File exchange formats for Networked television production

Hans Hoffmann
EBU Technical Department

While many digital TV production systems have been introduced in isolated "islands", there is increasing user demand to incorporate these islands into larger networked infrastructures. As a consequence, television production is currently undergoing a second phase of digitalization – the move towards IT-based technologies.

By way of an introduction to the AAF, GXF and MXF articles published in this issue of EBU Technical Review, this article presents – in a broader context – the file formats and systems issues that are likely to be encountered when migrating to "file-oriented" TV production.

Computer-based technology has already proven its usefulness in many applications within the professional broadcasting environment. Prominent examples can be found worldwide in the area of server-systems for production, post-production, playout and archiving. We are currently witnessing the first attempts to extend the application range of disk-based storage and networking to mobile applications in professional newsgathering as well as to news environments and the television production process.

The common denominator in all these applications is the transport of programme data and the storage of these data on non-linear media within partly-proprietary file formats. Programme exchanges can therefore only be carried out across platforms that can manage and exploit such proprietary file structures. User organizations, such as the EBU, have already expressed a strong requirement to share files between the systems of different vendors, in an non-proprietary way – within the studio as well as between different broadcast facilities. Sharing in this context refers to the exchange of content that is assembled in files, by means of removable media (hard-disk, optical disk, data tape) or, in particular, by directly accessing the content stored in files through standardized interfaces and network protocols. The operational and economical benefits that are obtained, when adopting networks and exchanging content in file form, can briefly be summarized as follows:

- multiple users can simultaneously access data that is related to a common project, within a distributed production environment;
- file exchange does not introduce picture-quality degradation because the compressed video accommodated in the file body can be transferred in its native compressed form;
- file exchange can be carried out through LANs and WANs at different speeds;
- the speed of the file transfer can be adapted to the available channel bandwidth (e.g. if the network provides for 10 Mbit/s, the file is transferred at that speed; if a faster network is available – and if the peripheral equipment can support it – then the transfer can be done at a higher speed);
- users can trade off the transfer costs versus the transfer time;
- metadata, audio, video and data can be transferred in one common wrapper;

- separation of the (storage) media and the content embedded in the file is possible;
- a horizontal system approach, following a layered model, is possible;
- systems can be built using readily-available computer equipment which might add economical benefits to the overall system costs, etc.

Within a distributed multi-user environment, the above-mentioned advantages can only be exploited if the source and destination systems can interoperate, which requires the file format and its payload and metadata to be well defined, standardized and open.

Regrettably, the vast majority of implementations currently offered on the broadcast market employ a variety of different and partly-proprietary file formats. Some of these have been directly adopted from the computer industry – such as the AVI file format – while others have been developed for more demanding applications in the professional broadcast world. It is noteworthy to mention that a very popular file format – the **General eXchange Format** (GXF) which is used in Grass Valley Group's Profile Server – was standardized by the SMPTE a while ago as SMPTE 360M [1]. In spite of this standardization (which was very important for the users and industry), it was recognized that more demanding requirements in the metadata area may not be met by this file format.

The need for a common, standardized and open file format – designed to cope with a broad range of requirements encountered within the wide gamut of television programme production – has become painfully obvious. Two good reasons for this are:

- the existence of different incompatible file formats, or file formats with limited functionalities, with most of them not complying with the needs of the professional broadcast industry (e.g. limited file size, no editing capabilities within a file, and no payload neutrality or metadata or KLV conformance etc.);
- the enthusiastic introduction of server-based NLE systems in most broadcast installations around the world.

Recognizing this strong demand for standardization, several bodies such as the Pro-MPEG Forum, the AAF Association, the EU GFORS Project and the EBU started an initiative about three years ago to develop a common file format called MXF (Material eXchange Format) for the exchange of TV programmes in production environments [2]. The SMPTE, and its technology committee for Wrappers and Metadata (W25), was chosen as the platform to carry out the required standardization.

From the user community, EBU project group P/PITV played a key role in defining the User Requirements (URs) and harmonizing the different industry attitudes within this initiative. Apart from other example implementations of this new file format (such as the Software Development Kit from the GFORS project), the EBU provided the platform for the development of an Open Source Code implementation of the MXF file format. The reason for the EBU engagement in a software project such as this (with strong support from the IRT in Munich and INESC Porto) was the necessity to provide feedback to the (paper) standardization work, to verify the URs in practical implementations, to provide source code to the users and manufacturers for rapid market penetration, and to permit users to become familiar with this new technology. The source code itself is freely available on a CVS from the EBU [contact: hoffmann@ebu.ch].

Transport and storage of files

A file can be considered as a "Container" or "Wrapper" for all the programme elements (digital audio, digital video, additional data and metadata) within a specific project. These elements can then be transferred over a network or stored on a storage medium as a single entity, which can easily be identified by a unique file name for unambiguous retrieval. The metadata provides a description of the essence accommodated in the "file body" and defines the exact relationship (contextual, timing etc.) of the various elements with one another. This allows audio and video to be processed separately, or the essence or the whole file to be linked to some additional metadata located on a database. A prerequisite for this functionality is that the structure of a file format must provide information which enables successful de-multiplexing (e.g. separation of the audio from the video), and identification and processing of the metadata and essence contained in the file. It needs to be

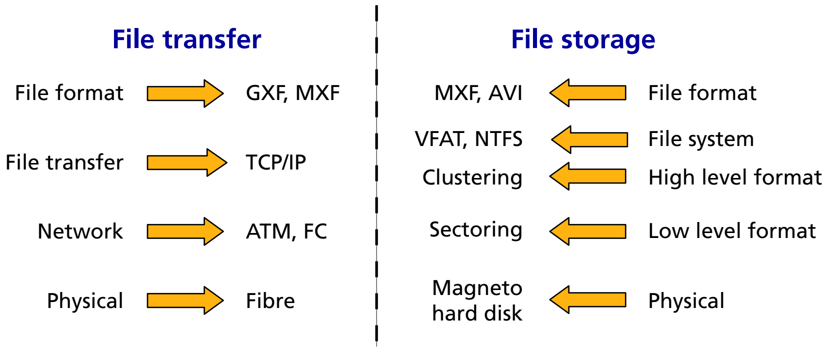

Figure 1
File transfer and file storage – the different layers of interoperability

stressed however that this information may not be sufficient to allow a successful interpretation or processing of the essence carried within the payload of the file (for example, the ability to decode compressed video with a certain picture quality).

It needs to be understood that the **file transfer** mechanism is independent of the **file format** and the payload located in the file. The definition of a file format is – in principle – independent of the transport mechanism used (layer model). However, the often-mentioned requirement to accommodate partial file transfers does require a degree of inter-relationship between the file format and the transport mechanism.

A similar layering of the functions can also be found when investigating the technologies involved for storing files. *Fig. 1* visualises the different layers to be considered for file transfer (left) and file storage (right).

If a real-time video signal is transferred onto a hard-disk-based server (e.g. for NLE), the incoming data stream is stored as a file. In the case of file transfers between servers over networks, the incoming signal is usually already in file form and may either be directly transferred to the storage medium, or it may need conversion to a different "file format" before storage. The requirement for high data throughput, fast non-linear access to the stored content as well as for the efficient usage of the storage medium, means that the file format may need to be restructured to achieve a best match between the inherent file structure and the structure of the segmented format of the storage medium. The latter is called **structured storage** or **low-level storage** format.

The main elements, which have to be considered when discussing files, are:

- The format used to transfer the information as a file – and this might exist only "on the wire". It can be different from the file format used to store the information.
- The low-/high-level storage format or structured storage which contains the file format.
- The file transfer protocols applied.
- The Application Programming Interface (API) and Operating System which are responsible for generating the access to the file stored on the media.

In professional TV production, even when using fully IT-based infrastructures, discussions on the constraints that are imposed when moving files between systems have not yet ended. In particular, the application of "streaming files" with real-time capabilities generates a challenge for typical IT networks, when considering the special requirements of TV production.

Streaming of a file

In the streaming mode, content is transferred in such a way that a certain timing relationship to a clock is maintained. This allows immediate display of the content (synchronous, isochronous transmission). Moreover, the transport system used must comply with Quality of Service (QoS). parameters These define the tolerances for bit-rate, delay, jitter/wander and bit error rate (BER). The network topology applied is point-to-point and point-to-multipoint (broadcast) with (usually) a unidirectional data transfer. Different methods with different technical performance are used today to achieve the required QoS parameters. The most popular are UDP or RSVP on IP networks, or a direct mapping of the file into the transport without any additional flow control protocols (e.g. direct mapping into ATM or FC).

With respect to the "streaming of a file" – which means to transfer and directly play out a file – the crucial prerequisites can be summarized as follows:

- essence must be arranged in the file body in a direct playable order;

- resynchronization information is distributed over the file to permit re-lock after interruption;
- the transport system, such as networks or unidirectional links, have to provide certain QoS parameters;
- depending on the network used, the file has to be transferred slightly faster than real-time to compensate for the terminal buffer delays, etc.;
- sufficient metadata information should be available to understand the payload to be played.

File transfer of a file

File transfer provides a reliable transport of the information with guaranteed delivery, even under adverse conditions. Often the term "generating a clone" or "generating a bit-by-bit copy" is also used in the discussions to emphasize that no difference between the sender and destination files is permitted. This can be achieved to a limited degree by either Forward Error Correction (FEC) or by using flow control protocols (TCP/IP) over bi-directional links that initiate a retransmission of corrupted packets, if necessary. The topologies applied include point-to-point and point-to-multipoint (reliable) transfer. Time-critical applications, in which a file has to arrive at the destination at a predetermined point in time, will also require to meet certain QoS parameters concerning bandwidth and bit-rate control. The latter is required if many users need to share the bandwidth on a network, in order to avoid the full consumption, by a single user, of the available network bandwidth.

The transfer time of a file is normally determined by the delay experienced when transiting the network and, in particular, by the flow-control protocols (e.g. TCP/IP) and the delays found in the source and destination servers (buffer memories, DMA transfers, disk access, etc.). Simple solutions to avoid blocking on the network include the use of protocols (XTP and FTPplus) which permit an adjustment for maximum bit-rate per user; more sophisticated solutions offer QoS parameters at the network level (ATM, FC).

In applications requiring faster-than-real-time transfer, the network must provide adequate bandwidth, and bandwidth-control in addition.

Abbreviations

AAF	Advanced Authoring Format
API	Application Programming Interface
ATM	Asynchronous Transfer Mode
AVI	(Microsoft) Audio-Video Interleaved
BER	Bit-Error Ratio
CVS	Concurrent Versions System
DMA	Direct Memory Access
EPG	Electronic Programme Guide
FC	Fibre Channel
FEC	Forward Error Correction
GXF	General eXchange Format
I/O	Input/Output
IP	Internet Protocol
ITU	International Telecommunication Union
KLV	(SMPTE) Key Length Value
LAN	Local Area Network
MPEG	Moving Picture Experts Group
MXF	Material eXchange Format
NLE	Non-Linear Editing
NTFS	(Microsoft) Windows NT File System
OS	Operating System
QoS	Quality of Service
RSVP	ReSource reserVation Protocol
SDTV	Standard-Definition Television
SMPTE	Society of Motion Picture and Television Engineers (USA)
TCP/IP	Transmission Control Protocol / Internet Protocol
TVA	TV-Anytime
UDP	User Datagram Protocol
UMID	(SMPTE) Unique Material Identifier
UR	User Requirements
VFAT	(Microsoft) Virtual File Allocation Table
WAN	Wide-Area Network
XML	Extensible Markup Language
XTP	Xpress Transport Protocol

User requirements for the file formats used in TV production

It has been mentioned earlier that the EBU was cooperating in the project to define a common file format (MXF), in particular on the definition of the User Requirements. The bulleted list below shows only those URs voted with the highest priority during this project. It is noteworthy to mention that the full list of URs may also become part of the engineering guideline for the MXF standard:

- easy to understand, apply and standardize;
- compression independent;
- low implementation overhead;
- must be open (as per the ITU definition);
- must provide identification of the payload;
- must be extensible in its header and body (KLV coding);
- must offer scalability (one frame ... many frames);
- must provide synchronization of multiple essence types;
- must be able to wrap video, audio, data essence and metadata;
- must permit direct mapping of the existing transfer formats;
- must uniquely identify the container framework;
- must be usable on all major platforms/OSs;
- must be application independent;
- must provide the means for partial file transfer;
- must provide the means for graceful recovery after interruption (although this is a transport issue as well);
- must provide cut-only editing capability (versioning);
- must be transport and storage-mechanism independent;
- must offer format expandability in an operational pattern (along a certain profile of operation);
- must offer easy conversion between files and streams.

File formats and metadata

In the broadcast community, the description and classification of metadata, together with the use of a defined coding scheme (KLV) including UMIDs, has been recently agreed and standardized by the SMPTE. Metadata can therefore be applied to define and describe the essential functions of a file format, as well as the payload of a file.

The (popular) diagram below *(Fig. 2)* shows how each item (essence or metadata) within the file is associated with a UMID when it is generated, transferred over a network, or stored on a hard disk or data tape.

It has been often mentioned in the discussions that the success of metadata in professional TV production will go hand-in-hand with the successful standardization of a common file format that supports the relevant metadata standards. On the other hand, there will also be a broad range of applications which store the metadata in databases where the file format will be a typical IT format such as XML. However, there will also be a number of applications in the TV production chain which require that:

a) some metadata are directly associated to the essence – this means that they are part of the file;

b) a link, from the file carrying the essence to an external storage, is provided and contains the metadata.

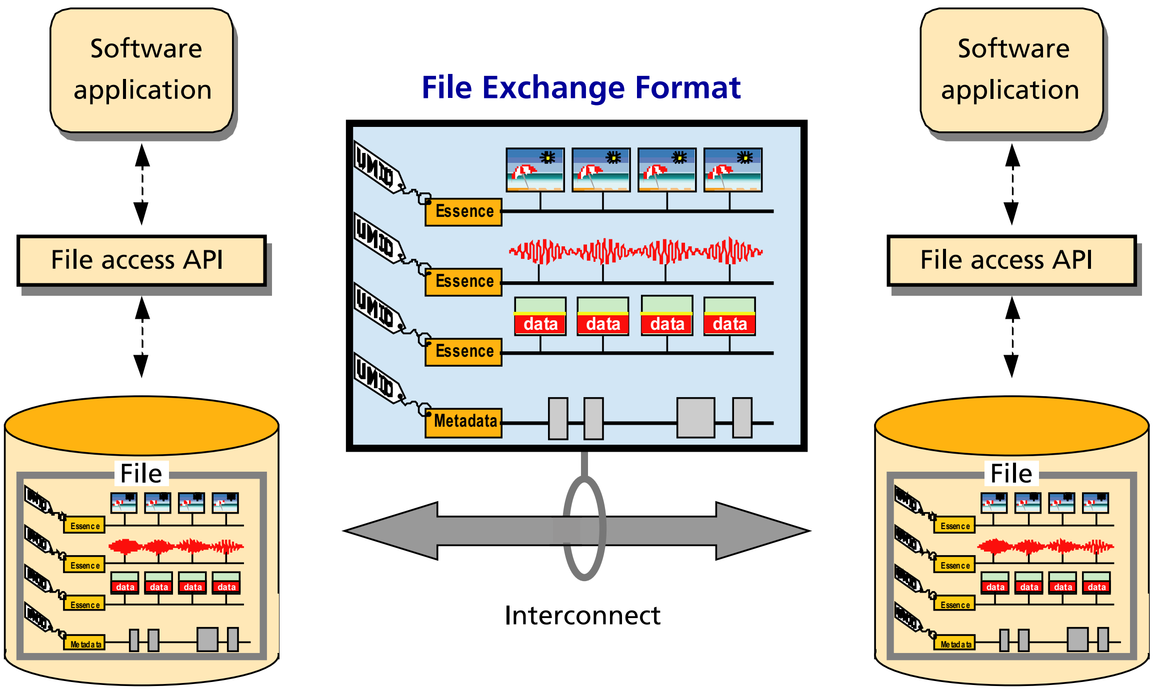

Figure 2
File transfer and storage – the different layers required

In particular, case b) above requires that the link from the file follows the UMID standards and is handled in a reliable way, especially by maintaining the appropriate source and destination information if either the file with the essence (e.g. MXF) or the metadata file (e.g. XML) is moved.

Structural metadata and user metadata

It has been described already that the proper acceptance of a file format used in professional broadcasting will have to enclose different complexity levels – depending on its intended application range. A file format to be used in post-production, such as AAF [3], will have to provide rich functionalities for picture and audio manipulation; a file format for TV-production purposes will require less complex functionalities (i.e. only simple edit cuts) but may include more production-oriented and real-time functions. These functional capabilities of a file format are described in certain metadata and are called **structural metadata**.

The other metadata which are added to the file, but which are not required by the function of the file itself, are called **user metadata**. User metadata add value to the file; examples are the name of the author, production location, name of participants, rights, budget information, shooting script, etc.

The MXF file format that is currently under standardization in the SMPTE supports the so-called **User Metadata Plug-In** interface. This means that the file format is able to transfer all types of user metadata which follow the rules described in the proposed MXF standard.

In general, it is important that the User Requirements for transparent metadata transfers are met, and that file formats treat metadata as “just another type of data” to be carried in the file.

Some system aspects

It becomes obvious that metadata, UMIDs, the metadata dictionaries and other standards which are available or under standardization can only be applied to the system if certain rules are defined. This directly affects the file format which has been chosen, because it provides the functionalities required for transferring essence, essence plus metadata, UMIDs, or essence plus a link to metadata stored at a different location (of course, there will also be IT file formats for metadata-only use).

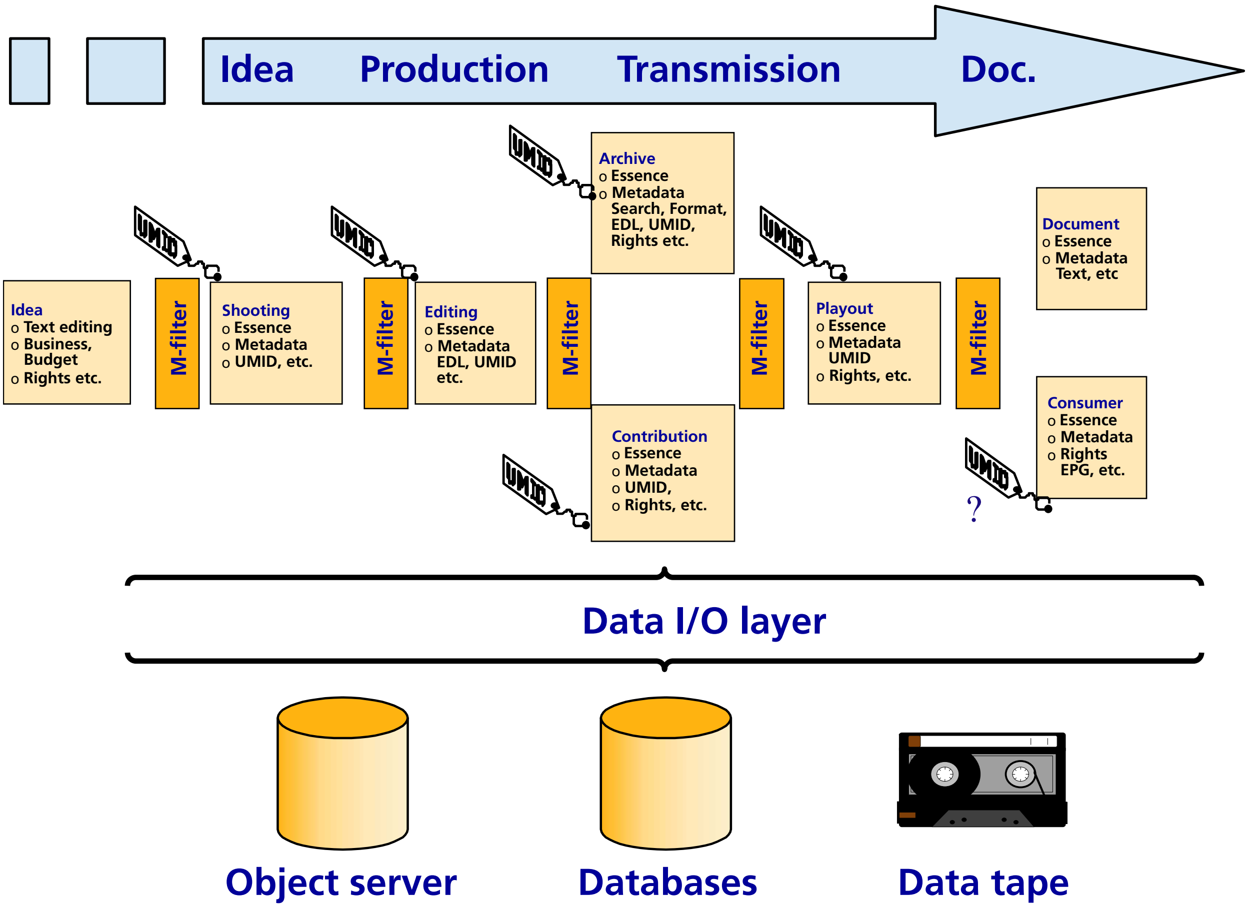

Figure 3
Simplified example of data flow through a system

Fig. 3 shows a simplified example for the flow of data through a system.

Depending on the functionality which is required at a certain position in the workflow, different metadata will be required or will have to be available (access rights) to the user (e.g. an editor will certainly not be interested in budget information). This function will have to be provided by a future metadata management system in conjunction with "metadata-filters" which will represent a serious challenge for system developers. Some of the rules for metadata and UMID handling can be summarized as follows:

- certain applications need certain metadata;
- the UMID is associated with each essence and renewed when essence is changed/created;
- metadata and UMID management will be a dynamic process through the whole data chain;
- filters will have to regulate this;
- infrastructures such as file formats will have to support the presence of metadata and UMIDs;
- storage will have to handle essence, UMIDs and metadata in either legacy equipment, object servers or databases;
- rules for "what goes into the archive" are required;
- rules for metadata exchange via internal and external lines are needed (EBU P/META and SMPTE standards);
- rules on what will be delivered to the consumer need to be defined (e.g. TV-Anytime).

A further "system issue" will be the coexistence of different – hopefully, a limited number of – file formats in the future TV production process. Investigations have clearly shown that one single file format such as GXF, AAF or the newly-developed MXF will not be able to meet the requirements of all the applications in a broadcast environment. For that reason, during the design of new file formats (such as MXF), it is important to consider not only extensibility but also the means of how to manage a conversion from one file format to another, with as little processing as possible. The MXF development, for example, tried to meet this requirement by including functions for an easy conversion between AAF and itself.

Hans Hoffmann was born in 1964 in Munich. After studying telecommunication engineering in Munich, he joined the IRT in 1993. At first he worked on research and development projects concerning the ITU-R BT.601 standard and the serial digital interface. With the introduction of compression, IT, server systems etc. in TV production, his work then moved towards these new technologies.

Mr Hoffmann chaired the EBU group P/BRRTV and P/PITV, which were both involved in standardization activities such as SDTI and file formats, and actively worked in the SMPTE engineering committees. He participated in the work of the joint EBU/SMPTE Task Force and chaired the subgroup on Networks and File Management. In 1998, he became chairman of the SMPTE technology committee on Networks and File Management. He is also Special Rapporteur on File Formats in the ITU-R.

In March 2000, Hans Hoffmann joined the EBU headquarters in Geneva as a senior engineer in the Technical Department. Here, he works in the area of new technologies for TV programme production, archiving, metadata and related issues. In particular, he is concerned with the strategic user questions which arise with the introduction of new technologies based on Information Technology. At the moment he serves in EBU project groups P/FTP, P/PITV, P/SPX and P/DTR.

Mr Hoffmann is a Fellow of the SMPTE and became SMPTE Engineering Director for Television in March 2002, but retains his post in the EBU.

Conclusions

With IT-based technologies being used increasingly within the broadcast industry, the use of file formats for programme exchange and storage will be mandatory for all aspects of future digital TV production (including graphics, post-production, archiving, contribution and distribution). This will apply not only to conventional SDTV production, but all the way up the quality scale to digital cinema theatres that are linked via new distribution paths. Towards this end, post-production houses will hopefully adopt the AAF file format (although not yet standardized) for complex off-line operations.

The development of the MXF file format – which provides the required interoperability (at least in the file format layer) – is another important step in the right direction.

However, more complex studies are still required in the following areas:

- user-defined metadata schemes in TV production;
- UMIDs and rules for the data chain;
- management systems (also for metadata and UMIDs);
- APIs for system interoperability (middleware);
- fast file transfer protocols;
- gateways/conversion between existing file formats and future formats (e.g. between GXF and MXF).

These "system issues" will have to be addressed, and the problems solved, if we want to fully exploit the benefits promised by adopting IT (and server-based) technologies in TV production.

Bibliography

[1] Bob Edge: **GXF – the General eXchange Format**
EBU Technical Review No. 291, July 2002.

[2] Bruce Devlin: **MXF – the Material eXchange Format**
EBU Technical Review No. 291, July 2002.

[3] Brad Gilmer: **AAF – the Advanced Authoring Format**
EBU Technical Review No. 291, July 2002.