

INTERNATIONAL STANDARD

ISO/IEC
10918-3

First edition
1997-05-01

Information technology — Digital compression and coding of continuous-tone still images: Extensions

*Technologies de l'information — Compression numérique et codage des
images fixes de nature photographique: Extensions*



Reference number
ISO/IEC 10918-3:1997(E)

AVA Documents
Document Sharing Hub

Contents

	<i>Page</i>
1 Scope	1
2 Normative references	1
3 Definitions, abbreviations, symbols and conventions	2
4 General	6
5 Compressed data format requirements	10
6 Encoder requirements	10
7 Decoder requirements	11
Annex A – Mathematical definitions	12
Annex B – Compressed data formats	14
Annex C – Variable quantization	28
Annex D – Selective refinement	31
Annex E – Tiling	34
Annex F – Still Picture Interchange File Format (SPIFF)	39
Annex G – Compliance testing	57
Annex H – Examples and guidelines	72
Annex I – Bibliography	80

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. 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.

International Standard ISO/IEC 10918-3 was prepared 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.84.

ISO/IEC 10918 consists of the following parts, under the general title *Information technology — Digital compression and coding of continuous-tone still images*:

- *Part 1: Requirements and guidelines*
- *Part 2: Compliance testing*
- *Part 3: Extensions*
- *Part 4: Registration procedures for JPEG profile, APPn marker and SPIFF profile ID marker*
- *Part 5: MHEG subset for base level implementation*

Annexes A to G form an integral part of this part of ISO/IEC 10918. Annexes H and I are for information only.

Introduction

This Recommendation | International Standard, Digital compression and coding of continuous-tone still images, is published as three parts:

- CCITT Rec. T.81 | ISO/IEC 10918-1: Requirements and guidelines;
- ITU-T Rec. T.83 | ISO/IEC 10918-2: Compliance testing;
- ITU-T Rec. T.84 | ISO/IEC 10918-3: Extensions.

This Recommendation | International Standard sets out requirements and guidelines for encoding and decoding extensions to the processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1, and for the coded representation of compressed image data of these extensions. This Recommendation | International Standard also defines tests for determining whether implementations comply with the requirements for the various encoding and decoding extensions.

This Recommendation | International Standard:

- defines extensions [including variable quantization, selective refinement, composite tiling, and a Still Picture Interchange File Format (SPIFF)] to processes for converting source image data to compressed image data;
- defines extensions to processes for converting compressed image data to reconstructed image data;
- defines coded representations for compressed image data;
- gives guidance and examples on how to implement these extensions in practice;
- describes compliance tests for these extensions.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – DIGITAL COMPRESSION AND CODING OF CONTINUOUS-TONE STILL IMAGES: EXTENSIONS

1 Scope

This Recommendation | International Standard is applicable to continuous-tone – grayscale or colour – digital still image data. It is applicable to a wide range of applications which require use of compressed images.

This Recommendation | International Standard:

- defines extensions [including variable quantization, selective refinement, tiling, and a Still Picture Interchange File Format (SPIFF)] to processes for converting source image data to compressed image data;
- defines extensions to processes for converting compressed image data to reconstructed image data;
- defines coded representations for compressed image data;
- gives guidance and examples on how to implement these extensions in practice;
- describes compliance tests for these extensions.

2 Normative references

The following Recommendations and International Standards contain provisions which, through references in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau (TSB) of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation H.262 (1995) | ISO/IEC 13818-2:1995, *Information technology – Generic coding of moving pictures and associated audio information: Video*.
- CCITT Recommendation T.81 (1992) | ISO/IEC 10918-1:1994, *Information technology – Digital compression and coding of continuous-tone still images – Requirements and guidelines*.
- ITU-T Recommendation T.82 (1993) | ISO/IEC 11544:1993, *Information technology – Coded representation of picture and audio information – Progressive bi-level image compression*.
- ITU-T Recommendation T.83 (1994) | ISO/IEC 10918-2:1995, *Information technology – Digital compression and coding of continuous-tone still images: Compliance testing*.

2.2 Additional references

- ISO 3166:1993¹⁾, *Codes for the representation of names of countries*.
- ISO 5807:1985, *Information processing – Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*.
- ISO 8601:1988, *Data elements and interchange formats – Information interchange – Representation of dates and times*.

¹⁾ Currently under revision.

- ISO 8859-1:1987, *Information processing – 8-bit single byte coded graphic character sets – Part 1: Latin alphabet No. 1.*
- ISO 8859-2:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 2: Latin alphabet No. 2.*
- ISO 8859-3:1988, *Information processing – 8-bit single-byte coded graphic character sets – Part 3: Latin alphabet No. 3.*
- ISO 8859-4:1988, *Information processing – 8-bit single-byte coded graphic character sets – Part 4: Latin alphabet No. 4.*
- ISO/IEC 8859-5:1988, *Information processing – 8-bit single-byte coded graphic character sets – Part 5: Latin/Cyrillic alphabet.*
- ISO 8859-6:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 6: Latin/Arabic alphabet.*
- ISO 8859-7:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 7: Latin/Greek alphabet.*
- ISO 8859-8:1988, *Information processing – 8-bit single-byte coded graphic character sets – Part 8: Latin/Hebrew alphabet.*
- ISO/IEC 8859-9:1989, *Information processing – 8-bit single-byte coded graphic character sets – Part 9: Latin alphabet No. 5.*
- ISO/IEC 8859-10:1992, *Information technology – 8-bit single-byte coded graphic character sets – Part 10: Latin alphabet No. 6.*
- ISO/IEC 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.*
- ISO/IEC 11172-2:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video.*
- ITU-T Recommendation T.4 (1993), *Standardization of group 3 facsimile apparatus for document transmission.*
- CCITT Recommendation T.6 (1988), *Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus.*
- ITU-T Recommendation T.30 (1993), *Procedures for document facsimile transmission in the general switched telephone network.*
- ITU-T Recommendation T.42 (1994), *Continuous-tone colour representation method for facsimile.*
- CCITT Recommendation T.51 (1992), *Latin based coded character sets for telematic services.*
- ITU-T Recommendation T.85 (1995), *Application profile for Recommendation T.82 – Progressive bi-level image compression (JBIG coding scheme) for facsimile apparatus.*
- CCITT Recommendation T.503 (1991), *A document application profile for the interchange of Group 4 facsimile documents.*
- CIE 1976 (L* a* b*) space, *CIE Publication No. 15.2, Colorimetry, 2nd. Ed. (1986).*
- Recommendation ITU-R BT.470-3 (1995), *Television systems.*
- Recommendation ITU-R BT.601-4 (1992), *Encoding parameters of digital television for studios.*
- Recommendation ITU-R BT.709-1, *Basic parameter values for the HDTV standards for the studio and for international programme exchange.*
- SMPTE 170M – 1994, *For television – Composite analog video signal – NTSC for studio applications.*
- SMPTE 240M – 1994, *For television – Signal parameters – 1125 line high definition production systems.*

3 Definitions, abbreviations, symbols and conventions

3.1 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

- 3.1.1 capability indicator:** A parameter of the version marker segment that indicates which of the processes of CCITT Rec. T.81 | ISO/IEC 10918-1 or the extensions of ITU-T Rec. T.84 | ISO/IEC 10918-3 are required to decode the bitstream which follows. Each bit of this parameter signals the presence of a particular capability.
- 3.1.2 component registration:** Specification of the spatial positioning of samples within components relative to the samples of other components.
- 3.1.3 component selective refinement:** A type of selective refinement which allows for a region of an image to contain fewer than the total number of colour components defined in the frame header.
- 3.1.4 composite tiling:** A type of tiling which allows for overlapping tiles without restrictions on tile size and other compression parameters.
- 3.1.5 grid:** A mathematical abstraction consisting of a two-dimensional array of elements used for composite tiling.
- 3.1.6 hierarchical selective refinement:** A type of selective refinement which allows for a region of an image to be further refined by the next differential frame of a hierarchical sequence.
- 3.1.7 hierarchical sequence:** A hierarchical progression of frames starting with a DHP marker segment.
- 3.1.8 image grid:** For composite tiled images, a grid associated with the down-sampled collaged image.
- 3.1.9 parameters:** Fixed length unsigned integers 4, 8, 16, or 32 bits in length, or variable length strings terminated by a single byte with all bits set to 0, used in the compressed data formats and the Still Picture Interchange File Format.
- 3.1.10 progressive selective refinement:** A type of selective refinement which allows for more non-zero DCT coefficients, more bits to the DCT coefficients, or both to be added to those previously coded in a region of an image.
- 3.1.11 pyramidal tiling:** A type of tiling which allows for multiple resolution levels. Tiles within each resolution level adhere to simple tiling definitions.
- 3.1.12 reference grid:** For composite tiled images, a grid associated with the up-sampled collaged image.
- 3.1.13 resolution:** Samples per unit of linear dimension.
- 3.1.14 selective refinement:** An extension which provides for further processing of a region of an image, typically for the purpose of image quality enhancement. There are three types of selective refinement: hierarchical selective refinement, progressive selective refinement, and component selective refinement.
- 3.1.15 simple tiling:** The definition of an image by an association of contiguous non-overlapping sub-images, each with identical parameters.
- 3.1.16 tile:** A sub-image of any size and resolution which is part of a larger image.
- 3.1.17 tiled image:** An image composed of rectangular sub-images. These sub-images are called tiles.
- 3.1.18 tile grid:** For composite tiled images, a grid associated with a tile.
- 3.1.19 variable quantization:** The process of rescaling quantization values used to quantize the AC DCT coefficients at any 8 by 8 block of an image, as described in Annex C.

3.2 Abbreviations

The abbreviations used in this Recommendation | International Standard are listed below.

- 3.2.1 JBIG:** Joint Bi-level Image experts Group – The joint ISO/ITU committee responsible for developing standards for bi-level image coding. It also refers to the standard produced by this committee: ITU-T Rec. T.82 | ISO/IEC 11544.
- 3.2.2 JFIF:** JPEG File Interchange Format – A basic file format specification for the delivery and interchange of JPEG compressed images.
- 3.2.3 JPEG:** Joint Photographic Experts Group – The joint ISO/ITU committee responsible for developing standards for continuous tone still picture coding. It also refers to the standards produced by this committee: CCITT Rec. T.81 | ISO/IEC 10918-1, ITU-T Rec. T.83 | ISO/IEC 10918-2, and ITU-T Rec. T.84 | ISO/IEC 10918-3.
- 3.2.4 JTIP:** JPEG Tiled Image Pyramid – A file organization where large images are tiled, reduced, and packed into a multi-resolution pyramidal JPEG image.

3.2.5 MPEG: Moving Pictures Expert Group – ISO-IEC committee responsible for developing standards for coding for video and audio. It also refers to the relevant standards produced by the group: MPEG1-ISO/IEC 11172 and MPEG2-ISO/IEC 13818.

3.2.6 SPIFF: Still Picture Interchange File Format – A file format defined by ITU-T Rec. T.84 | ISO/IEC 10918-3 intended for use by a wide variety of applications to exchange still pictures.

3.3 Symbols

In addition to the symbols used in CCITT Rec. T.81 | ISO/IEC 10918-1, the following symbols used in this Recommendation | International Standard are listed below:

BPS	Bits per sample (SPIFF file header parameter)
C	Compression type (SPIFF file header parameter)
CAP_i	Capability indicator – V + 1 bytes
CHARSET	Character set to be used to interpret the character data in SPIFF files
C_i	Component identifier
CONTLOC	File offset of contact information or zero (SPIFF directory entry parameter)
COPYRLOC	File offset of copyright information or zero (SPIFF directory entry parameter)
COPYRID	Copyright identifier (SPIFF directory entry parameter)
CREATLOC	File offset of creator identification or zero (SPIFF directory entry parameter)
CROFFSET_i	Component registration vertical and horizontal offsets (SPIFF directory entry parameter)
CR_{vo}	Component registration vertical offset
CR_{ho}	Component registration horizontal offset
DATE	Modification date (SPIFF directory entry parameter)
DCR	Define component registration marker
DESCLOC	File offset of the image description or zero (SPIFF directory entry parameter)
DQS	Define quantizer scale selection marker
DTI	Define tiled image marker
DTT	Define tile marker
DTTINDX	File offset of the offset list of DTT markers (SPIFF directory entry parameter)
EDATA	SPIFF directory entry data – The parameter field of an ETAG, which has a format specific to the corresponding ETAG value
ELEN	SPIFF directory entry length – The length of the directory entry in bytes, minus 2
EMN	SPIFF directory entry magic number – A two-byte code signalling the start of a directory entry
EOD	SPIFF end of directory
EODLEN	SPIFF EOD entry length – The exact length of the EOD entry in bytes
EODTAG	SPIFF end of directory tag – Identifies the EOD entry
ETAG	SPIFF directory entry tag
HEIGHT	Image height (SPIFF file header parameter)
HLEN	Header length (SPIFF file header parameter)
HRES	Horizontal resolution (SPIFF file header parameter)
IDENT	Additional identifier (SPIFF file header parameter)
IMGFLIP	Image flip (SPIFF directory entry parameter)
IMGOR	Image orientation (SPIFF directory entry parameter)
Lcr	Length of parameters in DCR segment
LEVAUT	Level of authenticity (SPIFF directory entry parameter)
Lqs	Length of parameters in DQS segment
Lrf	Length of parameters in SRF segment

Lrs	Length of parameters in SRS segment
Ltf	Length of parameters in DTT segment
Lti	Length of parameters in DTI segment
Lv	Length of parameters in VER segment
MN	Magic number – The first four bytes of a SPIFF file (SPIFF file header parameter)
NC	Number of components (SPIFF file header parameter)
NUMDTT	Number of DTT markers (SPIFF directory entry parameters)
NUMRST	Number of restart markers in a scan or zero (SPIFF directory entry parameter)
NUMSCAN	Number of scans (SPIFF directory entry parameter)
O_{hf}	Selectively refined frame horizontal offset
O_{hs}	Selectively refined scan horizontal offset
O_{vf}	Selectively refined frame vertical offset
O_{vs}	Selectively refined scan vertical offset
P	Profile identification (SPIFF file header parameter)
Q_SCALE	quantization scale
QS_CHANGE	Quantization scale change – The DC magnitude category that signals a new value of SCALE_CODE
R	Resolution units (SPIFF file header parameter)
REFNO1	Reference number one (SPIFF directory entry parameter)
REFNO2	Reference number two (SPIFF directory entry parameter)
REFNO3	Reference number three (SPIFF directory entry parameter)
REGAUT	Contact registration authority (SPIFF directory entry parameter)
REGCON	Registration country (SPIFF directory entry parameter)
REGID	Registration identifier (SPIFF directory entry parameter)
Rev	Revision number
RGhs	Reference grid width
RGvs	Reference grid height
RSTLIST	Restart list – The file offset to the start of the restart marker list for the scan or zero (SPIFF directory entry parameter)
S	Colour space in which the sample values define coordinates (SPIFF file header parameter)
SCALE_CODE	A 5-bit value which specifies the scale factor Q_SCALE
SCANEND	File offset of the first marker after the scan's compressed data (SPIFF directory entry parameter)
SCANLIST	Scan list containing one 4-word entry per scan (SPIFF directory entry parameter)
SCANSTRT	File offset to the X'FF' byte of the SOS (SPIFF directory entry parameter)
S_{hs}	Selectively refined scan horizontal size
SRF	Selectively refined frame marker
SRS	Selectively refined scan marker
STRLOC	SPIFF file offset of a character string or zero
S_{vs}	Selectively refined scan vertical size
Tc	Quantizer scale selector
TFho	Tile horizontal offset
TFhs	Tile horizontal scale
TFvo	Tile vertical offset
TFvs	Tile vertical scale
Tlhs	Tiled image horizontal scale

TIME	Modification time (SPIFF directory entry parameter)
TITLELOC	The file offset of the image title or zero (SPIFF directory entry parameter)
Tivs	Tiled image vertical scale
TNBPS	Thumbnail bits per sample (SPIFF directory entry parameter)
TNC	Thumbnail compression type (SPIFF directory entry parameter)
TNDATA	Thumbnail data offset (SPIFF directory entry parameter)
TNHEIGHT	Thumbnail vertical size (SPIFF directory entry parameter)
TNS	Thumbnail colour space (SPIFF directory entry parameter)
TNWIDTH	Thumbnail horizontal size (SPIFF directory entry parameter)
TRANCHAR	Transfer characteristics (SPIFF directory entry parameter)
TT	Tiling type
V	Version number
VER	Version marker
VERS	SPIFF version (SPIFF file header parameter)
VERSNLOC	File offset of the image version or zero (SPIFF directory entry parameter)
VRES	Vertical resolution (SPIFF file header parameter)
WIDTH	Image width (SPIFF file header parameter)

3.4 Conventions

The flowcharts use the conventions given in ISO 5807. One of the conventions is that arrows are not needed when the flow is from left-to-right and from top-to-bottom. Arrows are sometimes used in such cases to increase clarity.

4 General

The purpose of this clause is to give an overview of this Recommendation | International Standard. Another purpose is to introduce some of the terms which are defined in clause 3. (Terms defined in clause 3 of CCITT Rec. T.81 | ISO/IEC 10918-1 and clause 3 of ITU-T Rec. T.83 | ISO/IEC 10918-2 continue to apply in this Recommendation | International Standard.)

This Recommendation | International Standard defines extensions to the elements specified in CCITT Rec. T.81 | ISO/IEC 10918-1. This Recommendation | International Standard also describes compliance testing for embodiments of these extensions. Extensions which pertain to encoding or decoding are defined as procedures which may be used in combination with the encoding and decoding processes of CCITT Rec. T.81 | ISO/IEC 10918-1; no new processes are defined. This Recommendation | International Standard also defines extensions to the compressed data formats, i.e. interchange format and the abbreviated formats. Each encoding or decoding extension shall only be used in combination with particular coding processes and only in accordance with the requirements set forth herein. These extensions are backward compatible in the sense that decoders which implement these extensions will also support configuration subsets that are currently defined by CCITT Rec. T.81 | ISO/IEC 10918-1.

4.1 Extensions specified by this Recommendation | International Standard

The following extensions are specified:

- An extension which provides for variable quantization within a scan. The variable quantization extension may be used in conjunction with any of the DCT-based processes with the exception of the baseline process. The variable quantization extension provides for scaling of all quantization tables at the 8×8 block level;
- An extension which provides for selective refinement. The selective refinement extension refers to selecting rectangular regions of an image components for further refinement. There are several types of selective refinement:
 - a) The first type of selective refinement, referred to as hierarchical selective refinement, allows only a region of one or more components to be further refined by the next differential frame of a hierarchical sequence.

- b) The second type of selective refinement, referred to as progressive selective refinement, applies to the DCT-based progressive mode of operation. This type of selective refinement allows more non-zero DCT coefficients, more bits to the DCT coefficients, or both, to be added to a region of one or more component.
- c) The third type of selective refinement, referred to as component selective refinement, is used to specify a region of an image, which contains colour components which do not exist in other regions of the image.
- An extension which provides for tiling. The tiling extension is used to associate a number of sub-images, also called tiles, in order to form a single tiled image. The three types of tiling are summarized below:
 - a) For simple tiling, all tiles except possibly those on the right and bottom border have the same maximum dimensions, number of components, component IDs and scaling factors. Tiles (i.e. their components' arrays) are non-overlapping and contiguous. The tiles are coded sequentially from left-to-right and top-to-bottom.
 - b) For pyramidal tiling, multiple resolution versions of the same image (i.e. each version has different maximum dimensions) may be stored together in the same data stream. Each version of the image (also called a "resolution level") is stored as a tiled image with simple tiling.
 - c) For composite tiling, there are no restrictions except that all tiles shall have the same component identifiers.
- An extension which provides for the interchange of compressed image files between application environments. This extension is referred to as the Still Picture Interchange File Format (SPIFF) extension.
- Other extensions include addition of a version number to the compressed data format and increasing the limit on the number of data units in a minimum coded unit to 20.

The following subclauses describe these extensions in greater detail.

4.1.1 Variable quantization extension

The variable quantization extension is an enhancement to the quantization procedure of DCT-based processes which provides for changes to the quantization table values within a scan at the 8×8 block level. This extension may be used in conjunction with any of the DCT-based processes with the exception of the Baseline Process.

The quantization procedure as defined by CCITT Rec. T.81 | ISO/IEC 10918-1 is the step in the encoding process where each of the 64 DCT coefficients are quantized using one of 64 corresponding values from a quantization table. CCITT Rec. T.81 | ISO/IEC 10918-1 permits quantization tables to be redefined prior to the start of a scan but does not allow quantization table values to be changed within a scan. The variable quantization extension defined by this Recommendation | International Standard provides for scaling of quantization values at the 8×8 block level.

The variable quantization extension introduces a quantizer scale factor which may be coded in the compressed data stream at the start of any 8×8 block. The quantizer scale factor is used to scale the quantization table values which correspond to the AC coefficients in the quantization procedure. All defined quantization tables are scaled by the same quantizer scale factor.

This extension provides the following capabilities:

- The ability to compress an image to less than a bounded size with a single sequential pass over the image. The capability is valuable to applications which utilize a fixed-size compressed picture memory.
- The ability to use the masking properties of the human visual system more effectively, and thereby achieve greater compression rates for the same subjective quality.
- The ability to transcode, i.e. entropy decoding followed by entropy encoding, between some coded data representations defined by this Recommendation | International Standard and those defined by some other standards, including ISO/IEC 11172-2 and ITU-T Rec. H.262 | ISO/IEC 13818-2 (MPEG).

4.1.2 Selective refinement extension

The selective refinement extension is used to select a rectangular region of one or more components of an image for further refinement. The different types of selective refinement are described below.

4.1.2.1 Hierarchical selective refinement

Hierarchical selective refinement is used in the hierarchical mode of operation for refining a rectangular region of one or more components of an image. The location of the region of each component of the image to be selectively refined is specified immediately prior to a differential frame within an hierarchical sequence. The size of the region of each component is specified in the differential frame header. The difference image data reconstructed from the differential frame is then added only to the specified region of each component. One of the primary uses for this type of selective refinement is for coding a particular region of interest with greater detail than the remainder of the image.

4.1.2.2 Progressive selective refinement

The second type of selective refinement, referred to as progressive selective refinement, is used in the DCT-based progressive mode of operation. This type of selective refinement is used for similar reasons as hierarchical selective refinement. Progressive selective refinement may be applied to the DCT-based processes which use spectral selection, successive approximation, or both procedures in combination.

When progressive selective refinement is applied to a scan which uses the spectral selection procedure, more non-zero DCT coefficients are added to a region of each component. When it is applied to a scan which uses the successive approximation procedure, more bits are added to the DCT coefficients in a region of each component. Progressive selective refinement may also be applied to scans which use both procedures. In all cases, the location and size of the region of each component to be selectively refined is specified immediately prior to the scan which is used for selective refinement.

4.1.2.3 Component selective refinement

The third type of selective refinement, referred to as component selective refinement, may be used in all modes of operation for specifying a region of an image which contains colour components that do not exist in other regions of the image. The most common use for this type of selective refinement is for representing images which are mixed grayscale and colour.

4.1.3 Tiling extension

The tiling extension is used to associate a number of sub-images, also called tiles, in order to form a single tiled image. The tiling extension is also used to represent images which have maximum dimensions larger than 65535 samples on a side. The three types of tiling are described separately below.

4.1.3.1 Simple tiling

Simple tiling can be conceptualized as simply breaking up a larger image into smaller rectangular “tiles”. Simple tiling is useful for dividing a large image into easier to manage pieces and to allow random access to part of the compressed image.

For simple tiling, the tiles have the same maximum dimensions, with the exception of the tiles on the right and bottom sides of the image when either tiled image maximum dimension is not an integer multiple of the corresponding tile maximum dimension. Tiles are contiguous, non-overlapping and are coded sequentially from left-to-right and top-to-bottom. They all must have the same component identifiers, sampling factors, and be encoded with the same coding process. All tiles, with the possible exception of the tiles on the right and bottom sides of the image, must have an integral number of MCU rows and MCU columns.

4.1.3.2 Pyramidal tiling

Pyramidal tiling offers a method of storing multiple resolution versions of an image within the same compressed data stream. Pyramidal tiling is useful for providing direct access to lower resolution versions of a larger higher resolution image (i.e. versions with smaller maximum dimensions obtained by down-sampling of the image). For example, pyramidal tiling provides the capability to view large images on a screen, as required for browsing through “thumbnail” versions of images in a database or for side-by-side image comparison.

Pyramidal tiling allows for tiles of one resolution level to overlap those of other resolution levels. The lowest resolution level must be positioned first in the data stream and be followed by levels of increasing resolution. Tiles within a single resolution layer must conform to the rules for simple tiling.

4.1.3.3 Composite tiling

For composite tiling, there are no restrictions except that all tiles shall have the same component identifiers. Composite tiling is useful for relating diverse sub-images into an image collage, i.e. a single composite image.

In order to relate different images to each other as tiles of a composite image, it is necessary to define a geometry for each tile. The tile geometry is defined with respect to the tile grid which has width and height equal to the maximum horizontal and vertical dimensions of the image. The dimensions of each of the tile's components are computed from the width and height of the tile grid and the component sampling factors.

The tile geometry also requires the registration for all components to be defined. Component registration is simply the geometric relationship of the samples within a component to the tile grid. The DCR marker segment is used to define component registration by specifying the spatial position of the sample in the upper left corner of each component with respect to the tile grid.

The composite image uses a reference grid to specify the spatial relationship of the individual tiles. The tile grid of each tile is scaled and placed at a specified location within the reference grid. A third grid, the image grid, relates the resolution of the reference grid to the resolution of the intended output device. The use of a higher resolution reference grid can reduce border effects for multiple resolution images.

4.1.4 Still Picture Interchange File Format (SPIFF) extension

The Still Picture Interchange File Format (SPIFF) extension provides for the interchange of image files between application environments. SPIFF is a generic file format intended for interchange only and does not include many of the features found in application-specific formats.

SPIFF is based on the interchange format for compressed image data specified in CCITT Rec. T.81 | ISO/IEC 10918-1 and this Recommendation | International Standard. The interchange format omits certain parameters such as aspect ratio and colour space designation because they are not strictly required for decoding the image component values. SPIFF is a complete coded image representation, i.e. it includes all parameters necessary to reconstruct and present the decoded picture accurately on an output device.

4.1.5 Other extensions

A mandatory version number marker segment has been specified as part of the compressed data formats which utilize the extensions defined by this Recommendation | International Standard and for future extensions. The first version number marker segment has a version specifier which identifies the minimum level of functionality required to decode the entire compressed data stream (for images without composite tiling), or the first tile (for images with composite tiling). Additional version number marker segments may be inserted to identify portions of the stream which require lower levels of functionality (for images without composite tiling), or identify tiles which require other levels of functionality (for images with composite tiling). The version number marker segment also contains a capability indicator field which identifies the coding processes and extensions used in the compressed data stream.

Another extension defined by this Recommendation | International Standard increases the limit on the number of data units per Minimum Coded Unit (MCU) during interleaved encoding to 20. A data unit is a sample for lossless processes and an 8×8 block of samples for DCT-based processes.

4.2 Compliance tests for the extensions

The purpose of compliance tests is to provide designers, manufacturers, or users of a product with a set of procedures for determining whether the product meets a specified set of requirements with some confidence. In addition, the compliance tests specified herein are intended to achieve the following specific goals:

- increase the likelihood of compressed data interchange;
- decrease the likelihood that DCT-based encoders or decoders will yield reduced image quality as a result of computing the DCT or quantization procedures with insufficient accuracy;
- help implementors to meet this Recommendation | International Standard requirements for encoders and decoders as fully as possible.

For each of the processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1, the compliance tests for the above requirements are specified in ITU-T Rec. T.83 | ISO/IEC 10918-2. For each of the extensions defined by this Recommendation | International Standard, the compliance tests are specified in Annex G.

4.2.1 Availability of compliance test data

Standardized compliance test data is used to perform the encoder and decoder compliance tests. There are two types of compliance test data which are used by the encoder compliance tests: source image test data and encoder reference test data. Similarly, there are two types of compliance test data which are used by the decoder compliance tests: compressed test data and decoder reference test data.

Information about compliance test data for the encoder compliance tests and the generic decoder compliance tests is available from ISO and ITU to parties who wish to determine compliance of an encoder or decoder. Information about compliance test data for application-specific compliance tests should be obtained from the standards body which maintains standards for the particular application area.

Contact points:

- International Organization for Standardization
1, rue de Varembe
CH-1211 Genève 20, Switzerland
- International Telecommunication Union
Place des Nations
CH-1211 Genève 20, Switzerland

5 Compressed data format requirements

The compressed data formats specified by CCITT Rec. T.81 ISO/IEC 10918-1 are:

- a) the interchange format;
- b) the abbreviated format for compressed image data;
- c) the abbreviated format for table-specification data.

This Recommendation | International Standard extends the specification of these compressed data formats to support the use of the extensions in combination with the processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1.

The compressed data format requirements are that any compressed image data represented in interchange format, abbreviated format for compressed image data, or abbreviated format for table-specification data shall comply with the syntax and code assignments appropriate for the decoding process and extensions selected, as specified in Annex B of CCITT Rec. T.81 | ISO/IEC 10918-1 and Annex B of this Recommendation | International Standard.

6 Encoder requirements

An encoding process converts source image data to compressed image data. CCITT Rec. T.81 | ISO/IEC 10918-1 specifies a number of distinct encoding processes. This Recommendation | International Standard defines encoding extensions which may be used in combination with the encoding processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1.

An extended encoder is an embodiment of one (or more) of the encoding processes specified in CCITT Rec. T.81 | ISO/IEC 10918-1 used in combination with one or more of the encoding extensions specified herein. In order to comply with this Recommendation | International Standard, an extended encoder shall satisfy the requirements stated in clause 6 of CCITT Rec. T.81 | ISO/IEC 10918-1 and satisfy at least one of the following two additional requirements.

An extended encoder shall:

- a) with appropriate accuracy, convert source image data to compressed image data which comply with the interchange format syntax specified in Annex B for the encoding process(es) and extension(s) embodied by the encoder;
- b) with appropriate accuracy, convert source image data to compressed image data which comply with the abbreviated format for compressed image data syntax specified in Annex B for the encoding process(es) and extension(s) embodied by the encoder.

For each of the processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1, the compliance tests for the above requirements are specified in ITU-T Rec. T.83 | ISO/IEC 10918-2. For each of the extensions defined by this Recommendation | International Standard, the compliance tests for the above requirements are specified in Annex G.

NOTE – There is **no requirement** in this Recommendation | International Standard that any encoder which embodies one of the encoding processes and extensions shall be able to operate for all ranges of the parameters which are allowed for that process and extension. An encoder is only required to meet the compliance tests, and to generate the compressed data format according to Annex B for those parameter values which it does use.

7 Decoder requirements

A decoding process converts compressed image data to reconstructed image data. CCITT Rec. T.81 | ISO/IEC 10918-1 specifies a number of distinct decoding processes. This Recommendation | International Standard defines decoding extensions which may be used in combination with the decoding processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1.

An extended decoder is an embodiment of one (or more) of the decoding processes specified in CCITT Rec. T.81 | ISO/IEC 10918-1 used in combination with one or more of the decoding extensions specified herein. In order to comply with this Recommendation | International Standard, an extended decoder shall satisfy the requirements stated in clause 7 of CCITT Rec. T.81 | ISO/IEC 10918-1 and satisfy all three of the following additional requirements.

An extended decoder shall:

- a) with appropriate accuracy, convert to reconstructed image data any compressed image data with parameters within the range supported by the application, and which comply with the interchange format syntax specified in Annex B for the decoding process(es) and extension(s) embodied by the decoder;
- b) accept and store any table-specification data which comply with the abbreviated format for table-specification data syntax specified in Annex B for the decoding process(es) and extension(s) embodied by the decoder;
- c) with appropriate accuracy, convert to reconstructed image data any compressed image data which comply with the abbreviated format for compressed image data syntax specified in Annex B for the decoding process(es) and extension(s) embodied by the decoder, provided that the table-specification data required for decoding the compressed image data has previously been installed into the decoder.

Annex A

Mathematical definitions

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies the changes and additions to Annex A of CCITT Rec. T.81 | ISO/IEC 10918-1.

A.1 Geometries of image data

The following extensions are in addition to the specifications of CCITT Rec. T.81 | ISO/IEC 10918-1.

A.1.1 Simple tiling

In simple tiling, the tiles with the same maximum dimensions are referred to as 'nominal'. All tiles on the right border, apart from the bottom tile, have the same maximum vertical dimension as the nominal tiles and their maximum horizontal dimension is less than or equal to the maximum horizontal dimension of the nominal tiles. All tiles on the bottom border apart from the rightmost tile have the same maximum horizontal dimension as the nominal tiles and their maximum vertical dimension is equal to or less than the maximum vertical dimension of the nominal tiles. The maximum horizontal dimension of the bottom rightmost tile is the same as the maximum horizontal dimension of the right border tiles and its maximum vertical dimension is the same as the maximum vertical dimension of the bottom border tiles. The maximum horizontal dimension and maximum vertical dimension of the nominal tiles shall be multiples of $8 \cdot H_{\max}$ and $8 \cdot V_{\max}$ respectively so that no padding is required, except possibly in the case of the border tiles.

A.1.2 Tile geometry for composite tiling

For composite tiling, the concept of tile geometry is needed. Note that this concept is not included in CCITT Rec. T.81 | ISO/IEC 10918-1. The tile geometry is defined by additional meaning associated with the X, Y and sampling factors parameters.

The X and Y parameters of the frame header or DHP marker segment shall also define the width and height of the tile. The tile is associated with a rectangular grid made of $X \cdot Y$ square elements. The dimension of each element is one tile grid unit.

For each component the subsampling ratios shall be derived from the H_i and V_i sampling factors as follows:

- The horizontal subsampling ratio (SFH_i) is given by $H_{\max}/H_i$, where $H_{\max}$ is the largest H_i . The vertical subsampling ratio (SFV_i) is given by $V_{\max}/V_i$, where $V_{\max}$ is the largest V_i .

The registration of each component relative to the tile grid (and to each other component) is defined to give "even coverage" of the tile grid by the component samples. The vertical registration distance and horizontal registration distance are specified in units of half a tile grid unit.

The case of $H_{\max} = 4$ (or $V_{\max} = 4$) and where at least one of the other H_i (or V_i) has a value of 3, is not covered properly by the above. The composing of tiles in such a case is not specified and is left to the application.

The number of possible values for the horizontal registration distance of a component is given by $2 \cdot H_{\max}/H_i$. The number of possible values for the vertical registration distance of a component is given by $2 \cdot V_{\max}/V_i$.

Individual tiles used in composite tiling require a component registration selection (DCR) marker segment to specify the spatial positioning of the components.

A.1.3 Reference and image grids

Two additional grids are defined for composite tiling: a reference grid and an image grid. The dimension of each "grid element" of the reference grid is one reference grid unit. The dimension of each "grid element" of the image grid is one image grid unit.

The tile can be scaled up with an integer scaling factor in each dimension and put in the reference grid at the specified offset. The resulting tiled image described on the reference grid can then be scaled down with an integer scaling factor in each dimension, onto a third grid called the image grid. The up-scaling and down-scaling algorithms are not specified.

A.2 DCT coefficient quantization (informative) and dequantization (normative)

The following are extensions to the quantization and dequantization specified in CCITT Rec. T.81 | ISO/IEC 10918-1.

After the FDCT is computed for a block, each of the resulting 63 AC coefficients S_{vu} is quantized by the following equation, where '/' indicates integer division with truncation towards zero:

$$Sq_{vu} = \text{round}\{(S_{vu} * 16) / (Q_{vu} * Q_SCALE)\}$$

The quantizer scaling does not affect the quantization of the DC coefficient:

$$Sq_{00} = \text{round}(S_{00} / Q_{00})$$

The quantizer step size is the element Q_{vu} from the quantization table specified by the frame parameter T_{qi} . The scale factor Q_SCALE is the entry in Table C.1 whose index is the current value of $SCALE_CODE$ (see C.2).

At the decoder, the dequantization is specified by the following equations:

$$R_{vu} = Sq_{vu} * Q_{vu} * Q_SCALE / 16$$

and,

$$R_{00} = Sq_{00} * Q_{00}$$

Annex B

Compressed data formats

(This annex forms an integral part of this Recommendation | International Standard)

This annex specifies the changes and additions to Annex B of CCITT Rec. T.81 | ISO/IEC 10918-1.

B.1 General aspects of the compressed data format specifications

B.1.1 Marker assignments

All markers shall be assigned two-byte codes: a X'FF' byte followed by a second byte which is not equal to X'00' or X'FF'. The second byte is specified in Table B.1 for each defined marker. An asterisk (*) indicates a marker which is significant in its own right and is not the start of a marker segment.

B.1.2 Syntax

In B.2 to B.7 the interchange format syntax is specified. For the purpose of this Recommendation | International Standard, the syntax consists of:

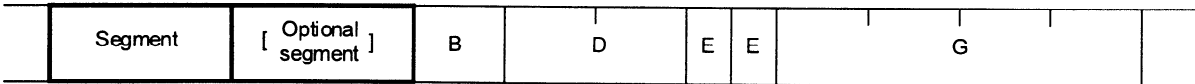
- the required ordering of markers, parameters, and entropy-coded segments;
- identification of optional or conditional constituent parts;
- the name, symbol, and definition of each marker and parameter;
- the allowed values of each parameter;
- any restrictions on the above which are specific to the various coding processes or extensions.

The ordering of constituent parts and the identification of which are optional or conditional is specified by the syntax figures in B.2 to B.7. Names, symbols, definitions, allowed values, conditions, and restrictions are specified immediately below each syntax figure.

B.1.3 Conventions for syntax figures

The syntax figures in B.2 to B.7 are a part of the interchange format specification. The following conventions, illustrated in Figure B.1, apply to these figures:

- **Segment indicator:** A thick-lined box encloses either a marker segment, an entropy-coded data segment, or combinations of these.
- **Optional/conditional indicator:** Square brackets indicate that a marker or marker segment is only optionally or conditionally present in the compressed image data.
- **Parameter/marker indicator:** A thin-lined box encloses either a marker or a single parameter.
- **Parameter length indicator:** The width of a thin-lined box is proportional to the parameter length (4, 8, 16, or 32 bits, shown as E, B, D, and G respectively in Figure B.1) of the marker or parameter it encloses; the width of thick-lined boxes is not meaningful.
- **Ordering:** In the interchange format, a parameter or marker shown in a figure precedes all of those shown to its right, and follows all of those shown to its left;
- **Entropy-coded data indicator:** Angled brackets indicate that the entity enclosed has been entropy encoded.



T0825210-96/d01

Figure B.1 – Syntax notation conventions

Table B.1 – Marker code assignments

Code assignment	Symbol	Description
Start of Frame markers, non-differential, Huffman coding		
X'FFC0'	SOF ₀	Baseline DCT
X'FFC1'	SOF ₁	Extended sequential DCT
X'FFC2'	SOF ₂	Progressive DCT
X'FFC3'	SOF ₃	Lossless (sequential)
Start of Frame markers, differential, Huffman coding		
X'FFC5'	SOF ₅	Differential sequential DCT
X'FFC6'	SOF ₆	Differential progressive DCT
X'FFC7'	SOF ₇	Differential lossless (sequential)
Start of Frame markers, non-differential, arithmetic coding		
X'FFC8'	JPG	Reserved for additional JPEG extensions
X'FFC9'	SOF ₉	Extended sequential DCT
X'FFCA'	SOF ₁₀	Progressive DCT
X'FFCB'	SOF ₁₁	Lossless (sequential)
Start of Frame markers, differential, arithmetic coding		
X'FFCD'	SOF ₁₃	Differential sequential DCT
X'FFCE'	SOF ₁₄	Differential progressive DCT
X'FFCF'	SOF ₁₅	Differential lossless (sequential)
Huffman table specification		
X'FFC4'	DHT	Define Huffman table(s)
Arithmetic coding conditioning specification		
X'FFCC'	DAC	Define arithmetic coding conditioning(s)
Restart interval termination		
X'FFD0' through X'FFD7'	RST _m *	Restart with modulo 8 count "m"
Other markers		
X'FFD8'	SOI*	Start of image
X'FFD9'	EOI*	End of image
X'FFDA'	SOS	Start of scan
X'FFDB'	DQT	Define quantization table(s)
X'FFDC'	DNL	Define number of lines
X'FFDD'	DRI	Define restart interval
X'FFDE'	DHP	Define hierarchical progression
X'FFDF'	EXP	Expand reference component(s)
X'FFE0' through X'FFEF'	APP _n	Reserved for application segments
X'FFF7' through X'FFFD'	JPG _n	Reserved for additional JPEG extensions
X'FFFE'	COM	Comment
Reserved markers		
X'FF01'	TEM*	For temporary private use in arithmetic coding
X'FF02' through X'FFBF'	RES	Reserved
Version 1 extensions		
X'FFF0'	VER	Version
X'FFF1'	DTI	Define tiled image
X'FFF2'	DTT	Define tile
X'FFF3'	SRF	Selectively refined frame
X'FFF4'	SRS	Selectively refined scan
X'FFF5'	DCR	Define component registration
X'FFF6'	DQS	Define quantizer scale selection

B.1.4 Conventions for symbols, code lengths, and values

Following the figure for each syntax defined in B.2 to B.7, the symbol, name, and definition for each marker and parameter shown in the figures, are specified. For each parameter, the length and allowed values are also specified in tabular form.

The following conventions apply to symbols for markers and parameters:

- all marker symbols have three upper-case letters, and some also have a subscript. Examples: SOI, SOF_n;
- all parameter symbols have one or two upper-case letters; some also have one or two lower-case letters and some have subscripts. Examples include: TFvs, Y, Nf, H_i, Tq_i.

B.2 General sequential and progressive syntax

This subclause specifies the interchange format syntax which applies to all coding processes for sequential DCT-based, progressive DCT-based, and lossless modes of operation.

B.2.1 Scan header syntax

Figure B.2 specifies the scan header which shall be present at the start of a scan. This header specifies which component(s) are contained in the scan, specifies the destinations from which the entropy tables to be used with each component are retrieved, and (for the progressive DCT) specifies which part of the DCT quantized coefficient data is contained in the scan. For lossless processes the scan parameters specify the predictor and the point transform.

NOTE – If there is only one image component present in a scan, that component is, by definition, non-interleaved. If there is more than one image component present in a scan, the components present are, by definition, interleaved.

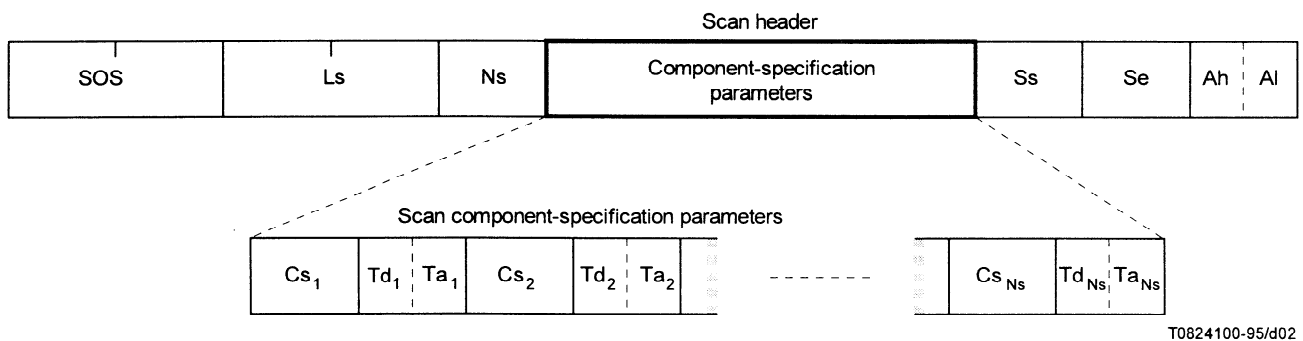


Figure B.2 – Scan header syntax

The marker and parameters shown in Figure B.2 are defined below. The size and allowed values of each parameter are given in Table B.2.

- SOS:** Start of scan marker – Marks the beginning of the scan parameters.
- Ls:** Scan header length – Specifies the length in bytes of the scan header shown in Figure B.2. This length parameter encodes the number of bytes in the marker segment, including the length parameter but excluding the two byte marker.
- Ns:** Number of image components in scan – Specifies the number of source image components in the scan. The value of Ns shall be equal to the number of sets of scan component parameters (Cs_j, Td_j, and Ta_j) present in the scan header.

- **Cs_j:** Scan component selector – Selects which of the N_f image components specified in the frame parameters shall be the jth component in the scan. Each Cs_j shall match one of the C_i values specified in the frame header, and the ordering in the scan header shall follow the ordering in the frame header. If N_s > 1, the order of interleaved components in the MCU is Cs₁ first, Cs₂ second, etc. If N_s > 1, the following restriction shall be placed on the image components contained in the scan:
 - For compressed data streams which do not use the 20 blocks per MCU extension defined in this Recommendation | International Standard:

$$\sum_{j=1}^{N_s} H_j \times V_j \leq 10$$

- For compressed data streams which use the 20 blocks per MCU extension defined in this Recommendation | International Standard:

$$\sum_{j=1}^{N_s} H_j \times V_j \leq 20$$

where H_j and V_j are the horizontal and vertical sampling factors for scan component j. These sampling factors are specified in the frame header for component i, where i is the frame component index for which frame component identifier C_i matches scan component selector Cs_j.

As an example, consider an image having 3 components with maximum dimensions of 512 lines and 512 samples per line, and with the following sampling factors:

Component 0	H ₀ = 4,	V ₀ = 1
Component 1	H ₁ = 1,	V ₁ = 2
Component 2	H ₂ = 2,	V ₂ = 2

Then the summation of H_j × V_j is (4 × 1) + (1 × 2) + (2 × 2) = 10.

The value of Cs_j shall be different from the values of Cs₁ to Cs_{j-1}.

- **Td_j:** DC entropy coding table destination selector – Specifies one of four possible DC entropy coding table destinations from which the entropy table needed for decoding of the DC coefficients of component Cs_j is retrieved. The DC entropy table shall have been installed in this destination (see B.2.4.2 and B.2.4.3 in CCITT Rec. T.81 | ISO/IEC 10918-1) by the time the decoder is ready to decode the current scan. This parameter specifies the entropy coding table destination for the lossless processes.
- **Ta_j:** AC entropy coding table destination selector – Specifies one of four possible AC entropy coding table destinations from which the entropy table needed for decoding of the AC coefficients of component Cs_j is retrieved. The AC entropy table selected shall have been installed in this destination (see B.2.4.2 and B.2.4.3 CCITT Rec. T.81 | ISO/IEC 10918-1) by the time the decoder is ready to decode the current scan. This parameter is zero for the lossless processes.
- **Ss:** Start of spectral or predictor selection – In the DCT modes of operation, this parameter specifies the first DCT coefficient in each block in zigzag order which shall be coded in the scan. This parameter shall be set to zero for the sequential DCT processes. In the lossless mode of operations this parameter is used to select the predictor.
- **Se:** End of spectral selection – Specifies the last DCT coefficient in each block in zigzag order which shall be coded in the scan. This parameter shall be set to 63 for the sequential DCT processes. In the lossless mode of operations this parameter has no meaning. It shall be set to zero.

- **Ah:** Successive approximation bit position high – This parameter specifies the point transform used in the preceding scan (i.e. successive approximation bit position low in the preceding scan) for the band of coefficients specified by Ss and Se. This parameter shall be set to zero for the first scan of each band of coefficients. In the lossless mode of operations this parameter has no meaning. It shall be set to zero.
- **Al:** Successive approximation bit position low or point transform – In the DCT modes of operation this parameter specifies the point transform, i.e. bit position low, used before coding the band of coefficients specified by Ss and Se. This parameter shall be set to zero for the sequential DCT processes. In the lossless mode of operations, this parameter specifies the point transform, Pt.

The entropy coding table destination selectors, Td_j and Ta_j, specify either Huffman tables (in frames using Huffman coding) or arithmetic coding tables (in frames using arithmetic coding). In the latter case, the entropy coding table destination selector specifies both an arithmetic coding conditioning table destination and an associated statistics area.

Table B.2 – Scan header parameter size and values

Parameter	Size (bits)	Values			
		Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Ls	16	6 + 2 × Ns			
Ns	8	1-4			
Csj	8	0-255 ^{a)}			
Tdj	4	0-1	0-3	0-3	0-3
Taj	4	0-1	0-3	0-3	0
Ss	8	0	0	0-63	1-7 ^{b)}
Se	8	63	63	Ss-63 ^{c)}	0
Ah	4	0	0	0-13	0
Al	4	0	0	0-13	0-15
a) Csj Shall be a member of the set of C_i specified in the frame header. b) 0 For lossless differential frames in the hierarchical mode. c) 0 If Ss equals zero. NOTE – The contents of this table are identical to Table B.3 in CCITT Rec. T.81 ISO/IEC 10918-1					

B.3 Version marker segment syntax

Inclusion of the version marker segment is mandatory for a data stream which uses any of the capabilities listed in Table B.5. The first version marker segment shall appear after SOI and before any other marker apart from APPn or COM marker segments. Except when using composite tiling, its parameters shall match the capabilities needed to decode the complete compressed image data stream. For composite tiling its parameters shall match the capabilities needed to decode the first tile (see Annex E).

A version marker segment with 0.0 version specifier shall not appear as the first (or only) version marker segment. This will provide protection for version 0.0 decoders which are not required to interpret (or jump over) version marker segments.

Other version marker segments may appear in any place of the bitstream where tables or miscellaneous marker segments are legal. These additional marker segments (if they appear) shall describe the capability needed to decode the subsequent part of the bitstream until the EOI marker or the next version marker segment. For images without composite tiling, these additional marker segments are optional. For composite tiling, if the level of functionality specified in the last encountered version marker segment which appears before a tile (and not within a tile) is not sufficient to decode the next tile, an additional marker segment shall appear before that tile that will specify the required functionality.

Figure B.3 specifies the version marker segment.

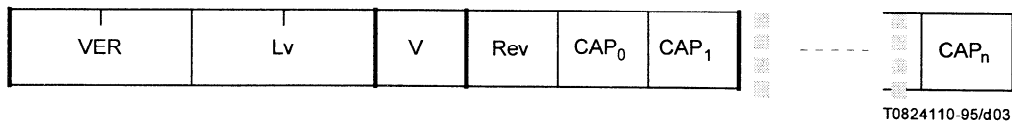


Figure B.3 – Version marker segment syntax

The marker and parameters shown in Figure B.3 are defined below. The size and allowed values of each parameter are given in Table B.3:

- **VER:** Version marker – Marks the beginning of the version marker parameters.
- **Lv:** Version marker segment length – Specifies the sum total length of all version marker segment parameters shown in Figure B.3
- **V:** Version number – Specifies the major version.
- **Rev:** Revision number – Specifies the minor revision of the version.

The two bytes composite parameter <version number>.<revision number> (**V.Rev**) is referred to as the Version specifier.

The version specifier for CCITT Rec. T.81 | ISO/IEC 10918-1 is 0.0. The first revision with the extensions described in ITU-T Rec. T.84 | ISO/IEC 10918-3 shall have the version specifier 1.0.

Encoders shall indicate the smallest version specifier having sufficient capabilities to decode the bitstream, except that a version marker segment with 0.0 version specifier shall not appear as the first (or only) version marker segment.

Decoders shall not attempt to decode the subsequent part of the bitstream if the major version number is greater than that implemented by the decoder. In this event, decoders shall parse but not decode the subsequent compressed data until the next version marker segment or the end of the compressed data stream is reached. If the major version number code is less than or equal to that implemented by the decoder, the decoder shall attempt to decode the subsequent part of the bitstream regardless of the value of the minor revision number.

- **CAP_i:** Capability indicator – Added capability indicator for version number *i*. The number of capability indicator bytes shall be equal to **V** + 1.

Table B.3 – Version marker segment parameter sizes and values

Parameter	Size (bits)	Values			
		Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Lv	16	5, V = 0 6, V = 1			
V	8	0, 1			
Rev	8	0			
CAP _i	8 8	CAP ₀ use Table B.4, version = 0 CAP ₁ use Table B.5, version = 1 CAP _i (i > 1) reserved for future versions			

The bits of the capability indicator byte for Version 0 have the following meaning (from most significant to least significant): Hierarchical, lossless, progressive, full progression, 12-bits, arithmetic, and extended. Only the 29 combinations defined in Table B.4 are legal for $V=0$. Note that for the hierarchical processes, the capability indicator coded is the largest of the binary values which represent the capabilities for each stage of the hierarchical progression. Also, if the first frame of a hierarchical progression uses a DCT-based process, the capability indicator shall indicate a DCT-based process whether or not there is a final lossless stage.

Table B.4 – Capability indicator byte for Version 0

Number	Coding process	CAP ₀ value
1	Baseline sequential	0000 0000
2	Extended sequential, Huffman, 8-bits	0000 0001
3	Extended sequential, arithmetic, 8-bits	0000 0011
4	Extended sequential, Huffman, 12-bits	0000 0101
5	Extended sequential, arithmetic, 12-bits	0000 0111
6	Spectral selection, Huffman, 8-bits	0001 0001
7	Spectral selection, arithmetic, 8-bits	0001 0011
8	Full progression, Huffman, 8-bits	0001 1001
9	Full progression, arithmetic, 8-bits	0001 1011
10	Spectral selection, Huffman, 12-bits	0001 0101
11	Spectral selection, arithmetic, 12-bits	0001 0111
12	Full progression, Huffman, 12-bits	0001 1101
13	Full progression, arithmetic, 12-bits	0001 1111
14	Lossless, Huffman	0010 0001
15	Lossless, arithmetic	0010 0011
16	Hierarchical, sequential Huffman, 8-bits	0100 0001
17	Hierarchical, sequential arithmetic, 8-bits	0100 0011
18	Hierarchical, sequential, Huffman, 12-bits	0100 0101
19	Hierarchical, sequential, arithmetic, 12-bits	0100 0111
20	Hierarchical, Spectral selection, Huffman, 8-bits	0101 0001
21	Hierarchical, Spectral selection, arithmetic, 8-bits	0101 0011
22	Hierarchical, Full progression, Huffman, 8-bits	0101 1001
23	Hierarchical, Full progression, arithmetic, 8-bits	0101 1011
24	Hierarchical, Spectral selection, Huffman, 12-bits	0101 0101
25	Hierarchical, Spectral selection, arithmetic, 12-bits	0101 0111
26	Hierarchical, Full progression, Huffman, 12-bits	0101 1101
27	Hierarchical, Full progression, arithmetic, 12-bits	0101 1111
28	Hierarchical, Lossless, Huffman	0110 0001
29	Hierarchical, Lossless, arithmetic	0110 0011

The bits of the additional capability indicator byte for Version 1 shall have the following meaning (from least significant to most significant): up to 20 blocks per MCU, variable quantization, selective refinement (3 bits), tiling type (2-bits), and 1 reserved bit. The reserved bit, bit 7, shall be set to zero (see Table B.5).

Table B.5 – Capability indicator byte for Version 1

Capability	Bit positions
10 < blocks per MCU <= 20	0xxx xxx1
Variable quantization	0xxx xx1x
Hierarchical selective refinement	0xxx x1xx
Progressive selective refinement	0xxx 1xxx
Component selective refinement	0xx1 xxxx
Tiling	
No tiling	000x xxxx
Simple tiling	001x xxxx
Pyramidal tiling	010x xxxx
Composite tiling	011x xxxx
NOTE – “x” indicates “don’t care”.	

B.4 Selective refinement syntax

B.4.1 General

Selective refinement refers to selecting a (rectangular) region of one or more of the image components for further refinement.

There are several types of selective refinement. One is when only a region of one or more components is further refined by the next differential frame of a hierarchical sequence. This type of refinement is called hierarchical selective refinement. The syntax for hierarchical selective refinement is specified in B.4.2.

A second type of selective refinement provides for the next scan to add more coefficients, more bits to the coefficients, or both, to a selected region of one or more components. This type includes spectral selection progressive selective refinement, successive approximation progressive selective refinement, and progressive selective refinement combining both spectral selection and successive approximation. The syntax for progressive selective refinement is specified in B.4.3.

The third type of selective refinement allows for some colour components to be used in selected regions of the image only. This type is called component selective refinement. The syntax for component selective refinement is the same as for progressive selective refinement specified in B.4.3.

B.4.2 Hierarchical selective refinement syntax

Hierarchical selective refinement of a differential frame is signalled by the inclusion of a Selectively Refined Frame (SRF) marker segment as one of the table-specification segments or miscellaneous marker segments preceding the differential frame. A differential frame which completely refines the previous layer (as in CCITT Rec. T.81 | ISO/IEC 10918-1) shall not have a preceding SRF marker segment.

Figure B.4 shows the Selectively Refined Frame (SRF) marker segment syntax.



T0824240-95/d04

Figure B.4 – Selectively refined frame marker segment syntax

The marker and parameters shown in Figure B.4 are defined below. The size and allowed values of each parameter are given in Table B.6.

- **SRF:** Selectively refined frame marker – Marks the beginning of the selectively refined frame parameters.
- **Lrf:** Selectively refined frame marker segment length – Specifies the sum total length of all selectively refined frame parameters shown in Figure B.4.
- **Ovf:** Selectively refined frame vertical offset – Specifies the maximum distance, in lines, between the top border of the refined region of each component and the top border of the component array (possibly upsampled from previous stages).
- **Ohf:** Selectively refined frame horizontal offset – Specifies the maximum distance, in samples, between the left border of the refined region of each component and the left border of the component (possibly upsampled from previous stages).

The maximum horizontal and vertical dimensions specified in the SOF marker segment of the differential frame which is preceded by a SRF marker segment shall specify the reduced maximum dimensions of the component regions being refined. The region of each component being refined shall fall entirely within the dimensions of the component array that was upsampled from the previous stages.

Table B.6 – Selectively refined frame parameter sizes and values

Parameter	Size (bits)	Values
Lrf	16	6
Ovf	16	$0 \leq \text{Ovf} \leq 2^{16} - 1$
Ohf	16	$0 \leq \text{Ohf} \leq 2^{16} - 1$

B.4.3 Progressive and component selective refinement syntax

Progressive and component selective refinement of a scan is signalled by the inclusion of a Selectively Refined Scan (SRS) marker segment as one of the table-specification segments or miscellaneous marker segments preceding that scan. A scan which completely “covers” the whole image (as in CCITT Rec. T.81 | ISO/IEC 10918-1) shall not have a preceding SRS marker segment. Also, component regions defined by a selectively refined scan shall not partially overlap the component regions defined by a preceding scan containing the same components.

Figure B.5 shows the selectively refined scan (SRS) marker segment syntax.



T0825220-96/d05

Figure B.5 – Selectively refined scan marker segment syntax

The marker and parameters shown in Figure B.5 are defined below. The size and allowed values of each parameter are given in Table B.7.

- **SRS:** Selectively refined scan marker – Marks the beginning of the selectively refined scan parameters.
- **Lrs:** Selectively refined scan marker segment length – Specifies the length of all selectively refined scan parameters shown in Table B.7.
- **Ovs:** Selectively refined scan vertical offset – Specifies the maximum distance, in lines, between the top border of the refined region of each component and the top border of the component array.