International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.44
(01/2005)

SERIES T: TERMINALS FOR TELEMATIC SERVICES

Mixed Raster Content (MRC)

ITU-T Recommendation T.44

ITU-T

International Telecommunication Union

ITU-T Recommendation T.44

Mixed Raster Content (MRC)

Summary

This Recommendation specifies the technical features of Mixed Raster Content (MRC) imaging format that enables efficient processing, interchange and archiving of raster-oriented pages containing a mixture of multi-level and bi-level images. This efficiency is realized through segmentation of the image into multiple layers (planes), as 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 palletized colours (contone) usually associated with naturally occurring images; bi-level detail associated with text and line-art; multi-level colours associated with the text and line-art. This Recommendation makes provisions for processing, interchange, and archiving of these image types in multiple separate layers. Recombining the layers in a prescribed manner regenerates the original image.

This edition of T.44 integrates Amendment 1 (which added a new Annex B which defines provisions for sharing resources across pages, stripes and layers and provisions for using colour tags as a means of representing text colour, that play an essential role in the application of JBIG2 of ITU-T Rec. T.88) with a number of additional modifications to the main body and its annexes that give the definition of the necessary technical specifications to support YCC colour space (as described in ITU-T Rec. T.42). This will allow the YCC colour space to be deployed using almost the same method as that of the LAB colour space.

Source

ITU-T Recommendation T.44 was approved on 8 January 2005 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure.

<u>This text includes the typo correction introduced by Erratum 1 (9 June 2005)</u>

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Introduction and background

The Mixed Raster Content (MRC) Recommendation is a way of describing raster-oriented (scanned and/or rasterized synthetic images) documents with both bi-level (text and/or line-art) and multi-level (colour/continuous-tone) data within a page. The goal of this MRC Recommendation is to make exchange of raster-oriented mixed content colour documents among users with varied communication systems possible with higher speed, higher image quality and modest computing resources (memory, storage and processing power).

The dramatic increase in exchange of electronic documents has raised customer expectations and requirements for raster-oriented documents. Colour must be exchanged just as gracefully and efficiently as black & white (bi-level) and quickly reproduce a copy of the original at the best possible image quality for that output device. The following technical relations can be associated with the customer requirements:

- efficient exchange of the raster data is directly related to the file size and compression ratios;
- image quality in a scan-anywhere, print-anywhere environment is directly related to the exchange of device-independent data forms and the rendering compromises made by the output engine;
- fast printing with modest resources is related to low complexity of the format.

The best approach to achieve high compression ratios and retain quality is to compress the different segments of the raster data according to their individual attributes. Text and line-art data (bi-level data) would be compressed with an approach that puts high emphasis on maintaining the detail and structure of the input. Pictures and colour gradients (multi-level data) would be compressed using an approach that puts a high emphasis on maintaining the smoothness and accuracy of the colours. These different data types (bi-level and multi-level) are often conceptualized as being on separate layers/planes within the page.

This separation of the data by importance of content (spatial detail vs colour) also directly implies that it is advantageous to use different resolutions for the different data, with a high spatial resolution used for text/line-art and high colour resolution for images/gradients.

This concept of data separation by importance of content has led to development of the base mode 3-layer model on which the MRC Recommendation is built. Provisions to extend the model beyond the base mode are defined in Annex A. The base mode 3-layer model identifies three basic data types that may be contained within a page. These are multi-level data associated with contone colour (continuous-tone and/or palletized colour) image for which mid-to-low spatial and high colour resolution is typically appropriate for good reproduction; bi-level data associated with high detail of text/line-art for which high spatial and low colour resolution is typically appropriate for good reproduction; multi-level data associated with multi-level colours of the text/line-art data for which mid-to-high spatial and mid-colour resolution is typically appropriate for good reproduction. Each page within the MRC model is processed independently. The data types within each page are represented in distinct layers (also referred to as planes) to be image processed, compressed and transmitted independently. Multi-level contone data may be represented in the lower layer, bi-level in the middle layer and multi-level data of text/line-art colours in the upper layer. The lower and upper layers will from here on be referenced as the background and foreground layers respectively, see Figure 1. The process of image regeneration is controlled by the middle bi-level layer that acts as a mask or selector to select whether pixels from the background contone layer or foreground text/line-art colour layer will be reproduced. Due to its selection function, this layer is referenced as the mask or selector layer; throughout this Recommendation the middle layer will be referenced as the mask layer. When the value of a mask layer pixel is one (1), the corresponding pixel from the foreground is selected and reproduced. When the value of the mask layer pixel is zero (0), the corresponding pixel from the background is selected and reproduced, see Figure 2.

Given limited device memory in many facsimile implementations and that mixed content pages often have a mixture of: text/line-art (monochrome or coloured) regions; contone image regions; text/line-art (monochrome or coloured) and contone image regions, there are provisions to subdivide the page into horizontal stripes that span the entire width of the page and isolate individual regions, see Figure 3. Stripes are composed of one or more layers as determined by the image type within the stripe. The mask layer must span the entire width and height of the stripe. The background and foreground layers need not span the width and height of the stripe. Reduction in the amount of white space coded in the background or foreground layers can be realized by taking advantage of the image width and height data included in the layer data stream and a horizontal and vertical offset provision. The default of the foreground base colour is black (layer base colour can be changed to any colour). The base colour is defined such that at mask pixel locations (value = 1) where a corresponding foreground pixel is not present, the foreground layer base colour is applied. The default of the background base colour is white (layer base colour can be changed to any colour). The base colour is defined such that at mask pixel locations (value = 0) where a corresponding contone image is not present, the background layer base colour is applied, see Figure 4.

The 3-layer model has 3 types of horizontal stripes that are implemented according to the type of data being addressed:

- 3-layer stripe (3LS), so referenced since it contains all three of the foreground, mask and background layers as in Figure 1. The 3LS is appropriate when addressing an image that contains both multi-coloured text/line-art and contone image or monochrome text/line-art on coloured background and contone image, as in stripes 3 and 5 of Figures 3 and 8;
- 2-layer stripe (2LS), so referenced since it contains coded data for two of the three layers (the third is set to a fixed value). The two layers may be mask and background, as in Figure 6a, or mask and foreground layers, as in Figure 6b. All combination of multiple layers shall include the mask layer. The 2LS is appropriate when addressing an image that contains monochrome text/line-art and contone image or coloured text/line-art and no contone image, as in stripes 2 and 7 of Figures 3 and 8;
- 1-layer stripe (1LS), so referenced since it contains coded data for only one of the three layers (the other two are set to fixed values). The one layer may be mask, as in Figure 7a, background, as in Figure 7b, or foreground, as in Figure 7c. The 1LS is appropriate when addressing an image that contains one of monochrome text/line-art, contone image or possibly richly coloured graphics, as in stripes 1, 4 and 6 of Figures 3 and 8.

Figure 8 provides an illustration of the various stripe types that may apply to the various image regions within a page.

The 3-layer model requires application of a multi-level coding scheme to the background and foreground layers. Any ITU-T multi-level coding (such as JPEG or JBIG, as defined in ITU-T Recs T.81 and T.43, respectively) may be used for the background or foreground. A bi-level coding scheme is required for the mask layer, any ITU-T bi-level coding (such as JBIG or MMR, as defined in ITU-T Recs T.85 and T.6, respectively) may be used, see Figure 5. The specific coders used throughout the page and over the various layers are identified at the start of each page. This information is provided by parameters in a Start of Page (SOP) Marker Segment. The spatial resolution of the mask layer, to be used throughout the page, is also identified by a SOP parameter. Layers with varied spatial resolutions may be combined within a stripe, the resolution of the foreground and background layers must be integral factors of the mask resolution layer, see Figure 5. The specific resolutions being used in the foreground and background layers are identified within a marker segment at the start of each layer within a stripe. A Start of Stripe marker segment contains parameters indicating type of stripe (1LS, 2LS or 3LS), the foreground and background layer base colour, offset of the foreground and/or background, the stripe height (number of lines) and the mask layer coded data length (number of octets).

An SOP marker segment denotes the beginning of an MRC page. This is followed by page data and terminated with a EOP (End of Page). The page data consists of stripes. During transmission, stripes are sent sequentially from the top of the page, stripe 1 through N, where N is an integer. Within a stripe, the mask layer is transmitted first, followed by the background and then the foreground as appropriate.

Recommendation T.44

Mixed Raster Content (MRC)

1 Scope

This Recommendation defines a means to efficiently represent raster-oriented pages that contain a mixture of multi-level and bi-level images. Any of the many ITU-T recommended encoding schemes, such as T.81 (JPEG) for the encoding of multi-level images and T.6 (MMR) for the encoding of bi-level images, may be combined within the context of this Recommendation. Similarly, ITU-T spatial and colour resolutions may be combined within a page. This Recommendation does not define new encodings or resolutions. The method of image segmentation is beyond the scope of this Recommendation; segmentation is left to manufacturers' implementation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ITU-T Recommendation T.4 (2003), *Standardization of Group 3 facsimile terminals for document transmission*.

– ITU-T Recommendation T.6 (1988), *Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus*. (Commonly referred to as MMR standard.)

– ITU-T Recommendation T.42 (2003), *Continuous-tone colour representation method for facsimile*.

– ITU-T Recommendation T.43 (1997), *Colour and gray-scale image representations using lossless coding scheme for facsimile*.

– ITU-T Recommendation T.81 (1992) | ISO/IEC 10918-1:1994, *Information technology – Digital compression and coding of continuous-tone still images – Requirements and guidelines*. (Commonly referred to as JPEG standard.)

– ITU-T Recommendation T.82 (1993) | ISO/IEC 11544:1993, *Information technology – Coded representation of picture and audio information – Progressive bi-level image compression*. (Commonly referred to as JBIG standard.)

– ITU-T Recommendation T.85 (1995), *Application profile for Recommendation T.82 – Progressive bi-level image compression (JBIG coding scheme) for facsimile apparatus*.

– ITU-T Recommendation T.86 (1998) | 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)*.

3 Definitions

The definitions contained in ITU-T Recs T.4, T.6, T.42, T.43, T.81, T.82 and T.85 apply, unless explicitly amended.

3.1 APP13 marker: Encoded as X'FFED', is the application marker, registered per ITU-T Rec. T.86, that uniquely identifies MRC.

3.2 end of page (EOP): Encoded as two consecutive JPEG EOI (X'FFD9FFD9').

3.3 JBIG: Joint Bi-level Image Experts Group, and also shorthand for the encoding method, described in ITU-T Rec. T.82, which was defined by this group.

NOTE – It is expected that JBIG will be changed to JBIG1 when referencing ITU-T Rec. T.82. This nomenclature change is the result of a new standard that is being developed by the JBIG committee. This new standard will be referenced as JBIG2.

3.4 JPEG: Joint Photographic Experts Group, and also shorthand for the encoding method, described in ITU-T Rec. T.81, which was defined by this group.

3.5 layer: An image, either multi-level or bi-level, that is to be combined with other images using the method described here. Layers are encoded using ITU-T coding methods. One or more layers may be used.

3.6 background layer: The "bottom" layer (layer 1), multi-level data associated with a contone image segment, in a 3-layer segmentation of a page containing a combination of bi-level and multi-level images.

At background pixel locations where the contone background image is not present, a background layer base colour (default of white) is applied. A means to define other values of background layer base colour is provided within the syntax described in clause 9.

3.7 contone layer: Continuous-tone and/or palletized colour. This definition is intended to account for both scanner and synthetic source image data. When a scanner is the source of an image, both continuous-tone and solid coloured images would be available as continuous-tone data. When the source of an image is synthetic, continuous-tone and solid coloured images may be available as continuous-tone or palletized colour data.

3.8 foreground layer: The "top" layer (layer 3), multi-level data associated with colours of text, graphics or line-art, in a 3-layer segmentation of a page containing a combination of bi-level and multi-level images.

At foreground pixel locations where the multi-level data associated with colours of text, graphics or line-art is not present, a foreground layer base colour (default of black) is applied. A means to define other values of foreground layer base colour is provided within the syntax described in clause 9.

3.9 image layer: Odd-numbered layer (e.g., layers 1, 3, 5, …), multi-level data associated with contone images, colours of texts, graphics or line-art, in a multi-layer segmentation of a page containing a combination of bi-level and multi-level images.

At image layer pixel locations, above layer 1, where the image is not present, a layer base colour, default of black, is applied. At layer 1 pixel locations where the image is not present, a layer base colour, default of white, is applied. A means to define other layer base colour values is provided within the syntax described in clause 9.

3.10 mask layer: Even-numbered layers (i.e., layers 2, 4, 6, …), bi-level data, in a multi-layer segmentation of a page containing a combination of bi-level and multi-level images. The bi-level mask layer selects the image layer directly above it or the image(s) below it to be visible. A corresponding image layer pixel above the mask layer is selected for reproduction when a mask layer pixel value is "1". A corresponding pixel from the image or collection of images below the mask is selected when a mask pixel value is "0".

The first mask layer (layer 2) may be distinguished as the main mask. The main mask selects the foreground or background to be visible. In a 3-layer segmentation it is simply referenced as the mask layer. When there is more than one mask layer, the other mask layers (layers 4, 6, 8, …) may be referenced as the overlay masks.

3.11 virtual mask layer: Even-numbered layer (i.e., layers 2, 4, 6, …) that contains no coded data. A virtual mask layer is used to establish dimensions of a page or stripe when there is no coded mask layer that spans the full dimensions of the page or stripe.

3.12 Modified Huffman (MH) is shorthand for the lossless bi-level one-dimensional encoding method described in ITU-T Rec. T.4.

3.13 modified modified READ (MMR) (READ is an acronym for Relative Element Address Designate) is shorthand for the lossless bi-level encoding method described in ITU-T Rec. T.6.

3.14 modified READ (MR) (READ is an acronym for Relative Element Address Designate) is shorthand for the lossless bi-level two-dimensional encoding method described in ITU-T Rec. T.4.

3.15 MRC magic number: MRC Magic number, encoded as the JPEG SOI (Start of Image – X'FFD8') to alert decoders that JPEG application markers registered per ITU-T Rec. T.86 follow.

3.16 start of page marker segment (SOP): Encoded as APP13 (X'FFED'), length of segment, SOP ident (MRC0), parameters.

3.17 start of stripe marker segment (SOSt): Encoded as APP13 (X'FFED'), length of segment, SOSt ident (MRC1), parameters.

3.18 stripe: An image swath, spanning the width of the page, that may consist of one or more layers.

3.19 termination number (TN): Encoded as the JPEG EOI (End of Image – X'FFD9') to alert decoders to the end of the initial JPEG application marker registered per ITU-T Rec. T.86. The TN is located directly after the SOP parameters.

4 Conventions

The conventions in ITU-T Rec. T.81 apply to this Recommendation.

5 Image representation

This Recommendation includes description of syntax for encapsulating one or more ITU-T encodings on a single page.

A page is composed from a set of page-wide stripes of image data, which are coded independently. The stripes are transmitted sequentially from the top to the bottom of the page.

In the base mode, stripes are composed of one to three (3) layers. Each layer is coded using a recommended ITU-T coding method. The base mode is mandatory and must be supported by all future modes. All future modes should support all previously defined modes unless specified otherwise.

Information required to decode the page, such as coding types used within the layers, is specified within the page header (start of page marker segment). Stripe height is specified within the stripe header (start of stripe marker segment).

Information required to decode a layer is included in the stripe header and the layer data.

In the base mode, the mask layer is transmitted first, followed by the background layer and then the foreground layer.

Details of the syntax are described below.

6 Stripe structure

In the base mode, stripes are composed of one to three layers; background layer, mask layer and foreground layer. Annex A makes provision for stripes to be composed of more than three layers. One or more layers may be assigned a fixed value (e.g., fixed colour value). Virtual mask layers and fixed value layers are not counted in classifying the stripe types as follows:

- Three-layer stripe: (3LS)
- Two-layer stripe: (2LS)
- One-layer stripe: (1LS)

6.1 Three-layer stripe (3LS)

3LS is the basic structure of this Recommendation. The 3LS contains foreground, mask and background layers, see Figure 1 and stripes 3 and 5 of Figure 8. It provides a means to transfer two images and a bi-level mask layer describing their recombination on the same page. This capability enables representation of richly coloured text, graphics, and line-art together with contone imagery in the same region using only multi-level and bi-level coding methods. It also enables representation of monochrome or coloured text, graphics, and line-art that resides on a coloured background together with contone image in the same region. The text/line-art colour is placed in the foreground layer and the contone image in the background layer. The bi-level mask plane is used to select which of the images is expressed at every pixel location within the stripe. It may contain the high detail text shape or rectangular outlines of text and contone image areas.

6.2 Two-layer stripe (2LS)

2LS is one of the special cases of the 3LS, in which one of the foreground or background layers is assigned a fixed colour value. The mask layer is mandatory in the 2LS stripe. The 2LS contains mask and background or mask and foreground layers. For the case of mask and background, the foreground is set to a layer base colour value (e.g., black), see Figure 6a and stripe 2 of Figure 8. It provides a means to transfer a continuous-tone image, a layer base colour value and a bi-level mask layer describing their recombination on the same page. This capability enables representation of monochrome text, graphics, and line-art together with contone imagery in the same region using only multi-level and bi-level coding methods. The monochrome text/line-art may overlay the coloured image. The text/line-art colour is represented by the fixed foreground value while the contone image is in the background layer. The bi-level mask plane is used to select which of the foreground layer base colour or the background images is expressed at every pixel location within the stripe. The mask contains the high detail shape of text, graphics or line-art. For the case of mask and foreground, the background is set to a layer base colour value (e.g., white), see Figure 6b and stripe 7 of Figure 8. It provides a means to transfer a colour foreground image, a background layer base colour value and a bi-level mask layer describing their recombination on the same page. This capability enables representation of coloured text, graphics, and line-art without contone imagery in the same region using multi-level and bi-level coding methods. The bi-level mask plane is used to select which of the foreground or the background images is expressed at every pixel location within the stripe.

6.3 One-layer stripe (1LS)

1LS is one of the special cases of the 3LS, in which two of the three layers are assigned a fixed value (e.g., fixed colour value in the case of fixed foreground or background layer). The 1LS contains a single coded layer. If the layer is coded using a bi-level coding method, the fixed foreground and background base layer colour values, defined within the stripe, are applied; a bi-level image is treated as a mask layer, see Figure 7a and stripe 1 of Figure 8. A page containing a single bi-level 1LS, where the foreground and background colours are the values of black and white respectively, is similar to a conventional bi-level facsimile page, see Figure 7a. There are two cases if the coding method is multi-level. Case 1: the mask is fixed to "0" (foreground colour is not applied) and the background layer base colour (e.g., white) is applied outside the area coded, see Figure 7b and stripes 4 and 6 of Figure 8. Case 2: the mask is fixed to "1" (background colour is not applied) and the foreground layer base colour (e.g., white in this case) is applied outside the area coded, see Figure 7c. To represent dimensions of pages that contain only case 1 or 2 stripe(s), where there is no coded mask data, a virtual mask layer should be assumed. The resolution of the virtual mask is fixed to that of the foreground or background layer while the dimensions of the virtual mask are fixed to the page dimensions, such as page width and height (number of scan lines). The width of the stripe is fixed to the page width. It is possible in cases 1 and 2 that the foreground or background layers may contain a fixed colour value (e.g., a layer base colour value); in effect there is no encoded colour data.

The 1LS is applied to areas containing only monochrome text, graphics (e.g., business graphics)/line-art, or continuous-tone image data.

7 Image coding

7.1 Spatial resolution

The resolution of the main mask layer is fixed for the entire page and defines the maximum resolution for the page. In general it is possible to define foreground and background layers of lower spatial resolution. Spatial resolution of all layers must be integral factors of the main mask layer resolution. All resolutions used must be square (i.e., same horizontal and vertical values) and conform to ITU-T recommended values. The main mask resolution is specified in the page header. The foreground and background resolutions are indicated in the layer data.

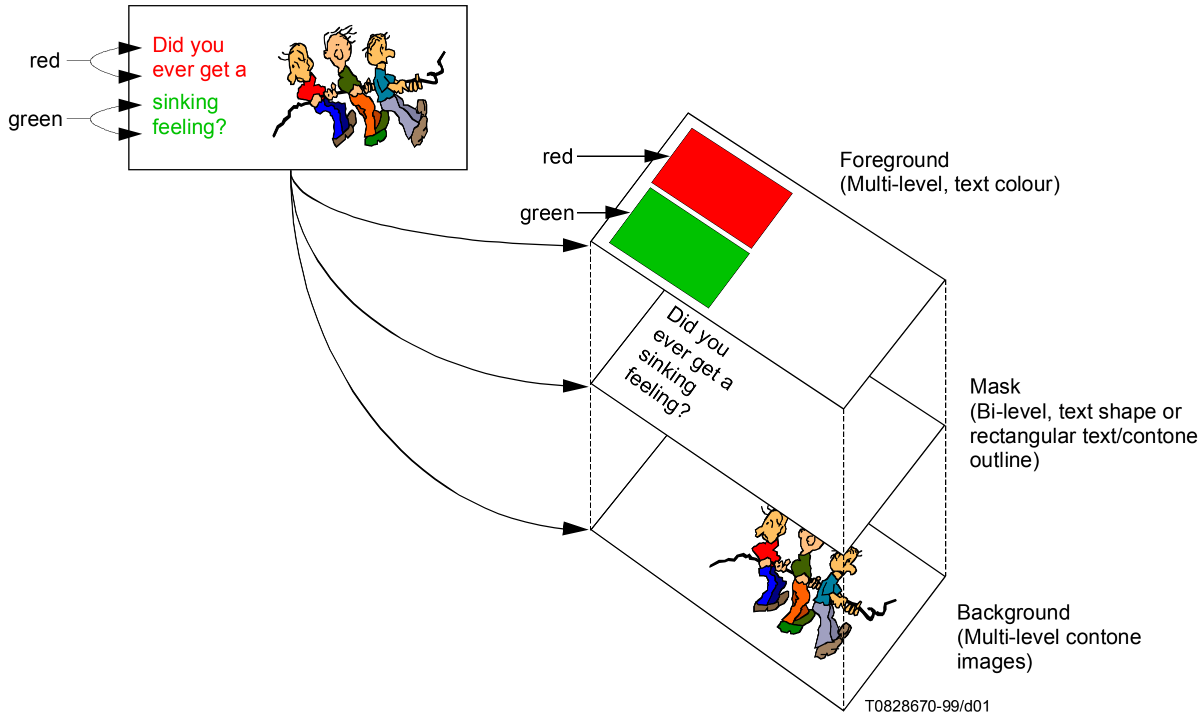

Figure 1/T.44

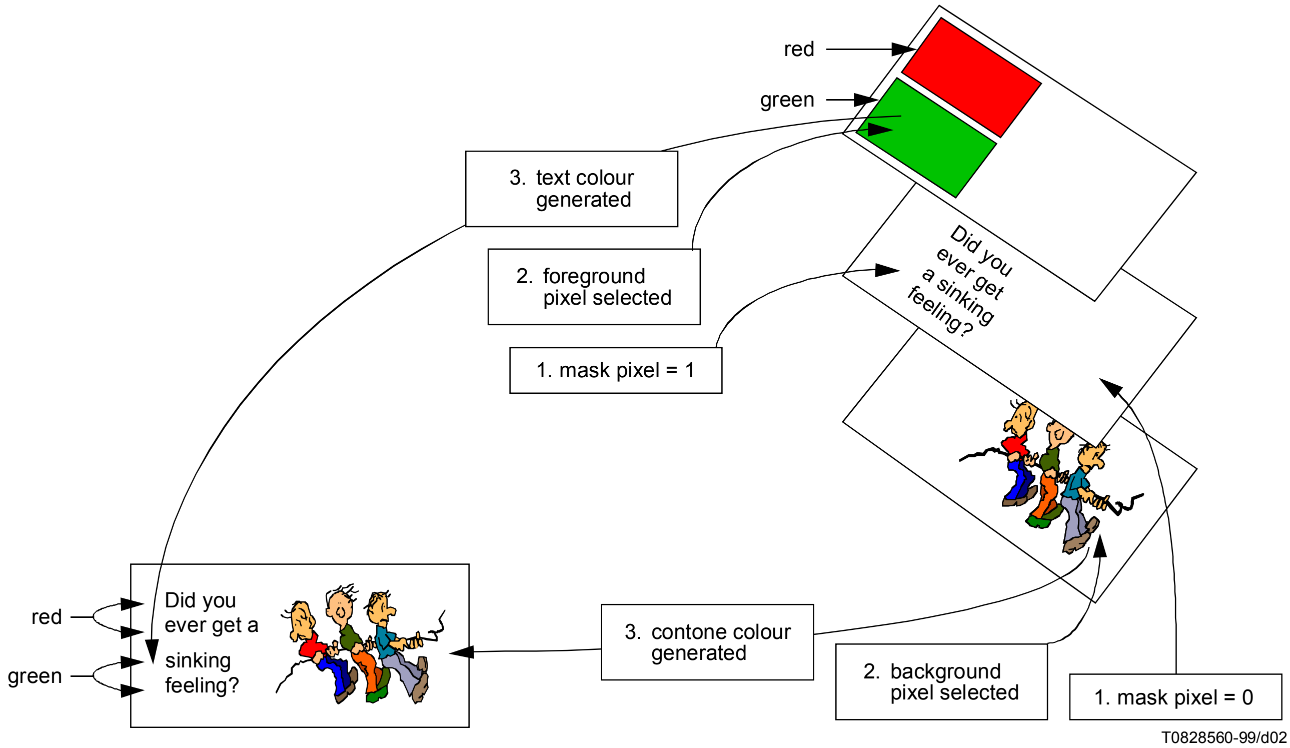

Figure 2/T.44

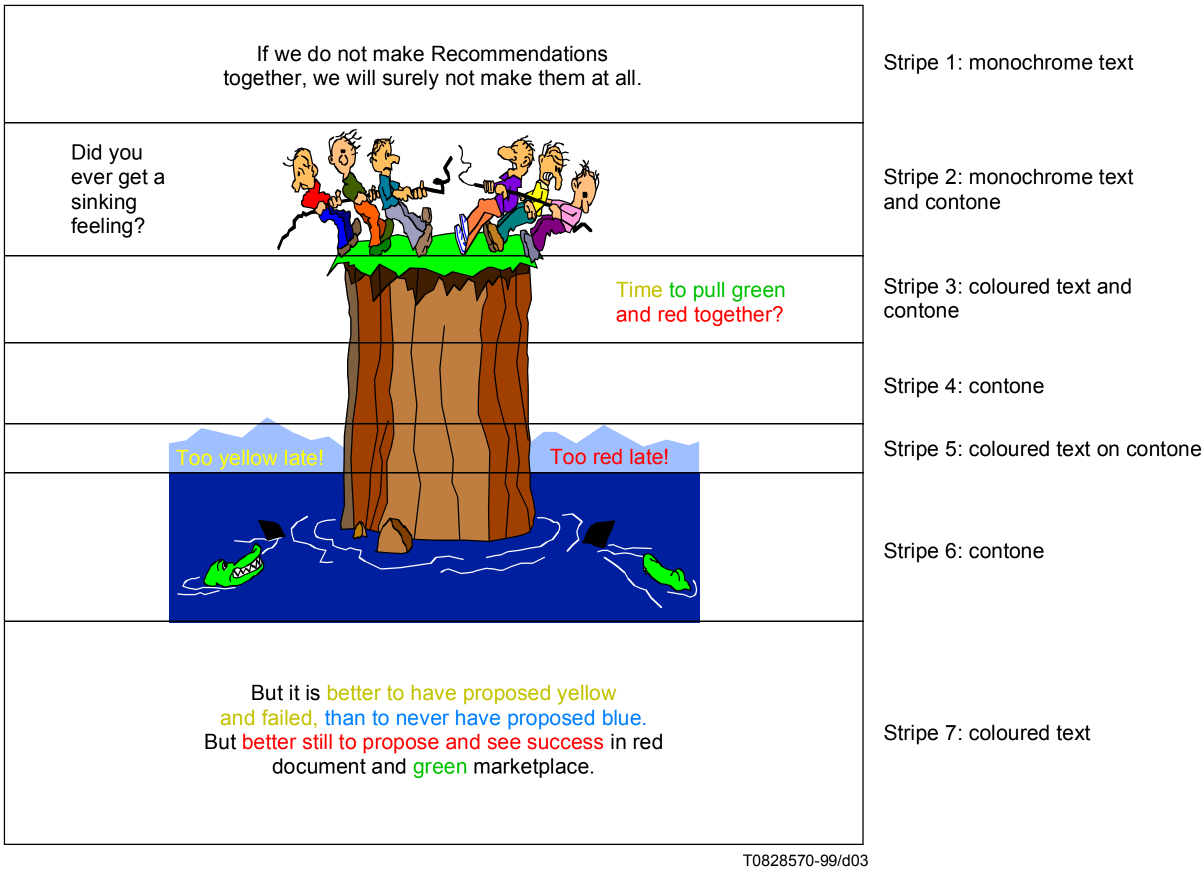

Figure 3/T.44

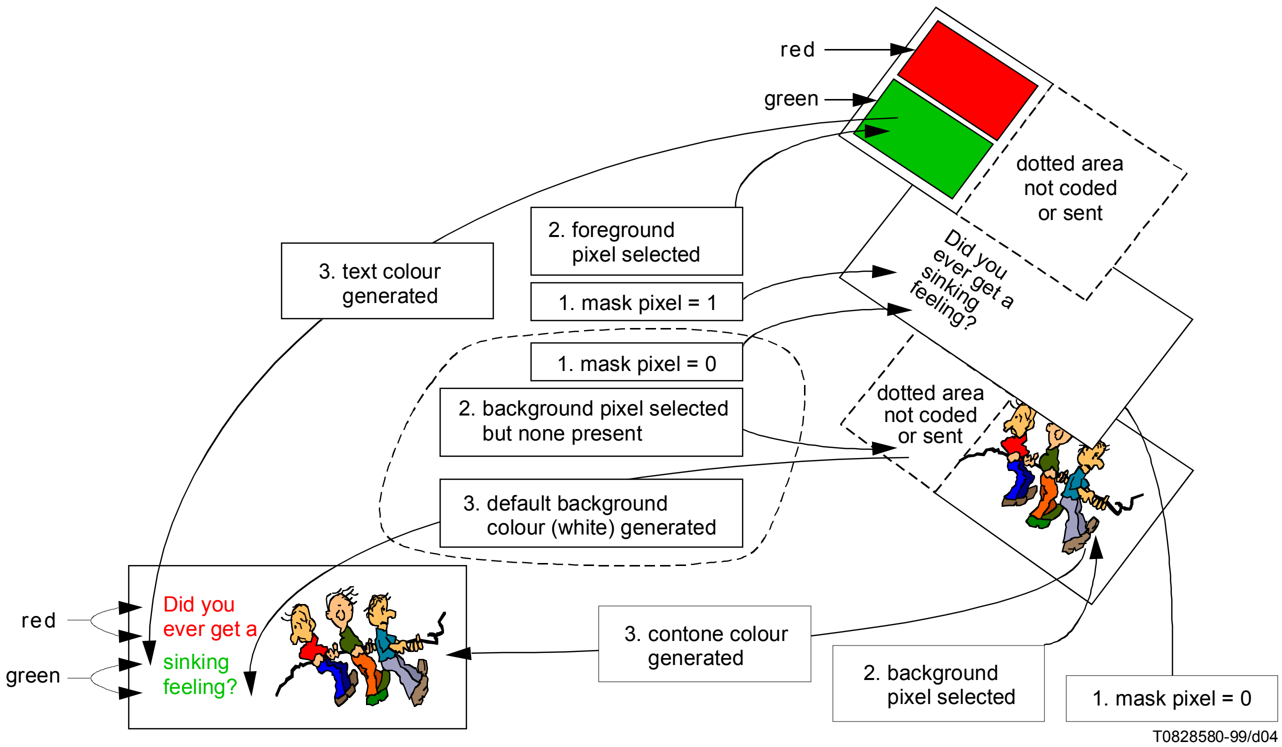

Figure 4/T.44

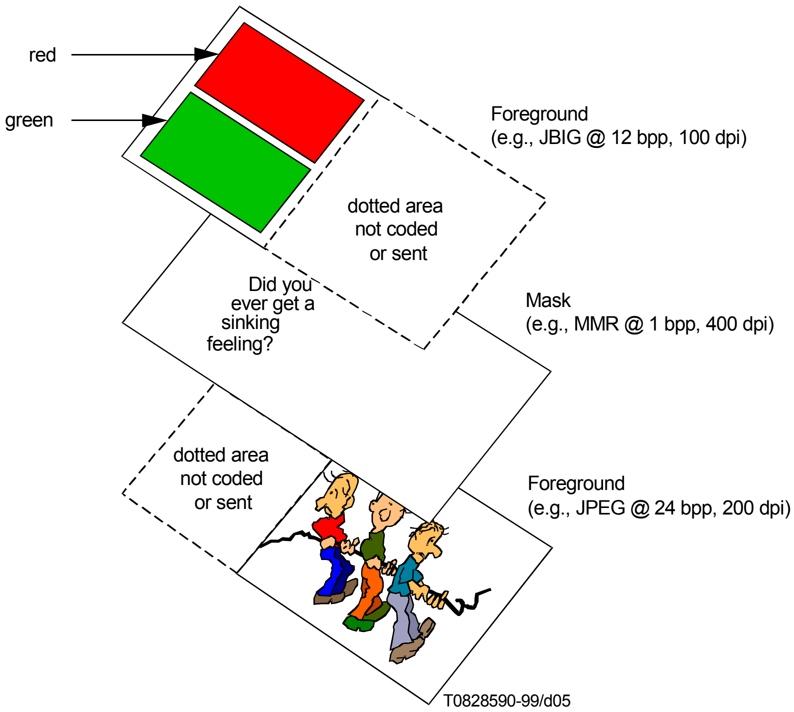

Figure 5/T.44

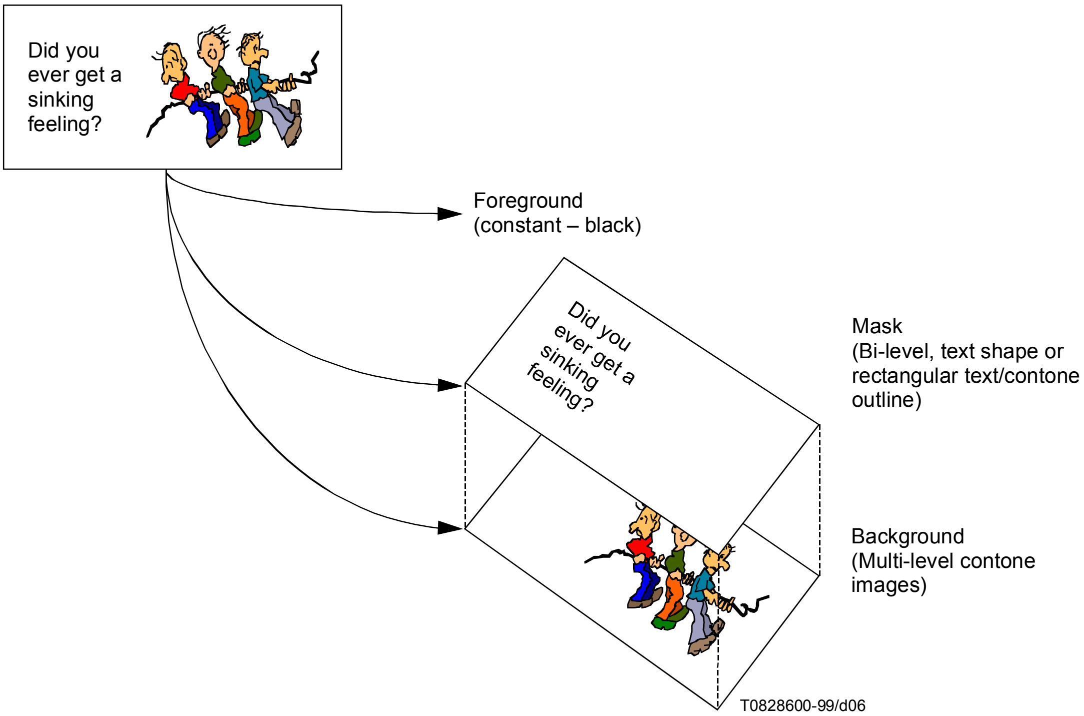

Figure 6a/T.44 – Mask and background

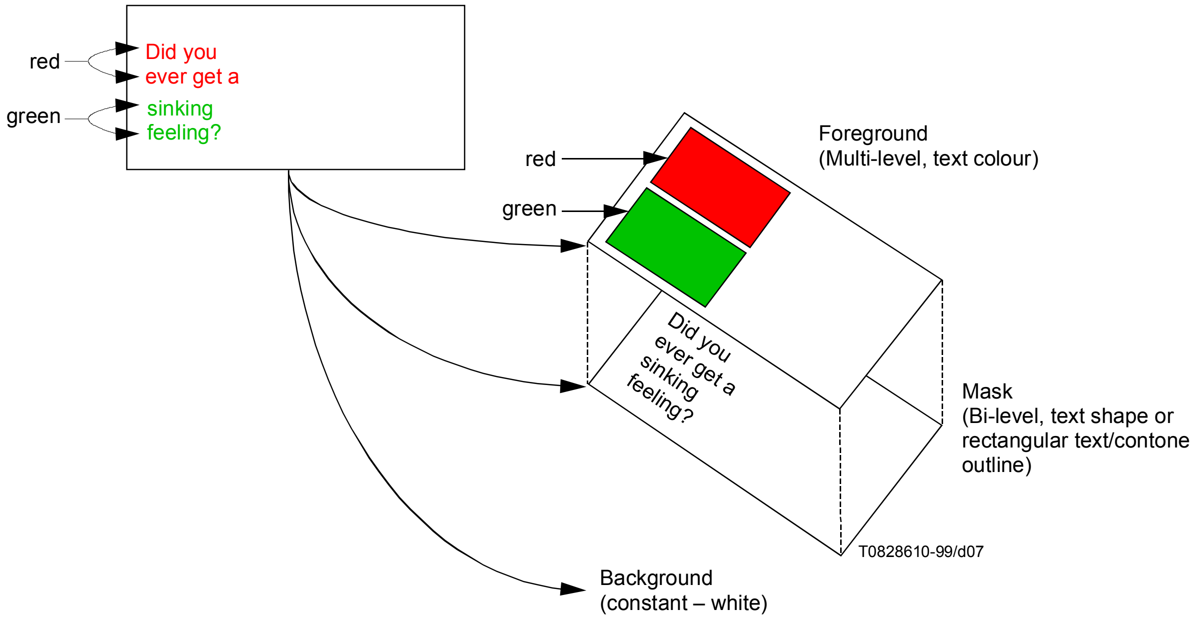

Figure 6b/T.44 – Mask and foreground

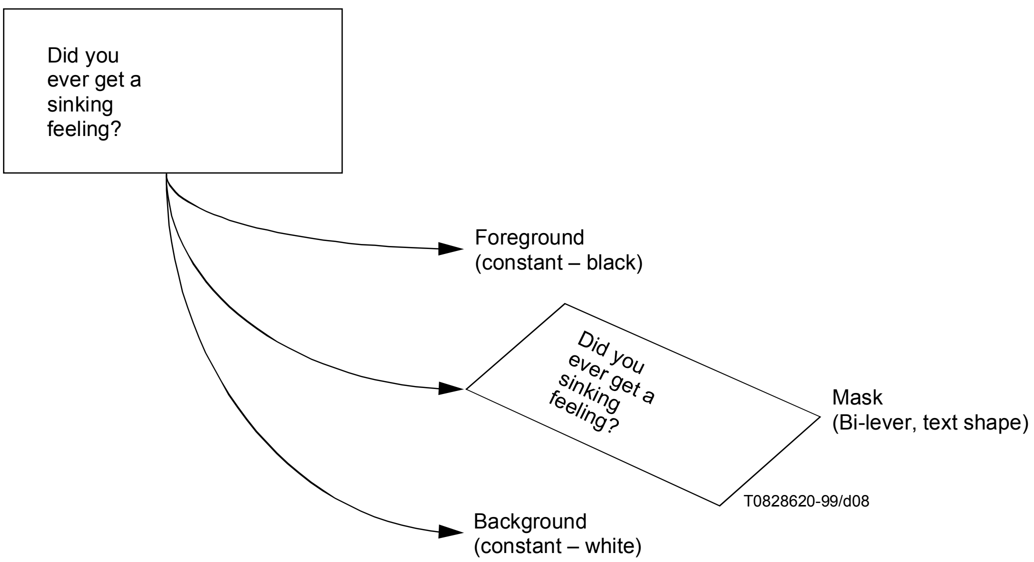

Figure 7a/T.44 – Mask

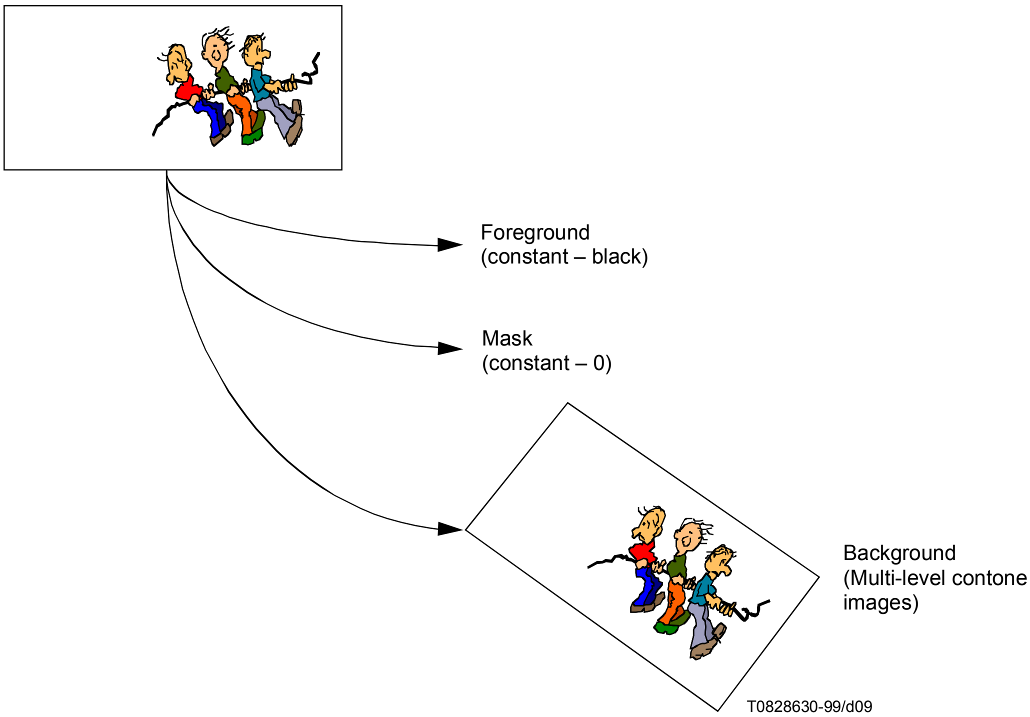

Figure 7b/T.44 – Background

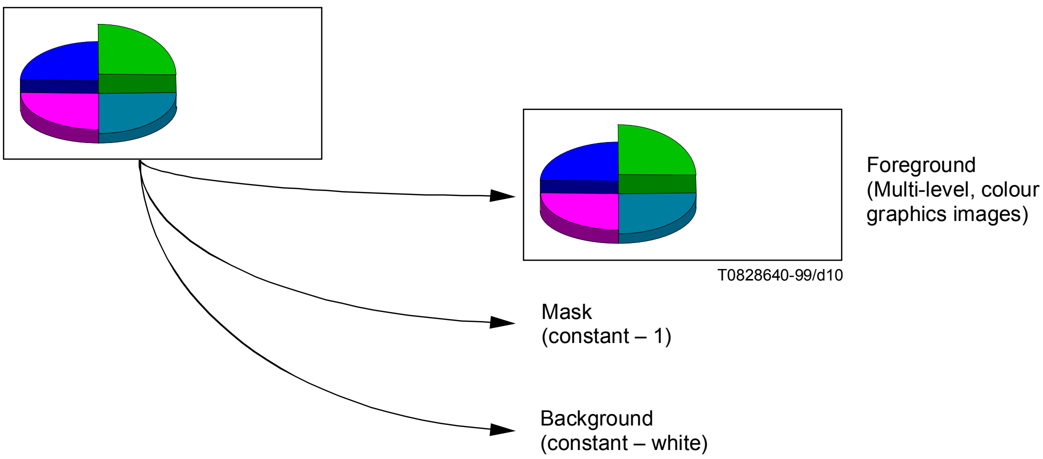

Figure 7c/T.44 – Foreground

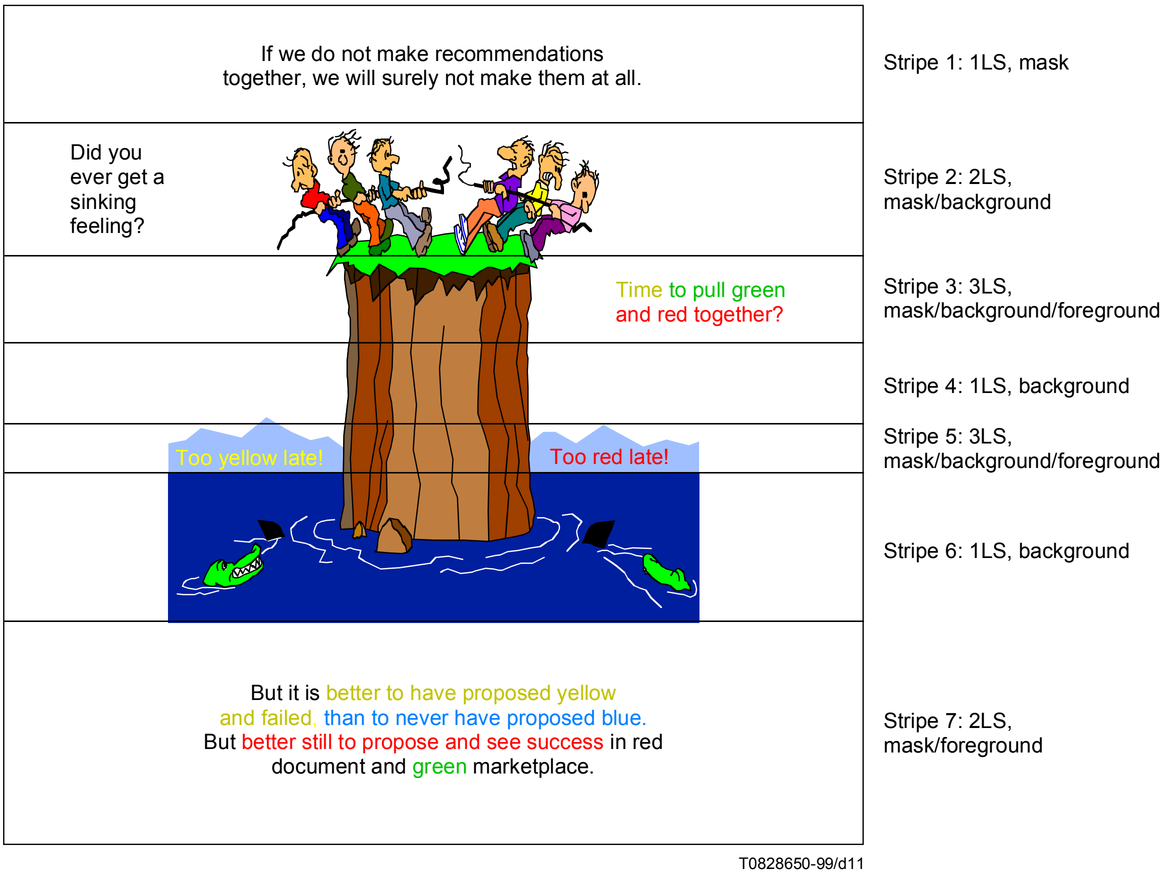

Figure 8/T.44

7.2 Stripe width and layer width

Stripes always span the entire width of a page. The main mask layer must always span the entire width.

This method takes advantage of the image width and height data included in the layer data stream, such as JPEG. A foreground and/or background layer (e.g., JPEG data) is not required to span the entire width. All layers must be fully contained within the boundaries of the stripe. In addition, a horizontal offset may be used to select a starting point to the right of the left stripe boundary. This offset is expressed in the main mask layer pixel units. A simple stripe containing only background (e.g., JPEG data) or foreground (e.g., T.43 JBIG data) image data may use this feature also.

7.3 Stripe height and layer height

To limit the data that must be buffered by an application, some applications may choose to limit the maximum height of two or more layer stripes (2LS & 3LS) to a specified number of lines (in mask layer resolution).

One-layer stripes (1LS) are not required to conform to a maximum stripe height, and are only limited by page size. Layers without coded data (i.e., virtual mask layers and image layers with only base layer colour) are not to be counted when considering whether a stripe is a 1LS, 2LS or 3LS stripe.

Stripe and main mask layer heights are always equal. Foreground and background layer heights are less than or equal to stripe heights. All layers must be fully contained within the boundaries of the stripe. In addition, a vertical offset may be used to select a starting point below the first scan line of the stripe. This offset is expressed relative to the first scan line at the top of the stripe and in the main mask pixel units. A simple stripe containing only background (e.g., JPEG) or foreground (e.g., JBIG) data may use this feature also.

7.4 Layer combination

Image layers are rendered sequentially in ascending order of layer number (i.e., layer 1 then 3). The background layer (i.e., layer 1), if present, shall be rendered first. Bi-level mask layers (even-numbered layers, such as layer 2) select pixels from their corresponding image layer (odd-numbered layer directly above the mask layer, such as layer 3) for rendering. A corresponding image layer pixel (directly above the mask layer pixel), or its layer base colour value, is selected when a mask pixel value is "1". The selected image layer pixel is rendered on top of any layer that may have been previously rendered. A corresponding image layer pixel shall not be rendered when the mask pixel value is "0". The pixel from the layer below the mask, or its layer base colour value, shall remain visible when the mask pixel value is "0". In the event of an image layer (i.e., layer 3), or portion thereof, without a corresponding mask layer, the image layer shall be rendered on top of any previously rendered layer.

8 Layer transmission order

In 3LS, the bi-level mask data is transmitted first, followed by the background layer and then the foreground layer. In 2LS, the bi-level mask image data is transmitted first, followed by the background or foreground layer.

9 Data format

9.1 Overview

The MRC image data consist of a series of: markers; parameter data that specify the image coder, image size, bit-resolution and spatial resolution; image data. The conventions of Annex B/T.81 are used broadly here. The JPEG registration body, per ITU-T Rec. T.86, has been used to register the marker code, APP13, classified as an application marker.

The MRC page structure for this application has the following elements: parameters, markers, and entropy-coded data segments. Parameters and markers are often organized into marker segments. Parameters are integers of length ½, 1, 2 or more octets. Markers are assigned two or more octet codes, an X'FF' octet followed by an octet not equal to X'00' or X'FF' and optionally proceeded by extra X'FF' octet codes. This base mode application defines marker segments to denote the Start of Page (SOP), optional marker segments and the Start of Stripe (SOSt). The MRC Magic Number (JPEG SOI) is used immediately preceding the application marker as part of the SOP marker segment. The JPEG EOI is used as a termination number located directly after the last SOP parameter. The end of a page (EOP) is defined as X'FFD9FFD9'. These markers are inserted by the encoder, and understood by the decoder in addition to all markers used for the coding methods, such as SOS (start of scan) of ITU-T Rec. T.81.

A distillation of the bit and byte ordering conventions of Annex B/T.81 that are applied throughout this Recommendation follows:

Bits are packed into octets starting at the most significant bit. If a decoder is reading a sequence of bits from a bit-stream, it shall first read the most significant bit of the first octet, the next most significant bit, and so on, then proceed to the next octet.

All multi-octet values shall be interpreted in a most-significant-first manner: the first octet of each value is most significant, and the last octet is the least significant.

9.2 Page data structure

The beginning of a MRC page is denoted by the Start of Page Marker Segment, followed by termination number, optional marker segments, page data and EOP. The optional marker segments are optional, unless otherwise stated. Their purpose is to provide insight into reproduction of the image and as such are typically not mandatory for image reproduction. Skipping of any unrecognized optional marker segment is appropriate. Page data consists of stripes 1 to N, as described in 9.2.1.

Marker segments defined in this Recommendation and located between the Start of Page Marker Segment (SOP) and EOP marker shall have the following structure, which is consistent with that of the SOP:

APP13 (X'FFED'), Length of segment, identifier, Optional length (if needed), parameters and/or segment data;

where the Optional length is included when two octets are insufficient to represent the Length of segment and four octets are needed.

In typical use, the 2-octet Length of segment should be sufficient to represent the length of the marker segment header and data, not including the 2-octet APP13 (X'FFED') marker itself; in which case the Optional length will be omitted. In cases where two octets are insufficient, the 2-octet Length of segment will take on a value of zero (0), and the Optional length will be used. If the 2-octet Length of segment is any value less than six (which is the clear minimal size of a marker segment for both the 2-octet length and the 4-octet identifier), then the Optional length is required. Values between one and five are reserved for future use.

All APP13 (X'FFED') markers currently adhere to this rule with the exception of EOH, which does not include the data length in the Length of segment, but includes that length separately as the only parameter in the marker segment header.

9.2.1 Start of page marker segment

Start of Page Marker Segment has the following structure:

MRC Magic Number, APP13, Length of segment, SOP ident, version, mask coder, image layer coders, mask resolution, width.

The Start of Page Marker Segment is defined as follows:

MRC Magic number:	2 octets	X'FFD8'
APP13 Marker:	2 octets	X'FFED'
Length of Segment:	2 octets	Length of segment in octets, MSB to LSB, as an integer value including the octet count itself but not including Magic Numbers or APP13.
SOP Ident:	4 octets	'MRC0', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D', X'52', X'43', X'00'). This X'00'-terminated string "MRC" uniquely identifies this marker segment as the start of page.
Version:	1 octet	Revision number, X'02' indicating revision "2".
Mode:	1 octet	X'01', indicating mode 1.0. Each mode identifies a different level of performance. Mode 1.0 identifies the base level of T.44 as defined by the contents of this Recommendation. Each incremental mode shall be defined in an Annex to this Recommendation and support the capabilities defined in this mode.
Mask layer coders:	1 or more octet(s)	With value indicating coder as shown in Table 1. The identified coders may be used in any mask layer. The main mask layer is the only mask (even-numbered) layer permitted in this mode (Mode 1.0) and only one coder may be selected. Only one coder shall be used for the main mask (layer 2). The value shall be fixed to zero "0" in the event that there is no mask layer coder (i.e., no coded mask layer data present).
Image layer coders:	1 or more octet(s)	With value indicating coders as shown in Table 2. The identified coders may be used in any image layer. Background and foreground are the only image (odd-numbered) layers permitted in Mode 1.0. Any one of the selected coder(s) may be used in an image layer. The value shall be fixed to zero "0" in the event that there is no image layer coder.
Main Mask resolution:	2 octets	Expressing vertical and horizontal resolution as a single integer value in units of pels/25.4 mm. Basic value is 200 pels/25.4 mm. The value shall be fixed to that of the image layer in the event that there is no coded mask data (layer) in the page.

Page width: 4 octets Expressing page width as a single integer value. For pages with two or more layers, the main mask layer image width defines the page width in using units of the main mask resolution. For pages with only a single layer foreground or background image, no coded mask data, a virtual mask (i.e., a mask layer without coded data) shall be used to define the page width.

Table 1/T.44 – Mask (even-numbered layer) coder octet(s)

Octet bit number	Coder used
LSB 0	One-dimensional T.4 (MH) coding
1	Two-dimensional T.4 (MR) coding
2	T.6 (MMR) coding
3	T.82 (JBIG1) coding applying ITU-T Rec. T.85
4	T.88 (JBIG2) coding, Annex B/ T.44 is required
5	Reserved
6	Reserved
MSB 7	Extend, add another octet that follows immediately
NOTE – New bi-level coders (i.e., a ~~5th,~~ 6th and 7th coder) would be assigned bit numbers ~~4,~~ 5 and 6 respectively. Bit 7, the extend bit, would be set when adding another octet to accommodate additional coders, such as an 8th which would be assigned to bit number 8.	

Table 2/T.44 – Image (odd-numbered layer) coder octet(s)

Octet bit number	Coder used
LSB 0	T.81 (JPEG) coding and ITU-T Rec. T.42/LAB
1	T.82 (JBIG1) coding applying ITU-T Recs T.43 and T.42/LAB
2	T.45 "Run-length Colour Encoding" and ITU-T Rec. T.42/LAB, Annex B/T.44 is required (Note 1)
3	T.81 (JPEG) coding and ITU-T Rec. T.42/YCC
4	T.82 (JBIG1) coding applying ITU-T Recs T.43 and T.42/YCC
5	T.45 "Run-length Colour Encoding" and ITU-T Rec. T.42/YCC, Annex B/T.44 is required (Note 1)
6	Reserved
MSB 7	Extend, add another octet that follows immediately
NOTE 1 – Coding scheme(s) referencing this Note shall use the SLC (Start of Layer Coded data) marker segment, defined in the Layer Data Structure clause of Annex A/T.44. This means that Mode 1 shall not be used with the referencing coding scheme. NOTE 2 – New multi-level coders (i.e., ~~4th through~~ 7th coder) would be assigned bit numbers ~~2 through~~ 6 ~~respectively~~. Bit 7, the extend bit, would be set when adding another octet to accommodate additional coders, such as an 8th which would be assigned to bit number 8. NOTE 3 – In Mode 1, the image code shall use either LAB (bits 0, 1 and 2) or YCC (bits 3, 4 and 5), but not both. Therefore, if any of bits 0, 1 or 2 is set, bits 3, 4 and 5 shall not be set. Conversely, if any of bits 3, 4 or 5 is set, bits 0, 1 and 2 shall not be set.	

9.2.2 Optional marker segments

The optional marker segments are optional, unless otherwise stated. Their purpose is to provide insight into reproduction of the image and as such are typically not mandatory for image reproduction. Skipping of any unrecognized optional marker segment is appropriate.

Optional marker segments (OMSx) consist of marker and associated parameters. The APP13 (X'FFED') marker initiates identification of the entry. Each optional marker segment is identified by the 3-octet ASCII string plus a hexadecimal count for 'MRCn'. The 'MRCn' identifier is a 4-octet value X'4D',X'52',X'43',X'n', where n may equal X'0A' (10) to a maximum of X'FE' (254). The optional marker segments are located after the Termination Number (TN).

Each optional marker segment (OMSx) has the following structure: APP13 Marker (X'FFED'), Length of entry, OMSx ident (MRCn), Optional length (if needed), entry parameters and/or entry data.

OMSx represents specific optional marker segments, where "x" is the value of a character used to distinguish each optional marker segment.

9.2.2.1 Layer base colour gamut range marker segment (OMSgl), MRC10 entry

This entry specifies the gamut range information of LAB for indicating image layer (i.e., odd-numbered layers such as background and/or foreground layers) base colour. Structure of the OMSgl marker segment entry is as follows:

APP13, length, OMSgl identifier, gamut range data for LAB.

The OMSgl marker segment is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length:	2 octets	Total entry field octet count, MSB to LSB, including the octet count itself, but excluding the APP13 marker.
OMSgl identifier:	4 octets	'MRC10', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D', X'52', X'43', X'0A'). This X'0A'-terminated string "MRC" uniquely identifies this entry marker as containing MRC information about optional gamut range data for LAB used to represent the layer base colour of the image layer in all stripes of the page.
Gamut range data:	12 octets	The data field contains six two-octet signed integers. For example, the two-octet signed integer X'0064' represents 100. The example gamut range L* = [0, 100], a* = [–85, 85], and b* = [–75, 125] would be represented by the code: X'0000', X'0064', X'0080', X'00AA', X'0060', X'00C8'

The calculation from a real value L* to an eight-bit value, L, is made as follows:

$$L = (255/Q) \times L^* + P$$

where the first integer of the first pair, P, contains the offset of the zero point in L* in the eight most significant bits. The second integer of the first pair, Q, contains the span of the gamut range in L*. Rounding to the nearest integer is performed. The second pair contains offset and range values for a*. The third pair contains offset and range values for b*. If the image is gray-scale (L* only), the field still contains six integers, but the last four integers are ignored.

NOTE – This description of gamut range is similar to APP1 (G3FAX1) defined in Annex E/T.4, except that 12-bit numbers are not defined.

9.2.2.2 Layer base colour illuminant marker segment (OMSi), MRC11 entry

This entry specifies the illuminant information for indicating image layer (i.e., odd-numbered layers such as background and/or foreground layers) base colour. Structure of OMSi entry is as follows:

APP13, length, OMSi ident, illuminant data.

This option is for further study with the exception of the default case; the specification of the default illuminant, CIE Illuminant D50 for LAB colour space, may be added for information.

For YCC colour space, D65 is the default illuminant, and the optional illuminant is not permitted. Therefore, this option is not applied for YCC colour space.

The OMSi segment is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length:	2 octets	Total entry field octet, MSB to LSB, count including the octet count itself, but excluding the entry marker.
OMSi identifier:	4 octets	'MRC11', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D', X'52', X'43', X'0B'). This X'0B'-terminated string "MRC" uniquely identifies this entry marker as containing MRC information about optional illuminant data for representing the layer base colour.
Illuminant data:	4 octets	The data consist of a four-octet code identifying the illuminant. In the case of a CIE standard illuminant, the four-octet code is one of the following:

– CIE Illuminant D50: X'00', X'44', X'35', X'30'

– CIE Illuminant D65: X'00', X'44', X'36', X'35'

– CIE Illuminant D75: X'00', X'44', X'37', X'35'

– CIE Illuminant SA: X'00', X'00', X'53', X'41'

– CIE Illuminant SC: X'00', X'00', X'53', X'43'

– CIE Illuminant F2: X'00', X'00', X'46', X'32'

– CIE Illuminant F7: X'00', X'00', X'46', X'37'

– CIE Illuminant F11: X'00', X'46', X'31', X'31'

NOTE – This description of illuminant is similar to APP1 (G3FAX2) defined in Annex E/T.4, except the colour temperature alone is not allowed.

9.2.2.3 Layer base colour gamut range marker segment (OMSgy), MRC09 entry

This entry specifies the gamut range information of YCC for indicating image layer (i.e., odd-numbered layers such as background and/or foreground layers) base colour. Structure of the OMSgy marker segment entry is as follows:

APP13, length, OMSgy identifier, gamut range data for YCC.

The OMSgy marker segment is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length:	2 octets	Total entry field octet count, MSB to LSB, including the octet count itself, but excluding the APP13 marker.
OMSgy identifier:	4 octets	'MRC9', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D', X'52', X'43', X'09'). This X'0C'-terminated string "MRC" uniquely identifies this entry marker as containing MRC information about optional gamut range data for YCC used to represent the layer base colour of the image layer in all stripes of the page.
Gamut range data:	12 octets	The data field contains six two-octet signed integers. For example, the two-octet signed integer X'03E8' represents 1000. The example gamut range Y = [0, 1.0], Cb = [–0.5, 0.5], and Cr = [–0.5,0.5] would be represented by the code: X'0000', X'03E8', X'0080', X'03E8', X'0080', X'03E8'

The calculation from a real value Y to an eight-bit value, NY, is made as follows:

$$NY = (255/(Q/1000)) \times Y + P$$

where the first integer of the first pair, P, contains the offset of the zero point in Y in the eight most significant bits. The second integer of the first pair, Q, contains an integer representation of the span of the gamut range in Y whereby the decimal point is implicitly in the third decimal place. Rounding to the nearest integer is performed. The second pair contains offset and range values for Cb. The third pair contains offset and range values for Cr.

9.2.2.4 MRC3 to MRC8 and 14 to MRC254 entry for future extensions

The entries from MRC3 to MRC8 are to be reserved for future structural marker segments while MRC14 to MRC254 are to be reserved for other future use, such as optional marker segments, encoder marker segments (see Annex A) and reproduction information.

9.2.3 TN (Termination Number)

This is the JPEG EOI (End of Image) to alert decoders to the end of the initial JPEG application markers, registered per ITU-T Rec. T.86. The TN is located after the last SOP parameter (i.e., page width).

TN: 2 octets X'FFD9'

9.3 Stripe data structure

The beginning of a stripe is denoted by the Start of Stripe Marker Segment, followed by stripe data.

The first layer represented is the mask layer, followed by the background layer, and next the foreground layer (as appropriate). When there are two or more layers, the mask layer shall always be one of them. In the case of background only pixel data, no mask or foreground pixel data, the mask shall be fixed to "0". In the case of foreground only pixel data, no mask or background pixel data, the mask shall be fixed to "1".

Start of Stripe Segment has the following structure:

APP13, Length of segment, SOSt ident, type of stripe, background layer base colour, foreground layer base colour, offset of background layer relative to upper left-hand pixel in the stripe, offset of foreground layer relative to upper left-hand pixel in the stripe, stripe height (number of lines), length of coded mask layer in number of octets.

In Mode 1 (base mode) all SOSt parameters must be present (i.e., layer base colour and offset values must be provided for both the foreground and background layers).

The Start of Stripe Marker Segment is defined as follows:

APP13 marker:		X'FFED'
Length of Segment:	2 octets	Length of segment in octets, MSB to LSB, as an integer value not including APP13.
SOSt Ident:	4 octets	'MRC1', represented as a 3-octet ASCII string plus a hexadecimal count (i.e., X'4D', X'52', X'43', X'01'). This X'01'-terminated string "MRC" uniquely identifies this marker segment as the start of stripe.
Type of stripe:	1 or more octet(s)	With value indicating stripe type as shown in Table 3. The corresponding bit shall be set to "1" for each layer present. When there are ≥ 2 layers, the main mask layer must be one of the layers (bit 1 set to 1). A maximum of three layers may be present in this mode (Mode 1.0).
Background layer base colour:	3 octets	Colour encoded using OMSgl for LAB Colour Space or OMSgy for YCC colour space, and the layer base colour gamut range. The default color space is LAB, and if Image layer coder is defined the same colour space as that is used. The value is white X'FF', X'80', X'60' for LAB Colour Space and X'FF', X'80', X'80' for YCC colour space unless otherwise specified. If available, custom gamut range may be applied from the optional marker segments.
Foreground layer base colour:	3 octets	Colour encoded using OMSgl for LAB Colour Space or OMSgy for YCC colour space, and the layer base colour gamut range. The default colour space is LAB, and if Image layer coder is defined the same colour space as that is used.

		The value is black X'00', X'80', X'60' for LAB colour space and X'FF', X'80', X'80' for YCC colour space unless specified otherwise. If available, custom gamut range may be applied from the optional marker segments.
Offset of background layer:	8 octets	Horizontal offset, vertical offset as 2 integer values in mask layer units, as appropriate. Offsets are relative to stripe first scan line and left boundary.
Offset of foreground layer:	8 octets	Horizontal offset, vertical offset as 2 integer values in mask layer units, as appropriate. Offsets are relative to stripe first scan line and left boundary.
Stripe height (lines):	4 octets	Height of stripe as an integer value. For images with two or more layers, the main mask layer height defines the stripe height. For single layer images, a virtual mask layer height defines the stripe height.
Mask layer length (octets):	4 octets	Coded length of the main mask layer as an integer value, when present. This value must be set to zero (0) when there is no coded mask data.

Table 3/T.44 – Type of stripe

Octet bit number	Layer used
LSB 0	Background layer (layer 1)
1	Main mask layer (layer 2)
2	Foreground layer (layer 3)
3	Layer 4
4	Layer 5
5	Layer 6
6	Layer 7
MSB 7	Extend, add another octet that follows immediately
NOTE – Refer to Annex A for stripes with 4 or more layers. Layers above seven (7) would require an additional octet for representation. Bit 7, the extend bit, would be set when adding another octet to accommodate an additional layer such as Layer 8 which would be represented by bit 8.	

9.4 EOP (End of Page)

This End of Page code indicates the end of MRC page.

EOP:	4 octets	X'FFD9', X'FFD9'

9.5 Layer data structure

Layers are coded using ITU-T coding methods indicated in the Start of Page Marker Segment. The coding method and resolution of the background and foreground layers are defined in the layer data. The resolutions of the background and foreground layers are restricted to ITU-T recommended values that must be integral factors of the main mask resolution. For example, if the mask resolution is 400 pels/25.4 mm, the background and foreground layer may each be either 100, 200 or 400 pels/25.4 mm.

Main Mask resolution:	2 octets	Expresses vertical and horizontal resolution as a single integer value in units of pels/25.4 mm. Basic value is 200 pels/25.4 mm. The value shall be fixed to that of the image layer in the event that there is no coded mask data (layer) in the page.

Page width: 4 octets Page width as a single integer value. For pages with two or more layers, the main mask layer image width defines the page width in units of main mask resolution. For pages with only a single layer foreground or background image, no coded mask data, a virtual mask (i.e., a mask layer without coded data) shall be used to define the page width.

9.6 Data format summary

9.6.1 High-level data format summary

<table>
<tr><td rowspan="2">SOP</td><td>SOP marker</td><td colspan="3">X'FFD8', X'FFED', Length, MRC0</td></tr>
<tr><td>Parameters</td><td colspan="3">Version, Mode, ...</td></tr>
<tr><td>TN</td><td colspan="4">X'FFD9'</td></tr>
<tr><td>OMSgl</td><td>OMSgl marker</td><td>X'FFED', Length, MRC10</td><td colspan="2">parameters</td></tr>
<tr><td>OMSi</td><td>OMSi marker</td><td>X'FFED', Length, MRC11</td><td colspan="2">parameters</td></tr>
<tr><td>OMSgy</td><td>OMSgy marker</td><td>X'FFED', Length, MRC9</td><td colspan="2">parameters</td></tr>
<tr><td rowspan="8">Page data</td><td rowspan="5">Stripe 1</td><td rowspan="2">SOSt</td><td>SOSt marker</td><td>X'FFED', Length, MRC1</td></tr>
<tr><td>Parameters</td><td>Type, B.G. layer base colour, ...</td></tr>
<tr><td rowspan="3">Stripe data</td><td>Mask layer</td><td>(layer data)</td></tr>
<tr><td>B.G. layer</td><td>(layer data)</td></tr>
<tr><td>F.G. layer</td><td>(layer data)</td></tr>
<tr><td>... ...</td><td colspan="3"></td></tr>
<tr><td rowspan="2">Stripe N</td><td colspan="3">SOSt</td></tr>
<tr><td colspan="3">Stripe data</td></tr>
<tr><td>EOP</td><td colspan="4">X'FFD9 FFD9'</td></tr>
</table>

9.6.2 Detail data format summary

MRC Magic Number
SOP marker segment
APP13 marker
Length of Segment
MRC0 SOP identifier
    Version
    Mode
    Mask coder
    Image layer coder
    Mask resolution
    Page width
TN

Layer base colour gamut marker segments
APP13
Length of Segment
MRC10 OMSgl identifier
    gamut range data

Layer base colour illuminant marker segments
APP13
Length of Segment
MRC11 OMSi identifier
    Illuminant data

OMSx (Optional marker segments)
APP13
Length of Segment
MRCn (n = 12 to 254) OMSx identifier
    optional maker segment data

Stripe 1
    SOSt marker segment
    APP13 marker
    Length of Segment
    MRC1 SOSt identifier
        Type of stripe
        Background layer base colour
        Foreground layer base colour
        Offset of background layer
        Offset of foreground layer
        Stripe height (lines)
        Mask layer length (octets), when appropriate
Stripe data
    Mask layer
        Layer coded data ----------

    Background layer
        Layer coded data ----------

    Foreground layer
        Layer coded data ----------

Stripe 2
    SOSt marker segment
    APP13 marker

Stripe data
    Mask layer
        Layer coded data ----------

    Background layer
        Layer coded data ----------

    Foreground layer
        Layer coded data ----------

Stripe 3

Stripe N

EOP (X'FFD9', X'FFD9')

Annex A

Mixed Raster Content (MRC) Modes 2 and 3

A.1 Scope

This annex defines Modes 2 and 3 to ITU-T Rec. T.44. Mode 2 adds SLC (start of layer coded data segment) support to the 3-layer model defined in Mode 1. Mode 3 adds SLC support and extends the model beyond three (3) layers to realize greater capability. Applications implementing Mode 2 shall support Mode 1 while applications supporting Mode 3 shall support modes 1 and 2. As with T.44 Mode 1, this annex does not define new encodings or resolutions. The method of image segmentation is beyond the scope of this annex, segmentation is left to the manufacturer's implementations.

A.2 References

See the main body of this Recommendation.

A.3 Definitions

The definitions of the main body of this Recommendation apply, plus the following definitions:

A.3.1 Encoder specific marker segment (EMSe): Encoded as APP13 (X'FFED'), length of segment, EMSe ident (MRC12 to 254), parameters/data. This category of marker segments provides information that is specific to the encoding/decoding of the image. These marker segments are not always present since they are encoder dependent. When present, parsing of the EMSe is required for proper decoding of the layer data stream for which they are defined, unless otherwise stated.

A.3.2 End of header marker segment (EOH): Encoded as APP13 (X'FFED'), length of segment, EOH ident (MRC255), parameters.

A.3.3 Start of layer coded marker segment (SLC): Encoded as APP13 (X'FFED'), length of segment, SLC ident (MRC2), parameters.

A.4 Conventions

See the main body of this Recommendation.

A.5 Image representation

This annex includes description of syntax for encapsulating two or more ITU-T encodings on a single page. The base mode is mandatory and shall be supported by this mode.

A page is composed from a set of page-wide stripes of image data. The stripes are transmitted sequentially from the top to the bottom of the page.

The stripes are composed of one or more layers. Each layer is coded using a recommended ITU-T coding method.

Information required to decode the page, such as coding types used within the layers, is specified within the page header (start of page marker segment). Stripe height is specified within the stripe header (start of stripe marker segment).

Information required to decode a layer is included in the stripe header and the layer data.

The main mask layer is transmitted first, followed by the background layer, followed by the foreground layer, and then any subsequent layer in increasing numeric order.

Details of the syntax are described below.

A.6 Stripe structure

Stripes are composed of one or more layers; background layer (layer 1), main mask layer (layer 2) foreground layer (layer 3), a series of overlay mask layers (even-numbered layers 4, 6, 8, …) and image layers (odd-numbered layers 5, 7, …). One or more images may be assigned a fixed colour value, while mask layers may be assigned a fixed bit value (i.e., 0, selecting the background or 1, selecting the foreground layer). Virtual mask layers and fixed value layers are not counted in classifying the stripe types as follows:

N layer stripe	(NLS)
:	
:	
Three-layer stripe	(3LS)
Two-layer stripe	(LS)
One-layer stripe	(1LS)

A.6.1 Three-layer stripe (3LS)

See the main body of this Recommendation.

A.6.2 Two-layer stripe (2LS)

See the main body of this Recommendation.

A.6.3 One-layer stripe (1LS)

See the main body of this Recommendation.

A.6.4 N-layer stripe (NLS)

N-layer stripes (NLS), where N is an integer, are an extension to the basic structure of ITU-T Rec. T.44, as defined in this annex. The NLS contains more than three (3) layers; see Figure A.1. It provides a means to transfer one or more multi-level image layers (background, foreground, layer 5, layer 7, …) and one or more bi-level mask layers (layers 2, 4, 6, …) that define layer recombination on the same page. Beyond layer 1 (background), the layers occur in pairs, 2 and 3, 4 and 5, etc. The main mask layer (layer 2) must span the full dimension of the stripe while other layers (i.e., layer 1, 3, 4, 5, …) may have an offset and dimensions that are less than those of the stripe. The offset and dimensions of the masks need not be the same as those of the corresponding image layers, see Figure A.1. This capability enables representation of richly coloured text, graphics, and line-art together with contone image using a combination of multi-level and bi-level coding methods.

A.7 Image coding

A.7.1 Spatial resolution

The resolution of the main mask layer is fixed for the entire page and defines the maximum resolution for the page. In general it is possible to define lower spatial resolution for other layers. Spatial resolution of all layers must be integral factors of the main mask resolution. All resolutions used must be square (i.e., same horizontal and vertical values) and conform to ITU-T recommended values. The main mask resolution is specified in the page header. The resolution of other layers is specified in the stripe data stream.

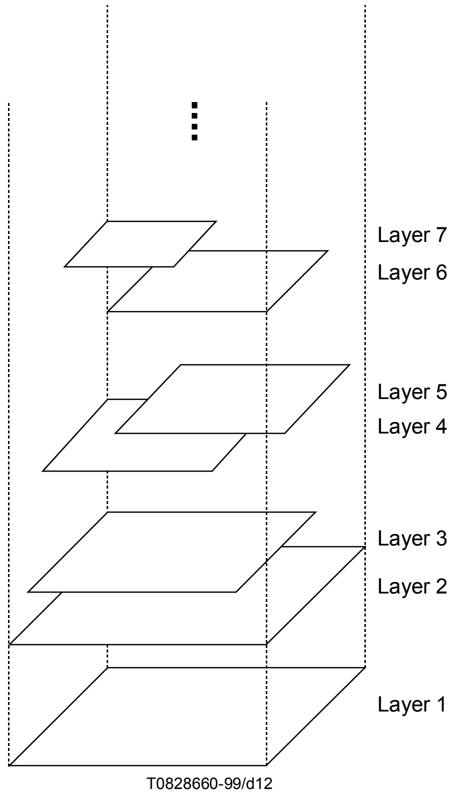

Figure A.1/T.44 – Mask and image layers in N-layer stripe

A.7.2 Stripe width and layer width

Stripes always span the entire width of a page. The main mask layer must always span the entire width.

This method takes advantage of the image width and height data included in the layer data stream. Layers other than the main mask are not required to span the entire width. All layers must be fully contained within the boundaries of the stripe. The width value of the mask layers (even-numbered layers) and the width value of the corresponding image layers (odd-numbered layers) are independent. In addition, for layers other than the main mask layer, a horizontal offset may be used to select a starting point to the right of the left stripe boundary. The offset value of the mask layers (even-numbered layers) and the offset value of the corresponding image layers (odd-numbered layers) are independent. This offset is expressed in the main mask pixel units in the layer data. A simple stripe containing only background (e.g., JPEG data) or foreground (e.g., T.43 JBIG data) image data may use this feature also.

A.7.3 Stripe height and layer height

To limit the data that must be buffered by an application, some applications may choose to limit the maximum height of two or more layer stripes (2LS to NLS) to a specified number of lines (in main mask layer resolution).

One-layer stripes (1LS) are not required to conform to a maximum stripe height, and are only limited by page size. Layers without coded data (i.e., virtual mask layers and image layers with only base layer colour) are not to be counted when considering whether a stripe is a 1LS, 2LS, 3LS or NLS stripe.

Stripe height and main mask layer height are always equal. Heights of layers, other than the main mask, are less than or equal to stripe heights. All layers must be fully contained within the boundaries of the stripe. The height value of the mask layers (even-numbered layers) and the height value of the corresponding image layers (odd-numbered layers) are independent. In addition, for layers other than the main mask layer, a vertical offset may be used to select a starting point below the first scan line of the stripe. The offset value of the mask layers (even-numbered layers) and the offset value of the corresponding image layers (odd-numbered layers) are independent. This offset is expressed relative to the first scan line at the top of the stripe and in the main mask pixel units. A simple stripe containing only background (e.g., JPEG) or foreground (e.g., T.43 JBIG) data may use this feature also.

A.7.4 Layer combination

Image layers are rendered sequentially in ascending order of layer number (i.e., layer 1, then 3, then 5, … then N). The background layer (i.e., layer 1), if present, shall be rendered first. Bi-level mask layers (even-numbered layers, such as layer 2) select pixels from their corresponding image layer (odd-numbered layer directly above the mask layer, such as layer 3) for rendering. A corresponding image layer pixel (directly above the mask layer pixel), or its layer base colour value, is selected when a mask pixel value is "1". The selected image layer pixel is rendered on top of any layer or layers that may have been previously rendered. A corresponding image layer pixel shall not be rendered when the mask pixel value is "0". The pixel from the layer or layer combination below the mask, or its layer base colour value, shall remain visible when the mask pixel value is "0". In the event of an image layer, or portion thereof, without a corresponding mask layer, the image layer shall be rendered on top of any previously rendered layers.

A.8 Layer transmission order

In NLS, the bi-level main mask data is transmitted first, followed by the background (layer 1), foreground (layer 3), layer 4, layer 5, …, layer N. In NLS without a background layer, the bi-level main mask image data is transmitted first, followed by the foreground layer, layer 4, layer 5, …, layer N.

A.9 Data format

A.9.1 Overview

The MRC image data consist of a series of: markers; parameters; data that specify the image coder, image size, bit resolution and spatial resolution; coded image data. The conventions of Annex B/T.81 are used broadly here. The JPEG registration body, per ITU-T Rec. T.86, has been used to register the marker code, APP13, classified as an application marker.

~~Markers and/or marker segments used in association with the coding methods (i.e., encoder marker and/or marker segments) may be defined outside of this Recommendation (i.e., foreign encoder markers and/or marker segments). Foreign encoder markers and/or marker segments may be located within or outside of the data stream. A foreign encoder marker located outside the data stream shall be of the APPn form (i.e., an X'FF octet followed by an octet not equal to X'00' or X'FF' and optionally preceded by extra X'FF' octet codes). Structure of a foreign encoder marker segment located outside the data stream shall be as follows:~~

~~APPn, Length of segment, identifier, parameter and/or data.~~

The MRC page structure for this application has the following elements: Parameters, markers, and entropy-coded data segments. Parameters and markers are often organized into marker segments. Parameters are integers of length ½, 1, 2 or more octets. Markers are assigned two or more octet codes, an X'FF' octet followed by an octet not equal to X'00' or X'FF' and optionally preceded by extra X'FF' octet codes. This application defines marker segments to denote the start of page (SOP), optional marker segments, the start of a stripe (SOSt), the start of layer coded data (SLC), encoder marker segments and end of header (EOH). The MRC Magic Number (JPEG SOI) is used immediately preceding the application marker as part of the SOP marker segment. The JPEG EOI is used as a termination number located directly after the last SOP parameter. The end of a page (EOP) is defined as X'FFD9FFD9'. These markers are inserted by the encoder, and understood by the decoder in addition to all markers used for the coding methods such as the SOI of ITU-T Rec. T.81.

Markers and/or marker segments used in association with the coding methods (i.e., encoder marker and/or marker segments) may be defined outside of this Recommendation (i.e., foreign encoder markers and/or marker segments). Foreign encoder markers and/or marker segments may be located within or outside of the data stream. A foreign encoder marker located outside the data stream shall be of the APPn form (i.e., an X'FF octet followed by an octet not equal to X'00' or X'FF' and optionally preceded by extra X'FF' octet codes). Structure of a foreign encoder marker segment located outside the data stream shall be as follows:

APPn, Length of segment, identifier, parameter and/or data.

A.9.2 Page data structure

The beginning of an MRC page is denoted by the Start of Page Marker Segment, followed by optional marker segment(s), Termination Number, page data, and EOP. The parameters of the optional marker segments are optional, unless otherwise stated. Their purpose is to provide insight into reproduction of the image and as such are typically not mandatory for image reproduction. Skipping of any unrecognized optional marker segment is appropriate. Page data consists of stripes 1 to n (where n is an integer), as described in the next subclause.

A.9.2.1 Start of page marker segment

Start of Page Marker Segment has the following structure:

MRC Magic Number, APP13, Length of segment, SOP ident, version, mask coders, image layer coders, main mask resolution, width.

The Start of Page Marker Segment is defined as follows:

MRC Magic Number:	2 octets	X'FFD8'
APP13 marker:	2 octets	X'FFED'
Length of Segment:	2 octets	Length of segment in octets, MSB to LSB, as an integer value including the octet count itself but not including Magic Number or APP13.
SOP Ident:	4 octets	'MRC0', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D', X'52', X'43', X'00'). This X'00'-terminated string "MRC" uniquely identifies this marker segment as the start of page.
Version:	1 octet	Revision number, X'00' indicting revision "0".
Mode:	1 octet	X'02', indicating Mode 2.0. Each mode identifies a different level of performance. Mode 2.0 identifies the SLC (start of layer coded marker segment) supported 3-layer mode of T.44 as defined by the contents of this annex. Applications supporting this mode shall support the capabilities defined in Mode 1.0. X'03', indicating Mode 3.0. Each mode identifies a different level of performance. Mode 3.0 identifies the SLC (start of layer coded marker segment) supported N-layer mode of T.44 as defined by the contents of this annex. Applications supporting this mode shall support the capabilities defined in Modes 1.0 and 2.0.
Mask coders:	1 or more octet(s)	With value indicating coder as shown in Table 1. The identified coders may be used in any mask layer. Only one coder shall be used in the main mask layer. Only one mask layer is present in Mode 2.0. More than one mask layer (i.e., main mask, layer 2, plus other even-numbered layers) may be present in Mode 3.0. The value shall be fixed to zero "0" in the event that there is no mask layer coder.
Image layer coders:	1 or more octet(s)	With value indicating coders as shown in Table 2. The identified coders may be used in any image layer. Only two image layers may be present in Mode 2.0. There is no restriction to the number of image layers that may be present in Mode 3.0 (i.e., image layers, layer 1, 3, plus other odd-numbered layers). The value shall be fixed to zero "0" in the event that there is no image layer coder.
Main Mask resolution:	2 octets	Expressing vertical and horizontal resolution as a single integer value in units of pels/25.4 mm. Basic value is 200 pels/25.4 mm. The value shall be fixed to that of the image layer in the event that there is no coded mask data (layer) in the page.

Page width:	4 octets	Expressing page width as a single integer value. For pages with two or more layers, the main mask layer image width defines the page width in units of main mask resolution. For pages with only a single layer foreground or background image, no coded mask data, a virtual mask (i.e., a mask layer without coded data) shall be used to define the page width.

A.9.2.2 Optional marker segments

See the main body of this Recommendation.

A.9.2.3 TN (Termination Number)

See the main body of this Recommendation.

A.9.3 Stripe data structure

The beginning of a stripe is denoted by the Start of Stripe Marker Segment, followed by stripe data.

The first layer represented is the main mask (layer 2), followed by the background (layer 1), followed by the foreground (layer 3), followed by layer 4, followed by layer 5, …, followed by layer N (as appropriate). When there are two or more layers, the main mask layer shall always be one of them. The stripe height is determined by the height of the first layer within the stripe.

Start of Stripe marker segment has the following structure:

APP13, Length of segment, SOSt ident, type of stripe.

The Start of Stripe Marker Segment is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length of Segment:	2 octets	Length of segment in octets, MSB to LSB, as an integer value not including APP13.
SOSt Ident:	4 octets	'MRC1', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D', X'52', X'43', X'01'). This X'01'-terminated string "MRC" uniquely identifies this marker segment as the start of stripe.
Type of stripe:	1 or more octet(s)	With value indicating stripe type as shown in Table 3. The corresponding bit shall be set to "1" for each layer present. When there are ≥ 2 layers, the main mask layer shall be one of the layers (bit 1 set to 1).

A.9.4 EOP

See the main body of this Recommendation.

A.9.5 Layer data structure

Layers are coded using ITU-T coding methods indicated in the Start of Page marker segment. A Start of Layer Coded Data (SLC) marker segment precedes the coded layer data. Parameters of the SLC include layer number, coder, resolution, coded image width and height, layer base colour and layer offset. One or more marker segments that contain encoding related parameters might follow the SLC. New encoder related marker segments may be defined as determined by encoding needs. They may be defined within or outside of this Recommendation. Those defined outside of this Recommendation are frequently referenced as foreign encoder marker segments. End of Header (EOH) marker segment terminates the SLC. The EOH contains the coded data length (octet count) of the layer. Encoder marker segments shall be located between the SLC and EOH. The resolutions of all layers are restricted to ITU-T recommended integral factors of the main mask resolution. For example, if the main mask resolution is 400 pels/25.4 mm, the other layers may each be either 100, 200 or 400 pels/25.4 mm.

A.9.5.1 Start of Layer Coded Data (SLC) marker segment

The start of layer coded data is uniquely identified by the SLC marker segment. This marker segment is mandatory for all layers. Structure of the SLC marker segment is as follows:

APP13, length, SLC ident ('MRC2'), layer number, coder, resolution, layer width, layer height, layer base colour, layer offset.

The marker segment is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length:	2 octets	Total entry field octet count, MSB to LSB, including the octet count itself. It excludes the APP13 marker, any other marker segments, such as encoder related marker segments and the EOH.
SLC ident:	4 octets	'MRC2', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D', X'52', X'43', X'02'). This X'02'-terminated string "MRC" uniquely identifies this marker segment as the start of layer coded data.
Layer number:	1 octet	Identifies the number, ordering, of the layer. The first SLC after the SOSt shall always have an even layer number, it is always a mask layer (i.e., a coded mask layer or a virtual mask layer).
Coder:	2 or more octets	Identify presence of coder and coder used, as per Table A.1. Table A.1 identifies the coder by referencing Table 1 or Table 2 and specifying octet bit number via the hexadecimal value of octets 2-N.
Resolution:	2 octets	Expressing vertical and horizontal resolutions as single integer values in units of pels/25.4 mm. Basic value is 200 pels/25.4 mm. For layers with no coded data, the value is set to that of the corresponding mask layer or image layer for image and mask layers respectively. For example, if layer three contains no coded data, then its resolution shall be set to that of layer 2, while if layer two contains no coded data, then its resolution shall be set to that of layer 1 or 3, whichever is present.
Layer width:	4 octets	Width of coded image data (number of pixels) in the layer, in main mask layer resolution unit, as an integer value.
Layer height:	4 octets	Number of scan-lines of coded data in the layer, in main mask layer resolution unit, as an integer value. The stripe height is defined by the layer height contained in the first SLC after the SOSt.
Layer base colour:	3 octets	Colour encoded using OMSgl for LAB Colour Space or OMSgy for YCC colour space, and the layer base colour gamut range. The default colour space is LAB, and if Coder is defined the same colour space as that is used. The value is fixed to X'00', X'00', X'00' for mask layers (even-numbered layers) since these layers do not have colours. For the Background layer (layer 1), the layer base colour is considered to be white X'FF', X'80', X'60' for LAB colour space and X'FF', X'80', X'80' for YCC colour space, unless specified otherwise. The layer base colour gamut and white point, per ITU-T Rec. T.42, are applied.

The optional marker segments, Layer Base Colour Gamut Range (OMSgl) for LAB or (OMSgy) for YCC and Layer Base Colour Illuminant (OMSi) for LAB shall be applied, if present.

For Foreground layers (odd-numbered layers 3 to N), the layer base colour is considered to be black X'00', X'80', X'60' for LAB Colour Space and X'00', X'80', X'80' for YCC colour space, unless specified otherwise. The layer base colour gamut and white point, per ITU-T Rec. T.42, are applied. The optional marker segments, Layer Base Colour Gamut Range (OMSgl) for LAB or (OMSgy) for YCC and Layer Base Colour Illuminant (OMSi) for LAB shall be applied, if present.

Offset:	8 octets	Horizontal offset and vertical offset as 2 integer values in main mask layer units, as appropriate. Offsets are relative to stripe first scan line and left boundary. Both the horizontal and vertical offsets of the main mask layer (layer 2) are fixed to X'00', X'00', X'00', X'00'.

Table A.1/T.44 – Coder identification

Octet 1			Octets 2..N
Octet 1 bit number	Bit value	Definition	Hex value: 'XXX---X'
0	0	No coded data, ignore bit 1	
	1	Bit 1 identifies coded data from Table 1 or Table 2	
1	0	Octets 2-N define octet bit number from Table 1	
	1	Octets 2-N define octet bit number from Table 2	
2-7	Reserved		

A.9.5.2 Encoder related marker segments (EMSe)

The encoder marker segments (EMSe) provide information that is specific to the encoding/decoding of the image. These marker segments are not always present since they are encoder dependent. When present, parsing of encoder marker segment(s) is required for proper decoding of the layer data stream for which they are defined, unless otherwise stated.

Encoder related marker segments consist of marker and associated parameters/data. The APP13 (X'FFED') marker initiates identification of each entry. Each encoder related marker segment is identified by the 3-octet ASCII string plus an hexadecimal count for 'MRCn'. The 'MRCn' identifier is a 4-octet value X'4D',X'52',X'43',X'n', where n may equal X'0C' (12) to a maximum of X'FE' (254).

Each encoder related marker segment has the following structure:

APP13 Marker (X'FFED'), Length of entry, encoder marker segment ident (MRCn), Optional length (if needed), parameters and/or data.

EMSe represents specific encoder marker segments, where "EMS" is replaced by an appropriate acronym used to distinguish each encoder marker segment.

Specific encoder marker segments are defined elsewhere within this Recommendation or outside of this Recommendation. A foreign encoder marker segment shall also have the structure:

APPn, Length of segment, identifier, parameter/data; where the APPn marker consists of an X'FF' octet followed by an octet not equal to X'00' or X'FF' and optionally preceded by extra X'FF' octet codes.

A.9.5.3 End of Header (EOH) marker segment

The EOH marker segment uniquely identifies the end of the SLC and any other marker segment(s) that may be present. This marker segment is mandatory. The EOH shall directly precede the coded data. Structure of the EOH marker segment is as follows:

APP13, length, EOH ident ('MRC255') and coded data length.

The marker segment is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length:	2 octets	Total entry field octet count, MSB to LSB, including the octet count itself. It excludes the APP13 marker.
EOH ident:	4 octets	'MRC255', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D', X'52', X'43', X'FF'). This X'FF'-terminated string "MRC" uniquely identifies this marker segment as the end of header.
Coded data length:	4 octets	Octet count of coded data in the layer.

A.9.6 Data format summary

A.9.6.1 High-level data format summary

<table>
<tr><td>SOP</td><td colspan="3">X'FFD8' X'FFED' Length, MRC0</td><td colspan="2">Version, Mode, ...</td></tr>
<tr><td>TN</td><td colspan="5">X'FFD9'</td></tr>
<tr><td>OMSgl</td><td colspan="3">X'FFED' Length, MRC10</td><td colspan="2">Gamut data for LAB</td></tr>
<tr><td>OMSi</td><td colspan="3">X'FFED' Length, MRC11</td><td colspan="2">Illuminant data for LAB</td></tr>
<tr><td>OMSgy</td><td colspan="3">X'FFED' Length, MRC9</td><td colspan="2">Gamut data for YCC</td></tr>
<tr><td rowspan="14">Page data</td><td rowspan="11">Stripe 1</td><td>SOSt</td><td colspan="2">X'FFED' Length MRC1</td><td>Type</td></tr>
<tr><td rowspan="10">Stripe data</td><td rowspan="3">Layer 2 (L2)</td><td>SLC X'FFED' Length, MRC2</td><td>Layer number, coder, res., width, height, layer base colour, offset</td></tr>
<tr><td>EOH X'FFED' Length, MRC255</td><td>coded data length</td></tr>
<tr><td colspan="2">coded data</td></tr>
<tr><td rowspan="4">L1</td><td colspan="2">SLC</td></tr>
<tr><td>EMSe X'FFED' Length, MRCn</td><td>Parameters/data</td></tr>
<tr><td colspan="2">EOH</td></tr>
<tr><td colspan="2">coded data</td></tr>
<tr><td colspan="3">L 3</td></tr>
<tr><td colspan="3">L 4</td></tr>
<tr><td colspan="3">- - - -</td></tr>
<tr><td colspan="5">.....</td></tr>
<tr><td rowspan="2">Stripe N</td><td colspan="4">SOSt</td></tr>
<tr><td colspan="4">Stripe data</td></tr>
<tr><td>EOP</td><td colspan="5">X'FFD9FFD9'</td></tr>
</table>

A.9.6.2 Detail data format summary

MRC Magic Number
SOP marker segment
APP13 marker
Length of Segment
MRC0 SOP identifier
    Version
    Mode
    Mask coder
    Image layer coder
    Main mask resolution
    Page width
TN

Layer base colour Gamut marker segment
APP13
Length of Segment
MRC10 OMSgl identifier
    gamut range data

Layer base colour Illuminant marker segment
APP13
Length of Segment
MRC11 OMSi identifier
    illuminant data

OMSx (optional marker segments)
APP13
Length of Segment
MRCn (n = 12 to 254) identifier

    …

Page data

Stripe 1
    SOSt marker segment
    APP13 marker
    Length of Segment
    MRC1 SOSt identifier
        Type of stripe

Stripe data
    Mask layer
    SLC marker segment
    APP13 marker
    Length of Segment
    MRC2 SLC identifier
        layer number
        coder
        Resolution
        Layer width
        Layer height
        Layer base colour
        Offset

Encoder marker segments
APP13 marker
Length of Segment
MRCn EMSe identifier
Parameters/data

EOH marker segment
APP13 marker
Length of Segment
MRC255 EOH identifier
coded data length
Layer coded data --------

Background layer
SLC marker segment
:
:
Encoder marker segments
…
…
EOH marker segment
Layer coded data --------

Foreground layer
SLC marker segment
:
:
Encoder marker segments
EOH marker segment
Layer coded data -------

Layer 4
SLC marker segment
:
:
Encoder marker segments
EOH marker segment
Layer coded data --------

Layer 5
SLC marker segment
:
:
Encoder marker segments
EOH marker segment
Layer coded data --------
:
:
Layer N
SLC marker segment
:
:
Encoder marker segments
EOH marker segment
Layer coded data -------

Stripe 2
SOSt marker segment
APP13 marker

Stripe data

Mask layer

Layer coded data --------

Background layer

Layer coded data --------

Foreground layer

Layer coded data -------

Layer 4

Layer coded data --------

Layer 5

Layer coded data --------

:

:

Layer N

Layer coded data -------

Stripe 3

Stripe N

EOP (X'FFD9', X'FFD9')

Annex B

MRC Mode 4 – Shared resources and colour tags

Introduction and background

Improved compression, in terms of both reduced size and less error for lossy methods, is achieved when the compression method closely models and matches the data to be compressed. This is giving rise to a new wave of compression methods that have explicit models of some types of data. MPEG4 includes support for describing objects moving over a static background, something that is seen in video images of the world. ITU-T Rec. T.88 | ISO/IEC 14492 (JBIG2) represents scanned bi-level data by segmenting it into text, halftone, and other regions, and then using separate specialized compression methods to store these regions. The text regions are compressed by extracting symbols (individual text characters), and forming symbol dictionaries. The same symbol shapes (each one representing a text character from a certain font in a certain size) are used across multiple text regions and pages, to improve compression. Halftone regions are similarly represented using dictionaries of halftone patterns.

Using these dictionaries, JBIG2 can achieve a large increase in compression relative to other bi-level image compression methods: 3-5 times greater compression than T.82 (JBIG1) or T.6 (MMR) is typical, and factors of over 20 times better compression than MMR have been observed.

Of course, in order to achieve these large compression factors, each piece of data must be used to the maximum possible extent. This means that a single symbol dictionary must be used by multiple pages wherever possible, which necessarily has consequences for any system using JBIG2, as most systems have considered pages to be completely independent entities.

An imaging model using JBIG2, such as MRC, should incorporate provisions for the use of shared data. This entails: having some way to define a shared resource, to be used by multiple coded entities (pages, stripes or layers); referring to that shared resource at the point where it is to be used; and later instructing the decoder that the resource is no longer required and may be flushed from memory. The shared data marker segment (SDM), introduced in this annex, is intended to provide this functionality.

It should be noted that SDM is not restricted to JBIG2; its structure is flexible and could be used for other coding methods. For example, a set of JPEG Huffman tables could be stored in a shared resource and then used by multiple JPEG-coded layers, reducing file size. Similarly, a palette table could be defined once, then used by multiple T.43-coded layers.

Another opportunity that is afforded by JBIG2 is improved compression of the foreground layer for documents containing coloured text. In most cases, if a document contains text, each individual text character is a single, flat, colour (e.g., black or red), and the number of such colours is limited. The foreground layer in this case looks like a number of coloured blobs, one for each character, each one having the shape of the corresponding character.

This foreground layer can be compressed using a new method that takes advantage of the JBIG2 structure. If the mask layer is compressed using JBIG2 symbols and/or halftone regions, then decoding it essentially yields a sequence of (XPositon, YPosition, Symbol ID) triples. Each triple indicates that the symbol (from some dictionary) specified by "Symbol ID" should be drawn at the location "(X, Y)". Simply augmenting a text region triple with a fourth component, the colour of that individual character (sometimes called the symbols "colour tag"), allows storage of the foreground layer in a very small amount of space: using run-length coding on those colours. The total space taken by the foreground layer can be as small as a few tens of bytes.

For example, if the mask layer contained two characters, an "R" in red and a "B" in blue, then the mask layer would decompress to:

(100, 0, "R")

(120, 0, "B")

and the foreground layer would decompress to:

(#7AD29C) [corresponding to CIELAB (48.0, 65.5, 48.0) using default gamut range]

(#3A9B1D) [corresponding to CIELAB (23.1, 20.4, –52.1) using default gamut range]

or some other suitable representation of the colours, such as indexes into a palette. Matching up the "R" symbol with the colour #7AD29C and drawing the symbol's shape in red gives the correct result. This is a single drawing operation, and is extremely efficient.

Storing the foreground layer in this manner, using colour tags, allows very compact representation, and efficient decoding. However, since the mask layer is transmitted before the foreground layer, the decoder needs to be warned that the upcoming foreground layer is simply a list of colours (one per JBIG2 symbol in the mask layer), not a complete image. For this reason, we need to put a flag in the mask layer that warns the decoder "foreground is compressed using colour tags". The decoder can then defer drawing the mask until the foreground layer has also been decoded.

B.1 Scope

This annex defines Mode 4 to ITU-T Rec. T.44, extending the MRC model to accommodate shared data and provisions for colour tags. The provisions of Mode 4 shall use the Mode 3 structure for implementation. Applications implementing Mode 4 shall support Modes 1, 2 and 3.

B.2 References

The references of the main body of this Recommendation apply, plus the following.

– ITU-T Recommendation T.45 (2000), *Run-length colour encoding.*

– ITU-T Recommendation T.88 (2000) | ISO/IEC 14492:2001, *Information technology – Lossy/lossless coding of bi-level images*. (Commonly referred to as JBIG2 standard.)

– ITU-T Recommendation T.89 (2001), *Application profiles for Recommendation T.88 – Lossy/lossless coding of bi-level images (JBIG2) for facsimile apparatus*.

B.3 Definitions

The definitions in Annex A apply, plus the following additional definitions:

B.3.1 Create shared data marker segment (SDMc), encoded as APP13 (X'FFED'), Length of segment, SDM ident (MRC3), Optional length (if needed), parameters, shared data.

B.3.2 Colour-interpreter encoder marker segment (CLIe), encoded as APP13 (X'FFED'), Length of segment, CLIe ident (MRC13), parameters.

B.3.3 Disposition shared data marker segment (SDMd), encoded as APP13 (X'FFED'), Length of segment, SDMd ident (MRC4), parameters.

B.3.4 Generic region: A region that codes pixels individually or in runs – a non-text or non-halftone region.

B.3.5 Halftone region: A region containing halftone patterns that is coded by drawing a set of patterns into a bitmap, placing the patterns according to a halftone grid.

B.3.6 Meta-data: Coding data external to the coded data stream that is required in the interpretation of the data stream and may be shared between pages and other document entities.

B.3.7 JBIG2 encoder marker segment (JB2e), encoded as APP13 (X'FFED'), Length of segment, JB2e ident (MRC12), parameters.

B.3.8 JBIG: Joint Bi-level Image Experts Group and also shorthand for the encoding methods, JBIG1 and JBIG2 described in ITU-T Recs T.82 and T.88 respectively, which were defined by this group.

B.3.9 Refinement region: A region that codes pixels by modifying a reference bitmap to produce an output bitmap.

B.3.10 Text region: A region containing text characters that is coded by drawing a set of symbol instances into a bitmap.

B.4 Shared data

JBIG2 compresses text regions by extracting symbols (individual text characters), and forming symbol dictionaries. The same symbol shapes (each one representing a text character from a certain font in a certain size) are used across multiple text regions and pages, to improve compression. Halftone regions are similarly represented using dictionaries of halftone patterns. The symbol dictionaries that are used over multiple regions and pages are referenced as shared data or shared resources. Using a single symbol dictionary over multiple pages, wherever possible, maximizes compression. The practice of using data over multiple pages contrasts with that of most systems where pages are considered to be completely independent entities.

MRC makes provision for shared resources by introducing three new functions:

1) The "create" function is used to establish a set of shared data. For future access, an ID (identification number) is assigned to the shared data when it is created/defined.

   There are four flags assigned to the "create" function to indicate the scope of application of the shared data:

   a) The "global" flag is used to indicate that the shared data is available for application over the entire document, across multiple pages.

   b) The "page" flag is used to indicate that the shared data is available for application over the rest of the current page, across multiple stripes.

   c) The "stripe" flag is used to indicate that the shared data is available for application over the rest of the current stripe, across multiple layers.

   d) The "layer" flag is used to indicate that the shared data is available for application over the current layer.

The scope flags are intended to reduce the need for the "forget" function, see item 2 below. A scope of "global" implies that the shared data are to be retained until the end of the document data stream, or until a "forget" or "use/forget" function indicates that they may be discarded, whichever comes first. A scope of "page" implies that the shared data can be discarded once the next EOP is encountered, or a "forget" or "use/forget" function combination indicates that they may be discarded, whichever comes first. A scope of "stripe" implies that the shared data can be discarded once the next SOSt is encountered, or a "forget" or "use/forget" function indicates that they may be discarded, whichever comes first. A scope of "layer" implies that the shared data can be discarded once the next SLC is encountered, or a "forget" or "use/forget" function combination indicates that they may be discarded, whichever comes first.

2) The "forget" function is used to instruct the decoder that the identified share data is no longer required and may be flushed from memory. The "forget" function may be applied to one or more sets of shared data at a time by referencing one or more shared data ID.

3) The "use" function is used to instruct the decoder to implement the identified share data in its decode operation. The "use" function may be applied to one or more sets of shared data at a time by referencing one or more shared data ID.

The "use" and "forget" functions may be used independently or in combination with each other. When used in combination, the decoder is instructed to use the identified shared data resources for that layer and then flush from memory.

B.5 Colour tags

In most cases, if a document contains text, each individual text character is a single, flat, colour (e.g., black or red), and the number of such colours is limited. The foreground layer in this case looks like a number of coloured blobs, one for each character, each one having the shape of the corresponding character.

Colour tags can be taken advantage of for documents containing coloured text based on its: improved compression of foreground image layers (odd numbered layers ≥ 3), high speed encode and decode, ease of transcode to PDLs (printer description languages). If the corresponding mask layer is compressed using JBIG2, then decoding it essentially yields a sequence of (X, Y, Symbol ID) triples. Each triple indicates that the symbol (from some dictionary) specified by "Symbol ID" should be drawn at the location "(X, Y)". Simply augmenting this triple with a fourth component, the colour of that individual character (sometimes called the symbols "colour tag"), allows storage of the foreground layer in a very small amount of space. The foreground is represented by a T.45 run-length encoded list of colours, one per JBIG2 symbol in the mask layer. The colours may be represented by discrete colours (i.e., in CIELAB or ITU-YCC space), indexed colours, such as in palette tables or 1 bit/component RGB/CMY(K).

A JBIG2-coded mask layer, within a stripe, might contain any combination of generic, halftone, refinement or text regions. Colour tags may only be used in association with JBIG2 mask layers, within a stripe, that contain only text regions (as colour tags "attach" to text). The foreground associated with a mask layer containing generic, refinement or halftone regions (or text and generic or text and halftone or text and refinement regions) shall be coded in the traditional MRC manner (i.e., using a multi-level coder). In other words, colour tags may only be used with foregrounds that are associated with mask layers within a stripe that contains only text regions.

B.5.1 Mask generation (rendering) when JBIG2 encoded

ITU-T Rec. T.88 defines a collection of encoding parameters and components, which are mixed and matched to generate various application profiles. Generation of the mask layer data stream will require knowledge of the specific profile used during JBIG2 encoding. Additionally, if colour tags are used to encode the foreground associated with a JBIG2 encoded mask it will be necessary to know this information prior to decoding the mask.

Clause B.6.3 makes provision for a T88Options function, used to notify the decoder of: the JBIG2 profile used, whether colour tags are used, any other parameters and/or data required to decode the data stream. The T88Options function uses a series of flag bits to identify each option.

The "tags to follow" options flag bit is used to alert JBIG2 decoders to defer drawing the mask layer until the foreground layer has also been decoded. Given that mask layers are transmitted before the corresponding foreground layers, the decoder needs to be warned that the upcoming foreground layer is simply a list of colours (one per JBIG2 symbol in the mask layer), not a complete image. For this reason, the "tags to follow" flag bit is needed in the mask layer to warn the decoder that the foreground is compressed using colour tags. The decoder can then defer drawing the mask until the foreground layer has also been decoded.

B.5.2 Foreground generation (rendering) when mask is JBIG2 encoded

If the mask layer is coded with JBIG2 and the foreground layer is coded with T.45, then the foreground image is the image obtained by:

- decoding the text regions in the mask layer into a list of (X, Y, Symbol ID) triples; ordered as found in the JBIG2 data;
- decoding the T.45 data into a list of corresponding colour values (CVAL); this list shall have the same number of elements as the list of triples;
- matching each (X, Y, Symbol ID) triple with the corresponding colour value, giving a list of (X, Y, Symbol ID, CVAL) quads;
- drawing these quads, in order from first to last, into the foreground image.

Thus, if two symbol instances overlap, the colour from the latter symbol instance overwrites the colour from the first one.

B.6 Data format

B.6.1 Overview

The second paragraph of A.9.1 is augmented to add shared data marker segment (SDMx) and a series of encoder marker segments (EMSe). The paragraph now reads as follows:

The MRC page structure for this application has the following elements: parameters, markers, and entropy-coded data segments. Parameters and markers are often organized into marker segments. Parameters are integers of length ½, 1, 2 or more octets. Markers are assigned two or more octet codes, an X'FF octet followed by an octet not equal to X'00' or X'FF' and optionally preceded by extra X'FF' octet codes. This application accommodates marker segments to denote the start of page (SOP), additional optional marker segments (OMSx), the start of a stripe (SOSt), the shared data marker segment (SDMx), the start of layer coded data (SLC), encoder related marker segments such as JBIG2 encoder marker segment (JB2e) and colour-interpreter encoder marker segment (CLIe), and end of header marker segment (EOH). The SDMx, JB2e and CLIe marker segments are defined in Annex B. The MRC Magic Number (i.e., JPEG SOI) is used immediately preceding the application marker as part of the SOP marker segment. Prior to the first SOSt, the JPEG EOI is used as a termination number. The end of a page (EOP) is defined as X'FFD9FFD9'. These markers are inserted by the encoder, and understood by the decoder in addition to all markers used for the coding methods, such as SOS (start of scan) of ITU-T Rec. T.81.

NOTE – Entire JBIG2 encoded data streams (i.e., including JBIG2 stripes and headers) are inserted directly following the EOH marker segment.

B.6.2 Start of page marker segment

Start of page marker segment is defined per Mode 3 of Annex A with revision to the "Mode:" description to read as follows:

Mode	1 octet	X'04', indicating Mode 4. Each mode identifies a different level of performance. Mode 4 identifies mandatory writer and reader provision for SDMx (shared data marker segment) along with optional writer and mandatory reader provision for colour tags. The SDMx and colour tag provisions shall be used in conjunction with the SLC (start of layer coded marker segment) supported N-layer mode of T.44 as defined by Mode 3 of Annex A. Applications supporting Mode 4 shall support the capabilities defined in Mode 3.

B.6.3 JBIG2 encoder marker segment (JB2e), MRC12 entry

This entry specifies parameters and/or data that are required in decoding a JBIG2 encoded data stream. It is used to specify the JBIG2 profile and whether colour tags or any future JBIG2 options are used in the data stream. The JBIG2 facsimile profiles are defined in ITU-T Rec. T.89. Structure of JB2e entry is as follows:

APP13, length, JB2e ident., parameters/data.

The JB2e encoder marker segment is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length of Segment:	2 octets	Length of segment in octets, MSB to LSB, as an integer value not including APP13.
JB2e Ident:	4 octets	'MRC12', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D',X'52',X'43',X'0C'). This X'0C'-terminated string "MRC" uniquely identifies this marker segment as the JBIG2 encoder marker segment.
T88Options:	1 or more octet(s)	With bit setting indicating applied option(s) flag, as shown in Table B.1. More than one bit may be set to indicate a combination of T88Options.

Table B.1/T.44 – T88Options octet(s)

Octet bit number	T88Options definition
LSB 0	JBIG2 Fax Profile 1 per ITU-T Rec. T.89
1	JBIG2 Fax Profile 2 per ITU-T Rec. T.89
2	JBIG2 Fax Profile 3 per ITU-T Rec. T.89
3	Reserved for JBIG2 Fax Profile to be defined in ITU-T Rec. T.89
4	Reserved for JBIG2 Fax Profile to be defined in ITU-T Rec. T.89
5	Reserved for JBIG2 Fax Profile to be defined in ITU-T Rec. T.89
6	Tags to follow – Used to alert the JBIG2 decoder to defer drawing the mask layer until the foreground layer has also been decoded. This is applied when the foreground colours are represented with colour tags. (Note 1.)
MSB 7	Extend, add another octet that follows immediately
NOTE 1 – If this bit is set, then JBIG2 Fax Profile 2 must be used (i.e., bit #1 must also be set). NOTE 2 – Bit 7, the extend bit, would be set when adding another octet to accommodate additional options, such as an 8th which would be assigned to bit number 8. NOTE 3 – Options for additional JBIG2 fax profiles will be added to bits 3 through 5 as they are defined in ITU-T Rec. T.89.	

B.6.4 Shared data marker (SDMx) segment

This marker segment makes provision for the use of shared data, shared resources. It provides a means to: create/define a shared resource that is available to be used by multiple coded entities (i.e., pages, stripes or layers); refer to a shared resource at the point where it is to be used; instruct the decoder that the shared resource is no longer required and may be flushed from memory. In applications of JBIG2 encoding, the SDMx is typically used in mask layers (even numbered layers). The SDMx may be located prior to SOSt marker segments (i.e., between the TN and the first SOSt or between end of coded data stream for one stripe and the SOSt for the next stripe), between SOSt and SLC, between SLC and EOH, between end of coded data for one layer and SLC for the next layer, or between end of stripe data and EOP. More than one SDMx may appear at the same location within the page structure.

Structure of each SDMx entry is as follows:

APP13 (X'FFED'), length, SDMx ident, Optional length (if needed), parameters/data.

SDMx represents specific shared data marker segments, where "x" is replaced by a specific character used to represent each shared data marker segment.

B.6.4.1 Create shared data marker (SDMc) segment

The SDMc is used in creating/defining a shared resource that is available to be used by multiple coded entities (i.e., pages, stripes or layers). Only one set of shared data may be created per SDMc.

The Create Shared Data Marker Segment (SDMc) is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length of Segment:	2 octets	Length of segment including length of shared data stream in octets, MSB to LSB, as an integer value not including APP13. Note that if two octets are insufficient, that the Optional length will be used, and this Length of Segment value will be zero.
SDMc Ident:	4 octets	'MRC3', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D',X'52',X'43',X'03'). This X'03'-terminated string "MRC" uniquely identifies this marker segment as the Create Shared Data Marker.
Optional length If (if needed)	4 octets	Length of segment when two octets are not sufficient. If used, length of segment should have a value of zero.
ID:	4 octets	With value providing unique identification of the shared data being created. IDs are unique, and as such their values shall not be reassigned within the document.
Scope:	1 octet	With value indicating breadth of application, as shown in Table B.2, of the shared data being created/defined.
Shared data stream	(Length of segment – 11 or 15 octets)	Data stream being created by this marker segment and referenced by the above ID.

Table B.2/T.44 – Share data scope octet

Octet value	Share data scope
0	Global – Shared data is available for application across the entire document (i.e., over more than one page).
1	Page – Shared data is available for application across an entire page (i.e., over more than one stripe).
2	Stripe – Shared data is available for application across an entire stripe (i.e., over more than one layer).
3	Layer – Shared data is available for application across an entire layer within a stripe (i.e., over one layer of a stripe).
4-255	Reserved

B.6.4.2 Disposition Shared Data Marker (SDMd) segment

The SDMd is used to: refer to a shared resource at the point where it is to be used; and/or instruct the decoder that the shared resource is no longer required and may be flushed from memory.
The Disposition Shared Data Marker Segment (SDMd) is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length of Segment:	2 octets	Length of segment in octets, MSB to LSB, as an integer value not including APP13 and the data when present. Note that two octets should be sufficient for the length, although the use of Optional length may still be used, if necessary.

SDMd Ident:	4 octets	'MRC4', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D',X'52',X'43',X'04'). This X'04'-terminated string "MRC" uniquely identifies this marker segment as the Disposition Shared Data Marker.
Disposition:	1 octet	With value indicating disposition, as shown in Table B.3, of the shared data referenced by the associated ID(s). The corresponding bit shall be set to "1" for each disposition type being applied. The "forget" and "use" dispositions may be used independently or in combination with each other. They may also apply to one or more "ID(s)" at a time. The "forget" disposition signals deletion of the shared data(s) referenced by the ID(s) below. The "Use" disposition signals application of the shared data(s) referenced by the ID(s) below. Combining the "forget" and "use" dispositions signals application and deletion at the end of the layer data stream of the shared data(s) referenced by the ID(s) below.
Count:	2 octets	With value indicating number of share data ID(s) being addressed by the "forget" and/or "use" disposition commands.
ID:	4 × Count	With value(s) identifying the shared data being addressed by the disposition commands.

Table B.3/T.44 – Share data disposition octet(s)

Octet bit number	Share data disposition
LSB 0	Use – apply shared data to the following layer
1	Forget – discard shared data: immediately, in the case that the "use" bit is not set; or after the layer, in the case that the "use" bit is set
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
MSB 7	Reserved
NOTE – New disposition commands (i.e., 3rd through 8th command) would be assigned bit numbers 2 through 7 respectively.	

B.6.5 Interpretation and representation of run-length colour encoded data

Colours values (CVAL) from T.45 "Run-length Colour Encoding" encoded layers shall be interpreted using parameters of the Start of Layer Coded Data (SLC) marker segment, defined in A.9.5.1, the Colour-interpreter Encoder Marker Segment (CLIe) and any foreign encoder marker segments (i.e., encoder marker segments defined outside of this Recommendation) appearing between the SLC and EOH pair. The CLIe and foreign encoder marker segments shall and may, respectively, be included to provide complete colour interpretation. CLIe is specified within this annex while foreign encoder marker segments, used in the specification of gamut range, illuminant data and palette data, are defined outside of this annex. As per Annex A, the CLIe and any foreign encoder marker segments shall be located between the SLC and the EOH (End of Header) pair. The EOH is immediately followed by the run-length encoded layer data.

There are a number of SLC parameters that are not required when ITU-T Rec. T.45 is used for colour value encoding. The value of the following unnecessary SLC parameters shall be set to "0" (zero) when ITU-T Rec. T.45 is used:

- resolution;
- width;
- height;

- layer base colour;
- offset.

The layer number and coder are the only SLC parameters that contain valid information when ITU-T Rec. T.45 is used.

B.6.5.1 Colour-interpreter Encoder Marker Segment (CLIe)

The CLIe identifies colour-encoding rules and is required to interpret T.45 encoded colour values (CVAL). The CLIe may be used for the interpretation of colour values that have been encoded with other encoders. This marker segment is mandatory for all run-length colour encoded layers. Structure of the CLIe is as follows:

APP13, length, CLIe ident. ('MRC13'), colour interpreter.

The marker segment is defined as follows:

APP13 marker:	2 octets	X'FFED'
Length of Segment:	2 octets	Total entry field octet count, MSB to LSB, including the octet count itself. It excludes the APP13 marker.
CLIe ident:	4 octets	'MRC13', represented as a 3-octet ASCII string plus an hexadecimal count (i.e., X'4D',X'52',X'43',X'0D'). This X'0D'-terminated string "MRC" uniquely identifies this marker segment as the Colour-interpreter Encoder Marker Segment, CLIe.
ColorInterpreter:	1 octet	With value indicating colour interpreter as shown in Table B.4. The interpreter specifies the colour space, bit depth (i.e., number of bits/component) and possibly other colour parameters such as gamut range, illuminant and white point.

Table B.4/T.44 – Colour interpreter octet(s)

Octet value	Coder used
0	Continuous-tone colour using CIELAB 8 bits/component per ITU-T Rec. T.42/LAB
1	Continuous-tone colour using CIELAB 12 bits/component per ITU-T Rec. T.42/LAB
2	Continuous-tone colour using ITU-YCC 8 bits/component per ITU-T Rec. T.42/YCC
3	Continuous-tone colour using ITU-YCC 12 bits/component per ITU-T Rec. T.42/YCC
4-15	Reserved
16	3-bit colour (1 bit/colour) using RGB primaries per 6.2.1/T.43 (Note 1)
17	3-bit colour (1 bit/colour) using CMY primaries per 6.2.1/T.43 (Note 1)
18	4-bit colour (1 bit/colour) using CMYK primaries per 6.2.1/T.43 (Note 1)
19-31	Reserved
32	Palettized colour image using the palette, LAB colour space and bit depth defined in the G3FAX6 marker segment per B.6.5.4.1, included within the same SLC (Note 2)
33	Palettized colour image using the palette, YCC colour space and bit depth defined in the G3FAX6 marker segment per B.6.5.4.1, included within the same SLC (Note 2)
34-255	Reserved
NOTE 1 – 1 bit/component image data is treated as a special case of palette image representation in which colour values are specified by a bit plane stack of named colours, per 6.2.1/T.43, rather than an exact colour value. Palette tables are omitted in these cases since there is no actual palette data. NOTE 2 – G3FAX6 marker segment, as specified in B.6.5.4, is required for colour value interpretation.	

B.6.5.2 Coded image gamut range data

The G3FAX1marker segment for LAB, defined in E.6.6/T.4, or the G3FAX4 maker segment for LAB and YCC, similarly defined as G3FAX1in E.6.6/T.4, shall be used in the interpretation of T.45 encoded colour values (CVAL) when it is present between the SLC and EOH. Presence of the G3FAX1 or the G3FAX4 is optional in the interpretation of T.45 encoded colour values (CVAL); however, if present, it shall be used. The G3FAX1 or the G3FAX4 may be used in the interpretation of colour values that have been encoded with other encoders. The G3FAX4 is similar to the OMSgl or OMGgy (layer base colour gamut range Optional Marker Segment) defined in 9.2.2.1 or 9.2.2.3, with the difference being that G3FAX4 is applied to the encoded data while OMSgl or OMGgy is applied to the layer base colour. The G3FAX4 marker segment is defined in B.6.5.2.1.

The same colour space as that defined by coder in SLC is used.

B.6.5.2.1 G3FAX4 marker segment

The G3FAX4 marker segment, referenced in B.6.5.2 and similarly defined as G3FAX1in E.6.6/T.4, is defined.

G3FAX4 for gamut range for LAB and YCC

X'FFE1' (APP1), length, G3FAX option identifier, gamut range data

The above terms are defined as follows:

Length: (Two octets) – Total APP1 field octet count including the octet count itself, but excluding the APP1 marker.

FAX identifier: (Six octets) – X'47', X'33', X'46', X'41', X'58', X'04' – This X'04'-terminated string "G3FAX" uniquely identifies this APP1 marker as containing FAX information about optional gamut range data. (The FAX option identifiers are referred to as G3FAX1 – G3FAX255, meaning the octet-terminated string "G3FAX", X'nn').

Gamut range data: (Twelve octets) – The data field contains six two-octet signed integers. For example:

X'0064' represents 100.

The calculation from a real value L* to an eight bit value, L, is made as follows:

$$L = (255/Q) \times L^* + P$$

where the first integer of the first pair, P, contains the offset of the zero point in L* in the eight most significant bits. The second integer of the first pair, Q, contains the span of the gamut range in L*. Rounding to the nearest integer is performed. The second pair contains offset and range values for a*. The third pair contains offset and range values for b*. If the image is gray-scale (L* only), the field still contains six integers, but the last four are ignored.

The calculation from a real value Y to an eight-bit value, NY, is made as follows:

$$NY = (255/(Q/1000)) \times Y + P$$

where the first integer of the first pair, P, contains the offset of the zero point in Y in the eight most significant bits. The second integer of the first pair, Q, contains the 1000 times of the span of the gamut range in Y. Rounding to the nearest integer is performed. The second pair contains offset and range values for Cb. The third pair contains offset and range values for Cr.

If the image is gray-scale (Y only), the field still contains six integers, but the last four integers are ignored.

NOTE – This representation is in accord with ITU-T Rec. T.42. When the twelve bits/pel/component option is used, the range and offset are represented as above in eight bits.

These represent the eight most significant bits of the zero-padded twelve-bit number in the offset, and the eight-bit integer range data as above. Appropriately higher precision calculation should be used.

For example, the gamut range L* = [0, 100], a* = [–85, 85], and b* = [–75, 125] would be selected by the code:

X'FFE1',X'0014',X'47',X'33',X'46',X'41',X'58',X'01',X'0000',X'0064',X'0080',X'00AA',X'0060',X'00C8'.

The other example gamut range Y = [0, 1.0], Cb = [–0.5, 0.5], and Cr = [–0.5,0.5] would be represented by the code:

X'FFE1',X'0014',X'47',X'33',X'46',X'41',X'58',X'01',X'0000', X'03E8', X'0080', X'03E8', X'0080', X'03E8'

B.6.5.3 Coded image illuminant data

The G3FAX2 marker segment for LAB, defined in E.6.7/T.4, or the G3FAX5 maker segment for LAB and YCC , similarly defined as G3FAX2 in E.6.6/T.4, shall be used in the interpretation of T.45 encoded colour values (CVAL) when it is present between the SLC and EOH. Presence of the G3FAX2 or the G3FAX5 is optional in the interpretation of T.45 encoded colour values (CVAL), however, if present it shall be used. The G3FAX2 or the G3FAX5 may be used for the interpretation of colour values that have been encoded with other encoders.

The G3FAX5 is similar to the OMSi (layer base colour illuminant Optional Marker Segment) defined in 9.2.2.2/ T.44, with the difference being that G3FAX5 is applied to the encoded data while OMSi is applied to the layer base colour. The G3FAX5 marker segment is defined in B.6.5.3.1.

The colour space to which G3FAX5 is applied, is only LAB. The default illuminant for YCC colour space is D65 and the other potion illuminant is not permitted for YCC colour space.

B.6.5.3.1 G3FAX5 marker segment

The G3FAX5 marker segment, referenced in B.6.5.3 and similarly defined as G3FAX2 in E.6.6/T.4, is defined.

G3FAX5 for illuminant data for LAB

X'FFE1' (APP1), length, G3FAX option identifier, illuminant data for LAB. This option is for further study with the exception of the default case; the specification of the default illuminant, CIE Illuminant D50, may be added for information.

Length:	(Two octets) – Total APP1 field octet count including the octet count itself, but excluding the APP1 marker.
FAX identifier:	(Six octets) – X'47', X'33', X'46', X'41', X'58', X'05'. This X'05'-terminated string "G3FAX" uniquely identifies this APP1 marker as containing optional illuminant data.
Illuminant data:	(Four octets) – The data consist of a four-octet code identifying the illuminant. In the case of a standard illuminant, the four octets are one of the following:
CIE Illuminant D50:	X'00', X'44', X'35', X'30'
CIE Illuminant D65:	X'00', X'44', X'36', X'35'
CIE Illuminant D75:	X'00', X'44', X'37' X'35'
CIE Illuminant SA:	X'00', X'00', X'53', X'41'
CIE Illuminant SC:	X'00', X'00', X'53', X'43'
CIE Illuminant F2:	X'00', X'00', X'46', X'32'
CIE Illuminant F7:	X'00', X'00', X'46', X'37'
CIE Illuminant F11:	X'00', X'46', X'31', X'31'
CIE Illuminant D50:	X'00', X'44', X'35', X'30'

In the case of a colour temperature alone, the four octets consist of the string 'CT', followed by the temperature of the source in degrees Kelvin represented by an unsigned two-octet integer. For example, a 7500°K illuminant is indicated by the code:

X'FFE1', X'000C', X'47', X'33', X'46', X'41', X'58', X'02', X'43', X'54', X'1D4C'.

B.6.5.4 Coded image palette data

The G3FAX3 marker segment for LAB, defined in 7.2.2.4/ T.43, or the G3FAX6 maker segment for LAB and YCC, similarly defined as G3FAX3 in E.6.6/T.4, shall be used in the interpretation of T.45 encoded colour values (CVAL) when the CVALs are defined in terms of palette indices. The G3FAX3 or the G3FAX6 marker segment shall be present between the SLC and EOH when T.45 encoded CVALs are defined in terms of palette indices. The G3FAX3 or the G3FAX6 marker segment may be used in the interpretation of colour values that have been encoded with other encoders.

The G3FAX6 marker segment is defined in B.6.5.4.1 A sample G3FAX6 code stream is also presented in B.6.5.4.2.

B.6.5.4.1 G3FAX6 marker segment

The G3FAX6 marker segment, referenced in B.6.5.4 and similarly defined in 7.2.2.4.1/T.43, is defined.

"G3FAX6 entry for colour palette table"

Colour palette table is specified using the Entry Marker X'FFE3' as follows:

X'FFE3' (Entry Marker), length (4 octets), FAX identifier 3, table ID, tentries, colour table data.

Length: (4 octets) – Total G3FAX3/G4FAX3 entry field octet count including the octet count itself, but excluding the Entry Marker.

FAX identifier 3: (6 octets) – X'47', X'3m', X'46', X'41', X'58', X'06' (m = 3 or 4). This identifier specifies G3FAX6 entry.

Table ID: (2 octets) – This specifies the type of colour palette table.

0: table specified in CIELAB or ITU-YCC space (8 bits/comp. precision).

4: table specified in CIELAB or ITU-YCC space (12 bits/comp. precision).

Tentries: (4 octets) – It specifies the number of the colour palette table entries. This value should have the following relations:

N: Number of bits specified in G3FAX0/G4FAX0.

mb: octets/component in the table:

1: 8 bits precision

2: 12 bits precision

2**(N – 1) < tentries ≤ 2**N

length = 16 + (3 * tentries * mb).

Colour table data: ((3 * tentries * mb) octets) – This data consists of tentries colour palette table entries. Each table entry, which consists of 3 components, is in sequential order from index = 0 to index = tentries – 1. Each component consists of one or two octets value. Its length is specified by the table ID. Each component value is represented by CIELAB space or ITU-YCC space defined in ITU-T Rec. T.42.

B.6.5.4.2 Code string example

The table below is an example of the code string for the following colour palette table. It assumes that the table is specified in CIELAB space (8 bit/comp. precision), $t_{entries}$ = 236.

Colour palette table example for 236 entries and 8 bit accuracy:

Index	**Component values (8 bits)**		
	L*	a*	b*
0	255	128	96
1	0	128	96
2	128	128	96
–	–	–	–
–	–	–	–
–	–	–	–
235	220	128	220

Code String Example:

Entry Marker X'FFE3'	length X'000002D4'	FAX-identifier " G3FAX '3' " X'47', X'33', X'46', X'41', X'58', X'03'	table ID = 0 X'0000'	$t_{entries}$ = 236 X'000000EC'

index = 0 (255,128,96) X'FF', X'80', X'60'	index = 1 (0,128,96) X'00', X'80', X'60'	index = 2 (128,128,96) X'80', X'80', X'60'	... ...	index = 235 (220,128,220) X'DC', X'80', X'DC'

B.6.6 Data format summary

B.6.6.1 High level data format summary

<table>
<tr><td>SOP</td><td colspan="5">X'FFD8 X'FFED', Length, MRC0, Version, Mode, ...</td></tr>
<tr><td>TN</td><td colspan="5">X'FFD9'</td></tr>
<tr><td>OMSgl</td><td colspan="5">X'FFED', Length, MRC10, Gamut data for LAB</td></tr>
<tr><td>OMSi</td><td colspan="5">X'FFED', Length, MRC11, Illuminant data for LAB</td></tr>
<tr><td>OMSgy</td><td colspan="5">X'FFED', Length, MRC9, Gamut data for YCC</td></tr>
<tr><td rowspan="17">Page data</td><td rowspan="14">Stripe 1</td><td>SOSt</td><td colspan="3">X'FFED', Length, MRC1, Type, stripe height</td></tr>
<tr><td rowspan="13">Stripe data</td><td rowspan="6">Layer 2 (L2)</td><td>SLC</td><td>X'FFED', Length, MRC2, Layer number, coder, res., width, height, layer base colour, offset</td></tr>
<tr><td>SDMc</td><td>X'FFE3', Length, MRC3, Optional length, IDs, scope, data</td></tr>
<tr><td>SDMd</td><td>X'FFED', Length, MRC4, Disposition, count, IDs</td></tr>
<tr><td>JB2e</td><td>X'FFED', Length, MRC12, T88Options</td></tr>
<tr><td>EOH</td><td>X'FFED', Length, MRC255, coded data length</td></tr>
<tr><td colspan="2">Coded Data</td></tr>
<tr><td>L1</td><td colspan="2"></td></tr>
<tr><td rowspan="4">L3</td><td colspan="2">SLC</td></tr>
<tr><td>CLIe</td><td>X'FFED', Length, MRC13, ColorInterpreter</td></tr>
<tr><td colspan="2">EOH</td></tr>
<tr><td colspan="2">Coded Data</td></tr>
<tr><td colspan="3">–
–
–</td></tr>
<tr><td colspan="3">LN</td></tr>
<tr><td colspan="5">–
–
–</td></tr>
<tr><td rowspan="2">Stripe N</td><td colspan="4">SOSt</td></tr>
<tr><td colspan="4">Stripe data</td></tr>
<tr><td>EOP</td><td colspan="5">X'FFD9FFD9'</td></tr>
</table>

B.6.6.2 Detail data format summary

MRC Magic Number

SOP marker segment

APP13 marker

Length of Segment

MRC0 SOP identifier

Version

Mode

Mask coder

Image layer coder

Mask resolution

Page width

TN

Layer base colour Gamut optional (OMSgl) marker segment

APP13

Length of Segment

MRC10 OMSgl identifier

gamut range data for LAB

Layer base colour Illuminant optional (OMSi) marker segment

APP13

Length of Segment

MRC11 OMSi identifier

illuminant data

Optional marker segments

APP13

Length of Segment

MRC9 OMSgy identifier

gamut range data for YCC

APP13

Length of Segment

MRCn (n = 14 to 254) identifier

...

Shared data marker (SDMc) segment

...

Shared data marker (SDMc) segment

...

Shared data marker (SDMd) segment

...

Page data

Stripe 1

SOSt marker segment

APP13 marker

Length of Segment

MRC1 SOSt identifier

Type of stripe

Stripe height

Stripe data

Main Mask layer (Layer 2)

SLC marker segment

APP13 marker

Length of segment

MRC2 SLC identifier

layer number

coder

Resolution

Layer width

Layer height

Layer base colour

Offset

SDMc marker segment

APP13 marker

Length of segment

MRC3 SDMc identifier

Optional length (if needed)

ID

Scope

Shared data --------

SDMd marker segment

APP13 marker

Length of segment

MRC4 SDMd identifier

Disposition

Count

ID

JBIG2 encoder marker (JB2e) segment

APP13 marker

Length of segment

MRC12 JB2e identifier

T88Options

End of header (EOH) marker segment

APP13 marker

Length of segment

MRC255 EOH identifier

coded data length

Layer coded data --------

Background layer

SLC marker segment

:

:

Encoder marker segments

...

...

EOH marker segment

Layer coded data --------

Foreground layer

SLC marker segment

:

:

CLIe marker segments

:

G3FAX4 marker segments

:

G3FAX5 marker segments

:

EOH marker segment

Layer coded data -------

Layer 4

SLC marker segment

:

:

SDMd marker segment

...

JBIG2 encoder marker (JB2e) segment

EOH marker segment

Layer coded data --------

Layer 5

SLC marker segment

:

:

CLIe marker segments

G3FAX6 marker segments

:

EOH marker segment

Layer coded data --------

:

:

Layer N

SLC marker segment

:

:

SDMc marker segments

...

EMSe marker segments

EOH marker segment

Layer coded data -------

SDMd marker segments

...

Stripe 2

SOSt marker segment

APP13 marker

Stripe data

Main Mask layer (Layer 2)

Layer coded data --------

Background layer

Layer coded data --------

Foreground layer

Layer coded data -------

Layer 4

Layer coded data --------

Layer 5

Layer coded data --------

:

:

Layer N

Layer coded data -------

Stripe 3

Stripe n

SDMd marker segment

EOP (X'FFD9', X'FFD9')

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	**Terminals for telematic services**
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2005