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)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

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*

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*

- 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*

— 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.

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

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 m^{th} 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.

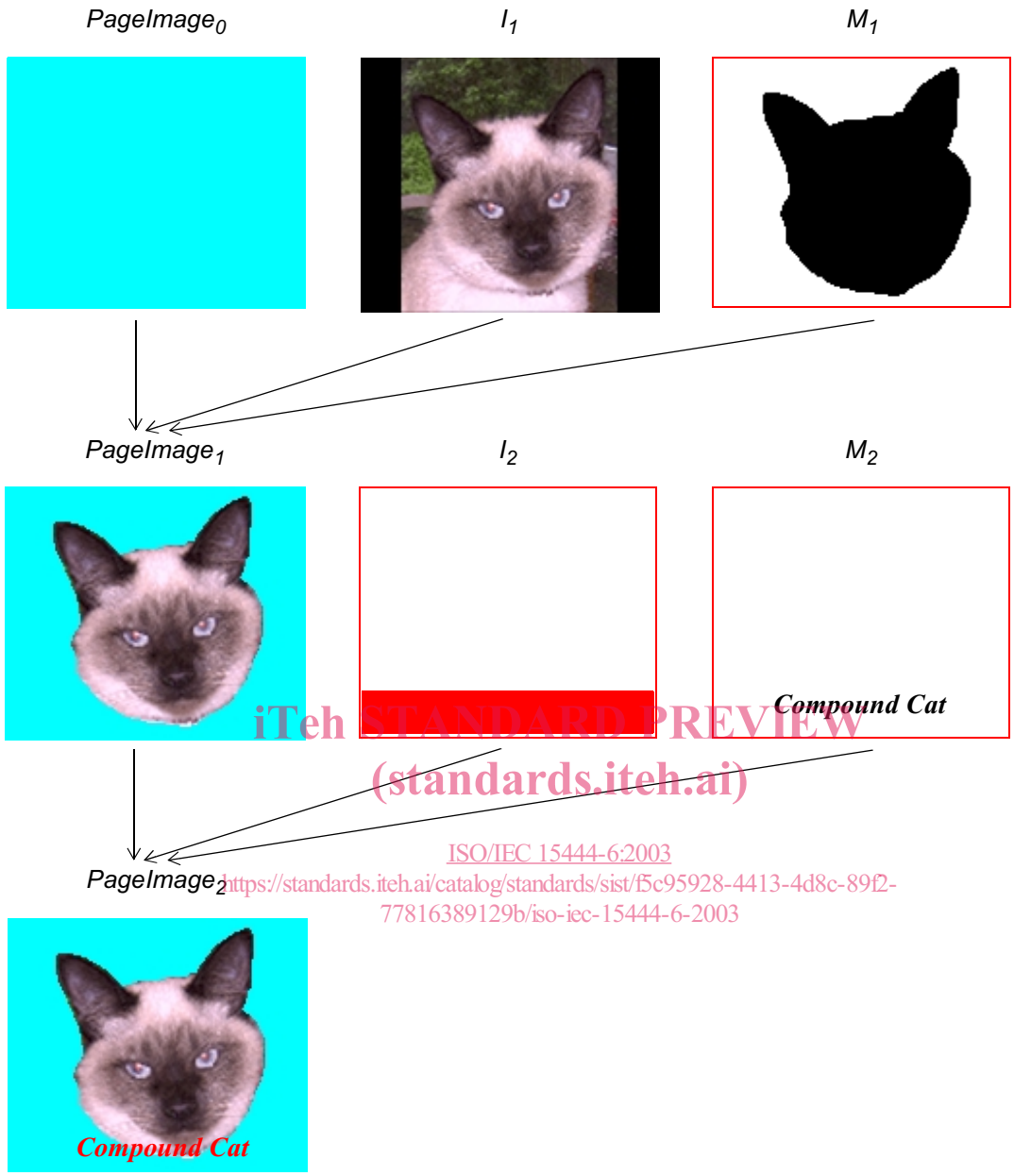

Figure 1 — Example compound document (Informative)

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.

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 location 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).

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.

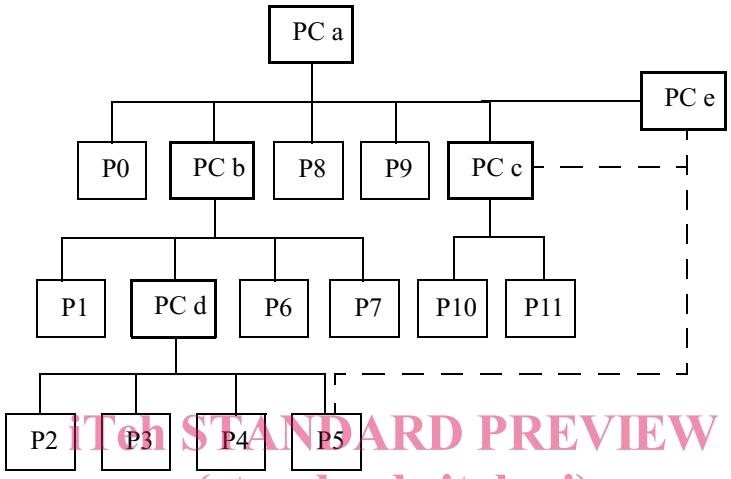

Figure 2 — Example of page collections and pages (Informative)

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.

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

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.

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.