crude circumstances, reliability was a problem. What happened next, however, is shrouded in myth and mystery. It is said that Chamberlin's head (and only!) salesman, Bill Fransen, travelled to the UK in search of a reliable supply of tape heads, approaching a Midlands-based company called Bradmatics. Intrigued why someone would want such a large order of tape heads, they met with Fransen and became intrigued by the product concept. What happened during those discussions will probably never be known but a new company was started called 'Mellotronics' and the Mellotron went into production.

Originally, the Tron was aimed at home organists with pre-recorded drum and accompaniment patterns on one keyboard with melodies and chords played on another. However, in the early '60s, the BBC thought this would be ideal for quickly playing sound effects, etc., in radio and TV productions (which, remember, were broadcast live in those days) and several customised Trons were build especially for the Beeb.

Then some bright spark had the idea of playing 'real' instruments such as strings, flute, choir, etc., and so the Mellotron as we know it was born and was rapidly adopted by many artists and bands in the mid-'60s, appearing on many a hit record. In the '70s, the Mellotron became synonymous with prog-rock bands such as Genesis, Yes, King Crimson and others who made the Mellotron a trademark of their sound.

With the advance of polyphonic synths, however, the Mellotron soon fell from grace (despite their magnificent sound, they were big and unreliable beasts to have to haul around from gig to gig!).

Of course, sampling as we know it came about in 1979 with the release of the seminal Fairlight CMI (Computer Musical Instrument). Originally, however, the Fairlight was intended to be a digital synthesiser and sampling was a kind of bolt-on added bonus. But it was the sampling that captured the imagination of everyone (although you'd have needed a cool \$40,000 or more to buy one!!!).

After the Fairlight, the race was on to bring sampling to the masses and the first off the starting block was Emu with their imaginatively titled 'Emulator' for a mere (!!) \$10,000. This was superseded by the Emulator II which, although more expensive (around \$18,000) was much more capable with filters and synth facilities.

Arguably one of the biggest advances in promoting 'affordable' sampling was the release of the Akai S900 in 1986 which just about offered everything musicians wanted of a sampler (at the time anyway) at a price that people were prepared to pay (around \$2,500). Of course, Emu and Akai superseded their products with ever more powerful instruments and other manufacturers entered the arena as well but ultimately, the market was dominated by Akai and Emu.

Until, that is, the advent of software samplers.

With computers getting more and more powerful with faster CPUs and bigger memory in the '90s, it became feasible to create a 'virtual' sampler that used the computer's gubbins instead of dedicated hardware. But therein lies a conundrum - whilst offering very powerful sample playback¹ and processing facilities, they lack the dedicated hardware required for efficient playback of samples with often low quality transposition (interpolation), poor and sometimes inaccurate loop playback (because of other strains being placed on the host computer's CPU that 'interrupt' playback), etc.. And whilst it's all well and good to have samples with high sampling rates and bit-depths that measure in gigabytes rather than megabytes (or kilobytes in the early samplers!), these are taxing on the CPU and reliable performance can often be difficult to achieve (and when a software sampler is hogging almost all of the host computer's resources, it can be difficult to run other music applications).

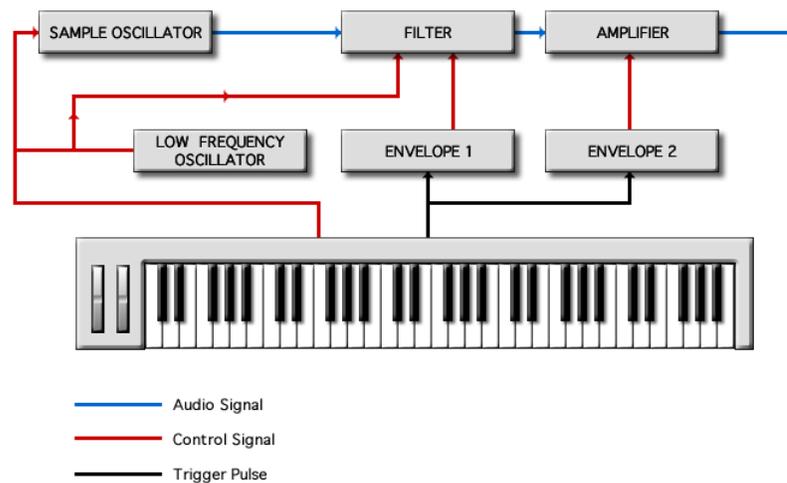
As such, there is still a place for a hardware sampler even today (despite what many would have us believe). It's true that the rack-mount format sampler has pretty much disappeared (though tens of thousands - or more - are still in use daily across the world) but this is probably due more to its form factor - rack-mount gear is generally on the decline because it can be so awkward to operate. But to say that the hardware sampler has had its day is incorrect because products such as Akai's MPC series have never been more popular.

And so it is with Fusion, which brings hi-end hardware sampling in a convenient and integrated workstation that is equally at home in the studio or on-stage. With a specification rivalling (if not exceeding!) many of the top hardware samplers ever made, this is not some afterthought or optional extra - it's an integral part of Fusion to be used alongside the other sound engines with almost unrivalled flexibility.

What follows is a 'generic' description of sampling in general and is not particularly Fusion-specific (that is described in a separate tutorial). Hopefully, with a fuller understanding of the basic principles of sampling, you can coax more from Fusion's powerful sampling and sample playback and processing facilities.

1 Very few software samplers actually 'sample' - they typically require samples to be recorded externally and then imported somehow.

Voice architecture....



Simplified block diagram of a typical sampler

If you have read the analog synthesis tutorial, you will no doubt notice that the diagram shown above looks uncannily similar to a typical analog synthesiser and indeed, they are - the oscillators are simply replaced with samples. After that, the signal flow is almost identical and the samples are processed by filters, envelopes, LFOs.

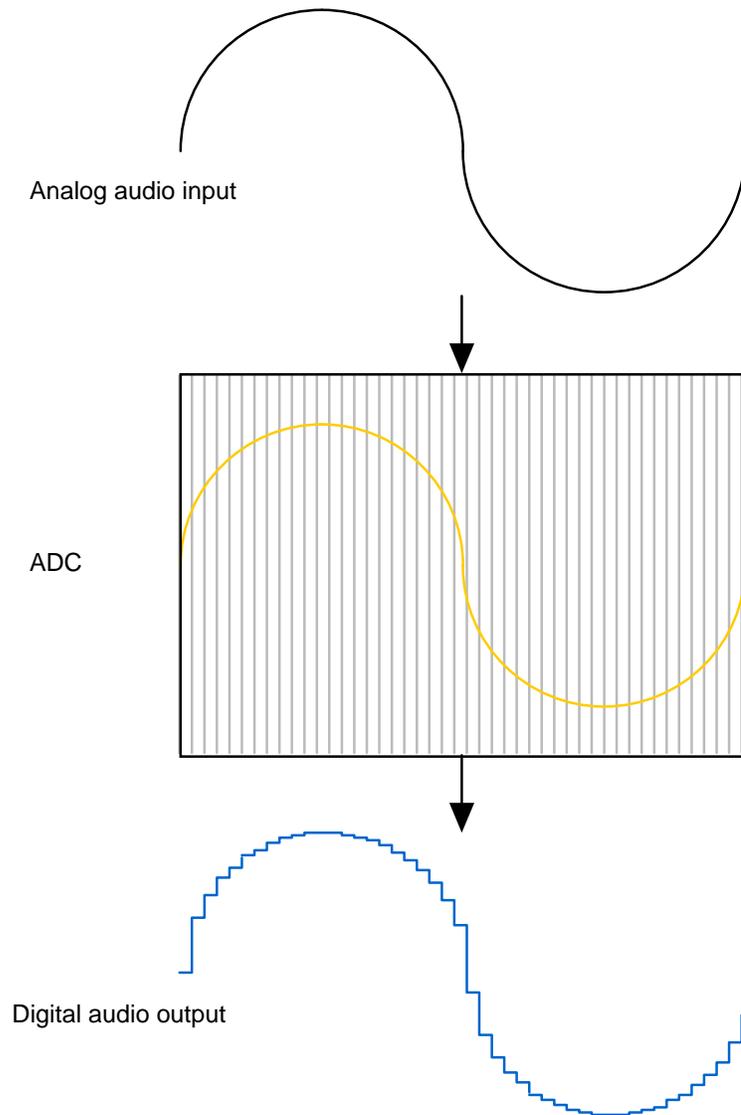
That said, the block labelled 'sample oscillator' above is not as simple as it looks. More often than not, there will be several sample oscillators per voice (typically four as on Fusion) and these can be used to layer samples and/or can be used to switch between different samples (typically using velocity) - we shall come to this in due course during this tutorial. For now, thinking of your sampler as a simple analog synth can help simplify things... it's just that instead of preset waveforms, almost *any* sound can be used as the sound source!

Let's now have a look at what a sample actually is.

Digitising audio

When you record a sample, it is a digital recording. As we know, microprocessors can only deal with noughts and ones. How, then, do we convert continuous analog audio into something that the microprocessor can make sense of?

When we record a signal digitally, it is actually sliced up into tiny fragments in an analog to digital converter (ADC):



As you can see, the continuous analog audio input is sliced up in the ADC and the output is a 'stepped' representation of the incoming signal. Because the waveform is stepped, each of those steps can be given a numeric value that a microprocessor can deal with.

However, because of those steps, the sound is not a truly accurate representation of the original signal. It follows, therefore, that the more slices we take, the more accurate the digital version will be. The number of slices is known as the sample rate.

The 'industry standard' sample rate today is 44.1kHz - that is, every second of audio is sliced 44,100 times. This gives a fairly accurate representation of the sound and is the standard adopted for CDs giving audio reproduction of up to 20kHz, the upper limits of human hearing. In fact, a good analogy can be made with movies.

As we know, movies are actually a series of still photographs which, when played back quickly in succession, gives the illusion of movement.

In the early days of movie making, the cameras used frame rates as low as 12 frames per second (FPS). This is why when we watch those early Charlie Chaplin, Buster Keaton, Keystone Cops and Laurel & Hardy movies, the movement looks jerky (and appropriately comical). Over time, the frame rate increased (it's now 24fps for film) and so, to all intents and purposes, movement appears to be totally smooth. Knowing this, you can begin to understand why early samplers didn't sound so good - with sampling at rates of just 20kHz or maybe 32kHz, fewer slices were taken and so the digitised signal was audibly 'jerky', manifesting itself in a muffled sound lacking in high frequencies. The Audio Engineering Society (AES) settled on 44.1kHz as the standard in 1985.

More recently, however, sample rates of 48kHz, 96kHz, 192kHz and higher are being used and whilst these *can* have a beneficial effect on audio quality, for individual *samples* of instrumental sounds, there are few - if any - practical benefits to be had.

But whilst the sample rate takes care of the number of slices taken on the horizontal axis, what of amplitude? This is measured in 'bits' and/or known as the 'bit-depth' and/or 'quantisation'. In a 1-bit system, you'd have just two levels - on or off... 1 or 0. However, with a 2-bit system, four levels are available (00, 01, 10 and 11). As you increase the number of bits, the range of possible amplitude values (and hence the potential dynamic range) increases exponentially. Thus a 3-bit system can provide 8 values, a 4-bit system can provide 16 values and so on.

In the early days of sampling, 8-bits was the technical limit of the time giving 256 levels which although significantly better, is still pretty poor providing a dynamic range/signal to noise ratio of just 49dB which is less than that of a cheap cassette recorder!

The next generation of samplers gave us 12-bits offering 4,096 amplitude variations and a dynamic range of 60dB whilst the next generation of 16-bit samplers offered a whopping 65,536 levels of amplitude representation and a highly respectable 96dB dynamic range (better even than professional reel-to-reel tape!). It was this 44.1kHz/16-bit system that was (understandably) adopted for CD mastering and is pretty much the practical standard for samplers.

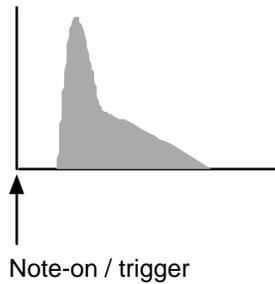
Today, many products use higher sampling rates and higher bit depths and whilst recordings of *performances* can benefit (for example, a band or orchestra playing), these new specs arguably have few real advantages on individual *samples* (for example, single, full level samples of individual piano notes or of drums, etc.) - in fact, in practice, hi-spec samples do little more than increase CPU, memory and disk usage with almost no practical audible benefit at all.

Now... if none of this makes any sense to you (or you are simply not interested!), that's fine! There is no real need to know any of this because these days, when you make a sample, what comes out is pretty much what went in and you don't really have to think about it. What follows, however, *is* relevant if you plan on making your own samples.

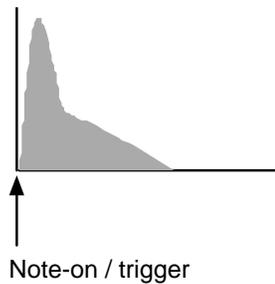
Trimming samples

It is essential to trim samples' start and end times very precisely.

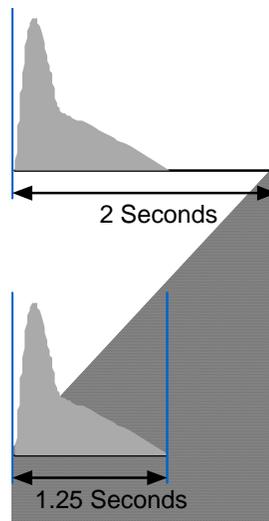
If the start time is not accurate and there is 'dead space' at the front, the samples will start late and out of time:



The sample start time *must* be set accurately to avoid this:



It is similarly important to set the sample end point accurately as well to avoid wasted memory, disk space and slower load times:



With proper end point trimming, we've just cut the sample length by almost half!!

As well as saving memory/disk space and improving load times, however, trimming the end point properly can also help optimise polyphony. If there is 'dead space' after the sample, when that sample is played/triggered, the actual sound is heard and finishes but while the silence after the actual sound is 'playing', you are actually using a voice of polyphony that won't be freed and available until the entire sample has stopped.

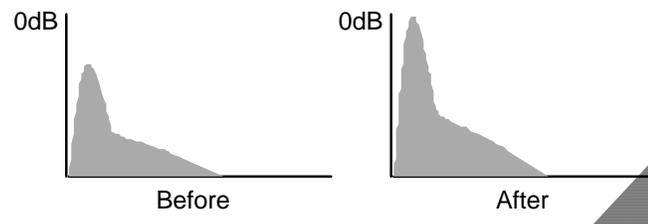
As you can see, it is *essential* to trim the sample start point accurately and it is good practice to trim the end of the sample properly as well. Good, efficient trimming can save memory, disk space, CPU usage, load times and voice usage - go figure!!!

Normalising

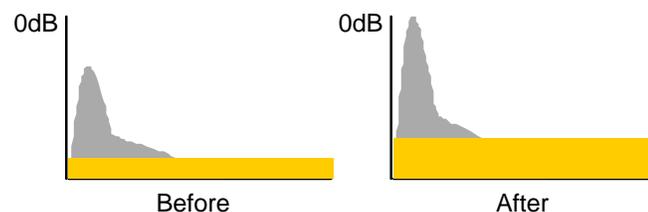
It is always good practice to record samples at as high a level as possible in order to get maximum level and dynamic range out of them (they can always be turned down in level if they are too loud).

As such, when making your own samples, try and record them as 'hot' as possible without clipping/distorting. Distortion in a digital audio system (such as a sampler) is a BAD thing. It's not like the smooth, creamy distortion enjoyed by guitarists in their valve amps, whatever, but is a nasty artefact that is almost impossible to remove and will spoil the quality of a sample.

One way to get a sample's level optimised is to use the NORMALISE function. This will analyse the sample, find its highest level and scale the whole sample's level up accordingly:



But be careful.... if there is any residual noise in the sample, that will also be increased in level:

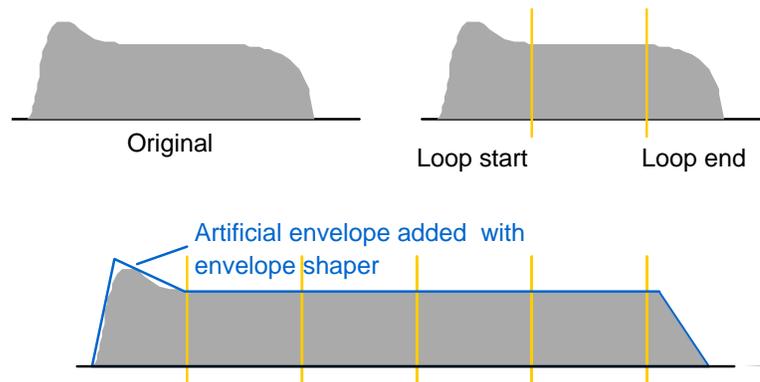


There's not much you can do about this however other than running the sample through some off-line noise reduction process such as is offered by Adobe Audition.

Looping

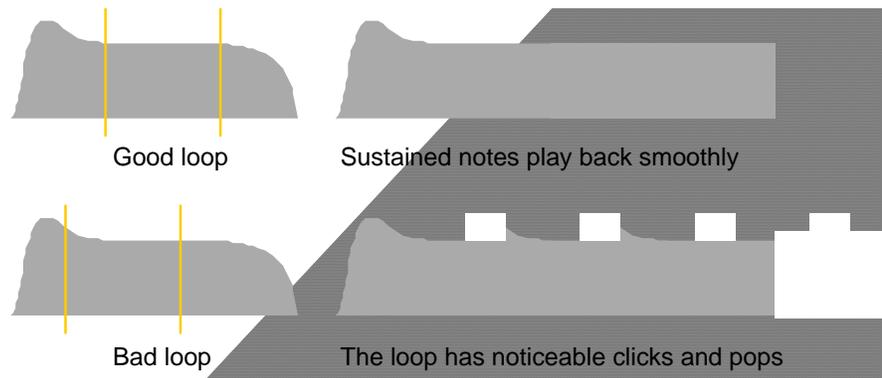
In the early days of samplers, memory was very, very expensive. As such, you simply could not afford to have lots of it (there were technical issues as well) and the original samplers boasted memory specs in kilobytes rather than megabytes (or, more recently, gigabytes). Because of this, samples could not be very long, lasting just a second or two. To overcome this and to artificially extend the length of samples, the process of LOOPING was developed².

Looping relies on the natural phenomenon that most instruments have a distinctive attack (percussive hammer strike on a piano or percussive stick hit on percussion instruments, violin/string section bow scrape, chuffy flute 'blow', raspy trumpet, whatever) but tend to settle down into a steady state of sustain. By carefully selecting loop points after the initial attack when the sound has settled down, it is possible to sustain the sound as long as you want and as long as you hold the note down:



The sound sustains for as long as you want and is artificially 'shaped' using an envelope generator.

In practice, however, even though the sound *appears* to have 'settled down', there can still be subtle changes in amplitude and/or tone in the sustained portion which can make finding the perfect loop quite tricky on occasions - not impossible, just tricky. When making a loop, therefore, you want to be looking at points in the sample where the level and the waveform are the same (or similar) at the loop points if you are to avoid clicks and pops.



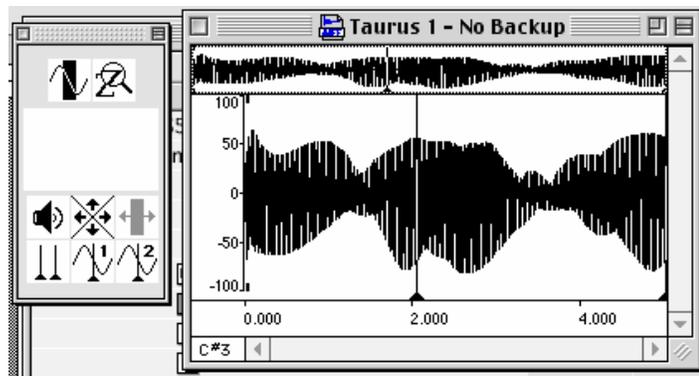
² This is not necessarily to be confused with looping drum patterns or musical phrases which, although related, typically refer to something else.

Even if you cannot avoid small clicks, don't despair because these can often be reduced using a small crossfade at the loop points. However, note that this can also introduce other artefacts such as phase shifts and cancellations which can be as noticeable as a bad loop!

Hardware samplers typically have many tools to assist with looping such as AUTOFIND (finds suitable points in the sample that look like they would loop well) and the aforementioned CROSSFADE. Many also allow you to audition loops in real-time as you adjust the loop points. These are facilities not always present on software sample editors.

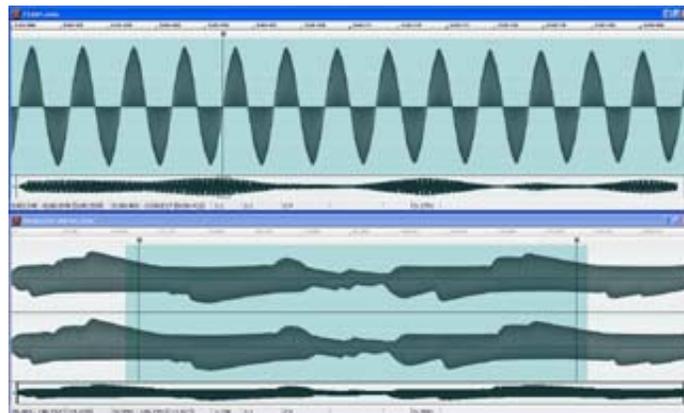
However, in recognition of the difficulties that can sometimes be associated with looping, software applications have been developed specifically for the purposes of looping.

Of these, Antares INFINITY is unarguably the best. It's been around for years and employs some seriously clever algorithms that create seamless loops almost with a single mouse click! There is one problem however - it only works with Mac OS9! In fact, if you look at the screen shot, you can see that its heritage lies in Mac OS7!



Unfortunately, it is unlikely that Infinity will ever be ported to OSX (and it doesn't work in OSX's Classic emulation mode either). It is not available for Windows either. Another small issue is it's large price tag!

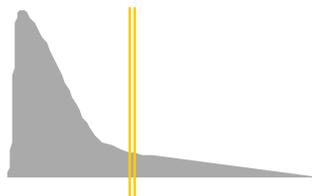
A piece of looping software worthy of note for Windows is the Zero-X Seamless Looper.



Although not a patch on Infinity, it can produce some reasonable results especially at its low price!

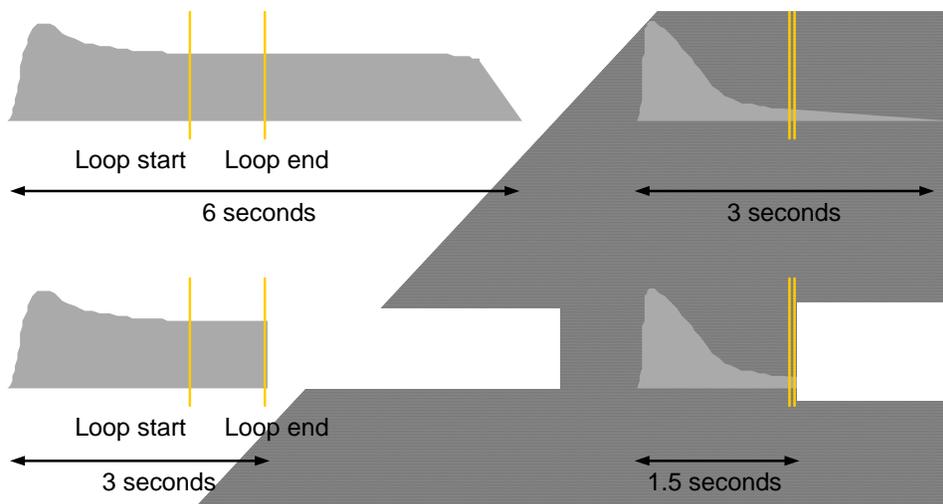
These applications are mentioned simply because these days, it is easy to interface modern hardware samplers with your Mac/PC, do the sample editing and looping on-screen there and then transfer the results back to the sampler. Fusion is no exception - by connecting it to your computer via USB, samples can be transferred back and forth between the two allowing you to use specialised programs such as these on your Mac/PC to help you with one of the trickiest aspects of sample editing. Neither of these applications will magically create a perfect loop for you and you have to use your judgement to pick potentially good loop points manually but they can certainly help you in your looping endeavours.

So far, we have only looked at long loops. Wherever possible, it is best to set as long a loop as possible to reduce the possibility of the ear picking up on any obvious repetition. However, there are occasions where a long loop is possible. This is especially true of instruments that naturally decay such as piano, guitar, harp, marimba, vibes, etc.. In this case, the practice is normally to set a very short loop somewhere towards the end of the sample:



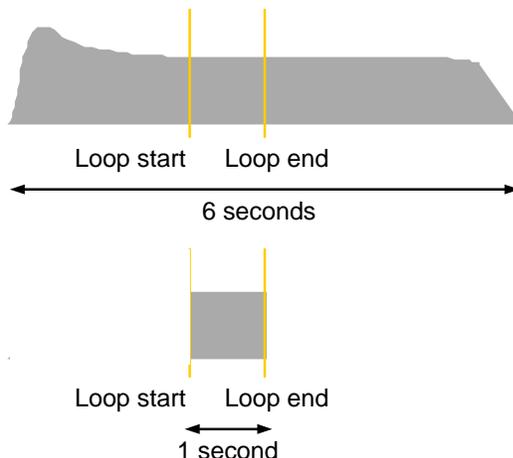
You hear the attack and 'body' of the sample but you nip in with a very short loop as soon as that has settled down. This loop might only be in the region of a few milliseconds and may represent a single cycle of the waveform at that point but it's a useful way of extending the sound and having more control over decay/release. In fact, this was how the early S&S synth sounds were made - sample just the attack and apply a single cycle loop immediately after it. It is still a viable technique today, especially when memory is at a premium.

Once you have found a good loop (however that is achieved - using your sampler's on-board facilities or via some external software editor... or a combination of both... whether it's a long loop or a short, single cycle loop), any data after the loop end can (should!) be discarded because it will never be used:

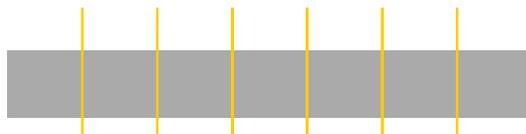


In both examples, we have halved the size of the sample by removing redundant data after the loop. Applying this to other samples can dramatically reduce overall memory usage, disk storage and load times.

In some circumstances, you *could* delete data *before* the loop as well. For example, in the case of sampling a sound to be used as a pad or a leadline, whatever - something that doesn't have a distinct attack - you can make further memory/disk space savings by discarding anything before the loop:



When played, you'll hear this solid 'block' of loop repeating for as long as you hold a note:



You can now apply an 'artificial' envelope to 're-shape' it as you want:



What you have done here is to take a small section of a longer sound, loop it round to sustain indefinitely but applied an envelope to restore the original's shape AND you saved yourself a LOT of memory, clawing back more than 80%!! This sample will also load in 1/6th of the time!!!

Of course, using the sampler's synth facilities, you can apply filters, etc., to the sound to create a good deal of variation out of a very small sample.

Looping is both a science and an art - there is a lot of 'theory' about how and where to set loop points but often, your ears will be the best judge. Loops are often best set using high quality enclosed headphones so that there is no acoustic noise and/or interference that can distract you and maybe mask inadequacies such as clicks and pops in the loop. And, like everything, practice makes perfect - don't be disheartened if your initial efforts don't work out well because it will become easier the more you do it. After a while, you'll be able to just look at a waveform and know (roughly) where to set loop points - after that, you can use the various tools to refine the loop.

Stereo or mono?

Good question!

If you have samples that are obviously stereo and *really* audibly benefit from being in stereo, it's worth using them but note that stereo samples use twice as much memory and will take twice as long to load. As such, it's worth questioning whether *overall* the sound will benefit from being in stereo when it's sitting in a mix. Sure, it might sound fabulous and impressive in isolation but that detail may well be largely lost when that sound is just another component in a mix.

Stereo samples also bring another problem - placing them in the overall stereo image. For example, you might have some impressive stereo lead line sample loaded but you might want to pan that to the left in your mix. With stereo samples, all you will actually do is change its left/right balance and when 'panned' to the left, what you will actually hear is just the left channel of that sample and the right channel will be lost. As such, it might be worth converting these to mono³.

It's worth mentioning that many samples that purport to be stereo are actually just mono samples recorded with one mic but recorded in 'stereo'. Those are definitely worth converting to mono as you will immediately halve the size/memory requirement/load time with no audible effect on the sound.

Another type of 'stereo' sample to watch out for are those that really are stereo (i.e. they have been recorded in stereo with two mics and with stereo ambience for example) but where the actual instrument is placed centrally. This is particularly true of drum sounds which are sometimes recorded this way. It's fine for kick and snare but can be problematic for drums such as toms or for cymbals, percussion, etc., which you may want to position in the stereo image manually because as soon as you try panning these, you upset the samples' stereo balance. In such cases, it's often best to convert these to mono so that you can pan them as you like⁴.

When performing live, it is likely that your PA will be mono or, if it is stereo, subtle stereo details in individual samples will largely go unnoticed in the excitement of the environment. In this case, it's worth converting to mono if only to speed up load times between songs.

It is not being suggested that stereo samples are bad and to be avoided - the above are merely observations of the *practical* applications of stereo samples.... If you want to use stereo samples, please feel free! Just consider *how* those samples will ultimately be used.

3 You can always add a stereo element to the sound by adding some subtle stereo effect such as reverb

Multi-sampling

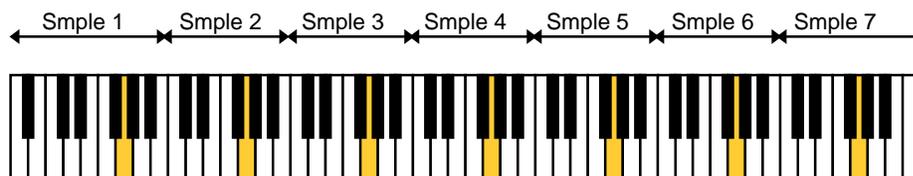
Wouldn't it be nice if all we had to do was take one sample of an instrument and then be able to play it across the entire keyboard and have it sound totally realistic?! If only!! Sadly, this is not possible... well - you *can* play a single sample across the entire keyboard range but as soon as the sound is transposed too far away from the pitch at which it was recorded, it won't sound much like the original! This is for a number of reasons.

Almost all musical instruments vary in tone across their range. A good example of this is the piano and if you play notes across its range, you can hear clearly differences in the different octaves. Sounds such as the human voice, saxophone, oboe and others have fixed frequency components called 'formants' which don't change pitch with different notes - when these are transposed on a sampler, those formants also change pitch and so the sound becomes very unnatural. In fact, the phenomenon is known as 'Munchkinisation' after the little characters in the movie classic, 'The Wizard of Oz', known as the Munchkins. Their voices were created by recording them at half speed and then speeding up playback on a tape machine - the vocal formants were also transposed in pitch giving rise to that characteristic (almost comical) squeaky character.

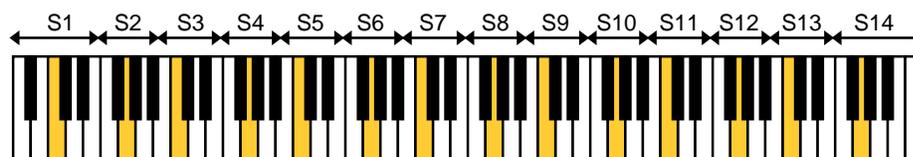
Also, almost all musical instruments' envelopes vary across their range. Once again, the piano is a good example - at low notes, it can take almost a minute for the sound to die away but the very high notes barely last a second. Similarly with guitars and other plucked instruments. Bowed instruments tend to vary their attack time and vibrato speeds being slower on low notes and sharper and faster on high notes.

Put these together and you can see why a complete instrument cannot be represented with just one sample across the keyboard range!

The solution to this is known as 'multi-sampling' where samples of the original instrument are taken at different pitches across its range. For example, you could take samples of a piano at G of every octave. By mapping these samples out from C-B in each octave on the sampler, you stand a better chance of getting a more accurate representation of the sound.



However, even one sample per octave might not be enough and so maybe you need samples at C and G of every octave.



Properly mapped out, your sampled piano is likely to be slightly more authentic.

Of course, the ideal is a sample for every note but therein lies another problem - memory! With so many samples, you need large amounts of memory in your sampler (not to mention a good deal of patience as you wait for them to load!).

Perhaps the best compromise is a sample every minor third (i.e. C, E#, F# and A) of every octave.



In this way, the original samples are never transposed more than a semitone either way and so the artefacts described on the previous page go largely unnoticed even on quite critical sounds (even vocal sounds which are notoriously difficult). Compared with an instrument sampled on every note, the size of the sampled instrument at minor thirds is also quite dramatically reduced to a third - instead of (in the case of a piano) having 88 samples, you will have 30 and instead of 300Mb (whatever), it will be a more modest 100Mb.

It really is worth trying to understand multi-sampling as it is fundamental to successful sampling.

It is also fundamental to sampling and laying out drum kits because although we're not dealing with pitched instruments here, we use the keyboard to trigger different drum sounds. For example:

C1	Kick 1
C#1	Claps
D1	Snare 1
D#1	Rimshot
E1	Snare 2
F1	Floor Tom
F#1	Closed hi-hat
G1	Low Tom
G#1	Medium hi-hat
A1	Mid Tom
A#1	Open hi-hat
B1	High Tom
C2	Crash cymbal

Etc..

The principles are almost exactly the same except that typically, each different drum is uniquely on its own key and not spread across several⁴.

Of course, unless you are doing your own sampling on Fusion, much of this is largely hidden from you because you are simply dealing with 'multi-samples' when assigning sounds to sampler programs and all the mapping, etc., has been done for you. In fact, this rather sets Fusion apart from other samplers - even if you buy an expensive library for a 'traditional' sampler and want to make your own program, YOU have to individually assign and map every single sample in the program. On Fusion, you simply select a pre-mapped multi-sample.

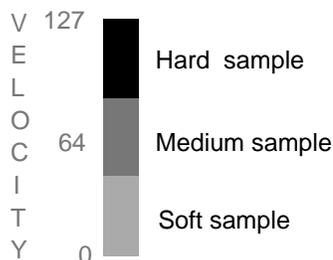
However, if you are planning on sampling your own sounds (your guitar or bass or piano ... or another synth/keyboard ... whatever), it's best to get acquainted with the concept of multi-sampling if you want to accurately represent it in your sampler.

⁴ For creating multi-sampled drum programs, Fusion helps to do away with much of the mapping complications of 'traditional' samplers by offering a special DRUM program where the keys are pre-defined and pre-mapped - all you do is slot your drum samples in on them.

Velocity switching

Just when you thought you were out of the water, there is yet another level of sampling that needs to be considered!

As well as instruments' tone varying across their pitch range, they also tend to vary according to how they are played. Once again, the piano is a good example - played softly and the tone is quite soft and muted; played hard and the tone is quite bright. The same is true of almost every instrument. To replicate this, you need to take several samples for each note - typically soft, medium and hard - and set those up so that you switch between them using velocity:



More recently, because of the ever increasing memory capacity of computers, developers of library for software samplers have gone completely bonkers, offering as many as 32 (or more!!!) velocity layers but frankly, this is not strictly necessary. In fact, velocity switching isn't always necessary at all because we can go some way to replicating this natural phenomenon by using a lowpass filter controlled by velocity to 'fake' the natural response of most instruments and it can be very effective (and arguably 'smoother' than abrupt switching between samples)⁵.

If you have read the Analog Synthesis Tutorial, you will have learned that the lowpass filter can be used to regulate high frequency components in a sound - with the cutoff frequency fully open, all the high frequencies pass through but as the cutoff frequency is lowered, so the higher harmonics are filtered out. If we control the cutoff frequency using velocity so that playing hard opens the filter fully (creating a bright sound) and playing softly brings the cutoff frequency down (creating a more muted sound), you can begin to see how such a filter can be employed to re-create what occurs in nature.

Of course, this won't go all the way in *accurately* duplicating the phenomenon 100% but in many circumstances, it can often be sufficient especially if the sound in question is going to be just a component part of a busier mix with other sounds/instruments vying for attention.

There is no denying that velocity switched samples *can* sound more dynamic but don't be fooled into believing that you *have* to take multiple samples for every note you sample. It can certainly be beneficial but is by no means *essential* depending on what it is you are sampling and ultimately how it will be used.

One also has to think of other practicalities. For example, is it really worth sampling an instrument at several different velocities if, at the end of the day, you're going to bang away at it at full velocity most of the time - you are filling your memory up with redundant samples that won't be played!!

5 Using a velocity controlled lowpass filter with velocity switched samples can help smooth out any abrupt transitions between the different samples.

Sample start control

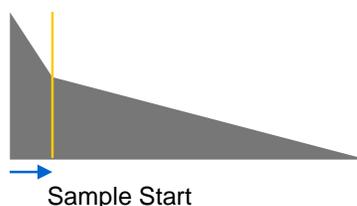
Another way getting more out of fewer (or even a single) velocity samples is to use velocity to control sample start position.

When an acoustic instrument is played hard, it will typically have a more pronounced attack. Once again, we will use the piano as an example. When a key is played softly, the hammer barely touches the strings resulting in a soft attack (not to mention a softer sound in general because few high frequencies in the string(s) have been 'agitated'). Play slightly harder and the hammer hits the strings slightly harder resulting in a more pronounced percussive attack (and a slightly brighter sound). Hit the key very hard, however, and the hammer hits the strings like a speeding bullet resulting in a very percussive attack (and also a very bright sound as the high frequency harmonics are excited into action).



However, we can go some way to emulating this by controlling sample start with velocity.

By taking just a hard sample and setting sample start a little way into the sample, we can effectively 'bypass' the initial, pronounced hammer component in the sound.



By controlling that sample start position with velocity so that soft playing bypasses the initial hammer 'thunk', medium playing reveals more of it and high velocity plays the hammer component in its entirety, we can go some way to emulating the sound's natural envelope at different velocities.



Of course, it will sound odd and somewhat unnatural repeatedly playing a bright sample at different velocities. A simple remedy to this is to use a lowpass filter controlled by velocity so that soft playing keeps the filter cutoff down to produce a more muted sound, medium playing opens the filter up a little for a slightly brighter sound and hard playing opens the filter right up allowing the bright sound to play as it should sound.

And by applying a decaying envelope to the same cutoff filter, we can also go some way to reproducing the natural phenomenon that high frequencies decay and die away quicker than low frequencies. With some careful programming, it's possible to get a lot of mileage out of just one velocity sample and actually make it sound passably realistic within the context of a mix.

It's not identical to 'real' velocity switching, of course, and it might not stand close scrutiny if the sound were to be played and highlighted in isolation but for non-critical sounds that will just sit in a mix or which will be played in less demanding live situations, it can be a remarkably effective way of saving memory as well as speeding up load times.

The same techniques can be used on many different sounds. Drums can benefit from this technique as can tuned percussion such as marimba or vibes. Scrapey, bowed instruments and many 'blown' instruments such as flute and/or brass can also benefit. Interestingly, samples of synth sounds with a pronounced attack (for example, a resonant synth bass with a distinctive sweep at the attack) can also be made surprisingly more playable.

Even if you don't control sample start with velocity, adjusting the sample start can have a dramatic effect on a sound and it's sometimes worth assigning the parameter to an assignable knob to have quick access to it.

Sampling - conclusion

We have covered the basics here. Successful sampling isn't easy especially if extensive multi-sampling and keyboard mapping is involved but, once you have a basic understanding of the principles involved, like everything, it's not exactly rocket science either. The key is to approach it logically, keep your wits about you and apply some common sense.

Looping *can* be difficult depending on the sound but there are numerous tools these days to make this easier. In really difficult cases, you have to be pragmatic and just accept that maybe an absolutely smooth and perfect loop is not achievable and/or some compromise is required. And if push comes to shove, don't loop it - record more than you think you'll ever require and forget looping completely... this is perfectly valid if a bit wasteful of memory.

And talking of which, if you think conserving memory has been somewhat over-emphasised during the course of this tutorial, it's for good reason - the more you can optimise a sample to have as small a 'footprint' as possible, the more samples you can load into memory, the quicker they will load (a prime consideration if you're playing live on-stage), the less disk space they will require to store and the less strain they will place on the CPU (thereby allowing greater polyphony). Many might regard such practices to be outdated, a bit 'old school' and largely irrelevant today but there is not a sound or situation that doesn't benefit from having lean, mean samples whether it's on-stage or in the studio!!!!

Synthesiser processing

In the early days of samplers, they offered very little in the way of synthesiser functions - a simple envelope generator perhaps and maybe a simple LFO to add vibrato.

Today, the distinction between sampler and synth is very blurred (in fact, many products purporting to be synths are actually sample based 'ROMplers' with multi-samples as their basic waveforms).

The sampler in Fusion has almost exactly the same 'post-processing' synthesis functions for samples as the VA synth engine. Please refer to the VA synth tutorial for details on the filters, envelopes, LFOs, controllers, etc..