Public Record Office Victoria
Advice to Victorian Agencies
June 2006, Version 2.1

Advice 13

PROS 99/007 has been replaced by *PROS 19/05 Create, Capture and Control Standard*.

Public offices that have implemented and configured a system in accordance with PROS 99/007 requirements — namely VERS Version 2 VEO creation — can continue to refer to the standard and its associated specifications and advices for the life of the system.

Public Record Office Victoria will continue to:

- Accept digital record transfers in VERS Version 2 VEO format.
- Test current vendor products against the PROS 99/007 requirements up until 30 June 2021. (After this time, PROV will only test vendor VEO creation validity for VERS Version 3 VEOs against PROS 19/05 requirements).

Vendors may continue to self-certify versions of their current products against PROS 99/007 up until 30 June 2025.

Last updated: 25 March 2020

Public Record Office Victoria
Advice to Victorian Agencies
June 2006, Version 2.1

Advice 13

Advice on VERS Long Term Preservation Formats PROS 99/007 (Version 2) Specification 4

State Government Victoria
Department for Victorian Communities

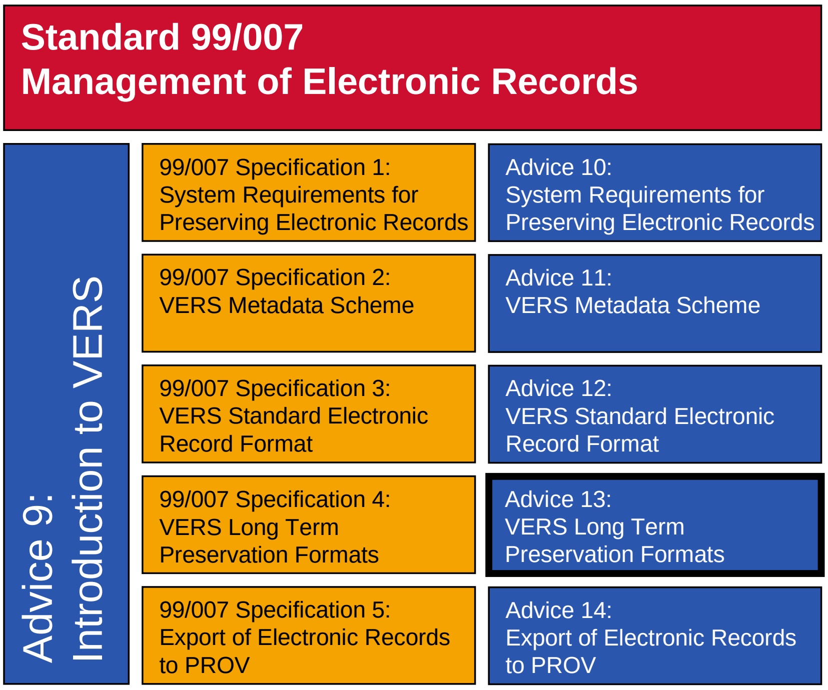
Further copies of this document can be obtained from the PROV Web site http://www.prov.vic.gov.au/

The State of Victoria gives no warranty that the information in this version is correct or complete, error free or contains no omissions. The State of Victoria shall not be liable for any loss howsoever caused whether due to negligence or otherwise arising from the use of this Advice.

Version	Version Date	Details
2.0	31 Jul 03	Released
2.1	30 Jun 06	Added PDF/A, JPEG, JPEG 2000, and MPEG

The Victorian Electronic Records Strategy (VERS)

This document is a guide to *PROS 99/007 Specification 4: VERS Long Term Preservation Formats*. The relationship between the VERS Standard, the Specifications that support this Standard, and the Introduction and Advices that explain VERS is shown below.



Standard 99/007
Management of Electronic Records

Advice 9: Introduction to VERS

99/007 Specification 1: System Requirements for Preserving Electronic Records

Advice 10: System Requirements for Preserving Electronic Records

99/007 Specification 2: VERS Metadata Scheme

Advice 11: VERS Metadata Scheme

99/007 Specification 3: VERS Standard Electronic Record Format

Advice 12: VERS Standard Electronic Record Format

99/007 Specification 4: VERS Long Term Preservation Formats

Advice 13: VERS Long Term Preservation Formats

99/007 Specification 5: Export of Electronic Records to PROV

Advice 14: Export of Electronic Records to PROV

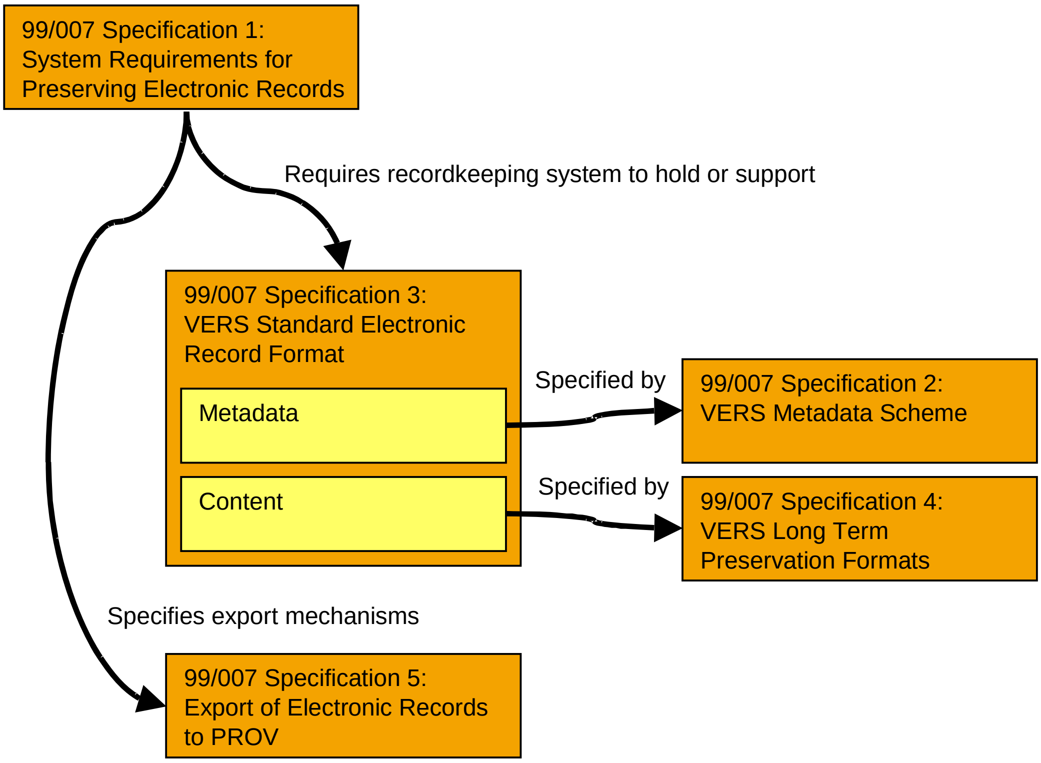
These documents have the following purposes:

- *Management of Electronic Records*. This document is the Standard itself and is primarily concerned with conformance. The technical requirements of the Standard are contained in five Specifications.
- *Introduction to VERS.* This document provides background information on the goals and the VERS approach to preservation. Nothing in this document imposes any requirements on agencies.
- *Specifications.* These five documents provide the technical requirements that support the Standard. Agencies *must* conform to the mandatory requirements of the specifications, *must* conform to the conditional requirements of the specifications if the appropriate conditions are satisfied, and *may* conform to the optional requirements. Some optional requirements are strongly recommended and these are noted as such.

  The five Specifications are:
  - *Specification 1: System Requirements for Preserving Electronic Records.* This document specifies the overall functions that a recordkeeping system must perform to preserve electronic records for a substantial period.
  - *Specification 2: VERS Metadata Scheme.* This document specifies the metadata that a recordkeeping system must hold to conform to VERS.
  - *Specification 3: VERS Standard Electronic Record Format*. This document contains the technical definition of the VERS Encapsulated Object (VEO) format; the mandatory long-term format for records.

  - *Specification 4: VERS Long Term Preservation Formats.* This document lists the data formats that PROV accepts as suitable for representing documents for a significant period.
  - *Specification 5: Export of Electronic Records to PROV.* This document lists the approved media and mechanisms by which PROV will accept an export of electronic records.
- *Advices.* These six documents provide background information, explanatory material, and examples in support of the Standard and associated Specifications. None of the information in the Advices imposes any requirement on agencies.

Relationship between Specifications. A second view of the relationship between the five Specifications is shown in the following diagram:


99/007 Specification 1: System Requirements for Preserving Electronic Records
Requires recordkeeping system to hold or support
99/007 Specification 3: VERS Standard Electronic Record Format
Metadata
Content
Specified by
99/007 Specification 2: VERS Metadata Scheme
Specified by
99/007 Specification 4: VERS Long Term Preservation Formats
Specifies export mechanisms
99/007 Specification 5: Export of Electronic Records to PROV

Specification 1 (System Requirements for Preserving Electronic Records) details the overall requirements on a recordkeeping system for preserving electronic records over a significant period. Amongst other requirements, the recordkeeping system must be capable of exporting the records in a standardised format.

The overall features of this standardised format are defined in *Specification 3 (VERS Standard Electronic Record Format)*, but some details are defined in two other Specifications. *Specification 2 (VERS Metadata Scheme)* defines the meaning and allowed values of the metadata that appears in a record. *Specification 4 (VERS Long Term Preservation Formats)* defines the formats in which the record content must be expressed.

Specification 5 (Export of Electronic Records to PROV) defines the mechanisms by which records are exported to PROV.

Relation to Version 1 of this Standard. This version of the VERS Standard completely replaces Version 1 of the Standard. Version 2 is identical in its base requirements, but makes those requirements clearer and more explicit. It also contains a number of conditional and optional extensions to Version 1.

Table of Contents

1 Introduction

Information in this Advice is purely informative. Nothing in this Advice imposes any requirements on a VERS compliant implementation; requirements are only imposed by the associated Specifications.

The purpose of this advice is to discuss the reasons why the long-term preservation formats acceptable to PROV have been chosen.

A long-term preservation format is one that is expected to continue to be capable of being rendered for a very long period of time.

2 Background

The theoretical background to the approach taken in VERS to preserving the ability to access the content of records is discussed in *Advice 9: Introduction to VERS*.

In summary, however, a major issue with preserving electronic records is ensuring that the contents of data files can be displayed for the indefinite future. Displaying the contents of a data file requires a program that interprets the data file and renders the information. This program may be the original application or a replacement. If the program is the original application, the challenge is keeping the original program running despite changes in the computer environment in which the program is running. If the program is a replacement, the challenge is ensuring that the replacement program renders the information in the data file faithfully.

Cost is a secondary, but important practical issue, when selecting a strategy for ensuring the continued display of electronic records. Unlike preserving paper records, all strategies for ensuring continued access to electronic records are active. That is, they require an archive to do things, and doing things requires resources (particularly people and money). An appropriate preservation strategy minimises the amount of resources required to ensure continued access.

2.1 Preservation strategy

VERS ensures continued access to the content of electronic records by converting the content to a long-term preservation format. The selection of a long-term preservation format is based on the following considerations:

- The format captures the essential characteristics of the original format. For PROV, the essential characteristic is that the long-term preservation format captures how the original user saw the record.
- That it is possible to re-implement a program to display the format from a specification. The worst case scenario is that a technician is required to implement a program from scratch with, initially, no other information than that contained in a VEO. In order to do this, they will require:
    - to be able to identify the format. This information is contained in metadata within the VEO. (Specifically these elements are M128 File Encoding and M131

Rendering Text. Further information about these elements is contained in *PROS 99/007 Specification 2: VERS Metadata Scheme*, and its associated Advice.)

- to be able to obtain a copy of the specification of the format. Normally, we would expect this specification to be in the form of a published document available from a library (or archive). Note that it is necessary to ensure that the software vendor supplying the program actually followed the specification; it is not unknown for vendors to unofficially extend the specification or not to implement it accurately.

The VERS Standard does not specify *when* the content of a record is to be converted to the long-term preservation format. It must be converted by the time of export to PROV, but it may be converted at any time prior to this.

The challenges with conversion are:

- ensuring that it is possible to convert the original. There are many reasons why it may not be possible to convert upon export. These include the lack of a suitable program to display the original content (e.g. the program may have been discarded, or may no longer run), or the file may be encrypted or protected by a password and the access key has been lost.
- ensuring that the conversion retains fidelity with the original. The main reason for loss of fidelity is simply that the conversion is being performed by a different program (or a different version of the program) than that used by the original creator of the record. A subsidiary issue of ensuring conversion fidelity is being able to demonstrate the fidelity of conversion. This is equivalent to ensuring, for example, that a microfilming or digitising project produces clear images of the original paper records.

Both of these challenges occur whenever the conversion occurs, but the challenges may be more severe the later that conversion occurs.

2.2 Cost minimisation strategy

Selection of long-term preservation formats has a critical impact on the cost of providing access to records over a long-term.

Each long-term preservation format accepted increases the cost of providing access. Each format normally requires a separate program in order to display it. The worst case scenario is that this program must be implemented from scratch at considerable expense. But even if programs are commercially available, it is necessary to purchase copies of the program. Note that it is not just a matter of obtaining these programs once — as an archive's systems change over time it will be necessary to replace these access programs periodically. At PROV we assume that this will be necessary every five years or so.

In addition to the cost of obtaining (and replacing) access programs, there is an ongoing cost in running these programs. This cost includes the cost of the hardware required to run the program and staff costs for installation, maintenance, and administration.

In the long-term it may be necessary to supersede a long-term preservation format. Typically this would occur if support for the original long-term preservation format was deemed to be too expensive and there was a suitable current format. Replacing a long-term preservation format requires a suitable conversion program. Such a conversion program is not simple, as it requires:

- monitoring to determine when the existing long-term preservation format is ceasing to be viable
- an evaluation, to determine an appropriate replacement format

- sourcing or writing of routines to perform the conversion
- identification of all instances of the superseded format in the collection
- conversion of all instances
- extensive testing to ensure that the conversion is accurate
- documentation of the conversion process.

Only the cost of the actual conversion step is dependent on the number of records to be converted. The cost of planning, testing, and documenting the conversion is largely independent of the number of records converted. To be more precise, the one-off cost of planning, testing, and documenting the conversion is amortised over the records converted, and hence is lower the more records there are in that format. For this reason, it makes economic sense to minimise the number of formats held in an archive. This reduces the number of formats that may need to be converted, and increases the number of records in that format.

3 Version History

3.1 Changes between Version 1 and Version 2

Specification 4 is new in Version 2 of the Standard, however some of the content was formerly in Specifications 1 and 3. In addition, the following changes have been made:

- plain text format has been explicitly added as an acceptable long-term format
- the requirements on PDF have been tightened
- the TIFF format has been added as an acceptable long-term format.

3.2 Changes between Version 2 and Version 2.1

Version 2.1 added the following file formats:

- PDF/A
- JPEG
- JPEG 2000
- MPEG

4 Compliance

Compliance to the associated Specification is achieved by the demonstration of accurate conversion to the required long-term format.

Accurate conversion means that the resulting file accurately reflects what the original creator of the record saw when they viewed the record. At a *minimum* the conversion should result in a file that contains the same information and appearance as if the record was printed from the originating application.

It is only necessary to demonstrate conversion to appropriate formats. For example, if a system only holds compliant TIFF images, the system will be compliant even though it cannot convert the TIFF images to text or to PDF.

5 Use of Encryption/Passwords/Copy Protection

Encryption, passwords, and copy protection are used to secure the contents of a file from unauthorised access.

- Encryption encrypts the contents of a file to prevent unauthorised access. Access is dependent on knowing the decryption key.
- Password protection is usually combined with encryption. The application will not open the file unless the correct password has been supplied.
- Copy protection uses a variety of mechanisms to prevent unauthorised copying. It is typically used to protect the rights of copyright holders.

All three mechanisms are designed to prevent unauthorised access. The purpose of an archive is to provide indefinite access to records. These two purposes can clash; just who is authorised to access a record is something that may change over the life of a record. Ultimately, records that are initially extremely confidential may be available for anyone to access.

The use of encryption and password protection opens the possibility of losing the record if the decryption key or password is lost. The potential for loss of keys or passwords is particularly high when there is a significant delay in converting the record to a long-term preservation format. While there are agencies that require the additional protection of encryption, PROV strongly recommends that protection against unauthorised access be implemented using the normal system security and that record contents not be encrypted or protected by passwords.

PROV will not accept any permanent records that are encrypted or protected by passwords, as the risk of losing the record through loss of the decryption key or password is too high.

The use of copy protection is likely to significantly impede conversion of the record content to a long-term preservation format. It is also likely to cause the loss of the record content due to the difficulty of re-implementing the copy protection scheme.

PROV will not accept any permanent records that use copy protection.

6 Selection of Long Term Preservation Formats

In selecting a long-term preservation format there are a number of issues to consider. These include:

- the characteristics of the record that it is necessary to preserve
- how easy it is to re-implement software to render the record, or to convert the record to another format.

6.1 What characteristics should be preserved?

The most critical issue when selecting a long-term preservation format is to decide what characteristics of the record need to be preserved.

At PROV, it was decided that the most critical characteristic to preserve was the appearance of the record. The preserved record should be as close as possible in appearance to what the original creator of the record saw. This means, for example, that:

- text should retain its font, weight, and colour. For example, a warning in bright red 40 point text should not be displayed in 12 point black characters when the record is viewed.
- objects such as images and text should not move from page to page depending on the idiosyncracies of the layout algorithms. For example, an image should be associated with its caption and not be arbitrarily relocated to another page.
- information not originally displayed should not be subsequently displayed. For example, when the record of an email is displayed, it should show the email as the original user saw it. The record should not show email headers that were not displayed to the original user (although these would be recorded in the record).

Accurately rendering record content is a key advantage of PDF, for example, over XML representations of documents. PDF precisely and accurately describes each page in a document. XML, on the other hand, describes the logical structure of the document. While stylesheets make it possible to indicate the desired appearance of an XML document, it is not possible to guarantee accuracy of rendition, for two reasons:

- there is no requirement for browsers to accurately apply the style specified in a stylesheet.
- the appearance of the pages depends on the layout algorithms used by the browser. Different algorithms, for example, could result in text being moved between pages.

Other organisations could make different decisions about the essential characteristics of a record that must be preserved. For example, it could be decided that it is only necessary to preserve the information in the record and a sense of the original appearance. In this case an XML representation might be appropriate. Other organisations, on the other hand, could feel that it is necessary to preserve the look and feel of the original application that created the document (this is particularly true where museums wish to preserve software as artefacts).

6.2 Re-implementing rendering software

There are often several possible file formats that will preserve the desired characteristics. In order to select amongst these alternative formats, PROV prefers the format that is easiest to re-implement.

The worst case preservation scenario is where it becomes necessary to re-implement, from scratch, software to render a record. In order to re-implement rendering software, the future developer will require:

- the ability to identify the format. This information is contained in metadata within the VEO. (Specifically these elements are M128 File Encoding and M131 Rendering Text. Further information about these elements is contained in *PROS 99/007 Specification 2: VERS Metadata Scheme* and its associated Advice.)
- the ability to obtain a copy of the specification of the format. Normally, we would expect this specification to be in the form of a published document available from a library (or archive).

6.2.1 Simple formats

The ideal format is one that is simple enough to allow it to be described it in a short piece of text that can be included in the M128 File Encoding element within the VEO. Very few formats are this simple.

6.2.2 Published formats

For records that are too complex to describe in a short piece of text, the preferred format is one that has been formally specified and published. The VEO must include a reference to the published specification in either the M128 File Encoding or M131 Rendering Text elements. An archive can build up a library of the specifications that it uses, or can rely on accessing the specifications through legal deposit libraries.

Almost all records are sufficiently complex to require an external published format. Consider a record that contains just text in several languages. In order to render the text it is necessary to convert the bit stream in the file into a sequence of character numbers. It is then necessary to map each character number into a glyph (the character image displayed on the screen). The Unicode standard [Unicode] describes thousands of character glyphs and has several mechanisms for converting the character numbers into bit streams. It is clear that this complex specification could not be summarised in the M128 File Encoding element. Instead, this element will contain a reference to the Unicode standard.

Most electronic records are a great deal more complex than a simple text file. For example, consider a document such as this Advice. Although most of the content is text, the characters have formatting applied (e.g. colour, font, and weight). The characters are combined to form higher level formatting units (e.g. paragraphs) which have their own formatting applied. Some paragraphs have particular characteristics (e.g. they are numbered, form indexes, or a headers). Finally, the document contains objects that are not textual, but images. These images may have their own specifications (e.g. JPEG, TIFF, GIF). The result is that any specification rich enough to cater for all the features of current electronic documents is very complex, and will often contain references to other specifications.

Where there are a choice of several suitable formats, the following criteria are used to select between them:

- *Widely used formats*. Many formats are developed but are not widely adopted. Widely adopted formats are preferred for long-term preservation as it is far more likely that software will continue to be available to render the format.
- *Non proprietary formats*. Non proprietary formats (e.g. international standards) are preferred over proprietary formats. One advantage of non proprietary formats is that the specification is independent of a particular vendor. It is much harder, then, for vendors to add additional features that use undocumented extensions to the format.
- *Independent implementations*. Formats that have several independently written implementations are preferred over formats where there is only one implementation. Independent implementations help ensure that vendors accurately implement the specification. It should be noted that significant ‘re-badging’ occurs in the computer industry and this can make it difficult to determine how many independent implementations there actually are. In a given situation it may appear that there are several independent implementations, but, in fact, all the implementations may use the same code licensed from the original developer.

6.3 Industry standard formats

For many types of records there is no suitable format with a published specification. In this case, VERS recommends that an 'industry standard' format be selected. These formats need not be published.

The long-term preservation strategy is that records in an 'industry standard' format must be converted before the format becomes obsolete. This approach is more risky than adopting a format that has a published specification for two reasons:

- It is necessary to monitor the viability of the 'industry standard' format and to convert the records before the format becomes unviable. If the monitoring fails, and records are left in an industry standard format after this format ceases to be used, the records may be lost as it may not be possible to obtain software that will render the record.
- It is necessary to accept whatever conversion accuracy is produced by the available conversion utilities, as an archive cannot implement its own conversion untilies.

The basis for using an 'industry standard' format is economics. If the industry standard product has the major share of the marketplace, it is unlikely to be replaced easily or quickly. Consequently, records in this format will be accessible for a reasonable period.

Further, if the industry standard product is ever replaced, the replacement products are almost certain to read the proprietary format of the product they replace. Again, this is simple economics. If a new product cannot read and manipulate the data formats used by the industry leader, the new product is unlikely to gain market share.

7 Standard Long Term Preservation Formats

7.1 Text

Text files are those which contain only text; typically letters, numbers, punctuation, spaces, and tab characters. Text files were once very common, but few files on most office computer systems are likely to contain only plain text, as modern office applications support documents with sophisticated formatting and images. In the future, plain text files may become more common. HTML pages and XML documents, for example, are plain text files. Other examples of text files include software source files, and some emails. On Windows file systems many text files have the extension '.txt'.

Text files can be opened using the simplest text editors such as Notepad and vi, and more complex programs such as Word and Wordpad.

A text file basically consists of a sequence of characters which are represented in the text file as numbers. In order to display the contents of a text file it is necessary convert the bit stream in the text file into a sequence of numbers and then convert the numbers into the characters. Both conversions are defined by standards.

The current root of English language textual encodings is ASCII (American Standard Code for the Interchange of Information) which dates from 1968 and which subsequently became an international standard (ISO 646:1991) [ISO646]. ASCII (ISO 646) was subsequently extended in the 1980s to allow coverage of some non English languages. The new standard is known as ISO 8859 [ISO8859]. The important thing to note is that in all three standards

(ASCII, ISO 646, and ISO 8859) the first 127 characters are identical, so English text is represented identically in all three.

Text files with non Latin characters should be represented as Unicode [Unicode]. Unicode itself is defined by the Unicode Consortium, but a functionally equivalent standard is produced by ISO as ISO/IEC 10646 [ISO10646]. Unicode defines virtually all known characters in most languages. The standard continually evolves, mainly by the addition of new characters for additional languages. For this reason, any object that conforms to the current version of the Unicode standard now, will conform to future versions of the standard.

Like all character standards, Unicode defines the character glyphs and assigns each of them a number. Unlike the other character standards, the character numbers can be physically encoded in a digital file in a variety of different ways. The common physical encoding mechanisms are UTF-8 and UTF-16. UTF-8 has the characteristic that the characters available in ASCII are encoded identically to ASCII. Thus Unicode text containing only the characters available in ASCII and encoded in UTF-8 is identical to the same text encoded as ASCII.

In summary, if the text is English it should be expressed in ASCII. The resulting characters will be identical irrespective of whether they are considered to be encoded in ISO 646, ISO 8859-1, and Unicode (ISO 10646) represented in UTF-8.

Languages other than English should be expressed in Unicode (ISO 10646) represented in UTF-8.

7.2 PDF (Portable Document Format) and PDF/A

The Portable Document Format (PDF) is a proprietary format defined by Adobe Systems Incorporated. The specification for PDF has been formally published as a book [PDF], which makes it suitable for a long-term preservation format.

The PDF format is designed to precisely describe pages of documents; one purpose of PDF is to ensure that documents are printed identically irrespective of the printer used. This is a key benefit of using PDF in preserving record content: in PDF each page is rendered exactly as the creator intended. Unlike other representations (e.g. Word, or XML) a new version of the PDF reader will not result in text or other objects moving around a page or between pages, or changing font size or colour.

PDF-A is an international standard that prohibits the use of certain features of PDF. The prohibited features are those which may make it difficult to render (display) PDF files in the future. Such features include:

- Not including the font definitions in the file. All PDF-A files must include the font definitions of all characters included in the file. Not including the font definitions makes the PDF file smaller, but means that the file cannot be accurately displayed.
- Externally referenced content. All of the content of a PDF-A file must be included in the file. Content that exists outside a PDF file is most likely to be lost.
- Undefined formats. PDF files may include arbitrary Javascript programs, sound files, or video files. The standard leaves these formats undefined.

PDF is very widely used as a pre-press tool (i.e. preparing publications for printing) and in making available electronic copies of printed publications. There are at least two free viewers: Adobe Reader (available from Adobe Systems at http://www.adobe.com/products/acrobat/readstep2.html visited 10 May 2006), and Ghostscript (available from http://www.cs.wisc.edu/~ghost/index.html visited 15 June 2003).

VERS specifies PDF 1.4. This is generated by Acrobat 5.x. (Acrobat 4.x generated PDF 1.3, and Acrobat 3.x generated PDF 1.2).

There are a number of minor features of PDF that may make it difficult to preserve the ability to render portions of a PDF document. These have been prohibited in PDF-A and in VERS. It should be noted that most of these problematic features are very obscure and would be unlikely to appear in documents produced by Adobe Distiller.

These include the following:

- PDF files must be self-contained. There are a number of options in the PDF specification that allow components of a PDF document to be external to the PDF file.

  These include:
  - All external files must be embedded. VERS requires that all external files be embedded in the final PDF file. A PDF document is composed of multiple object streams, and these object streams may be contained in ‘external files’. However, the specification allows these external files to be actually embedded in the final PDF file (this is the default behaviour when generating PDF when using Adobe Distiller).
  - All fonts must be embedded within the PDF file. This is controlled by an option in Adobe Distiller.

    Note that even after embedding the fonts within a PDF file there are some issues with TrueType fonts (see section 5.5.5, p.333, [PDF]). We assume that the standard PDF generation tools (such as Adobe Distiller) implement the detailed recommendations in this section. Note that these issues apply to any archival format that uses TrueType fonts.
  - Reference Xobjects may be used, but the external file must be embedded in the containing document.
  - The Open Prepress Interface (OPI) must not be used. OPI is a mechanism where large high-resolution images produced during the pre-press production process are stored separately to the PDF file.
- The PDF encryption option is prohibited for the reasons give in section 5.
- Rendering options dependent on specific output devices are prohibited. These include:
  - CIE-based colour to device colour.
  - Conversions amongst device colour space.
  - Transfer functions.
  - Halftones.
  - Scan conversion.
  - Overprinting.

  These options are designed to allow the fine control of printing on specific printers (or classes of printers).
- Javascript actions associated with the document are prohibited.
- The following types of annotations are prohibited:
  - Plugin extensions
  - File Attachment annotations, as it may not be possible to identify the application necessary to render the attachment.
  - Sound annotations, as it may not be possible to identify the application necessary to play the sound.
  - Movie annotations, as it may not be possible to identify the application necessary to play the movie.
  - Any Javascript actions, and the Launch, SubmitForm, or ImportData actions associated with Form annotations. Apart from requiring the inclusion of a Javascript interpreter (and hence specification), the result of executing Javascript

can be almost anything (including references to external objects). Hence, it is extremely difficult to preserve the ability to evaluate Javascript over the long-term.

7.3 TIFF (Tagged Image File Format)

TIFF files are used for raster images. Raster images are represented by an array of dots (pixels); the most typical examples are the result of scanning a document or image. When compared with other raster file formats (e.g. GIF), TIFF is very widely supported (by most, if not all, image processing and viewing applications), and supports sophisticated colour management features. Consequently, TIFF is recommended for use when preserving a scanned image.

TIFF files must conform to the Baseline TIFF specification [TIFF].

In addition, the following compression extensions are allowed:

- CCITT bilevel encodings.
- LZW compression. Note that LZW compression is patented. This is actually of benefit in an archive, as a patent, by definition, contains sufficient detail to re-implement the algorithm.

No other extensions are supported.

7.4 JPEG

JPEG files are very widely used to represent continuous tone images (e.g. photographs and greyscale images) due to the excellent compression algorithm. JPEG images are particularly widespread on the internet. JPEG should not be used to compress images with sharp edges as the compression algorithm often results in compression artefacts (known as ‘ringing’). These artefacts take the form of fuzzy echo lines parallel to the sharp edge. A second compression artefact is that the picture is tiled with small rectangles that blur the fine detail of the image. These artefacts result from the compression algorithm. If these artefacts are not acceptable, TIFF or JPEG 2000 should be used.

JPEG is defined by an international standard [ISO10918]. Note that identical standards (except for the identifier) are issued by ISO and the ITU.

Technically, the JPEG standard supports both lossless and lossy compression. With lossless compression, when an image is compressed and then decompressed the resulting image is identical to the original image (i.e. no information is lost). With lossy compression, the uncompressed image will be different from the original image. In practice, almost no use is made of the lossless mode in JPEG and almost all JPEG images use lossy compression. (Note that there is also a JPEG Lossless standard (ISO 14495-1 or ITU-T T.87) which is a completely separate standard that is not supported by VERS.)

The use of lossy compression is a contentious issue for long term preservation of images. Ideally, lossless compression should be used for archival master images. This is because:

- The image preserved should be the highest quality possible.
- Future preservation actions on a lossy compressed image may reduce the quality of the image even further. For example, if the JPEG image format became obsolete and it was necessary to migrate the image to a new format, it would be necessary to uncompress the JPEG image and recompress it using the new format. The

uncompressed image will have lost information when compared with the original image, and the new compression may well lose additional information. It is even theoretically possible that the interaction between the two lossy compression algorithms could result in extremely poor results.

In practice, it is possible to make too much of this argument for the following reasons:

- For many images accurate colour rendition is not essential. For example, documents in an archive are often written on coloured stock, or are printed using simple printing processes. Precise colour control is consequently not necessary. With such documents the compression algorithm should be chosen to preserve the detail of the image and to give the feel of the original.
- JPEG is a well defined standard. While it may become obsolete, it is very unlikely that an archive will be unable to re-implement the decompression software when required. Consequently, it is unlikely that JPEG images will need to be migrated to another format. If it ever was required to migrate to another format, advances in compression techniques should mean that a lossless compression format could be selected to avoid the second loss of quality.

For these reasons, we would strongly recommend for images:

- where colour fidelity is important, we recommend using a lossless compression (e.g. TIFF or JPEG 2000)
- that have sharp edges, or it is desired to retain the fine detail, we would recommend using a lossless compression algorithm or JPEG 2000.
- where the slight loss of colour fidelity is acceptable, we consider that JPEG is an acceptable format. This includes where the paper stock is coloured, the ink is coloured, or printing process is too simple to ensure accurate colour rendition.
- that are already compressed using JPEG when received, the existing compression should be retained. There is no benefit in uncompressing a JPEG image, and then recompressing using a lossless compression algorithm as the information has already been lost. In fact, re-compressing using a lossless algorithm may simply be hiding the fact that information has been lost by a previous JPEG compression.

The basic JPEG standard defines several profiles. All are accepted by VERS:

- Baseline. This is a sequential DCT (lossy) process with scans with 1 to 4 components, each of which uses 8 bit samples. It uses Huffman coding with 2 AC and 2 DC tables.
- Extended. This extends the Baseline profile. It allows sequential or progressive encoding, 8 or 12 bit samples, and either arithmetic or Huffman encoding with 4 AC and 4 DC tables.
- Lossless. This is a lossless process.
- Hierarchial. This uses multiple frames, each of which can either use the extended or lossless processes.

The JPEG standard defines an interchange format. This format specifies the bitstream used to encode an image. The ‘abbreviated format for compressed image data’ is not accepted as a long term preservation format by VERS. Images using this abbreviated format do not include some of the tables required to decompress the image (instead, it is assumed that they are built into the decoder). Consequently, this format is not appropriate for long term preservation. In practice, it is unlikely that any normal generated JPEG image would use this mode as it could not be decoded by all viewers.

JFIF (JPEG File Interchange Format) [JFIF] defines a further profile of JPEG. This profile is not included in the international standard, but was defined by Eric Hamilton of C-Cube Microsystems. This profile is both a restriction and extension of the standard interchange format. Many, if not most, JPEG files are actually JFIF files. It strongly recommends use of the Baseline JPEG profile and a YCbCr colour space. It uses an application specific extension to include the version, units, X pixel density, Y pixel density, and a thumbnail. Standard decoders will be able to decompress JFIF files and will ignore the application specific extensions. If it desired to check if a particular file is JFIF encoded, the file should commence with the hex string 'FF D8 FF E0 xx xx 9A 96 49 00' (where the 'xx' may be any hex number).

There are a number of other encodings of JPEG. These include SPIFF, EXIF, and JTIP. These have not been investigated due to their rarity and are not accepted as long term preservation formats for VERS.

7.5 JPEG 2000

JPEG 2000 (ISO 15444) was developed as a more modern replacement for JPEG. Its most important features are greater compression for equal quality of resulting imaging and improved lossless compression.

The improvement in compression, while present, is not great. [SantaCruz] comments that "there have not been any truly significant [advances] in image compression efficiency in the past decade". The improvements in compression are mainly seen at very high and very low bitrates (i.e. the number of bits to encode a pixel, or bpp). At the 'near visual lossless' performance, JPEG 2000 is only around 20% better than JPEG. The trade-off is that computational complexity for JPEG 2000 is significantly higher. In 2000, JPEG codecs were around 3 times faster than JPEG 2000 codecs and, while it was expected that some improvement would be made, this difference was never expected to close.

[Jakulin] describes the artifacts introduced by JPEG 2000 and compares them (using the same images) with the original image and JPEG. The conclusion was that, although JPEG 2000 solves some of JPEG's problems at lower bit rates, it introduces others. The author recommends that JPEG 2000 is more appropriate for images that do not have noisy textures. He goes on to suggest that between 2.5 and 3 bpp (bits per pixel), JPEG 2000 is superior, and above 3 bpp, there is no difference. For black and white images JPEG 2000 is superior to JPEG, but not as good as a lossless black and white compression (e.g 16 colour greyscale PNG).

[Gormish] concluded that JPEG 2000 "is unlikely to replace JPEG in low complexity applications at bitrates in the range where JPEG performs well".

The standard consists of a large number of parts:

- *Part 1: Core coding system.* Published as an international standard in 2000 and revised in September 2004. Intended as a royalty and license free core with maximum capability to interchange between implementations. Includes the file format (JP2).
- *Part 2: Extensions.* Published as an international standard in May 2004. Extensions for circumstances where "interchange is less important than other requirements (e.g. ability to handle a particular type of data)" [1]. It is not expected that JPEG 2000 files using the part 2 extensions will be handled by part 1 decoders. Since interchange is critically important in an archival setting, the part 2 extensions should not be accepted without consideration of their archival impact. Includes a new file format (JPX) that supports multiple compositing layers, animation, and extended colour spaces

- *Part 3: Motion JPEG 2000.* Published as an international standard in November 2004, and currently under review. This defines a file format (MJ2 or MJP2) for storing motion sequences of images. Inter frame coding is not supported. This has now been effectively superseded by Part 12.
- *Part 4: Conformance testing.* Published as an international standard in 2002 and a revised in December 2004. Compliance tests for JPEG 2000 Part 1. Includes a set of compliance classes for decoders.
- *Part 5: Reference software.* Reference implementations in C and Java for Part 1. Published as an international standard in November 2003.
- *Part 6: Compound image file format.* Published as an international standard in 2003. Compound images include both text and continuous tone photographs. This part defines the JPM file format, which uses the Mixed Raster Content model of ISO 16485. It supports multi-page documents with many objects per page. The objects can be compressed using different algorithms (e.g. JBIG for b/w images). JPM is an extension of JP2, and uses content defined in JPX (Part 2).
- Part 7: Not produced.
- *Part 8: JPSEC.* This part examines security aspects of JPEG 2000. It includes encryption of the image or metadata, source authentication, data integrity, conditional access, and ownership protection. This part is currently a Draft International Standard.
- *Part 9: Interactivity tools, APIs and protocols.* Published as an international standard in November 2005. Interactive protocols and API. This part defines a protocol (JPIP) which sits on top of HTTP and allows access to images (or portions of images). It is designed to avoid retransmission of image portions. It is primarily focussed on access to Part 1 images, but provides access to some of the format extensions in Part 2. It also allows selection amongst multiple codestreams in JPX (Part 2), MJ2 (Part 3), and JPM (Part 6).
- *Part 10: JP3D.* This part is concerned with the compression of 3 dimensional ‘images’ (actually data arrays).
- *Part 11: JPWL.* This part provides additional error detection and correction to allow for transmission of Part 1 images over an error prone wireless connection. In particular it attempts to protect the header. Since this only adds additional information to an existing image during transmission, it should not be an issue for a long term preservation format.
- *Part 12: ISO Base Media File Format.* Published as an international standard in 2004 and revised in April 2005. This is a common format for both JPEG 2000 and MPEG-4 files. It is intended to provide a base file format for future applications.

VERS currently only accepts images encoded using Part 1 of the standard. This is primarily due to the lack of time required to investigate the other Parts, but also due to the immaturity of most of the remaining parts.

Restricting acceptance of JPEG 2000 images to those conforming to Part 1 means that the images must be encoded using the JP2 format. Other parts of the standard define a variety of other formats:

- JPX. Defined in Part 2 to support the extensions defined in that part. It appears that the extensions can either be represented in a way that is upwardly compatible with JP2, or in an incompatible way.
- MJ2/MPJ2. Defined in Part 3 to support sequences of frames encoded in JPEG 2000. Appear to be superseded by the ISO Base Media File Format.

- JPM. An extension of JP2 for representing compound documents. A JPM file contains a hierarchy of page collections, pages, layout objects, masks and image objects. A JP2 decoder should ignore the components it does not recognise.
- ISO Base Media File Format. This is a common format for JPEG 2000 and MPEG-4 files. Does not appear (yet) to be widely used.

7.6 MPEG-4

For video encoding we have chosen MPEG-4 (ISO/IEC 14496) as a long term preservation format.

MPEG-4 was designed to support interactive multimedia on viewers ranging from handheld screens (e.g. iPod) to high definition TV (HDTV). It provides an almost bewildering array of technologies. Apart from multiple methods of compressing conventional video and audio, MPEG-4 includes facilities to represent shapes, animations, text, and so on. Unlike a conventional video stream, MPEG-4 is defined in terms of objects within scenes. These objects may be derived from ‘real sources’ (e.g. conventional video), or ‘synthetic sources’ (e.g. CAD, computer animation). Synthetic sources even include symbolic representations of musical scores.

MPEG-4 was chosen over MPEG-2 (the format used on DVDs) for the following reasons:

- Superior compression of the video stream.
- Better support for non-television video. This includes non-standard frame sizes, and in particular, non-standard frame rates. This is important in a government archive where much of the preserved footage may not be captured from conventional television footage, for example, CCTV footage.
- Potential for future improvements in the standard. Whereas MPEG-2 appears to have completed development, MPEG-4 is still being actively developed.

The MPEG-4 standard currently (May 2006) consists of 22 parts, not all of which have been confirmed as international standards (these parts are listed at the end of this section). Because of the complexity of MPEG-4, VERS will only accept MPEG-4 files derived from real sources, that is, conventional video and audio. We accept the following parts of the standard:

- Part 1 Systems. This part defines the overall structure of an MPEG-4 object. Note that we do not accept the file format defined in the 2001 edition of this part (in Section 13), as this has been removed from the 2004 edition and replaced by the file format defined in Part 14.
- Part 2 Visual. This defines the compression of video streams.
- Part 3 Audio. This defines the compression of audio streams.
- Part 10 Advanced Video Coding (AVC). This is an improved video compression scheme, which is identical to the ITU-T H.264 compression scheme.
- Part 14 MP4 File Format. This replaces the file format originally defined in Part 1.
- Part 15 Advanced Video Coding (AVC) File Format

It should be noted that Part 4 of the standard describes testing for conformance to the standard and Part 5 contains a reference implementation. These can be used to ensure that an implementation is correct. The reference implementation in Part 5 conforms to the standard, but may not be efficient or produce good results. (It was noted in one reference that the vendors developing the standard do not have any incentive to make available a free high quality MPEG-4 implementation.) It appears that both Part 4 and 5 are updated as new parts of ISO 14496 become international standards.

Advanced Video Coding (AVC, also known as ITU-T H.264) is a newer video compression algorithm for MPEG-4 that attempts to achieve flexible good quality video at bit rates 50% or less than previous compression without significant increases in decompression complexity. AVC appears to have greater take-up than the original video coding specified in Part 2. We have chosen to support the basic AVC profiles. The Fidelity Range Extensions allow higher quality, but currently lack support.

Conformance to the MPEG-4 standards is specified in terms of profiles and conformance levels (see [Koenen]). A profile is a collection of tools (which represent bit stream syntaxes) which can be used to render onto a device. A level is essentially a measure of complexity; typically a bit rate that will allow a particular level of performance. A conformance point is the combination of a profile at a specific level (very crudely, a set of bit streams that can be handled at a particular bit rate). The standard defines relatively few conformance points.

- Visual Profiles. We limit video objects to the Main L4 visual profile. This allows an MPEG-4 file to represent a HDTV video stream with a bit rate of 38.4 Megabits/section. We do not accept the Simple Scaleable, N-bit, Scaleable Texture, Simple FA, Basic Animated Texture, or Hybrid profiles. These other profiles mainly represent video information from synthetic sources. For AVC we accept the Baseline, Extended, and Main Profiles up to L4. This allows encoding of HDTV at 60 or 30 frames per second.
- Audio Profile. We accept the Main L4 audio profile. This profile handles all audio bitstreams (including synthetic sources) defined in MPEG-4 with a bit rate sufficient to encode a 5.1 channel object. We could have restricted the VERS to the Scalable profile (which excludes the synthetic sources), but this also excludes the AAC Main and AAC SSR tools.
- Graphics Profile. We do not accept any graphics profile, as these represent synthetic sources.
- System Profiles. There are two aspects to system profiles: Scene Graphs and Object Descriptors. VERS accepts the Core Object Descriptor profile (this is the only profile defined). VERS accepts the Audio and Simple 2D Scene Graph profiles; the Complete 2D and Complete profiles require the display device to be able to manipulate objects.

There are several alternative file formats for MPEG-4 files. Of these, the original format which was defined in Section 13 ISO/IEC 14496-1:2001 seems to be obsolete as it is no longer included in the 2004 edition. The replacement file format is defined in Part 14 (MP File Format), and this is the format that VERS will accept. There is another format (ISO Base Media File Format – BMFF) defined in Part 12. The BMFF is a joint format and is shared with JPEG 2000, however, it does not seem to be widely used. Finally, there is the Advanced Video Coding (AVC) File Format (Part 15) which is used to encode the AVC compressed video.

Where audio is captured along with video, we recommend that MPEG Audio layer 3 compression is used to encode the audio within MPEG-4. MP3 produces small audio files with acceptable quality.

MPEG-4 currently consists of 22 parts. In the following list, the current (May 2006) international standards are identified, together with any approved corrigendum and amendments. Corrigenda correct minor mistakes in the published standard, while amendments extend the standard.

- Part 1 Systems (ISO/IEC 14496-1:2004, ISO/IEC 14496-1:2004/Amd 1:2005, ISO/IEC 14496-1:2004/Amd8:2004)

- Part 2 Visual (ISO/IEC 14496-2:2004, ISO/IEC 14496-2:2004/Cor 1:2004, ISO/IEC 14496-2:2004/Amd 1:2004, ISO/IEC 14496-2:2004/Amd 2:2005)
- Part 3 Audio (ISO/IEC 14496-3:2005)
- Part 4 Conformance testing (ISO/IEC 14496-4:2004, ISO/IEC 14496-4:2004/Cor 1:2005, ISO/IEC 14496-4:2004/Amd 1:2005, ISO/IEC 14496-4:2004/Amd 1:2005/Cor 1:2005, ISO/IEC 14496-4:2004/Amd 2:2005, ISO/IEC 14496-4:2004/Amd 3:2005, ISO/IEC 14496-4:2004/Amd 4:2005, ISO/IEC 14496-4:2004/Amd 5:2005, ISO/IEC 14496-4:2004/Amd 6:2005, ISO/IEC 14496-4:2004/Amd 7:2005, ISO/IEC 14496-4:2004/Amd 8:2005, ISO/IEC 14496-4:2004/Amd 9:2006, ISO/IEC 14496-4:2004:Amd 10:2005)
- Part 5 Reference Software (ISO/IEC 14496-5:2001, ISO/IEC 14496-5:2001/Amd 1:2002, ISO/IEC 14496-5:2001/Amd 2:2002, ISO/IEC 14496-5:2001/Amd 3:2002, ISO/IEC 14496-5:2001/Amd 4:2002, ISO/IEC 14496-5:2001/Amd 5:2002, ISO/IEC 14496-5:2001/Amd 6:2002, ISO/IEC 14496-5:2001/Amd 7:2002, ISO/IEC 14496-5:2001/Amd 8:2002)
- Part 6 Delivery Multimedia Integration Framework (DMIF) (ISO/IEC 14496-6:2000)
- Part 7 Optimised software for MPEG-4 tools (ISO/IEC 14496-7:2004)
- Part 8 Carriage of ISO/IEC 14496 objects over IP framework (ISO/IEC 14496-8:2004)
- Part 9 Reference Hardware Description (ISO/IEC 14496-9:2004)
- Part 10 Advanced Video Coding (ISO/IEC 14496-10:2005)
- Part 11 Scene Description and Application Engine (ISO/IEC 14496-11:2005)
- Part 12 ISO Base Media File Format (ISO/IEC 14496-12:2005, ISO/IEC 14496-12:2005/Cor 1:2005, ISO/IEC 14496-12:2005/Cor 2:2006)
- Part 13 Intellectual Property Management and Protection (IPMP) Extensions (ISO/IEC 14496-13:2004)
- Part 14 MP4 File Format (ISO/IEC 14496-14:2003, ISO/IEC 14496-14:2003/Cor 1:2006)
- Part 15 Advanced Video Coding (AVC) File Format (ISO/IEC 14496-15:2004, ISO/IEC 14496-15:2004/Cor 1:2006, ISO/IEC 14496-15:2003/Amd 1:2006)
- Part 16 Animation Framework eXtension (AFX) (ISO/IEC 14496-16:2004, ISO/IEC 14496-16:2004/Cor 1:2005, ISO/IEC 14496-16:2004/Cor 2:2005, ISO/IEC 14496-16:2004/Amd 1:2006)
- Part 17 Streaming Text Format (ISO/IEC 14496-17:2006)
- Part 18 Font compression and streaming (ISO/IEC 14496-18:2004)
- Part 19 Synthesized Texture Stream (ISO/IEC 14496-19:2004)
- Part 20 Lightweight Application Scene Representation
- Part 21 MPEG-J Extension for rendering
- Part 22 Open Font Format

8 References

[Gormish] An Overview of JPEG-2000, M.W. Marcellin, M.J. Gormish, A. Bilgin, M.P. Boliek, Proc of IEEE Data Compression Conference, 2000, p523-541, http://rii.ricoh.com/~gormish/pdf/dcc2000_jpeg2000_note.pdf visited 1 March 2006.

[Jakulin] Baseline JPEG and JPEG2000 Artifacts Illustrated, Aleks Jakulin, 2004, http://ai.fri.uni-lj.si/~aleks/jpeg/artifacts.htm, visited 2 March 2006.

[Koenen] Profiles and Levels in MPEG-4: Approach and Overview, Rob Koenen, http://www.chiariglione.org/mpeg/tutorials/papers/icj-mpeg4-si/11-Profiles_paper/11-Profiles_paper.htm visited 10 May 2006.

[ISO646] ISO 7-bit coded character set for information interchange, International Organization for Standardization/International Electrotechnical Commission, ISO/IEC 646:1991.

[ISO8859] 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No 1, International Organization for Standardization/International Electrotechnical Commission, ISO/IEC 8859-1:1998. Note that there are a further 15 parts to this standard to cover non English languages; Unicode (ISO 10646) should be used instead of these parts.

[ISO10646] Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane, International Organization for Standardization/International Electrotechnical Commission, ISO/IEC 10646-1:2000, Universal Multiple-Octet Coded Character Set (UCS) – Part 1 Amendment 1: Mathematical symbols and other characters, Organization for Standardization/International Electrotechnical Commission, ISO/IEC 10646-1:2000/Amd 1:2002, Universal Multiple-Octet Coded Character Set (UCS) – Part 2: Supplementary Planes, International Organization for Standardization/International Electrotechnical Commission, ISO/IEC 10646-2:2001. Note that these are identical in content to Unicode.

[ISO10918] Information Technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines, International Organization for Standardization/International Electrotechnical Commission, ISO/IEC 190918-1:1994. Note that the ITU standard ITU-T T.81 is identical to the ISO/IEC version.

[JFIF] JPEG File Interchange Format, Version 1.02, Eric Hamilton, C-Cube Microsystems, 1 September 1992, (http://www.jpeg.org/public/jfif.pdf visited 1 May 2006)

[PDF] PDF Reference, third edition, Adobe Portable Document Format, Version 1.4, Adobe Systems Incorporated, Addison Wesley, 2001, ISBN 0-201-75839-3 (http://partners.adobe.com/public/developer/en/pdf/PDFReference.pdf visited 30 June 2006), as modified in 'Errata for PDF Reference, third edition' (http://partners.adobe.com/public/developer/en/pdf/PDF14errata.txt visited 30 June 2006).

[PDFA] PDF-Archive, AIIM International, http://www.aiim.org/standards.asp?ID=25013 visited 15 June 2003.

[SantaCruz] A study of JPEG 2000 still image coding versus other standards, Diego Santa Cruz, Touradj Ebrahimi, ISO/IES JTC1/SC29/WG1, http://www.jpeg.org/public/wg1n1814.pdf visited 2 March 2006. (To be published in X European Signal Processing Conference, Tampere, Sept 2000.

[TIFF] TIFF Revision 6.0, Final – June 3, 1992 (http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf visited 30 June 2006.

[Unicode] The Unicode Standard, Version 3.0, The Unicode Consortium, Addison-Wesley, 2000, ISBN 0-201-61633-5, http://www.unicode.org/unicode/uni2book/u2.html visited 30 January 2003.