Network Working Group
Request for Comments: 2306
Category: Informational

G. Parsons
Northern Telecom
J. Rafferty
Human Communications

March 1998

Tag Image File Format (TIFF) - F Profile for Facsimile

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Overview

This document describes in detail the definition of TIFF-F that is used to store facsimile images. The TIFF-F encoding has been folklore with no standard reference definition before this document.

Internet Fax Working Group

This document is a product of the IETF Internet Fax Working Group. All comments on this document should be forwarded to the email distribution list at <ietf-fax@imc.org>.

1. Abstract

This document references the Tag Image File Format (TIFF) to define the F profile of TIFF for facsimile (TIFF-F) as a file format that may be used for the storage and interchange of facsimile images.

2. TIFF Definition

TIFF (Tag Image File Format) Revision 6.0 is defined in detail within [TIFF].

A brief review of concepts used in TIFF is included in this document as background information, but the reader is directed to the original TIFF specification [TIFF] to obtain specific technical details.

2.1 Baseline TIFF and Applications

TIFF provides a method to describe and store raster image data. A primary goal of TIFF is to provide a rich environment within which implementations can exchange image data. [TIFF] characterizes Baseline TIFF as being the core of TIFF, the essentials that all mainstream TIFF developers should support in their products. Applications of TIFF are defined by using Baseline TIFF as a starting point and then defining "extensions" to TIFF that are used for the specific "application", as well as specifying any other differences from Baseline TIFF.

3. TIFF-F Definition

3.1 Introduction

Though it has been in common usage for many years, TIFF-F has previously never been documented in the form of a standard. An informal TIFF-F document was originally created by a small group of fax experts led by Joe Campbell. The existence of TIFF-F is noted in [TIFF] but it is not defined. This document defines the F application of [TIFF]. For ease of reference, the term TIFF-F will be used throughout this document as a shorthand for "F Profile of TIFF for Facsimile". TIFF-F files are intended for use with the image/tiff MIME media content-type which includes support for the "application" parameter (e.g., application=faxbw).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [REQ].

3.1.1 TIFF-F Historical Background

Up until TIFF 6.0, TIFF supported various "Classes" which defined the use of TIFF for various applications. Classes were used to support specific applications and in this spirit, TIFF-F has been known historically as "TIFF Class F". Previous informal TIFF-F documents used the "Class F" terminology.

As of TIFF 6.0 [TIFF], the TIFF Class concept has been eliminated in favor of the concept of Baseline TIFF. Therefore, this document updates the definition of TIFF-F as the F profile of TIFF for facsimile, by using Baseline TIFF as defined in [TIFF] as the starting point and then defining the differences from Baseline TIFF which apply for TIFF-F. In almost all cases, the resulting definition of TIFF-F fields and values remains consistent with those used historically in earlier definitions of TIFF Class F. Where some

of the values for fields have been updated to provide more precise conformance with the ITU-T [T.4] and [T.30] fax recommendations, these differences are noted.

3.1.2 Overview

The intent of this specification is to document:

1) The fields and values which are applicable for this F profile of TIFF for facsimile.
2) A minimum set of TIFF-F fields and values which should be able to interwork with virtually all historic TIFF-F readers.
3) A broader range of values for the traditional TIFF-F fields that will provide support for the most widely used facsimile compressions, page sizes and resolutions, consistent with the ITU-T [T.4] and [T.30] recommendations.

The structure of the TIFF-F definition will be as follows. A brief review of the structure of TIFF files and practical guidelines for the writing and reading of multi-page TIFF-F files is provided in sections 3.1.3 and 3.1.4.

A review of TIFF-F fields follows. Section 3.2 reviews the fields from Baseline TIFF that are applicable for black and white (bi-level) images and are also used by TIFF-F.

Section 3.3 reviews the other required TIFF-F fields. Several fields that are specific extensions for TIFF-F are reviewed in section 3.4. There are also fields that may be helpful, but are not required. These recommended fields are listed in the section 3.5. Section 3.6 defines the requirements for the minimum subset of TIFF-F fields and values to maximize interoperability. Several technical topics, including implementation issues and warnings are discussed in subsequent sections. Finally, section 3.9 introduces the TIFF-F Reader and Writer. A table of the required and recommended fields for a TIFF-F Reader is provided, along with details on the permitted set of values.

3.1.3 Structure of TIFF Files

The structure of TIFF files is specified within [TIFF]. In this section, a short summary of the TIFF structure is included for the informational purposes. In addition, some practical guidelines for the use of this structure in reading and writing TIFF-F files are addressed in the following section 3.1.4. The structure for writing "minimum subset" TIFF-F files is defined in section 3.6.2.

A TIFF file begins with an 8-byte image file header that defines the byte order used within a file (see section 3.9.1), includes a magic number sequence that identifies the content as a TIFF file, and then uses an offset to point to the first Image File Directory (IFD). An IFD is a sequence of tagged fields, sorted in ascending order (by tag value), that contains attributes of an image and pointers to the image data. TIFF fields (also called entries) contain a tag, its type (e.g. short, long, rational, etc.), a count (which indicates the number of values/offsets) and a value/offset. However, the actual value for the field will only be present if it fits into 4 bytes; otherwise, an offset will be used to point to the location of the data associated with the field. In turn, this offset may itself be used to point to an array of offsets.

For the case of facsimile data, many documents consist of a series of multiple pages. Within TIFF, these may be represented using more than one IFD within the TIFF file. Each IFD defines a subfile whose type is given in the NewSubfileType field. For the case of facsimile data that is placed in a TIFF-F file, each facsimile page in a multi-page document has its own IFD. For bi- level facsimile files, multiple IFDs are organized as a linked list, with the last entry in each IFD pointing to the next IFD (the pointer in the last IFD is 0). (There is also another technique for organizing multiple IFDs as a tree, that uses the SubIFDs field, but this technique is not applicable for TIFF-F images.) Within each IFD, the location of the related image data is defined by using fields that are associated with strips. These fields identify the size of strips (in rows), the number of bytes per strip after compression and a strip offset, which is used to point to the actual location of the image strip.

TIFF has a very flexible file structure, but the use of some practical guidelines for implementors when writing multi-page TIFF-F files can produce TIFF structures which are easier for readers to process. This is especially for implementations in environments such as facsimile terminals where a complex file structure is difficult to support.

3.1.4 Practical Guidelines for Writing/Reading Multi-Page TIFF-F Files

Traditionally, historical TIFF-F has required readers and writers to be able to handle multi-page TIFF-F files. Based on the experience of various TIFF-F implementors, it has been seen that the implementation of TIFF-F can be greatly simplified if certain practical guidelines are followed when writing multi-page TIFF-F files. However, for interchange robustness, TIFF-F readers SHOULD be prepared to read TIFF files whose structure is consistent with [TIFF], which supports a more flexible file structure than is recommended here.

The structure for a multi-page TIFF-F file will include one IFD per page of the document. Therefore, each IFD will define the attributes for a single page. For simplicity, the writer of TIFF- F files SHOULD present IFDs in the same order as the actual sequence of pages. (The pages are numbered within TIFF-F beginning with page 0 as the first page and then ascending (i.e. 0, 1, 2,...). However, as noted in section 3.1.3, any field values over 4 bytes will be stored separately from the IFD. TIFF-F readers SHOULD expect IFDs to be presented in page order, but be able to handle exceptions.

Per [TIFF], the exact placement of image data is not specified. However, the strip offsets for each strip of image are defined from within each IFD. Where possible, a second simplifying assumption for the writing of TIFF-F files is to specify that the image data for each page of a multi-page document SHOULD be contained within a single strip (i.e. one image strip per fax page). The use of a single image strip per page is very useful for implementations such as store and forward messaging, where the file is usually prepared in advance of the transmission, but other assumptions may apply for the size of the image strip for implementations which require the use of "streaming" techniques (see section 3.7.6). In the event a different image strip size assumption has been used (e.g. constant size for image strips which may be less than the page size), this will immediately be evident from the values/offsets of the fields that are related to strips. From the TIFF-F reader standpoint, one image strip per page permits the image data to be found through reference via a single offset, resulting in a much simplified image structure and faster processing.

A third simplifying assumption is that each IFD SHOULD be placed in the TIFF-F file structure at a point which precedes the image which the IFD describes. If any long field values are present (see section 3.1.3) then these SHOULD be placed after their referencing IFD and before the image data they describe.

A fourth simplifying assumption for TIFF-F writers and readers is to place the actual image data in a physical order within the TIFF file structure which is consistent with the logical page order. In practice, TIFF-F readers will need to use the strip offsets to find the exact physical location of the image data, whether or not it is presented in logical page order.

TIFF-F writers MAY make a fifth simplifying assumption, in which the IFD, the value data and the image data for which the IFD has offsets precede the next image IFD. These elements MUST precede the next image IFD in the minimum set TIFF-F files (see section 3.6.2). However, this principle has been relaxed in the case of TIFF-F to reflect past practices.

So, a TIFF-F file which is structured using the guidelines of this section will essentially be composed of a linked list of IFDs, presented in ascending page order, which in turn each point to a single page of image data (one strip per page), where the pages of image data are also placed in a logical page order within the TIFF-F file structure. (The pages of image data may themselves be stored in a contiguous manner, at the option of the implementor).

3.2 Baseline TIFF Required Fields for BiLevel Images

Baseline TIFF per [TIFF] requires that the following fields be present for all BiLevel Images: ImageWidth, ImageLength, Compression, PhotometricInterpretation, StripOffsets, RowsPerStrip, StripByteCounts, XResolution, YResolution and ResolutionUnit. TIFF-F uses all of these fields, but in some cases specifies a different range of acceptable values than Baseline TIFF. Per [TIFF], if fields are omitted, the Baseline TIFF default value(if specified) will apply.

In the field definitions which follow in this section and subsequent sections, the fields will be presented in the following form:

Fieldname (tag-number) = values (if applicable). TYPE

A brief summary of the Baseline TIFF fields and their use in TIFF-F follows:

ImageWidth(256) = 1728, 2048, 2432, 2592, 3072, 3648, 3456, 4096, 4864.

SHORT or LONG. These are the fixed page widths in pixels. The permissible values are dependent upon X and Y resolutions as shown in sections 2 and 3 of [T.4] and reproduced here for convenience:

XResolution x Yresolution	ImageWidth
204x98, 204x196, 204x391, 200x100, 200x200	1728, 2048, 2432
300x300	2592, 3072, 3648
408x391, 400x400	3456, 4096, 4864

Historical TIFF-F did not include support for the following widths related to higher resolutions: 2592, 3072, 3648, 3456, 4096 and 4864. Historical TIFF-F documents also included the following values related to A5 and A6 widths: 816 and 1216. Per the most recent version of [T.4], A5 and A6 documents are no

longer supported in Group 3 facsimile, so the related width values are now obsolete. See section 3.8.2 for more information on inch/metric equivalencies and other implementation details.

ImageLength (257). SHORT or LONG. LONG recommended.
The total number of scan lines in the image.

Compression (259) = 3,4. SHORT.
This is a required TIFF-F field. The permitted values for TIFF-F purposes are 3 and 4 as shown. The default value per Baseline TIFF is 1 (Uncompressed), but this value is invalid for facsimile images. Baseline TIFF also permits use of value 2 (Modified Huffman encoding), but the data is presented in a form which does not contain EOLs. Instead, TIFF-F specifies the value 3 for encoding one-dimensional T.4 Modified Huffman or 2-dimensional Modified READ data. The detailed settings which apply for T.4 encoded data are specified using the T4Options field. TIFF-F also permits use of the value 4 for the compression field, which indicates that the data is coded using a [T.6] compression method (i.e the Modified Modified READ two-dimensional method). The detailed settings which apply for T.6 encoded data are specified using the T6Options field.

Please refer to the definitions of the T4Options and T6Options fields in section 3.3, and section 3.8 for more information on the encoding of images and conventions used within TIFF-F.

PhotometricInterpretation (260) = 0,1. SHORT.
This field allows notation of an inverted ("negative") image:
0 = normal
1 = inverted

StripOffsets (273). SHORT or LONG.
For each strip, the offset of that strip. The offset is measured from the beginning of the file. If a page is expressed as one large strip, there is one such entry per page.

RowsPerStrip (278). SHORT or LONG. LONG recommended.
The number of scan lines per strip. When a page is expressed as one large strip, this is the same as the ImageLength field.

StripByteCounts (279). LONG or SHORT. LONG recommended.
For each strip, the number of bytes in that strip. If a page is expressed as one large strip, this is the total number of bytes in the page after compression. Note that the choice of LONG or SHORT depends upon the size of the strip.

ResolutionUnit (296) = 2,3. SHORT.
The units of measure for resolution:
2 = Inch
3 = Centimeter

TIFF-F has traditionally used inch based measures.

XResolution (282) = 204, 200, 300, 400, 408 (inches). RATIONAL.
The horizontal resolution of the TIFF-F image expressed in pixels per resolution unit. The values of 200 and 408 have been added to the historical TIFF-F values, for consistency with [T.30]. Some existing TIFF-F implementations may also support values of 77 (cm). See section 3.8.2 for more information on inch/metric equivalencies and other implementation details.

YResolution (283) = 98, 196, 100, 200, 300, 391, 400 (inches). RATIONAL.
The vertical resolution of the TIFF-F image expressed in pixels per resolution unit. The values of 100, 200, and 391 have been added to the historical TIFF-F values, for consistency with [T.30]. Some existing TIFF-F implementations may also support values of 77, 38.5 (cm). See section 3.8.2 for more information on inch/metric equivalencies and other implementation details.

3.3 TIFF-F Required Fields

In addition to the Baseline TIFF fields, there are additional required fields for TIFF-F. A review of the additional required fields for TIFF-F follows:

BitsPerSample (258) = 1. SHORT.
Since TIFF-F is only used for black-and-white facsimile images, the value is 1 (the default) for all files.

FillOrder (266) = 1, 2. SHORT.
TIFF F readers must be able to read data in both bit orders, but the vast majority of facsimile products store data LSB first, exactly as it appears on the telephone line.
1 = Most Significant Bit first.
2 = Least Significant Bit first.

NewSubFileType (254)= (Bit 1 = 1). LONG.
This field is made up of 32 flag bits. Unused bits are expected to be 0 and bit 0 is the low order bit. Bit 0 is set to 0 for TIFF-F. Bit 1 is always set to 1 for TIFF-F, indicating a single page of a multi-page image. The same bit

settings are used when TIFF-F is used for a one page fax image. See sections 3.1.1 and 3.1.2 for more details on the structure of multi-page TIFF-F image files.

PageNumber (297). SHORT/SHORT.
This field specifies the page numbers in the fax document. The field comprises two SHORT values: the first value is the page number, the second is the total number of pages. Single-page documents therefore use 0000/0001 hex. If the second value is 0, the total number of pages in the document is not available.

SamplesPerPixel (277) = 1. SHORT.
The value of 1 denotes a bi-level, grayscale, or palette color image.

There is also a requirement to include either the T4Options or the T6Options field in a TIFF-F IFD, depending upon the setting of the Compression field. These fields are defined in the next section on TIFF extensions.

3.4 TIFF-F Extensions

These are fields which are extensions beyond the required TIFF-F fields. The following fields have been defined as extensions in [TIFF].

T4Options (292) (Bit 0 = 0 or 1, Bit 1 = 0, Bit 2 = 0 or 1). LONG.
This field is required if the value for the compression field has been set to 3. The values are set as shown below for TIFF-F. For TIFF-F, uncompressed data is not allowed and EOLs MAY be byte aligned (see section 3.8.3).

```
bit 0 = 0 for 1-Dimensional, 1 for 2-Dimensional (MR)
bit 1 = must be 0 (uncompressed data not allowed)
bit 2 = 0 for non-byte-aligned EOLs or 1 for byte-
        aligned EOLs
```

This field is made up of a set of 32 flag bits. Unused bits must be set to 0. Bit 0 is the low order bit. Please note that T4Options was known as G3Options in earlier versions of TIFF and TIFF-F. The data in a TIFF-F image encoded using one of the T.4 methods is not terminated with an RTC (see section 3.8.5).

T6Options (293) = (Bit 0 = 0, Bit 1 = 0) LONG.
This field is required for TIFF-F if value of the compression field has been set to 4. The value for this field is made up of a set of 32 flag bits. Setting bit 0 to 0 indicates that the data is compressed using the Modified Modified READ (MMR) two-

dimensional compression method. MMR compressed Data is two-dimensional and does not use EOLs. Each MMR encoded image MUST include an "end-of-facsimile-block" (EOFB) code at the end of each coded strip (see section 3.8.6). Uncompressed data is not applicable for bi-level facsimile images, so that bit 1 must be set to 0. Unused bits must be set to 0. Bit 0 is the low-order bit. The default value is 0 (all bits 0).

bit 0 = 0 for 2-Dimensional
bit 1 = must be 0 (uncompressed data not allowed)

In earlier versions of TIFF, this field was named Group4Options. The significance has not changed and the present definition is compatible.

In addition, three new fields, defined as TIFF-F extensions, describe page quality. The information contained in these fields is usually obtained from receiving facsimile hardware (if applicable). These fields are optional. They SHOULD NOT be used in writing TIFF-F files for facsimile image data that is error corrected or otherwise guaranteed not to have coding errors.

Some implementations need to understand exactly the error content of the data. For example, a CAD program might wish to verify that a file has a low error level before importing it into a high- accuracy document. Because Group 3 facsimile devices do not necessarily perform error correction on the image data, the quality of a received page must be inferred from the pixel count of decoded scan lines. A "good" scan line is defined as a line that, when decoded, contains the correct number of pixels. Conversely, a "bad" scan line is defined as a line that, when decoded, comprises an incorrect number of pixels.

BadFaxLines (326). SHORT or LONG
This field reports the number of scan lines with an incorrect number of pixels encountered by the facsimile during reception (but not necessarily in the file).

Note: PercentBad = (BadFaxLines/ImageLength) * 100

CleanFaxData (327). SHORT
N =
0 = Data contains no lines with incorrect pixel counts or regenerated lines (i.e., computer generated)
1 = Lines with an incorrect pixel count were regenerated by receiving device

2 = Lines with an incorrect pixel count are in the data and were not regenerated by receiving device (i.e. data contains bad scan lines)

Many facsimile devices do not actually output bad lines. Instead, the previous good line is repeated in place of a bad line. Although this substitution, known as line regeneration, results in a visual improvement to the image, the data is nevertheless corrupted. The CleanFaxData field describes the error content of the data. That is, when the BadFaxLines and ImageLength fields indicate that the facsimile device encountered lines with an incorrect number of pixels during reception, the CleanFaxData field indicates whether these bad lines are actually still in the data or if the receiving facsimile device replaced them with regenerated lines.

ConsecutiveBadFaxLines (328). LONG or SHORT.
This field reports the maximum number of consecutive lines containing an incorrect number of pixels encountered by the facsimile device during reception (but not necessarily in the file).

The BadFaxLines and ImageLength data indicate only the quantity of such lines. The ConsecutiveBadFaxLines field is an indicator of their distribution and may therefore be a better general indicator of perceived image quality.

3.5 Recommended Fields

hese are fields that MAY be used in encoding TIFF-F files, but are ptional in nature and may be ignored by many TIFF readers. These ields are called recommended consistent with historical TIFF-F ractice.

BadFaxLines (326) [defined in section 3.4]

CleanFaxData (327) [defined in section 3.4]

ConsecutiveBadFaxLines (328) [defined in section 3.4]

DateTime (306). ASCII.
Date and time in the format YYYY:MM:DD HH:MM:SS, in 24-hour format. String length including NUL byte is 20 bytes. Space between DD and HH.

DocumentName (269). ASCII.
This is the name of the document from which the document was scanned.

ImageDescription (270). ASCII.
This is an ASCII string describing the contents of the image.

Orientation (274). SHORT.
This field is designated as "Recommended" for consistency with historical TIFF-F, but is also a Baseline TIFF field with a default value of 1 per [TIFF]. The default value of 1 applies if the field is omitted, but for clarity, TIFF-F writers SHOULD include this field. This field might be useful for displayers that always want to show the same orientation, regardless of the image. The default value of 1 is "0th row is visual top of image, and 0th column is the visual left." An 180-degree rotation is 3. See [TIFF] for an explanation of other values.

Software (305). ASCII.
The optional name and release number of the software package that created the image.

3.6 Requirements for TIFF-F Minimum Subset

This section defines the requirements for a minimum subset of TIFF-F fields and values that all TIFF-F readers SHOULD support to maximize interoperability with current and historical TIFF-F implementations. The TIFF-F structure for writing minimum subset files is also defined.

3.6.1 Summary of Minimum Subset Fields and Values

A summary of the minimum subset TIFF-F fields and values is provided in the following table. The required fields for the minimum subset are shown under the column labeled "Field". The values for these fields in the minimum subset are shown under the column labeled "Minimum".

Field	Minimum	Comment
BitsPerSample	1	one bit per sample
Compression	3	3 for T.4 (MH)
FillOrder	2	LSB first
ImageWidth	1728	
ImageLength		required
NewSubFileType	Bit 1 = 1	single page of multipage file
PageNumber	X/X	pg/tot, 0 base, tot in 1st IFD
PhotometricInterp	0	0 is white
ResolutionUnit	2	inches (default)
RowsPerStrip	=ImageLength	
SamplesPerPixel	1	one sample per pixel

StripByteCounts		required
StripOffsets		required
T4Options	Bit 0 = 0	MH
	Bit 1 = 0	
	Bit 2 = 0,1	Non-Byte-aligned, Byte-Aligned EOLs
XResolution	204	Units is per inch
YResolution	196,98	Units is per inch

3.6.2 TIFF-F Minimum Subset File Structure

For implementations which need to write minimum subset TIFF-F files, the file structure shown in Figure 3.1 MUST be used:

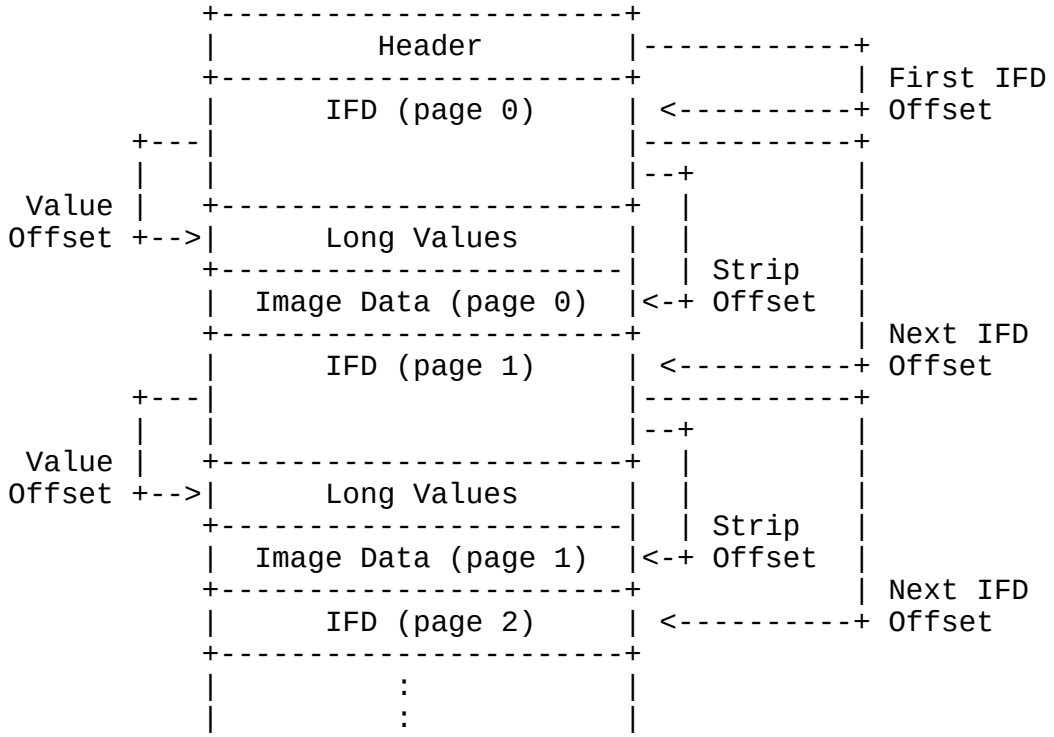

Figure 3.1 TIFF-F Minimum Subset File Structure

As depicted in Figure 3.1, the IFD of each page precedes the related Image Data for that page. If present, any long field values appear between the IFD and the image data for that page. For multiple page documents, each IFD/image pair is immediately followed by the next IFD/image pair in logical page order within the file structure, until all pages have been defined.

The format for the TIFF Header is as defined in [TIFF]. When writing TIFF-F minimum subset files, the value for the byte order in the Header SHOULD be II (0x4949, denoting that the bytes in the TIFF file are in LSB first (little-endian) order.

This results in a TIFF header whose content is as shown in Figure 3.2.

Offset	Description	Type	Value
0	Byte Order	Short	0x4949 (II)
2	Version	Short	42
4	Offset of 0th IFD	Long	0x 0000 0008

Figure 3.2: Image File Header for Minimum Subset TIFF-F Files

3.7 Technical Implementation Issues

3.7.1 Strips

Those new to TIFF may not be familiar with the concept of "strips" embodied in the three fields RowsPerStrip, StripByteCount, StripOffsets.

In general, third-party implementations that read and write TIFF files expect the image to be divided into "strips," also known as "bands." Each strip contains a few lines of the image. By using strips, a TIFF reader need not load the entire image into memory, thus enabling it to fetch and decompress small random portions of the image as necessary.

The dimensions of a strip are described by the RowsPerStrip and StripByteCount fields. The location in the TIFF file of each strip is contained in the StripOffsets field.

The size of TIFF-F strips is application dependent. The recommended approach for multi-page TIFF-F images is to represent each page as a single strip.

3.7.2 Bit Order

The default bit order in Baseline TIFF per [TIFF] is indicated by FillOrder=1, where bits are not reversed before being stored. However, TIFF-F typically utilizes the setting of FillOrder=2, where the bit order within bytes is reversed before storage (i.e., bits are stored with the Least Significant Bit first).

Facsimile data appears on the phone line in bit-reversed order relative to its description in CCITT Recommendation T.4. Therefore, a wide majority of facsimile implementations choose this natural order for storage. Nevertheless, TIFF-F readers must be able to read data in both bit orders.

3.7.3 Multi-Page

Many existing implementations already read TIFF-F like files, but do not support the multi- page field. Since a multi-page format greatly simplifies file management in fax application software, TIFF-F specifies multi-page documents (NewSubfileType = 2) as the standard case.

3.7.4 Compression

In Group 3 facsimile, there are three compression methods which had been standardized as of 1994 and are in common use. The ITU-T T.4 recommendation defines a one-dimensional compression method known as Modified Huffman (MH) and a two-dimensional method known as Modified READ (MR) (READ is short for Relative Element Address Designate). In 1984, a somewhat more efficient compression method known as Modified Modified READ (MMR) was defined in the T.6 recommendation. It was originally defined for use with Group 4 facsimile, so that this compression method has been commonly called Group 4 compression. In 1991, the MMR method was approved for use in Group 3 facsimile and has since been widely utilized.

TIFF-F permits three different compression methods. In the most common practice, the one-dimensional compression method (Modified Huffman) is used. This is specified by setting the value of the Compression field to 3 and then setting bit 0 of the T4Options field to 0. Alternatively, the two dimensional Modified READ method (which is much less frequently used in historical TIFF-F implementations) may be selected by setting bit 0 to a value of 1.

Optionally, depending upon the implementation requirements, the more efficient two-dimensional compression method from T.6 (i.e. MMR or "Group 4 compression") may be selected. This method is selected by

setting the value of the Compression field to 4 and then setting the value of the first two bits (and all unused bits) of T6options to 0. More information to aid the implementer in making a compression selection is contained in section 3.8 on Implementation Warnings.

3.7.5 Example Use of Page-quality Fields

Here are examples for writing the CleanFaxData, BadFaxLines, and ConsecutiveBadFaxLines fields:

1. Facsimile hardware does not provide page quality information: MUST NOT write page-quality fields.
2. Facsimile hardware provides page quality information, but reports no bad lines. Write only BadFaxLines = 0.
3. Facsimile hardware provides page quality information, and reports bad lines. Write both BadFaxLines and ConsecutiveBadFaxLines. Also write CleanFaxData = 1 or 2 if the hardware's regeneration capability is known.
4. Source image data stream is error-corrected or otherwise guaranteed to be error-free such as for a computer generated file: SHOULD NOT write page-quality fields.

3.7.6 Use of TIFF-F for Streaming Applications

TIFF-F has historically been used for handling fax image files in implementations such as store and forward messaging where the entire size of the file is known in advance. While TIFF-F may also possibly be used as a file format for cases such as streaming applications, different assumptions may be required than those provided in this document (e.g., the entire size and number of pages within the image are not known in advance). As a result, a definition for the streaming application of TIFF-F is beyond the scope of this document.

3.7.7 TIFF-F Export and Import

Fax implementations that do not wish to support TIFF-F as a native format may elect to support it as import/export medium.

Export

It is recommended that implementations export multiple page TIFF-F files without manipulating fields and values. Historically, some TIFF-F writers have attempted to produce individual single-page TIFF-F files with modified NewSubFileType and PageNumber (page one-of-one) values for export purposes. However, there is no easy way to link such multiple single page files together into a logical multiple page document, so that this practice is not recommended.

Import

A TIFF-F reader MUST be able to handle a TIFF-F file containing multiple pages.

3.8 Implementation Warnings

3.8.1 Uncompressed data

TIFF-F requires the ability to read and write at least one-dimensional T.4 Huffman ("compressed") data. Uncompressed data is not allowed. This means that the "Uncompressed" bit in T4Options or T6Options must be set to 0.

3.8.2 Encoding and Resolution

Since two-dimensional encoding is not required for Group 3 compatibility, some historic TIFF-F readers have not been able to read such files. The minimum subset of TIFF-F REQUIRES support for one dimensional (Modified Huffman) files, so this choice maximizes portability. However, implementers seeking greater efficiency SHOULD use T.6 MMR compression when writing TIFF-F files. Some TIFF-F readers will also support two-dimensional Modified READ files. Implementers that wish to have the maximum flexibility in reading TIFF-F files SHOULD support all three of these compression methods (MH, MR and MMR).

For the case of resolution, almost all facsimile products support both standard (98 dpi) vertical resolution and "fine" (196 dpi) resolution. Therefore, fine-resolution files are quite portable in the real world.

In 1993, the ITU-T added support for higher resolutions in the T.30 recommendation including 200 x 200, 300 x 300, 400 x 400 in dots per inch based units. At the same time, support was added for metric dimensions which are equivalent to the following inch based resolutions: 391v x 204h and 391v x 408h. Therefore, the full set of inch-based equivalents of the new resolutions are supported in the TIFF-F writer, since they may appear in some image data streams received from Group 3 facsimile devices. However, many facsimile terminals and older versions of TIFF-F readers are likely to not support the use of these higher resolutions.

Per [T.4], it is permissible for implementations to treat the following XResolution values as being equivalent: <204,200> and <400,408>. In a similar respect, the following YResolution values

may also be treated as being equivalent: <98, 100>, <196, 200>, and <391, 400>. These equivalencies were allowed by [T.4] to permit conversions between inch and metric based facsimile terminals.

In a similar respect, the optional support of metric based resolutions in the TIFF-F reader (i.e. 77 x 38.5 cm) is included for completeness, since they are used in some legacy TIFF-F implementations, but this use is not recommended for the creation of TIFF-F files by a writer.

3.8.3 EOL byte-aligned

The historical convention for TIFF-F has been that all EOLs in Modified Huffman or Modified READ data must be byte-aligned. However, Baseline TIFF has permitted use of non-byte-aligned EOLs by default, so that a large percentage of TIFF-F reader implementations support both conventions. Therefore, the minimum subset of TIFF-F as defined in this document includes support for both byte-aligned and non-byte-aligned EOLs.

An EOL is said to be byte-aligned when Fill bits have been added as necessary before EOL codes such that EOL always ends on a byte boundary, thus ensuring an EOL-sequence of a one byte preceded by a zero nibble: xxxx0000 00000001.

Modified Huffman encoding encodes bits, not bytes. This means that the end-of-line token may end in the middle of a byte. In byte alignment, extra zero bits (Fill) are added so that the first bit of data following an EOL begins on a byte boundary. In effect, byte alignment relieves application software of the burden of bit-shifting every byte while parsing scan lines for line-oriented image manipulation (such as writing a TIFF file).

For Modified READ encoding, each line is terminated by an EOL and a one bit tag bit. Per [T.4], the value of the tag bit is 0 if the next line contains two dimensional data and 1 if the next line is a reference line. To maintain byte alignment, fill bits are added before the EOL/tag bit sequence, so that the first bit of data following an MR tag bit begins on a byte boundary.

3.8.4 EOL

As illustrated in FIGURE 1/T.4 in [T.4], facsimile documents encoded with Modified Huffman begin with an EOL (which in TIFF-F may be byte-aligned). The last line of the image is not terminated by an EOL. In a similar respect, images encoded with Modified READ two dimensional encoding begin with an EOL, followed by a tag bit.

3.8.5 RTC Exclusion

Aside from EOLs, TIFF-F files have historically only contained image data. This means that implementations which wish to maintain strict conformance with the rules in [TIFF] and compatibility with historical TIFF-F, SHOULD NOT include the Return To Control sequence (RTC) (consisting of 6 consecutive EOLs) when writing TIFF- F files. However, implementations which need to support "transparency" of [T.4] image data MAY include RTCs when writing TIFF-F files if the flag settings of the T4Options field are set for non-byte aligned MH or MR image data. Implementors of TIFF readers should also be aware that there are some existing TIFF-F implementations which include the RTC sequence in MH/MR image data. Therefore, TIFF-F readers MUST be able to process files which do not include RTCs and SHOULD be able to process files which do include RTCs.

3.8.6 Use of EOFB for T.6 Compressed Images

TIFF-F pages which are encoded with the T.6 Modified Modified READ compression method MUST include an "end-of-facsimile-block" (EOFB) code at the end of each coded strip. Per [TIFF], the EOFB code is followed by pad bits as needed to align on a byte boundary. TIFF readers SHOULD ignore any bits other than pad bits beyond the EOFB.

3.9 TIFF-F Fields Summary

Implementations may choose to implement a TIFF-F Reader, TIFF-F Writer or both, depending upon application requirements. The TIFF- F Reader is typically used to read an existing TIFF-F file which resides on a computer or peripheral device. The TIFF-F Writer is typically used to convert a bi-level image bit stream into a TIFF-F compliant file. For many Internet applications, only the Reader needs to be implemented. The specific field support required for TIFF-F Readers and Writers is summarized below.

3.9.1 TIFF Reader

The fields in the following table are specified for a TIFF-F Reader. The range of values for required and recommended fields are as shown. The minimum subset of values are also shown. If required fields are omitted in a TIFF-F file, the Baseline TIFF default value will apply. Image data must not have any coding errors. In the table, certain fields have a value that is a sequence of flag bits (e.g. T4Options). An implementation should test the setting of the relevant flag bits individually to allow extensions to the sequence of flag bits to be appropriately ignored.

As noted within [TIFF], a TIFF file begins with an 8-byte image file header, of which the first two bytes (0-1) contain the byte order within the file. The permissible values are:

II- Byte order from least significant byte to the most significant byte (little-endian)
MM - byte order is always from most significant to least significant (big-endian)

For a TIFF-F Reader, the legal values are:

ByteOrder: MM,II (Either byte order is allowed)

3.9.1.1 Fields for TIFF-F Reader

Recommended Fields in the table are shown with an asterisk (*).

Other fields may be present, but they should be of an informational nature, so that a reader can elect to ignore them.

Informational fields which are often present in TIFF-F images are: Software, Datetime, BadFaxLines, CleanFaxData and ConsecutiveBadFaxLines.

Field	Values	Minimum	Comment
BitsPerSample	1	1	one bit per sample
Compression	3,4	3	3 for T.4 (MH, MR)
			4 for T.6 - MMR
FillOrder	2,1	2	LSB first or MSB first
ImageWidth	1728, 2048, 2432, 2592, 3072, 3648, 3456, 4096, 4864	1728	depends on XResolution
ImageLength	>0		required
NewSubFileType	Bit 1 = 1	Bit 1 = 1	single page of multipage file
Orientation *	1		1st row=top left, 1st col=top
PageNumber	X/X	0/1	pg/tot, 0 base, tot in 1st IFD
PhotometricInterp	0,1	0	0 is white
ResolutionUnit	2,3	2	inches (default)
RowsPerStrip	=ImageLength or other	=ImageLength	
SamplesPerPixel	1	1	one sample per pixel
StripByteCounts	>0		required

StripOffsets	>0		required
T4Options	Bit 0 = 0,1 Bit 1 = 0 Bit 2 = 0,1	Bit 0 = 0 Bit 1 = 0 Bit 2 = 0,1	MH,MR(incl if not MMR) Non-Byte-aligned and Byte-Aligned EOLs
T6Options	0		MMR (incl only if MMR)
XResolution	204,200,300, 400,408, 77	204	If unit is per inch If unit is per cm
YResolution	196,98,100, 200,300,391, 400, 77,38.5	196,98	If unit is per inch If unit is per cm

3.9.2 TIFF-F Writer

For the case of writing (creating) a TIFF-F file format from an image data stream or other raster data, implementations SHOULD write files which can be read by a TIFF-F Reader as defined in 3.9.1. It is recommended that all fields from the table in 3.9.1.1 SHOULD be included when writing TIFF-F files in order to minimize dependencies on default values. Image data must not have any coding errors.

Other fields may be present, but they should be of an informational nature, so that a Reader may elect to ignore them.

For the case of writing "minimum subset" TIFF-F files, the rules defined in section 3.6 apply.

Informational fields that may be useful for TIFF-F files are:
Software, Datetime, BadFaxLines, ConsecutiveBadFaxLines

TIFF Writers SHOULD only generate the fields that describe facsimile image quality when the image has been generated from a fax image data stream where error correction (e.g. Group 3 Error Correction Mode) was not used. These fields are: CleanFaxData, BadFaxLines and ConsecutiveBadFaxLines.

4. MIME sub-type image/tiff

[TIFFREG] describes the registration of the MIME content-type image/tiff to refer to TIFF 6.0 encoded image data. When transported by MIME, the TIFF content defined by this document must be encoded within an image/tiff content type. In addition, an optional "application" parameter is defined for image/tiff to identify a particular application’s subset of TIFF and TIFF extensions for the

encoded image data, if it is known. Typically, this would be used to assist the recipient in dispatching a suitable rendering package to handle the display or processing of the image file.

4.1 Refinement of MIME sub-type image/tiff for Application F

Since this document defines a facsimile specific profile of TIFF, it is useful to note an appropriate application parameter for the image/tiff MIME content-type.

The "faxbw" application parameter is defined for black and white facsimile. It is suitable for use by applications that can process one or more TIFF for facsimile profiles or subsets used for the encoding of black and white facsimile data.

Since this document defines a profile of TIFF for facsimile which is suitable for use with black and white facsimile image data, applications which use this profile or its minimum subset should set the value of the application parameter to "faxbw".

An example of the use of the image/tiff MIME Content-type with the application parameter set with the value "faxbw" follows:

Example:

```
Content-type: image/tiff; application=faxbw
```

In this example, use of this parameter value will enable applications to identify the content as being within a profile or subset of TIFF for Facsimile that is suitable for encoding black and white image data, before attempting to process the image data.

5. Implementation Usage

5.1 Internet Fax Usage

The usage of TIFF-F is envisioned as a component of Internet Fax. It is anticipated that Internet Fax may use both a TIFF-F Reader and TIFF-F Writer. The details of the Internet Fax services and their use of TIFF-F will be specified in other documents.

5.2 VPIM Usage

The Application F of TIFF (i.e. TIFF-F content) is a secondary component of the VPIM Message as defined in [VPIM2]. Voice messaging systems can often handle fax store-and-forward capabilities in addition to traditional voice message store-and- forward functions.

As a result, TIFF-F fax messages can optionally be sent between compliant VPIM systems, and may be rejected if the recipient system cannot deal with fax.

Refer to the VPIM Specification for proper usage of this content.

6. Security Considerations

This document describes the encoding for TIFF-F, which is a profile of the TIFF encoding for facsimile. As such, it does not create any security issues not already identified in [TIFFREG], in its use of fields as defined in [TIFF]. There are also new TIFF fields defined within this specification, but they are of a purely descriptive nature, so that no new security risks are incurred.

Further, the encoding specified in this document does not in any way preclude the use of any Internet security protocol to encrypt, authenticate, or non-repudiate TIFF-F encoded facsimile messages.

7. Authors' Addresses

Glenn W. Parsons
Northern Telecom
P.O. Box 3511, Station C
Ottawa, ON K1Y 4H7
Canada
Phone: +1-613-763-7582
Fax: +1-613-763-2697
Email: Glenn.Parsons@Nortel.ca

James Rafferty
Human Communications
12 Kevin Drive
Danbury, CT 06811-2901
USA
Phone: +1-203-746-4367
Fax: +1-203-746-4367
Email: Jrafferty@worldnet.att.net

8. References

[MIME1] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[MIME4] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", RFC 2048, November 1996.

[REQ] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[T.30] ITU-T Recommendation T.30 - "Procedures for Document Facsimile Transmission in the General Switched Telephone Network", June, 1996

[T.4] ITU-T Recommendation T.4 - "Standardization of Group 3 Facsimile Apparatus for Document Transmission", June, 1996

[T.6] ITU-T Recommendation T.6 - "Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus", March, 1993

[TIFF] Adobe Developers Association, TIFF (TM) Revision 6.0 - Final, June 3, 1992.

[TIFFREG] Parsons, G., Rafferty, J. and S. Zilles, "Tag Image File Format (TIFF) - image/tiff: MIME Sub-type Registration ", RFC 2302, March 1998.

[VPIM2] G. Vaudreuil and G. Parsons, "Voice Profile for Internet Mail - version 2", Work In Progress, <draft-ema-vpim-06.txt>, November 1997.

9. Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.