BRITISH STANDARD

BS ISO/IEC 15444-6:2003

Corrected and reprinted

Information technology — JPEG 2000 image coding system —

Part 6: Compound image file format

ICS 35.040

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI Documents Document Sharing Hub British Standards

Full document details: BS ISO/IEC 15444-6:2003

National foreword

This British Standard reproduces verbatim ISO/IEC 15444-6:2003 and implements it as the UK national standard.

The UK participation in its preparation was entrusted to Technical Committee IST/37, Coding of picture, audio, multimedia and hypermedia information, which has the responsibility to:

— aid enquirers to understand the text;

— present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;

— monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international publications referred to in this document may be found in the *BSI Catalogue* under the section entitled "International Standards Correspondence Index", or by using the "Search" facility of the *BSI Electronic Catalogue* or of British Standards Online.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the ISO/IEC title page, pages ii to iv, pages 1 to 71 and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 2 December 2003

© BSI 2 December 2003

ISBN 0 580 43010 3

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

INTERNATIONAL STANDARD

ISO/IEC 15444-6

First edition
2003-10-15

Information technology — JPEG 2000 image coding system —

Part 6: Compound image file format

Technologies de l'information — Système de codage d'image JPEG 2000 —

Partie 6: Format de fichier d'image de composant

Reference number
ISO/IEC 15444-6:2003(E)

AVA Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

Contents

Page

Foreword . . . v
1 Scope . . . 1
2 Normative references. . . . 1
2.1 Identical Recommendations | International Standards . . . 1
2.2 Additional References . . . 2
3 Terms and definitions . . . 3
4 Abbreviations. . . . 4
5 General Description. . . . 4
5.1 Mixed Raster Content Model. . . . 4
5.2 File Elements and Structure . . . 6
5.3 JPM Use Scenarios (Informative) . . . 17
Annex A Compound Image File Structure . . . 20
A.1 File Identification . . . 20
A.2 File Organization . . . 20
A.3 Box Definition . . . 22
A.4 Boxes Used in a Compound Image File . . . 22
Annex B Box Definitions. . . . 25
B.1 File Level Boxes . . . 25
B.2 Page Level Boxes . . . 36
B.3 Layout Object Level Boxes . . . 39
B.4 Object Level Boxes . . . 42
B.5 Codestream Element Boxes . . . 47
B.6 General / Common Boxes . . . 51
Annex C Metadata . . . 66
C.1 Adding intellectual property rights information in JPM . . . 66
C.2 Adding vendor specific information to the JPM file format . . . 66
Annex D Profiles. . . . 67
D.1 JPM Profiles . . . 67
D.2 Decompression Profiles . . . 68
Annex E Guidelines for Constructing URLs for JPM Files . . . 69
E.1 Pages and Layout Objects . . . 69
E.2 Metadata boxes. . . . 69
E.3 Labels . . . 70
E.4 Page Collections . . . 70
E.5 Page Thumbnails . . . 70
E.6 Document Thumbnail . . . 70
E.7 Byte Ranges . . . 70

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 15444-6 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 15444 consists of the following parts, under the general title *Information technology — JPEG 2000 image coding system*:

— *Part 1: Core coding system*

— *Part 2: Extensions*

— *Part 3: Motion JPEG 2000*

— *Part 4: Conformance testing*

— *Part 5: Reference software*

— *Part 6: Compound image file format*

— *Part 8: Secure JPEG 2000*

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

Information technology — JPEG 2000 image coding system —

Part 6:
Compound image file format

1 Scope

This International Standard defines a normative but optional file format for storing compound images using the JPEG 2000 file format family architecture. This format is an extension of the JP2 file format defined in ITU-T Rec T.800 | ISO/IEC 15444-1 Annex I and uses boxes defined for both the JP2 file format and the JPX file format defined in ITU-T Rec T.801 | ISO/IEC 15444-2 Annex M. This standard is useful for applications storing multiple pages, images with mixed content, and/or images that need more structure than provided in JP2. Applications that implement this file format shall implement it as described in this International Standard.

This International Standard

— specifies a binary container for multiple bi-level and continuous-tone images used to represent a compound image,

— specifies a mechanism by which multiple images can be combined into a single compound image, based on the Mixed Raster Content model

— specifies a mechanism for grouping multiple images in a hierarchy of layout objects, pages and page collections,

— specifies a mechanism for storing JPEG 2000 and other compressed image data formats,

— specifies a mechanism by which metadata can be included in files specified by this International Standard.

2 Normative references

2.1 Identical Recommendations | International Standards

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

— ISO 5807:1985, *Information processing - Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*

— ISO 8601:2000, *Data elements and interchange formats - Information interchange - Representation of dates and times*

— ISO/IEC 646:1991, *Information technology - ISO 7-bit coded character set for information interchange*

— ISO/IEC 8859-1:1998, *Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1*

— ISO/IEC 11578:1996, *Information technology - Open Systems Interconnection - Remote Procedure Call (RPC)*

— ISO 3166-1:1997, *Codes for the representation of names of countries and their subdivisions - Part 1: Country codes*

— ISO 3166-2:1998, *Codes for the representation of names of countries and their subdivisions - Part 2: Country subdivision code*

AVA Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

— ITU-T Recommendation T.4, *Standardization of Group 3 facsimile terminals for document transmission*

— ITU-T Recommendation T.6, *Facsimile coding schemes and coding control functions for group 4 facsimile apparatus*

— ITU-T Recommendation T.42, *Continuous-tone colour representation method for facsimile*

— ITU-T Recommendation T.44 | ISO/IEC 16485:2000, *Information technology - Mixed Raster Content (MRC)*

— ITU-T Recommendation T.44 Amendment 1, *Accommodation of new Annex B*

— ITU-T Recommendation T.45, *Run-length colour encoding*

— ITU-T Recommendation T.81 | ISO/IEC 10918-1:1994, *Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines*

— ITU-T Recommendation T.82 | ISO/IEC 11544:1993, *Information technology - Coded representation of picture and audio information - Progressive bi-level image compression*

— ITU-T Recommendation T.83 | ISO/IEC 10918-2:1995, *Information technology - Digital compression and coding of continuous-tone still images: Compliance testing*

— ITU-T Recommendation T.84 | ISO/IEC 10918-3:1997, *Information technology - Digital compression and coding of continuous-tone still images: Extensions*

— ITU-T Recommendation T.84 Amendment 1 | ISO/IEC 10918-3:1997/Amd 1:1999, *Provisions to allow registration of new compression types and versions in the SPIFF header*

— ITU-T Recommendation T.86 | ISO/IEC 10918-4:1999, *Information technology - Digital compression and coding of continuous-tone still images: Registration of JPEG Profiles, SPIFF Profiles, SPIFF Tags, SPIFF colour Spaces, APPn Markers, SPIFF Compression types and Registration authorities (REGAUT)*

— ITU-T Recommendation T.87 | ISO/IEC 14495-1:2000, *Information technology - Lossless and near-lossless compression of continuous-tone still images - Baseline*

— ITU-T Recommendation T.88 | ISO/IEC 14492:2001, *Information technology - Lossy/lossless coding of bi-level images*

— ITU-T Recommendation T.89, *Application profiles for Recommendation T.88 - Lossy/lossless coding of bi-level images (JBIG2) for facsimile*

— ITU-T Recommendation T.800 | ISO/IEC 15444-1:2000, *Information technology - JPEG 2000 image coding system - Part 1: Core coding system*

— ITU-T Recommendation T.801 | ISO/IEC 15444-2, *Information technology - JPEG 2000 image coding system - Part 2: Extensions*

— ITU-T Recommendation T.803 | ISO/IEC 15444-4, *Information technology - JPEG 2000 image coding system - Part 4: Conformance testing*

2.2 Additional References

— IEEE Std. 754-1985 R1990, *IEEE Standard for Binary Floating-Point Arithmetic*

— IETF RFC 1766, *Tags for the Identification of Languages*, March 1995

— IETF RFC 2279, UTF–8, *A transformation format of ISO 10646*, January 1998

— ICC.1:1998-09, *International Color Consortium, File Format for Color Profiles*

— IEC 61966-2-1:1999-10, *Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB*

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

— IEC 61966-2-1/Amd.1: *Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB*

— W3C, *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation 6 October 2000, <http://www.w3.org/TR/REC-xml>

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ITU-T Rec T.800 | ISO/IEC 15444-1 Clause 3 and the following apply.

3.1 base colour: The colour of an object for which no image data is available.

3.2 BasePage: The original state of the page before it is rendered with layout objects.

3.3 box: A portion of the file format defined by a length and a unique box type. Boxes of some types may contain other boxes.

3.4 component: A two-dimensional array of samples.

3.5 compound image: An image that may contain scanned images, synthetic images or both, and that preferably requires a mix of continuous tone and bi-level compression methods.

3.6 file format: A codestream or codestreams and additional support and information not explicitly required for decoding of the codestream or codestreams. Examples of such support data include text fields providing security and historical information, data to support the placement of multiple codestreams within a given data file, and data to support exchange between platforms or conversions to other file formats.

3.7 fragment: A portion of the codestream for an image. Clause 5.2.6 describes fragment usage.

3.8 JP2 file: The name of a file in the file format described in ITU-T Rec T.800 | ISO/IEC 15444-1. Structurally, a JP2 file is a contiguous sequence of boxes.

3.9 JPM file: The name of a file in the file format described in this International Standard. A JPM file can contain one or more pages, composed from one or more layout objects, each of which is composed from at most two objects. Structurally, a JPM file is a contiguous sequence of boxes.

3.10 JPX file: The name of a file in the file format described in ITU-T Rec T.801 | ISO/IEC 15444-2. Structurally, a JPX file is a contiguous sequence of boxes.

3.11 layout object: An entity that comprises at most two paired objects or MRC layers.

3.12 main page collection: The main page collection contains all pages and page collections in a file.

3.13 mask object: An object that is used to select the samples of a corresponding image object that are to be imaged on a page.

3.14 metadata: Additional data associated with the image data beyond the image data.

3.15 MRC: Mixed Raster Content; a multi-layer imaging model described in ITU-T Recommendation T.44 | ISO/IEC 16485.

3.16 object: An image that is part of a layout object; an MRC layer.

3.17 page: The largest collection of layout objects that can be imaged independently of any other layout objects; a canvas or frame for imaging.

3.18 page collection: A collection of pages logically grouped together in a JPM file. Each page must be contained in at least one page collection**.**

3.19 primary page collection: A page collection which provides back and forward navigation in the main document associated with a page.

3.20 PageImage: The image created by rendering the BasePage with the layout objects. The $PageImage_k$ is the images created by rendering the BasePage with the first k layout objects.

3.21 profile: A subset of all possible field values in a file.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

3.22 superbox: A box that itself contains a contiguous sequence of boxes (and only a contiguous sequence of boxes).

4 Abbreviations

For the purposes of this International Standard, the following abbreviations apply. The abbreviations defined in ITU-T Rec T.800 | ISO/IEC 15444-1 Clause 4 also apply to this International Standard

CCITT: International Telegraph and Telephone Consultative Committee, now ITU-T

DPI: Dots per inch

IPR: Intellectual Property Rights

JP2: JPEG 2000 File Format defined in ITU-T Rec T.800 | ISO/IEC 15444-1

JPX: JPEG 2000 File Format defined in ITU-T Rec T.801 | ISO/IEC 15444-2; JPEG 2000 File Format E tended

JPM: JPEG 2000 File Format defined in this International Standard; JPEG 2000 File Format - Multi-layer

MRC: Mixed Raster Content

UUID: Universal Unique Identifier

5 General Description

The purpose of this clause is to give an overview of this International Standard. Terms defined in previous clauses in this International Standard will also be introduced. (Terms defined in Clause 3 and 4 in ITU-T Rec T.800 | ISO/IEC 15444-1 continue to apply in this International Standard.) Throughout this International Standard, text formatted as a NOTE in the following form is informative only:

NOTE - Informative text appears here.

This International Standard defines a file format for storing compound images using the JPEG 2000 file format family architecture. A compound image file contains multiple images, both contiguous tone and bi-level, together with composition models describing how the individual images are combined to generate the compound image. This International Standard is based on the multi-layer Mixed Raster Content (MRC) imaging model, defined in ITU-T T.44 | ISO/IEC 16485.

This International Standard defines a member of the JPEG 2000 file format family that enables the efficient processing, interchange and archiving of raster-oriented pages containing a mixture of multi-level and bi-level images. This efficiency is realized by representing the mixed-content image using multiple layers, determined by image type, and applying image specific encoding, spatial and colour resolution processing. A rasterized page may contain one or more image types, such as: multi-level continuous-tone or palettized (contone) content usually associated with naturally occurring images; bi-level detail associated with text and line-art; and multi-level colours associated with the text and line-art. This International Standard makes provisions for processing, interchange, and archiving of these image types in multiple layers and defines composition models which regenerate the desired image.

5.1 Mixed Raster Content Model

A file that conforms with this International Standard contains one or more pages. The *PageImage* associated with a page is generated by combining the page's layout objects with the *BasePage*.

The *BasePage* is the initial *PageImage* before any layout objects have been rendered. The *BasePage* has the same width and height as the page and is either transparent or filled with a single colour. *BasePage* is defined in 5.2.3.1. The Layout Objects are applied sequentially, in an order defined by the Layout Object Identifier field

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

in the Layout Object Header boxes, to the *BasePage* to create the final *PageImage*. The Layout Object with Layout Object Identifier value of 0 is the page thumbnail and is not used in creating the *PageImage*.

Associated with each Layout Object is a mask *M* and an image *I*. The mask *M* is an opacity image and has only one component; *I* can be greyscale or colour, with one or more components. *M* and *I* are defined in Clause 5.2.4. Both *M* and *I* have the same width and height as the page.

The following equations show the model for combining the *BasePage* and a sequence of *n* layout objects with non-zero Layout Object Identifier values to create the final *PageImage*. We will use the notation *N[c][x,y]* to refer to the sample value at position *(x,y)* in component *c* of an image *N*.

$$\mathrm{PageImage}_0[c][x,y] = \mathrm{BasePage}[c][x,y] \tag{1}$$

$$\mathrm{PageImage}_m[c][x,y] = \frac{(s_m - M_m[0][x,y]) \times \mathrm{PageImage}_{m-1}[c][x,y] + M_m[0][x,y] \times I_m[c][x,y]}{S_m} \qquad m = 1, \ldots, n \tag{2}$$

$$\mathrm{PageImage}[c][x,y] = \mathrm{PageImage}_n[c][x,y] \tag{3}$$

where $M_m[c][x,y]$ and $I_m[c][x,y]$ are the image sample values of component *c* at position *(x,y)* of the mth Layout Object's mask *M* and image *I* respectively, and $s_m = 2^{\text{bit depth of } M_m} - 1$. *M*, *I* and *BasePage* are defined in 5.2.4 and 5.2.3.1. *PageImage*[c][x,y] is the final page image, obtained after combining all *n* layout objects associated with the page.

NOTE - Figure 1 shows a simple example of a *PageImage* constructed from a *BasePage* with a single solid colour and two Layout Objects. Note that white in the masks M_0 and M_1 denotes a value of 0.

NOTE - In a JPM file, the mask and image objects in a layout object typically have different spatial resolutions. Therefore, they must be scaled to the same resolution and to the page resolution before they are combined to create a page image according to Equations 1-3. In a JPM file, the size of the mask, image and page are specified in page grid units, but their resolutions may not be specified. The scaling of the mask and image is specified by a scale factor. The method of scaling is not specified in this International Standard, although informative guidelines for scaling of the mask and image are provided. The page image would then be scaled to the resolution of the output device for rendering on a display or printer. As described here, there would be two separate scaling operations: in the first, the mask and image of successive layout objects are scaled to a common resolution and combined on the page; in the second, the resulting page image is scaled to the device resolution. In practice, these two scaling operations are often combined into a single, device-dependent and implementation-specific scaling operation.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

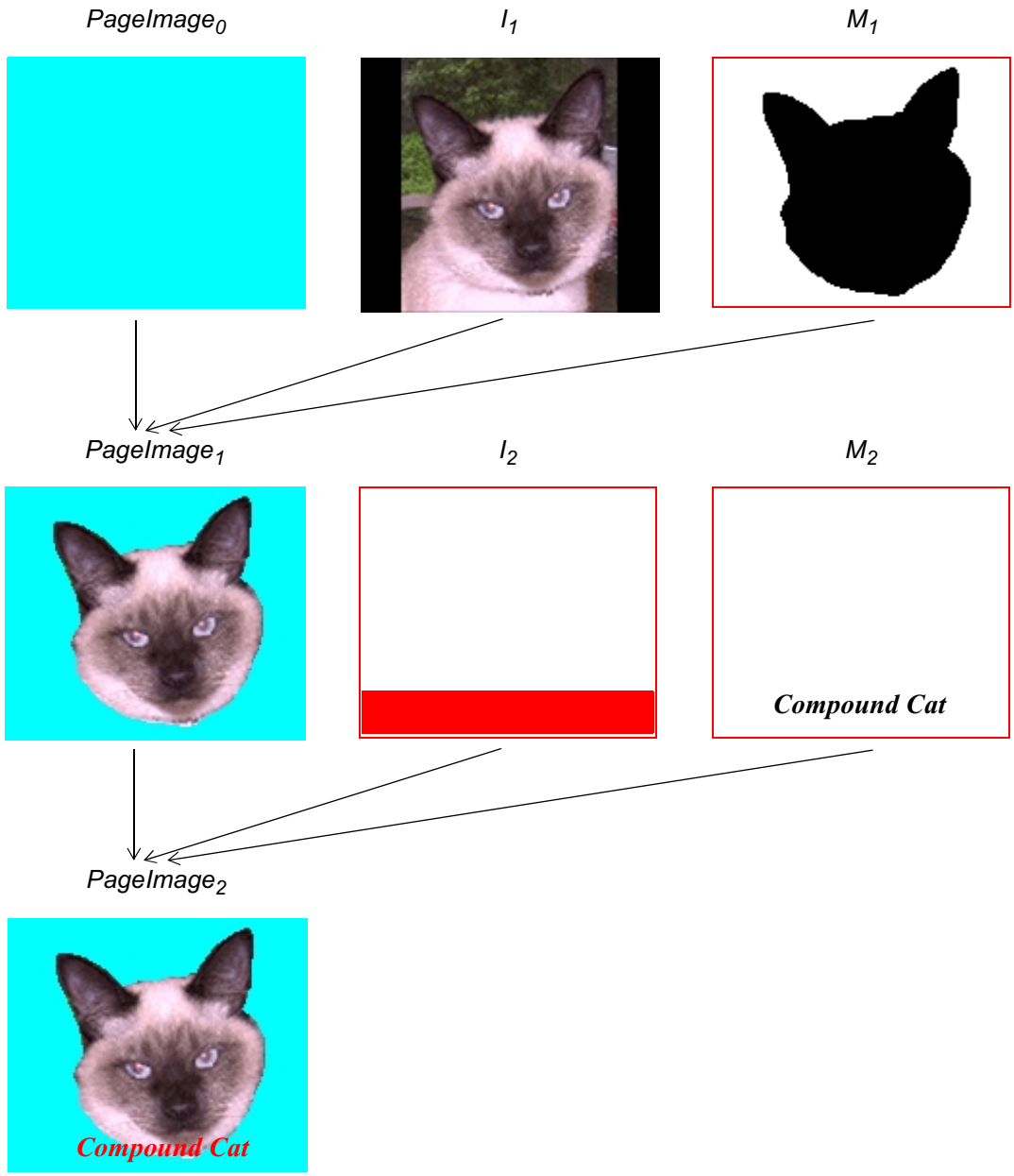

Figure 1 — Example compound document (Informative)

5.2 File Elements and Structure

The files that conform to the format defined in this International Standard are called JPM files. At its core, a JPM file is a sequence of pages, where each page in turn is a sequence of layout objects. A layout object normally consists of a mask object and an image object. Mask and image objects are composited to build up the final page image according to equations 1-3. The key elements or boxes of a JPM file are: page collections, page, layout object, and object. An object points to its image data directly via a Contiguous Codestream box or indirectly via a Fragment Table box. Like all members of the JPEG 2000 file format family, a JPM file begins with a JPEG 2000 Signature Box and a File Type Box. A Compound Image Header Box, containing general information about the file, is then followed by Page Collection, Page, Fragment Table, Contiguous Codestream, Media Data and general metadata boxes. Refer to Annex B for constraints on the location of boxes in a JPM file.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

This list illustrates the hierarchical relationship between the key elements in a JPM file. A particular order of these boxes is not implied. Full details of all the boxes may be found in Annex B.

- JPEG 2000 Signature
- File Type
- Compound Image Header
- Page Collection
- Page
  - Layout Object
    - Mask Object
    - Image Object
- Fragment Table
- Codestream Fragment
- Contiguous Codestream

5.2.1 Page Collections

A JPM compatible file consists of a sequence of pages, each represented by a Page box which occurs at the top level of the file and each of which can be rendered independently of any other page. Page collections are used to logically group pages in a JPM file. Page collections can be logically nested, so that a page collection can itself consist of one or more page collections and/or one or more pages. Page collections referred to from other page collections are called subsidiary page collections. All pages in a JPM file must be pointed to by at least one page collection.

A page can be said to be contained in a page collection, but this does not mean that the Page box is located within the Page Collection box. It is not. Page boxes and Page Collection boxes both occur at the top level of the file and are not contained within other boxes.

A JPM file contains one page collection known as the main page collection which is used to locate all pages of the file. Any additional page collections in a JPM file are logically nested within the main page collection (see Clause 5.2.2.3). A Page Collection box contains an optional Label box, optional metadata (XML and/or UUID) boxes, and a Page Table box that contains the locations of the pages and page collections belonging to the page collection.

It is recommended that optimized files have the main page collection located near the beginning of the file. While Page boxes and Page Collection boxes occur at the top level of the file, they may be located in external files. This case may be viewed as equivalent to being at the top level of the file.

Each page in a JPM file has a primary page collection. The purpose of a primary page collection is to enable navigation in the primary document to which the page belongs using the sequential order within the primary page collection, thus providing support for previous page and next page commands.

Every Page Collection box contains a Page Table box. The Page Table box entries point to the locations of the Page and Page Collection boxes within the page collection. A flag for each entry specifies whether the location is that of a Page box or Page Collection box, as well as indicating whether the box contains a thumbnail or metadata.

By walking the tree of pages and logically nested page collections in a page's primary page collection, all pages in the page's primary document can be reached. Every page (with the exception of a self-contained JPM file containing only a single page) has a Primary Page Collection Locator or PPCLoc box. This box points to the primary page collection of the page and provides an index, PIx, into that page collection's page table where the page is referenced. Then the next page and previous page can be found by walking one page forward or backward from the current page.

NOTE - Multiple page collections can exist in a JPM file. Some may have functions other than basic navigation. A table of figures could point to those pages containing figures. A section table or chapter table might point to only the first pages of sections or chapters. Any page collections of this sort must be auxiliary page collections, since they provide redundant pointers to pages and page collections and are sparse rather than comprehensive (see Clause 5.2.2.3).

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

NOTE - Figure 3 illustrates a logical grouping of page collections PC and pages P in a JPM file. PCa is the main page collection and the primary page collection for pages P0, P8 and P9 and page collections PCb, PCc and PCe. In a JPM file, the Page Table box of the Page Collection box for PCa would reference the Page boxes for P0, P8 and P9, and the Page Collection boxes for PCb, PCc and PCe. The Primary Page Collection Locator boxes in these Page and Page Collection boxes would reference the Page Collection box for PCa.

PCb is a subsidiary page collection of PCa and the primary page collection for pages P1, P6 and P7 and for page collection PCd. PCd is the primary page collection for pages P2, P3, P4 and P5. PCc is the primary page collection for pages P10 and P11.

PCe is an auxiliary page collection, which references page collection PCc and page P5.

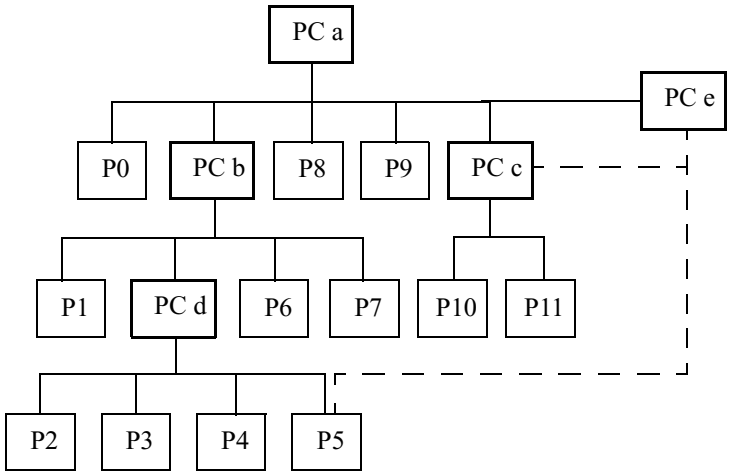

Figure 2 — Example of page collections and pages (Informative)

5.2.2.1 Main Page Collection

Each JPM file has one main page collection. The purpose of the file's main page collection is to comprehensively list all pages (and subsidiary page collections) within a JPM file. This allows random seeking to any of the pages in the file.

The Page boxes for these pages occur at the top level of the file format, as do Media Data boxes containing fragments of codestreams. Since these Media Data boxes may be large and since they would likely be interspersed with Page boxes, the Page boxes might be widely separated in the file. In a JPM file containing a client copy of several browsed pages of a server's JPM file, each successively viewed page and some portion of its codestream fragments would be appended in turn to the bottom of the file. An update to the main page collection allows each of these new pages to be located.

The main page collection comprehensively references all pages and page collections by means of a hierarchical arrangement. Some pages may be referenced from a page collection beneath the main page collection, but all pages and page collections are part of the tree structure of the main page collection.

A JPM file optimized for browsing would have the main page collection near the front of the file. On the other hand, a client copy created during a browsing session would likely have the main page collection appended to the end of the file each time it is modified. The old main page collection's Page Collection box could then be left in place but have its box type changed from "pcol" to "free". A later garbage collection step could delete these Free boxes. Because of cases like this, the file format has a pointer to the main page collection included in the Compound Image Header box near the top of the file. This makes it easy to locate the main Page Collection box.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

5.2.2.2 Primary Page Collection

Each page or page collection has a primary page collection. By way of distinction, each JPM file has a main page collection. A primary page collection is a property of a page or page collection, not a property of a JPM file.

The primary page collection of a page is the page collection where a JPM reading application would find the "next page" and "previous page" for a current page.

A Primary Page Collection Locator or PPCLoc box must appear in all Page boxes and all Page Collection boxes, with the one exception detailed in the next paragraph. The PPCLoc box provides a pointer to the primary page collection of the page or page collection. This backward pointer enables a comprehensive tree walk to find all the pages and page collections in the file.

The PPCLoc box is optional only in the case of a single-page, self-contained JPM file (i.e. one which has no external references). In this case, it appears that the page has no primary page collection, but in fact the file's main page collection functions as the page's primary page collection.

The primary page collection of a page or page collection may not be in the local JPM file in which that Page box or Page Collection box is located. The local file may, for example, have three single pages, each of which has been copied from one of three different remote files. Keeping the original primary page collection pointers for these pages pointing to the original remote files allows a user to keep browsing the original source document via next page and previous page commands.

The main page collection for a file may be a copy of a subsidiary page collection on a remote server, for instance, in which case it would have the parent page collection on that remote server as its primary page collection. When the main page collection does not have a primary page collection, then the main Page Collection Box shall not contain a PPCLoc box.

5.2.2.3 Auxiliary Page Collections

Auxiliary page collections are page collections which provide redundant pointers to pages already pointed to in the logical tree structure of the main page collection. Examples of auxiliary page collections could include a List of Figures or a List of Tables. Auxiliary page collections still appear in the logical tree structure of the main page collection so that they can be located, but a flag is used to mark them as auxiliary. This way, a decoding application knows they are not to be used to perform a comprehensive tree walk through all the pages referenced in the main page collection.

Auxiliary page collections appear in the main page collection to assist an application in locating them within the file, but they are not part of the logical tree that is walked to comprehensively locate all pages. They instead provide a redundant means of reaching selected pages and thus should be ignored when trying to determine the natural page order of the file. Auxiliary page collections can appear down in the hierarchy of the main page collection; they need not occur at the top level.

Auxiliary page collections should be labeled by means of a Label box in order to be useful to a receiving application. If they are labeled, then a decoding application could present the label to the end user and offer such options as "next page in List of Figures" and "previous page in List of Figures."

As an example, if page 17 appears in the "List of Figures" page collection, the decoding application would return to that page collection to get the next or previous page containing a figure, but would return to the primary page collection for page 17 (by means of the PPCLoc box in page 17's Page box) to find the next page or previous page.

5.2.3 Pages

A page in a JPM file is represented by a Page box, a superbox that consists of a Page Header box, containing general information about the page, a Page Collection Locator box, containing the location of the page's primary page collection, an optional Base Colour box, which describes the base page colour, optional Metadata

AVA Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

boxes, and Layout Object boxes, one for each layout object on the page. Full details of these boxes may be found in Annex B.2.

Pages occur at the top level of the file format. This means incremental updates to the file may be accomplished by simply appending new pages to the end of the file. Page boxes and boxes containing codestreams or portions of codestreams may be intermixed in the file.

With the exception of document thumbnails, each image in a JPM files is logically associated with at least one page.

5.2.3.1 Base Page

The *BasePage* is the initial *PageImage* before any layout objects have been rendered. Let *page_width* and *page_height* be the width and height of the page respectively, as signalled in the Page Header box.

Let *spc* be the colourspace in which the *I* images are to be combined to generate a *PageImage*. *BasePage* is an image with dimensions *page_width* and *page_height* and colourspace *spc*.

If the PColour field of Page Header box is 0 then the *BasePage* is transparent and contains the sample values of any underlying image, converted to the colourspace *spc*.

If the PColour field of the Page Header box is 1 then every *BasePage* sample contains the representation of *white* in the colourspace *spc*.

If the PColour field of the Page Header box is 2 then every *BasePage* sample contains the representation of *black* in the colourspace *spc*.

If the PColour field of the Page Header box is 255 then there is a mandatory Base Colour box within the Page box and every *BasePage* sample contains the *spc* representation of the colour indicated by the Base Colour box.

5.2.4 Layout Objects

Within a Page Box, there are as many Layout Object boxes as there are layout objects on the page. A Layout Object box is a superbox that consists of a Layout Object Header box, containing general information about the layout object, optional boxes containing metadata associated with the layout object, and one or two Object boxes - either an image Object box and/or mask Object box, or a combined image/mask Object box.

Each Object box contains an Object Header box identifying whether the object represents an image, a mask or a combined image/mask, specifying the location of any codestream associated with the object and containing positioning information for the object.

An image object typically has a codestream associated with it, but may not, in which case the NoCodestream field in the image Object header is set to 1. An image object may also have an associated constant colour or image base colour. If an image object does not have an associated codestream, then it must have a defined image base colour. A mask object or image/mask object, if present, must have a codestream associated with it and have NoCodestream=0.

If an object has an associated codestream then the Object Header box is followed by an optional Object Scale box and a JP2 Header box containing boxes describing the object data: an Image Header, Colour Specification, optional Bits Per Component, Palette, Component Mapping and Channel Definition boxes.

Associated with each layout object is a single component opacity mask *M* and an image *I*, each with the same width and height as the containing page.

The *I* images for the layout objects to be combined to generate a *PageImage* must all use the same colourspace and bit-depth. The colourspace to be used for the *I* images of layout objects may be decided by the implementor and is not defined by this International Standard.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

The general methods for generating the mask *M* and the image *I* associated with a layout object are described in 5.2.4.2 and 5.2.4.4.

5.2.4.1 describes the special case of generating a mask *M* and an image *I* from a JBIG 2 codestream with an associated ITU-T Rec. 45 encoding of colour tags.

5.2.4.1 Colour Tagged JBIG 2 Layout Objects

If the Layout Object box contains an image Object box with a compression type of ITU-T Rec. T.45 (Run Length) coding then it must also contain a mask Object box with a compression type of ITU-T Rec. T.88 (JBIG 2), and both must have the NoCodestream field in the Object Header boxes set to 0. In addition, the following fields must be the same for the image and mask Objects:

— the OVoff and OHoff fields in the Object Header boxes;

— the VRN, VRD, HRN, VRD fields in the Scale boxes;

— the HEIGHT and WIDTH fields in the Image Header box in the JP2 Header boxes.

In 5.2.4.3.1, *m_orig* is the image obtained by decompressing the JBIG 2 codestream associated with the mask Object box.

In 5.2.4.4.1, *i_orig* is the image obtained by applying the colour tags associated with the image Object box to the symbol occurrences in the JBIG 2 mask codestream as described in ITU-T Rec. T.45.

5.2.4.2 Mask *M* for a Layout Object

If the Layout Object box does not contain a mask Object box or a combined image/mask Object box, then the mask *M* for the layout object is defined to have a bit-depth of 1 and to have value 1 at all sample locations, i.e. *M[0][x,y] = 1*.

Sub-clauses 5.2.4.3.1 - 5.2.4.3.5 define *M* when a codestream is associated with a mask Object box or a combined image/mask Object box in the Layout Object box.

The first stage in generating *M*, described in 5.2.4.3.1, is to decode the mask object, converting to a bit-depth of 8 if the mask has a lower bit-depth, and ensuring that the samples use a “min is white” representation.

The second stage, described in 5.2.4.3.2, is to scale the mask.

The third stage, described in 5.2.4.3.3, is to clip the mask from the top and from the left.

The fourth stage, described in 5.2.4.3.4, is to position the mask on the page.

The fifth and final stage, described in 5.2.4.3.5, is to clip the mask to the layout window.

These stages are described individually for clarity and an efficient JPM decoder may be able to combine one or more of these stages.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

Figure 3 illustrates the five stages for an example mask. Note that white in the images denotes a value of 0.

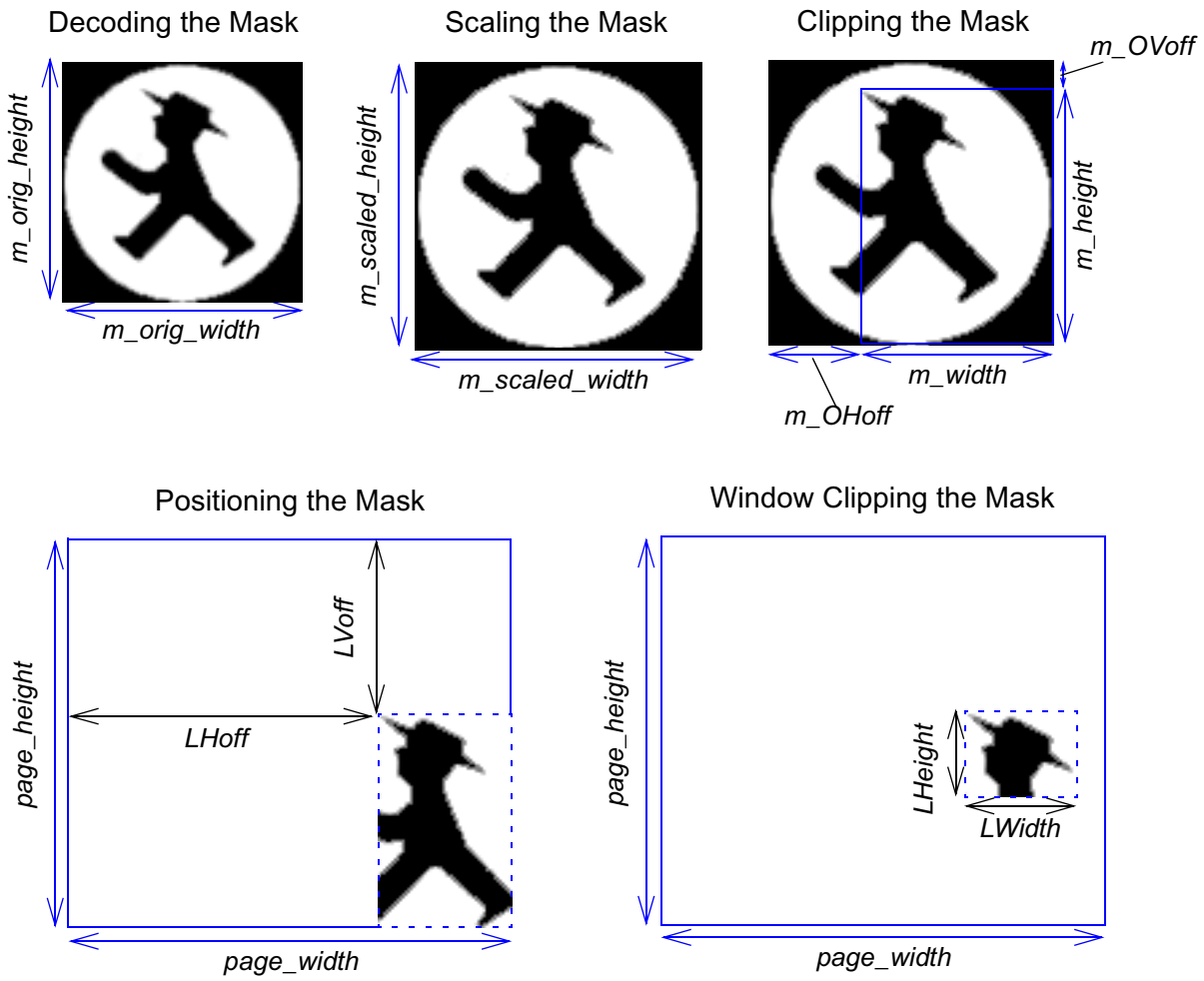

Figure 3 — Example of generating M (Informative)

5.2.4.3.1 Decoding the Mask

If a mask Object box is used then let *m_orig* be the image obtained by decompressing the associated codestream, otherwise let *m_orig* be the image containing only the opacity component of the image obtained by decompressing the associated codestream. Let *m_orig_width* and *m_orig_height* be the width and height of *m_orig* and let *b* be the bit-depth of *m_orig*.

If *b* is 1 then, unless indicated otherwise by the codestream, samples of value 0 are assumed to represent white while samples of value 1 are assumed to represent black.

If *b* is greater than 1 then, unless indicated otherwise by the codestream, samples of value 0 are assumed to represent black while samples of value $2^b - 1$ are assumed to represent white.

Let *grey* be an image with bit-depth *g* = *max(8, b)* and the same width and height as *m_orig*. *grey*, defined as follows:

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

$$\text{grey}[0][x, y] = \begin{cases} (2^b - 1 - \text{m_orig}[0][x,y]) \times \dfrac{2^g - 1}{2^b - 1} & \text{if 0 represents black in m_orig} \\ \text{m_orig}[0][x,y] \times \dfrac{2^g - 1}{2^b - 1} & \text{otherwise} \end{cases} \quad 4)$$

5.2.4.3.2 Scaling the Mask

If an Object Scale box is contained in the mask Object box or combined image/mask Object box then let *m_VRN*, *m_VRD*, *m_HRN* and *m_HRD* be the scaling factors contained in this box, otherwise let *m_VRN*, *m_VRD*, *m_HRN* and *m_HRD* all be 1. Let *m_scaled* be the result of scaling *grey* vertically by the ratio $\frac{\text{m_VRN}}{\text{m_VRD}}$ and horizontally by the ratio $\frac{\text{m_HRN}}{\text{m_HRD}}$, generating an image with bit-depth *g* and dimensions:

$$\text{m_scaled_width} = \left\lfloor \frac{\text{m_HRN}}{\text{m_HRD}} \times \text{m_orig_width} \right\rfloor \text{ and } \text{m_scaled_height} = \left\lfloor \frac{\text{m_VRN}}{\text{m_VRD}} \times \text{m_orig_height} \right\rfloor . \quad (5)$$

NOTE - The method for scaling a binary or grey mask up or down is implementation specific. For example, the scaling may use nearest neighbour or bilinear interpolation.

5.2.4.3.3 Clipping the Mask

Let *m_OVoff* and *m_OHoff* be the vertical and horizontal offsets in the mask Object box or combined image/ mask Object box.

Let *m_clipped* be an image with bit-depth *g*, width *m_width = max(0, m_scaled_width - m_OHoff)* and *m_height = max(0, m_scaled_height - m_OVoff)*. *m_clipped* is defined as follows:

$$\text{m_clipped}[0][x,y] = \text{m_scaled}[0][x + \text{m_OHoff}, y + \text{m_OVoff}] \quad (6)$$

5.2.4.3.4 Positioning the Mask

Let *page_width* and *page_height* be the width of the page containing the layout object, as signalled in the Page Header box. Let *LVoff* and *LHoff* be the layout vertical and horizontal offsets in the Layout Object Header box.

Let *m_pos* be an image with a width of *page_width* and height of *page_height*, defined as follows:

$$\text{m_pos}[0][x, y] = \begin{cases} \text{m_clipped}[0][x - \text{LHoff}, y - \text{LVoff}] & (0 \le (x - \text{LHoff}) < \text{m_width}) \text{ and } (0 \le (y - \text{LVoff}) < \text{m_height}) \\ 2^g - 1 & \text{otherwise} \end{cases}$$

(7)

5.2.4.3.5 Window Clipping the Mask

Let *M* be an image with a width of *page_width* and height of *page height*. Let *LWidth* and *LHeight* be the layout object width and layout object height as specified in the Layout Object Header box. *M* is defined as follows:

$$M[0][x, y] = \begin{cases} \text{m_pos}[0][x, y] & (0 \le (x - \text{LHoff}) < \text{LWidth}) \text{ and } (0 \le (y - \text{LVoff}) < \text{LHeight}) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

5.2.4.4 Image *I* for a Layout Object

Let *spc* be the colourspace in which the *I* images are to be combined to generate a *PageImage*.

If the Layout Object box contains an image in either an image Object box or a combined image/mask Object box and also contains a Base Colour box, then let *base* be the representation in colourspace *spc* of the colour indicated by the Base Colour box; otherwise let *base* be the representation in colourspace *spc* of the colour *black*.

If the Layout Object box does not contain an image Object box or a combined image/mask Object box, then the image *I* for the layout object is defined to be an image in colourspace *spc* where every sample has the colour *base*.

If the Layout Object does contain an image Object box or a combined image/mask Object box, but with no associated codestream, then the image *I* for the layout object is also defined to be an image in colourspace *spc* where every sample has the colour *base*.

Sub-clauses 5.2.4.4.1 - 5.2.4.4.5 define *I* when a codestream is associated with a image Object box or a combined image/mask Object box in the Layout Object box.

The first stage in generating *I*, described in 5.2.4.4.1, is to decode the image object and convert the image to the *spc* colourspace.

The second stage, described in 5.2.4.4.2, is to scale the image.

The third stage, described in 5.2.4.4.3, is to clip the image from the top and from the left.

The fourth stage, described in 5.2.4.4.4, is to position the image on the page.

The fifth and final stage, described in 5.2.4.4.5, is to clip the image to the layout window.

These stages, as with those to generate the mask *M*, are described individually for clarity and an efficient JPM decoder may be able to combine one or more of these stages

5.2.4.4.1 Decoding the Image

If an image Object box is used then let *i_orig* be the image obtained by decompressing the associated codestream, otherwise let *i_orig* be the image containing all but the last component of the image obtained by decompressing the associated codestream. Let *i_orig_width* and *i_orig_height* be the width and height of *i_orig*.

Let *conv* be the image obtained by converting *orig* to the colourspace *spc*.

5.2.4.4.2 Scaling the Image

If an Object Scale box is contained in the image Object box or combined image/mask Object box then let *i_VRN*, *i_VRD*, *i_HRN* and *i_HRD* be the scaling factors contained in this box, otherwise let *i_VRN*, *i_VRD*, *i_HRN* and *i_HRD* all be 1. Let *i_scaled* be the result of scaling *conv* vertically by the ratio $\frac{\text{i_VRN}}{\text{i_VRD}}$ and horizontally by the ratio $\frac{\text{i_HRN}}{\text{i_HRD}}$, generating an image with dimensions:

$$\text{i_scaled_width} = \left\lfloor \frac{\text{i_HRN}}{\text{i_HRD}} \times \text{i_orig_width} \right\rfloor \text{ and } \text{i_scaled_height} = \left\lfloor \frac{\text{i_VRN}}{\text{i_VRD}} \times \text{i_orig_height} \right\rfloor . \quad (9)$$

NOTE - An image may be scaled up using bilinear interpolation or, if it's a palette image, using nearest neighbour

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

interpolation. An image may be scaled down using averaging followed by subsampling, or if it's a palette image, using subsampling. If the image is compressed using JPEG 2000, an implementation may also choose to use the progressive-by-resolution decompression feature to obtain a power-of-two reduced resolution image for scaling.

5.2.4.4.3 Clipping the Image

Let *i_OVoff* and *i_OHoff* be the vertical and horizontal offsets in the image Object box or combined image/mask Object box.

Let *i_clipped* be an image with bit-depth *g*, width *i_width = max(0, i_scaled_width - i_OHoff)* and *i_height = max(0, i_scaled_height - i_OVoff)*. *i_clipped* is defined as follows:

$$\text{i_clipped}[0][x,y] = \text{i_scaled}[0][x + \text{i_OHoff},\ y + \text{i_OVoff}] \tag{10}$$

5.2.4.4.4 Positioning the Image

Let *page_width* and *page_height* be the width of the page containing the layout object, as signalled in the Page Header box. Let *LVoff* and *LHoff* be the layout vertical and horizontal offsets in the Layout Object Header box.

Let *i_pos* be an image with a width of *page_width* and height of *page_height*, defined as follows:

$$\text{i_pos}[0][x,y] = \begin{cases} \text{i_clipped}[0][x - \text{LHoff},\ y - \text{LVoff}] & (0 \le (x - \text{LHoff}) < \text{i_width})\ \text{AND}\ (0 \le (y - \text{LVoff}) < \text{i_height}) \\ \text{base}[c] & \text{otherwise} \end{cases} \tag{11}$$

5.2.4.4.5 Window Clipping the Image

Let *I* be an image with a width of *page_width* and height of *page height*. Let *LWidth* and *LHeight* be the layout object width and layout object height as specified in the Layout Object Header box. *I* is defined as follows:

$$I[c][x,y] = \begin{cases} \text{i_pos}[c][x,y] & (0 \le (x - \text{LHoff}) < \text{LWidth})\ \text{and}\ (0 \le (y - \text{LVoff}) < \text{LHeight}) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

5.2.5 Thumbnails

Thumbnails are an optional feature which allow an overview of a document or a page to be presented to an end user. There are two types of thumbnails: document thumbnails and page thumbnails.

A document thumbnail offers an overview representation of an entire document, much as the cover or spine of a book serves to do. It may simply be a repetition of the page thumbnail for the title page. Document thumbnails may be JP2 compatible or not.

A JPM file with a document thumbnail is created by adding a JP2 Header box for the thumbnail image immediately after the File Type box. The codestream for the thumbnail image must be contained in the first Contiguous Codestream box following the JP2 Header box.

NOTE - A JP2-compatible document thumbnail can be used in the construction of a JPM file that also conforms to the JP2 file format defined in ITU-T Rec T.800 | ISO/IEC 15444-1. A JP2 compatible document thumbnail must use JPEG 2000 as the coder and appears to a JP2 reader to be a valid JP2 file. The JP2 compatibility flag is set in the File Type box by adding the string "jp2\040" to the compatibility string. A JP2 reader shall then look for a JP2 Header box and jump over any following boxes until it finds a Contiguous Codestream box. It would then access the image represented by the data in that box. When a document thumbnail is present, the file would contain a JP2 Header Box, and a Contiguous Codestream Box near the beginning of the file, somewhere after the Compound Image Header Box and before the first Page Box, usually before the Page Collection Box.

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

A page thumbnail is simply the layout object in a page with layout object identifier 0. A page thumbnail layout object must be the first layout object in a page. Its presence is signalled by a flag in the page table which indicates that a given page pointed to from the page table contains a page thumbnail. Page thumbnails are optional.

NOTE - Page thumbnails allow an overview of each individual page to be displayed. This can be used during navigation, for example to render a small icon near each node of a tree-like table of contents. It can also be used to allow the display of a more useful initial image onto a new page than simply the page's background colour while waiting for additional content to arrive.

5.2.6 Contiguous and Fragmented Codestreams

The codestream data for the objects of a page may either be a single contiguous codestream, or be composed of one or more codestream fragments. A codestream consisting of fragments is accessed via a Fragment Table box. A Fragment List box lists the locations in the file, or external pointers to another file, for the codestream fragments. Each codestream fragment is contained within a Media Data box. If the codestream is not fragmented then the object shall contain the location of a Contiguous Codestream box containing the codestream data. Full details of these boxes may be found in Annex B.5.

The fragment table mechanism permits any codestream to be broken up into an ordered list of fragments, each pointed to by its own offset and data reference. A data reference indexes into the data reference table and produces a string listing the filename or URI of an external file where the fragment is to be found. If the data reference is 0, the fragment is located in the current file. This mechanism allows fragments to be placed in an optimal position to support streaming of the data to a client application from a server, or to allow progressive refinement of multiple codestreams simultaneously.

Fragments may start or stop at any byte boundary of the codestream. The length of each fragment is specified by the length parameter retrieved from the fragment table. If a fragment does not end at a point in the codestream where a full code block or full raster line can be decoded, the decoding application holds this additional information until the next fragment has been retrieved. The next fragment of the byte stream is then appended to the residual data from the prior fragment and then decoding is resumed. Codestreams having any compression type may be decoded in this way.

Fragment table entries always point to codestreams. All header information required to decode the codestream, such as width and height, compression type or bit depth is stored in fields in the Object box.

5.2.7 Shared Data

A file can contain multiple boxes or fragments containing identical data. File size can be reduced by sharing these boxes, including by reference a box or fragment that occurs in one part of a JPM file in a different part of the same file or in a different JPM file. The two mechanisms for sharing boxes in a JPM file are Shared-Data Reference box and Cross-Reference box. Wherever a Shared-Data Reference or a Cross-Reference box occurs in the file, the data it references is treated as if it exists at the point in the file where the reference is located. The reference can be by means of either an identifier, in the case of a Shared-Data Reference box (Annex B.1.8), or a data reference including an offset and length, in the case of a Cross-Reference box (Annex B.6.4).

A Shared-Data Reference box in a file uses an identifier that is defined in the same file; therefore this mechanism shall only be used to share data within a file. The Cross-Reference box uses a data reference and therefore may be used to share data across files. In the case of a reference to an offset from the beginning of the file and length, the offset applies before any substitution implied by the reference occurs. A shared data substitution would not be understood by a JP2-only reader, so that a JP2-conforming JPM file must include directly any box whose presence in the file is necessary for JP2 conformance.

5.2.8 Metadata

This International Standard allows metadata to be attached to any structural element of a JPM file, including the file or document itself, page collections, pages, layout objects and objects. This is indicated in the box

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003

definitions in Annex B by the inclusion of a “metadata” box in the data field of the container box. Whenever there is a reference to a metadata in a box definition, it is understood that the box may contain an XML box or a UUID box with metadata. These “metadata” boxes are defined in Annex C.

5.3 JPM Use Scenarios (Informative)

5.3.1 Self-Contained Files and Files with External References

Clause 5.2 outlines the structure of a JPM file. In a local file, typically all image content and parameter boxes are contained in a single file for ease of tracking and maintenance, and because local storage is available and local bandwidths are high.

The multiple layout object model is a method of representing and constructing document images, which allows access paradigms based on the multiple objects in addition to providing compression advantages due to applying the optimal compression method to each object. In addition, document images have an internal structure that implies a reading order, unlike photographic images, so that breaking them into layout objects can bring streaming advantages where only the small area of the page (a column, say, or a paragraph) that is of interest is brought over and refined.

The fragment table and cross reference features of the JPEG 2000 family of file formats provide enormous flexibility in the construction of files, whose contents are distributed across multiple files, some of which may be located remotely on networks such as the Internet. This sub-clause examines some of the file organisations enabled by these features.

Useful JPM files tend to have structural information boxes at the top and large codestream objects either at the bottom of the file or remotely located or both. This allows rapid parsing and seeking among the component parts of the file by first getting the necessary structural information and then seeking to the offsets specified in the local or remote files.

Structural information includes such information as Page Table boxes, Page Collection boxes, Label boxes and Fragment Table boxes.

Page Table boxes are a key structural feature. They refer to Page boxes via page references. These references contain an offset, length and data reference for the Page Box associated with each page in the complete file. Each of these data references may be zero, indicating a page reference to a Page box in the current file, or non-zero, indicating that the data reference value is to be used as an index into the Data Reference box. The Data Reference box contains an enumerated list of strings, each string being a file name or a URI pointing to an Internet file.

5.3.2 Files prepared for the Internet

JPM files prepared for use on the Internet will typically already have been decomposed into layers and perhaps also into separate layout objects. These preparation steps each give a measure of improved streaming performance in browsing remote JPM files.

Such files would likely have structural information placed near the top of the file, with codestreams occurring later. Codestreams and subsets of codestreams are accessible directly by seeking to a point in the file indicated by entries in the fragment table associated with that codestream.

5.3.3 Page Collection Example

The main page collection of a very large multi-volume set of books such as an encyclopedia should not point to each page individually; such a data structure would be too large to be brought over to a client before even the first page can be displayed. Instead, the main page collection might serve as what amounts to a table of contents and point to each of the front matter pages individually, then to page collections that each cover entire volumes: Volume A, Volume B and so on. Each of these “volume” page collections would reference “section” page collections that would reference “article” page collections, and so on. Each of these page collections could have an appropriate label applied by means of Label boxes in their Page Collection boxes.

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 15444-6:2003

A search of metadata for a term occurring on a few of the pages returns a new “search results” JPM file containing a single main “search results” page collection pointing to a handful of pages. The first entry in this page collection box would point directly to the first search result page. This would actually point to one of the pages in the encyclopedia over on the server via an external reference. By navigating the “search results” page collection, the application can let the user go to next search hit and previous search hit. If the user then finds the article of interest and wants to continue reading onto a page back in the remote encyclopedia where no search hit occurred, they would then need a next page function as well.

The client software would go to the PPCLoc box found in that page's Page box. This entry has a pointer to the primary Page Collection box back in the main JPM file. This primary Page Collection box will then be used by the client software to first find the current page’s entry in that page collection by means of the PIx portion of the PPCLoc box. The PIx + 1 entry in that page collection would be the next page. By that means, the previous and next pages can be found.

5.3.4 Page Collections in a Client / Server Context

The JPM file format places Page boxes and Page Collection boxes at the file level (not within other superboxes) to assist with incremental browsing of pages. New pages or page collections drawn over from a server may simply be appended to the end of the local (incomplete) copy of the file.

In a client/server environment such as the web, it would be common for a user of a client application to browse only certain pages of a multi-page JPM file on a server. An initial link could point the client application to a specific page in the server’s JPM file. The client application would create a new local JPM file and could choose to continuously update it as portions of the server file are retrieved. This would ensure that the browser cache always contains a valid JPM file when browsing ceases.

This local file would have a main page collection. The main page collection would initially contain just one page. As the user begins fetching additional pages, these would be appended to the local file and the main page collection’s page table updated to point to each new local copy of a server page. At the end of browsing, a stand-alone file having no clear connection to the server file could exist on the client.

A better way of performing this browsing step would be to copy the server’s main page collection to the client prior to downloading any pages. This page collection would be fixed up to point to the server copies of the pages by creating a data reference to the server URL. Then, as the first page is brought down to the client, the main page collection is updated to point to the local page, while continuing to point to the server for all other pages.

5.3.5 Example of a Client/Server Interaction

In a client/server system, a user might want to browse a multi-page JPM file on a server, and to begin on a page in the middle of that remote file. Perhaps a search engine found a hit on the JPM file metadata associated with that page. The URI returned by the search engine would point to the JPM file and could contain a construction such as http://webimaging.org/test.jpm?page=page17. This would instruct the JPM decoder subsystem of the browser to abort the download of the full JPM file as soon as the key structural boxes at the top of the box have completed downloading. Then, byte serving protocols would be used to retrieve only the ranges of bytes needed to render the data in the Page box associated through the Label box to label “page17.” Each Object box in each Layout Object box of a given Page box may contain a fragment table reference to a Fragment Table Box. By retrieving only enough of each layout object’s Fragments to render a rough initial version of the page, a full rendering of all the layout objects can be deferred. The end user could in parallel begin moving the mouse over the browser’s display window to indicate which layout objects might better be refined next, based on their having been indicated of interest.

During this interactive browsing session, fragments are streaming over to the client copy of the JPM file in an order that is quite different than the order in the server copy. Page 17 comes first in the client copy, where page 1 was more likely first in the server copy. Similarly, the layout objects within page 17 will have a different order in the client version of the page, since the user’s mouse movements dictated their retrieval. The client version can nevertheless form a perfectly equivalent surrogate for the server version because the page tables and fragment tables in the client copy will have started out pointing entirely to fragments on the server. As

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 15444-6:2003