Tag Image File Format (TIFF) - Class F

Status of this Memo

This document is an Internet Draft. Internet Drafts are working
documents of the Internet Engineering Task Force (IETF), its Areas,
and its Working Groups. Note that other groups may also distribute
working documents as Internet Drafts.

Internet Drafts are valid for a maximum of six months and may be
updated, replaced, or obsoleted by other documents at any time. It
is inappropriate to use Internet Drafts as reference material or to
cite them other than as a "work in progress".

To learn the current status of any Internet-Draft, please check the
"1id-abstracts.txt" listing contained in the Internet-Drafts Shadow
Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe),
munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or
ftp.isi.edu (US West Coast).

Overview

This document describes in detail the definition of TIFF Class F
that is used to store facsimile images and also describes the
registration refinement of the MIME sub-type image/tiff. The Class
F tag encoding has been folklore with no specific 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 formalizes the reference for the Tag Image File
Format (TIFF) and formally defines TIFF Class F that is used to
store facsimile images. As well, the original registration for
image/TIFF from RFC 1528 is refined by this document.

2. TIFF Definition

TIFF (Tag Image File Format) Revision 6.0 is defined in detail by
Adobe in [TIFF]. The documentation can be obtained from Adobe at:

Adobe Developers Association
Adobe Systems Incorporated
1585 Charleston Road
Only an introduction to the baseline TIFF is included in this document. The reader is directed to the original TIFF specification [TIFF] to obtain specific technical details.

2.1 TIFF Scope

TIFF describes image data that typically comes from scanners, frame grabbers, and paint- and photo-retouching programs. TIFF is not a printer language or page description language. The purpose of TIFF is to describe and store raster image data. A primary goal of TIFF is to provide a rich environment within which applications can exchange image data. This richness is required to take advantage of the varying capabilities of scanners and other imaging devices. Though TIFF is a rich format, it can easily be used for simple scanners and applications as well because the number of required fields is small.

2.2 TIFF Features

- TIFF is capable of describing bilevel, grayscale, palette-color, and full-color image data in several color spaces.

- TIFF includes a number of compression schemes that allow developers to choose the best space or time tradeoff for their applications.

- TIFF is not tied to specific scanners, printers, or computer display hardware.

- TIFF is portable. It does not favor particular operating systems, file systems, compilers, or processors.
- TIFF is designed to be extensible—to evolve gracefully as new needs arise.

- TIFF allows the inclusion of an unlimited amount of private or special-purpose information.

3. TIFF Class F Definition

Though it has been in common usage for many years, TIFF Class F (or 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 serves as the formal definition for TIFF Class F for Internet applications.

3.1 The Spirit of TIFF Class F

TIFF Classes reduce the information burden on TIFF readers and writers that wish to support narrow applications. For example, Appendix G-1 of TIFF 5.0 stated that classes enable TIFF readers "to know when they can stop adding TIFF features." In other words, defining a Class enables applications interested only in reading that Class to give up if the characteristic tags and values are not present. Therefore, TIFF Class F insists on a rather narrow definition of tags. In a general TIFF file, for example, the writer would be free to create single-page documents without the NewSubFiletype and PageNumber tags. Not so for a Class F file, where the multi-page tag is required even for a single page.

TIFF Class F is a sub-class of Class B (Bilevel) [TIFFB]. That is, all tags that are required in Class B are also required in Class F. For some common tags, however, Class F limits the range of acceptable values. The YResolution tag, for example, is a Class B tag, but within Class F only a limited set of values is permitted. Such tags are listed in section 3.2

Section 3.3 discusses other Class B tags that have a specific meaning when applied to facsimile images. Several new Class F tags are discussed in section 3.4. There are also tags that may be helpful but are not required. These recommended tags are listed in the section 3.5.

Several technical topics, including implementation issues, warnings and conformance are discussed in subsequent sections. Finally, section 3.9 introduces the TIFF-F Reader and Writer. A table of the required TIFF-F tags for each of these implementations is summarized.
3.2 Class F Required Tags

FillOrder = 1, 2. SHORT.
TIFF Class 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.

ImageWidth = 1728, 2048, 2482, 2592, 3072, 3723, 3456, 4096, 4964. SHORT or LONG. These are the fixed page widths in pixels defined in CCITT Group 3.

NewSubFileType = 2. LONG.
The value 2 identifies a single page of a multi-page image.

PageNumber. SHORT/SHORT.
This tag specifies the page numbers in the fax document. The tag comprises two SHORT values: the first value is the page number, the second is the total number of pages. Single-page documents therefore use 00000001 hex.

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

XResolution = 204, 200, 300, 400, 406 (inches). RATIONAL.
The horizontal resolution of the image expressed in pixels per resolution unit. Some existing TIFF-F implementations may also support values of 77 (cm).

YResolution = 98, 196, 100, 200, 300, 392, 400 (inches). RATIONAL.
The vertical resolution of the image expressed in pixels per resolution unit. Some existing TIFF-F implementations may also support values of 77, 38.5 (cm).

3.2.1 Class F Tags for Compression Selection

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 Addressing). 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 Class F permits two different compression methods. By default, the one-dimensional compression method is used. Optionally, depending upon the application, the more efficient two-dimensional compression method from T.6 (i.e. MMR or "Group 4 compression") may be selected. More information to aid the implementor in making this selection is contained in section 3.8 on Implementation Warnings.

The tags used to specify the compression are shown below:

**Compression = 3,4. SHORT.**

This is a required Class F tag. Modified Huffman, one-dimensional encoding with "byte-aligned" EOLs is selected by setting 3. 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 1 byte preceded by a zero nibble: xxxx0000 00000001. The data in a Class F image is not terminated with an RTC (see sections 3.8.4 and 3.8.5). For two-dimensional encoding, set the value of the compression tag to 4 (see section 3.8.2).

**T4Options = 4,5. LONG.**

This tag is required for TIFF Class-F if the one-dimensional compression method has been specified (i.e. by setting the value of the Compression tag to 3). Data is one-dimensional and EOLs must be byte-aligned. Uncompressed data is not allowed.

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

**T6Options = 0 LONG.**

This tag is required for TIFF Class F if the two-dimensional compression method is specified (i.e. by setting the value of the Compression tag to 4). Data is two-dimensional and there is no use of EOLs. Uncompressed data is not allowed. In earlier versions of TIFF, this tag was named Group4Options. The significance has not changed and the present definition is compatible. 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. The default value is 0 (all bits 0).

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

3.3 Class B (Bilevel) Required Tags

Although these tags are already required in Class B (Bi-Level) files, an explanation of their usage for facsimile images may be helpful.

**BitsPerSample = 1. SHORT.**

Since facsimile is a black-and-white medium, this must be 1 (the default) for all files.
ImageLength. SHORT or LONG. LONG recommended.
The total number of scan lines in the image.

PhotometricInterp = 0,1. SHORT.
This tag allows notation of an inverted ("negative") image:
0 = normal
1 = inverted

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

RowsPerStrip. 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 tag.

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

StripByteCounts. SHORT or LONG. SHORT 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.

StripOffsets. 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.

3.4 New Class F Tags

There are only three new tags for Class F. All three tags describe
page quality. The information contained in these tags is usually
obtained from the receiving facsimile hardware, but since not all
devices are capable of reporting this information, the tags are
optional.

Some applications 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
Tag = 326 (146 hex)
Type = SHORT or LONG
This tag 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
Tag = 327 (147 hex)
Type = SHORT
N = 0
  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 existed, but were not regenerated by receiving device

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 tag describes the error content of the data. That is, when the BadFaxLines and ImageLength tags indicate that the facsimile device encountered lines with an incorrect number of pixels during reception, the CleanFaxData tag indicates whether these lines are actually in the data or if the receiving facsimile device replaced them with regenerated lines.

ConsecutiveBadFaxLines
Tag = 328 (148 hex)
Type = LONG or SHORT

This tag 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 tag is an indicator of their distribution and may therefore be a better general indicator of perceived image quality.

3.5 Recommended Tags

BadFaxLines. LONG.
The number of "bad" scan lines encountered by the facsimile during reception.
CleanFaxData = 0, 1, 2. BYTE.
This tag indicates whether lines with incorrect pixel count are actually in the data or if the receiving facsimile device replaced them with regenerated lines.
- 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 existed, but were not regenerated by receiving device

ConsecutiveBadFaxLines. LONG or SHORT.
The maximum number of consecutive scan lines with incorrect pixel count encountered by the facsimile device reception.

DateTime. 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. ASCII.
This is the name of the document from which the document was scanned.

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

Orientation. SHORT.
This tag 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.

3.6 Technical Implementation Issues

3.6.1 Strips

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

In general, third-party applications 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 tags. The location in the TIFF file of each strip is contained in the StripOffsets tag.
The TIFF documentation suggests using strips of an arbitrary size of about 8K. Although various application programs assert that they "prefer" banded images, research failed to uncover a single existing application that could not read a single-strip page where they could read the same file in a multi-strip format. Indeed, applications seem to be more sensitive to the total size of the decoded image and are not particularly fussy about banding. This result is not surprising, considering that most desktop publishing programs are prepared to deal with massively larger images than those one finds in facsimile. In short, each page may be represented as a single strip of any length.

In fact, there may be a compelling reason to employ a strip size equal to the length of one A4 page (297 mm). When a document is imaged, it may be of any length. Not all fax machines, however, can accept unlimited length documents. Worse, the remote machine’s page-length capability is not known until the fax connection has been established. The solution is for the transmitting fax device to image long documents into A4-size strips, then seam them together at transmission, after the capabilities of the remote fax machine is known.

3.6.2 Bit Order

Although the TIFF 6.0 documentation lists the FillOrder tag in the category "No Longer Recommended," Class F resurrects it.

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 applications choose this natural order for storage. Nevertheless, TIFF Class F readers must be able to read data in both bit orders.

3.6.3. Multi-Page

Many existing applications already read Class F-like files, but do not support the multi-page tag. Since a multi-page format greatly simplifies file management in fax application software, Class F specifies multi-page documents (NewSubfileType = 2).

3.6.4. Two-dimensional Encoding

PC Fax applications that wish to support two-dimensional encoding may do so by setting Bit 0 in the Group3Options tag. Please see section 3.8.2.

3.6.5. Example Use of Page-quality Tags

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

1. Facsimile hardware does not provide page quality information: write no tags.
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. Computer generated file: write CleanFaxData = 0.

3.7 TIFF Class F Conformance

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

Export
The simplest form of support is a Class F writer that produces individual single-page Class F files with the proper NewSubFile tag and the PageNumber (page one-of-one) tag.

Import
A Class F reader must be able to handle a Class F file containing multiple pages.

3.8 Implementation Warnings

3.8.1 Uncompressed data

Class 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, Class F readers may decline to read such files. Therefore, for maximum portability, applications may choose to write only one-dimensional files. Although the same argument technically holds for "fine" (196 dpi) vertical resolution, only a tiny fraction of facsimile products support only 98 dpi. 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: 392v x 203h and 392v x 406h. Therefore, the inch-based equivalents of the new resolutions are supported in the TIFF 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. In a similar respect, the optional support of metric based resolutions in the TIFF-F reader (i.e. 77 x 38.5 cm) is
3.8.3 EOL byte-aligned

In the spirit of TIFF, all EOLs in data must be byte-aligned. 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.

Recall that 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).

3.8.4 EOL

As illustrated in FIGURE 1/T.4 in [T.4], facsimile documents begin with an EOL (which in Class F is byte-aligned). The last line of the image is not terminated by an EOL.

3.8.5 RTC Exclusion

Aside from EOLs, TIFF Class F files contain only image data. This means that the Return To Control sequence (RTC) is specifically prohibited. Exclusion of RTCs not only makes possible the simple concatenation of images, it eliminates the mischief—failed communications and unreadable images—that their mistreatment inevitably produces.

3.9 TIFF-F Tags 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. Despite the semantic difference between the Reader and the Writer, it is often sufficient to implement only the Reader for most Internet applications. The specific tag support required for each of these variations is summarized below.

3.9.1 TIFF Reader

The tags in following table are specified for a TIFF Reader. Legal values are as shown. If required tags are omitted, the default value will apply. Image data must not have any coding errors.
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 Tags for TIFF-F Reader

<table>
<thead>
<tr>
<th>Tag</th>
<th>Legal</th>
<th>Default</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitsPerSample</td>
<td>1</td>
<td>1</td>
<td>one bit per sample</td>
</tr>
<tr>
<td>CleanFaxData</td>
<td>0</td>
<td>0</td>
<td>data has no errors</td>
</tr>
<tr>
<td>Compression</td>
<td>3, 4</td>
<td>3</td>
<td>T.4 bi-level encoding, MH or T.6, MMR</td>
</tr>
<tr>
<td>FillOrder</td>
<td>2, 1</td>
<td>2</td>
<td>LSB first or MSB first</td>
</tr>
<tr>
<td>ImageWidth</td>
<td>1728, 2048, 2482, 2592, 3072, 3723, 3456, 4096, 4964</td>
<td>depends on XResolution</td>
<td></td>
</tr>
<tr>
<td>ImageLength</td>
<td>&gt;0</td>
<td></td>
<td>required</td>
</tr>
<tr>
<td>NewSubFileType</td>
<td>2</td>
<td>2</td>
<td>single page of multipage file</td>
</tr>
<tr>
<td>Orientation</td>
<td>1</td>
<td>1</td>
<td>1st row=top left, 1st col=top</td>
</tr>
<tr>
<td>PageNumber</td>
<td>X/X</td>
<td>0/1</td>
<td>pg/tot, 0 base, tot in 1st IFD</td>
</tr>
<tr>
<td>PhotometricInterp</td>
<td>0</td>
<td>0</td>
<td>0 is white</td>
</tr>
<tr>
<td>ResolutionUnit</td>
<td>2, 3</td>
<td>2</td>
<td>inches (default)</td>
</tr>
<tr>
<td>RowsPerStrip</td>
<td>=ImageLength</td>
<td>=ImageLength</td>
<td>one sample per pixel</td>
</tr>
<tr>
<td>SamplesPerPixel</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>StripByteCounts</td>
<td>&gt;0</td>
<td></td>
<td>required</td>
</tr>
<tr>
<td>StripOffsets</td>
<td>&gt;0</td>
<td></td>
<td>required</td>
</tr>
<tr>
<td>T4Options</td>
<td>4</td>
<td>4</td>
<td>MH (incl if not MMR)</td>
</tr>
<tr>
<td>T6Options</td>
<td>0</td>
<td>0</td>
<td>MMR (incl only if MMR)</td>
</tr>
<tr>
<td>Xresolution</td>
<td>204, 200, 77,</td>
<td>204</td>
<td></td>
</tr>
<tr>
<td></td>
<td>300, 400, 406</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yresolution</td>
<td>196, 98, 100,</td>
<td>196</td>
<td></td>
</tr>
<tr>
<td></td>
<td>200, 77, 38.5,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>300, 392, 400</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Other tags may be present, but must be of the sort that can be ignored safely by implementations (i.e. purely informational).

Recommended informational tags are:
Software, Datetime, BadFaxLines, ConsecutiveBadFaxLines

Parsons, Rafferty
Expires 7/31/97
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 use the TIFF-F tags in the following table as a default. The use of default tags and values for the TIFF-F writer is intended to encourage consistent use of TIFF-F in Internet applications. Image data must not have any coding errors.

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.

For a TIFF-F Writer, the legal value is:

ByteOrder: II (least significant byte to the most significant byte)

### 3.9.2.1 TIFF-F Writer Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Legal Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitsPerSample</td>
<td>1</td>
<td>one bit per sample</td>
</tr>
<tr>
<td>Compression</td>
<td>3</td>
<td>T.4 bi-level encoding, MH</td>
</tr>
<tr>
<td>FillOrder</td>
<td>2</td>
<td>LSB first</td>
</tr>
<tr>
<td>ImageWidth</td>
<td>1728, 2048, 2482, 2592, 3072, 3723, 3456, 4096, 4964</td>
<td>depends on XResolution</td>
</tr>
<tr>
<td>ImageLength</td>
<td>&gt; 0</td>
<td></td>
</tr>
<tr>
<td>NewSubFileType</td>
<td>2</td>
<td>single page of multi-page file</td>
</tr>
<tr>
<td>PageNumber</td>
<td>X/X</td>
<td>pg/tot, 0 base, tot in 1st IFD</td>
</tr>
<tr>
<td>PhotometricInterp</td>
<td>0</td>
<td>0 is white</td>
</tr>
<tr>
<td>ResolutionUnit</td>
<td>2</td>
<td>inches</td>
</tr>
<tr>
<td>RowsPerStrip</td>
<td>&gt;0</td>
<td>must be same as ImageLength</td>
</tr>
<tr>
<td>SamplesPerPixel</td>
<td>1</td>
<td>one sample per pixel</td>
</tr>
<tr>
<td>StripByteCounts</td>
<td>&gt;0</td>
<td>as appropriate</td>
</tr>
<tr>
<td>StripOffsets</td>
<td>&gt;0</td>
<td>as appropriate</td>
</tr>
<tr>
<td>T4Options</td>
<td>4</td>
<td>MH, byte aligned EOL</td>
</tr>
<tr>
<td>Xresolution</td>
<td>204,200, 300,400,406</td>
<td></td>
</tr>
<tr>
<td>Yresolution</td>
<td>196,98,100, 200,300, 392,400</td>
<td></td>
</tr>
</tbody>
</table>

For some applications, it may be preferable to use T.6 compression in place of one-dimensional T.4 compression. For this case, the defaults are revised as specified in section 3.9.2.3.
3.9.2.2 Optional TIFF-F Writer Tags

The following tags are optional. If they are present, they must contain the values as shown:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Legal Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CleanFaxData</td>
<td>0</td>
<td>data doesn’t contain bad scan lines</td>
</tr>
<tr>
<td>Orientation</td>
<td>1</td>
<td>1st row = top left, 1st col = top</td>
</tr>
</tbody>
</table>

Recommended informational tags are:
Software, Datetime, BadFaxLines, ConsecutiveBadFaxLines

3.9.2.3 TIFF-F Writer Tags for T.6 Compression Option

For some applications, a TIFF-F writer may choose to use the T.6 compression option in place of the one-dimensional Modified Huffman standardized in [T.4]. In this case, the rules for the TIFF-F Writer tags and values apply, but the except as specified below:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Legal Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression</td>
<td>4</td>
<td>ITU-T T.6 encoding, MMR</td>
</tr>
<tr>
<td>T6Options</td>
<td>0</td>
<td>Replaces T4Options Tag</td>
</tr>
</tbody>
</table>

Other tags may be present, but must be of the sort that can be ignored safely by applications (i.e. purely for information).

4. MIME sub-type image/TIFF

The image/TIFF sub-type was originally defined in RFC 1528 as containing TIFF 5.0 encoded image data.

This document, in section 6, re-defines the original image/TIFF definition to refer to TIFF 6.0 encoded image data. The TIFF 6.0 specification is a cleaner document and is compatible with previous TIFF 5.0 encoded image data. Further, an optional "class" parameter is defined for image/TIFF to identify the TIFF Class of the encoded image data, if it is known.
5. Implementation Usage

5.1 Internet Fax Usage

The usage of TIFF-F is envisaged to be a primary component of Internet Fax. It is anticipated that Internet Fax will make use of both a TIFF-F Reader and TIFF-F Writer. The details of these applications will be specified in other documents.

5.2 VPIM Usage

The image/TIFF sub-type with the Class F parameter (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, this sub-type is used to hold fax messages within the multipart/voice-message content that is sent between compliant VPIM systems. In this context, the fax content is optional and may be rejected if the recipient system cannot deal with fax. VPIM implementations must at least implement and support the TIFF-F Reader.

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

To: ietf-types@iana.org
Subject: Registration of Standard MIME media type image/TIFF

MIME media type name: image
MIME subtype name: TIFF
Required parameters: none
Optional parameters: class

The Classes of TIFF are denoted by letters. There are currently five valid values of class:
   B - Bilevel
   G - Grayscale
   P - Palette
   R - RGB, and
   F - Bi-Level Facsimile

There is no default value for class, as the absence of the class parameter indicates that the class of the encoded TIFF image is unknown or unnecessary to be known. The onus is on the displaying software to determine the class (if necessary) and present the image to the user.

Encoding considerations: Binary or Base-64 generally preferred

Security considerations: none

Interoperability considerations:

The ability of implementations to handle all the defined classes of TIFF may not be ubiquitous. As a result, the absence of the class parameter would force implementations to decode and attempt to display the encoded TIFF image data in order to determine if it could actually be viewed.

Published specification:

TIFF (Tag Image File Format) and most of the classes are defined in:
   TIFF (TM) Revision 6.0 - Final - June 3, 1992

Adobe Developers Association
Adobe Systems Incorporated
1585 Charleston Road
P.O. Box 7900 Mountain View, CA 94039-7900

A copy of this specification can be found in:
TIFF Class F is defined in this document in section 3

Applications which use this media type:

primarily fax and voice messaging

Additional information:

Magic number(s):
   II (little-endian):  49 49 42 00 hex
   MM (big-endian):    4D 4D 00 42 hex

File extension(s): .TIF
Macintosh File Type Code(s): TIFF

Person & email address to contact for further information:

   Glenn W. Parsons
   Glenn.Parsons@Nortel.ca

   James Rafferty
   Jrafferty@worldnet.att.net

Intended usage: COMMON

Author/Change controller:

   James Rafferty

7. Authors’ Addresses

   Glenn W. Parsons
   Nortel Technology
   P.O. Box 3511, Station C
   Ottawa, ON K1Y 4H7
   Canada
   Phone: +1-613-763-7582
   Fax:   +1-613-763-8385
   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


