

# Universal Preservation Format

## Part 2: Technical Requirements

Dave MacCarn  
dave\_maccarn@wgbh.org  
WGBH Educational Foundation  
last revised March 22, 1999

1. Introduction
2. Analog Component
3. Digital Component
  - 3.1 Wrapper
  - 3.2 Essence
  - 3.3 Metadata
  - 3.4 Other Object Types
4. Open Standards
5. Media Compiler
6. Other Options and Features

### 1. Introduction

The Universal Preservation Format (UPF) is a storage system for electronically generated content. UPF requires a storage technology that is "self-describing." A "self-describing" storage technology is one where the structure of the storage need not be known. The structure is disclosed internally in the storage. Access to the storage system is described. Once access is obtained, all information for the retrieval of stored content is described as well as the structure of the content. This "self-description" protects against technological obsolescence.

Note: There are two main format concepts used here. First, the physical storage format or media (e.g. a hard disk mechanism or data tape,) and second, the content storage technology (i.e. the file format in which the content is stored.) Using the example of a video tape system, these two formats are bound together within the machine. The content is encoded in some byte order or format (e.g. samples per line, bytes per sample, compressed or uncompressed.) This encoding is stored on a physical medium in a specialized order (decollated blocks of video or audio bytes.) UPF removes this bond by abstracting the encoded content (i.e. setting it free) from the recording system.

The "self-describing" storage technology uses a Wrapper concept (see below) to hold the content (defined as Metadata plus Essence.) Physically connected to the content is a description, in analog form (i.e. human readable blueprint,) of the physical format housing the storage format. This description describes the construction of a "reader" for the physical format, thereby insuring recovery of the content, even in the absence of the original recording mechanism (i.e. physical storage devise.) This requirement is key to UPF and defines UPF as a hybrid technology, one with an analog component and a digital component.

Note: It is possible to have a subset of the UPF, one that doesn't contain the analog component. This subset is one in which the content is stored in a Wrapper on a known physical format (e.g. DLT tape.) This form would be subject to changes in the technology of the physical format (there will always be another tape or disk format). But as long as the physical format is readable (the specification of the drive mechanism, byte order, etc. is known), the content will be recoverable. This allows for an assumed less expensive storage system but with all the problems of migration. However, this form provides for a common content storage format, allowing other content formats to be merged.

## 2. Analog Component

The analog component of UPF would be a human readable system that describes the content stored on the physical media. It also contains the information (i.e. text, pictures, specifications) to construct a system to read the physical media in order to retrieve the content. Because the content is stored within a storage structure, the blueprints also must explain how to access the digital storage structure that contains the content. The analog system must be attached to the media to prevent loss of this information.

For example, a system like microfiche would store instructions that can be read simply with the aid of magnification. These instructions would first explain what is contained on the physical media (e.g. the title, producer, etc. of a video program) and that it is stored in a digital form in a storage structure. Then the instructions explain how to read the digital media and would describe such things as the recording mechanism (e.g. DVD-ROM, phase-change optical, DLT etc.) and include full manufacturer specifications for elements such as, transport system, heads, byte coding, etc. Additional information needed to access the content within the retrieved storage structure is also available. The combination of the analog component and the digital storage structure could be stored on a single piece of media that uses one side for the storage of the analog information and the other side for the storage of the digital information. Norsam Technologies, Inc<sup>1</sup> is developing a technology for such a combine storage system.

## 3. Digital Component

### 3.1 Wrapper

The digital component of UPF would be a content storage technology (also known as a storage structure or Wrapper) that contains information about the content and which is stored as objects. The information in the Wrapper describes not only the location of content within the Wrapper but also all information about the content (i.e. it's type.) The Wrapper could be defined as a "self-describing" database. This Wrapper can contain any type of digital object (Essence or Metadata, e.g. text, still or moving images, sound, etc.) "The fundamental purposes of a Wrapper are (i) to gather together programme materials and related information (both by inclusion and by reference to material stored elsewhere) and (ii) to identify the pieces of information and thus facilitate the placing of information into the Wrapper, the retrieval of information from the Wrapper, and the management of transactions involving the information." <sup>2</sup>

The Wrapper would be capable of describing and defining the content and its structure. This would include the type of the object and all information needed to reconstruct the object. (This information could be referred to as another type of Metadata.)

An example of an object is a Rich Text Format (RTF) file. The structure (algorithm) of a RTF file is defined as:

*An RTF file has the following syntax:*

*<File>*

*{ <header> <document> }*

*This syntax is the standard RTF syntax; any RTF reader must be able to correctly interpret RTF written to this syntax. It is worth mentioning again that RTF readers do not have to use all control words, but they must be able to harmlessly ignore unknown (or unused) control*

words, and they must correctly skip over destinations marked with the \\* control symbol. There may, however, be RTF writers that generate RTF that does not conform to this syntax, and as such, RTF readers should be robust enough to handle some minor variations. Nonetheless, if an RTF writer generates RTF conforming to this specification, then any correct RTF reader should be able to interpret it.

### **Header**

The header has the following syntax:

```
<header>  
\rtf <charset> \deff? <fonttbl> <filetbl>? <colortbl>? <stylesheet>? <revtbl>?
```

### **RTF Version**

An entire RTF file is considered a group and must be enclosed in braces. The control word \rtfN must follow the opening brace. The numeric parameter N identifies the major version of the RTF Specification used. The RTF standard described in this Application Note, although titled as version 1.4, continues to correspond syntactically to RTF Specification Version 1. Therefore, the numeric parameter N for the \rtf control word should still be emitted as 1.

### **Character Set**

After specifying the RTF version, you must declare the character set used in this document. The control word for the character set must precede any plain text or any table control words. The RTF Specification currently supports the following character sets.

#### **Control word**

#### **Character set**

\ansi

ANSI (the default)

\mac

Apple Macintosh

\pc

IBM PC code page 437

\pca

IBM PC code page 850, used by IBM Personal System/2Æ (not implemented in version 1 of Microsoft Word for OS/2)

### **Font Table**

The \fonttbl control word introduces the font table group. Unique \fn control words define each font available in the document, and are used to reference that font throughout the document. This group has the following syntax:

...

Or the another example of an object is an ITU-R 601, 4:2:2 format video. ITU-R 601 is defined as:

*ITU-R BT 601 4:2:2 format:*

*Video Signal*

*3 components*

*Luminance Y*

*Chrominance Cb, Cr*

*differentiation for 25 Hz countries and 30Hz countries.*

*25 frames per second (30 frames per second), that is 50 fields/s (60 fields/s)*

*625 lines/frame (525 lines/frame)*

*for luminance: 864 samples/line (858 samples/line), that means for both cases a sampling frequency of 13.5 MHz.*

*for chrominance: 432 samples/line (429 samples/line), that means for both cases a sampling frequency of 6.75 MHz.*

*sampling structure: orthogonal.*

*sample coding: uniform PCM either 8 bits or 10 bits.*

*quantization levels (for 8 bits samples) and analogical signal:*

*0 and 255 only for sync.*

*from 1 to 254 for video.*

*luminance: 16 = black, 235 = white*

*chrominance: 128 = no chrominance*

...

(this includes byte order, sample rate, bits per sample, etc.)

The Wrapper could also contain single or multiple objects (Essence or Metadata) of varying size. The Wrapper may point to other Wrappers, thus permitting content to be spaced over multiple storage systems. Metadata in each Wrapper would track and index the content.

[For information on the internal structure (i.e. storage technology) of Wrappers see: Apple's Bento<sup>3</sup>, IronDoc<sup>4</sup> and Microsoft's Component Object Model.<sup>5</sup>]

### **3.2 Essence**

Essence is any electronically generated data that represents images, sound and text, etc. Essence types could be video, audio and data of many kinds including graphic, still image, captions, text and other data that might be needed by other stored Essence. Certain types of Essence may be treated as Metadata, such as captions, which might be included visually in a program and be used as an index to a program.

Essence may be compressed or non-compressed as dictated by the original source. It is important to note that Essence is to be stored in its native format. The structure of the Essence depends upon its encoding scheme (e.g. a TIFF file.) This scheme would be identified and defined in the Metadata.

Essence would be available by sequential or random access depending on the physical storage mechanism.

### **3.3 Metadata**

Metadata is generally defined as data about data. Metadata would be a stored object type. This object type would have other Metadata describing its type and structure. For example, Metadata might be a MARC record<sup>6</sup>. A Metadata object would describe a MARC record and defined it as:

*USMARC Concise Bibliographic: Leader and Directory  
LEADER*

*A fixed field that comprises the first 24 character positions (00-23) of each record and provides information for the processing of the record.*

*Character Positions*

*00-04 - Logical record length*

*The computer-generated, five-character numeric string that specifies the length of the entire record. The number is right justified and each unused position contains a zero.*

*05 - Record status*

*A one-character code that indicates the relation of the record to a file.*

*a - Increase in encoding level*

*The Encoding level (Leader/17) of the record has been changed to a higher encoding level.*

*c - Corrected or revised*

*A change other than in the Encoding level code has been made to the record.*

*d - Deleted*

*n - New*

*p - Increase in encoding level from prepublication*

*The cataloging level of a prepublication record has changed because of the availability of the published item.*

**06 - Type of record**

*A one-character code that indicates the characteristics of and defines the components of the record.*

*a - Language material*

*This code is also used for microforms of language material.*

*c - Printed music*

*This code is also used for microforms of printed music.*

*d - Manuscript music*

*This code is also used for microforms of manuscript music.*

*e - Printed map*

*This code is also used for microforms of printed maps.*

*f - Manuscript map*

*This code is also used for microforms of manuscript maps.*

*g - Projected medium*

*The described item is a motion picture, videorecording, filmstrip, slide, transparency, or material specifically designed for overhead projection.*

*i - Nonmusical sound recording*

*j - Musical sound recording*

*k - Two-dimensional nonprojectable graphic*

*This code is used for items such as activity cards, charts, collages, computer graphics, drawings, duplication masters, flash cards, paintings, photonegatives, photoprints, pictures, postcards, posters, prints, spirit masters, study prints, technical drawings, transparency masters, photomechanical reproductions, and reproductions of any of these.*

*m - Computer file*

...

Any number of Metadata types would be accommodated. Some types might be keywords or databases, but all are required to be identified and defined.

One essential Metadata type would be a Unique Material Identifier. Unique Material Identifiers should subscribe to a single standard format and be established in a single registry. This Metadata type would also be classed as permanent. If the Essence is modified through versioning, the Unique Material Identifier would reflect the versioning.

[For a discussion of Metadata types and characteristics see reference 1.]

Certain types of Metadata might be duplicated elsewhere, e.g. indexes, keywords and other information that is stored in a database external to UPF, which refers to Essence stored in the UPF system.

### **3.4 Other Object Types.**

Other types of stored objects would include Application Programming Interfaces (API) to allow for the creation of any number of content management systems that would have full access to the stored content in the UPF.

#### **4. Open Standards**

The UPF would be an open standard. By its nature UPF requires a complete “self-described” system, the technical specifications of the system would be required to be in the public domain.

#### **5. Media Compiler**

A device that might be called a “media compiler” would control the UPF. This media compiler would handle the assembling of the Essence and Metadata objects and prepare them for storage in the UPF. This preparation is required because the sources (i.e. databases, media libraries, tape machine etc.) of the objects may be varied. User interaction may be needed to select the location of the objects (e.g. they could exist somewhere on a network.)

It is desirable to make the compilation process as automated as possible. This automation would use a standard open protocol for interconnection to other automation systems, be they databases or other storage systems.

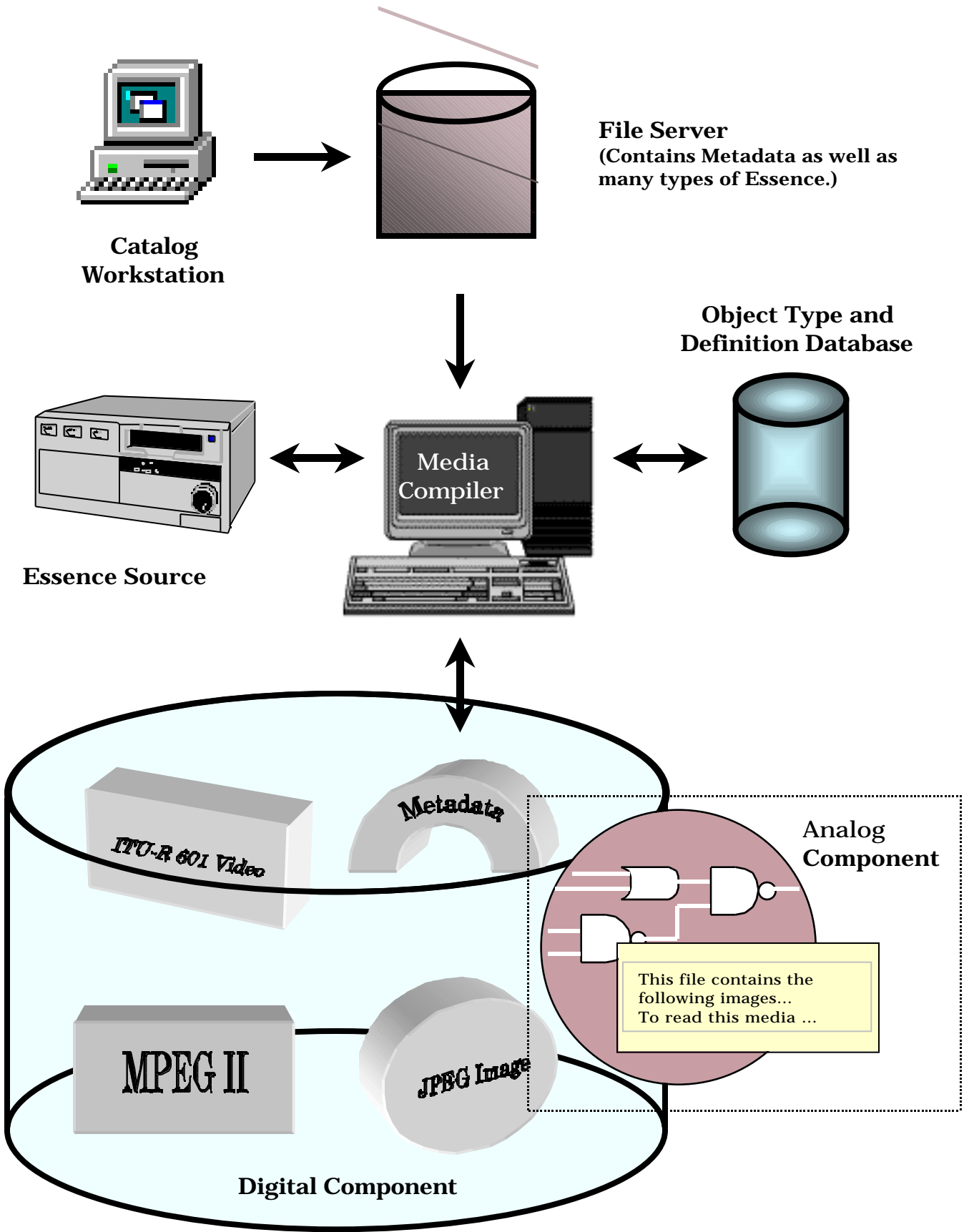
Object typing would be handled automatically where possible. The media compiler would contain a database of known object types and, as much as possible, be able to recognize the supplied objects and write the required Metadata to define and describe the objects.

The media compiler would also handle the retrieval of Essence and Metadata. Standard interfaces would be provided to export the retrieved materials to the requesting systems, be they a database for Metadata or image playback systems for Essence. In some cases it will be necessary to convert objects to contemporary or future systems. For example, videotape recorded in ITU-R 601 format and stored in the UPF might be required to playback to a D-7 videotape machine or to some as yet defined format of the future. Another example is the recovery of a bit-mapped image (BMP) and converted for a JPEG viewer.

#### **6. Other Options and Features**

Manufacturers of media compilers could define other options and features for their UPF storage systems to differentiate their products. However, the UPF could not be made proprietary.

# Media Compiler and UPF Storage System



---

<sup>1</sup> Norsam Technologies, Inc.: see <http://www.norsam.com/>

<sup>2</sup> SMPTE and EBU, Task Force for Harmonized Standards for Exchange of Program Material as Bitstreams, Final Report: Analyses and Results, July 1998. see [http://www.ebu.ch/pmc\\_es\\_tf.html](http://www.ebu.ch/pmc_es_tf.html)

<sup>3</sup> Apple's Bento: see [http://info.wgbh.org/upf/bento\\_design\\_overview.html](http://info.wgbh.org/upf/bento_design_overview.html)

<sup>4</sup> Iron Doc: see <http://www.best.com/~mccusker/irondoc/irondoc.htm>

<sup>5</sup> Microsoft's Component Object Model: see [http://www.microsoft.com/wpaper/Com\\_modl.asp](http://www.microsoft.com/wpaper/Com_modl.asp)

<sup>6</sup> MARC Standards: see <http://lcweb.loc.gov/marc/marc.html>