iXML

Paul Isaacs of Applied Audio Consultants takes a look at a metadata standard that is beginning to improve our working lives.

If you have not yet heard of 'iXML' you might be forgiven for concluding that it is a derivative of the XML programming language adapted by Apple Mac who, with their iMACs and iPODs, have been largely responsible for thrusting the lower case 'i' into the forefront of the consumer's psyche and thus making it the star of one of the most successful product branding exercises this century.

So, you may be surprised to know that the 'i' actually refers to none other than the IBS, our very own Institute of Broadcast Sound, and it acknowledges the fact that the IBS played a key role in iXML's conception. In this article, I will present an overview of iXML and how this metadata interchange standard is beginning to demonstrate its potential as an effective aid to production workflow.

Industry Standards

You might be wondering whether there is any need – let alone any room – for yet another 'standard' in an industry where so called standards come and go like *Pop Idols* and *Big Brother* contestants! Although iXML is not yet a formal 'standard' the plan has always been to have it accepted as one, and it has already been adopted by many well-known audio manufacturers – and with good reason. It is comprehensive, extensible, straightforward to implement, and does not break compatibility with existing processes.

The form of iXML is essentially a chunk of metadata embedded within a Broadcast Wave audio file (although it can also be embedded in other file types too), and it fully describes the attributes of the recorded audio. The Wave format is founded upon a chunk-style architecture that allows for optional chunks of data to be introduced without upsetting compatibility; tools that are not programmed to read the iXML chunk will simply ignore it.

Exploiting the benefits of the Extensible Markup Language (XML), iXML provides a simple text-based method for storing and exchanging metadata. It is best considered as an enrichment and expansion of the widely used BEXT chunk – although that is beginning to show its age now, in an era where the metadata storage requirements of non-linear multi-track recording exceeds what BEXT is able to provide.

Small Beginnings

The foundations for iXML were conceived in an open debate between manufacturers and users at the Olympia Production Show in early 2004. Healthy debate highlighted a range of issues relating to post-production workflow that needed addressing in order to help current film and TV workflow practices. Furthermore, there seemed to be a tangible desire amongst the manufacturers present to collaborate on developing a solution. Infuriatingly, though, that debate ended just at the point where a glimmer of hope for a solution was emerging.

> **"The IBS played a key role in iXML's conception"**

In an attempt to prevent this progress from going to waste, I approached John Andrews (IBS Chairman) and Mark Yonge (MIBS and AES Standards Manager) to suggest that the IBS was the ideal forum to take up the mantle of ensuring that matters progressed. They wholeheartedly agreed and set about planning a brainstorming meeting at the BBC's White City offices in July. This meeting was attended by key engineering representatives from many of the non-linear editing (NLE) workstation, location recorder, and post-production software manufacturers, along with representatives of several leading UK post-production facilities.

In addition, the IBS established a closed subgroup web forum to provide a convenient means for manufacturers and other key industry figures to share and develop ideas. The forum's list of subscribers included the likes of Aaton, Apple, Ambient, Avid, BBC, Dark Matter Digital, Digidesign, Fairlight, Fostex, Gallery, HHB, Lightworks, Merging Technologies, Nagra, Sadie, Sound Devices, Sound of Voices, Steinberg, Synchroarts, Videosonics and Zaxcom. The whole collaboration was characterised by a noticeable lack of 'competitive hat wearing' – perhaps because there was no longer a need for those present to conceal evidence of loss of hair due to it being pulled out in frustration!

The July meeting included in-depth discussions about audio file naming conventions and embedded metadata. It was clear that a metadata interchange standard solely based on file name structure had serious restrictions. A file name is too easily changed either manually or automatically, accidentally or otherwise during transfer. Furthermore, file naming systems have different formal naming rules, 'legal' character sets and limited file name lengths. Even without the inherent limitations of file-naming, it quickly became apparent that it was unlikely a formalised file name structure could ever be agreed upon – some users wanted descriptive file names based on scene and take number, while others wanted them based on a roll ID plus incrementing segment number.

In the end, the manufacturers decided they would focus their efforts on an embedded, non-tamperable solution, leaving exchange of information via file name as a plan B – file names were still deemed to be of some use as they provide post facilities with another way to locate and conform audio files.

Simple Implementation

For iXML to be widely and quickly adopted, all agreed that it should be very simple to

"iXML is metadata embedded within a Broadcast Wave file"

implement. Incredibly, the first lines of iXML code were being drafted by a handful of engineers during the course of that meeting at the BBC – a sure sign of its simplicity and potential.

Thirty months later, iXML has evolved into a mature specification, which, by way of the additional storage space and metadata that it contains, offers a significant number of advantages over its predecessor, BEXT. Apart from logging the majority of BEXT metadata items such as scene name, take number, track name, time-code, frame rate, notes and tape ID, iXML introduces some new metadata objects including *Project, Circled, Speed, Sync-point-list, History, File-set*, and *Track-list*. Some of these objects contain several sub-objects, but it is beyond the scope of this article to discuss all these in detail. Instead, I'll focus on several of the more interesting metadata items... Did I just say 'interesting' and 'metadata' in the same sentence? Someone help me!

The *Sync-point-list* object provides a method for communicating the position of multiple cue points and/or event regions within an audio file. This is currently being used to great effect by the likes of Aaton and Gallery to log a slate mark within the audio file when a clapper is closed. Aaton's Cantar recorder can auto-detect a clap by analysing an audio file's characteristics just after the start of a recording. Gallery's Mark Gilbert and Peter Schneider of Gotham Sound have also successfully tested a prototype 'METAslate' system using a clapper that transmits a pulse to a recorder's GPI inputs via RF at the moment it is closed. Both Aaton's Cantar and Gallery's Metacorder log auto-detected claps within iXML's *Sync-point-list* as a sample count from the beginning of the audio file, thus eliminating sync problems in telecine and post-sync issues relating to timecode drift or blurred smartslate time-code images.

Aaton's Indaw system has the ability to locate directly to these sync points, and Metacorder's Final Cut Pro XML export delivers pre-assembled masterclips which include all the METASlate sync points right into the FCP bin ready to use. For assistants syncing dailies in InDaw or Final Cut Pro, the *sync-point-list* object dramatically speeds up the process, and it is quite likely we will see other manufacturers adopt the *sync-point-list* for these purposes in the near future.

Mono or Poly?

One of iXML's key strengths is its ability to provide a cross-reference between sibling files from the same take, particularly in a multi-track recording consisting of several mono files or a combination of mono and poly files. The question of whether to record mono or poly files has been around for some time, and even now can be a cause for concern. To summarise, an 8-track mono file recording is represented by eight separate audio files, whereas an 8-track poly file recording is represented by a single file containing all eight tracks interleaved together.

Both formats have advantages and disadvantages, but most importantly it is how AVID handles mono and poly files that determines which is preferable. Avid is compatible with both, but in general its system does not work so smoothly with mono files. This is because Avid has no method for associating imported mono files from the same take, whereas an imported poly file is de-interleaved into its constituent mono files with each linked to the other by a common *Tape-ID*. This is particularly useful for tracking back to, and auto-conforming from, original audio files.

Built in to iXML is the potential for circumventing decisions of whether to record mono or poly files, as it provides an inherent association between parent and sibling files using its *File-set* and *History* objects. The *File-set* object's main purpose is to convey the total number of sibling files belonging to a take, and then to associate them together by giving each the same ID, known as a *Family-UID*. This allows conforming software to rebuild the family of files that make up a particular recording. This is further aided by the *History* object which allows tracking of a file's origins using the *Parent-filename* and *Parent-UID*. Moreover, the *Original-filename* parameter allows for the tracking back to a parent file that may have been renamed accidentally or otherwise.

With increasingly more tracks being recorded on location, the requirement to identify and describe a particular track has become more important than ever. The *Track-list* object goes significantly further than BEXT in describing track properties. Not only does it log descriptive track names, it also keeps a record of *Channel-index* – the original track number that the audio was recorded on. For example, if Take 1 comprises of a four-track poly file made up from tracks 1, 3, 5, and 7, and Take 2 comprises of another four-track file, but one made up from tracks 2, 4, 6, and 8, a workstation is able to import and place those takes on the correct original tracks, as opposed to both takes appearing as tracks 1-4 – which wouldn't be very helpful to an editor!

In addition, a *Function* parameter identifies the explicit purpose of the audio track. It can log whether it is a Left channel, or Right, Mid, Side, Left Surround, or whatever – the iXML specification provides a dictionary of possible track functions. Sound Of Voices' PrismX software is a polyphonic file audition tool that reads the *Function* parameter and automatically applies the appropriate panning, MS decoding or WXYZ (Soundfield B-format) surround decoding, thus saving valuable mix setup time.

NTSC Conversion

Although non-linear file-based recording has provided the industry with a wealth of benefits, it has also introduced several large cans of worms. Apart from having the potential for resolving the aforementioned poly/mono file issues, iXML also provides tags for helping post facilities resolve the playback speed issues commonly associated with 29.97 fps NTSC and more recently 23.98 fps HDTV workflows.

Much of the confusion arises from the fact that most post production equipment (and even different versions of the same equipment) designed to handle video and/or audio do not deal with sample rates other than 48kHz in a universally predictable way – some will sample rate convert, others will adjust their playback speed, and some will simply not work correctly at all. Actually, it is far more complex than that – every production decision such as which shooting format (film, SD or HD), deliverable medium (film, DVD, TV, HDTV) can affect the choice of audio sample rate used on location. It is therefore not surprising that post facilities frequently encounter audio that drifts or that does not sync easily with picture.

There are some well-known solutions for handling the more common workflows. For instance, in the film to NTSC workflow,

film at 24 fps is converted to video at 29.97 fps by using a 2:3 pull-down process that results in an extra six frames of video for every 24 frames of film – providing a total of 30fps. To achieve the NTSC video rate of 29.97fps it is necessary to slow down the telecine transfer speed by 0.1%, and consequently, it is also necessary to apply the same 0.1% pull down to the audio that was recorded on location.

To ensure the audio ends up at the 48kHz sample rate required by a video recorder after that 0.1% pull down has been applied, the source audio files are often recorded on location 0.1% faster – at the rate of 48.048kHz. However, it is common for these files to be 'fake-stamped' in their file header as 48kHz, and this fake stamping causes devices such as the Fostex DV40 and DV824 to playback the 48.048kHz files at a true 48kHz – in other words 0.1% slow, which is exactly what is required to maintain sync with the 0.1% slowed picture.

This is a relatively simple and well-known method, but there are countless alternative methodologies designed to handle different scenarios that would benefit from 'knowing' not only the file header sample rate stamp, but also the actual recording sample rate. The *Digitiser-sample-rate* parameter within iXML logs the actual sample rate for that purpose. Playing back at the *Digitiser-sample-rate* will enable the true recorded speed without any speed adjustment. There is also a *Timestamp-sample-rate* parameter which records how the crucial BWF timestamp (stored as a sample count since midnight) was calculated by the recording application. These extra elements of information effectively tell the whole story about a recording's speed, and allow a post facility to determine correct playback speed rather than having to waste time guessing how each recording was made.

Summary

There is a great deal more to iXML than I have the space to describe here. For those interested in finding out more, a full description of the iXML specification is available at *www.ixml.info* – a website kindly provided courtesy of Mark Gilbert at the Gallery.

Time will tell whether iXML becomes a long-term fixture of our industry, but the signs are very positive; Aaton, Digidesign, Fostex, Gallery, HHB, Merging Technologies, Sadie, Sound Devices, Sound of Voices, Synchroarts and Zaxcom have already released iXML-compatible products, and the list continues to grow.

In recent weeks, the iXML group has agreed the use of an iXML logo to promote the format. Manufacturers who publish (and make available online) an iXML implementation chart to help clients plan their workflows, are able to advertise their product as iXML compatible by displaying the iXML logo.

"iXML has evolved into a mature specification"