Data Exchange File Format™

Version 2.5

March 25, 1994

Written by John Bono (206) 487-7497

TABLE OF CONTENTS

1.0 Overview

1.1 Introduction

This document contains the information necessary to read and write images and other information to and from ultrasound systems' disks using a "universal" file format, designed to promote the compatibility of image files between ultrasound acquisition, image review and printing products. The universal file format is referred to as the Data Exchange File Format™ (DEFF™). Digital images using DEFF are in a form appropriate for archiving, analysis, display, and printing.

The specification of Data Exchange File Format (DEFF) usage is separated into the following parts:

Disk File System including: physical media, low level and high level disk formats
Image Locator Database Specification
Base Class
Image Class
Camera Ready Class
Archive Class
Pictorial Class

The DEFF image files are grouped with each other through the use of locator databases.

The DEFF format is used to store many different types of data such as Camera Ready single frames, cineloops, etc. The various files all have the same structure and have certain parameters in common such as Date/Time. However, there are many types of data which are specific to an individual class of DEFF file.

1.1.1 Related Documents

Tag Image File Format, Revision 5.0 (Final). Aldus Corporation, 1987, 1988. Available from Aldus Developer's Desk.
(206) 628-6593 voice, (206) 343-4240 fax, or Aldus main number

1.2 Disk

DEFF currently uses only two disk types, the ISO standard 5.25" 600 MB magneto-optical and the ISO standard 3.5" 128 MB magneto-optical . The low level format is Sony mode 3 and the high level format is compatible with MS-DOS 4.0.

(MS-DOS is a trademark of Microsoft Inc.)

1.3 File Relationships

On a DEFF disk there will be one or more Locator Data Bases. These locator DBs will point to DEFF files which contain images or other data. Each DEFF file which does contain an image may optionally have a pictorial image DEFF file associated with that image.

Currently there are two locator DBs: one as a print queue for printing and one for tracking all other DEFF files.

1.4 Class Hierarchy

The DEFF specification is organized into "classes." This allows localized specification of concepts and data values. This aids both maintenance of the specification and also transference of its contents to users who need only a partial implementation of DEFF file reading and writing.

The following figures indicate the class hierarchy for both the locator database and the DEFF data files. An example in the DEFF data class structure: the Camera Ready Class is a subclass of Image Class and the Image Class is a subset of the Base DEFF Class. This indicates that a camera ready DEFF file contains all the tags in the Base, Image, and Camera Ready classes.

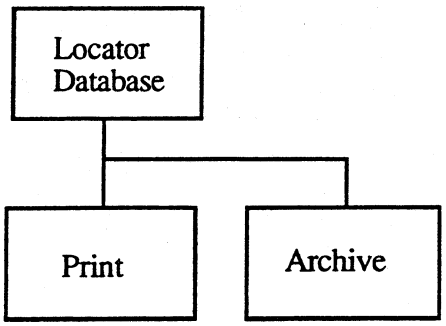


Figure 1.4-1 DEFF Locator Database Classes

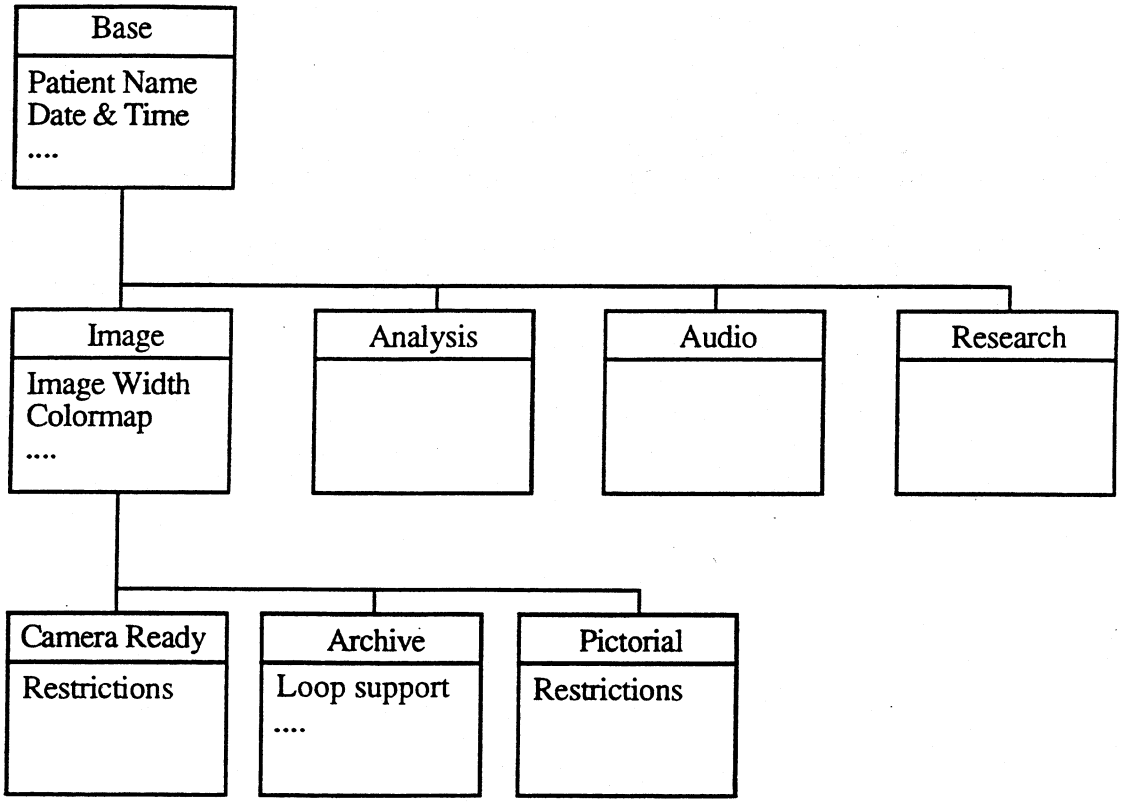


Figure 1.4-2 DEFF Data Class Structure

1.5 Class Descriptions

The Base DEFF Class specification contains information pertinent to all DEFF files. This includes, but is not limited to, the data title, description of the machine that created the DEFF file, version number, etc.

A subclass of the DEFF Base Class, the Image Class, is used for all types of images but usually requires an Image subclass to exist in the DEFF file as well. The Image Class specifies parameters which all common to all images such as image width and height, location of the pixels, and some limited description of the pixels.

A subclass to the Image Class, the Camera Ready Class, is intended to be used for printing images or export from the DEFF environment. Camera ready DEFF files adhere completely to the standard Tagged Image File Format (TIFF). They may also contain additional tags which are not necessary to use to render the image and which in no way violate the TIFF standard. Camera ready files contain only a single frame image. They use only 8 bit gray, 24 bit RGB, or 8 bit palette pixel types. Camera ready DEFF files may be read by many popular off-the-shelf software products.

Another subset of the Image Class, the Pictorial Class, is intended to be used as a "small" image that represents a larger image or cineloop. It is expected to be combined with other pictorial images to create an "pictorial image directory."

The Archive Class is also a subclass to the Image Class. It contains most of the extensions to TIFF necessary to use the TIFF format for ultrasound purposes. It contains such things as the ability to store more than one frame, acquisition time of each frame in a loop, heart rate, etc.

In addition to images, DEFF files can store a variety of data items. Classes have been defined which deal with these non-data items such as "raw" ultrasound data, analysis data, and audio data.

2.0 Disk File System

2.1 Introduction

This section specifies the disk filing system used for the Data Exchange File Format. It contains information covering the physical media, the low level disk format, and the high level disk format.

2.2 Physical Media

Currently only two physical disk media are supported. This may change in the future.

The physical disk media used by DEFF are:

1) 5.25" magneto-optical (MO) rewritable disk with 512 bytes per sector. It is compatible with the standards defined in:
ISO/IEC JTC1 DIS10089-2A Continuous Composite Format and Cartridge Standard
and
ANSI X3B11.
The total formatted disk capacity is approximately 600 MB.

2) 3.5" magneto-optical (MO) rewritable disk with 512 bytes per sector. It is compatible with the standards defined in:
ISO/IEC 10090 Continuous Composite Format and Cartridge Standard
The total formatted disk capacity is approximately 125 MB (128MB is used as an informal reference only).

2.3 Low Level Disk Format

The low level formatting is done using Sony Mode 3 Fixed format.

2.4 High Level Disk Format

The 5.25" MO disk is double sided. Each side of the disk is treated as a independent volume containing approximately 300 MB per side. The 3.5" MO disk is single sided with all of the approximately 128 MB on the one side.

The media is formatted to appear as an unpartitioned disk compatible with MS-DOS(tm) 4.0. This means that the disk contains a customary boot sector, root directory, and file allocation tables (FAT).

The printing and archival databases are contained in the root directory of the disk. Images and other information are contained in the root directory or subdirectories on the disk.

Other than the disk capacity, there is no limit to the number of image files that could be contained on the disk.

If the disk is formatted by a DEFF compliant device, the boot sector, sector 0 of track 0, is formatted as follows:

Byte(s)	Value	Description
0-2	varies	Jump instruction to loader (NOPs) Note 1.
3-10	"ddddddddd"	The formatting DOS (vendor specific). Note 2.
11-12	0x0200	512 bytes/sector
13	0x20	32 sectors /cluster. A value of 0x10 (16) is also supported.
14-15	0x0001	1 sector in boot record
16	0x02	2 file allocation table (FAT) Note 3.
17-18	0x0200	512 root directory entries
19-20	0x0000	Flag for more than 65536 sectors/disk. Use offset 32 value.
21	0xF8	Flag for disk type = hard disk
22-23	varies	Sectors/FAT. Varies from disk to disk.
For 5.25" disk:		
24-25	0x001F	31 sectors/track
For 3.5" disk:		
24-25	0x0019	25 sectors/track
26-27	0x0001	1 side (head) per disk
28-31	0x00000000	0 reserved or hidden sectors
32-35	varies	Total sectors/disk. Varies from disk to disk.
36-37	0x0000	Physical drive number = 0
38	0x29	Extended boot record signature = 41
39-42	undefined	Volume serial number. Note 4.
43-53	"vvvvvvvvvvv"	The volume ID (vendor specific)
54-61	"FAT16 "	The file system label
:		
510-511	0x55AA	Signature flag

Note 1: These three bytes should be either 0xEB0090 (indicating a relative jump) or 0x909090 indicating NOPs. The bytes are for booting off the optical drive which DEFF does not support so they are not really used for any purpose other than by some programs which use them to validate the disk. The use of 0xEB0090 is known to be more generally compatible and is the reccommended choice. Readers of DEFF disks should ignore this field.
Note 2: While any eight characters appear to be valid in this field, the use of "MSDOS4.0" is known to be the preferred choice for this string. Some systems, upon finding this field not set to "MSDOS4.0" will ignore the sectors_per_fat field and use there own calculation. This may cause an error due to the calculation resulting in a different value than the sectors_per_fat field.
Note 3: Two FATs are recommended. One FAT could also be used but again may cause some incompatibility.
Note 4: The serial number may be any four bytes. A random or sequential number is preferred but is not required.

(MS-DOS is a trademark of Microsoft)

3.0 Locator Data Base

3.1 Introduction

This section covers the format of locator databases of the Data Exchange File Format. These locator databases are used to relate together individual DEFF files.

3.2 Locator Databases

There are two locator databases used. Each database allows for the presentation of an image list as well as providing direction to the location of the image files. Each database is contained in a single file. The two files reside in the root directory. Both databases have the same structure.

One database is used as a list of images which are considered as a print queue. All the images referenced in that database are to be printed.

The second database is used as a list of images in a local archive queue. This is a master list of all DEFF files contained on the disk.

"PRINTQ.DB" is used as a directory of images to be printed.
"ARCHIVEQ.DB" is used for a general directory of all DEFF files on the disk.

The justification for having two databases is:
1. The PRINTQ.DB is unencumbered with a variety of options, formats, and information not relating to printing.
2. The ARCHIVEQ.DB can change to a more sophisticated form without affecting the printing devices.
3. The order and content of the images in the print queue can be manipulated independently of the time of acquisition.
This allows the print reader development to be simple while at the same time reducing the chance of possible corruption and allowing for future DEFF upgrades.

3.3 Locator Database Structure

Each database is composed of four areas:
- Header
- In-use records
- Deleted records
- Free Area

The Header is the first part of the file, the Free Area is the last.

In the middle, records may be either in-use (valid) or marked as deleted. These records will be variable length depending on the software revision that wrote them. Any one software version is expected to write each record using a fixed length. Deleted records are never re-used in-place.
Rather, the file is packed to reclaim space in the Free Area. This approach allows the same disk to be used by more than one system while each system may have different software version. It also insures that the database records are always maintained in roughly chronological order.

(The order is not perfectly chronological if two different systems write to the disk and have unsynchronized clocks.)

The Free Area may be used to reserve space on the disk for the locator data base file. If a large Free Area is created when the data base file is initially created, then fragmentation of the locator database file will be minimized.

3.3.1 Format Definitions

CR is 0x0D
LF is 0x0A
FF is 0x0C
NULL is 0x00
"byte" refers to 8 bit data, but stored as two ASCII characters
"word" refers to 16 bit data, but stored as four characters
"long" refers to 32 bit data, but stored as eight characters

"ASCII hex word" refers to a string of four characters which when evaluated as a hexidecimal number yield a 16 bit value. For example, the string "0202" represents the number 0x202 or 514 decimal.

The locator databases are intended to be viewable using a standard text editor. Therefore, all fields are defined in such a way as to avoid what a text editor would see as embedded control characters. No fields are expected to contain control characters other thean CR, LF, and FF.

Strings do not have terminators and may not have embedded control characters other than CR, LF, and FF. When strings are smaller than the field size the string is contained in the first part of the field with spaces (" ") added as padding.

In fields with string constants, the string values are considered case sensitive. For example, "Vi" is different than "VI".

3.3.2 Header Description

The header has the following format:

Length	Title	Description
8	headerSize	ASCII hex long Size of this header, in bytes. Includes headerSize field, FF, and NULL. Always "00000022" for this specification version.
2		CR LF
8	freeAreaOffset	ASCII hex long Offset from the beginning of the file to the location of the free area. Never points to embedded Deleted Records. This is the location that a new record will be added. It may point to the end of the file.
2		CR LF
4	inUseCount	ASCII hex word Number of In-use records.
2		CR LF
4	deletedCount	ASCII hex word Number of embedded Deleted Records. This value is used to determine when packing of the database should occur.
2		CR LF

2	FF NULL The NULL is required.

---

34 bytes total

3.3.3 In-Use Record Description

Each record in the file will have the following format:

Length	Title	Description
4	recordLength	ASCII hex word Size in bytes of this record. Includes recordLength field and <FF>. For In-Use records, a value of 010A is used for this revision level. Note that not all records in the file necessarily have the same length if, for instance, some records were written by different versions of software.
2		CR LF
8	swVersion	ASCII string Software revision that created this entry. Always "WXYZ0001" for this version. The WXYZ is company specific and each product should use a different string. If the company initials "WXYZ" are less than four characters, the initials are left justified followed by zeroes. The "0001" (decimal) portion of the revision string is intended to be common to all DEFF locator databases regardless of originating company. Thus, the last four digits can be used, independently of the first four, to indicate the locator database record format.
2		CR LF
11	filename	ASCII string The first 8 characters are the path and file. The last 3 characters are always the file extension. Always in one of two forms: sslfffffeee or fffffffeee where the ss is the subdirectory name of between 1 and 6 characters, fffff is the filename of between 1 and 8 characters, and eee is the file extension. Note that the "l" is included but the "dot" is left out of the file name of "fffff.eee." The 'l' may occur in positions from slfffffeee to ssssssslfeee or may be absent as in fffffffeee. If the filename is less than the field size then the filename will be padded after the name with spaces. For example, using '+' for the spaces, sslfff+++e++ indicates a filename of "fff.e" residing in subdirectory "ss." See the following section on filename specification requirements.
2		CR LF

61	patientInfo	ASCII string This field contains four items as follows: "|patientID|lastName|firstName|middle|" The quotes are not included but the vertical bars "|" are included. If the items do not fit in the field size (61 characters) then the string is truncated to fit the field and at least the last vertical bar will be missing. "middle" may be either an initial, name, or blank (space). If the string does not fill the field, the string is padded to 61 characters using spaces after the last vertical bar. If the patient ID and names are not separable, the vertical bars are omitted and the entire string is to be treated as if it were a patient ID. Each application which adds records to the database must guarantee that the patientID is unique to the disk, meaning that a single patient name must have a unique patientID.
2		CR LF
19	dateTime	ASCII string Date and time of the acquisition. Format is "YYYY:MM:DD HH:MM:SS" not including the quotes but including the colons and space. The hours "HH" are in 24 hour clock format.
2		CR LF
120	applicInfo	ASCII string Interpretation is dependent on the type of locator database. All tags have the same format: "FF|xx " The quotes are not included, the FF is a two character flag tag, the vertical bar, '|', is included, xx is an ASCII string indicating the value (usually a hex byte, except L2), and the trailing space is always included to separate the tags. All tags are optional and may occur in any order. The applicInfo field is padded with spaces to the field size of 120. Any unrecognized tags should be silently ignored by the reader. Common tags are common to all locator databases.

Common Tags

Tag	Value Description
La|	Obsolete, do not use when creating new records. ASCII string of fixed length of 19 characters. Label of image. Expected to be used for user labeling. The tag separating space between this tag and the next is not considered part of the 19 characters.
L2|	ASCII string of fixed length of 30 characters. Label of image. Expected to be used for user labeling. The tag separating space between this tag and the next is not considered part of the 30 characters.

ARCHIVEQ.DB Tags

Tag	Value Description
API	ASCII hex byte Also Printed. 00 when the image is not also destined to the printer. 01 when image was destined to the printer. Once set, it is never cleared. It is used for creating reprints in the same manner as the originals.
Vil	ASCII hex byte View number
NVI	ASCII hex byte Number of views in this stage
AEI	ASCII hex byte Archive Class image file exists. 00 when no associated Archive Class image file exists. 01 when Archive Class image file exists. Archive Class image filename is "sslfffff.ACI" where "sslfffff" is the same as in the filename field above, and "ACI" is a constant.
CEI	ASCII hex byte Camera Ready Class image file exists. 00 when no associated Camera Ready image file exists. 01 when Camera Ready image file exists. Camera Ready image filename is "sslfffff.CRI" where "sslfffff" is the same as in the filename field above, and "CRI" is a constant.
PEI	ASCII hex byte Pictorial Class image file exists. 00 when no associated Pictorial Image file exists. 01 when Pictorial Image file exists. Pictorial Image filename is "sslfffff.PDI" where "sslfffff" is the same as in the filename field above, and "PDI" is a constant.
REI	ASCII hex byte Results Class data file exists. 00 when no associated Results data file exists. 01 when Results data file exists. Results data filename is "sslfffff.RCD" where "sslfffff" is the same as in the filename field above, and "RCD" is a constant.
UEI	ASCII hex byte A unspecified type of file exists. 00 when no associated file exists. 01 when file exists. Filename is "sslfffff.eee" as specified in the filename field above.
PNI	ASCII string of fixed length of 20 characters. Protocol name. The title of the protocol associated with the image.
EXI	ASCII string of fixed length of 20 characters. Exam name. The title of the exam associated with the image.
VFI	ASCII string of fixed length of 6 characters. Vendor flags. Specific use is vendor dependent. It has meaning only when first qualifying it by the vendor/product name in the swVersion field.
ACI	ASCII hex byte Archive complete. Set when the image has been transferred off disk to some "central" archive. 00 = not yet archived. 01 = image was archived.

PRINTQ.DB Tags

Tag	Value Description
USI	ASCII hex byte UserID. Used when the printer has a setup storage which is indicated by a userID number. The meaning of the USI tag is vendor specific. For example, it may be used to select contrast curves or route images to various printers. Writers of this field should use a default setting of 01 and increment upward from there.
PCI	ASCII hex byte Print Completed. 00 = Not yet printed, 01 = image was printed
FFI	ASCII hex byte Form Feed. Used to indicate that after this image is printed, the film or paper should be advanced to the next sheet. 00 = no, 01 = do a form feed, default is no. Note: Some printers may choose to ignore this field.
DRI	ASCII hex byte Obsolete. Do not use when creating new records. Should be ignored if found. [Note: This flag is reserved and should not be reused]
DII	ASCII hex byte Obsolete. Do not use when creating new records. Should be ignored if found by printer. [Note: This flag is reserved and should not be reused]

2		CR LF
4	vendorSpec1	Four ASCII characters Specific use is vendor dependent. Thus, this field is reserved for use by each vendor independently and has meaning only when first qualifying it by the vendor name in the swVersion field. The characters used here are to be printable, not control characters.
2		CR LF
4	vendorSpec2	Four ASCII characters Specific use is vendor dependent. Thus, this field is reserved for use by each vendor independently and has meaning only when first qualifying it by the vendor name in the swVersion field. The characters used here are to be printable, not control characters.
2		CR LF
4	vendorSpec3	Four ASCII characters Specific use is vendor dependent. Thus, this field is reserved for use by each vendor independently and has meaning only when first qualifying it by the vendor name in the swVersion field. The characters used here are to be printable, not control characters.
2		CR LF
4	grayColorFlag	ASCII hex word When the DB record gives information about an image file then this field indicates whether the image is gray scale only or whether the

		image contains color. Note that this flag does not indicate the pixel type. Thus, a gray image may have a palette or RGB pixel type. Value — Interpretation 8000 — Gray scale image 8001 — Color image 0-5 — Reserved A001-A007 — Reserved
2		CR LF
4	numberFrames	ASCII hex word When the DB record gives information about an image file then this field indicates the number of frames in the image file.
2		CR LF
1		FF

---

266 (0x10A) bytes total

3.3.4 Deleted Record Description

Each deleted record in the file will have the following format:

Length	Title	Description
4	emptyFlag	ASCII hex word 2's complement negative of the size in bytes of this deleted record. Note that not all records in the file necessarily have the same length if, for instance, some records were written by different versions of software. Includes emptyFlag field and <FF>. Always FEF6 for this version of the software. An obsolete previous version of the locator database used zero (0000) to indicate a deleted record. In that case the end-of-record mark had to be searched for since the number of bytes in the record was not known.
x		Some number of bytes.
1		FF

3.4 Using the Locator Databases

This is a description of the required operational procedure of using the disk's locator databases. It is important that every implementation using DEFF operate within the procedure so that systems from different vendors can interoperate correctly.

The ARCHIVEQ.DB database contains a master list of all the images and other data. The PRINTQ.DB database acts as a simple print queue list to convey that a set of images is desired to be printed.

The following discussion refers to an "acquisition system". In general, an acquisition system is the source of images. An acquisition system could be an ultrasound system, a video-to-disk image capture device, a review station, etc.

Whenever an image is written to the disk by an acquisition system, a record is created in the ARCHIVEQ.DB database. Thus, the ARCHIVEQ.DB database will contain a master list of all types of DEFF files. A single record will indicate the types of image files created which are all associated with the same image content. For example, a single record may indicate that there is an Archive Class image, a Camera Ready Class Image, and a Pictorial Directory Class image of which all three were created from the same acquisition but are stored in different formats and different files.

The database record "indicates" the presence of the image/data files using the "Exists" flags of: Archive Class Exists (AE), Camera Ready Class Exists (CE), Pictorial Class Exists (PE), Results Class Exists (RE), and Unspecified File Exists (UE). For each flag that is set, there will be a file of that type stored on the disk. An image file which could be formatted as either a Camera Ready Image or an Archive Image shall be formatted as a Camera Ready Image file.

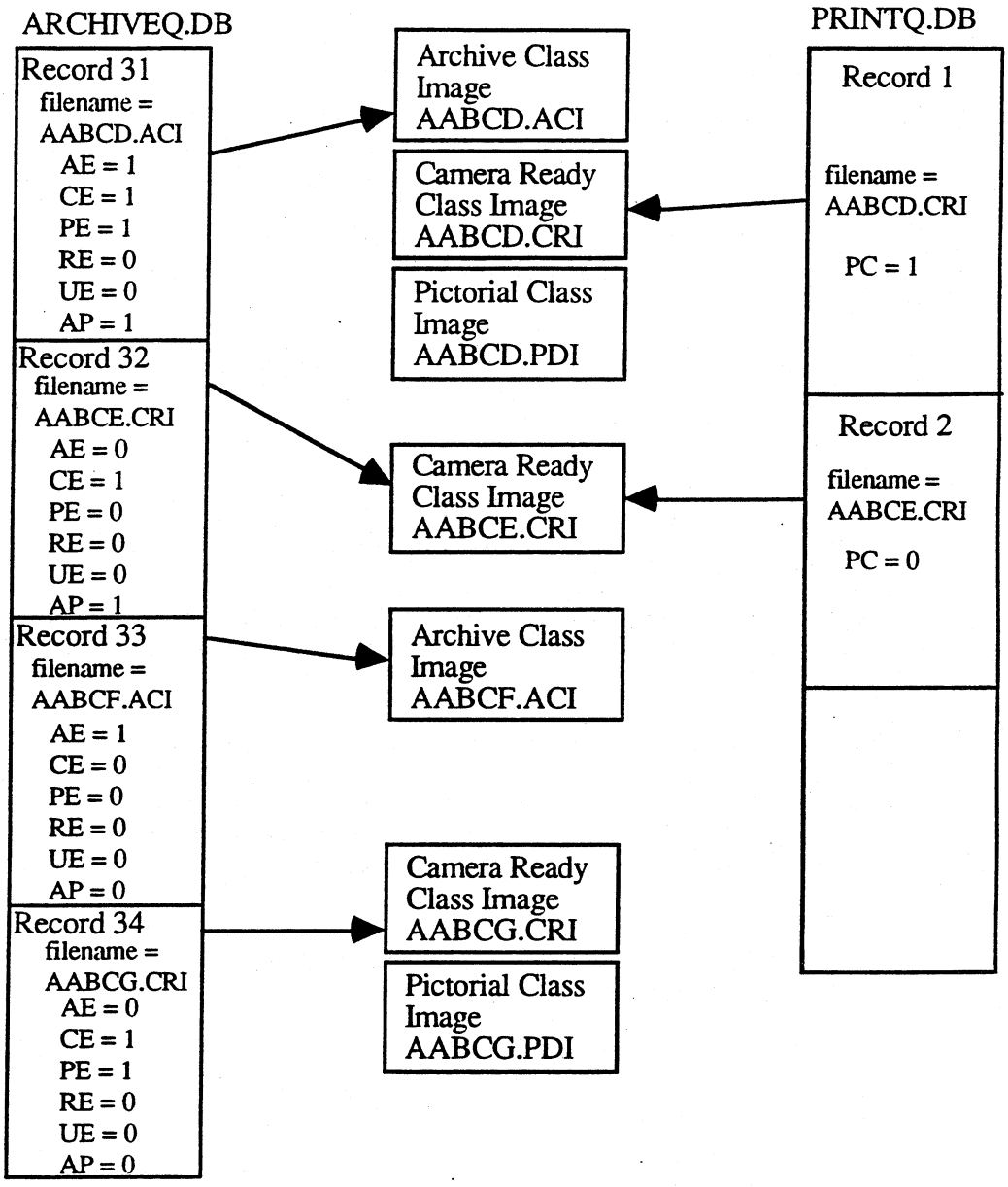
When it is desired to indicate to a printer that the printer should print an image, a record is created in the PRINTQ.DB that has a filename field which refers to the same Camera Ready image file as indicated in one of the ARCHIVEQ.DB records.

The figure below gives examples of the databases in use. The first example shows an image set which has an Archive Class, Camera Ready Class, and Pictorial Class image. Record 31 in the ARCHIVEQ.DB "points" to the Archive Class Image. The flag AE indicates the existence of the Archive Class image, the flag CE indicates the existence of the Camera Ready image, and the flag PE indicates the existence of the Pictorial Class image. No Results data or Unspecified file exists so RE and UE are both zero. The Camera Ready image is also "pointed" to by the PRINTQ.DB. Print Complete (PC) is set so it is known that this image has been printed at least once. Also Printed (AP) indicates image was referenced at some time by a record in the PRINTQ.DB.

Example 2 shows Record 32 in the ARCHIVEQ.DB indicating through the flags the presence of only the Camera Ready image file. The filename extension is ".CRI" and thus the record "points" to the Camera Ready image. The PRINTQ.DB Record 2 also points to this image and because Print Complete is not set, the image is known to not yet have been printed.

Example 3 shows Record 33 pointing to an Archive Class image. No print record is associated with that image. Also Printed (AP) being zero indicates that the image was never referenced by a record in the PRINTQ.DB file.

Example 4 shows Record 34 pointing to a Camera Ready image which is not and was never in the PRINTQ.DB. It also has a Pictorial image associated with it.



3.4.1 Filename Requirements

The database for archiving is always named "ARCHIVEQ.DB".
The database for printing is always named "PRINTQ.DB".

As described in the description for the locator database field for "filename", there are three parts to a filename as in: "ssfffff.eee".

The "ss" portion describing the optional single subdirectory path has no restrictions placed on its value other than those of MS-DOS and the maximum allowable length of six characters.

The "fffff" portion is required to be a sequential base 26 number (i.e. letters only: AAAAA to ZZZZZ). When creating a new filename, there are two approaches which can be used. In case 1, an application first decides what subdirectory the file will be stored to. Then the last record

in the ARCHIVEQ.DB which uses that subdirectory must be found. The new filename should be created using one plus the "fffff" found in last record. In case 2, an application may just use the database's last record's filename plus one. Note that with this approach, filenames are only unique to a subdirectory.

For example, sequential files could be named AAABY.ACI, AAABZ.ACI, AAACA.ACI, AAACB.ACI, etc. Alphabetical overflow (ZZZZZ to AAAAA) must also be handled.

The "eee" portion is a constant for the these types of DEFF files:

Extension	File Type
ACI	Archive Class Image file
CRI	Camera Ready Image file
PDI	Pictorial Class Image file
RCD	Results Class Data
UCD	Unprocessed (Research) Class Data
ACS	Audio Class Samples

The file types above refer to the lowest subclass contained within a file. For instance, an Archive Class Image file is one containing the DEFF Base Class, Image Class, and Archive Class.

Implementations should always check for the existence of a file of the name being given to a newly created file in order to avoid unexpectedly overwriting an existing file.

3.4.2 Acquisition System Operation

The acquisition systems are responsible for all management of the files on the disk. This includes creating and deleting database records and creating and deleting image files.

The acquisition system operational procedure is as follows:

On an acquisition system the operator inserts an optical disk.
If the ARCHIVEQ.DB and PRINTQ.DB files do not exist
    then they are created.
Inside the PRINTQ.DB, for each record with the print complete flag set (PC = 1)
    the record is deleted as specified in section 3.3.4.
If the databases contain deleted records
    then the databases are optionally compressed to remove deleted records

The operator enters his/her print User Identification Number in the user setups.
The operator scans or recalls a previously saved image.

When the operator activates the save (non-print) image control the ARCHIVEQ.DB database and image files are updated as follows:
    A new ARCHIVEQ.DB record is created
    The patient name and date are filled in. (If a patient ID and/or name is not available, the unit #, and a large sequential number should be used.)
    The filename field is filled in with the image file name (either fffff.CRI or fffff.ACI)
    Label (L2) is set to what is desired in recall directory listings, usually the user annotation
    If the image is Camera Ready
        Archive Class Exists is set to zero (AE=0)
        Camera Ready Class Exists is set to one (CE=1)
        The Camera Ready Class image file is written to disk (fffff.CRI)
    else
        Archive Class Exists is set to one (AE=1)
        Camera Ready Class Exists is set to zero (CE=0)
        The Archive Class image file is written to disk (fffff.ACI)
    If a pictorial directory is supported
        Pictorial Class Image Exists flag is set to one (PE=1).
        the Pictorial Class image file is written to disk (fffff.PDI)
    Else
        Pictorial Class Image Exists flag is set to zero (PE=0).
Also Printed (AP) is not set (AP=0) indicating that the image was not originally destined to be sent to the printer.

When the operator activates the print-frame control the ARCHIVEQ.DB database and image files are updated as follows:

A new ARCHIVEQ.DB record is created
The filename field is filled in with the image file name (fffff.CRI)
Label (L2) is set to what is desired in recall directory listings, usually the user annotation
Camera Ready Class Exists is set to one (CE=1)
The Camera Ready Class image file is written to disk (fffff.CRI)
If the acquisition desires a native Archive Class file to be kept also
    Archive Class Exists is set to one (AE=1)
    The Archive Class image file is written to disk (fffff.ACI)
else
    Archive Class Exists is set to zero (AE=0)
If a pictorial directory is supported
    Pictorial Class Image Exists (PE) flag is set to one.
    the Pictorial Class image file is written to disk (fffff.PDI)
Else
    Pictorial Class Image Exists flag is set to zero (PE=0).

Also Printed (AP) is set (AP=1) indicating that the image is destined to be sent to the printer.
The PRINTQ.DB database is updated as follows:

A new record is created
Patient name and date are filled in.
The filename field is filled in with the image file name (fffff.CRI) The filename must always be the camera ready version.
Label (L2) is set to what is desired in print directory listings, usually the user annotation
Print Complete is set to zero (PC=0)

If the operator looks at the image directory (i.e. ARCHIVEQ. DB database) and desires to delete an image then:
Find the image record in the ARCHIVEQ.DB database.
If Camera Ready Exists is set (CE=1)
    If the image is referenced in the PrintQ database (Finding it requires a database search.)
        If the PrintQ Print Complete flag is zero
            If it is not okay to delete an unprinted image
                Abort this entire deleting operation
        Delete PrintQ record
    Delete camera ready image (fffff.CRI)
If Archive Class Exists is set (AE=1), delete archive image (fffff.ACI)
If Pictorial Directory Image Exists is set (PE=1), delete PDI image (fffff.PDI)
If Results Class Data Exists is set (RE=1), delete RCD image (fffff.RCD)
If Unspecified File Exists flag is set (UE =1), delete file as named in filename field.
The ARCHIVEQ.DB record is deleted.

Optical disk is (eventually) removed from the acquisition system. The disk may or may not now go through the printing system procedure outlined below.

Note: Eventually, on the acquisition system the disk is re-inserted.
The ARCHIVEQ.DB database still points to the printed images. The PRINTQ.DB database will also point to the images but (usually) the print complete flag is set. A re-inserted disk may or may not have been sent to a printer and may or may not have completed the printing. In any case, the procedure as stated above is followed once again and the acquisition to camera cycle repeats.

3.4.3 Printing System Operation

Using the locator database for printing requires the reading device to check the patientInfo field of each record to determine the last image for each patient. This is necessary to avoid the possibility of printing images of two different patients on one sheet.

On printing system the optical disk is inserted into a printing system.
A temporary variable, SavedPatientInfo, is set to some "impossible" name
While(PRINTQ.DB record is deleted or has an unsupported version
    Record is skipped.
If no valid record is found
    then printing aborts
If the Print Complete flag is set (PC=1) then the printer has the option of providing one of the following implementation choices:

1. Skip the record. No re-print supported.
2. Ask the user if reprint is desired (Preferably done just once at initial printer setup.)
3. Reprint image only on user demand.

If the previously printed image(s) is to be re-printed, it is handled just like first time prints.
If the printing device supports the User Identification Number (Us)
then from the found record, User Identification Number (Us) extracted and used to setup the printer defaults, select the camera, or some other vendor specific purpose
If the sheet (i.e. film) format allows for multiple images on one sheet
    then
    SavedPatientInfo is compared with the record's patientInfo
    If this image belongs to the same patient
        then the image may be added to the same film if possible (See note below)
        else
        SavedPatientInfo is set from the record's patientInfo field and
        a new sheet is used for this new patient.
If the grayColorFlag indicates a non-supported image type
    printing of this image is aborted or marked for conversion or some other handling
The image file name is extracted from the record.
The file image is read. Printing takes place now if appropriate for that device.
The Print Complete flag is set (PC=1)
If the Form Feed flag (FF) is set
    then this sheet is considered full
The above image reading procedure repeats until there are no more images listed in the PRINTQ.DB data base. Or possibly, the process could be aborted in the middle with images left on the disk which have not yet been printed.

Note: Multiple images belonging to the same patient (i.e. patientInfo field is identical) may have various pixel matrix sizes, aspect ratios, pixel types, etc. Handling such situations is a printer implementation issue and is not specified here.

4.0 Base Class

4.1 Introduction

This section specifies the parameters which are common to all DEFF data files. Other separate sections cover the subclasses of DEFF used to store specific types of data.

4.2 File Structure

The DEFF file (see figure 2.0-1) consists of a TIFF header, followed by a singly linked list of TIFF Image File Directories (IFDs). The File Header and TIFF IFD contains, as it's last element, a four-byte file offset to the (possibly) next TIFF IFD. The link offset of the last TIFF IFD contains NULL (0), indicating the end of file. DEFF only uses a single TIFF IFD. (Note: DEFF 2.1 and previous versions, used multiple IFDs linked through use of the Next IFD Offset. For this reason, in implementations not compatible with DEFF 2.1 and prior, the Next IFD Offset should be written as a zero but not checked when reading old files.)

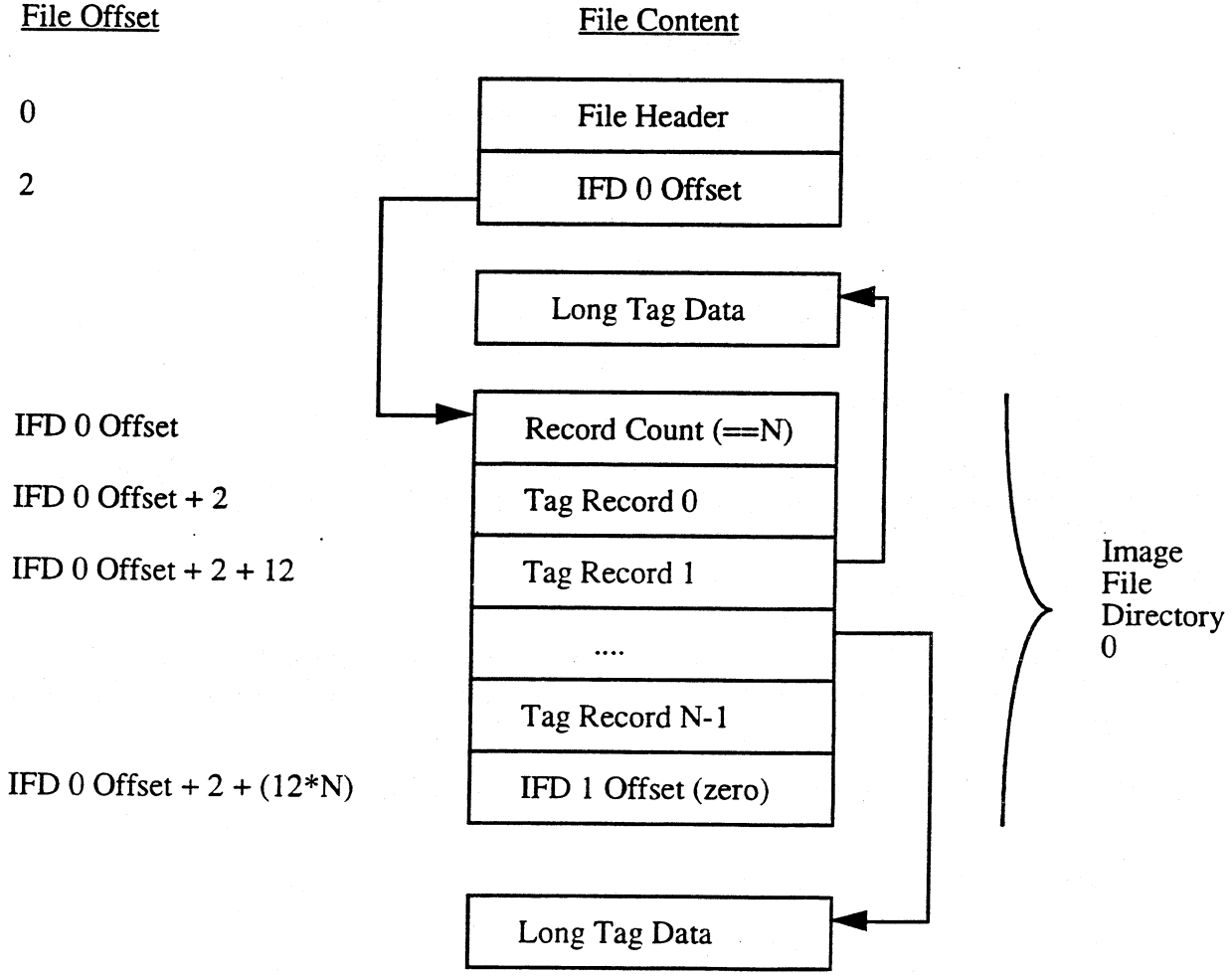


Figure 4.2-1 Data Exchange File Format

4.2.1 File Header

The first element of the File Header (figure 2.1-1) is the Byte order. The allowed values are:

"II" (0x4949) Indicates Intel byte ordering (low byte first, then successively higher bytes).

"MM" (0x4D4D) Indicates Motorola byte ordering (high byte first, then successively lower bytes).

In both formats, character strings are stored left-to-right in sequential bytes.

The second element of the File Header is the "version number." The only allowed value is 0x002A. This value helps TIFF readers to be sure that they are reading a TIFF file.

The third and final element of the File Header is the four- byte file offset of the first Image File Directory.

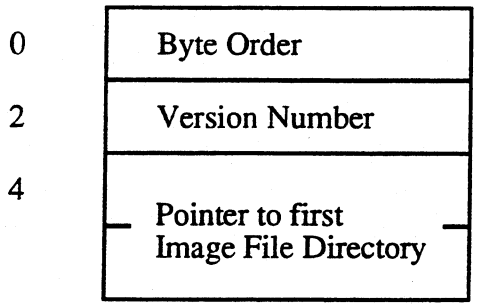


Figure 4.2.1-1 File Header Format

4.2.2 Image File Directory

Each Image File Directory (IFD) is arranged with the two-byte tag count first, followed by its tag records in ascending numerical order, followed last by the four-byte file offset of the next IFD (Zero now, but used in previous DEFF versions).

4.2.3 Tag Record Format

Each Tag Record (figure 2.3.-1) begins with the tag number, a two-byte integer that indicates the meaning of the tag. For example, a tag value of 0x11A is the "X resolution" tag, giving the number of pixels per inch horizontally of the raster image.

The next two bytes give the data type of data contained in the tag:

0x0001	BYTE	8 bit bytes
0x0002	ASCII	8 bit characters
0x0003	SHORT	16 bit unsigned integers
0x0004	LONG	32 bit unsigned integer
0x0005	RATIONAL	Two 32 bit unsigned integers, the first being the numerator of a fraction, the second being the denominator.

The next four bytes give the number of data items contained in the Tag Record. All tag values are assumed to be arrays where the length field indicates the length of the array. The units of length are the data type of the Tag Record. For example, if the data type is SHORT, and the length is 8, then the total data length is 2 bytes/short * 8 shorts = 16 bytes.

ASCII length is handled somewhat differently: ASCII strings are terminated with a NULL (0), and are further padded with NULLs to an even number of bytes. The length includes the string and it's NULL terminator, but not the padding characters. DEFF will usually write a fixed pre-defined number of characters for a given string. Thus, there will usually be more than one padding character. DEFF requires that a string contain no control characters other than Line Feed (LF, 0x0A) and Form Feed (FF, 0x0C). Thus the end of a line is marked with a single LF rather than a LF and CR combination.

The last four bytes of the tag can have one of two purposes. If the total data length is less than or equal to four bytes, the tag data is contained here. If the total data length is greater than four bytes, these bytes give a pointer (file offset) to the actual data. TIFF places no restrictions on where this data actually resides (it can follow the IFD, follow the last IFD, or be between two other IFDs). However, the standard DEFF approach will be to write the header, the Artist tag data, the first (and only) IFD, the first IFD's data.

For examples, consult the reference sections.

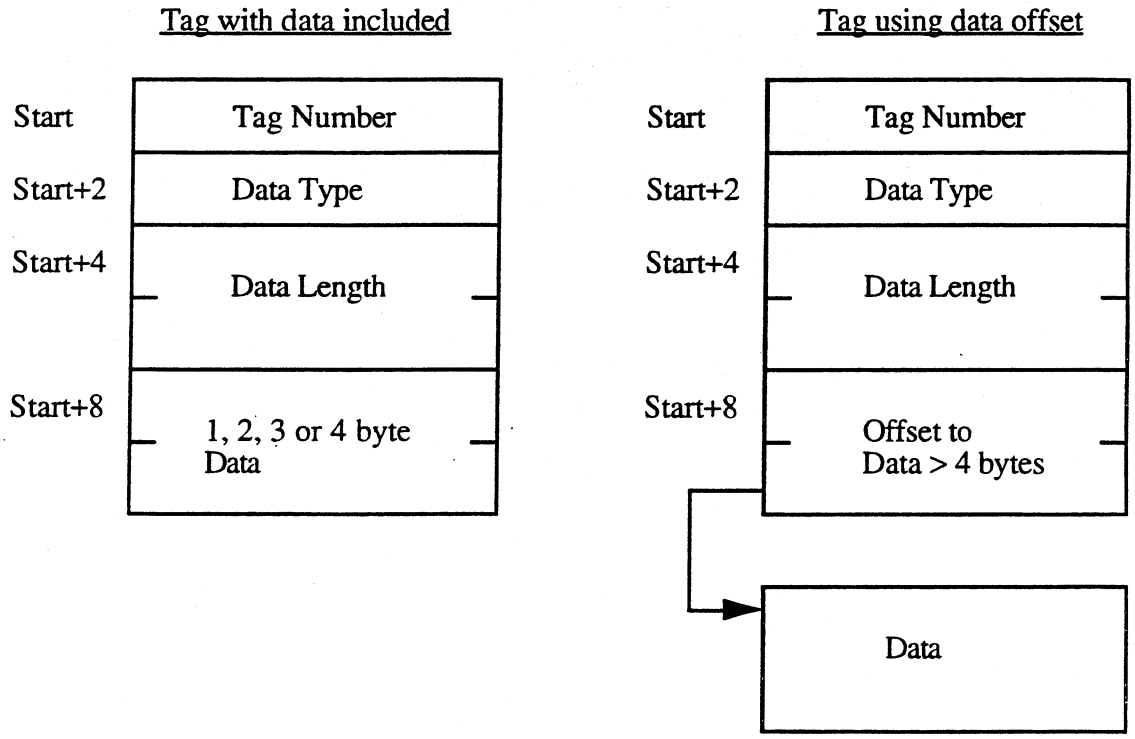


Figure 4.2.3-1 Tag Record Formats

4.2.4 DEFF Extensions to TIFF

For image review purposes, it is necessary to add more information to the DEFF file than supported by the TIFF standard. In order to maintain TIFF reader compatibility, it is necessary to add tags which do not conflict with the TIFF standard. The TIFF standard allows for this condition by providing "private" tags. These tags are registered with Aldus so that no other TIFF writer will use the same tag number. In order to support many TIFF vendors, Aldus issues only five tags to a given vendor. Since this is inadequate to support the quantity of information to be added, an indirect approach is taken where one of the official private tags points to more tags which are not TIFF compatible.

DEFF calls this indirect pointer tag "ExtendedTagsOffset." ExtendedTagsOffset points to the DEFF specific tags contained in a special Extended IFD. The Extended IFD is in official TIFF format. The tag numbers are not official TIFF but do not need to be since no TIFF reader will encounter them. Thus, there will not be TIFF reader conflicts caused by these tags. See figure 4.2.4-1.

Five official TIFF tags have been reserved for ultrasound use with consent from Aldus. These tags are numbered 0x8440 through 0x8444 inclusive. The tags are used as:

Tag Number	Use
0x8440	Archive Pixel Type
0x8441	Palette Ranges
0x8442	Extended Tags Offset
0x8443	unused
0x8444	unused

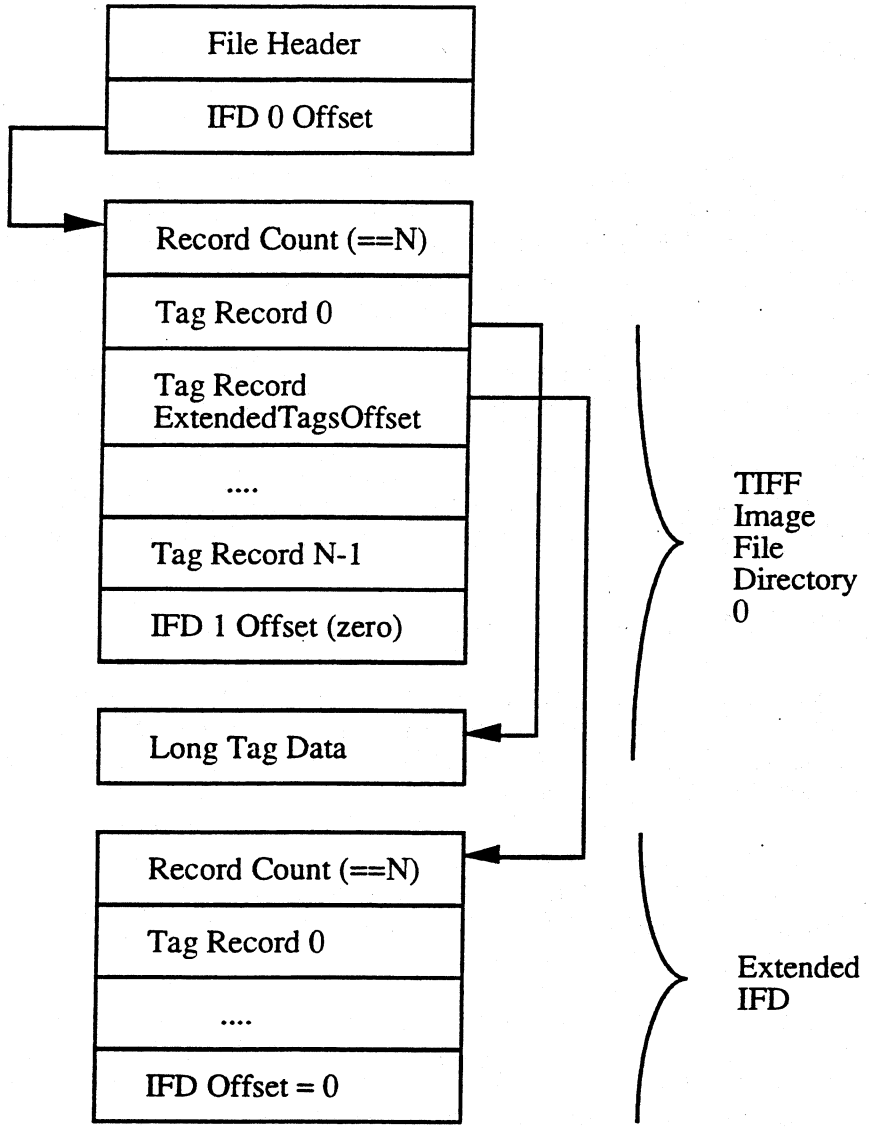


Figure 4.2.4-1 TIFF and Extended Image File Directories

4.3 DEFF Specific Concepts

4.3.1 Loops of Frames

The TIFF specification is not designed to handle multiple frames (loops), or non-image data. By TIFF definition, all tag records in an IFD are associated only with a single frame. Most of the parameters in a DEFF file change only on a loop basis rather than on a frame by frame basis. This data includes the copyright notice, the color palette, loop descriptions, make, model, and even ImageLength and ImageWidth.

To handle data which does change on a frame by frame basis, the affected tag will be expanded into an array whose size is a multiple of the number of frames in the loop.

Most TIFF readers can only handle single images and cannot handle loops of any sort. Therefore, where such compatibility is a concern (i.e. camera ready and other export files), only single image DEFF files will be produced with all TIFF rules followed.

4.3.2 Backward and Forward Compatibility

A key concept of DEFF files is the expected ability of new software versions to be able to read old versions of DEFF files. The reverse is also expected, new DEFF files should be able to be read by old software versions. This capability is supported by writing the DEFF reader software to adhere to several concepts:

- Ignore tags that it doesn't recognize.
- When possible, make up defaults for tags that aren't in the file.
- Disable functions which require tag values not contained in the file.
- Avoid re-writing a file to add tags.

The last item is required to avoid missing the copying of tags which the reader did not recognize in the original file. However, in some cases, the file must be rewritten to add tags which were not in the original file. If the tag addition is done by old software which doesn't now about tags added in a new DEFF version, then the new DEFF tags will be unavoidably lost and the new file will created to a DEFF version older than the DEFF version of the original.

4.3.3 Future Tag Numbers

In an effort to avoid coordination problems when adding enhancements to the DEFF specification and DEFF handling software, addition of new tags will be restricted to specific ranges of tag numbers, according to the "Reserved Tag" numbers as listed for a given DEFF class.

It is expected that the DEFF document and all software will be eventually updated to handle any additional tags in a DEFF subclass that is supported by that software. The separation of the tag number ranges is simply to avoid two independent uses of the same tag number due to separate software development efforts.

This DEFF section only defines the file format for the base or common DEFF tags.
It is anticipated that future enhancements will be required. It is expected that all such future enhancements will be organized as sub classes to the base described here.
Although tag number ranges are reserved for a given class, a subclass may have tag numbers which do not fall in the reserved range. Thus, the TIFF approach of having no significance implied by the tag number is maintained.

4.3.4 Required vs. Optional Tags

Each tag specified in this section is marked as either required or optional. If the tag is marked as required, then when writing that class of DEFF file, the DEFF writer must put the tag in the file and the DEFF reader can count on it being there.

Optional tags are written when they are applicable and if the specific version of the DEFF writer supports the tag. They are not left out simply because they contain the same value as the default. Thus the DEFF reader cannot be assured that the optional tag is included in the file. If the optional tag is missing, the reader will have to use the default specified in this section or do without that tag's value.

Note that required tags can never be deleted since old software will demand their presence. Thus they must be chosen very carefully.

4.3.5 Numbering Conventions

When numbering items, for example frames, the index numbering always starts with zero and goes to N-1. When indicating the number of items, the value always represents the number of items, N, as if counting from one. Thus, an N frame cineloop will have frames numbered from 0 to N-1.

This approach implies that using an index value of zero as a flag is not allowed since the first item is indexed as item 0.

4.3.6 Version Control

The DEFF Version tag indicates the sequential integral version of the DEFF specification that was used to generate the file. This number is defined in Appendix A for each version of DEFF. The DEFF Version tag along with the Make, Model, and Software tags gives a complete description of the version of the DEFF file.

Software that generates DEFF files should fill in the DEFF version with the latest number with which it is compatible.

Software that reads DEFF files should follow a compatibility determination process such as:

```
If mySoftwareDEFFLevel < fileDEFFLevel
    My software is older than the file's DEFF version so read the file using only the
    portion my software needs
Else
    Compare fileDEFFLevel and Software Version with a VersionsWithKnownProblems
    table
    If no known problems
            Read the file using only the tags supported in that fileDEFFLevel
Else
    If the known problem is irrecoverable
    Abort the read
Else
    Read the file and try to translate the file into a usable form
```

4.4 Tag Descriptions

Tags already defined in revision 5.0 of the TIFF standard will be used as defined. Extensions to the standard will be used when a tag does not exist, or the use of an existing tag would be non-standard. These two categories of tags are described separately.

Not all TIFF-defined tags are described here. Such tags are not used or supported in DEFF files. For this reason, and to support backward compatibility of images to the maximum extent possible, unrecognized tags should be ignored when loading these files.

In tag data fields of an enumerated type, a value with the MSB set indicates a private vendor specific usage. Multiple vendors may use the same enumerated value for different purposes.

4.4.1 DEFF Base Class Tag List

This list contains only those tags common to all DEFF files. It also includes tag ranges or numbers reserved for special purposes. All numbers are in hexadecimal. The "Required" field indicates whether the tag is required for this particular class. If the field is "cond" then inclusion of the tag is dependent on some other tag's value.

IFD	Tag #	Type	Required	Description
TIFF	010E	ascii	Yes	Image Description
TIFF	010F	ascii	Yes	Make
TIFF	0110	ascii	Yes	Model
TIFF	0131	ascii	Yes	Software
TIFF	0132	ascii	Yes	Date/Time
TIFF	013B	ascii	Yes	Artist
TIFF	8442	long	Yes	Extended Tags Offset
Ext	401F	long	No	Patient Demographics
Ext	400A	ascii	Yes	Patient/Exam Information (formerly Image Information)
Ext	400E	ascii	Yes	Original Machine
Ext	401C	long	Yes	DEFF Version
Ext	401E	long	Yes	DEFF Subclass
Ext	6002	ascii	No	Source Machine

Obsolete tags:

Ext	6000	Source Offset

Reserved tags:

TIFF	8443	Officially registered tag Reserved for future use
TIFF	8444	Officially registered tag Reserved for future use
Ext	4020-4FFF	Reserved for Base Class specification
Ext	5000-5FFF	Reserved for Research Class specification
Ext	6004-6FFF	Reserved for special definition
Ext	9000-9FFF	Reserved for Image Class specification
Ext	A000-AFFF	Reserved for Audio Class specification
Ext	B000-BFFF	Reserved for Analysis Class specification
Ext	F000-FFFF	Reserved for vendor dependent experimental use

4.4.2 TIFF IFD Tags

Artist

Tag	0x013B
Type	ASCII
Length	254
Range	n/a
Required	

This tag record contains a copyright and/or proprietary notice. The actual null terminated copyright string resides in the DEFF file immediately after the file header.

It is recommended that any copyright notice that is put into the file allow for the customary clinical copying of the image from one vendor's equipment to another. Otherwise, the ability to send images from one company's acquisition system to another company's review system would be legally prevented.

Date / Time

Tag	0x0132
Type	ASCII
Length	20
Range	n/a
Required	

This tag stores the null-terminated ASCII Date and Time at which the file was created or last modified. The form is

"yyyy:mm:dd hh:mm:ss"

where hours (hh) is 24-hour format local time. Conversion country-specific form is the responsibility of the application program.

Extended Tags Offset

Tag	0x8442
Type	LONG
Length	1
Range	$0..2^{32}-1$
Required	

The Extended Tags Offset points to the non-TIFF compatible Extended IFD which contains the tags specific to DEFF.

This Extended IFD is formatted exactly the same as a standard TIFF IFD. This tag is included in every TIFF IFD. Note that the size of the Extended IFD and data is variable and is not indicated anywhere.

The existence on this tag is the prime indicator that the file is a DEFF file as opposed to a straight TIFF file.

Image Description

Tag	0x010E
Type	ASCII
Length	1..96
Range	n/a
Required	

This tag stores the null-terminated ASCII string indicating an image description which may be used for the image title. Suggested content is patient ID followed by an image title. The separator should be a space character. This description may possibly be used for the window title in a windowing based system implementation.

Make

Tag	0x010F
Type	ASCII
Length	1..32
Range	n/a
Required	

This tag stores the null-terminated ASCII string indicating the manufacturer of the ultrasound or review station that last modified the image.

Model

Tag	0x0110
Type	ASCII
Length	1..32
Range	n/a
Required	

This tag stores the null-terminated ASCII string indicating the model of the ultrasound or review station that last modified the image.

Software

Tag	0x0131
Type	ASCII
Length	1..32
Range	n/a
Required	

This tag is required in DEFF files to store the null-terminated ASCII version number of the software in the ultrasound system or review station that last modified the image.

4.4.3 DEFF Specific Tags

These tags are contained only in a special Extended IFD pointed to by the ExtendedTagsOffset tag. The tag numbers are not official TIFF but do not need to be since no TIFF reader will encounter them. Thus, there will not be TIFF reader conflicts caused by these tags.

DEFF Subclass

Tag 0x401E
Type LONG
Length 1
Range 0..4
Required in DEFF 2.0 and later.
Optional otherwise
Default: 1 (Image)

Indicates the subclass to the DEFF Base Class which is contained in the DEFF file. This is the highest level tag indicating what is in the DEFF file. Since all DEFF files before version B4 were of the Image Class it is assumed that if the tag is not included in the DEFF file that the subclass is Image.

This tag is defined as:

Value	Subclass
0	None
1	Image
2	Research
3	Analysis
4	Audio
5	Experimental

Note that the Experimental subclass is to be used for vendor specific trials of new features and is not intended to be identifiable by other DEFF readers. Thus, any reader that encounters the use of the experimental subclass can abort the reading since the rest of the file is proprietary to the company that created the file.

DEFF Version

Tag 0x401C
Type LONG
Length 1
Range 0..(2**32)-1
Required in DEFF 2.0 and newer versions.

The DEFF Version tag contains a integral number which represents the version of the DEFF file.

Appendix A lists the version numbers for each version of changes to the DEFF specification. Note that the version number increments every time a new DEFF specification is produced regardless of whether or not that DEFF specification was ever released. This is expected to lead to rapid changes to the value in the DEFF Version tag.

Note that this tag was added at DEFF B2 and so older DEFF files will most likely not have this tag. Also note that the DEFF Version tag is really for informational purposes only and the preferred approach to determination of compatiblity is to check for the tags themselves as is done in TIFF.

See appendix A for the latest version but some old, possibly not ever used, versions are:

Version Number	Version Name	Description
1	A1	
2	A2	
3	A3	
4	A4	
5	A5	
6	A6	The officially released DEFF revision A specification.
7	B1	
8	B2	
9	B3	Document name change to Raster DEFF.
10	2.0	Public version. DEFF specification changed to class hierarchy.
11	2.1	Made compatible across industry.
12	2.2	More industry compatible. Use of single IFD.
12	2.3	Exact same data format as DEFF 2.2 so version number is same.
12	2.4	Exact same data format as DEFF 2.2 so version number is same.
12	2.5	Exact same data format as DEFF 2.2 so version number is same.

Patient Demographics

Tag 0x401F
Type LONG
Length 5
Range n/a
Optional No default

This tag contains a collection of patient demographic parameters in this order:

Age
The patient age at the time of the exam. Units of days.
Weight
The patient's weight. Units of grams.
Height
The patient's height. Units of cm. Same as patient length when the patient is an infant.
PD Spare 1
Currently undefined.
PD Spare 2
Currently undefined.

Patient/Exam Information (formerly Image Information)

Tag 0x400A
Type ASCII
Length 7 * 64
Range n/a
Required

This tag contains a collection of equal size strings in this order:

Patient ID (PID)
The Patient ID (PID) to which the image belongs.
Patient Last Name
Patient First Name
Patient Middle Name
Middle name may also be the middle initial. If it is just the initial, then a period must also be included. Middle name may also be blank or "NMI" (no middle initial).
Protocol Name
The null-terminated ASCII name of the protocol used to acquire the image.
Protocol Stage Name
The null-terminated ASCII name of the protocol stage in which the image was acquired (single-stage acquisitions typically have no stage name).
View Name
The null-terminated ASCII name of the diagnostic view, the protocol view, or some other image identifying title.

Original Machine

Tag	0x400E
Type	ASCII
Length	3 * 32
Range	n/a
Required	

This tag contains a collection of equal size strings in this order:

Make
: The null-terminated ASCII Manufacturer of the ultrasound or review station that originally produced the DEFF file.

Model
: The null-terminated ASCII Model of the ultrasound or review station that originally produced the DEFF file.

Software
: The null-terminated ASCII version number of the software in the ultrasound or review station that originally produced the DEFF file.

Source Machine

Tag	0x6002
Type	ASCII
Length	3 * 32
Range	n/a
Optional	

This tag contains information describing the ultrasound system that actually created the image. This may be different than the Original Machine. The Source Machine is the system that did the ultrasound scanning while the Original Machine is the one that converted the image into a DEFF file. These two machines are different when video frame grabbing or file format conversion is done.

This tag contains a collection of equal size strings in this order:

Make
: The null-terminated ASCII Manufacturer of the ultrasound machine that originally created the image.

Model
: The null-terminated ASCII Model of the ultrasound or review station that originally created the image.

Software
: The null-terminated ASCII version number of the software in the ultrasound or review station that originally created the image.

5.0 Image Class

5.1 Introduction

This section covers all DEFF files which contains images. However, the subclasses of the Image Class are also required to define which of the possible types of images that the DEFF file contains.

5.2 File Structure

Image DEFF Class files use the same structure as described in the DEFF base class specification.

5.3 DEFF Specific Concepts

5.3.1 Image Files

DEFF can be used for many different forms of files. The files do not have to contain images. The Image Class is concerned only with DEFF files which do contain images. The Image Class is further broken down into subclass depending on the type of image it contains. Some of the subclasses are Archive, Camera Ready, and Pictorial.

5.3.2 Image Aspect Ratio and Aspect Restoration

The term "aspect ratio" usually refers to the (screen or) image aspect ratio defined as:

$$\text{Image Aspect Ratio} = \frac{\text{horizontal image width}}{\text{vertical image height}}$$

Multiplying by one gives:

$$\text{Image Aspect Ratio} = \frac{\text{horiz image width}}{\text{vert image height}} \times \frac{\text{\# horiz pixels}}{\text{\# horiz pixels}} \times \frac{\text{\# vert pixels}}{\text{\# vert pixels}}$$

$$\text{Image Aspect Ratio} = \frac{\text{\# vert pixels}}{\text{vert image height}} \times \frac{\text{horiz image width}}{\text{\# horiz pixels}} \times \frac{\text{\# horiz pixels}}{\text{\# vert pixels}}$$

Substituting in the DEFF tags' names yields:

$$\text{Image Aspect Ratio} = \text{YResolution} \times \frac{1}{\text{XResolution}} \times \frac{\text{ImageWidth}}{\text{ImageLength}}$$

Also sometimes used is the pixel aspect ratio defined as:

Pixel Aspect Ratio = horizontal pixel width / vertical pixel height

Pixel Aspect Ratio = YResolution / XResolution

5.3.3 Compression

When used, image compression will be done on an image by image basis. Only the image data is compressed, not the header, IFDs, or non-image data.

The compression algorithms used are as specified in the TIFF standard. Both Packbits and LZW with horizontal difference preprocessing are supported.

5.3.4 Graphics

At the Image Class level, all graphics are assumed to be embedded in the image. This includes graphics associated with measurements, called tracings. A future subclass may possibly define graphics separately from the image.

5.3.5 Regions

It is necessary to know which part of an image is 2D data, Doppler data, or M-mode data, in order to perform some post processing on images. This is especially important when the images are acquired in dual or triple mode. Each part of an image which represents a different kind of data is referred to as a region.

In addition to knowing the location of the region, the transformation from pixel coordinates to physical coordinates is needed. This scaling information is used to perform real world measurements and analysis using the image data.

The Spatial Calibration tag provides the regions' locations and scaling information. It describes all the regions of an image and therefore the length of the tag is variable and depends on the NumberOfRegions tag.

Redundantly to the Spatial Calibration, the RegionArea and RegionScale tags also provide pixel to physical correlation. In new implementations of DEFF readers and writers, use of these tags is discouraged in favor of use of the Spatial Calibration tag.

Note that the scaling information taken from the Spatial Calibration tag or the Region Area/ Scale tags applies to the image after correction for the capture decimation (see Archive class) is applied and also after correction for the an interlaced image is applied.

5.4 Tag Descriptions

Tags defined in revision 5.0 of the TIFF standard will be used as defined. Extensions to the standard will be used when a tag does not exist, or the use of an existing tag would be non-standard.

5.4.1 Tag List

All numbers are in hexadecimal. The "Required" field indicates whether the tag is required for this particular class. If the field is "cond" then inclusion of the tag is dependent on some other tag's value.

IFD	Tag #	Type	Required	Description
TIFF	00FE	long	yes	New Subfile Type
TIFF	0100	long	yes	Image Width
TIFF	0101	long	yes	Image Length
TIFF	0102	short	yes	Bits Per Sample
TIFF	0103	short	yes	Compression
TIFF	0106	short	yes	Photometric Interpretation
TIFF	0111	long	yes	Strip Offsets
TIFF	0115	short	yes	Samples Per Pixel
TIFF	0116	long	yes	Rows Per Strip
TIFF	0117	long	yes	Strip Byte Counts
TIFF	011A	rational	yes	X Resolution
TIFF	011B	rational	yes	Y Resolution
TIFF	011C	short	yes	Planar Configuration
TIFF	0128	short	no	Resolution Unit
TIFF	0129	short	yes	Page Number
TIFF	013D	short	cond	Predictor
TIFF	0140	short	cond	Color Map
Ext	9101	short	no	Frame Interlace
Ext	9000	long	yes	Frame Strips
Ext	401D	short	yes	Image Subclass (formerly File Tag Level)
Ext	400C	short	cond	Number Of Regions
Ext	4011	short	*cond	Region Area
Ext	4012	rational	*cond	Region Scale
Ext	9002	long	cond	Spatial Calibration

* Note that these tags are not needed in DEFF 2.2 and newer because their functionality has been replaced by a new tag. See appendix for description of these tags.

Obsolete tags:

Ext	9001	Calibration Supported

Reserved Tags:

Ext	9003-90FF	Reserved for Image Class
Ext	9100-91FF	Reserved for Archive Class
Ext	9200-92FF	Reserved for Camera Ready Class
Ext	9300-93FF	Reserved for Pictorial Class
Ext	9400-9FFF	Reserved for future Image subclasses

5.4.2 TIFF IFD Tags

Bits Per Sample

Tag	0x0102
Type	SHORT
Length	== Samples Per Pixel
Range	8, 16
Required	

This tag record, if present, determines the number of bits representing each primary color component of a pixel.

Currently supported values are 8 and 16. However, TIFF readers will not be able to handle the DEFF usage of 16 bits per pixel. This is due to the use of a non-standard pixel type.

Color Map

Tag	0x0140
Type	SHORT
Length	(see text)
Range	0 .. 65535
Conditional	Required for palette images, Optional otherwise
Default	none

This tag record is required for 8 bit pixel types with Photometric Interpretation = 3 and for some more-than-8 bit types.

Length is 3 * 256 for 8 BitsPerSample types. Other pixel types will not necessarily have a length that is 3 * 2**BitsPerSample and the length must be taken from the tag's length field. This greatly reduces the color map size when, for example, a 16 bit pixel actually only uses 10 significant bits. Thus, when the BitsPerSample is not 8 bits, this file is not TIFF compatible.

All the red entries come first, then all the green, then all the blue. The pixel colors are stored left justified. 0,0,0 represents black. 65535, 65535, 65535 represents white.

This tag will not be included when the images are 24 bit RGB, or when the pixel type can make the palette values be implied.

Note that the "color" map may contain only grey values for a non-color image.

Compression

Tag 0x0103
Type SHORT
Length 1
Range 1, 5, 32773
Required

Value Description
1 Indicates that no data compression is used.
5 Indicates the use of the LZW compression algorithm as described in "Tag Image File Format Specification, Rev. 5.0 (Final)", Appendix F.
32773 Indicates the use of the Packbits compression algorithm as described in "Tag Image File Format Specification, Rev. 5.0 (Final)", Appendix C.

Image Length

Tag 0x0101
Type LONG
Length 1
Range 0..65535
Required

This record describes the length (Y dimension) of the number of pixels as actually stored in the file. (This is usually, but not necessarily, the same as the Y dimension of a correctly displayed image. There would be a difference if for example, only one field of a frame is stored in the file.)

Image Width

Tag 0x0100
Type LONG
Length 1
Range 0..65535
Required

This record describes the width (X dimension) of the image, in pixels.

New Subfile Type

Tag	0x00FE
Type	LONG
Length	1
Range	0
Required	

This tag must always have a value of zero. It is included because TIFF recommends that it be present.

Page Number

Tag	0x0129
Type	SHORT
Length	2
Range	(see text)
Required	

Frame Number

The current frame number. Because Page Number is a TIFF tag, Frame Number is only used to refer to the first frame. Thus, this value must always be zero.

Total Frames in File

The total number of image frames in the file.

This value is used to determine the number of frames in a loop and is therefore used as a multiplier for all tags whose length depends on the number of frames in the file.

Photometric Interpretation

Tag 0x0106
Type SHORT
Length 1
Range 1..3, 32820
Required

This tag record describes how to interpret the values in a pixel.

Value	Description
1	Gray. Minimum pixel sample value (0) is black
2	RGB. Minimum pixel sample value (0) is the minimum intensity for that primary color.
3	Palette. Value is described by a single sample. The sample is used as an index into the Palette. A palette may be used for a color or grey image.
32820	Other type of interpretation. Use an extended tag to determine the interpretation. This value is registered with Aldus.

This tag may be overridden by a pixel type tag contained in a subclass of the Image class.

Planar Configuration

Tag 0x011C
Type SHORT
Length 1
Range 1, 2
Required

Value	Description
1	The values for each pixel are stored contiguously (RGB of one pixel, followed by RGB of the next, etc.). Red is first (lowest address) followed by Green, then Blue.
2	The values for each pixel are stored in planes, with all the Red values for an image stored in one array, followed by the Green values in another, followed by the Blue values.

If the Samples Per Pixel tag is 1 (non RGB image), this tag is irrelevant, and TIFF says should not be present. However, to make other tag decoding easier, DEFF requires that this tag be present and that when SamplesPerPixel = 1, then PlanarConfiguration must be 1.

Predictor

Tag 0x013D
Type SHORT
Length 1
Range 1, 2
Conditional Required if Compression = LZW, optional otherwise.

Used when Compression == 5 (LZW).

Value	Description
1	Indicates no prediction scheme is used before coding.
2	Horizontal differencing prediction is used.

See "Tag Image File Format Specification, Rev. 5.0 (Final)", Appendices F and I.

It is expected that this tag value will always be 2 for LZW compressed images since it greatly reduces the compressed image size.

Resolution Unit

Tag 0x0128
Type SHORT
Length 1
Range 1..3
Optional Default = 2

This record controls the interpretation of the X resolution and Y resolution Tag Records.

Value	Description
1	No absolute unit of measurement
2	Inch
3	Centimeter

Rows Per Strip

Tag 0x0116
Type LONG (see text)
Length 1
Range 0..(2**32)-1
Required

The image data can be organized into strips. This tag contains the number of pixel rows per "strip" of image data.

DEFF requires the presence of this tag and its value must be equal to Image Length. Thus, all data is contained in the same strip (Unless PlanarConfig = 2).

Although the TIFF standard allows the value to be either SHORT or LONG, DEFF requires the use of LONG values exclusively.

Samples Per Pixel

Tag	0x0115
Type	SHORT
Length	1
Range	1..3
Required	

This tag record determines the number of primary color components of a pixel.

Strip Byte Counts

Tag	0x0117
Type	LONG
Length	SamplesPerPixel
Range	0..(2**32)-1
Required	

This tag contains the number of bytes for each strip of the first image's data. It is used to control the number of bytes fetched for each strip and is essential when the data is compressed. It is also required when the image is not compressed.

DEFF requires that RowsPerStrip = ImageLength. Thus, if PlanarConfiguration = 1, the length field is one, else if PlanarConfiguration = 2 the length field will be SamplesPerPixel.

Strip Offsets

Tag 0x0111
Type LONG
Length SamplesPerPixel
Range 0..(2**32)-1
Required

This tag contains the offset(s) to the first image contained in the DEFF file.

DEFF requires that RowsPerStrip = ImageLength. Thus, if PlanarConfiguration = 1, the length field is one, else if PlanarConfiguration = 2 the length field will be SamplesPerPixel.

Although the TIFF standard allows the value to be either SHORT or LONG, DEFF requires the use of LONG values exclusively.

X Resolution

Tag 0x011A
Type RATIONAL
Length 1
Range 1/((2**32)-1)..(2**32)-1
Required

This tag is used to indicate the pixels per Resolution Unit horizontally.

Y Resolution

Tag 0x011B
Type RATIONAL
Length 1
Range 1/((2**32)-1)..(2**32)-1
Required

This tag indicates the vertical pixels per Resolution Unit. When the image is interlaced, this tag indicates the Y Resolution of the field, not the frame. Thus, the field can be correctly displayed by TIFF readers.

5.4.3 Extended IFD Tags

Extended IFD tags are contained only in a special Extended IFD pointed to by the ExtendedTagsOffset tag.

Frame Interlace

Tag 0x9101
Type SHORT
Length 1
Range 0..1
Optional Default = 0, non-interlaced

This tag indicates whether the file contains images which are stored in a non-interlaced or interlaced format. The value of this field affects the content of the Frame Strips tag. A non-interlaced file is one where each row of an image is store sequentially in the order of 0, 1, 2, 3 ... An interlaced file is one which is stored every other row (even rows) followed by the other alternate rows (odd rows) as in 0, 2, 4, ... 1, 3, 5,...

Value	Description
0	Non-interlaced
1	Interlaced

Note that if the image is stored in interlaced format, some tag values refer to the field while others refer to the frame. The table below indicates the whether a tag's value refers to the field or the frame:

Tag	Non-interlaced	Interlaced
all scaling tags	frame	frame
CaptureDecimation	frame	frame
Frame Strips	frame	field
ImageLength	frame	field
PageNumber	frame	frame
PlanarConfiguration	frame	field Note 1
RowsPerStrip	frame	field
StripByteCounts	frame	field Note 2
StripOffsets	frame	field Note 2
X Resolution	frame	frame
Y Resolution	frame	field

Note 1: If the frames are interlaced and PlanarConfiguration = 2, then the image storage occurs in the order of all field 0 planes followed by all field 1 planes.

Note 2: If the frames are interlaced and PlanarConfiguration = 2, then the length of these fields will be SamplesPerPixel and the combination of the values will refer to a field.

Frame Strips

Tag 0x9000
Type LONG
Length 2 * NumberOfPlanes * TotalFields * TotalFramesInFile
Range $0..2^{32}-1$
Required

This tag contains a series of two word "sets". Each set contains a location and size of one pixel data strip. The two word set consists of:

Strip Offset
: Byte offset to the frame data. (Note: The first set will have this value equal to the offset in the StripOffsets tag.)

Strip ByteCount
: The number of bytes in the strip. (Note: The set will have this value equal to the byte count in the Strip Byte Counts tag.)

To calculate the length of this FrameStrips tag the formula is:

TagLength = 2 * NumberOfPlanes * TotalFields * TotalFramesInFile

The NumberOfPlanes is found from:
if PlanarConfig = 1
then NumberOfPlanes = 1
if PlanarConfig = 2
then NumberOfPlanes = SamplesPerPixel
(Note: This works because DEFF requires that RowsPerStrip = ImageLength.)

The TotalFields is found from:
if FrameInterlace = non-interlaced
then TotalFields = 1
if FrameInterlace = interlaced
then TotalFields = 2

TotalFramesInFile is taken from the Page Number tag.

The sequential ordering of the sets follows the formula:

```
for frame = 0 to TotalFramesInFile-1
    for field = 0 to TotalFields-1
        for plane = 0 to NumberOfPlanes-1
            StripOffset
            StripByteCount
```

Some implicit ordering of the sets is required:

Planes: For PlanarConfig = 2, and SamplesPerPixel = 3, the red plane is followed by the green plane which is followed by the blue plane. For PlanarConfig = 2, and SamplesPerPixel = 2, the least significant byte's plane is followed by the most significant byte's plane.

Fields: For any given frame, the even field's set(s) will always occur before the odd field's set.

Frames: The frames' sets will always occur in sequential order. So frame 0 is followed by frame 1 followed by frame 2 and so on.

Note that it is possible to have successive strip offsets to point to the same field/frame strip data. This allows for simple duplication of fields and frames without requiring the redundant storage.

Image Subclass (formerly File Tag Level)

Tag 0x401D
Type SHORT
Length 1
Range 0..4
Required in DEFF B4 (pre 2.0) and later
Optional otherwise
Default 0 (Archive)

This tag indicates what Image Class' subclass is present in the file. In general, this tag is used to determine which other tags to expect to be contained in the DEFF file.

The defined values are:

Value	Meaning
0	Archive (formerly "Normal" and "Raster")
1	Camera Ready
2	Import / Export (not currently used)
3	Pictorial
4	None

Number Of Regions

Tag 0x400C
Type SHORT
Length 1
Range n/a
Conditional Required if Spatial Calibration tag is present.
Default Zero regions if tag is not present.

This record gives the number of distinct scaling regions in this image loop.

A value of 0 indicates that no region information is provided.

Spatial Calibration

Tag 0x9002
Type LONG
Length 31 * NumberOfRegions
Range N/A
Conditional In DEFF 2.2 and newer, tag is required if NumberOfRegions > 0

This tag is used to specify the type, location, and scaling information of various data regions within the image contained in this file. The data organization consists of all the parameters for region 0, followed by all the parameters for region 1, etc. until the NumberOfRegions is reached. The parameters of a single region are listed below.

Note that the scaling information taken from this tag applies to the image after correction for the capture decimation is applied and also after correction for the an interlaced image is applied.

Also note that it is expected that in a future version of DEFF that the reference pixel location will be defined for various "standard" types of regions.

Some of the parameters are of RATIONAL type so the numerator is stored as a LONG followed by the denominator stored as a second LONG. Note that in order to avoid roundoff errors DEFF readers should promote these values to floating point and then perform the divide to create a floating point numerator/denominator value.

Region Spatial Format

Enumerated value indicating the spatial organization of the data within the region.

Value	Meaning
0	None or not applicable
1	2D (echo or flow)
2	M-mode (echo or flow)
3	Spectral (CW or PW Doppler, etc.)
4	Waveform (physio, Doppler, etc.)
5	Graphics
6	Steered 2D
8000-FFFF	Vendor specific experimental regions

Region Data Type

Enumerated value indicating the type of data within the region.

Value	Meaning
0	None or not applicable
1	Echo
2	Velocity
3	Variance
4	Flow intensity (power)
5	Computed border
6	Doppler Mean
7	Doppler Mode
8	Doppler Max
9	ECG
10	Pulse
11	Phono
12	Gray bar
13	Color bar
8000-FFFF	Vendor specific experimental data types

Region Flags

Flags used for special handling of the region.

Bit 0 (LSB) Transparency.

1 = Transparent, 0 = Opaque If the region is transparent, then measurements may be done on regions underneath this region. This is most useful for ECG overlays. Default = 0.

Bit 1 Scaling protection.

If the region is protected, the region cannot be manually scaled. That is, the data in the Region Scale tag record that corresponds to the current region cannot be changed. This feature will be used with images captured by a review station using hidden digital and by other images where scaling is computed by the respective system. The protection bit is used to tell the system that the scaling for that area was computed internally to the machine (i.e.: hidden digital or digital capture) and the user cannot edit that information. In a manually scaled image (i.e.: from a competitor's video tape), it is possible to edit the scaling done for that image.1 = Protected, 0 = Not Protected. Default = 0. The ultrasound machine that produces a DEFF file should always set this bit to a 1.

Bit 2 Doppler Scale Type.

Valid for PW and CW Doppler regions only. Indicates which type of scale marking is used in the region. 0 = velocity, 1 = frequency, 0 is default.

Bits 3-31 Reserved for future use.

Must be set to zero by a DEFF writer, and ignored by a DEFF reader.

Region Location Minimum X
Region Location Minimum Y
Region Location Minimum Z
Region Location Maximum X
Region Location Maximum Y
Region Location Maximum Z

The bounds of a rectangular volume specifying the location of the region. The upper left corner of the image is X = 0, Y = 0, and the lower right corner is X = ImageWidth-1, Y = ImageLength-1. Z is the frame number. Thus, a region will be specified as within the bounds of (0, 0, 0) to (ImageWidth-1, ImageLength-1, TotalFramesInFile-1). Note that because Z is the frame number, the physical units of Z may be either temporal (i.e. time) or spatial (i.e. volume).

Reference Pixel Location X
Reference Pixel Location Y
Reference Pixel Location Z

These values define the location of a virtual "reference" pixel. The reference pixel is used to tie the image's pixel coordinate system to the physical coordinate system. For example, the reference pixel could define where a depth of zero centimeters occurs in a 2D image, or it could define where the baseline (i.e. zero frequency) resides in a spectral display. The Reference Pixel Location uses the same coordinates as used in Region Location but the location is not required to be within the region or even within the image boundary or loop. For this reason, the Reference Pixel Location can be either positive or negative. Thus, Reference Pixel Location X, Y, and Z are 2's complement signed LONG values.

Physical Units In X Direction
Physical Units In Y Direction
Physical Units In Z Direction

Enumerated values indicating the physical units of the three dimensions of the region. Each of the X, Y, and Z values uses the encoding as follows:

Value	Meaning
0	None or not applicable
1	Percent
2	dB
3	cm
4	seconds
5	hertz ($seconds^{-1}$)
6	dB/second
7	cm/second
8	cm^2
9	cm^2/second
10	cm^3
11	cm^3/second
12	degrees

Reference Pixel Physical X Numerator and Denominator
Reference Pixel Physical Y Numerator and Denominator
Reference Pixel Physical Z Numerator and Denominator

The physical value at the reference pixel location. These fields are signed RATIONAL. The units are as specified in the PhysicalUnits fields.

Physical Delta X Numerator and Denominator
Physical Delta Y Numerator and Denominator
Physical Delta Z Numerator and Denominator

The physical value increments per positive pixel increment (or positive frame increment in the Z direction case). These fields are signed RATIONAL. The units are as specified in the PhysicalUnits fields.

Transducer Frequency

The transducer excitation center frequency actually used to collect the data in the region. Units of Hertz.

Pulse Repetition Frequency

The ultrasound PRF used to collect the data in the region. Units of hertz.

Region Specific Data Numerator and Denominator

This word allows for a parameter to be stored which is dependent on the type of region.

Exact specification is TBD. Anticipated use includes:

Region Type	Region Specific Data
Doppler	Doppler Correction Angle
Steered 2D	Steering Angle
All others	Reserved

6.0 Camera Ready Class

6.1 Introduction

This section describes a class of DEFF designed to promote the compatibility of image files between ultrasound systems and cameras, and PCs.

Camera Ready DEFF files are compatible with Tag Image File Format (TIFF) files.

6.2 File Structure

The Camera Ready Class uses the same file structure as the DEFF base class. The tags contained in the Extended IFD are necessary on writing but not required to read the file. Thus, camera ready DEFF files are compatible with software which can read TIFF images.

6.3 DEFF Specific Concepts

6.3.1 Camera Ready Files

It is desirable to be able to save a single frame from an ultrasound display to a DEFF file and to have the file format support either a review station or a camera. Unfortunately, this is not always possible due to conflicting requirements of flexibility vs. simplicity. When an image is in DEFF file which is expected to be compatible with a camera, the DEFF file is referred to as "camera ready."

A camera ready file has certain characteristics which are a subset of the DEFF specification. The most notable characteristics are:

- Single image, not loops.
- Pixel format is 8 bit gray, 8 bit palette, or 24 bit RGB.
- All graphics are embedded in the pixel data, no graphics lists.
- DEFF specific tags are not needed to be read to reproduce an image

6.4 Tag Descriptions

Tags already defined in revision 5.0 of the TIFF standard are used as defined.

6.4.1 Tag List

All numbers are in hexadecimal. The "Required" field indicates whether the tag is required for this particular class. If the field is "cond" then inclusion of the tag is dependent on some other tag's value.

IFD	Tag #	Type	Required	Description
Ext	9200	long	no	Camera Ready Version

Reserved Tags:

Ext	9201-92FF	Reserved for Camera Ready Class

6.4.2 Tag Restrictions

The Camera Ready Class restricts the usage of several tags to a subset of their defined values. This makes Camera Ready Class images less variable and eases implementation. This section defines the restrictions to the Camera Ready Class' superclass tags.

Bits Per Sample
This tag record determines the number of bits representing each primary color component of a pixel.
Must be: 8

Compression
Must be: 1 indicating that no data compression is used.

Frame Strips
Must only use the frame offsets, not fields.

Frame Interlace
Must be zero, non-interlaced. (This is an optional tag with a zero default.)

Page Number
This tag is used to specify the current frame number (the first value), and the total number of frames in the file (the second value).
Frame number must be: 0
Total frames must be: 1

Photometric Interpretation
This tag record describes how to interpret the values in a pixel.
Must be one of:

Value	Description
1	Gray
2	RGB
3	Palette color

Samples Per Pixel
This tag record determines the number of primary color components of a pixel.
Must be either: 1 for Gray scale or palette images (default)
3 for RGB images

6.4.3 Tag Definitions

Camera Ready Version

Tag	0x9200
Type	LONG
Length	1
Range	0..(2**32)-1
Optional	Default 1, DEFF 2.0 compatible

The Camera Ready Version tag contains an integral number which represents compatibility with the ability to display or print this Camera Ready Class DEFF file.

The version number increments only when changes to this class has rendered inoperable the ability to print or display the image using software which was only compatible up to the last version. This is expected to occur very infrequently.

Note that this tag was added at DEFF version 2.2 thus older DEFF files will not have this tag.

Version Number	DEFF Spec	Reason for change
1	2.0	Original Camera Ready Class.
2	2.2	Allowance for PlanarConfig = 2. Image Description tag length increased.

7.0 Archive Class

7.1 Introduction

This section describes the specification of the Archive DEFF Class tags only.

7.2 File Structure

Archive DEFF Class files use the same structure as described in the DEFF specification.

7.3 DEFF Specific Concepts

7.3.1 Multiple Images in One DEFF File

A typical Archive Class DEFF file will contain one or more frames. When either a single frame or a multiple frame loop is stored, all tags are contained in the first TIFF IFD and Extended IFD, followed by all the image and other data.

When an acquisition does not consist of continuous frames, for example multiple beats consisting of only 8 frames per beat, then each of the "beats" is stored in a separate DEFF file.

7.3.2 Image Aspect Ratio and Aspect Restoration

The Archive Class redefines the aspect ratio from the definition provided in the Image Class. This is necessary due to the ability to have capture decimation as defined below.

The tags X Resolution and Y Resolution indicate the number of pixels per inch (or per Resolution Unit) in the DEFF file image. Note that the DEFF image aspect is not necessarily the same as the capture screen aspect due to decimations and other conversions occurring during the DEFF file writing process.

The X and Y Resolution tags can be used to correct differences between DEFF image aspect and review screen aspect. The DEFF image aspect ratio is simply:

$$\text{Image Aspect Ratio} = \frac{\text{horizontal image width}}{\text{vertical image height}}$$

Substituting in the DEFF tags' names yields:

$$\text{Image Aspect Ratio} = \frac{\text{YResolution}}{\text{XResolution}} \times \frac{\text{ImageWidth}}{\text{ImageLength}}$$

Also included in the DEFF file is the tag, Capture Decimation, which includes both the X and Y decimations used during image capture. This tag is provided *for convenience only* and is not required to be used to convert the DEFF image aspect to the review system aspect. Capture Decimation allows the DEFF file reader to determine if simple pixel decimation occurred during the image capture process. This tag indicates the capture method as follows:

Resolution Terminology	Capture Decimation X	Capture Decimation Y
High Resolution	1	1
Medium Resolution	1	2
Low Resolution	2	2

The original screen size can be found by:

X screen = XResolution x XDecimation
Y screen = YResolution x YDecimation

Note that the scaling information taken from the Spatial Calibration tag or the Region Area/ Scale tags applies to the image after correction for the capture decimation is applied and also after correction for the an interlaced image is applied.

7.3.3 Capture Window Coordinates

For any image that was captured at the full size of the original system's display, this discussion is not really relevant.

This discussion assumes that the image aspect restoration described above has taken place.

There are three rectangular areas of pixels: Original Screen, Restored Image, and Replay Window. All three areas have their origin at the upper left and the X dimension increasing horizontally to the right and the Y dimension increasing vertically down. They all have the same units: pixels of the original screen. See figure 3.8-1.

The Original Screen is the display of the system that captured the image.

The Restored Image is the DEFF file image after any Capture Decimation has been "undone."

The Replay Window is the portion of the Restored Image that is most desirable to be shown when screen space is limited.

To translate images between these three rectangular areas, the values of Replay Window X and Y, and Replay Window Pan X and Y are defined.

Replay Window X and Y indicate the offset from the Original Screen origin to the Replay Window origin.

Replay Window Pan X and Y indicate the offset from the Restored Image origin to the Replay Window origin.

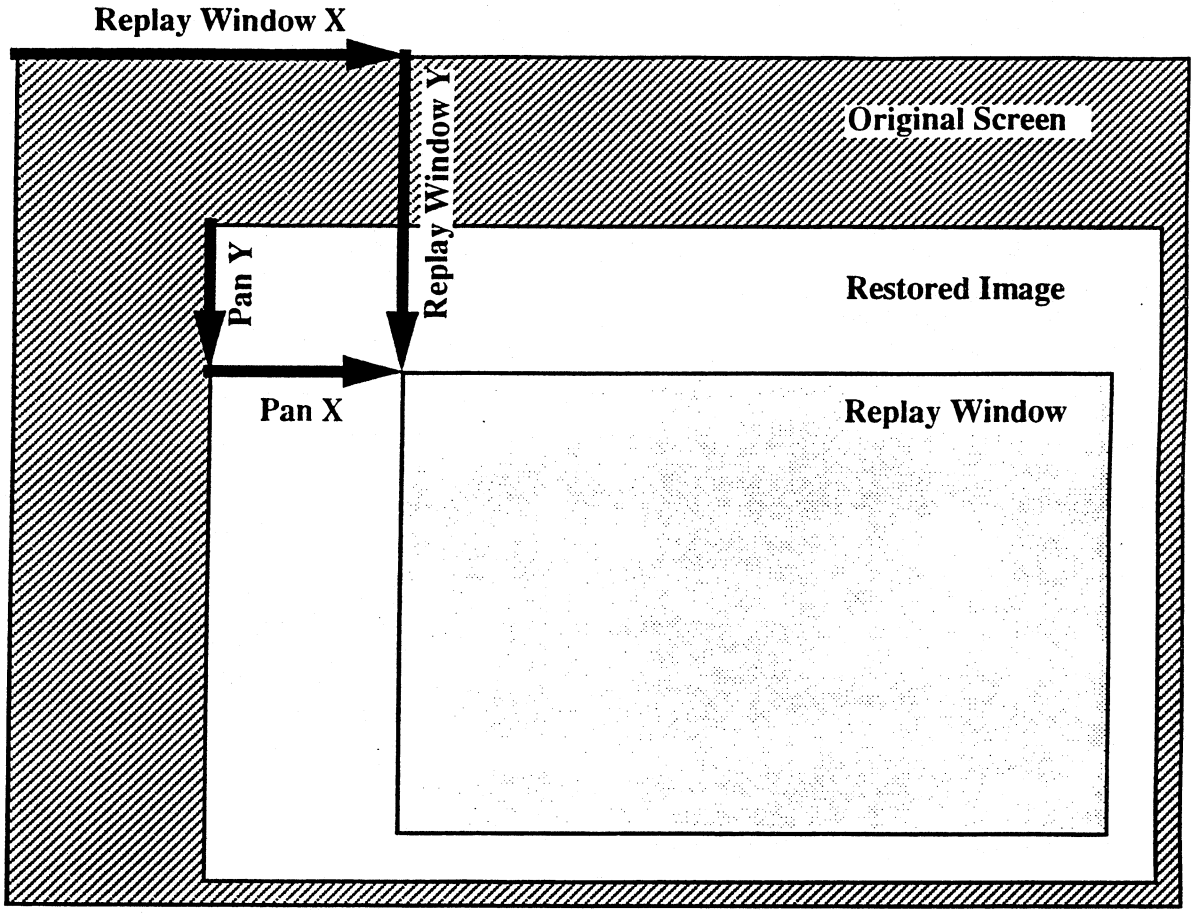


Figure 7.3.3-1 Area Translations

The offset from the Original Screen origin to the Restored Image Origin is found by:

Restored Image Origin(X,Y) =
Replay Window (X,Y) - Replay Window Pan (X,Y)

This seems like an unusual approach but it allows the DEFF file X and Y values to always be positive.

The size of the Original Screen is not indicated in the DEFF file, it is not needed.

The size of the Restored Image is:

Restored Image X = ImageWidth * Capture Decimation X
Restored Image Y = ImageLength * Capture Decimation Y

The size of the Replay Window is indicated in the Replay Area tag as Replay Window Size X and Y.

7.3.4 Time Synchronization

There are many parameters included in the DEFF files which allow for the replay of a cineloop at the same rate as the capture. Some parameters also help to synchronize loops of differing number of frames and/or acquisition duration.

The Page Number tag contains the actual number of frames in the DEFF file. This will be referred to as N frames.

The Trigger Time A value is the time from the ECG R-wave to the start of the Capture Initial Delay. The Trigger Time B value is measured from the R-wave as well but is currently not used.

The Capture Initial Delay is the time from the acquisition trigger to the start of the frame 0. To get the time from the R-wave to the start of frame 0, the Trigger A value must be added to the Capture Initial Delay value. All other timing is referenced from the start of frame 0 except for the Timers' values.

The Number of Timers tag indicates the number of application timers running during the acquisition. The Application Timers tag gives their values. These are long term timers started and stopped by the user. The timer values do not have any specific meaning associated with each timer. They may be used for such things as timing the start of exercise to the peak of exercise, or from the peak exercise to the completion of acquisition of all images. Different stages of the protocol may use different timers.

In DEFF 2.2 and newer:

The Frame Timing tag is used to determine the capture and replay timing of each individual frame. Use of the Frame Timing tag is recommended over use of Capture Period and Playback Period tags.

In DEFF 2.1 and older:

For each frame J in the cineloop, where $0 <= J < N$, there will be a frame time tag associated with that frame.

Frame Time Tag (J) = Frame0Start to FrameJEnd

The capture period is given by the Capture Period tag which is

Capture Period = Interim Delay * Number of Frames

where Interim Delay is the time between the start of Frame A and the start of Frame A+1 and the Number of Frames is the number of frames in one Triggered Acquisition. Thus an 8 frame, 50ms/frame loop would have a capture period value of 400ms. (The user will see this as a standard 350ms acquisition).

The playback period is set by the user as part of a protocol. Playback Period = UserDesiredDelayBetweenReplayFrames * M
where M is the number of frames in one Triggered Acquisition.

7.3.5 Scrolling / Sweeping Data Capture

Presently, scrolling M-mode and Doppler data will be captured as 2D raster frames. The frames will be a loop of up to 20 seconds in which each frame acquisition is spaced at about 1 second intervals. This approach requires no special tags or data formats specific to the scrolling data storage.

When the raster image contains sweeping data there will be a small gap in the data where old data is removed and new data will be drawn. Measurements should not take place across the gap but are usually not prevented by the displaying system.

7.3.6 ECG Trace

The storage approach for ECG waveforms is under development. Thus, this desciption is only one possible implementation at this time.

Some image loops will contain a separate ECG waveform trace. In these cases, the replaying system will have the choice of whether or not to overlay the ECG trace upon the images.

This form of ECG trace is stationary and does not scroll or sweep. The color of the ECG trace is defined by the ECG Trace Color tag. The waveform is specified by an one dimensional array of y points in the ECG Trace Points tag. The x dimension is assumed to increment with each point.

The entire waveform fits within the Region Area defined for the ECG trace. Region Area defines both the size and position of the window. The first ECG point is at the left edge of the region area window and the last point is at the right edge of the window. A vertical height of zero indicates the trace is at the bottom of the window while a vertical height of 63 indicates that the trace is 63 pixels above the bottom of the window. The right edge of the ECG waveform corresponds to the last image captured.

When the R-wave occurs, a counter starts to run. When the counter reaches the Trigger Time A value, a flag is set in the ECG Trace Point data indicating this position. The counter is cleared and continues to run. When the counter reaches Capture Initial Delay, the capture of Frame 0 begins. Thus to determine the start of Frame 0 from the R-wave, Trigger Time A must be added to Capture Initial Delay.

The Region Scale tag indicates the seconds per undecimated pixels of the ECG waveform.

7.3.7 Graphics

No low level graphics drawing tags (e.g. point, line, circle, etc.) are defined at present. It is yet to be determined if these primitives will be required.

Most graphics are assumed to be embedded in the image. The values of items embedded in the image are often specified in separate tags. For instance, patient name is both embedded in the image as pixels and included in the Image Information tag as a text string.

The ECG trace described above requires processing by the image displaying system to overlay it on top of an image.

7.4 Tag Descriptions

7.4.1 Tag List

The tags listed here are for the Archive DEFF Class only. All tags included in superclasses to the Archive DEFF Class must also be included in Archive DEFF files. All numbers are in hexadecimal. The "Required" field indicates whether the tag is required for this particular class. If the field is "cond" then inclusion of the tag is dependent on some other tag's value.

IFD	Tag #	Type	Required	Description
TIFF	8440	short	yes	Archive Pixel Type (formerly Image Pixel Type & Raster Pixel Type)
TIFF	8441	short	*no	Palette Ranges
Ext	9102	long	cond	Application Timers
Ext	4001	ascii no	Audio	Comment File
Ext	4002	short	no	Capture Decimation
Ext	4003	rational	no	Capture Initial Delay
Ext	4004	rational	*no	Capture Period
Ext	6003	ascii	no	Color Map Identifier
Ext	9100	short	cond	Compressed Color Map
Ext	401B	byte	no	ECG Trace Color
Ext	401A	byte	*no	ECG Trace Points
Ext	9104	long	no	Frame Timing
Ext	4018	short	no	Heart Information
Ext	4017	short	no	Loop Graphics' Attributes
Ext	400D	short	yes	Number of Timers
Ext	6001	short	no	Organ Scan
Ext	400F	rational	*no	Playback Period
Ext	4010	short	no	Primary Frame
Ext	4007	short	yes	Protocol Information (File Identification)
Ext	4013	short	no	Replay Area
Ext	4014	short	no	Selected Beat
Ext	4019	rational	no	Trigger Times
Ext	4016	short	no	Trim Points

* Note that these tags are not needed in DEFF 2.2 and newer because their functionality has been or will be replaced by a new tag. See appendix for description of these tags.

Reserved Tags:

Ext	9105-91FF	Reserved for Archive Class

Obsolete Tags:

Ext	4000	Audio Frame Comment
Ext	4005	Debug Frame Comment
Ext	4006	Debug Loop Comment
Ext	4008	Frame Time Tag
Ext	4009	Heart Rate. Replaced by tag 4018, Heart Information.
Ext	400B	Number of Triggered Acquisitions
Ext	4015	Timers
Ext	9103	Timing Supported

7.4.2 Tag Restrictions

Photometric Interpretation

This Image Class tag is overridden by the Archive Class' Archive Pixel Type tag.

7.4.3 TIFF IFD Tags

Archive Pixel Type (formerly Image Pixel Type)

Tag 0x8440
Type SHORT
Length 1
Range 1..7
Required

This record specifies the kind of data contained in this file. Legal values are:

Value Description

1 Gray 8 bit raster data. 0 = black, 255 = white.
2 RGB 8 bit per primary raster data. 0,0,0 = black, 255, 255, 255 = white.
3 Pre-lookup 8 bit raster data (palette color). Requires ColorMap tag to be present.

4 Mixed RGB/gray raster data. Packed into one 16 bit word per pixel. Pixels vary on a pixel by pixel basis and are either MSB = 1, then 5 bit red, 5 bit green, and 5 bit blue or MSB = 0 next 7 don't cares and 8 bits gray. To be compatible with an UM9 HDI system, the don't cares must be set to 0100110 binary. The 16 bits are stored 8 bit Least Significant Byte first and then 8 bit Most Significant Byte second, regardless of the overall file's byte order. Used for VCR playback captured images. This pixel format will not have a palette. It does have the following values for other tags:
Photometric interpretation: 32820 ("other")
Samples per pixel: 1
Bits per sample: 16

5 Pre-lookup 11 bit raster data. Stored as one 16 bit word per pixel with the 5 MSBs zero. Requires ColorMap and PaletteRanges tags to be present. ColorMap will contain 3 times 2**11 entries. BitsPerSample will be 16. PaletteRanges will allow determination of the type of data represented by the 11 bit index. This format is for pre-look up table color with full variance. It also allows for future expansion of echo and frequency field sizes. Restricting to 11 bits instead of 16 allows for a much smaller palette size which is especially important for single image files.

6 Pre-lookup 16 bit raster data. Stored as one 16 bit word per pixel. Requires ColorMap tag to be present. ColorMap will contain 3 times 2**16 entries. BitsPerSample will be 16.

7 Pre-lookup 16 bit raster data. Requires use of the CompressedColormap. SamplesPerPixel=2, BitsPerSample=8,8 (length=2), and PlanarConfig=2. Each 16 bit pixel is made up of two samples (Thus, SamplesPerPixel=2). All of a frame's first (LS byte) samples are stored in the first 8 bit plane followed by all of the second (MS byte) samples in the second 8 bit plane (Thus, PlanarConfig=2). Combining the two samples into a 16 bit word provides an index into the compressed colormap.

100 Non-pixel data. Now obsolete.
Note that values of 1, 2, and 3 are similar to the Image Class' Photometric Interpretation tag.

7.4.4 Extended IFD Tags

These tags are contained only in a special Extended IFD pointed to by the ExtendedTagsOffset tag. The tag numbers are not official TIFF but do not need to be since no TIFF reader will encounter them. Thus, there will not be TIFF reader conflicts caused by these tags.

Application Timers

Tag	0x9102
Type	LONG
Length	NumberOfTimers
Range	0..(2**32)-1
Conditional	Required if Number of Timers != 0
	Optional if Number of Timers = 0

This tag gives the values associated with each of the application timers which were visible when this loop was acquired. Units of milliseconds.

Audio Comment File

Tag	0x4001
Type	ASCII
Length	1..32
Range	n/a
Optional	
Default	n/a

This tag, if present, is used to attach an audio comment to the image represented by the DEFF file. The argument is the null-terminated filename of an audio data type DEFF file, assumed to be present on the same device and in the same directory as the current DEFF file.

Capture Decimation

Tag	0x4002
Type	SHORT
Length	2
Range	1,2
Optional	
Default	1

This tag gives the operator-selected decimation factor applied upon capture to the image contained in this DEFF file.

The first number is the horizontal decimation and the second number is the vertical decimation.

Note that this tag is supplied for convenience only and that the X Resolution and Y Resolution tags are independent of this tag's values.

Capture Initial Delay

Tag 0x4003
Type RATIONAL
Length 1
Range 1/((2**32)-1)..(2**32)-1
Optional
Default 0/1

This tag gives the programmed delay from acquisition trigger to the capture of the first frame. It is carried for informational purposes, since it does not affect display of the image loop. The units of measure are seconds.

Color Map Identifier

Tag 0x6003
Type ASCII
Length 41 maximum
Range N/A
Optional

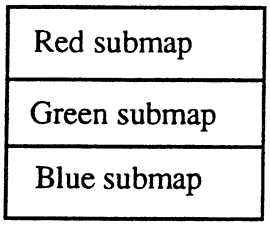
A variable length ASCII string defining the type of colormap used. This string is intended to help avoid loading a colormap which has been already recalled into a reviewing system.

Compressed Color Map

Tag 0x9100
Type SHORT
Length variable
Range 0 .. 65535
Conditional Required for images requiring a compressed palette. Optional otherwise.
Default none

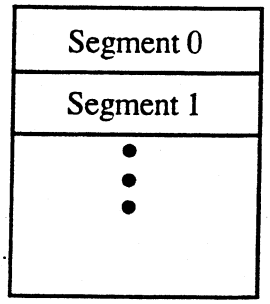
This tag contains a compressed version of the color map necessary to render the image into red, green, and blue components of pixels. In its uncompressed state, the colormap will have $2^{(SamplesPerPixel * BitsPerSample)}$ RGB tuples. Each RGB tuple consists of a 16 bit red component, a 16 bit green component, and a 16 bit blue component. The pixel components are stored left justified. A RGB of 0,0,0 represents black. 65535, 65535, 65535 represents white.

A compressed colormap is divided into a red submap followed by a green submap followed by a blue submap as in:



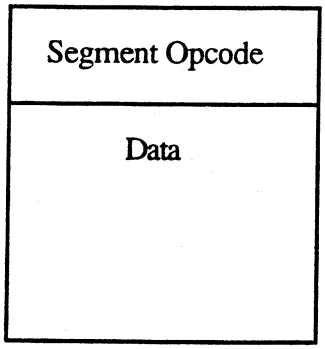
Compressed Color Map

Each submap consists of a series of segments as in:



Submap

Each segment consists of an opcode followed by some data as in:

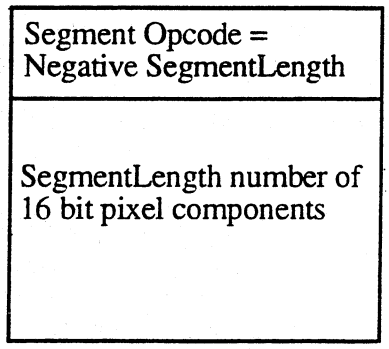


Segment

There are three types of segments: discrete, linear, and indirect. The segment type is identified by the 2's complement value of the opcode as follows:

Opcode	Segment Type
< 0	discrete segment
> 0	linear segment
0	indirect segment

The discrete segment is used to represent a series of palette components which are not monotonic with respect to their predecessors or successors. The absolute value of the opcode is SegmentLength indicating the number of 16 bit pixel components following the opcode. The discrete segment's format is thus:

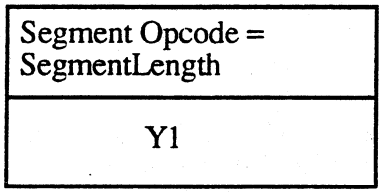


Discrete Segment

The linear segment represents a series of palette components whose values may be represented by a straight line.

X = palette address, Y = value contained in the palette
(X_0,Y_0) = end of the previous segment
(X_0+SegmentLength, Y_1) = end of this linear segment
Where: SegmentLength is passed in the segment opcode.
Y_1 is contained in the data portion of this segment.

During decompression, the application should "connect" the previous segment's endpoint, (X_0,Y0), with this segment's endpoint, (X_0+SegmentLength, Y_1), using a straight line, by computing the values for each point between the endpoints. Note that because the linear segment uses the end point from the previous segment, a linear segment cannot be the first segment in a submap. The linear segment's format is:



Linear Segment

The indirect segment allows the re-use of repetitive regions within a submap without re-specifying the pixel component values. The opcode is zero followed by the number of segments to copy and one offset pointer to the first segment to copy. The byte offset is relative to the beginning of the submap in which the indirect segment occurs. For example, if an green submap's indirect segment offset wants to point to the first segment in the submap, then the offset will be zero. The offset is a 32 bit value but is stored in the segment as a least significant 16 bit value followed by a most significant 16 bit value. Note that an indirect segment is not allowed to point to or copy another indirect segment. This avoids need for recursion and also avoids possibilities of infinite loops. The format of an indirect segment is:

Segment Opcode = Zero
Number of segments to copy
Least significant 16 bits of byte offset to first segment to copy
Most significant 16 bits of byte offset to first segment to copy

Indirect Segment

ECG Trace Color

Tag 0x401B
Type BYTE
Length 3
Range 0..255
Optional Default = 255, 255, 255

The storage approach for ECG waveforms is under development. Thus, this description is only one possible implementation at this time.

This tag contains the RGB color to be used when drawing an ECG waveform.
The byte order is Red, Green, Blue. 0 is no intensity, 255 is full intensity.

Frame Timing

Tag	0x9104
Type	LONG
Length	2 * TotalFramesInFile
Range	N/A
Optional	Default 0, no timing information available

TotalFramesInFile is taken from the PageNumber tag. The repeated words associated with each frame are:

Capture FrameDuration
: The time from the start of acquisition of this frame to the start of acquisition of the next frame. For the last frame in a loop, the time is from the start of acquisition of this frame to the when the start of the next frame would have occurred. Units of microseconds.

Playback Frame Duration
: The time from the start of display of this frame to the start of display of the next frame. For the last frame in a loop, the time is from the start of display of this frame to the when the start of display of the first frame should occur. Units of microseconds. This value is possibly different than the Capture Frame Duration if, for example, the user desired a replay speed slower than the capture speed.

A 32 bit unsigned value with units of microseconds can represent a maximum duration of 71.6 minutes which is longer that any conceivable need.

Note that it is legal for a loop of length one and thus this tag may be included but is not very useful.

Heart Information

Tag	0x4018
Type	SHORT
Length	4
Range	0..65535
Optional	
Default	0, 0, 0, 0

This tag gives the four values for the entire loop which are associated with the heart:

Heart Rate
: This is the patient's actual heart rate during the cineloop capture. A value of 0 indicates no reliable heart rate exists on the system. Units are in beats / second.

Percent of Maximum Heart Rate
: This is the actual heart rate divided by the theoretical maximum heart rate of this patient. Units are tenths of a percent. For example, if the heart rate is 70 and the maximum heart rate is 100, then this value is 700.

Blood Pressure at Systole
: This is the "top" blood pressure number.

Blood Pressure at Diastole
: This is the "bottom" blood pressure number.

Loop Graphics' Attributes

Tag 0x4017
Type SHORT
Length 96 (4 * 24)
Range n/a
Optional

This tag is used to indicate various display attributes of each of several graphics items which may appear as review station overlays on an image. The original acquisition system will not likely use this tag. The value of the graphic item is not included here but rather in various other tags.

There are 16 graphic items identified plus 8 spare items in this tag. Each graphic item has several attributes. All the attributes of one graphics item will occur before the next graphic item.

The attributes are:

Flags
Bit 0 indicates if the graphic item is present in the image. 1 = present.
Bit 1 indicates whether it is possible to disable (not display) the item. 1 = possible to disable
Bits 2 and 3 indicate a Corner Relation.
This is an enumerated type value indicating how to use the Location X and Location Y values. The X and Y values will be measured from one corner of the image to the same corner of a box surrounding the graphic item.

Bit 3	Bit 2	Meaning
0	0	use the upper left corners
0	1	use the upper right corners.
1	0	use the lower right corners.
1	1	use the lower left corners.

Bits 4 - 15 not used and must be set to zero.

Location X
Location Y

These are relative distance from an image corner to a graphics item corner. Units are image pixels (same units as Image Width and Image Length). These numbers are always positive.

Color

This is the color that the graphic used. Not used in DEFF rev A but must be set to zero.

Each of the graphics items has all of the above attributes. The graphics items are:

Frame Number
Event Timers
Blood Pressure
Heart Rate
Percent of Maximum Heart Rate
Date
Time
View Name
Stage Name
Protocol Name
Scaling Info (What is this?)
Time from R Wave
User Annotation (Value not in tags.)
Measurements. (Value not in tags.)
Spare 7
Patient Name
Patient ID
Spare 0
Spare 1
Spare 2
Spare 3
Spare 4
Spare 5
Spare 6

Number of Timers

Tag	0x400D
Type	SHORT
Length	1
Range	0..4
Required	

This tag gives the number of application timers visible at the time of acquisition of the image in the current file.

A value of 0 indicates that no timer information is provided.

Organ Scan

Tag	0x6001
Type	SHORT
Length	1
Range	0, 1
Optional	
Default	0

This tag is a flag to indicate that the DEFF file contains a loop which was acquired as an organ scan. An organ scan is a loop captured while moving the scanhead across an organ. The flag may be used to play the loop in alternating forward and reverse directions, instead of in a continually repeated increasing order.

0 = Loop is not an organ scan, or unknown image content
1 = Loop is an organ scan

Primary Frame

Tag	0x4010
Type	SHORT
Length	1
Range	0..Number of Pages - 1 (from Page Number tag)
Optional	Default is zero.

This tag gives the page (frame) number of the "Primary" frame, that is, the frame regarded by the operator as being representative of the entire image loop contained in the current DEFF file.

Protocol Information (formerly File Identification)

Tag	0x4007
Type	SHORT
Length	7
Range	n/a
Required	

This tag contains a collection of integer parameters in this order:

Study ID
    The Study ID (SID) to which the image belongs.
Sequence Number
    The Sequence Number (SN) to which the image belongs.
Image ID
    The Image ID (IID) of the image.
Number of Stages
    The total number of stages in the protocol. The default is 1.
Stage Number
    The protocol stage number (0..maxstages-1) of the cineloop in this file. The default is 0.
Number of Views
    The total number of views in the current protocol stage. The default is 1.
View Number
    The protocol view number (0..maxviews-1) of the cineloop in this file. The default is 0.

Replay Area

Tag	0x4013
Type	SHORT
Length	7
Range	0..65535
Optional	
Default	See text.

This tag contains a collection of integer parameters in this order:

Replay Window Position X
: The operator-selected default playback position for the image loop contained in this file. Units are pixels from the screen origin (top left). Default = 0.

Replay Window Position Y
: The operator-selected default playback position for the image loop contained in this file. Units are pixels from the screen origin (top left). Default = 0.

Replay Window Size X
: The operator-selected default window size into which the image contained in this file is loaded. Units are pixels from the screen origin (top left). If the tag is not present, the image is loaded into a window the same size as the Image Length and Image Width. Units are in pixels. Default = ImageWidth*CaptureDecimationX

Replay Window Size Y
: The operator-selected default window size into which the image contained in this file is loaded. Units are in pixels. Default = ImageLength*CaptureDecimationY

Replay Window Pan X
: The operator-selected default panning coordinates into the Replay Window at image load time. Units are pixels from the capture image origin (top left). Default = 0.

Replay Window Pan Y
: The operator-selected default panning coordinates into the Replay Window at image load time. Units are pixels from the capture image origin (top left). Default = 0.

Replay Screen Number
: When loading images from files back into the review station or ultrasound machine, the system could use multiple "collections" or "replay screens" to group the images for display. This number indicates the default replay screen number. Default = 0.

Note that all numbers are always positive.

Selected Beat

Tag	0x4014
Type	SHORT
Length	1
Range	0, 2
Optional	
Default	0

When multibeat image loops are acquired, the operator is given the opportunity to select the best beat. This beat is the one loaded when the image file is recalled. All beats may be stored, however, so that this selection process can be repeated later. Each beat is stored in a separate DEFF file. This tag is required if more than one beat is stored.

Value	Meaning
0	No best beat was selected.
1	This file does not contain the selected best beat.
2	This file does contain the selected best beat.

Trigger Times

Tag 0x4019
Type RATIONAL
Length 2
Range 1/((2**32)-1)..(2**32)-1
Optional
Default 0/1, 0/1

Trigger Time A
This value gives the programmed delay from ECG R-wave to the acquisition trigger A, which starts the Initial Capture Delay.
Trigger Time B
This value is Trigger B.

Trigger Times A and B occur in conjunction with the flags in the ECG Trace tag. Therefore, when the flag occurs in the ECG Trace data, the time from the R-wave of that point in the ECG waveform is given by the Trigger A and B values in this tag.

The units of measure are seconds.

Trim Points

Tag 0x4016
Type SHORT
Length 2
Range 0..(2**16)-1
Optional
Default 0 and N-1

If a cineloop is included in the DEFF file which has a larger number of frames than the number of frames between the two cineloop trim points, then this tag contains the frame numbers of the first frame inside the trim and the last frame inside the trim.

If there are no trim points specified or the file only contains the points inside the trim, then the first trim point is zero and the last is N-1. Where N is the number of frames in the file.

Start Loop Trim
The frame number of the first frame inside the trim boundary.
Stop Loop Trim
The frame number of the last frame inside the trim boundary.

8.0 Pictorial Class

8.1 Introduction

This section describes an image class which allows a pictorial directory to be created. The directory would consist of many small images, each representing one DEFF file available for review.

8.2 File Structure

Pictorial Class DEFF files use the same structure as described in the Base Class DEFF specification.

8.3 DEFF Specific Concepts

8.3.1 Pictorial Directory Image Files

A pictorial directory is a way for a ultrasound system or review station to present to the user a file directory consisting of pictures. To fit many pictures, or images, on one display, each image must be very small.

To support this feature, a specific DEFF file subset is defined as a pictorial directory file. A pictorial directory file has certain characteristics which are a subset of the DEFF specification. The most notable characteristics are:

-Image size is one/sixteenth the area of "normal" images
-Single image, not loops.
-Pixel format is limited to one type
-All graphics are embedded in the pixel data, no graphics lists.

8.4 Tag Descriptions

8.4.1 Tag List

All numbers are in hexadecimal. The "Required" field indicates whether the tag is required for this particular class. If the field is "cond" then inclusion of the tag is dependent on some other tag's value.

IFD	Tag #	Type	Required	Description
Ext	9300	short	yes	Pictorial Pixel Type
Ext	9301	long	no	Pictorial Version

Obsolete tags:
None

Reserved Tags:

Ext	9302-93FF	Reserved for Pictorial Class

8.4.2 Tag Restrictions

The Pictorial Class restricts the usage of several Image Class tags to a subset of their defined values. This makes Pictorial Class images less variable and eases implementation. This section defines the restrictions to the Pictorial Class' superclass tags.

Bits Per Sample
This tag record determines the number of bits representing each primary color component of a pixel.
Must be: 16

Photometric Interpretation
This tag record describes how to interpret the values in a pixel. DEFF values are:
Must be: 32820 Other type of interpretation.

Samples Per Pixel
This tag record determines the number of primary color components of a pixel.
Must be: 1 Gray scale or palette images (default)

8.4.3 TIFF Specific Tags

None.

8.4.4 DEFF Specific Tags

These tags are contained only in a special Extended IFD pointed to by the ExtendedTagsOffset tag. The tag numbers are not official TIFF but do not need to be since no TIFF reader will encounter them. Thus, there will not be TIFF reader conflicts caused by these tags.

Pictorial Pixel Type

Tag 0x9300
Type SHORT
Length 1
Range 1..5
Required in DEFF B4 and later
Optional otherwise
Default 4 (Mixed RGB/gray, 555/8)

This record specifies the kind of data contained in this file. Legal values are:

Value	Meaning

4 Mixed RGB/gray raster data. Packed into one 16 bit word per pixel. Pixels vary on a pixel by pixel basis and are either MSB = 1, then 5 bit red, 5 bit green, and 5 bit blue or MSB = 0 next 7 don't cares and 8 bits gray. To be compatible with an UM9 HDI system, the don't cares must be set to 0100110 binary. The 16 bits are stored 8 bit Least Significant Byte first and then 8 bit Most Significant Byte second, regardless of the overall file's byte order. Used for VCR playback captured images. This pixel format will not have a palette. It does have the following values for other tags:

Photometric interpretation:	32820 ("other")
Samples per pixel:	1
Bits per sample:	16

Only the single value is defined at this time in order to allow any type of captured image to be represented by pictorial images that do not conflict when all presented on a display at the same time.

The pixels are stored in non-interlaced rows.

Pictorial Version

Tag	0x9301
Type	LONG
Length	1
Range	0..(2**32)-1
Optional	Default 1, DEFF 2.0 compatible

The Pictorial Version tag contains an integral number which represents compatibility with the ability to display this pictorial class DEFF file.

The version number increments only when changes to this class has rendered inoperable the ability to display the image using software which was only compatible up to the last version. This is expected to occur very infrequently.

Note that this tag was added at DEFF version 2.2 thus older DEFF files will not have this tag.

Version Number	DEFF Spec	Reason for change
1	2.0	Original Pictorial Class

Appendix A Revision History

DEFF 2.5 Changes

Locator Database
Added format section statement restricting control character use.
Clarified swVersion to be per product not necessarily per company.
Fixed typo in filename field example.
Added requirment for applications to insure patientID field unique to disk.
Added the Protocol Name (PN) flag.
Added the Exam (EX) flag.
Added the Vendor Flags (VF) flag.
Added the Archive Complete (AC) flag.
Clarified the use of the UserID (US) flag.
Clarified the use of the FormFeed (FF) flag.
Added requirement that the vendorSpec fields be printable.
Deleted "currently always the case" from grayColorFlag and numberFrames.
Clarified Filename requirments in that section.
Fixed typo in Acquisition System Operation description.
Added note to Printer System Operation explaining possible image format changes occurring for one patient.

Base Class
Added version number for DEFF 2.5. Same as DEFF 2.2.

Image Class
Completely changed Aspect Ratio concept description.
Removed aspect ratio formula from XResolution tag description.

Archive Class
Corrected Aspect Ratio concept description.

DEFF 2.4 Changes

Locator Database
Made obsolete the Delete Record (DR) and Delete Image (DI) flags.
Clarified definition of Also Printed (AP) flag.
Added the Archive Class image exists flag (AE).
Added the Camera Ready Class image exists flag CE).
Added the Pictorial Class image exists flag (PE).
Added the Results Class data exists flag (RE).
Added the Unspecified file exists flag (UE).
Completely revised the sections describing the use of the locator databases.
Added section on filename requirements.

Image Class
Added note in Compression tag on use of LZW algorithm requiring a license.

Base Class
Added version number for DEFF 2.4. Same as DEFF 2.2.

DEFF 2.3 Changes

Overview
Added 3.5" MO disk.

Disk File System
Added 3.5" MO disk.

Locator Database
Changed statement so that chronological record order is not guaranteed.
Added note that implementations should check for the existence of a file of the name being given to a newly created file in order to avoid overwriting an existing file.

Base Class
Added statement that in all enumerated type fields, a value with the MSB set indicates a private vendor specific usage. Multiple vendors may use the same enumerated value for different purposes.
Added version number for DEFF 2.3 and note about preference to check tags, not version.

Image Class
Changed error in Number of Regions inclusion condition to be required if Spatial Calibration tag is present.
Added units (of hertz) to Spatial Calibration tag Transducer Frequency field.

Tag Index

Strip Offsets, 50
Trigger Times, 77
Trim Points, 77
X Resolution, 50
Y Resolution, 50