

---

---

**Information technology — Lossy/lossless  
coding of bi-level images**

**AMENDMENT 3: Extension to colour coding**

*Technologies de l'information — Codage avec ou sans perte des  
images au trait*

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**  
AMENDEMENT 3. Extension au codage de la couleur

[ISO/IEC 14492:2001/Amd 3:2012](https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-ac6ff9861231/iso-iec-14492-2001-amd-3-2012)

[https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-  
ac6ff9861231/iso-iec-14492-2001-amd-3-2012](https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-ac6ff9861231/iso-iec-14492-2001-amd-3-2012)

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

[ISO/IEC 14492:2001/Amd 3:2012](https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-ac6ff9861231/iso-iec-14492-2001-amd-3-2012)

<https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-ac6ff9861231/iso-iec-14492-2001-amd-3-2012>



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2012

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

## Foreword

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

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

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

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

Amendment 3 to ISO/IEC 14492:2001 was prepared jointly by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T. The identical text is published as ITU-T Rec. T.88 (2000 E)/Amd.3.

(standards.iteh.ai)

[ISO/IEC 14492:2001/Amd 3:2012](https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-ac6ff9861231/iso-iec-14492-2001-amd-3-2012)

<https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-ac6ff9861231/iso-iec-14492-2001-amd-3-2012>

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

ISO/IEC 14492:2001/Amd 3:2012

<https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-ac6ff9861231/iso-iec-14492-2001-amd-3-2012>

INTERNATIONAL STANDARD  
ITU-T RECOMMENDATION

## Information technology – Lossy/lossless coding of bi-level images

## Amendment 3

## Extension to colour coding

## 1) Clause 2, Normative References

Append a new reference as follows (with the additions underlined):

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

## 2) Clause 4.2, Symbol definitions

Insert new symbols ***COLEXTFLAG***, ***CPCOMPLEN***, ***CPDEFCOLS***, ***CPEXCOLS***, ***CPNCOMP***, ***CPNVALS***, ***GBCOLS***, ***GBCOMBOP***, ***GBFGCOLID***, ***SBCOLS***, ***SBCOLSECTSIZE*** and ***SBFGCOLID*** as follows (with the additions underlined):

(Symbols left untouched)

Clow Low-order 16 bits of C

**COLEXTFLAG** A parameter indicating whether the generic region segment is extended to represent a coloured bitmap

**CONTEXT** The values of the pixels in a template used in the generic or generic refinement decoding procedure

**CPCOMPLEN** The length (in bytes) of each component's value

**CPDEFCOLS** The default colour set

**CPEXCOLS** The colours defined in the colour palette segment

**CPNCOMP** The number of colour components

**CPNVALS** The number of colour values coded in this segment

(Symbols left untouched)

**GB** The prefix used for many of the variables associated with a generic (bitmap) region decoding procedure

**GBCOLS** An array containing the colours used in a generic region segment

**GBCOMBOP** The combination operator used in a generic region decoding procedure

**GBFGCOLID** The 4-byte integer indicating the foreground colour of the generic region segment

(Symbols left untouched)

**SB** The prefix used for many of the variables associated with a symbol (bitmap) region decoding procedure

**SBCOLS** An array of colours used in a text region segment

**SBCOLSECTSIZE** The colour section length

(Symbols left untouched)

**SBDEFPIXEL** The default for pixels in a text region

**SBFGCOLID** The three 1-byte integers indicating the foreground colour of the text region segment

(Symbols left untouched)

**3) Clause 6.2.2, Input parameters**

Add new rows of **GBCOLS**, **GBCOMBOP** and **COLEXTFLAG** after **GBATY<sub>4</sub>**, and add a new Note f), in Table 2 as follows (with the additions underlined):

**Table 2 – Parameters for the generic region decoding procedure**

Name	Type	Size (bits)	Signed?	Description and restrictions
<i>(Rows left untouched)</i>				
<b>GBATY<sub>4</sub></b>	Integer	8	Y	The Y location of the adaptive template pixel A <sub>4</sub> . <sup>b)</sup>
<u><b>GBCOLS</b></u>	<u>Array of colours (colour palette)</u>			<u>An array containing the colours used in this generic region.<sup>f)</sup></u>
<u><b>GBCOMBOP</b></u>	<u>Operator</u>			<u>The combination operator for this generic region. Shall take on the value REPLACE.<sup>f)</sup></u>
<u><b>COLEXTFLAG</b></u>	<u>Integer</u>	<u>1</u>	<u>N</u>	<u>A parameter indicating whether the generic region segment is extended to represent coloured bitmap.</u>
<i>(Notes left untouched)</i>				
<sup>f)</sup> <u>Unused if COLEXTFLAG = 0.</u>				

iTech STANDARD PREVIEW  
(standards.iteh.ai)

**4) Clause 6.2.4, Variables used in decoding**

Add a new row of **GBFGCOLID** after **CONTEXT**, and add a new Note b) in Table 4 as follows (with the additions underlined):

<https://standards.iteh.ai/catalog/standards/sist/517d311e-95b9-4a0d-9022-ac6ff9861231/iso-iec-14492-2001-amd-3-2012>

**Table 4 – Variables used in the generic region decoding procedure**

Name	Type	Size (bits)	Signed?	Description and restrictions
<i>(Rows left untouched)</i>				
<b>CONTEXT</b>	Integer	16	N	The values of the pixels in the template. <sup>a)</sup>
<u><b>GBFGCOLID</b></u>	<u>Integer</u>	<u>32</u>	<u>N</u>	<u>The 4-byte integer indicating the colour palette ID of the foreground colour for the generic region segment.<sup>b)</sup></u>
<i>(Notes left untouched)</i>				
<sup>b)</sup> <u>Unused if COLEXTFLAG = 0.</u>				

**5) New clause 6.2.7, Colour extension of generic region segment**

Add a new clause 6.2.7 after clause 6.2.6, as follows:

**6.2.7 Colour extension of generic region segment**

If **COLEXTFLAG** is 1, after the decoding procedure described in 6.2.5 or 6.2.6, read **GBFGCOLID** which is a 4-byte integer indicating the colour palette ID, and set the colour specified by **GBCOLS[GBFGCOLID]** to the segment's foreground colour.

The background colour of the segment of which **COLEXTFLAG** is 1 is regarded as transparent.

NOTE – **GBFGCOLID** (the foreground colour information) has a fixed 4-byte field and is put at the end of the region segment only when **COLEXTFLAG** is 1. The decoder can find the address of **GBFGCOLID** by subtracting 4 from the segment data length recorded in the segment header.

6) Clause 6.4.2, Input parameters

Revise the description and restrictions of **SBCOMBOP**, add new rows of **COLEXTFLAG** and **SBCOLS** after **SBRATY<sub>2</sub>**, and new Notes e) and f) in Table 9 as follows (with the additions underlined):

Table 9 – Parameters for the text region decoding procedure

Name	Type	Size (bits)	Signed?	Description and restrictions
<i>(Rows left untouched)</i>				
<b>SBCOMBOP</b>	Operator			The combination operator for this text region. May take on the values OR, AND, XOR, XNOR and REPLACE. <sup>e)</sup>
<i>(Rows left untouched)</i>				
<b>SBRATY<sub>2</sub></b>	Integer	8	Y	The Y location of the adaptive template pixel RA <sub>2</sub> . <sup>c)</sup>
<b><u>COLEXTFLAG</u></b>	<u>Integer</u>	<u>1</u>	<u>N</u>	<u>A parameter indicating whether the generic region segment is extended to represent coloured bitmap.</u>
<b><u>SBCOLS</u></b>	<u>Array of colours</u>			<u>An array containing the colours used in this text region.<sup>f)</sup></u>
<i>(Notes left untouched)</i>				
e) <u>REPLACE operator is used if and only if COLEXTFLAG = 1.</u>				
f) <u>Unused if COLEXTFLAG = 0.</u>				

7) Clause 6.4.4, Variables used in decoding

Add new rows of **SBCOLSECTSIZE** and **SBFGCOLID** after **HO<sub>1</sub>**, and Note b) after Note a) in Table 11 as follows (with the additions underlined):

Table 11 – Variables used in the text region decoding procedure

Name	Type	Size (bits)	Signed?	Description and restrictions
<i>(Rows left untouched)</i>				
<b>HO<sub>1</sub></b>	Integer	32	N	The height of <b>IBO<sub>1</sub></b> . <sup>a)</sup>
<b><u>SBCOLSECTSIZE</u></b>	<u>Integer</u>	<u>32</u>	<u>N</u>	<u>The colour section length.<sup>b)</sup></u>
<b><u>SBFGCOLID</u></b>	<u>Array of integers</u>			<u>An array of colour palette ID, indicating the colour of each symbol instance.<sup>b)</sup></u>
a) <u>Unused if SDREFINE = 0.</u>				
b) <u>Unused if COLEXTFLAG = 0.</u>				

8) Clause 6.4.5, Decoding the text region

Insert Figure AMD3-1 after Figure 17 as follows:

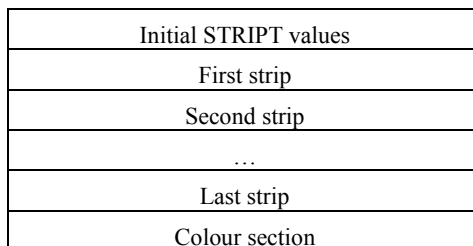
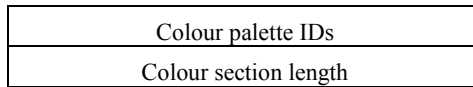


Figure AMD3-1 – Coded structure of a text region extended for colour text

**9) Clause 6.4.5, Decoding the text region**

Insert Figure AMD3-2 after Figure 20:



**Figure AMD3-2 – Coded structure of colour section**

**10) Clause 6.4.5, Decoding the text region**

Insert new Notes after Figure AMD3-2:

NOTE 3bis – If **COLEXTFLAG** is 1 then a colour section, which specifies the colours of all symbol instances, is put at the end of the region as shown in Figure AMD3-1. The colour section consists of two parts as shown in Figure AMD3-2: colour palette IDs and colour section length. The colour palette IDs, which is encoded in the data structure based on Rec. ITU-T T.45, contains the information that specifies the colour of every symbol instances in the segment. The colour section length is a 4-byte field that contains the length of the colour section, in bytes.

NOTE 3ter – The background colour of the segment of which **COLEXTFLAG** is 1 is regarded as transparent.

**11) Clause 6.4.5, Decoding the text region**

Change the procedures itemized by 3) and 4) to 4) and 5) respectively, and insert a new procedure 3) as follows (with the modifications and additions underlined):

- 3) If **COLEXTFLAG** is 1, decode the colour section as described in 6.4.12.

(Procedures left untouched)

- 4) Decode each strip as follows:

(Procedures left untouched)

- 5) After all the strips have been decoded, the current contents of **SBREG** are the results that shall be obtained by every decoder, whether it performs this exact sequence of steps or not.

**12) Clause 6.4.5, Decoding the text region**

In the sub-procedure 4)-c), which is previously itemized by 3)-c), change the sub-sub-procedures itemized by ix), x) and xi) to x), xi) and xii) respectively, and insert a new sub-sub-procedure ix) after procedure viii), as follows (with the modifications and additions underlined):

- ix) If **COLEXTFLAG** is 1, set the colour specified by **SBCOLS**[**SBFGCOLID**][**NINSTANCES**] to the foreground colour of the symbol instance bitmap  $IB_l$ .
- x) Draw  $IB_l$  into **SBREG**. Combine each pixel of  $IB_l$  with the current value of the corresponding pixel in **SBREG**, using the combination operator specified by **SBCOMBOP**. Write the results of each combination into that pixel in **SBREG**.
- xi) Update **CURS** as follows:

(Procedures left untouched)

- xii) Set:

$$NINSTANCES = NINSTANCES + 1$$

**13) Clause 6.4.5, Decoding the text region**

Insert a new Note at the end of 6.4.5, as follows:

NOTE 6 – The colour palette (**SBCOLS**) is created by concatenating the default colour set (**CPDEFCOLS**) and the additional colours (**CPEXCOLS**) defined in the colour palette segment referred to by this segment.



**14) New clause 6.4.12, Colour section**

Add a new clause 6.4.12 after clause 6.4.11.5, as follows:

**6.4.12 Colour section**

If **COLEXTFLAG** is 1, the colour information recorded in the colour section needs to be decoded as shown in this clause.

**6.4.12.1 Colour section length**

The decoding procedure of the colour section begins with achieving its top address by subtracting the **SBCOLSECTSIZE** from the segment data length defined in the segment header. **SBCOLSECTSIZE** is a 4-byte field at the end of the text region segment data part. The decoder can find the end of the data part by reading the segment data length in the segment header.

**6.4.12.2 Colour palette IDs**

The field of the colour palette IDs is decoded in accordance with Rec. ITU-T T.45 and the result is stored in **SBFGCOLID**. The number of elements in **SBFGCOLID** is equal to the number of symbol instances in the text region segment (**SBNUMINSTANCES**).

NOTE – If the codestream has the coloured text region segment, the ITU-T T.45 for colour palette IDs in the colour section is restricted as follows:

- The number of colour components is 1 (**NCOMP** = 1 in the header of ITU-T T.45 codestream).
- The length of each colour component's value is 1-byte (**COMPLEN** = 1 in the header of ITU-T T.45 codestream).
- The number of colour values is equal to **SBNUMINSTANCES** (**NVALS** = **SBNUMINSTANCES** in the header of ITU-T T.45 codestream).

**15) New clause 6.8, Colour palette decoding procedure**

Add a new clause 6.8 after clause 6.7.5, as follows:

**6.8 Colour palette decoding procedure****6.8.1 General description**

This decoding procedure is used to decode a set of colours; these colours can then be used by generic and text region decoding procedures.

**6.8.2 Input parameters**

The colour palette segment requires no input parameter.

The parameters to this decoding procedure are shown in Table AMD3-1.

**Table AMD3-1 – Parameters for the colour palette decoding procedure**

Name	Type	Size (bits)	Signed?	Description and restrictions
<b>CPNCOMP</b>	Integer	8	N	The number of colour components.
<b>CPCOMPLEN</b>	Integer	8	N	The length (in bytes) of each component's value. Make take on the values 1, 2 or 4.
<b>CPNVALS</b>	Integer	8	N	The number of colour values coded in this segment.