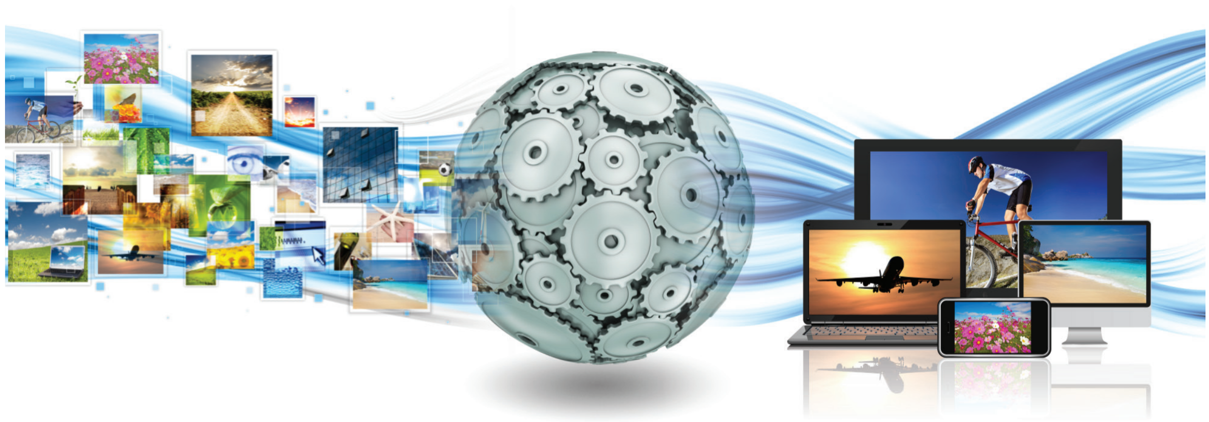


# The Essentials of Video Transcoding

## Enabling Acquisition, Interoperability, and Distribution in Digital Media

The intent of this paper is to provide an overview of the transcoding basics you'll need to know in order to make informed decisions about implementing transcoding in your workflow. That involves explaining the elements of the various types of digital media files used for different purposes, how transcoding works upon those elements, what challenges might arise in moving from one format to another, and what workflows might be most effective for transcoding in various common situations.

### Contents



### Introduction

It's been more than three decades now since digital media emerged from the lab into real-world applications, igniting a communications revolution that continues to play out today. First up was the Compact Disc, featuring compression-free LPCM digital audio that was designed to fully reconstruct the source at playback. With digital video, on the other hand, it was clear from the outset that most recording, storage, and playback situations wouldn't have the data capacity required to describe a series of images with complete accuracy. And so the race was on to develop compression/decompression algorithms (codecs) that could adequately capture a video source while requiring as little data bandwidth as possible.

The good news is that astonishing progress has been made in the ability of codecs to allocate bandwidth effectively to the parts of a signal that are most critical to our perception of the content. Take, for example, the Advanced Audio Codec (AAC) popularized by Apple via iTunes. Its fidelity may be debatable for purists, but the fact is that 128 kbps AAC delivers audio quality that is acceptable to a great many people in a great many situations, and it uses only one-eleventh of the bitrate of CD-quality LPCM (16-bit/44.1 kHz). Even more impressive are the capabilities of the h.264 codec, which supports encoding of 1080p HD material at bit rates of 6Mb/s or less – a compression ratio of 250:1 or more!

With success, of course, comes competition and complication. Codecs make possible the extension of digital content into new realms – the desktop, the smart phone, etc. – each of which has constraints that are best addressed by optimizing a codec's performance for a particular platform in a particular situation. The result is an ever-expanding array of specialized codecs. Some codecs are used for content acquisition, others for editing and post-production, still others for storage and archiving, and still more for delivery to viewers on everything from big-screen TVs to mobile devices. Because content is increasingly used in a multitude of settings, it's become increasingly important to be able to move it efficiently from one digital format to another without degrading its quality. Simply put, that is the purpose of transcoding.

Perhaps the easiest way to realize the importance of transcoding is to imagine a world in which it doesn't exist. Suppose, for example, that you manage production for a local TV news team that has nabbed a bombshell interview. You've got an exclusive on the story only until the following day, and you need to get it on the evening news. Unfortunately, the digital video format of your ENG camera uses a different codec than that of your non-linear editor (NLE). So before you can even begin to edit you'll have to play the raw footage from the camera, decompressing it in real time while simultaneously redigitizing it in a format that your NLE can handle. Now suppose that you have to go through the same painfully slow routine to get the edited version to your broadcast server, and again to make a backup copy for archiving, and again to make a set of files at different bitrates for different devices to stream from your station's website. The inefficiency multiplies as you move through the process.

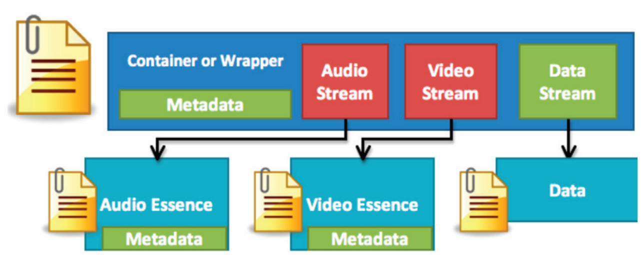
Transcoding bypasses this tedious, inefficient scenario. It allows programs to be converted from codec to codec faster than real-time using standard IT hardware rather than more costly video routers and switches. And it facilitates automation because it avoids manual steps such as routing signal or transporting storage media. Intelligently implemented and managed, transcoding vastly improves throughput while preserving quality.

## Digital media formats

The first rule of digital media file formats and codecs is that their number is always growing. A glance back in time will reveal some older formats that have fallen by the wayside, but also shows that there are far more formats in regular use today than there were fifteen or even five years ago. Different formats arise to fill different needs (more on this later), but they all share some basic characteristics.

At the simplest level a video-capable file format is a container or wrapper for component parts of at least three types. One type is video stream data and another is audio stream data; together the elementary streams of these two content types are referred to as the file's "essence." The third component is metadata, which is data that is not itself essence but is instead information about the essence.

Metadata can be anything from a simple file ID to a set of information about the content, such as the owner (rights holder), the artist, the year of creation, etc. In addition to these three, container formats may also support additional elementary content streams, such as the overlay graphics used in DVD Video menus and subtitles.




Component parts of a media container file

The following are examples of container file types that are in common use today:

- QuickTime MOV, an Apple-developed cross-platform multimedia format
- MPEG-2 Transport Stream and Program Stream, both developed by the Motion Picture Experts Group, with applications in optical media (DVD-Video, Blu-ray Disc) and digital broadcast (e.g. ATSC).
- WMV and MP4 for online distribution
- GXF (General eXchange Format), developed by Grass Valley and standardized by SMPTE
- LXF, a broadcast server format developed by Harris (now named "Imagine Communications")
- MXF (Material eXchange Format), standardized by SMPTE

Container formats such as those listed above differ from one another in a multitude of ways. At the overall file level, these include the structure of the header (identifying information at the start of a file that tells a hardware device or software algorithm the type of data contained in the file), the organization of the component parts within the container, the way elementary streams are packetized and interleaved, the amount and organization of the file-level metadata, and the support for simultaneous inclusion of more than one audio and/or video stream, such as audio streams for multiple languages or video streams at different bitrates (e.g. for adaptive streaming formats such as Apple HLS or MPEG-DASH).

At the essence level, format differences relate to the technical attributes of the elementary streams. For video, these include the frame size (resolution) and shape (aspect ratio), frame rate, color depth, bit rate, etc. For audio, such attributes include the sample rate, the bit depth, and the number of channels.

Container file formats also differ greatly in the type and organization of the metadata they support, both at the file level and the stream level. In this context, metadata generally falls into one of two categories:

- **Playback metadata** is time-synchronized data intended for use by the playback device to correctly render the content to the viewer, such as:
  - captions, including VBI (Line 21) or teletext in SD and VANC in HD;
  - timecode (e.g.VITC);
  - aspect ratio (e.g. AFD);
  - V-Chip settings;
  - copy protection (e.g. CGMS).
- **Content metadata** describes the content in ways that are relevant to a particular application, workflow, or usage, such as:
  - for Web streaming: title, description, artist, etc.;
  - for television commercials: Ad-ID, Advertiser, Brand, Agency, Title, etc.;
  - for sporting events: event name, team names, athlete names, etc.

File formats are designed with various purposes in mind, so naturally the metadata schemes used by those formats are not all the same. A metadata scheme designed for one purpose may not include the same attributes as a scheme designed for a different purpose, which can greatly complicate the task of preserving metadata in the transcoding process.

## The essence of compression

In many cases the most important difference between file formats, and the reason transcoding is required between them, is the codecs they support and the amount of compression applied to the video and audio streams that comprise each format's essence. Those factors are determined by the purpose for which each format was defined. The common video file formats fall into a half-dozen different usage categories, including acquisition, editing, archiving, broadcast servers, distribution, and proxy formats.

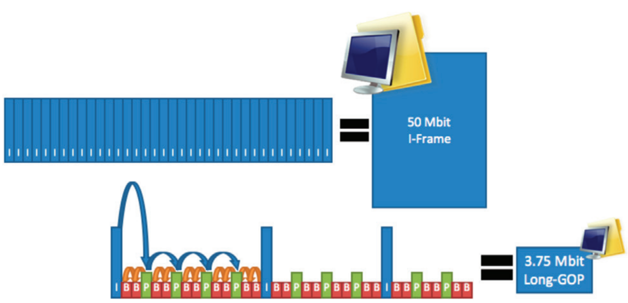
To understand why different compression schemes are used in these different categories we need a quick refresher about how compression works. In video, compression schemes are generally based on one or both of two approaches to reducing the overall amount of data needed to represent motion pictures:

- Intra-frame compression reduces the data used to store each individual frame.
- Inter-frame compression reduces the data used to store a series of frames.

Both compression approaches start with the assumption that video contains a fair amount of redundancy. Within a given frame, for example, there may be many areas of adjacent pixels whose color and brightness values are identical. "Lossless" intraframe compression simply expresses those areas more concisely. Rather than repeat the values for each individual pixel, the values may be stored once along with the number of sequential pixels to which they apply.

If greater compression is required, you then move into the realm of "lossy" compression, meaning that you begin discarding some of the information that you would need in order to reconstruct the original image with perfect accuracy. Adjacent pixels that are similar in color and brightness will be treated as if they are actually identical, so there's more redundancy and therefore less data is required to describe the overall picture. This reduces bandwidth requirements but increases the chance that the viewer will begin to see compression artifacts at playback, such as "blockiness" rather than smooth color gradients.

Inter-frame compression also assumes redundancy, in this case over time across a "group of pictures" (GOP) that may be as short as 3 frames or as long as a few hundred frames, depending on the format and specific use case. The first frame in each GOP, (called the I-frame in MPEG-2, or the "IDR Frame" in h.264 and h.265) is stored in full while the subsequent frames (called B-frames and P-frames) are stored only in terms of how they differ from the I-frame or from each other. The greater the frame-to frame redundancy, the less data it takes to store the differences. If limited bandwidth requires greater compression, a pixel that is similar from one frame to the next may be treated as if it is identical, thereby increasing redundancy. Inter-frame compression allows a drastic reduction in the overall amount of data, but it also means that at playback the only frame that can be reconstructed independently (without reference to other frames) will be the I-frame.




Compared to codecs using intra-frame compression only (top), inter-frame compression (bottom) reduces bitrate by storing only frame-to-frame difference data for most frames.

## The right tool for the job

The way that intra-frame and inter-frame compression are used (or not) for the video stream in a given file format has a direct impact on the suitability of that format for various tasks. If you're choosing a format for editing, its video codec must support independent access to every frame, so intra-frame compression is usually the ideal choice. If you're choosing a format for longterm archiving, then independent access to every frame is not a priority, so you can significantly reduce your bitrate and file size by using both intra-frame and inter-frame compression. Distinct but similar considerations apply to audio as well, with differences in supported audio codecs once again affecting suitability for different purposes.

The following are examples of different applications for which container formats are used, and the types of codecs that are commonly used to compress essence in those applications:

- Editing applications use formats such as Apple's ProRes, Avid's DNxHD, and Sony's IMX. These formats typically use only intra-frame compression, resulting in video streams comprised entirely of I-frames that can each be decompressed on their own. The quality of the source is maintained and the material is easy to seek and edit on individual frames. But resulting bitrates can be high: 150 Mb/s, 220 Mb/s, or even higher. The easiest approach to editing the accompanying audio streams is to leave them as uncompressed LPCM.
- Archiving applications require high quality, and so may be handled using JPEG2000 or DV video elementary streams within a QuickTime or AVI container. But reducing storage requirements is often a higher priority than maintaining independent access to every frame, so inter-frame compression may be applied as well, typically using long-GOP (15-frame) formats such as MXF wrapped MPEG-2 (or h.264) at 36Mbps or higher. Audio streams are often left as PCM or encoded to Dolby E, which uses two channels of 20-bit PCM to store up to eight channels of sound.
- Broadcast server formats are designed for use with proprietary systems, such as the Grass Valley K2, Harmonic Spectrum, or Imagine Nexio, that serve as the source from which stations pull program for the feed to the broadcast transmitter or cable signal distribution system. Editability is not a requirement. The video compression employed is typically long-GOP MPEG-2 or h.264. Audio may or may not be compressed.

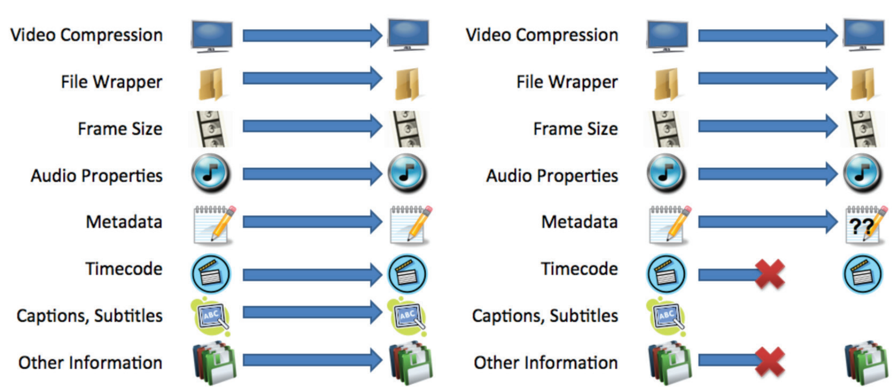
- Distribution is a diverse category covering everything from CableLabs MPEG-2 transport streams for video on demand (VOD) to adaptive streaming formats (Apple HLS, Microsoft Smooth Streaming, Adobe HDS or MPEG-DASH) for "new media" distribution via desktop or mobile device. A single container may include streams of different bitrates and resolutions (e.g. 320 x 240 or 1280 x 720). H.264 is a common video codec for this application. For audio, common codecs include Dolby Digital (AC-3), DTS, Advanced Audio Coding (AAC), MP3, and Windows Media Audio (WMA).
- Proxy formats are small, highly-compressed stand-ins for full-quality source files, allowing fast, easy viewing of material when bandwidth or playback power is limited. A stock footage library, for example, may use proxies to allow prospective customers to browse for suitable scenes, after which they are supplied with a high-resolution, lightly compressed version of the footage they choose. Similarly, editors can use proxies to quickly find the start and end timecode points of takes that they want to include in their final production, without being in a high-powered editing suite. Since the priority is to make the files small, proxies typically use lossy inter-frame video compression.

## Bridging the gaps

With a fuller understanding of file formats and their codecs, the need for media exchange between diverse systems is clear. Production, post production, archiving, and distribution are like islands, each operating based on its own requirements, standards, and practices. Transcoding bridges the gaps between them, enabling media exchange between systems and making possible the vast range of uses to which digital media are now put.

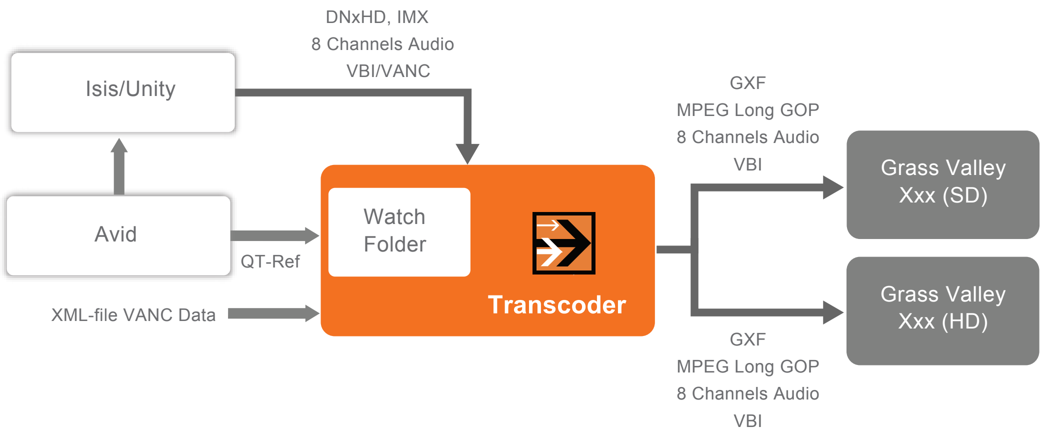
The basic idea of transcoding is quite straightforward: make content from a source in one format playable by a system using a different format. If the desired attributes (e.g. frame size, frame rate, etc.) are the same for both source and destination formats, and the destination container supports the codec used on the source's elementary streams, then it may be possible to "re-wrap" (direct encode) those streams in the destination format. But in all other cases the source streams must each be transcoded, which means decoding to an uncompressed state, still in the digital domain, while nearly-simultaneously re-encoding with a different codec and/or different attributes. The transcoding workflow may also allow streams to be manipulated in other ways while they are in the uncompressed state between decoding and re-encoding. An audio stream, for example, may be adjusted for loudness, or a video stream may be rescaled, color corrected, or letterboxed.

In an ideal world, transcoded material would retain all the media and metadata present in the source. In practice, the extent to which all aspects of the source are effectively preserved may be limited by the capabilities of the destination format. Metadata is particularly problematic in this regard. Content metadata that cannot be mapped from a source attribute to a corresponding destination attribute will be lost because there is nowhere to put it in the destination file.




Transcoding ideally results in the destination fully representing the source (left), but file format incompatibilities mean that some aspects of the source may be left behind (right).

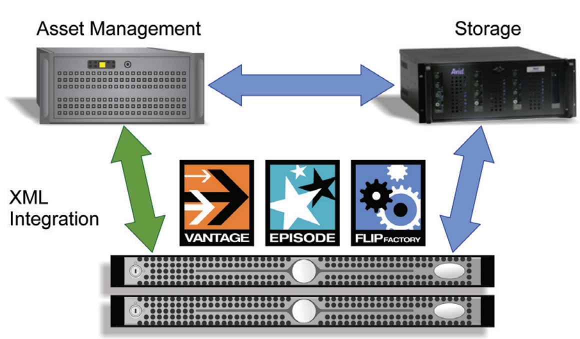
A post-to-broad-cast transcoding workflow in which content from the NLE's SAN is converted for use on the broadcast servers of a station or network.




## Transcoding workflows

Just as the file formats used for digital media vary greatly depending on their purpose, so too do the workflows used in various segments of the media industry. We'll look at a few here to get a sense of how transcoding is implemented in various situations.

One of the simplest configurations is the integration of transcoding into digital asset management (DAM) applications, where the content is stored in a form that is different from the form it will be used in if needed. A news organization, for example, would typically use a DAM system to store footage that is acquired and broadcast live as events unfold. Some of that footage would later be pulled from the DAM system and transcoded from the DAM-compatible format into an NLE-compatible format so that it can be edited into a regularly scheduled news show. The transcoding element of that asset retrieval process should be automated for transparency to the user, which can be achieved either through a watch-folder system or software coding that addresses interfaces supported by the DAM system.

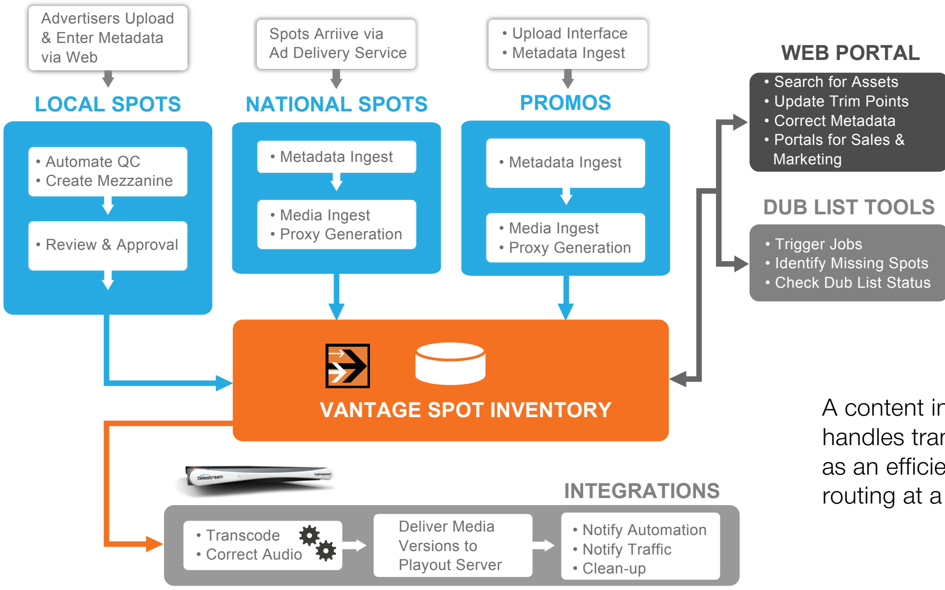



A transcoding system that supports API-level integration can enable assets stored on a DAM system to be transparently readied for editing on an NLE.

A somewhat more complex scenario is post-to-broadcast, where the finished output of an NLE is moved to a broadcast server. The illustration at the top of this page, for example, shows an Avid NLE system with an ISIS or Unity storage area network (SAN). When post production is complete, including compositing and rendering of effects, the SAN will contain a set of finished video segments in the NLE's working format, in this case DNxHD.

The NLE will create a QuickTime reference file that is comprised of a series of timecode references to segments of this DNxHD video on the SAN. The QT-Ref file is output to a watch folder, alerting the transcoding system that there's work to do and pointing it to the source material to transcode. The transcoder then accesses the SAN and transcodes the DNxHD material specified in the QT-ref file to a destination format that is compatible with the broadcast server, in this case to Grass Valley's proprietary GXF format for both an HD feed and an SD feed.

Configuration gets even more complex when designing a workflow for broadcast stations or cable networks, where content comes from multiple sources, both local and remote, and may be either prerecorded or live. For maximum efficiency, such systems must integrate seamlessly with a variety of content sources, including in-house NLEs used for production of promos and tags, delivery networks such as Pathfire or Pitch Blue that provide syndicated content, and digital catch servers that ingest commercials from delivery services such as Extreme Reach, Javelin, and Hula MX.




A content ingestion system that handles transcoding can serve as an efficient hub for content routing at a station or network.

As shown above, a multi-input system can maximize efficiency by serving as the hub through which content from the entire range of sources passes on its way to on-air servers as well as to servers for online distribution.

Another complex scenario – one whose importance has increased dramatically in recent years – is content repurposing for Web or OTT dsitribution. In this situation the content is programming that has been created originally for broadcast (terrestrial or cable) and is being prepared for downloading to computer desktops and mobile devices, for example via the iTunes Store. As shown in the next block diagram, a series of steps are involved, including not only changing the resolution, e.g. from HD (1920 x 1080) to iPhone-compatible widescreen (640 x 360), but also things like stitching the between-ad segments into a single file and perhaps adding an appropriate logo in the lower right corner.

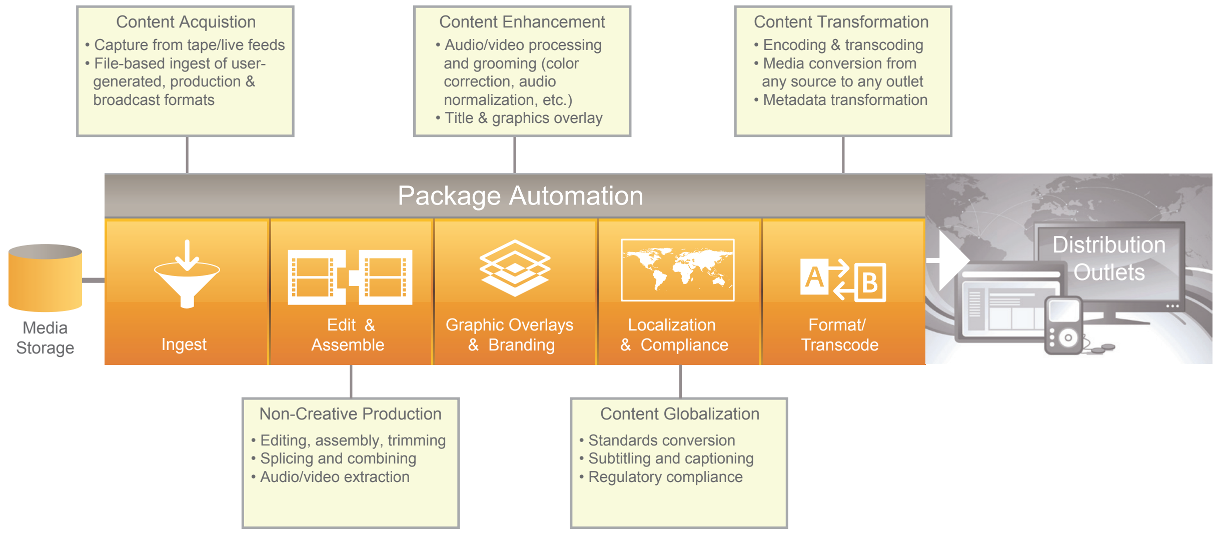
The imperative is to complete all of these steps with as little manual intervention as possible, because human labor is expensive and the revenue from repurposed media is incremental and uncertain. The most effective transcoding system for this application will be one that can also handle all of these related tasks within a single, automated environment. That environment should require minimal operator involvement, intelligently routing files based on their media characteristics or metadata while also incorporating quality checking and analysis.

Regardless of the specifics of the individual situation, the fundamental question facing a designer who is integrating transcoding capabilities is where the transcoders fit into the workflow.

A large centralized transcoding capability can be easier to manage and to generate consistent results, but it also requires a flexible network that can support multiple workflows for different types of tasks. Multiple dedicated transcode systems, on the other hand, may each individually be simpler, but can present more of a maintenance challenge overall. The greater the volume of work being done, the more likely it is that a centralized system will ultimately prove more efficient.

## Conclusion

As we've learned, a fully-realized transcoding solution must take into account each of the major compo-nents – audio streams, video streams, and both playback and content metadata – of the file formats it supports. It must be automatable so that the overall process can execute reliably with little operator intervention.




The ideal new media workflow integrates transcoding into a unified, automation-intensive environment providing multi-step processing

And it must be designed to integrate seamlessly with components handling other tasks (stitching, logo overlays, proxy generation, etc.) that also need to occur at the same point in the workflow, so that content can flow as smoothly as possible from source to destination.

Because codecs and file formats are each tailored for a specific application, their capabilities are inherently distinct, and the features of one don't always port easily to another. As long as these differences persist, there remains the possibility that some aspect of a source will be lost in translation. But it's precisely because of those differences that transcoding is such an indispensable part of today's–and tomorrow's–production and distribution processes. If the past is any guide, the future holds ever-increasing ways in which we create, edit, store, transmit, and use digital media. Transcoding enables the interoperability that makes that future possible.

## About Telestream and Vantage Transcode

Telestream provides world-class live and on-demand digital video tools and workflow solutions that allow consumers and businesses to transform video on the desktop and across the enterprise. Telestream products span the entire digital media lifecycle, including video capture and ingest; live and on-demand encoding and transcoding; playout, delivery, and live streaming; as well as automation and management of the entire workflow.

Media, entertainment, corporations, education, government, content creators, and consumers all rely on Telestream to simplify their digital media workflows. Robust products, reliable customer support, and quick response to evolving needs in complex video environments have gained Telestream a worldwide reputation as the leading provider for automated and transcoding-based digital media workflow solutions.

Telestream Vantage Transcode simplifies file format conversion in today's multi-format, multi-vendor video environments. Video and audio are automatically converted with metadata between all the major SD and HD broadcast server, edit system, streaming server, cable VOD server, web, mobile and handheld file formats. Telestream's complete range of Vantage products take transcoding workflows to the next level – by adding automated decision making and bringing complex process steps together into an automated, easy-to-manage workflow. Vantage enterprise-class system management products provide an even higher level of visibility and control for high-volume video workflows.

## To learn more

Visit us on the web at www.telestream.net
or call Telestream at 1.530.470.1300.

telestream

www.telestream.net | info@telestream.net | tel +1 530 470 1300