OGC GeoTIFF Standard

Table of Contents

Open Geospatial Consortium
Submission Date: <yyyy-mm-dd>
Approval Date: <yyyy-mm-dd>
Publication Date: <2018-12-16>
External identifier of this OGC® document: http://www.opengis.net/doc/{doc-type}/{standard}/{m.n}
Internal reference number of this OGC® document: YY-nnnrx
Version: 0.3
Category: OGC® <Implementation>
Editor: Ted Habermann

OGC GeoTIFF Standard

Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Standard
Document subtype:	Not Applicable
Document stage:	Draft
Document language:	English

you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

i. Abstract

This profile specifies the requirements and encoding rules for using the Tagged Image File Format (TIFF) for the exchange of georeferenced imagery. It formalizes the existing community standard for the Geographic Tagged Image File Format (GeoTIFF) file format.

ii. Keywords

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, geotiff, tiff

iii. Preface

The GeoTIFF format was initially developed during the early 1990's (N. Ritter & Ruth, 1997) in order to leverage a mature platform independent file format (TIFF) by adding metadata required for describing and using geographic image data. TIFF met the requirements for an underlying format, as it was lossless and extensible. In September 1994, SPOT Image Corp proposed a GeoTIFF structure that was limited to Universal Transverse Mercator (N. Ritter & Ruth, 1997). The proposed GeoTIFF specification has augmented and formalized by Niles and Ruth as Revision 1.0, specification version 1.8.2 in November 1995 (N. Ritter & Ruth,1995). This specification is currently the official GeoTIFF specification (GeoTIFF, n.d).

The GeoTIFF format is used throughout the geospatial and earth science communities to share geographic image data. That usage inevitably leads to identification of new requirements and needs for profiles, extensions, and improvements to the original GeoTIFF Specification. The OGC is well established as a forum for standardization in the GeoTIFF producer and user communities and, as such, it provides an inclusive standardization process for those communities. This document is the first step in the process of integration of the GeoTIFF into that standardization process. Once GeoTIFF is ensconced in the OGC, the standard can be evolved using a formal process.

Suggested additions, changes, and comments on this standard are welcome and encouraged. Such suggestions may be submitted by email message or by submitting an official OGC Change Request using the online CR application:

```
https://portal.opengeospatial.org/public_ogc/change_request.php
```

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

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

iv. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

The HDF Group

v. Submitters

All questions regarding this submission should be directed to the editor or the submitters:

Name	Affiliation
Ted Habermann	The HDF Group
Emmanuel Devys	IGN
Charles Heazel	Heazeltech LLC
Roger Lott	International Association of Oil & Gas Producers (IOGP)
Even Rouault	Spatialys

Chapter 1. Scope

This OGC® Standard defines the Geographic Tagged Image File Format (GeoTIFF) file format and the requirements to which every GeoTIFF file must adhere. GeoTIFF defines a set of TIFF tags provided to describe all "Cartographic" information associated with TIFF imagery that originates from satellite imaging systems, scanned aerial photography, scanned maps, digital elevation models, or as a result of geographic analyses. Its aim is to allow means for tying a raster image to a known model space or earth-based coordinate reference system, and for describing those coordinate reference systems.

The tags documented in this spec are to be considered completely orthogonal to the raster-data descriptions of the TIFF spec, and impose no restrictions on how the standard TIFF tags are to be interpreted, which color spaces or compression types are to be used, etc.

This minor revision of GeoTIFF focuses on updating the current GeoTIFF community specification and aligning it with current OGC standardization practice, as well as enabling the use of coordinate reference systems that have been included in EPSG register since the GeoTIFF 1.0 specification dating from 1995. It should therefore be backward compatible with the 1.0 version, both for coordinate reference systems based on EPSG register codes, or user-defined coordinate reference systems.

Chapter 2. Conformance

This standard defines 120 Requirements grouped into 29 Requirements Classes.

These Requirements address five (5) standardization target types:

- Underlying TIFF Requirements
- GeoTIFF Configuration GeoKeys
- Geographic CRS Parameter GeoKeys
- Projected CRS Parameter GeoKeys
- Vertical CRS Parameter GeoKeys

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site.

In order to conform to this OGC® encoding standard, a software implementation shall comply with all of the conformance classes specified in Annex A (normative).

This version 1.0 of the OGC® GeoTIFF standard is a formalized version of the GeoTIFF specification currently in common use. Any discrepancies between this specification and the community GeoTIFF standard identified in Chapter 3 should be brought to the attention of the OGC® for resolution.

Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

- Association Adobe Developers, ***TIFF Specification Revision 6.0***, June 3, 1992
- Ritter, Niles & Ruth, Mike, ***GeoTIFF Format Specification***, October 31, 1995

Chapter 4. Terms and Definitions

For the purposes of this document, the following additional terms and definitions apply.

4.1. band

range of wavelengths of electromagnetic radiation that produce a single response by a sensing device

```
[Source: 19101-2:2018, 3.1]
```

Note: At the pixel level, a band is represented as one of the vector values of the pixel. At image level, band i of an image is the rectangular array of i^{th} sample values from the pixel vectors.

4.2. cell

rectangular area in raster space, in which a single pixel value is filled

```
[Source: GeoTIFF Format Specification, October 31, 1995]
```

4.3. code

representation of a label according to a specified scheme

```
[Source: ISO 19118:2011, 4.3]
```

4.4. coordinate

one of a sequence of numbers designating the position of a point

Note: In a spatial coordinate reference system, the coordinate numbers are qualified by units.

```
[Source: ISO 19111:2019, 3.1.5]
```

4.5. coordinate reference system

coordinate system that is related to an object by a datum

Note 1: Geodetic and vertical datums are referred to as reference frames.

Note 2: For geodetic and vertical reference frames, the object will be the Earth. In planetary applications, geodetic and vertical reference frames may be applied to other celestial bodies.

```
[Source: ISO 19111:2019, 3.1.9]
```

NOTE This term is also called Model Coordinate Reference system in the context of this document, and was called Model Coordinate system in the 1995 GeoTIFF v1.0.

4.6. coordinate system

set of mathematical rules for specifying how coordinates are to be assigned to points

```
[Source: ISO 19111:2019, 3.1.11]
```

4.7. correspondence model

functional relationship between ground and image coordinates based on the correlation be-tween a set of ground control points and their corresponding image coordinates

```
[Source: ISO/TS 19130:2010, 4.3]
```

4.8. datum

reference frame

parameter or set of parameters that realize the position of the origin, the scale, and the orientation of a coordinate system

```
[Source ISO 19111:2019, 3.1.15]
```

4.9. device space

coordinate space referencing scanner, printers and display devices

```
[Source: GeoTIFF Format Specification, October 31, 1995]
```

4.10. double

8-byte IEEE double precision floating point

4.11. ellipsoid

reference ellipsoid

<geodesy> geometric reference surface embedded in 3D Euclidean space formed by an ellipse that is rotated about a main axis

Note: For the Earth the ellipsoid is bi-axial with rotation about the polar axis. This results in an oblate ellipsoid with the midpoint of the foci located at the nominal centre of the Earth.

```
[Source: ISO 19111:2019, 3.1.22]
```

4.12. flattening

f

ratio of the difference between the semi-major (a) and semi-minor axis (b) of an ellipsoid to the semi-major axis; f = (a - b)/a

Note: Sometimes inverse flattening 1/f = a/(a - b) is given instead; 1/f is also known as reciprocal flattening.

```
[Source: ISO 19111:2019, 3.1.28]
```

4.13. geocoding

translation of one form of location into another

```
[Source: ISO 19133:2005, 4.4]
```

NOTE	In the 1995 GeoTIFF v1.0, "an image is geocoded if a precise algorithm for determining the earth-location of each point in the image is defined".

4.14. geographic coordinate reference system

coordinate reference system that has a geodetic reference frame and an ellipsoidal coordinate system

```
[Source: ISO 19111:2019, 3.1.35]
```

Note: This allows the assignment of a Latitude-Longitude vector to a location on earth (plus optionally a geodetic height).

NOTE	In the the 1995 GeoTIFF v1.0, this term was "geographic coordinate system"

4.15. geokey

in GeoTIFF, a GeoKey is equivalent in function to a TIFF tag, but uses a different storage mechanism

```
[Source: GeoTIFF Format Specification, October 31, 1995]
```

4.16. georeferencing

geopositioning an object using a Correspondence Model derived from a set of points for which both ground and image coordinates are known

```
[Source: ISO 19130:2010, 4.37]
```

NOTE In the 1995 GeoTIFF v1.0, "An image is georeferenced if the location of its pixels in some model space is defined, but the transformation tying model space to the earth is not known".

4.17. GeoTIFF

standard for storing georeference and geocoding information in a TIFF 6.0 compliant raster file

```
[Source: GeoTIFF Format Specification, October 31, 1995]
```

4.18. grid

network composed of two or more sets of curves in which the members of each set intersect the members of the other sets in an algorithmic way

Note: The curves partition a space into grid cells.

```
[Source: ISO 19123:2005, 4.1.23]
```

4.19. imagery

representation of phenomena as images produced electronically and/or optical techniques

Note: In this document, it is assumed that the phenomena have been sensed or detected by one or more devices such as radar, cameras, photometers, and infra-red and multispectral scanners

```
[Source: 19101-2:2018, 3.14]
```

4.20. meridian

intersection of an ellipsoid by a plane containing the shortest axis of the ellipsoid

Note: This term is generally used the describe the pole-to-pole arc rather than the complete closed figure.

```
[Source: 19111:2019, 3.1.42]
```

4.21. metadata

information about a resource

```
[Source: ISO 19115-1:2014, 4.10]
```

4.22. model space

space in a coordinate reference system related to the earth or a part of the earth

4.23. mosaic

an image composed of two or more separately collected (sensed) images

Note: Additional XML metadata may be used to identify the cut-lines (boundaries and parameters for the images used to compose the mosaic.

4.24. orthorectified grid

orthoimage

image in which by orthogonal projection to a reference surface, displacement of image points due to sensor orientation and terrain relief has been removed

Note: The amount of displacement depends on the resolution and the level of detail of the elevation information and on the software implementation.

```
[Source: 19101-2:2008, 3.25]
```

```
georectified grid created using ground control points and elevation data where
constant scale is maintained throughout the grid
```

4.25. parallel

line of constant latitude, parallel to the equator

```
[Source: GeoTIFF Format Specification, October 31, 1995]
```

4.26. pixel

smallest element of a digital image to which attributes are assigned.

Note 1: This term originated as a contraction of “picture element”.

Note 2: Related to the concept of a grid cell.

```
[Source: 19101-2:2008, 3.28]
```

4.27. prime meridian

meridian from which the longitudes of other meridians are quantified

```
[Source: ISO 19111:2019, 3.1.50]
```

4.28. projected coordinate reference system

coordinate reference system derived from a geographic coordinate reference system by applying a map projection

Note 1: May be two- or three-dimensional, the dimension being equal to that of the geographic coordinate reference system from which it is derived.

Note 2: In the three-dimensional case the horizontal coordinates (geodetic latitude and geodetic longitude coordinates) are projected to northing and easting and the ellipsoidal height is unchanged.

```
[Source ISO 19111:2019, 3.1.51]
```

NOTE	In the the 1995 GeoTIFF v1.0, this term was "projected coordinate system"

4.29. projection

projected coordinate reference system

coordinate conversion from an ellipsoidal coordinate system to a plane

```
[Source: ISO 19111:2019, 3.1.40]
```

4.30. raster

raster space

usually rectangular pattern of parallel scanning lines forming or corresponding to the display on a cathode ray tube

```
Note:   A raster is a type of grid.
```

```
[Source: ISO 19123:2005, 4.1.30]
```

NOTE In the the 1995 GeoTIFF v1.0, "A continuous planar space in which pixel values are visually realized."

4.31. rational <TIFF>

a *rational* value is a fractional value represented by the ratio of two unsigned 4-byte integers

4.32. rectified grid

georectified grid

grid for which there is an affine transformation between the grid coordinates and the coordinates of an external coordinate reference system

Note: If the coordinate reference system is related to the earth by a datum, the grid is a georectified grid.

```
 [Source: ISO 19123:2005, 4.1.32]
=== referenceable grid
__georeferenceable grid__
```

grid associated with a transformation that can be used to convert grid coordinate values to values of coordinates referenced to an external coordinate reference system

```
Note: If the coordinate reference system is related to the earth by a datum, the grid
is a georeferenceable grid.
```

```
[Source: ISO 19123:2005, 4.1.33]
```

4.33. short

2-byte IEEE signed integer

4.34. tag <TIFF>

a tag is packet of numerical or ASCII values, which have a numerical "Tag" ID indicating their information content

```
[Source: GeoTIFF Format Specification, October 31, 1995]
```

4.35. vertical coordinate reference system

one-dimensional coordinate reference system based on a vertical reference frame

```
[Source: ISO 19111:2019, 3.1.70]
```

Chapter 5. Conventions

5.1. Abbreviations

5.1.1. ASCII

[American Standard Code for Information Interchange] The predominant character set encoding of present-day computers.

5.1.2. CRS

Coordinate Reference System

5.1.3. EPSG

Geodetic parameter dataset initiated by the now-disbanded European Petroleum Survey Group and now maintained at http://www.epsg-registry.org by the International Association of Oil and Gas Producers (IOGP).

5.1.4. IEEE

Institute of Electrical and Electronics Engineers, Inc.

5.1.5. IFD

In TIFF format, an Image File Directory, containing all the TIFF tags for one image in the file (there may be more than one).

5.1.6. POSC

Petrotechnical Open Software Corporation.

5.1.7. TIFF

Acronym for Tagged Image File Format; a platform-independent, extensive specification for storing raster data and ancillary information in a single file.

5.1.8. URI

Uniform Resource Identifier

5.1.9. USGS

United States Geological Survey

5.1.10. 2D

two-dimensional

5.1.11. 3D

three-dimensional

5.1.12. Identifiers

The normative provisions in this specification are denoted by the URI

http://www.opengis.net/spec/{standard}/{m.n}

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

5.2. Symbols

5.2.1. a

semi-major axis of ellipsoid

5.2.2. b

semi-minor axis of bi-axial ellipsoid

5.2.3. f

flattening

Chapter 6. Clauses not Containing Normative Material

The following section of this Standard is included for informative purposes only. It serves to assist in correct implementation of this standard but do not impose any additional requirements.

- Annex B: The GeoTIFF File Structure: This Annex provides a description of the community GeoTIFF specification and supporting information on how various elements of that specification should be used.
- Annex D: Recommendations for describing compound and geographic 3D CRSs: This Annex provides recommendations for describing 3D Model CRS subtypes to which heights are referenced (compound and geographic 3D) that are not fully supported in this community specification.
- Annex E: Summary of GeoKey IDs and names: This Annex lists the names and IDs of all GeoKeys supported in this specification.
- Annex F: Examples: This Annex contains a set of illustrative examples of GeoTIFF tags contents.

Chapter 7. Requirements

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

This OGC Standard specifies the content and structure of a group of industry-standard tag sets for the management of georeferenced or geocoded raster imagery using Aldus-Adobe’s public domain Tagged-Image File Format (TIFF).

This standard defines a set of TIFF tags provided to describe all "Cartographic" information associated with TIFF imagery that originates from satellite imaging systems, scanned aerial photography, scanned maps, digital elevation models, or as a result of geographic analyses. Its aim is to allow means for tying a raster image to a known model space or earth-referenced coordinate reference system, and for describing that CRS.

7.1. Underlying TIFF Requirements

This Standardization Target addresses the core Requirements Classes applicable to all GeoTIFF files.

A GeoTIFF file is a TIFF 6.0 file and inherits the file structure as described in the corresponding portion of the TIFF spec. In addition, all GeoTIFF specific information is encoded in several reserved TIFF tags. The following requirements formalize compliance with TIFF and the GeoTIFF reserved tag structure.

The tags documented in this standard are to be considered completely orthogonal to the raster-data descriptions of the TIFF spec, and impose no restrictions on how the standard TIFF tags are to be interpreted, which color spaces or compression types are to be used, etc.

7.1.1. Requirements Class TIFF

A GeoTIFF file is a valid TIFF 6.0 file.

Requirements Class: TIFF	
http://www.opengis.net/spec/GeoTIFF/0.0/Core	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/TIFF *A GeoTIFF file SHALL be compliant with the TIFF 6.0 specification*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/DataGeoTags *GeoTIFF files SHALL encode all GeoTIFF specific information using the specified reserved TIFF tags*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/DataTypes *GeoTIFF implementation software SHALL support all documented TIFF 6.0 tag data-types, and in particular the IEEE double-precision floating point "DOUBLE" type*

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ByteOrder *GeoTIFF implementation software SHALL honor the 'byte-order' indicator by performing byte swaps as necesssary to provide an accurate in-memory representation of the values encoded in the GeoTIFF file*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/TagSort *The TIFF tags in a GeoTIFF file SHALL be written out to the file with the tag-IDs sorted in ascending order*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeySort *The GeoKey entries in a GeoTIFF file SHALL be written out to the file with the key-IDs sorted in ascending order*

7.1.2. Requirements Class GeoKeyDirectoryTag

The GeoKeyDirectoryTag Requirements Class specifies the requirements for implementing the reserved GeoKeyDirectoryTag TIFF tag.

A GeoTIFF file stores projection parameters in a set of "Keys" which are virtually identical in function to a TIFF tag, but have one more level of abstraction above TIFF. Like a tag, a Key has an ID number ranging from 0 to 65535, but unlike TIFF tags, all key ID's are available for use in GeoTIFF parameter definitions.

The Keys in GeoTIFF (also called "GeoKeys") are all referenced from the GeoKeyDirectoryTag tag. The first four keys form the GeoKey Directory Header. The keys which make up this header are: KeyDirectoryVersion, KeyRevision, MinorRevision, and NumberOfKeys.

The GeoKey Directory Header is followed by <NumberOfKeys> Key Entries. Each Key Entry consists of four values:

- "KeyID" gives the key-ID value of the Key (identical in function to TIFF tag ID, but completely independent of TIFF tag-space).
- "TIFFTagLocation" indicates which TIFF tag contains the value(s) of the Key: if TIFFTagLocation is 0, then the value is SHORT, and is contained in the "Value_Offset" entry. Otherwise, the type (format) of the value is implied by the TIFF-Type of the tag containing the value.
- "Count" indicates the number of values in this key.
- "Value_Offset" Value_Offset indicates the index- offset **into** the TagArray indicated by TIFFTagLocation, if it is nonzero. If TIFFTagLocation=0, then Value_Offset contains the actual (SHORT) value of the Key, and Count=1 is implied. Note that the offset is not a byte-offset, but rather an index based on the natural data type of the specified tag array.

Requirements Class: GeoKeyDirectoryTag	
http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.ID *The GeoKeyDirectoryTag SHALL have ID = 34735*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.type *The GeoKeyDirectoryTag SHALL have type = SHORT (2-byte unsigned integer)*

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.count *The GeoKeyDirectoryTag SHALL include at least 4 keys (short integers) as header information*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyDirectoryVersion *The first unsigned short integer in the GeoKeyDirectoryTag SHALL hold the KeyDirectoryVersion.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyDirectoryVersionValue *The value of KeyDirectoryVersion SHALL be 1.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyRevision *The second unsigned short integer in the GeoKeyDirectoryTag SHALL hold the KeyRevision.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyRevisionValue *The value of KeyRevision SHALL be 1.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.minorRevision *The third unsigned short integer in the GeoKeyDirectoryTag SHALL hold the MinorRevision.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.minorRevisionValue *The MinorRevision for this standard SHALL be O or 1.* 0 = 1995 version 1 = 2019 version
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.numberOfKeys *The fourth unsigned short integer in the GeoKeyDirectoryTag SHALL hold the NumberOfKeys defined in the rest of the GeoKeyDirectoryTag.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyEntrySetCount *The GeoKeyDirectoryTag SHALL hold NumberOfKeys KeyEntry Sets in addition to the header information*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.KeyEntry *Each Key Entry in the Key Entry Set SHALL include 4 unsigned short integer values*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyEntryKeyID *The first unsigned short integer in the Key Entry SHALL hold the key identifier.*

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyEntryTIFFTagLocation *The second unsigned short integer in the Key Entry SHALL hold the TIFF Tag Location. The value of this entry shall be a valid GeoTIFF tag identifier or a zero (0)*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyEntryKeyCount *The Third unsigned short integer in the Key Entry SHALL indicate the number of values associated with this key.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyDirectoryTag.keyEntryValueOffset *The fourth unsigned short integer in the Key Entry SHALL hold either the key value (if TIFF Tag location = 0) or the index into the tag indicated by the TIFF Tag Location value.*

7.1.3. Requirements Class GeoKeyCode

For consistency, several key codes have the same meaning in all implemented GeoKeys

*Requirements Class: GeoKeyCode	
http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyCode	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyCode.undefined *GeoKeys with a value of zero SHALL indicate intentionally omitted parameters*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyCode.userDefined *GeoKeys with a value of 32767 SHALL indicate user-defined parameters*

The "undefined" code means that this parameter is intentionally omitted.

In some cases, additional GeoKeys are required when the "User-Defined" is used. Those requirements are included within a Requirements Class where appropriate.

7.1.4. Requirements Class GeoShortParams

The following requirements govern the storage of parameter values when there are two or more values of type Short.

Requirements Class: GeoShortParams	
http://www.opengis.net/spec/GeoTIFF/0.0/GeoShortParams	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoShortParams.Criteria *In the case where a Parameter of type Short has more than one value, those values SHALL be stored in the GeoKeysDirectory*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoShortParams.Location. *Parameter values stored in the GeoKeysDirectory SHALL appear after the last Key Entry*

7.1.5. Requirements Class GeoDoubleParamsTag

The following requirements govern the storage of parameter values when the values are of type Double.

Requirements Class: GeoDoubleParamsTag	
http://www.opengis.net/spec/GeoTIFF/0.0/GeoDoubleParamsTag	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoDoubleParamsTag.ID *The GeoDoubleParamsTag SHALL have ID = 34736*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoDoubleParamsTag.count *The GeoDoubleParamsTag MAY hold any number of key parameters with type = Double.*

7.1.6. Requirements Class GeoAsciiParamsTag

The following requirements govern the storage of parameter values when the values are of type ASCII. All text values in a TIFF file must be null-terminated ASCII.

Requirements Class: GeoAsciiParamsTag	
http://www.opengis.net/spec/GeoTIFF/0.0/GeoAsciiParamsTag	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoAsciiParamsTag.ID *The GeoAsciiParamsTag SHALL have ID = 34737*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoAsciiParamsTag.count *The GeoAsciiParamsTag MAY hold any number of key parameters with type = ASCII. (May not be necessary, the same as keyentry.count)*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoAsciiParamsTag.NULLRead *Pipe characters ("|") in the GeoAsciiParamsTag SHALL be converted NULL characters before returning strings to the client*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoAsciiParamsTag.NULLWrite *NULL characters used to terminate strings in the GeoAsciiParamsTag SHALL be converted to a "|" (pipe) prior to being written into the GeoAsciiParamsTag*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeoAsciiParamsTag.type *The GeoAsciiParamsTag SHALL have type = ASCII*

7.2. GeoTIFF Configuration GeoKeys

This Standardization Target addresses encoding of Configuration values essential for interpreting the rest of the GeoKeys.

7.2.1. Requirements Class GTRasterTypeGeoKey

This establishes the Raster Space used; there are currently only two options, namely *RasterPixelIsPoint* and *RasterPixelIsArea*. No user-defined raster spaces are currently supported.

For variance in imaging display parameters, such as pixel aspect-ratios, use the standard TIFF 6.0 device-space tags. The use of this geokey is **highly recommended** for accurate georeferencing of raster.

Requirements Class: GTRasterTypeGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.ID *The GTRasterTypeGeoKey SHALL have ID = 1025*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.type *The GTRasterTypeGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.value *The GTRasterTypeGeoKey value SHALL be:* * 0 to indicate that the Raster type is undefined or unknown; or * 1 to indicate that the Raster type is PixelIsArea; or * 2 to indicate that the Raster type is PixelIsPoint; or * 32767 to indicate that the Raster type is user-defined. Recommendation: the use of 0 (undefined) or 32767 (user-defined) is not recommended
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.reserved *GTRasterTypeGeoKey values in the range 3-32766 SHALL be reserved*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.private *GTRasterTypeGeoKey values in the range 32768-65535 SHALL be private*

7.2.2. Requirements Class GTModelTypeGeoKey

This GeoKey defines the type of Model coordinate reference system used, to which the transformation from the raster space is made:

- Model CRS is unknown or unspecified;
- Model CRS is a Geographic CRS;
- Model CRS is a Geocentric CRS;
- Model CRS is a Projected CRS;
- Model CRS is user-defined.

If the Model coordinate reference system is from the GeoTIFF standard CRS register (i.e EPSG register), then only the registered CRS code need be specified. See Requirements for definition of Model CRS (when Model CRS is from GeoTIFF CRS register).

If the Model coordinate reference system is not from the GeoTIFF standard CRS register, then it requires the specification of some or all CRS elements. See Requirements for definition of user-defined Model CRS.

The GeoTIFF v1.0 format has also been used to describe pseudo-3D compound CRSs consisting of a projected CRS and a vertical CRS or a geographic 2D CRS and a vertical CRS, as well as a geographic 3D CRS. In this document, this usage is permitted but not explicitly described through the GTModelTypeGeoKey. Recommendations are given in Annex D.

Requirements Class: GTModelTypeGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.required *A GeoTIFF file SHALL include a GTModelTypeGeoKey*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.ID *The GTModelTypeGeoKey SHALL have ID = 1024*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.type *The GTModelTypeGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.value *The GTModelTypeGeoKey value SHALL be:* * 0 to indicate that the Model CRS in undefined or unknown; or * 1 to indicate that the Model CRS is a 2D projected coordinate reference system, indicated by the value of the ProjectedCRSGeoKey; or * 2 to indicate that the Model CRS is a 2D geographic coordinate reference system, indicated by the value of the GeodeticCRSGeoKey; or * 3 to indicate that the Model CRS is a geocentric Cartesian 3D coordinate reference system, indicated by the value of the GeodeticCRSGeoKey; or * 32767 to indicate that the Model CRS type is user-defined.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.reserved *GTModelTypeGeoKey values in the range 4-32766 SHALL be reserved*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.private *GTModelTypeGeoKey values in the range 32768-65535 SHALL be private*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.projCRS *If the GTModelTypeGeoKey value is 1 (Model CRS is a projected 2D CRS) then the GeoTIFF file SHALL include a ProjectedCRSGeoKey.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.geogCRS *If the GTModelTypeGeoKey value is 2 (Model CRS is a geographic 2D CRS) then the GeoTIFF file SHALL include a GeodeticCRSGeoKey.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.geocenCRS *If the GTModelTypeGeoKey value is 3 (Model CRS is a geocentric CRS) then the GeoTIFF file SHALL include a GeodeticCRSGeoKey.*

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.user-defined *If the GTModelTypeGeoKey value is 32767 (user-defined) then the GeoTIFF file SHALL include at least one of ProjectedCRSGeoKey of GeodeticCRSGeoKey and the GTCitationGeoKey SHALL be populated.*

NOTE	The GTCitationGeoKey is also provided to give an ASCII reference to published documentation on the overall configuration of the GeoTIFF file (see Citation GeoKeys).

7.3. Raster to Model Coordinate Transformation Requirements

For most common applications, the transformation between raster space and model space may be defined with a set of raster-to-model tiepoints and scaling parameters. The *ModelTiepointTag* and *ModelPixesScaleTag* may be used for this purpose.

Alternatively, the *ModelTransformationTag* may be used to specify the transformation matrix between the raster space (and its dependent pixel-value space) and the (possibly 3D) model space.

7.3.1. Requirements Class ModelTiepointTag

Requirements Class: ModelTiepointTag	
http://www.opengis.net/spec/GeoTIFF/0.0/ModelTiepointTag	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelTiepointTag.ID *The ModelTiepointTag SHALL have ID = 33922*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelTiepointTag.type *The ModelTiepointTag SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelTiepointTag.count *The ModelTiepointTag SHALL have 6 values for each of the K tiepoints*

7.3.2. Requirements Class ModelPixelScaleTag

Requirements Class: ModelPixelScaleTag	
http://www.opengis.net/spec/GeoTIFF/0.0/ModelPixelScaleTag	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelPixelScaleTag.ID *The ModelPixelScaleTag SHALL have ID = 33550*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelPixelScaleTag.type *The ModelPixelScaleTag SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelPixelScaleTag.count *The ModelPixelScaleTag SHALL have 3 values representing the scale factor in the X, Y, and Z directions*

7.3.3. Requirements Class ModelTransformationTag

Requirements Class: ModelTransformationTag	
http://www.opengis.net/spec/GeoTIFF/0.0/ModelTransformationTag	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelTransformationTag.ID *The ModelTransformationTag SHALL have ID = 34264*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelTransformationTag.type *The ModelTransformationTag SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ModelTransformationTag.count *The ModelTransformationTag SHALL have 16 values representing the terms of the 4 by 4 transformation matrix. The terms SHALL be in row-major order*

7.4. Requirements for definition of Model CRS (when Model CRS is from GeoTIFF CRS register)

7.4.1. Overview

When the Model CRS is included in the GeoTIFF CRS register (i.e EPSG register), only its register code is required. For 2D projected CRSs ("map grids") the code is given through the ProjectedCRSGeoKey. For geodetic CRSs (geographic 2D and geocentric CRSs) the code is given through the GeodeticCRSGeoKey. If the Model CRS is a pseudo-3D compound CRS consisting of either a projected 2D CRS or a geographic 2D CRS together with a vertical CRS, the code of the vertical component is given through the VerticalGeoKey.

NOTE It is a practice in some Communities to indicate a geographic 3D CRS (with ellipsoidal height) by including the code of the geographic 3D CRS in a VerticalGeoKey. See Annex D.

ProjectedCRSGeoKey

This key is used to specify the projected coordinate reference system from the GeoTIFF CRS register or to indicate that the Model CRS is a user-defined projected coordinate reference system.

NOTE In GeoTIFF 1.0 this key was called ProjectedCSTypeGeoKey

Requirements Class: ProjectedCRSGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCRSGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCRSGeoKey.ID *The ProjectedCRSGeoKey SHALL have ID = 3072*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCRSGeoKey.type *The ProjectedCRSGeoKey SHALL have type = SHORT*

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCRSGeoKey.obsolete *ProjectedCRSGeoKey values in the range 1-1000 SHALL be obsolete EPSG/POSC Datum Codes*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCRSGeoKey.reserved *ProjectedCRSGeoKey values in the range 1001-1023 SHALL be reserved.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCRSGeoKey.EPSG *ProjectedCRSGeoKey values in the range 1024-32766 SHALL be EPSG Projected CRS Codes* NOTE: In GeoTIFF v1.0 the range was 20000-32760. Several values in this range have been deprecated or deleted from the EPSG Dataset and should no longer be used. See Annex B.5.4, Table 1.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCRSGeoKey.user-defined *A ProjectedCRSGeoKey value of 32767 SHALL be a user-defined projected CRS. If the ProjectedCRSGeoKey value is 32767 (User-Defined) then the ProjectedCitationGeoKey, GeodeticCRSGeoKey and ProjectionGeoKey SHALL be populated.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCRSGeoKey.private *ProjectedCRSGeoKey values in the range 32768-65535 SHALL be private*

GeodeticCRSGeoKey

This key is provided to specify the geodetic (geographic or geocentric) coordinate reference system from the GeoTIFF CRS register or to indicate that the Model CRS is a user-defined geodetic coordinate reference system.

NOTE	In GeoTIFF 1.0 this key was called GeographicTypeGeoKey. Geodetic CRS is a superset of geographic 2D CRS, geographic 3D CRS and geocentric (earth-centred 3D Cartesian) CRS.

Requirements Class: GeodeticCRSGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.ID *The GeodeticCRSGeoKey SHALL have ID = 2048*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.type *The GeodeticCRSGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.obsolete *GeodeticCRSGeoKey values in the range 1-1000 SHALL be obsolete EPSG/POSC Geographic Codes*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.reserved *GeodeticCRSGeoKey values in the range 1001-1023 SHALL be reserved.* NOTE: In GeoTIFF v1.0 the reserved ranges were 1001-3999 and 5000-32766.

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.EPSG *GeodeticCRSGeoKey values in the range 1024-32766 SHALL be EPSG geodetic CRS codes (geographic 2D CRS, geographic 3D CRS and geocentric CRS)* NOTE: In GeoTIFF v1.0 the range was 4000-4999. Several values in this range have been deprecated or deleted from the EPSG Dataset and should no longer be used. See Annex B.5.4, Table 2.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.user-defined *If the GeodeticCRSGeoKey value is 32767 (User-Defined) then the GeodeticCitationGeoKey, GeodeticDatumGeoKey and at least one of GeogAngularUnitsGeoKey or GeogLinearUnitsGeoKey SHALL be populated.* NOTE: if the user-defined CRS is geographic 2D, GeogAngularUnitsGeoKey should be populated, if the user-defined CRS is geographic 3D, both GeogAngularUnitsGeoKey and GeogLinearUnitsGeoKey should be populated, if the user-defined CRS is geocentric, GeogLinearUnitsGeoKey should be populated.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.private *GGeodeticCRSGeoKey values in the range 32768-65535 SHALL be private*

VerticalGeoKey

This key is provided to specify the vertical coordinate reference system from the GeoTIFF CRS register or to indicate that the CRS is a user-defined vertical coordinate reference system.

NOTE In GeoTIFF 1.0 this key was called VerticalCSTypeGeoKey. In GeoTIFF v1.0 vertical coordinate reference systems were described in draft form, with the statement "Vertical coordinate systems are not yet implemented. These sections are provided for future development, and any vertical coordinate systems in the current revision must be defined using the VerticalCitationGeoKey".

Requirements Class: VerticalGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/VerticalGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalGeoKey.ID *The VerticalGeoKey SHALL have ID = 4096*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalGeoKey.type *The VerticalGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalGeoKey.reserved *VerticalGeoKey values in the range 1-1023 SHALL be reserved* NOTE: In GeoTIFF v1.0 the reserved ranges were 0001-4999 and 6000-32766.

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalGeoKey.EPSG *VerticalGeoKey values in the range 1024-32766 SHALL be* ***either*** *EPSG Vertical CRS Codes* ***or*** *EPSG geographic 3D CRS codes* NOTE: In GeoTIFF v1.0 the ranges were 5000-5099 and 5200-5999. As at 2018-05-29 no EPSG vertical CRSs have been or are in this range. Values in this range have been and are used as EPSG vertical datum codes; in this document their use as codes for vertical CRSs is deprecated.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalGeoKey.userdefined *If the VerticalGeoKey value is 32767 (User-Defined) then the VerticalCitationGeoKey and VerticalDatumGeoKey SHALL be populated.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalGeoKey.private *VerticalGeoKey values in the range 32768-65535 SHALL be private*

7.4.2. Citation GeoKeys

These keys are used to describe Model CRS elements through ASCII free text. A citation may be included with a CRS identified through the GeoTIFF CRS register (7.4.1). A citation is mandatory for a user-defined CRSs and CRS objects (7.4.3). The *GeodeticCitationGeoKey*, *ProjectedCitationGeoKey* and *VerticalCitationGeoKey* are used with CRSs and CRS components. Moreover, the *GTCitationGeoKey* is provided to give an ASCII reference to published documentation on the overall configuration of the GeoTIFF file. Moreover,

NOTE	In GeoTIFF 1.0 the GeodeticCitationGeoKey key was called GeogCitationGeoKey and the ProjectedCitationGeoKey key was called PCSCitationGeoKey

Requirements Class: CitationGeoKeys	
http://www.opengis.net/spec/GeoTIFF/0.0/CitationGeoKeys	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/CitationGeoKeys.ID *The GeodeticCitationGeoKey SHALL have ID = 2049* *The ProjectedCitationGeoKey SHALL have ID = 3073* *The VerticalCitationGeoKey SHALL have ID = 4097* *The GTCitationGeoKey SHALL have ID = 1026* NOTE: This is provided to give an ASCII reference to published documentation on the overall configuration of this GeoTIFF file.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/CitationGeoKeys.type *The CitationGeoKeys SHALL have type = ASCII*

7.5. Requirements for definition of user-defined Model CRS

The GeoKeys described in this section are needed only when Model CRSs are not available from the

GeoTIFF CRS register and the CRS or one or more of its component objects is user-defined, that is if one or more of ProjectedCRSGeoKey, GeodeticCRSGeoKey or VerticalGeoKey has a value of 32767.

NOTE	Anyone not interested in constructing a user-defined system can ignore that whole section.

7.5.1. Units GeoKeys

These keys are used to specify Units of Measure (UoM) through the identification of a unit from the GeoTIFF CRS register or to indicate that the unit is user-defined

The **GeogAngularUnitsGeoKey** key is used to specify the angular unit for:

- the axes in user-defined geographic 2D CRSs;
- the horizontal axes in user-defined geographic 3D CRSs;
- the longitude from the reference meridian in user-defined prime meridians;
- user-defined map projection parameters that are angles.

The **GeogAzimuthUnitsGeoKey** key is used to specify the angular unit for user-defined map projection parameters when these differ from the angular unit described through the GeogAngularUnitsGeoKey.

The **GeogLinearUnitsGeoKey** key is used to specify the linear unit for:

- the axes in user-defined geocentric Cartesian CRSs;
- the height axis of a user-defined geographic 3D CRS;
- for user-defined ellipsoid axes.

The **ProjLinearUnitsGeoKey** key is used to specify the linear units for:

- the axes of a user-defined projected CRS;
- map projection parameters that are lengths.

The **VerticalUnitsGeoKey** key is used to specify the linear unit for:

- the axis of a user-defined vertical CRS.

Requirements Class: UnitsGeoKey
http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.ID *The GeogAngularUnitsGeoKey SHALL have ID = 2054* *The GeogAzimuthUnitsGeoKey SHALL have ID = 2060* *The GeogLinearUnitsGeoKey SHALL have ID = 2052* *The ProjLinearUnitsGeoKey SHALL have ID = 3076* *The VerticalUnitsGeoKey SHALL have ID = 4099*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.type *The GeogAngularUnitsGeoKey, the GeogAzimuthUnitsGeoKey, the GeogLinearUnitsGeoKey, the ProjLinearUnitsGeoKey and the VerticalUnitsGeoKey SHALL each have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.obsolete *GeogAngularUnitsGeoKey, GeogAzimuthUnitsGeoKey, GeogLinearUnitsGeoKey, ProjLinearUnitsGeoKey and VerticalUnitsGeoKey values in the range 1-1023 SHALL be obsolete GeoTIFF codes.* NOTE: In GeoTIFF v1.0 the range was 0001-2000
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.angular *GeogAngularUnitsGeoKey and GeogAzimuthUnitsGeoKey values in the range 1024-32766 SHALL be EPSG Unit Of Measure (UOM) codes with type = angle.* NOTE: In GeoTIFF v1.0 the range was 9100-9199
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.linear *GeogLinearUnitsGeoKey, ProjLinearUnitsGeoKey and VerticalUnitsGeoKey values in the range 1024-32766 SHALL be EPSG Unit Of Measure (UOM) codes with type = length.* NOTE: In GeoTIFF v1.0 the range was 9000-9099. Several values in this range have been deprecated or deleted from the EPSG Dataset and should no longer be used. See Annex B.5.4, Table 3.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.userdefinedAngular *A GeogAngularUnitsGeoKey or a GeogAzimuthUnitsGeoKey value of 32767 SHALL be a user-defined angular unit. If the value is 32767 (User-Defined) then the GeodeticCitationGeoKey and the GeogAngularUnitSizeGeoKey SHALL be populated*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.userdefinedGeogLinear *A GeogLinearUnitsGeoKey value of 32767 SHALL be a user-defined linear unit. If the value is 32767 (User-Defined) then the GeodeticCitationGeoKey and the GeogLinearUnitSizeGeoKey SHALL be populated*

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.userdefinedProjLinear *A ProjLinearUnitsGeoKey value of 32767 SHALL be a user-defined linear unit. If the value is 32767 (User-Defined) then the ProjectedCitationGeoKey and the ProjLinearUnitSizeGeoKey SHALL be populated.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.userdefinedVertical *A VerticalUnitsGeoKey value of 32767 (user defined) SHALL not be used* NOTE: The rationale for this is that it would imply to fill a VerticalUnitSizeGeoKey, which does not exist in GeoTIFF 1.0. The user should use of the EPSG linear unit.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitsGeoKey.private *GeogAngularUnitsGeoKey, GeogAzimuthUnitsGeoKey, GeogLinearUnitsGeoKey, ProjLinearUnitsGeoKey and VerticalUnitsGeoKey values in the range 32768-65535 SHALL be private.*

7.5.2. Unit Size GeoKeys

These keys allow the definition of size of user-defined angular and linear units, given in the SI base unit for that unit type (meters for length, radians for angle).

Requirements Class: UnitSizeGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/UnitSizeGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitSizeGeoKey.ID *The GeogAngularUnitSizeGeoKey SHALL have ID = 2055* *The GeogLinearUnitSizeGeoKey SHALL have ID = 2053* *The ProjLinearUnitSizeGeoKey SHALL have ID = 3077*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitSizeGeoKey.type *The GeogAngularUnitSizeGeoKey, GeogLinearUnitSizeGeoKey and ProjLinearUnitSizeGeoKey SHALL each have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/UnitSizeGeoKey.units *The units of the GeogAngularUnitSizeGeoKey value SHALL be radians.* *The units of the GeogLinearUnitSizeGeoKey value SHALL be meters.* *The units of the ProjLinearUnitSizeGeoKey value SHALL be meters.*

7.5.3. Geodetic Datum

GeodeticDatumGeoKey

This key is used to specify a geodetic datum from the GeoTIFF CRS register, or to indicate that the geodetic datum or one or both of its component ellipsoid or prime meridian is user-defined.

NOTE	In GeoTIFF 1.0 this key was called GeogGeodeticDatumGeoKey.

Requirements Class: GeodeticDatumGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.ID *The GeodeticDatumGeoKey SHALL have ID = 2050*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.type *The GeodeticDatumGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.obsolete *GeodeticDatumGeoKey values in the range 1-1000 SHALL be obsolete EPSG/POSC Datum Codes*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.reserved *GeodeticDatumGeoKey values in the range 1001-1023 SHALL be reserved.* NOTE: In GeoTIFF v1.0 the reserved ranges were 1001-5999 and 7000-32766.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.EPSG *GeodeticDatumGeoKey values in the range 1024-32766 SHALL be EPSG geodetic datum codes.* NOTE: In GeoTIFF v1.0 the range was 6000-6999. Several values in this range have been deprecated or deleted from the EPSG Dataset and should no longer be used. See Annex B.5.4, Table 4.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.user-defined *If the GeodeticDatumGeoKey value is 32767 (User-Defined) then the GeodeticCitationGeoKey, PrimeMeridianGeoKey and EllipsoidGeoKey SHALL be populated.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.private *GeodeticDatumGeoKey values in the range 32768-65535 SHALL be private*

PrimeMeridianGeoKey

This key is used to specify a Prime Meridian from the GeoTIFF CRS register or to indicate that the Prime Meridian is user-defined. The default is Greenwich, England.

NOTE	In GeoTIFF 1.0 this key was called GeogPrimeMeridianGeoKey.

Requirements Class: PrimeMeridianGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianGeoKey.ID *The PrimeMeridianGeoKey SHALL have ID = 2051*

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianGeoKey.type *The PrimeMeridianGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianGeoKey.obsolete *PrimeMeridianGeoKey values in the range 1-100 SHALL be obsolete EPSG/POSC Datum Codes*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianGeoKey.reserved *PrimeMeridianGeoKey values in the range 101-1023 SHALL be reserved* NOTE: In GeoTIFF v1.0 the range was 101-7999 and 9000-32766
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianGeoKey.EPSG *PrimeMeridianGeoKey values in the range 1024-32766 SHALL be EPSG Prime Meridian Codes* NOTE: In GeoTIFF v1.0 the range was 8000-8999
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianGeoKey.user-defined *If the PrimeMeridianGeoKey value is 32767 (User-Defined) then the GeodeticCitationGeoKey, and PrimeMeridianLongGeoKey SHALL be populated*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianGeoKey.private *PrimeMeridianGeoKey values in the range 32768-65535 SHALL be private*

PrimeMeridianLongitudeGeoKey

This key allows definition of a user-defined Prime Meridian, the location of which is defined by its longitude relative to the international reference meridian (for the earth this is Greenwich).

NOTE In GeoTIFF 1.0 this key was called GeogPrimeMeridianLongGeoKey.

Requirements Class: PrimeMeridianLongitudeGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianLongitudeGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianLongitudeGeoKey.ID *The PrimeMeridianLongitudeGeoKey SHALL have ID = 2061*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianLongitudeGeoKey.type *The PrimeMeridianLongitudeGeoKey SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/PrimeMeridianLongitudeGeoKey.units *The unit for the PrimeMeridianLongitudeGeoKey value SHALL be GeogAngularUnits*

Ellipsoid

EllipsoidGeoKey

This key is provided to specify an ellipsoid (or sphere) from the GeoTIFF CRS register or to indicate that the ellipsoid (or sphere) is user-defined.

NOTE In GeoTIFF 1.0 this key was called GeogEllipsoidGeoKey.

Requirements Class: EllipsoidGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidGeoKey.ID *The EllipsoidGeoKey SHALL have ID = 2056*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidGeoKey.type *The EllipsoidGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidGeoKey.obsolete *EllipsoidGeoKey values in the range 1-1000 SHALL be obsolete EPSG/POSC Datum Codes*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidGeoKey.EPSG *EllipsoidGeoKey values in the range 1024-32766 SHALL be EPSG ellipsoid Codes* NOTE: In GeoTIFF v1.0 the range was 7000-7999. Several values in this range have been deprecated or deleted from the EPSG Dataset and should no longer be used. See Annex B.5.4, Table 5.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidGeoKey.user-defined *If the EllipsoidGeoKey value is 32767 (User-Defined) then the GTCitationGeoKey and the EllipsoidSemiMajorAxisGeoKey SHALL be populated together with the one of either the EllipsoidSemiMinorAxisGeoKey or the EllipsoidInvFlatteningGeoKey.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidGeoKey.private *EllipsoidGeoKey values in the range 32768-65535 SHALL be private*

Ellipsoid parameter GeoKeys

These keys are used to specify the size and shape of a user-defined ellipsoid or sphere used as the model of the earth. Only bi-axial ellipsoids and spheres are catered for. An ellipsoid is defined through two parameters, its semi-major axis *(a)* and either its semi-minor axis *(b)* or its inverse flattening *(1/f)* where $1/f = a/(a - b)$. If the model is a sphere, $1/f$ is infinite so a and b must be used, with the value of b set to that of a.

EllipsoidSemiMajorAxisGeoKey

This key is provided to specify the first defining parameter of a user-defined bi-axial ellipsoid or a user-defined sphere. It allows the specification of the ellipsoid semi-major axis *(a)* or the sphere radius.

NOTE In GeoTIFF 1.0 this key was called GeogSemiMajorAxisGeoKey.

Requirements Class: EllipsoidSemiMajorAxisGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMajorAxisGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMajorAxisGeoKey.ID *The EllipsoidSemiMajorAxisGeoKey SHALL have ID = 2057*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMajorAxisGeoKey.type *The EllipsoidSemiMajorAxisGeoKey SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMajorAxisGeoKey.units *The units of the EllipsoidSemiMajorAxisGeoKey SHALL be defined by the value of GeogLinearUnitsGeoKey*

EllipsoidSemiMinorAxisGeoKey

This key is provided to specify the second defining parameter of a user-defined bi-axial ellipsoid or of a user-defined sphere. It allows the specification of the ellipsoid semi-minor axis *(b)* or the sphere radius.

NOTE In GeoTIFF 1.0 this key was called GeogSemiMinorAxisGeoKey.

Requirements Class: EllipsoidSemiMinorAxisGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMinorAxisGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMinorAxisGeoKey.ID *The EllipsoidSemiMinorAxisGeoKey SHALL have ID = 2058*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMinorAxisGeoKey.type *The EllipsoidSemiMinorAxisGeoKey SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMinorAxisGeoKey.units *The units of the EllipsoidSemiMinorAxisGeoKey SHALL be defined by the value of GeogLinearUnitsGeoKey*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidSemiMinorAxisGeoKey.sphere *If the Model CRS is a sphere, the value of the EllipsoidSemiMinorAxisGeoKey SHALL equal that of the EllipsoidSemiMajorAxisGeoKey.*

EllipsoidInvFlatteningGeoKey

This key is provided to specify the second defining parameter of a user-defined bi-axial ellipsoid. It allows the specification of the ellipsoid inverse flattening *(1/f)*. It is a ratio and does not require a unit.

NOTE In GeoTIFF 1.0 this key was called GeogInvFlatteningGeoKey.

Requirements Class: EllipsoidInvFlatteningGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidInvFlatteningGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidInvFlatteningGeoKey.ID *The EllipsoidInvFlatteningGeoKey SHALL have ID = 2059*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/EllipsoidInvFlatteningGeoKey.type *The EllipsoidInvFlatteningGeoKey SHALL have type = DOUBLE*

7.5.4. Vertical Datum

This key may be used to specify the vertical datum for a user-defined vertical coordinate reference system.

Requirements Class VerticalDatumGeoKey

Requirements Class: VerticalDatumGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey.ID *The VerticalDatumGeoKey SHALL have ID = 4098*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey.type *The VerticalDatumGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey.reserved *VerticalDatumGeoKey values in the range 1-1023 SHALL be reserved* NOTE: In GeoTIFF v1.0 the reserved range was 16384-32766.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey.EPSG *VerticalDatumGeoKey values in the range 1024-32766 SHALL be EPSG vertical datum codes* NOTE: In GeoTIFF v1.0 the range was given as 1-16383 but without reference to EPSG.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey.user-defined *If the VerticalDatumGeoKey value is 32767 (User-Defined) then the VerticalCitationGeoKey SHALL be populated.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey.private *VerticalDatumGeoKey values in the range 32768-65535 SHALL be private*

7.5.5. Map Projection GeoKeys

Requirements Class ProjectionGeoKey

The *ProjectionGeoKey* key is used to specify a map projection from the GeoTIFF CRS register or to indicate that the map projection is user-defined. In the EPSG Dataset a map projection is a coordinate conversion, a subtype of coordinate operation.

Requirements Class: ProjectionGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.ID *The ProjectionGeoKey SHALL have ID = 3074*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.type *The ProjectionGeoKey SHALL have type = SHORT*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.obsolete *ProjectionGeoKey values in the range 1-1023 SHALL be obsolete EPSG/POSC map projection codes* NOTE: In GeoTIFF v1.0 the range was 1-9999.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.EPSG *ProjectionGeoKey values in the range 1024-32766 SHALL be valid EPSG map projection (coordinate operation) codes* NOTE: In GeoTIFF v1.0 the range was 10000-19999. Several values in this range have been deprecated or deleted from the EPSG Dataset. See Annex B.5.4, Table 6.
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.user_defined *If the ProjectionGeoKey value is 32767 (User-Defined) then the ProjectedCitationGeoKey, ProjectionMethodGeoKey, and ProjLinearUnitsGeoKey SHALL be populated*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.private *ProjectionGeoKey values in the range 32768-65535 SHALL be private*

Requirements Class ProjMethodGeoKey (coordinate operation method)

The *ProjMethodGeoKey* key is used to specify a map projection method from the GeoTIFF v1.0 coordinate transformation code list (Annex C), or to indicate that the map projection method is user-defined.

NOTE In GeoTIFF 1.0 this key was called ProjCoordTransGeoKey.

NOTE GeoTIFF v1.0 did not make reference to the EPSG coordinate operation methods (a future version of GeoTIFF might do this).

Requirements Class: ProjMethodGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/ProjMethodGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjMethodGeoKey.ID *The ProjMethodGeoKey SHALL have ID = 3075*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjMethodGeoKey.type *The ProjMethodGeoKey SHALL have type = SHORT*

Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjMethodGeoKey.transform *ProjMethodGeoKey values in the range 1-27 SHALL be GeoTIFF map projection method codes* NOTE: See Annex C
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjMethodGeoKey.reserved *ProjMethodGeoKey values in the range 28-32766 SHALL be reserved*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjMethodGeoKey.user_defined *If the ProjectionMethodGeoKey value is 32767 (User-Defined) then the ProjectedCitationGeoKey and keys for each map projection parameter (coordinate operation parameter) appropriate to that method SHALL be populated.*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjMethodGeoKey.private *ProjMethodGeoKey values in the range 32768-65535 SHALL be private*

Map Projection parameters (coordinate operation parameters)

Each map projection method requires several map projection parameters (coordinate operation parameters). GeoTIFF v1.0 did not specify which parameters should be associated with which methods, nor make reference to the EPSG coordinate operation parameters associated with each method (a future version of GeoTIFF might do this).

Requirements Class ProjAngularParameters

Requirements Class: ProjAngularParameters	
http://www.opengis.net/spec/GeoTIFF/0.0/ProjAngularParameters	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjAngularParameters.ID *The ProjStdParallel1GeoKey SHALL have ID = 3078* *The ProjStdParallel2GeoKey SHALL have ID = 3079* *The ProjNatOriginLongGeoKey SHALL have ID = 3080* *The ProjNatOriginLatGeoKey SHALL have ID = 3081* *The ProjFalseOriginLongGeoKey SHALL have ID = 3084* *The ProjFalseOriginLatGeoKey SHALL have ID = 3085* *The ProjCenterLongGeoKey SHALL have ID = 3088* *The ProjCenterLatGeoKey SHALL have ID = 3089* *The ProjStraightVertPoleLongGeoKey SHALL have ID = 3095*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjAngularParameters.type *The ProjAngularParameters SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjAngularParameters.units *All parameters in this requirements class SHALL have units as specified by the GeogAngularUnitsGeoKey*

Requirements Class ProjAzimuthAngleGeoKey

Requirements Class: ProjAzimuthAngleGeoKey	
http://www.opengis.net/spec/GeoTIFF/0.0/ProjAzimuthAngleGeoKey	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjAzimuthAngleGeoKey.ID The ProjAzimuthAngleGeoKey SHALL have ID = 3094_
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjAzimuthAngleGeoKey.type *The ProjAzimuthAngleGeoKey SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjAzimuthAngleGeoKey.units *The ProjAzimuthAngleGeoKey SHALL have units as specified by the GeogAzimuthUnitsGeoKey*

Requirements Class ProjLinearParameters

Requirements Class: ProjLinearParameters	
http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearParameters	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearParameters.ID *The ProjFalseEastingGeoKey SHALL have ID = 3082* *The ProjFalseNorthingGeoKey SHALL have ID = 3083* *The ProjFalseOriginEastingGeoKey SHALL have ID = 3086* *The ProjFalseOriginNorthingGeoKey SHALL have ID = 3087* *The ProjCenterEastingGeoKey SHALL have ID = 3090* *The ProjCenterNorthingGeoKey SHALL have ID = 3091*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearParameters.type *All parameters in this requirements class SHALL have type = DOUBLE*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearParameters.units *All parameters in this requirements class SHALL have units as specified by the ProjLinearUnitsGeoKey*

Requirements Class ProjScalarParameters

Requirements Class: ProjScalarParameters	
http://www.opengis.net/spec/GeoTIFF/0.0/ProjScalarParameters	
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjScalarParameters.ID *The ProjScaleAtNatOriginGeoKey SHALL have ID = 3092* *The ProjScaleAtCenterGeoKey SHALL have ID = 3093*
Requirement	http://www.opengis.net/spec/GeoTIFF/0.0/ProjScalarParameters.type *All parameters in this requirements class SHALL have type = DOUBLE*

Chapter 8. Media Types for any data encoding(s)

A section describing the MIME-types to be used is mandatory for any standard involving data encodings. If no suitable MIME type exists in http://www.iana.org/assignments/media-types/index.html then this section may be used to define a new MIME type for registration with IANA.

A GeoTIFF file is a TIFF file. It is common to use the tiff MIME type, image/tiff according to [RFC3302]. OGC GMLCOV GeoTIFF extension (OGC 12-100r1) specifies image/tiff as MIME identifier (cf. Requirement #5). The recommendation for a specific MIME type such as image/tiff+geo, or image/tiff-geo is under consideration in this revision, and should be discussed within the OGC and the imagery communities.

Annex A: Conformance Class Abstract Test Suite (Normative)

NOTE	Ensure that there is a conformance class for each requirements class and a test for each requirement (identified by requirement name and number)

A.1. Conformance Class A

A.1.1. Requirement 1

Test id:	/conf/conf-class-a/req-name-1
Requirement:	/req/req-class-a/req-name-1
Test purpose:	Verify that...
Test method:	Inspect...

A.1.2. Requirement 2

Annex B: GeoTIFF File Structure and GeoTIFF CRS and models principles (Informative)

B.1. The GeoTIFF File Structure

B.1.1. Introduction

The current GeoTIFF specification (Ritter and Ruth, 1995) includes a detailed description of the structural approach used in GeoTIFF and the semantics and values of the tags. The tag specifications are included in Clause 7 of this standard as requirements. This Annex provides an informative overview of the structure of a GeoTIFF file and tags. Much of this information is excerpted from Ritter and Ruth, 1995.

GeoTIFF fully complies with the TIFF 6.0 specifications, and its extensions do not in any way go against the TIFF recommendations, nor do they limit the scope of raster data supported by TIFF.

GeoTIFF uses a small set of reserved TIFF tags to store a broad range of georeferencing information, catering to geodetic as well as projected coordinate reference system needs. No information is stored in private structures, IFD's or other mechanisms that would hide information from naive TIFF reading software.

GeoTIFF uses a "MetaTag" (GeoKey) approach to encode dozens of information elements into just 6 tags, taking advantage of TIFF platform-independent data format representation to avoid cross-platform interchange difficulties. These keys are designed in a manner parallel to standard TIFF tags, and closely follow the TIFF discipline in their structure and layout. New keys may be defined as needs arise, within the current framework, and without requiring the allocation of new tags from Aldus/Adobe.

GeoTIFF uses numerical codes to describe coordinate reference systems, datums, ellipsoids, etc. The codes are derived from the EPSG list compiled by the International Association of Oil and Gas Producers, and mechanisms for adding user-defined systems or their components have been established. The GeoTIFF information content is designed to be compatible with the data decomposition approach used by the National Spatial Data Infrastructure (NSDI) of the U.S. Federal Geographic Data Committee (FGDC).

While GeoTIFF provides a robust framework for specifying a broad class of existing coordinate reference systems, it is also fully extensible, permitting internal, private or proprietary information storage. However, since this standard arose from the need to avoid multiple proprietary encoding systems, use of private implementations is to be discouraged.

B.1.2. GeoTIFF Design Considerations

Every effort has been made to adhere to the philosophy of TIFF data abstraction. The GeoTIFF tags conform to a hierarchical data structure of tags and keys, similar to the tags which have been implemented in the "basic" and "extended" TIFF tags already supported in TIFF Version 6 specification. The following are some points considered in the design of GeoTIFF:

- Private binary structures, while permitted under the TIFF spec, are in general difficult to maintain, and are intrinsically platform- dependent. Whenever possible, information should be sorted into their intrinsic data-types, and placed into appropriately named tags. Also, implementers of TIFF readers would be more willing to honor a new tag specification if it does not require parsing novel binary structures.
- Any Tag value which is to be used as a "keyword" switch or modifier mistakes of mis-spelling a keyword, as well as facilitating an implementation in code using the "switch/case" features of most languages. In general, scanning ASCII strings for keywords (CaseINSensitiVE?) is a hazardous (not to mention slower and more complex) operation.
- True "Extensibility" strongly suggests that the Tags defined have a sufficiently abstract definition so that the same tag and its values may be used and interpreted in different ways as more complex information spaces are developed. For example, the old SubFileType tag (255) had to be obsoleted and replaced with a NewSubFileType tag, because images began appearing which could not fit into the narrowly defined classes for that Tag. Conversely, the YCbCrSubsampling Tag has taken on new meaning and importance as the JPEG compression standard for TIFF becomes finalized.

B.1.3. GeoTIFF Software Requirements

GeoTIFF requires support for all documented TIFF 6.0 tag data-types, and in particular requires the IEEE double-precision floating-point "DOUBLE" type tag. Most of the parameters for georeferencing will not have sufficient accuracy with single-precision IEEE, nor with RATIONAL format storage. The only other alternative for storing high-precision values would be to encode as ASCII, but this does not conform to TIFF recommendations for data encoding.

It is worth emphasizing here that the TIFF spec indicates that TIFF-compliant readers shall honor the 'byte-order' indicator, meaning that 4-byte integers from files created on opposite order machines will be swapped in software, and that 8-byte DOUBLE’s will be 8-byte swapped.

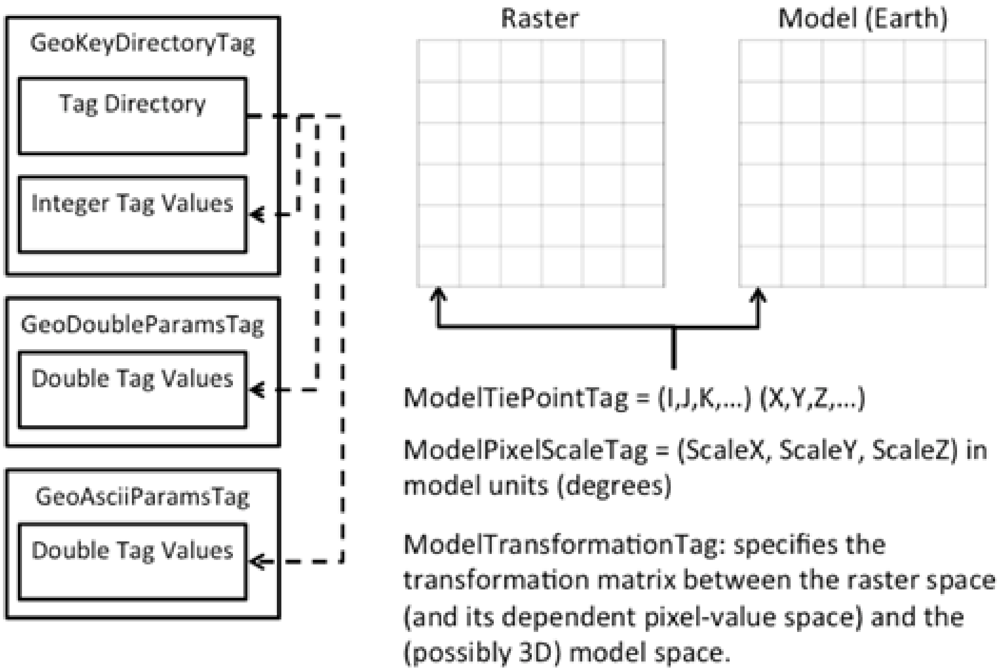
A GeoTIFF reader/writer, in addition to supporting the standard TIFF tag types, must also have an additional module which can parse the "Geokey" MetaTag information. A public-domain software package for performing this function is available; (libgeotiff) see https://trac.osgeo.org/geotiff#libgeotiff.

B.1.4. GeoTIFF File and "Key" Structure

This section describes the abstract file-format and "GeoKey" data storage mechanism used in GeoTIFF. Uses of this mechanism for implementing georeferencing and geocoding is detailed in section 2.6 and section 2.7 of Ritter and Ruth, 1995.

A GeoTIFF file is a TIFF 6.0 file, and inherits the file structure as described in the corresponding portion of the TIFF spec. All GeoTIFF specific information is encoded in several additional reserved TIFF tags, and contains no private Image File Directories (IFD’s), binary structures or other private information invisible to standard TIFF readers.

The number and type of parameters that would be required to describe most popular projection types would, if implemented as separate TIFF tags, likely require dozens or even hundred of tags, exhausting the limited resources of the TIFF tag-space. On the other hand, a private IFD, while providing thousands of free tags, is limited in that its tag-values are invisible to non-savvy TIFF readers (which don't know that the IFD_OFFSET tag value points to a private IFD).



To avoid these problems, a GeoTIFF file stores projection parameters in a set of "Keys" which are virtually identical in function to a "Tag", but has one more level of abstraction above TIFF. Effectively, it is a sort of "Meta-Tag". A Key works with formatted tag-values of a TIFF file the way that a TIFF file deals with the raw bytes of a data file. Like a tag, a Key has an ID number ranging from 0 to 65535, but unlike TIFF tags, all key ID's are available for use in GeoTIFF parameter definitions.

The Keys in GeoTIFF (also call "GeoKeys") are all referenced from the GeoKeyDirectoryTag, which defined as follows (Clause 7.1):

```
GeoKeyDirectoryTag:
    Tag = 34735 (87AF.H)
    Type = SHORT (2-byte unsigned short)
    N = variable, >= 4
    Alias: ProjectionInfoTag, CoordSystemInfoTag
```

This tag may be used to store the GeoKey Directory, which defines and references the "GeoKeys", as described below.

The tag is an array of unsigned SHORT values, which are primarily grouped into blocks of 4. The first 4 values are special, and contain GeoKey directory header information. The header values consist of the following information, in order:

Header=\{KeyDirectoryVersion, KeyRevision, MinorRevision, NumberOfKeys} where

- *KeyDirectoryVersion* indicates the current version of Key implementation, and will only change if this Tag’s Key structure is changed. (Similar to the TIFFVersion (42)). The current DirectoryVersion number is 1. This value will most likely never change, and may be used to ensure that this is a valid Key-implementation.

- *KeyRevision* indicates what revision of Key-Sets are used.
- *MinorRevision* indicates what set of Key-codes are used. The complete revision number is denoted <KeyRevision>.<MinorRevision>
- *NumberOfKeys* indicates how many Keys are defined by the rest of this Tag.

This header is immediately followed by a collection of <NumberOfKeys> KeyEntry sets, each of which is also 4-SHORTS long. Each KeyEntry is modeled on the "TIFFEntry" format of the TIFF directory header, and isof the form:

- KeyEntry = { KeyID, TIFFTagLocation, Count, Value_Offset } where
- *KeyID* gives the key-ID value of the Key (identical in function to TIFF tag ID, but completely independent of TIFF tag-space),
- *TIFFTagLocation* indicates which TIFF tag contains the value(s) of the Key: if TIFFTagLocation is 0, then the value is SHORT, and is contained in the "Value_Offset" entry. Otherwise, the type (format) of the value is implied by the TIFF-Type of the tag containing the value.
- *Count* indicates the number of values in this key.
- *Value_Offset* Value_Offset indicates the index-offset **into** the TagArray indicated by TIFFTagLocation, if it is nonzero. If TIFFTagLocation=0, then Value_Offset contains the actual (SHORT) value of the Key, and Count=1 is implied. **Note that the offset is not a byte-offset, but rather an index based on the natural data type of the specified tag array.**

Following the KeyEntry definitions, the KeyDirectory tag may also contain additional values. For example, if a Key requires multiple SHORT values, they shall be placed at the end of this tag, and the KeyEntry will set TIFFTagLocation=GeoKeyDirectoryTag, with the Value_Offset pointing to the location of the value(s).

All key-values which are not of type SHORT are to be stored in one of the following two tags, based on their format:

```
GeoDoubleParamsTag:
    Tag = 34736 (87BO.H)
    Type = DOUBLE (IEEE Double precision)
    N = variable
```

This tag is used to store all of the DOUBLE valued GeoKeys, referenced by the GeoKeyDirectoryTag. The meaning of any value of this double array is determined from the GeoKeyDirectoryTag reference pointing to it. FLOAT values should first be converted to DOUBLE and stored here.

```
GeoAsciiParamsTag:
    Tag = 34737 (87B1.H)
    Type = ASCII
    N = variable
```

This tag is used to store all of the ASCII valued GeoKeys, referenced by the GeoKeyDirectoryTag. Since keys use offsets into tags, any special comments may be placed at the beginning of this tag. For the most part, the only keys that are ASCII valued are "Citation" keys, giving documentation and references for obscure projections, datums, etc.

Note on ASCII Keys:

Special handling is required for ASCII-valued keys. While it is true that TIFF 6.0 permits multiple NULL-delimited strings within a single ASCII tag, the secondary strings might not appear in the output of naive "tiffdump" programs. For this reason, the null delimiter of each ASCII Key value shall be converted to a "|" (pipe) character before being installed back into the ASCII holding tag, so that a dump of the tag will look like this.

```
AsciiTag="first_value|second_value|etc...last_value|"
```

A baseline GeoTIFF-reader must check for and convert the final "|" pipe character of a key back into a NULL before returning it to the client software.

GeoKey Sort Order:

In the TIFF spec it is required that TIFF tags be written out to the file in tag-ID sorted order. This is done to avoid forcing software to perform N-squared sort operations when reading and writing tags.

To follow the TIFF philosophy, GeoTIFF-writers shall store the GeoKey entries in key-sorted order within the GeoKeyDirectoryTag.

```
Example:
  GeoKeyDirectoryTag=( 1, 1, 2, 6,
                    1024, 0, 1, 2,
                    1026, 34737,12, 0,
                    2048, 0, 1, 32767,
                    2049, 34737,14, 12,
                    2050, 0, 1, 6,
                    2051, 34736, 1, 0 )
  GeoDoubleParamsTag(34736)=(1.5)
  GeoAsciiParamsTag(34737)=("Custom File|My Geographic|")
```

The first line indicates that this is a Version 1 GeoTIFF GeoKey directory, the keys are Rev. 1.2, and there are 6 Keys defined in this tag.

The next line indicates that the first Key (ID=1024 = GTModelTypeGeoKey) has the value 2

(Geographic 2D), explicitly placed in the entry list (since TIFFTagLocation=0). The next line indicates that the Key 1026 (the GTCitationGeoKey) is listed in the GeoAsciiParamsTag (34737) array, starting at offset 0 (the first in array), and running for 12 bytes and so has the value "Custom File" (the "|" is converted to a null delimiter at the end). Going further down the list, the Key 2051 (GeogLinearUnitSizeGeoKey) is located in the GeoDoubleParamsTag (34736), at offset 0 and has the value 1.5; the value of key 2049 (GeogCitationGeoKey) is "My Geographic".

The TIFF layer handles all the problems of data structure, platform independence, format types, etc, by specifying byte-offsets, byte-order format and count, while the Key describes its key values at the TIFF level by specifying Tag number, array-index, and count. Since all TIFF information occurs in TIFF arrays of some sort, we have a robust method for storing anything in a Key that would occur in a Tag.

With this Key-value approach, there are 65536 Keys which have all the flexibility of TIFF tag, with the added advantage that a TIFF dump will provide all the information that exists in the GeoTIFF implementation.

This GeoKey mechanism is used extensively in Clause 7 where the parameters for defining Coordinate Systems and their underlying Coordinate Reference Systems (and projection if applicable) are defined.

B.2. GeoTIFF Models of the earth

B.2.1. Ellipsoid

The geoid - the earth stripped of all topography - forms a reference surface for the earth. However, because it is related to the earth's gravity field, the geoid is a very complex surface; indeed, at a detailed level its description is not well known. The geoid is therefore not used in practical mapping.

It has been found that an oblate ellipsoid (an ellipse rotated about its minor axis) is a good approximation to the shape of the geoid and therefore a good model of the earth. Many approximations exist: several hundred ellipsoids have been defined for scientific purposes and about 30 are in day-to-day use for Earth mapping. The size and shape of these bi-axial ellipsoids can be defined through two parameters. GeoTIFF requires one of these to be

the semi-major axis (a),

and the second to be either the inverse flattening (1/f) or the semi-minor axis (b).

Other ellipsoid parameters needed for cartographic applications, for example the eccentricity, can easily be calculated from the two defining parameters. Note that GeoTIFF uses the modern geodesy convention for the symbol (b) for the semi-minor axis. No provision is made for mapping other planets in which a tri-dimensional (tri-axial) ellipsoid might be required, where (b) would represent the semi-median axis and (c) the semi-minor axis.

Historical models exist which use a spherical approximation; such models are not recommended for modern applications, but if needed the size of a model sphere may be defined by specifying identical values for the semi-major and semi-minor axes; the inverse flattening cannot be used as it becomes infinite for perfect spheres.

Numeric codes for ellipsoids regularly used for earth mapping are included in the GeoTIFF reference lists.

B.2.2. Prime Meridian

The coordinate axes of the system referencing points on an ellipsoid are called latitude and longitude. More precisely, geodetic latitude and longitude are required in this GeoTIFF standard. A discussion of the several other types of latitude and longitude is beyond the scope of this document as they are not required for conventional georeferencing.

Geodetic latitude is defined to be the angle subtended with the ellipsoid’s equatorial plane by a perpendicular through the surface of the ellipsoid from a point. Latitude is positive if north of the equator, negative if south.

Geodetic longitude is defined to be the angle measured about the minor (polar) axis of the ellipsoid from a prime meridian (see below) to the meridian through a point, positive if east of the prime meridian and negative if west. Unlike latitude, which has a natural origin at the equator, there is no feature on the ellipsoid which forms a natural origin for the measurement of longitude. The zero longitude can be any defined meridian. Historically, nations have used the meridian through their national astronomical observatories, giving rise to several prime meridians. By international convention, the meridian through Greenwich, England is the standard prime meridian. Longitude is only unambiguous if the longitude of its prime meridian relative to Greenwich is given. Prime meridians other than Greenwich that are sometimes used for earth mapping are included in the GeoTIFF reference lists.

B.2.3. Geodetic Datum (Geodetic Reference Frame)

As well as there being several ellipsoids in use to model the earth, any one particular ellipsoid can have its location and orientation relative to the earth defined in different ways. If the relationship between the ellipsoid and the earth is changed, then the coordinates of a point will change. Conversely, for coordinates to uniquely describe a location, the relationship between the earth and the ellipsoid must be defined. This relationship is described by a geodetic datum or geodetic reference frame. An exact geodetic definition of geodetic datums and reference frames is beyond the scope of GeoTIFF. However the GeoTIFF standard requires that:

1. if geocentric/geographic/projected CRS is in the GeoTIFF CRS register: cite the CRS code, through the GeodeticCRSGeoKey if geocentric or geographic, or if projected through the ProjectedCRSGeoKey. The geodetic datum is included in that register CRS definition.
2. if the GeoTIFF CRS register contains the geodetic datum but not the geocentric/geographic/projected CRS: cite the CRS as user-defined (32767) through the GeodeticCRSGeoKey if geocentric or geographic, or through the ProjectedCRSGeoKey if projected, and then cite the geodetic datum code from the CRS register through the GeodeticDatumGeoKey.
3. if geodetic datum is not in the GeoTIFF CRS register (so the GeoTIFF CRS register will not have any CRSs associated with the datum): cite the CRS as user-defined (32767) through the GeodeticCRSGeoKey if geocentric or geographic, or through the ProjectedCRSGeoKey if

projected, and then cite the geodetic datum code as user-defined through the GeodeticDatumGeoKey. Requirement http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.user-defined then requires that the GeodeticCitationGeoKey, PrimeMeridianGeoKey and EllipsoidGeoKey be populated". The definition of the user-defined geodetic datum in addition to its ellipsoid and prime meridian should be described through the geodetic citation.

B.3. Coordinate Reference Systems in GeoTIFF

In the TIFF/GeoTIFF framework, there are essentially three different spaces in which coordinates may be defined. The spaces are:

1. The raster space (Image space) R, used to reference the pixel values in an image,
2. The Device space D, and
3. The Model space, M, used to reference points on the earth.

In the sections that follow we shall discuss the relevance and use of each of these spaces, and their corresponding coordinate systems, from the standpoint of GeoTIFF.

B.3.1. Device Space and GeoTIFF

In standard TIFF 6.0 there are tags that relate raster space R with device space D, such as monitor, scanner or printer. The list of such tags consists of the following:

```
ResolutionUnit (296)
XResolution (282)
YResolution (283)
Orientation (274)
XPosition (286)
YPosition (287)
```

In GeoTIFF, provision is made to identify earth-referenced coordinate systems (model space M) and to relate M space with R space. This provision is independent of and can co-exist with the relationship between raster and device spaces. To emphasize the distinction, this spec shall not refer to "X" and "Y" raster coordinates, but rather to raster space "J" (row) and "I" (column) coordinate variables instead, as defined in section *Raster Space*.

B.3.2. Raster Space

Raster Data

Raster data consists of spatially coherent, digitally stored numerical data, collected from sensors, scanners, or in other ways numerically derived. The manner in which this storage is implemented in a TIFF file is described in the standard TIFF specification (see TIFF Specification Revision 6.0).

Raster data values, as read in from a file, are organized by software into two-dimensional arrays, the indices of the arrays being used as coordinates. There may also be additional indices for multispectral data, but these indices do not refer to spatial coordinates but spectral, and so are not of concern here.

Many different types of raster data may be georeferenced, and there may be subtle ways in which the nature of the data itself influences how the coordinate system (Raster Space) is defined for raster data. For example, pixel data derived from imaging devices and sensors represent aggregate values collected over a small, finite, geographic area, and so it is natural to define coordinate systems in which the pixel value is thought of as filling an area. On the other hand, digital elevations models may consist of discrete "postings", which may best be considered as point measurements at the vertices of a grid, and not in the interior of a cell.

Raster Space

The choice of origin for raster space is not entirely arbitrary, and depends upon the nature of the data collected. Raster space coordinates shall be referred to by their pixel types, i.e., as "PixelIsArea" or "PixelIsPoint".

Note: For simplicity, both raster spaces documented below use a fixed pixel size and spacing of 1. Information regarding the visual representation of this data, such as pixels with non-unit aspect ratios, scales, orientations, etc., are best communicated with the TIFF 6.0 standard tags.

"PixelIsArea" Raster Space

The "PixelIsArea" raster grid space R, which is the default, uses coordinates I and J, with (0,0) denoting the upper-left corner of the image, and increasing I to the right, increasing J down. The first pixel-value fills the square grid cell with the bounds:

```
top-left = (0,0), bottom-right = (1,1)
```

and so on; by extension this one-by-one grid cell is also referred to as a pixel. An N by M pixel image covers an area with the mathematically defined bounds (0,0), (N,M).

```
(0,0)
 +---+---+-> I
 | * | * |
 +---+---+       Standard (PixelIsArea) TIFF Raster space R,
 | (1,1)   (2,1) showing the areas (*) of several pixels.
 |
 J
```

"PixelIsPoint" Raster Space

The PixelIsPoint raster grid space R uses the same coordinate axis names as used in PixelIsArea Raster space, with increasing I to the right, increasing J down. The first pixel-value however, is realized as a point value located at (0,0). An N by M pixel image consists of points which fill the mathematically defined bounds (0,0), (N-1,M-1).

```
(0,0) (1,0)
 *-------*------> I
 |       |
 |       |      PixelIsPoint TIFF Raster space R,
 *-------*      showing the location (*) of several pixels.
 | (1,1)
 J
```

If a point-pixel image were to be displayed on a display device with pixel cells having the same size as the raster spacing, then the upper-left corner of the displayed image would be located in raster space at (-0.5, -0.5).

B.3.3. Model Coordinate Reference Systems (Model space)

'Real world' coordinate reference systems are imposed on models of the earth, hence the term *model coordinate reference system* used in GeoTIFF. To georeference an image in GeoTIFF, you must specify a Raster Space coordinate system, choose a Model coordinate reference system, and specify a transformation between these two, as described in B.6.

To describe a location uniquely, a coordinate must be referenced to an adequately defined Model coordinate reference system. If the Model coordinate reference system is from the GeoTIFF standard definitions (B.5.4), the only reference required is the standard coordinate reference system code. If the Model coordinate reference system is non-standard, it must be defined.

The following subtypes of Model coordinate reference system (CRS) are recognized in GeoTIFF:

- Geographic
- Geocentric
- Projected ('map grid')
- Vertical

Projected ('map grid') and geographical 2D CRSs form two-dimensional horizontal coordinate systems (i.e. horizontal with respect to the earth's surface). Height is not part of these systems. To describe a position in three dimensions using these 2D systems it is necessary to consider height as a second one-dimensional vertical coordinate reference system in a 2D + 1D pseudo 3D compound CRS structure. Recommendations for describing compound CRSs are given in Annex D.

True spatial 3D CRS subtypes are geocentric and geographic 3D. See Annex D for recommendations for describing geographic 3D CRSs.

Geodetic Coordinate Reference Systems

A geodetic coordinate reference system is created by associating a coordinate system - a set of axes - with a geodetic datum. Subtypes of geodetic CRS supported by GeoTIFF are:

- geocentric, when the coordinate system is a 3-dimensional Cartesian coordinate system with its origin at or near the centre of the earth. The Z-axis is in or parallel to the earth's axis of rotation (or to the axis around which the rotational axis precesses). The X-axis is in the plane of the equator and passes through its intersection with the prime meridian, and the Y-axis is in the plane of the equator forming a right-handed coordinate system with the X and Z axes.

- geographic, when the coordinate system is ellipsoidal, i.e. latitude and longitude in the 2D case and in the 3D case additionally with ellipsoidal height. GeoTIFF v1.0 did not clearly define whether geographic CRSs are 2D or 3D.

Geocentric coordinates are readily converted to and from geographic 3D coordinates. Geographic 2D coordinates cannot be converted to geocentric coordinates without some assumption regarding height.

Projected Coordinate Reference Systems

Before digital computing capabilities were available, calculation on the surface of an ellipsoid was a non-trivial task. Reduction of the ellipsoid surface to a plane facilitated spatial calculations. A geographical coordinate reference system cannot be represented on a plane surface without distortion. Map projections are conversions of ellipsoidal coordinates (latitude and longitude) to Cartesian coordinates in a plane. A map projection consists of a coordinate operation method (through which the characteristics of the distortions are controlled) and a set of defining parameters specific to the method which are parameters in the method formulas, together with specified values for the set of coordinate operation parameters required by the projection method. A projected coordinate reference system results from the application of a map projection to a geographic coordinate reference system, associated with a planar coordinate system, in practice almost always Cartesian.

Vertical Coordinate Reference Systems

Many uses of GeoTIFF will be limited to a two-dimensional, horizontal, description of location for which geographic 2D coordinate reference systems and projected 2D coordinate reference systems are adequate. If a three-dimensional description of location is required, GeoTIFF allows this either through a geocentric coordinate reference system, or through the use of a geographic 3D coordinate reference system (where the vertical component is height above the ellipsoid), or by defining a 1D vertical coordinate reference system and using this together with a geographic 2D or projected coordinate reference system in an implicit compound CRS structure. Vertical CRS are referenced to a vertical reference surface (vertical datum) at or close to the geoid, and associated with a 1D vertical coordinate system in which heights or depths are given. Through increasing use of satellite navigation and positioning systems the ellipsoid is increasingly being used as a vertical reference surface. Heights above the ellipsoid are expressed as part of a geographic 3D CRS, but are not referenced to a vertical CRS (see Annex D). The ellipsoid surface may be offset vertically from the reference surface for a vertical CRS approximating the geoid by up to +/- 100m, and generally the two surfaces will not be exactly parallel to each other.

B.4. Defining Geographic Coordinate Reference Systems

Within the GeoTIFF standard a Model coordinate reference system (geocentric, geographic, projected or vertical) can be identified either by

```
the code of a standard coordinate reference system
```

or by

```
a user-defined system.
```

B.4.1. Standard Model Coordinate Reference Systems

In GeoTIFF, standard CRSs are identified through reference to an EPSG CRS code. This is sufficient to define the CRS component objects. Further information on EPSG codes is given in B.5.4.

NOTE	This document removes the reference to the specific EPSG codes listed in the 1995 GeoTIFF v1.0 specification and replaces it by **allowing reference to any code in the EPSG Dataset**, including codes for any objects introduced into the EPSG Dataset after publication of this document.

B.4.2. User-defined Model Coordinate Reference Systems

GeoTIFF attempts to allow Model CRSs that are not described in the standard CRS register to be defined through user-defined keys. However the provisions made are limited in that:

- no provision was made for fully describing coordinate system. Although axis units could be described, provision for describing axis order and positive direction was omitted.
- there is ambiguity in the provision for describing user-defined map projections. Codes for some common map projection methods and map projection parameters were provided, but neither the method nor the parameter were defined. Inferences may be made from the listed map projection method names and map projection parameter names, but ambiguity remains so interoperability is not guaranteed.

In practice, user-defined Model CRS definition is limited to the following cases:

1. a user-defined projected CRS which uses a base geographic CRS and a map projection that are both individually available from the GeoTIFF CRS register but, in the register, not associated together. EPSG geogCRS code needs citing through Requirement http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.EPSG, EPSG projection code needs citing through Requirement http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.EPSG
2. a user-defined projected CRS which uses a user-defined geographic CRS with a map projection that is available from the GeoTIFF CRS register. GeogCRS needs defining as in Requirement http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.userdefined, EPSG projection code needs citing through Requirement http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.EPSG
3. a user-defined geographic CRS available from the GeoTIFF CRS register and a map projection not in EPSG register. EPSG geogCRS code needs citing through Requirement http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticCRSGeoKey.EPSG, projection needs defining through Requirement http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.user_defined

using the v1.0 provisions (use the names in annex C).

4. Neither base GeogCRS or map projection is in EPSG. GeogCRS needs defining, projection needs defining through Requirement http://www.opengis.net/spec/GeoTIFF/0.0/ProjectionGeoKey.user_defined using the v1.0 provisions (the names in annex C).

But even for these cases, unless stated through a Citation, some assumption is required about axis positive direction and axis order. Based on the standards available for US FGDC metadata and in the EPSG Dataset when the GeoTIFF v1.0 specification was written, it is assumed that:

- projected CRS axes are easting, northing;
- geographic 2D CRS axes are longitude east, latitude north;

NOTE	Users must note that this GeoTIFF practise is not in line with ISO TC211 and OGC rules for geographic 2D CRS axis order.

- vertical CRS axis is height up;

and that values for the transformation from raster space to model space are consistent with these.

User-defined geographic 2D CRS

For a user-defined geographic 2D CRS the user is expected to provide:

- geocentric coordinate reference system name (through the *GeodeticCitationGeoKey*),
- geodetic datum through the *GeodeticDatumGeoKey*, either
  - the geodetic datum code (if available through standard EPSG code), or
  - user-defined geodetic datum name and other defining information:
    - the geodetic datum name (through the *GeodeticCitationGeoKey*),
    - the ellipsoid (through the *EllipsoidGeoKey*, see User-defined ellipsoid), and
    - the prime meridian (through the *PrimeMeridianGeoKey*, see User-defined prime meridian)
- axis unit through the *GeogAngularUnitsGeoKey*, either
  - angle unit code (if available through standard EPSG code), or
  - user-defined angle unit name (through the *GeodeticCitationGeoKey*) and scaling from SI base unit of radian (through the *GeogAngularUnitSizeGeoKey*).
- if the CRS uses a user-defined ellipsoid, the ellipsoid axis unit through the *GeogLinearUnitsGeoKey*, either
  - length unit code (if available through standard EPSG code), or
  - user-defined length unit name (through the *GeodeticCitationGeoKey*) and scaling from SI base unit of meter (through the *GeogLinearUnitSizeGeoKey*).

User-defined geocentric CRS

For a user-defined geocentric CRS the user is expected to provide:

- geocentric coordinate reference system name (through the *GeodeticCitationGeoKey*),
- geodetic datum through the *GeodeticDatumGeoKey*, either
  - the geodetic datum code (if available through standard EPSG code), or
  - user-defined geodetic datum name and other defining information:
    - the geodetic datum name (through the *GeodeticCitationGeoKey*),
    - the ellipsoid (through the *EllipsoidGeoKey*, see User-defined ellipsoid), and
    - the prime meridian (through the *PrimeMeridianGeoKey*, see User-defined prime meridian)
- axis unit through the *GeogLinearUnitsGeoKey*, either
  - length unit code (if available through standard EPSG code), or
  - user-defined length unit name (through the *GeodeticCitationGeoKey*) and scaling from SI base unit of meter (through the *GeogLinearUnitSizeGeoKey*).
- if the CRS uses a user-defined prime meridian, prime meridian Greenwich longitude unit through the *GeogAngularUnitsGeoKey*, either
  - angle unit code (if available through standard EPSG code), or
  - user-defined angle unit name (through the *GeodeticCitationGeoKey*) and scaling from SI base unit of radian (through the *GeogAngularUnitSizeGeoKey*).

User-defined ellipsoid

For any user-defined geocentric, geographic 3D or geographic 2D CRS an ellipsoid needs to be identified. The user is expected to provide:

- ellipsoid through the *EllipsoidGeoKey*, either
  - the ellipsoid code (if available through standard EPSG code), or
  - the user-defined ellipsoid name and other defining information:
    - the ellipsoid name (through the *GeodeticCitationGeoKey*),
    - the ellipsoid semi-major axis (through the *EllipsoidSemiMajorAxisGeoKey*)
    - either the ellipsoid semi-minor axis (through the *EllipsoidSemiMinorAxisGeoKey*) or the ellipsoid inverse flattening (through the *EllipsoidInvFlatteningGeoKey*).
    - The units for the ellipsoid axis or axes:
- For geocentric the ellipsoid axis or axes values must given in the length unit defined through the *GeogLinearUnitsGeoKey* already required (see User-defined geocentric CRS).
- For geographic 2D CRSs, then a *GeogLinearUnitsGeoKey* is additionally required.

User-defined prime meridian

For any user-defined geocentric, geographic 3D or geographic 2D CRS a prime meridian needs to be identified whenever it is not Greenwich. (If no prime meridian is identified, it should be assumed to be Greenwich). The user is expected to provide:

- Prime meridian through the *PrimeMeridianGeoKey*, either
  - the prime meridian code (if available through standard EPSG code), or
  - the user-defined prime meridian name and other defining information:
    - the prime meridian name (through the *GeodeticCitationGeoKey*),
    - the prime meridian longitude (through the *PrimeMeridianLongitudeGeoKey*),
    - the units for the prime meridian longitude:
- For geographic CRSs the prime meridian longitude value must given in the angle unit defined through the *GeogAngularUnitsGeoKey* already required (see User-defined geographic 2D CRS).
- For geocentric CRSs, then a *GeogAngularUnitsGeoKey* is additionally required.

User-defined Projected Coordinate Reference Systems

For a user-defined projected CRS the user is expected to provide:

- projected coordinate reference system name (through *ProjectedCitationGeoKey*),
- base geographic CRS (either standard EPSG code or user-defined, see User-defined geographic 2D CRS)
- map projection through the *ProjectionGeoKey*, either
  - map projection code (if available through standard EPSG code), or
  - user-defined map projection (see below).
- axis unit through *ProjLinearUnitsGeoKey*, either
  - length unit code (if available through standard EPSG code), or
  - user-defined length unit name (through the *ProjectedCitationGeoKey*) and scaling from SI base unit of meter (through the *ProjLinearUnitSizeGeoKey*).

User-defined map projection

For a user-defined map projection the user is expected to provide:

- map projection name (through *ProjectedCitationGeoKey*),
- map projection method (through *ProjMethodGeoKey*),
- map projection parameter values (using a set of keys appropriate to the map projection method).
  - For map projection parameters that are lengths the parameter value needs to be expressed in the units defined through the *ProjLinearUnitsGeoKey*.
  - For map projection parameters that are angles the parameter value needs to be expressed in the units defined through the *GeogAngularUnitsGeoKey*, which is required in the base geographic CRS description, except for azimuths when the value needs to be expressed in the units defined through a *GeogAzimuthUnitsGeoKey*.
- if the map projection method requires a parameter that is an azimuth, the azimuth unit through a *GeogAzimuthUnitsGeoKey*.

User-defined Vertical Coordinate Reference Systems

For a user-defined vertical CRS the user is expected to provide:

*vertical coordinate reference system name (through *VerticalCitationGeoKey*), * user-defined vertical datum through *VerticalDatumGeoKey*, either **the vertical datum code (if available through standard EPSG code), or** the vertical datum name and other defining information (through the *VerticalCitationGeoKey*) * vertical axis unit through *VerticalUnitsGeoKey*, either **linear unit code (if available through standard EPSG code), or** linear unit name (through *VerticalCitationGeoKey*) and scaling from SI base unit of meter (through *GeogLinearUnitSizeGeoKey*).

B.5. Coordinate Transformations

The purpose of GeoTIFF is to allow the definitive identification of georeferenced locations within a raster dataset. This is generally accomplished through tying raster space coordinates to a model space coordinate system, when no further information is required. In the GeoTIFF nomenclature, "georeferencing" refers to tying raster space to a model space M, while "geocoding" refers to defining how the model space M assigns coordinates to points on the earth.

The three tags defined below may be used for defining the relationship between R and M, and the relationship may be diagrammed as:

```
        ModelPixelScaleTag
         ModelTiepointTag
 R ------------ OR --------------> M
(I,J,K) ModelTransformationTag (X,Y,Z)
```

The next section describes these Baseline georeferencing tags in detail.

B.5.1. GeoTIFF Tags for Coordinate Transformations

For most common applications, the transformation between raster and model space may be defined with a set of raster-to-model tiepoints and scaling parameters. The following two tags may be used for this purpose:

```
ModelTiepointTag:
      Tag = 33922 (8482.H)
      Type = DOUBLE (IEEE Double precision)
      N = 6*K, K = number of tiepoints
      Alias: GeoreferenceTag
```

This tag stores raster→model tiepoint pairs in the order

```
ModelTiepointTag = (...,I,J,K, X,Y,Z...),
```

where (I,J,K) is the point at location (I,J) in raster space with pixel-value K, and (X,Y,Z) is a vector in model space. In most cases the model space is only two-dimensional, in which case both K and Z should be set to zero; this third dimension is provided in anticipation of support for 3D digital elevation models and vertical coordinate systems.

A raster image may be georeferenced simply by specifying its location, size and orientation in the model coordinate space M. This may be done by specifying the location of three of the four bounding corner points. However, tiepoints are only to be considered exact at the points specified; thus defining such a set of bounding tiepoints does **not** imply that the model space locations of the interior of the image may be exactly computed by a linear interpolation of these tiepoints.

However, since the relationship between the Raster space and the model space will often be an exact, affine transformation, this relationship can be defined using one set of tiepoints and the "ModelPixelScaleTag", described below, which gives the vertical and horizontal raster grid cell size, specified in model units.

If possible, the first tiepoint placed in this tag shall be the one establishing the location of the point (0,0) in raster space. However, if this is not possible (for example, if (0,0) is goes to a part of model space in which the projection is ill-defined), then there is no particular order in which the tiepoints need be listed.

For orthorectification or mosaicking applications a large number of tiepoints may be specified on a mesh over the raster image. However, the definition of associated grid interpolation methods is not in the scope of the current GeoTIFF spec.

Remark: As mentioned in section 2.5.1, all GeoTIFF information is independent of the XPosition, YPosition, and Orientation tags of the standard TIFF 6.0 spec.

The next two tags are optional tags provided for defining exact affine transformations between raster and model space; baseline GeoTIFF files may use either, but shall never use both within the same TIFF image directory.

```
ModelPixelScaleTag:
     Tag = 33550
     Type = DOUBLE (IEEE Double precision)
     N = 3
```

This tag may be used to specify the size of raster pixel spacing in the model space units, when the raster space can be embedded in the model space coordinate system without rotation, and consists of the following 3 values:

```
ModelPixelScaleTag = (ScaleX, ScaleY, ScaleZ)
```

where ScaleX and ScaleY give the horizontal and vertical spacing of raster pixels. The ScaleZ is primarily used to map the pixel value of a digital elevation model into the correct Z-scale, and so for most other purposes this value should be zero (since most model spaces are 2-D, with Z=0).

A single tiepoint in the ModelTiepointTag, together with this tag, completely determine the relationship between raster and model space; thus they comprise the two tags which Baseline GeoTIFF files most often will use to place a raster image into a "standard position" in model space.

Like the Tiepoint tag, this tag information is independent of the XPosition, YPosition, Resolution and Orientation tags of the standard TIFF 6.0 spec. However, simple reversals of orientation between raster and model space (e.g. horizontal or vertical flips) may be indicated by reversal of sign in the corresponding component of the ModelPixelScaleTag. GeoTIFF compliant readers must honor this sign-reversal convention.

This tag must not be used if the raster image requires rotation or shearing to place it into the standard model space. In such cases the transformation shall be defined with the more general ModelTransformationTag, defined below.

```
ModelTransformationTag
     Tag = 34264 (85D8.H)
     Type = DOUBLE
     N = 16
```

This tag may be used to specify the transformation matrix between the raster space (and its dependent pixel-value space) and the (possibly 3D) model space. If specified, the tag shall have the following organization:

```
ModelTransformationTag = (a,b,c,d,e....m,n,o,p).
```

where

```
model                  image
coords =     matrix  * coords
|- -|     |-       -|  |- -|
| X |     | a b c d |  | I |
| | |     |         |  |   |
| Y |     | e f g h |  | J |
|   |  =  |         |  |   |
| Z |     | i j k l |  | K |
| | |     |         |  |   |
| 1 |     | m n o p |  | 1 |
|- -|     |-       -|  |- -|
```

By convention, and without loss of generality, the following parameters are currently hard-coded and will always be the same (but must be specified nonetheless):

```
m = n = o = 0, p = 1.
```

For Baseline GeoTIFF, the model space is always 2-D, and so the matrix will have the more limited form:

```
|- -|   |-       -| |- -|
| X |   | a b 0 d | | I |
| | |   |         | |   |
| Y |   | e f 0 h | | J |
|   | = |         | |   |
| Z |   | 0 0 0 0 | | K |
| | |   |         | |   |
| 1 |   | 0 0 0 1 | | 1 |
|- -|   |-       -| |- -|
```

Values "d" and "h" will often be used to represent translations in X and Y, and so will not necessarily be zero. All 16 values should be specified, in all cases. Only the raster-to-model transformation is defined; if the inverse transformation is required it must be computed by the client, to the desired accuracy.

This matrix tag should not be used if the ModelTiepointTag and the ModelPixelScaleTag are already defined. If only a single tiepoint (I,J,K,X,Y,Z) is specified, and the ModelPixelScale = (Sx, Sy, Sz) is specified, then the corresponding transformation matrix may be computed from them as:

```
|-               -|
| Sx  0.0 0.0 Tx  |    Tx = X - I/Sx
| 0.0 -Sy 0.0 Ty  |    Ty = Y + J/Sy
| 0.0 0.0 Sz  Tz  |    Tz = Z - K/Sz (if not 0)
| 0.0 0.0 0.0 1.0 |
|-               -|
```

where the -Sy is due the reversal of direction from J increasing- down in raster space to Y increasing-up in model space.

Like the Tiepoint tag, this tag information is independent of the XPosition, YPosition, and Orientation tags of the standard TIFF 6.0 spec.

B.5.2. Cookbook for Defining Transformations

Here is a 4-step guide to producing a set of Baseline GeoTIFF tags for defining coordinate transformation information of a raster dataset.

```
Step 1: Establish the Raster Space coordinate system used: RasterPixelIsArea or
RasterPixelIsPoint.
```

```
Step 2: Establish/define the model space Type in which the image is to be
georeferenced. Usually this will be a projected coordinate Reference system. If you
are geocoding this data set, then the model space is defined to be the corresponding
geographic, geocentric or projected coordinate reference system.
```

```
Step 3: Identify the nature of the transformations needed to tie the raster data down
to the model space coordinate reference system:
```

```
Case 1: The model-location of a raster point (x,y) is known, but not the scale or
orientations: Use the ModelTiepointTag to define the (X,Y,Z) coordinates of the known
raster point.
```

```
Case 2: The location of three non-collinear raster points are known exactly, but the
linearity of the transformation is not known.
```

```
Use the ModelTiepointTag to define the (X,Y,Z) coordinates of all three known raster
points. Do not compute or define the ModelPixelScale or ModelTransformation tag.
```

```
Case 3: The position and scale of the data is known exactly, and no rotation or
shearing is needed to fit into the model space.
```

```
Use the ModelTiepointTag to define the (X,Y,Z) coordinates of the known raster point,
and the ModelPixelScaleTag to specify the scale.
```

```
Case 4: The raster data requires rotation and/or lateral shearing to fit into the
defined model space:
```

```
Use the ModelTransformation matrix to define the transformation.
```

```
Case 5: The raster data cannot be fit into the model space with a simple affine
transformation (rubber-sheeting required). Use only the ModelTiepoint tag, and specify
as many tiepoints as your application requires. Note, however, that this is not a
Baseline GeoTIFF implementation, and should not be used for interchange; it is
recommended that the image be geometrically rectified first, and put into a standard
projected coordinate reference system.
```

```
Step 4: Install the defined tag values in the TIFF file and close it.
```

B.6. Geocoding Raster Data

A geocoded image is a georeferenced image as described in section Coordinate Transformations, which also specifies a model space coordinate reference system (CRS) between the model space M (to which the raster space has been tied) and the earth. The relationship can be diagrammed, including the associated TIFF tags, as follows:

```
   ModelPixelScaleTag
   ModelTiepointTag             GeoKeyDirectoryTag CRS
R -------- OR ---------------> M --------- AND -----------> Earth
   ModelTransformationTag       GeoDoubleParamsTag
                                GeoAsciiParamsTag
```

The geocoding coordinate system is defined by the GeoKeyDirectoryTag, while the Georeferencing information (T) is defined by the ModelTiepointTag and the ModelPixelScale, or ModelTransformationTag. Since these two systems are independent of each other, the tags used to store the parameters are separated from each other in the GeoTIFF file to emphasize the orthogonality.

B.7. Model CRS Reference Parameters

Most of the GeoTIFF standard definitions for model ('real world') coordinate reference systems and their component elements are based on the hierarchical system developed for the EPSG Geodetic Parameter Dataset ('EPSG Dataset'). The complete set of EPSG definitions is available at http://www.epsg-registry.org.

The EPSG Dataset is maintained by the Geodesy Subcommittee of the International Association of Oil and Gas Producers (IOGP). It follows the ISO 19111 / OGC Abstract Specification Topic 2 data model for describing the definitions of coordinate reference systems (CRSs). CRSs and coordinate operations are composed of a number of objects and attributes. Some of these objects themselves are composed objects and attributes, in a nested structure. Each release of new or revised data is indicated by the EPSG Dataset version number. Since 1999 (from EPSG Dataset v5.0 and later) EPSG policy has been to never remove any invalid data but instead to leave it in the Dataset with its status set to deprecated. Deprecated data contains a significant error (significant defined as having impact on the result of applying a transformation or conversion) and is invalid. As such, since 1999 reference to the version of the EPSG Dataset to qualify codes of entities within the Dataset has been unnecessary. Using EPSG Dataset versions 5.0 and 9.3 as examples, crs:EPSG:5.0:4326 and crs:EPSG:9.3:4326 and crs:EPSG::4326 reference the same object. The terms of use of the EPSG Dataset are given at http://www.epsg.org/Termsofuse.aspx.

B.7.1. EPSG coding of objects

Within the EPSG Dataset each object has a code. There have been three generations of coding:

1. In v1.x of the publicly-available EPSG Dataset (1994-1996, published by the Petrotechnical Open Software Corporation, POSC), codes were alphanumeric. The initial letter indicated the object type, and objects within each type were then assigned sequential numbers.
2. With the introduction of GeoTIFF v1.0, EPSG Dataset v2.1 object codes were changed to integer values in the range 1024 through 32766. This overall code range was divided into non-overlapping sub-ranges, with one sub-range range for each object type. At that time, all EPSG object codes were unique. The GeoTIFF v1.0 specification was written at this time, and the EPSG

code ranges for object types were written into the GeoTIFF v1.0 specification.

3. However as the number of items in the EPSG Dataset grew, some of the object code sub-ranges became fully assigned. The unique code system broke down. Since 2006, all object types have been separately assigned codes within the range 1024 through 32766. Within each object type codes remain unique, but one code value may be used for several object types. For example, code 4326 is used for both a CRS and for a geographic extent (in EPSG called 'area'). Codes at and just above the lower end of the range 1024 through 32766 may be used by numerous object types: for example by the year 2018 code 1026 has been assigned to 10 different object types. EPSG codes therefore are only unique when the object type is disclosed. EPSG::4326 is ambiguous, crs:EPSG::4326 and area:EPSG::4326 are unambiguous.

The GeoTIFF v1.0 specification refers to "obsolete EPSG/POSC codes". These refer to the numeric part of the alphanumeric coding in (i) above. These values had been used in some GeoTIFF v0.x files and for backward compatibility with those earlier files GeoTIFF v1.0 retained references to them. As all of these alphanumeric codes were changed to the integer coding in (ii) above, reference to these obsolete codes should now be unnecessary. In effect, for model CRS GeoKeys the obsolete code range may be treated as a reserved code range. Note: 'EPSG/POSC obsolete codes' refers specifically to the coding in generation (i) above, and should not be confused with codes from generations (ii) and (iii) which have been given the status of 'deprecated'.

A reference to an EPSG coordinate reference system code is sufficient for a complete definition: it implies use of the CRS components (datum, ellipsoid, map projection, etc.) that are associated with that CRS in the EPSG Dataset definition. The EPSG codes for coordinate reference system components should only be referenced when describing a user-defined coordinate reference system.

Deprecated and deleted EPSG codes

Geotiff v1.0 sections 6.3.2, 6.3.3 and 6.3.4 listed codes for EPSG objects available at that time. Most of these remain valid, although there may have been minor changes to names and other non-essential detail. However since the publication of GeoTIFF v1.0 several of these object records have been either deprecated or deleted from the EPSG Dataset. Deprecated records usually have been superseded by a later record, with a cross reference included in the EPSG Dataset. Deleted records have been removed from the EPSG dataset and the code could be (and in some cases has been) reused for a totally different object. Particular care is needed in the treatment of GeoTIFF files that have used these deleted object records, *shown in italics* in tables B.1 through B.6 below. These tables list the EPSG codes listed in GeoTIFF v1.0 that are no longer valid and which should no longer be used. The list was correct at 29th May 2018. For any later deprecation and replacement of records refer to the EPSG Dataset. Note that the names in the GeoTIFF v1.0 specification and given in these tables have prefixes and underscores not found in the EPSG Dataset.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
PCS_AGD66_AMG_zone_48	20248	Deprecated in EPSG, no replacement.
PCS_AGD84_AMG_zone_48	20348	Deprecated in EPSG, no replacement.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
PCS_AGD84_AMG_zone_57	*20357*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_AGD84_AMG_zone_58	20358	Deprecated in EPSG, no replacement.
PCS_Lisbon_Portugese_Grid	*20700*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo13	*20973*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo15	*20975*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo17	*20977*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo19	*20979*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo21	*20981*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo23	*20983*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo25	*20985*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo27	*20987*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo29	*20989*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo31	*20991*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo33	*20993*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Arc_1950_Lo35	*20995*	*Deleted in EPSG, code not re-used as at 2018-05-29, no replacement.*
PCS_Batavia_NEIEZ	21100	Deprecated in EPSG, replaced by 3001 Batavia / NEIEZ.
PCS_Beijing_Gauss_13N	21473	Deprecated in EPSG, replaced by 21453 Beijing 1954 / Gauss-Kruger 13N.
PCS_Beijing_Gauss_14N	21474	Deprecated in EPSG, replaced by 21454 Beijing 1954 / Gauss-Kruger 14N.
PCS_Beijing_Gauss_15N	21475	Deprecated in EPSG, replaced by 21455 Beijing 1954 / Gauss-Kruger 15N.
PCS_Beijing_Gauss_16N	21476	Deprecated in EPSG, replaced by 21456 Beijing 1954 / Gauss-Kruger 16N.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
PCS_Beijing_Gauss_17N	21477	Deprecated in EPSG, replaced by 21457 Beijing 1954 / Gauss-Kruger 17N.
PCS_Beijing_Gauss_18N	21478	Deprecated in EPSG, replaced by 21458 Beijing 1954 / Gauss-Kruger 18N.
PCS_Beijing_Gauss_19N	21479	Deprecated in EPSG, replaced by 21459 Beijing 1954 / Gauss-Kruger 19N.
PCS_Beijing_Gauss_20N	21480	Deprecated in EPSG, replaced by 21460 Beijing 1954 / Gauss-Kruger 20N.
PCS_Beijing_Gauss_21N	21481	Deprecated in EPSG, replaced by 21461 Beijing 1954 / Gauss-Kruger 21N.
PCS_Beijing_Gauss_22N	21482	Deprecated in EPSG, replaced by 21462 Beijing 1954 / Gauss-Kruger 22N.
PCS_Beijing_Gauss_23N	21483	Deprecated in EPSG, replaced by 21463 Beijing 1954 / Gauss-Kruger 23N.
PCS_Bern_1898_Swiss_Old	*21790*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 21780 Bern 1898 (Bern) / LV03C.*
PCS_Bogota_UTM_zone_17N	21817	Deprecated in EPSG, no replacement.
PCS_Bogota_Colombia_3W	21891	Deprecated in EPSG, replaced by 21896 Bogota 1975 / Colombia West zone.<
PCS_Bogota_Colombia_Bogota	21892	Deprecated in EPSG, replaced by 21897 Bogota 1975 / Colombia Bogota zone.
PCS_Bogota_Colombia_3E	21893	Deprecated in EPSG, replaced by 21898 Bogota 1975 / Colombia East Central zone.
PCS_Bogota_Colombia_6E	21894	Deprecated in EPSG, replaced by 21899 Bogota 1975 / Colombia East.
PCS_Douala_UTM_zone_32N	22832	Deprecated in EPSG, replaced by 2214 Douala 1948 / AOF west.
PCS_Garoua_UTM_zone_33N	23433	Deprecated in EPSG, replaced by 2312 Garoua / UTM zone 33N.
PCS_ID74_UTM_zone_53N	23853	Deprecated in EPSG, no replacement.
PCS_ID74_UTM_zone_46S	23886	Deprecated in EPSG, no replacement.
PCS_Kalianpur_India_IVb	*24384*	*Deleted in EPSG, code not re-used as at 2018-05-29.*
PCS_La_Canoa_UTM_zone_21N	*24721*	*Deleted in EPSG, code not re-used as at 2018-05-29.*
PCS_Mhast_UTM_zone_32S	26432	Deprecated in EPSG, replaced by 3353 Mhast (onshore) / UTM zone 32S and 3354 Mhast (offshore) / UTM zone 32S.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
PCS_Monte_Mario_Italy_1	26591	Deprecated in EPSG, replaced by 3003 Monte Mario / Italy zone 1.
PCS_Monte_Mario_Italy_2	26592	Deprecated in EPSG, replaced by 3004 Monte Mario / Italy zone 2.
PCS_NAD27_California_VII	26747	Deprecated in EPSG, replaced by 26799 NAD27 / California zone VII.
PCS_NAD27_Hawaii_zone_1	*26761*	*Deleted in EPSG, code not re-used as at 2018-05-29.*
PCS_NAD27_Hawaii_zone_2	*26762*	*Deleted in EPSG, code not re-used as at 2018-05-29.*
PCS_NAD27_Hawaii_zone_3	*26763*	*Deleted in EPSG, code not re-used as at 2018-05-29.*
PCS_NAD27_Hawaii_zone_4	*26764*	*Deleted in EPSG, code not re-used as at 2018-05-29.*
PCS_NAD27_Hawaii_zone_5	*26765*	*Deleted in EPSG, code not re-used as at 2018-05-29.*
PCS_NAD27_BLM_14N_feet	*26774*	*This GeoTIFF v1.0 entry is incorrect. 26774 is NAD27 / Indiana West. NAD27 / BLM 14N (feet) is 32074, see below.*
PCS_NAD27_BLM_15N_feet	*26775*	*This GeoTIFF v1.0 entry is incorrect. 26775 is NAD27 / Iowa North. NAD27 / BLM 15N (feet) is 32075, see below.*
PCS_NAD27_BLM_16N_feet	*26776*	*This GeoTIFF v1.0 entry is incorrect. 26776 is NAD27 / Iowa South. NAD27 / BLM 16N (feet) is 32076, see below.*
PCS_NAD27_BLM_17N_feet	*26777*	*This GeoTIFF v1.0 entry is incorrect. 26777 is NAD27 / Kansas North. NAD27 / BLM 17N (feet) is 32077, see below.*
PCS_NAD27_Michigan_North	*26788*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 6966 NAD27 / Michigan North.*
PCS_NAD27_Michigan_Central	*26789*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 6201 NAD27 / Michigan Central.*
PCS_NAD27_Michigan_South	*26790*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 6202 NAD27 / Michigan South.*
PCS_NAD_Michigan_Michigan_East	26801	Deprecated in EPSG, replaced by 5623 NAD27 Michigan / Michigan East.
PCS_NAD_Michigan_Michigan_Old_Central	26802	Deprecated in EPSG, replaced by 5624 NAD27 Michigan / Michigan Old Central.
PCS_NAD_Michigan_Michigan_West	26803	Deprecated in EPSG, replaced by 5625 NAD27 Michigan / Michigan West.
PCS_NAD83_Kentucky_North	26979	Deprecated in EPSG, replaced by 2205 NAD83 / Kentucky North.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
PCS_Nahrwan_1967_UTM_38N	27038	Deprecated in EPSG, replaced by 7006 Nahrwan 1934 / UTM zone 38N.
PCS_NTF_France_I	27581	Deprecated in EPSG, replaced by 27571 NTF (Paris) / Lambert zone I.
PCS_NTF_France_II	27582	Deprecated in EPSG, replaced by 27572 NTF (Paris) / Lambert zone II.
PCS_NTF_France_III	27583	Deprecated in EPSG, replaced by 27573 NTF (Paris) / Lambert zone III.
PCS_NTF_Nord_France	27591	Deprecated in EPSG, replaced by 27561 NTF (Paris) / Nord France.
PCS_NTF_Centre_France	27592	Deprecated in EPSG, replaced by 27562 NTF (Paris) / Centre France.
PCS_NTF_Sud_France	27593	Deprecated in EPSG, replaced by 27563 NTF (Paris) / Sud France.
PCS_Pulkovo_Gauss_4N	28464	Deprecated in EPSG, replaced by 2494 Pulkovo 1942 / Gauss-Kruger CM 21E.
PCS_Pulkovo_Gauss_5N	28465	Deprecated in EPSG, replaced by 2495 Pulkovo 1942 / Gauss-Kruger CM 27E.
PCS_Pulkovo_Gauss_6N	28466	Deprecated in EPSG, replaced by 2496 Pulkovo 1942 / Gauss-Kruger CM 33E.
PCS_Pulkovo_Gauss_7N	28467	Deprecated in EPSG, replaced by 2497 Pulkovo 1942 / Gauss-Kruger CM 39E.
PCS_Pulkovo_Gauss_8N	28468	Deprecated in EPSG, replaced by 2498 Pulkovo 1942 / Gauss-Kruger CM 45E.
PCS_Pulkovo_Gauss_9N	28469	Deprecated in EPSG, replaced by 2499 Pulkovo 1942 / Gauss-Kruger CM 51E.
PCS_Pulkovo_Gauss_10N	28470	Deprecated in EPSG, replaced by 2500 Pulkovo 1942 / Gauss-Kruger CM 57E.
PCS_Pulkovo_Gauss_11N	28471	Deprecated in EPSG, replaced by 2501 Pulkovo 1942 / Gauss-Kruger CM 63E.
PCS_Pulkovo_Gauss_12N	28472	Deprecated in EPSG, replaced by 2502 Pulkovo 1942 / Gauss-Kruger CM 69E.
PCS_Pulkovo_Gauss_13N	28473	Deprecated in EPSG, replaced by 2503 Pulkovo 1942 / Gauss-Kruger CM 75E.
PCS_Pulkovo_Gauss_14N	28474	Deprecated in EPSG, replaced by 2504 Pulkovo 1942 / Gauss-Kruger CM 81E.
PCS_Pulkovo_Gauss_15N	28475	Deprecated in EPSG, replaced by 2505 Pulkovo 1942 / Gauss-Kruger CM 87E.
PCS_Pulkovo_Gauss_16N	28476	Deprecated in EPSG, replaced by 2506 Pulkovo 1942 / Gauss-Kruger CM 93E.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
PCS_Pulkovo_Gauss_17N	28477	Deprecated in EPSG, replaced by 2507 Pulkovo 1942 / Gauss-Kruger CM 99E.
PCS_Pulkovo_Gauss_18N	28478	Deprecated in EPSG, replaced by 2508 Pulkovo 1942 / Gauss-Kruger CM 105E.
PCS_Pulkovo_Gauss_19N	28479	Deprecated in EPSG, replaced by 2509 Pulkovo 1942 / Gauss-Kruger CM 111E.
PCS_Pulkovo_Gauss_20N	28480	Deprecated in EPSG, replaced by 2510 Pulkovo 1942 / Gauss-Kruger CM 117E.
PCS_Pulkovo_Gauss_21N	28481	Deprecated in EPSG, replaced by 2511 Pulkovo 1942 / Gauss-Kruger CM 123E.
PCS_Pulkovo_Gauss_22N	28482	Deprecated in EPSG, replaced by 2512 Pulkovo 1942 / Gauss-Kruger CM 129E.
PCS_Pulkovo_Gauss_23N	28483	Deprecated in EPSG, replaced by 2513 Pulkovo 1942 / Gauss-Kruger CM 135E.
PCS_Pulkovo_Gauss_24N	28484	Deprecated in EPSG, replaced by 2514 Pulkovo 1942 / Gauss-Kruger CM 141E.
PCS_Pulkovo_Gauss_25N	28485	Deprecated in EPSG, replaced by 2515 Pulkovo 1942 / Gauss-Kruger CM 147E.
PCS_Pulkovo_Gauss_26N	28486	Deprecated in EPSG, replaced by 2516 Pulkovo 1942 / Gauss-Kruger CM 153E.
PCS_Pulkovo_Gauss_27N	28487	Deprecated in EPSG, replaced by 2517 Pulkovo 1942 / Gauss-Kruger CM 159E.
PCS_Pulkovo_Gauss_28N	28488	Deprecated in EPSG, replaced by 2518 Pulkovo 1942 / Gauss-Kruger CM 165E.
PCS_Pulkovo_Gauss_29N	28489	Deprecated in EPSG, replaced by 2519 Pulkovo 1942 / Gauss-Kruger CM 171E.
PCS_Pulkovo_Gauss_30N	28490	Deprecated in EPSG, replaced by 2520 Pulkovo 1942 / Gauss-Kruger CM 177E.
PCS_Pulkovo_Gauss_31N	28491	Deprecated in EPSG, replaced by 2521 Pulkovo 1942 / Gauss-Kruger CM 177W.
PCS_Pulkovo_Gauss_32N	28492	Deprecated in EPSG, replaced by 2522 Pulkovo 1942 / Gauss-Kruger CM 171W.
PCS_SAD69_UTM_zone_18N	29118	Deprecated in EPSG, replaced by 29168 SAD69 / UTM zone 18N.
PCS_SAD69_UTM_zone_19N	29119	Deprecated in EPSG, replaced by 29169 SAD69 / UTM zone 19N.
PCS_SAD69_UTM_zone_20N	29120	Deprecated in EPSG, replaced by 29170 SAD69 / UTM zone 20N.
PCS_SAD69_UTM_zone_21N	29121	Deprecated in EPSG, replaced by 29171 SAD69 / UTM zone 21N.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
PCS_SAD69_UTM_zone_22N	29122	Deprecated in EPSG, replaced by 29172 SAD69 / UTM zone 22N.
PCS_SAD69_UTM_zone_17S	29177	Deprecated in EPSG, replaced by 29187 SAD69 / UTM zone 17S.
PCS_SAD69_UTM_zone_18S	29178	Deprecated in EPSG, replaced by 29188 SAD69 / UTM zone 18S.
PCS_SAD69_UTM_zone_19S	29179	Deprecated in EPSG, replaced by 29189 SAD69 / UTM zone 19S.
PCS_SAD69_UTM_zone_20S	29180	Deprecated in EPSG, replaced by 29190 SAD69 / UTM zone 20S.
PCS_SAD69_UTM_zone_21S	29181	Deprecated in EPSG, replaced by 29191 SAD69 / UTM zone 21S.
PCS_SAD69_UTM_zone_22S	29182	Deprecated in EPSG, replaced by 29192 SAD69 / UTM zone 22S.
PCS_SAD69_UTM_zone_23S	29183	Deprecated in EPSG, replaced by 29193 SAD69 / UTM zone 23S.
PCS_SAD69_UTM_zone_24S	29184	Deprecated in EPSG, replaced by 29194 SAD69 / UTM zone 24S.
PCS_SAD69_UTM_zone_25S	29185	Deprecated in EPSG, replaced by 29195 SAD69 / UTM zone 25S.
PCS_Sudan_UTM_zone_35N	29635	Deprecated in EPSG, replaced by 20135 Adindan / UTM zone 35N.
PCS_Sudan_UTM_zone_36N	29636	Deprecated in EPSG, replaced by 20136 Adindan / UTM zone 36N.
PCS_Tananarive_Laborde	29700	Deprecated in EPSG, replaced by 20701 Tananarive (Paris) / Laborde Grid and 29702 Tananarive (Paris) / Laborde Grid approximation.
PCS_Timbalai_1948_Borneo	*29800*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 29873 Timbalai 1948 / RSO Borneo (m).*
PCS_TM65_Irish_Nat_Grid	29900	Deprecated in EPSG, replaced by 29903 TM65 / Irish Grid.
PCS_Voirol_Unifie_N_Algerie	*30591*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 30791 Nord Sahara 1959 / Nord Algerie.*
PCS_Voirol_Unifie_S_Algerie	*30592*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 30792 Nord Sahara 1959 / Sud Algerie.*
PCS_Bern_1938_Swiss_New	*30600*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 21780 Bern 1898 (Bern) / LV03C.*
PCS_MGI_Austria_West	31291	Deprecated in EPSG, replaced by 31281 MGI (Ferro) / Austria West Zone.
PCS_MGI_Austria_Central	31292	Deprecated in EPSG, replaced by 31282 MGI (Ferro) / Austria Central Zone.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
PCS_MGI_Austria_East	31293	Deprecated in EPSG, replaced by 31283 MGI (Ferro) / Austria East Zone.
PCS_DHDN_Germany_zone_1	*31491*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 5520 DHDN / 3-degree Gauss-Kruger zone 1.*
PCS_DHDN_Germany_zone_2	*31492*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 31466 DHDN / 3-degree Gauss-Kruger zone 2.*
PCS_DHDN_Germany_zone_3	*31493*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 31467 DHDN / 3-degree Gauss-Kruger zone 3.*
PCS_DHDN_Germany_zone_4	*31494*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 31468 DHDN / 3-degree Gauss-Kruger zone 4.*
PCS_DHDN_Germany_zone_5	*31495*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 31469 DHDN / 3-degree Gauss-Kruger zone 5.*
PCS_NAD27_New_York_Long_Is	32018	Deprecated in EPSG, replaced by 4456 NAD27 / New York Long Island.
PCS_NAD27_Pennsylvania_S	32029	Deprecated in EPSG, replaced by 4455 NAD27 / Pennsylvania South.
PCS_NAD27_Tennessee	32036	Deprecated in EPSG, replaced by 2204 NAD27 / Tennessee.
PCS_NAD27_Puerto_Rico	*32059*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 3991 Puerto Rico State Plane CS of 1927.*
PCS_NAD27_St_Croix	*32060*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 3992 Puerto Rico / St. Croix.*
PCS_NAD27_BLM_14N_feet	32074	Deprecated in EPSG, replaced by 32064 NAD27 / BLM 14N (ftUS).
PCS_NAD27_BLM_15N_feet	32075	Deprecated in EPSG, replaced by 32065 NAD27 / BLM 15N (ftUS).
PCS_NAD27_BLM_16N_feet	32076	Deprecated in EPSG, replaced by 32066 NAD27 / BLM 16N (ftUS).
PCS_NAD27_BLM_17N_feet	32077	Deprecated in EPSG, replaced by 32067 NAD27 / BLM 17N (ftUS).

Table B.1 - Deprecated and deleted EPSG Projected CRS codes

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
GCS_Bern_1898	*4217*	*Deleted in EPSG, code reassigned to NAD83 / BLM 59N (ftUS) projected CRS, entity not replaced.*

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
GCS_Cote_d_Ivoire	4226	Deprecated in EPSG, replaced by 4142 Locodjo 1965 and 4143 Abidjan 1987.
GCS_Douala	4228	Deprecated in EPSG, replaced by 4192 Douala 1948.
GCS_Gandajika_1970	4233	Deprecated in EPSG, replaced by 4684 Gan 1970 and 4685 Gandajika.
GCS_Garoua	4234	Deprecated in EPSG, replaced by 4197 Garoua.
GCS_Guyane_Francaise	4235	Deprecated in EPSG, replaced by 4623 CSG67.
GCS_Manoca	4260	Deprecated in EPSG, replaced by 4193 Manoca 1962.
GCS_Mhast	4264	Deprecated in EPSG, replaced by 4704 Mhast (onshore) and 4705 Mhast (offshore).
GCS_NAD_Michigan	4268	Deprecated in EPSG, replaced by 4267 NAD27.
GCS_Qornoq	4287	Deprecated in EPSG, replaced by 4194 Qornoq 1927.
GCS_RT38	*4290*	*Deleted in EPSG, code re-used for a coordinate transformation, entity replaced by 4308 RT38.*
GCS_SAD69	4291	Deprecated in EPSG, replaced by 4618 SAD69.
GCS_Segora	4294	Deprecated in EPSG, replaced by 4613 Segara.
GCS_Sudan	4296	Deprecated in EPSG, replaced by 4201 Adindan.
GCS_Voirol_Unifie	*4305*	*Deleted in EPSG, code re-used for a map projection, entity replaced by 4307 Nord Sahara 1959.*
GCS_Voirol_Unifie_Paris	*4812*	*Deleted in EPSG, code reassigned to New Beijing / 3-degree Gauss-Kruger CM 132E projected CRS, entity replaced by 4819 Nord Sahara 1959 (Paris).*
GCS_NDG_Paris	4902	Deprecated in EPSG, replaced by 4901 ATF (Paris)
GCSE_Clarke1866Michigan	4009	Deprecated in EPSG, no replacement.
GCSE_Everest1830_1975Definition	*4017*	*Deleted in EPSG, code reassigned to MOLDREF99 geographic 3D CRS, entity replaced by 4045 Unknown datum based upon the Everest 1830 (1975 Definition) ellipsoid.*
GCSE_International1967	*4023*	*Deleted in EPSG, code reassigned to MOLDREF99 geographic 2D CRS, entity replaced by 4036 Unknown datum based upon the GRS 1967 ellipsoid.*
GCSE_NWL10D	*4026*	*Deleted in EPSG, code reassigned to MOLDREF99 / Moldova TM projected CRS, entity not replaced.*
GCSE_Sphere	4035	Deprecated in EPSG, replaced by 4047 Unspecified datum based upon the GRS 1980 Authalic Sphere.

Table B.2 - Deprecated and deleted EPSG Geodetic CRS codes

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
Linear_Foot_Modified_American	*9004*	*Deleted in EPSG, code not re-used as at 2018-05-29.*
Linear_Foot_Indian	*9006*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9080 Indian foot, 9081 Indian foot (1937), 9082 Indian foot (1962) and 9083 Indian foot (1975).*
Linear_Link	*9007*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9098 link.*
Linear_Link_Benoit	*9008*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9053 British link (Benoit 1895 A) and 9063 British link (Benoit 1895 B).*
Linear_Link_Sears	*9009*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9043 British link (Sears 1922).*
Linear_Chain_Benoit	*9010*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9052 British chain (Benoit 1895 A) and 9062 British chain (Benoit 1895 B).*
Linear_Chain_Sears	*9011*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9042 British chain (Sears 1922).*
Linear_Yard_Sears	*9012*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9040 British yard (Sears 1922).*
Linear_Yard_Indian	*9013*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9084 Indian yard, 9085 Indian yard (1937), 9086 Indian yard (1962) and 9087 Indian yard (1975).*
Linear_Mile_International_Nautical	*9015*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 9030 nautical mile.*

Table B.3 - Deprecated and deleted EPSG Unit of Measure codes

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
Datum_Bern_1898	*6217*	*Deleted in EPSG, code not re-used as at 2018-05-29, entity not replaced.*
Datum_Cote_d_Ivoire	6226	Deprecated in EPSG, replaced by 6142 Locodjo 1965 and 6143 Abidjan 1987.
Datum_Douala	6228	Deprecated in EPSG, replaced by 6192 Douala 1948.
Datum_Gandajika_1970	6233	Deprecated in EPSG, replaced by 6684 Gan 1970 and 6685 Gandajika.
Datum_Garoua	6234	Deprecated in EPSG, replaced by 6197 Garoua.

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
Datum_Guyane_Francaise	6235	Deprecated in EPSG, replaced by 6623 CSG67.
Datum_Manoca	6260	Deprecated in EPSG, replaced by 6193 Manoca 1962.
Datum_Mhast	6264	Deprecated in EPSG, replaced by 6704 Mhast (onshore) and 6705 Mhast (offshore).
Datum_NAD_Michigan	6268	Deprecated in EPSG, replaced by 6267 North American Datum 1927.
Datum_Qornoq	6287	Deprecated in EPSG, replaced by 6194 Qornoq 1927.
Datum_RT38	*6290*	*Deleted in EPSG, code not re-used as at 2018-05-29, entity replaced by 6308 Stockholm 1938.*
Datum_South_American_Datum_1969	6291	Deprecated in EPSG, replaced by 6618 SAD69.
Datum_Segora	6294	Deprecated in EPSG, replaced by 6613 Gunung Segara.
Datum_Sudan	6296	Deprecated in EPSG, replaced by 6201 Adindan.
Datum_Voirol_Unifie_1960	*6305*	*Deleted in EPSG, code not re-used as at 2018-05-29, entity replaced by 6307 Nord Sahara 1959.*
Datum_Nord_de_Guerre	6902	Deprecated in EPSG, replaced by 6901 ATF (Paris)

Table B.4 - Deprecated and deleted EPSG Geodetic Datum codes

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
Ellipse_Bessel_Namibia	7006	Deprecated in EPSG, replaced by 7046 Bessel Namibia (GLM).
Ellipse_Clarke_1866_Michigan	7009	Deprecated in EPSG, replaced by 7008 Clarke 1866.
Ellipse_Everest_1830_1975_Definition	*7017*	*Deleted in EPSG, code not re-used as at 2018-05-29, entity not replaced.*
Ellipse_International_1967	*7023*	*Deleted in EPSG, code not re-used as at 2018-05-29, entity replaced by 7036 GRS 1967 and 7050 GRS 1967 Modified.*
Ellipse_NWL_10D	*7026*	*Deleted in EPSG, code not re-used as at 2018-05-29, entity not replaced.*
Ellipse_Sphere	7035	Deprecated in EPSG, replaced by 7047 GRS 1980 Authalic Sphere.

Table B.5 - Deprecated and deleted EPSG Ellipsoid codes

Name given in GeoTIFF v1.0	EPSG CRS Code	Comment
Proj_California_CS27_VII	10407	Deprecated in EPSG, replaced by 10408 California CS27 zone VII.
Proj_Kentucky_CS83_North	11631	Deprecated in EPSG, replaced by 15303 Kentucky CS83 North zone (metres).
Proj_Michigan_CS27_North	12111	Deprecated in EPSG, replaced by 6965 Michigan CS27 North zone.
Proj_Michigan_CS27_Central	12112	Deprecated in EPSG, replaced by 6198 Michigan CS27 Central zone.
Proj_Michigan_CS27_South	12113	Deprecated in EPSG, replaced by 6199 Michigan CS27 South zone.
Proj_New_York_CS27_Long_Island	13104	Deprecated in EPSG, replaced by 4454 New York CS27 Long Island zone.
Proj_Pennsylvania_CS27_South	13702	Deprecated in EPSG, replaced by 4436 Pennsylvania CS27 South zone.
Proj_BLM_14N_feet	*15914*	*BLM zone 14N (US survey feet) (incorrect unit in GeoTIFF v1.0 specification).*
Proj_BLM_15N_feet	*15915*	*BLM zone 15N (US survey feet) (incorrect unit in GeoTIFF v1.0 specification).*
Proj_BLM_16N_feet	*15916*	*BLM zone 16N (US survey feet) (incorrect unit in GeoTIFF v1.0 specification).*
Proj_BLM_17N_feet	*15917*	*BLM zone 17N (US survey feet) (incorrect unit in GeoTIFF v1.0 specification).*
Proj_RSO_Borneo	*19912*	*Deleted in EPSG, code not re-used as at 2018-05-29, replaced by 19956 Rectified Skew Orthomorphic Borneo Grid (chains), 19957 Rectified Skew Orthomorphic Borneo Grid (feet) and 19958 Rectified Skew Orthomorphic Borneo Grid (metres).*

Table B.6 - Deprecated and deleted EPSG Map projection codes

Annex C: GeoTIFF Map Projection Method codes (Normative)

C.1. Map Projection methods

GeoTIFF v1.0 lists a number of map projection methods (which it calls 'coordinate transformations'). These are names, without any formulas or clear citation. As such they are an ambiguous identification of method. It is not possible to identify whether the formulas are ellipsoidal or spherical. If it is assumed that they are ellipsoidal formulas, for some the name does not adequately distinguish between method variations where different formulas lead to significantly different results in application. The codes for these listed methods are given in table C.1.

GeoTIFF v1.0 Map Projection Method Name	GeoTIFF v1.0 Map Projection Method Name Alias	GeoTIFF v1.0 Value for ProjMethodGeoKey
TransverseMercator	GaussBoaga; GaussKruger	1
TransvMercator_Modified_Alaska	AlaskaConformal	2
ObliqueMercator	ObliqueMercator_Hotine	3
ObliqueMercator_Laborde		4
ObliqueMercator_Rosenmund	SwissObliqueCylindrical	5
ObliqueMercator_Spherical		6
Mercator		7
LambertConfConic_2SP		8
LambertConfConic_Helmert		9
LambertAzimEqualArea		10
AlbersEqualArea		11
AzimuthalEquidistant		12
EquidistantConic		13
Stereographic		14
PolarStereographic		15
ObliqueStereographic		16
Equirectangular		17
CassiniSoldner	TransvEquidistCylindrical	18
Gnomonic		19
MillerCylindrical		20

GeoTIFF v1.0 Map Projection Method Name	GeoTIFF v1.0 Map Projection Method Name Alias	GeoTIFF v1.0 Value for ProjMethodGeoKey
Orthographic		21
Polyconic		22
Robinson		23
Sinusoidal		24
VanDerGrinten		25
NewZealandMapGrid		26
TransvMercator_SouthOriented		27

Table C.1 - Codes for GeoTIFF v1.0 map projection methods

NOTE ProjMethodGeoKey was called ProjCoordTransGeoKey in GeoTIFF v1.0

C.2. Map Projection parameters

GeoTIFF v1.0 lists a number of map projection parameters. These are names, without any definition or clear citation. As such they are ambiguous. It is also not clear which parameters should be used for which methods. The map projection parameter GeoKey codes are listed in table C.2.

GeoTIFF v1.0 Map Projection Parameter Name	GeoTIFF v1.0 Map Projection Parameter Alias	GeoTIFF v1.0 GeoKey	Parameter Value Unit
ProjStdParallel1GeoKey	ProjStdParallelGeoKey	3078	GeogAngularUnits
ProjStdParallel2GeoKey		3079	GeogAngularUnits
ProjNatOriginLongGeoKey	ProjOriginLongGeoKey	3080	GeogAngularUnits
ProjNatOriginLatGeoKey	ProjOriginLatGeoKey	3081	GeogAngularUnits
ProjFalseEastingGeoKey		3082	ProjLinearUnits
ProjFalseNorthingGeoKey		3083	ProjLinearUnits
ProjFalseOriginLongGeoKey		3084	GeogAngularUnits
ProjFalseOriginLatGeoKey		3085	GeogAngularUnits
ProjFalseOriginEastingGeoKey		3086	ProjLinearUnits
ProjFalseOriginNorthingGeoKey		3087	ProjLinearUnits

GeoTIFF v1.0 Map Projection Parameter Name	**GeoTIFF v1.0 Map Projection Parameter Alias**	**GeoTIFF v1.0 GeoKey**	**Parameter Value Unit**
ProjCenterLongGeoKey		3088	GeogAngularUnits
ProjCenterLatGeoKey		3089	GeogAngularUnits
ProjCenterEastingGeoKey		3090	ProjLinearUnits
ProjCenterNorthingGeoKey		3091	ProjLinearUnits
ProjScaleAtNatOriginGeoKey	ProjScaleAtOriginGeoKey	3092	ratio
ProjScaleAtCenterGeoKey		3093	ratio
ProjAzimuthAngleGeoKey		3094	GeogAzimuthUnits
ProjStraightVertPoleLongGeoKey		3095	GeogAngularUnits

Table C.2 - GeoKey codes for GeoTIFF v1.0 map projection parameters

Annex D: Recommendations for describing compound and geographic 3D CRSs (for grids containing height values)

(Informative)

D.1. Introduction

It is reminded here that GeoTIFF presently only handles Geographic 2D CRS or Projected 2D CRS, or geocentric (intrinsically 3D) as ModelTypeGeoKey. It is also reminded that a verticalCRS is a 1D gravity-related CRS and that ellipsoidal height cannot be 1D, only as the height part of a geographic 3D CRS. In all cases, when the GeoTIFF file contains digital elevation models (Terrain or Surface model, or gridded bathymetric data), the vertical reference is provided by the coumpound CRSs mechanism (Horizontal 2D CRS + VerticalGeoKey) documented below. However, it should be clear that the value of the VerticalGeoKey may be a verticalCRS (case of 1D gravity-related CRS) or describe a geographic 3D CRS (case of ellipsoïdal height). Therefore this key is named VerticalGeoKey (and can't be called VerticalCRSGeoKey).

D.2. Compound CRSs

A future version of GeoTIFF may expand the values of the GTModelTypeGeoKey to explicitly describe the Model CRS being compound. This document does not do that.

Recommendation: For a 2D+1D compound CRS comprised of projected CRS + vertical CRS, GTModelTypeGeoKey value should be 1 and both a ProjectedCRSGeoKey and a VerticalGeoKey should be given. The ProjectedCRSGeoKey should describe a 2D projected CRS.

Recommendation: For a 2D+1D compound CRS comprised of geographic 2D CRS + vertical CRS, GTModelTypeGeoKey value should be 2 and both a GeodeticCRSGeoKey and a VerticalGeoKey should be given. The GeodeticCRSGeoKey should describe a geographic 2D CRS.

D.3. Geographic 3D CRS (case of ellipsoïdal height)

A future version of GeoTIFF may expand the values of the GTModelTypeGeoKey to explicitly differentiate between geographic 2D and geographic 3D CRSs. This document does not do that. In this document there are two possible means of describing a geographic 3D CRS:

a) give GTModelTypeGeoKey value of 2 and identify a geographic 2D CRS through the GeodeticCRSGeoKey, giving the geographic 3D CRS description through the VerticalGeoKey. The geographic 2D CRS should be the horizontal component of the geographic 3D CRS. (This is the **recommended option**)

b) give GTModelTypeGeoKey value of 2 and use the VerticalGeoKey to describe an ellipsoid from which the ellipsoid height is measured using the GeoTIFF v1.0 codes in Table D.1 below.

NOTE This option is documented only for compatibility with some past Community practices. In this document a 1D ellipsoidal height is an invalid concept and use of these codes is deprecated.

None of these are long-term solutions. Until a major revision of this specification is available, recommended practice for producers/writers is to use (a). Readers should be prepared for any of these options.

Ellipsoid Name	GeoTIFF v1.0 Vertical Code
Airy_1830_ellipsoid	5001
Airy_Modified_1849_ellipsoid	5002
ANS_ellipsoid	5003
Bessel_1841_ellipsoid	5004
Bessel_Modified_ellipsoid	5005
Bessel_Namibia_ellipsoid	5006
Clarke_1858_ellipsoid	5007
Clarke_1866_ellipsoid	5008
Clarke_1880_Benoit_ellipsoid	5010
Clarke_1880_IGN_ellipsoid	5011
Clarke_1880_RGS_ellipsoid	5012
Clarke_1880_Arc_ellipsoid	5013
Clarke_1880_SGA_1922_ellipsoid	5014
Everest_1830_1937_Adjustment_ellipsoid	5015
Everest_1830_1967_Definition_ellipsoid	5016
Everest_1830_1975_Definition_ellipsoid	5017
Everest_1830_Modified_ellipsoid	5018
GRS_1980_ellipsoid	5019
Helmert_1906_ellipsoid	5020
Indonesian_National_Spheroid_ellipsoid	5021
International_1924_ellipsoid	5022
International_1967_ellipsoid	5023
Krassowsky_1940_ellipsoid	5024
NWL_9D_ellipsoid	5025
NWL_10D_ellipsoid	5026
Plessis_1817_ellipsoid	5027
Struve_1860_ellipsoid	5028
War_Office_ellipsoid	5029
WGS_84_ellipsoid	5030

Ellipsoid Name	**GeoTIFF v1.0 Vertical Code**
GEM_10C_ellipsoid	5031
OSU86F_ellipsoid	5032
OSU91A_ellipsoid	5033

Table D.1 - Deprecated codes from GeoTIFF v1.0 indicating Geographic 3D CRS ellipsoid heights (corresponding to option b)

Annex E: Summary of GeoKey IDs and names

(Informative)

Table 1. Summary of GeoKey IDs and names

Key ID	Type	GeoTIFF v1.0 key name	GeoTIFF v1.0 key alias	This document key name
GeoTIFF Configuration Keys				
1024	Short	GTModelTypeGeoKey		(as GeoTIFF v1.0)
1025	Short	GTRasterTypeGeoKey		(as GeoTIFF v1.0)
1026	Ascii	GTCitationGeoKey		(as GeoTIFF v1.0)
Geodetic CRS Parameter Keys				
2048	Short	GeographicTypeGeoKey		GeodeticCRSGeoKey
2049	Ascii	GeogCitationGeoKey		GeodeticCitationGeoKey
2050	Short	GeogGeodeticDatumGeoKey		GeodeticDatumGeoKey
2051	Short	GeogPrimeMeridianGeoKey		PrimeMeridianGeoKey
2052	Short	GeogLinearUnitsGeoKey		(as GeoTIFF v1.0)
2053	Double	GeogLinearUnitSizeGeoKey		(as GeoTIFF v1.0)
2054	Short	GeogAngularUnitsGeoKey		(as GeoTIFF v1.0)
2055	Double	GeogAngularUnitSizeGeoKey		(as GeoTIFF v1.0)
2056	Short	GeogEllipsoidGeoKey		EllipsoidGeoKey
2057	Double	GeogSemiMajorAxisGeoKey		EllipsoidSemiMajorAxisGeoKey
2058	Double	GeogSemiMinorAxisGeoKey		EllipsoidSemiMinorAxisGeoKey
2059	Double	GeogInvFlatteningGeoKey		EllipsoidInvFlatteningGeoKey
2061	Double	GeogPrimeMeridianLongGeoKey		PrimeMeridianLongitudeGeoKey
Projected CRS Parameter Keys				
2060	Short	GeogAzimuthUnitsGeoKey		(as GeoTIFF v1.0)

Key ID	Type	GeoTIFF v1.0 key name	GeoTIFF v1.0 key alias	This document key name
3072	Short	ProjectedCSTypeGeoKey		ProjectedCRSGeoKey
3073	Ascii	PCSCitationGeoKey		ProjectedCitationGeoKey
3074	Short	ProjectionGeoKey		(as GeoTIFF v1.0)
3075	Short	ProjCoordTransGeoKey		ProjMethodGeoKey
3076	Short	ProjLinearUnitsGeoKey		(as GeoTIFF v1.0)
3077	Double	ProjLinearUnitSizeGeoKey		(as GeoTIFF v1.0)
3078	Double	ProjStdParallel1GeoKey	ProjStdParallelGeoKey	(as GeoTIFF v1.0)
3079	Double	ProjStdParallel2GeoKey		(as GeoTIFF v1.0)
3080	Double	ProjNatOriginLongGeoKey	ProjOriginLongGeoKey	(as GeoTIFF v1.0)
3081	Double	ProjNatOriginLatGeoKey	ProjOriginLatGeoKey	(as GeoTIFF v1.0)
3082	Double	ProjFalseEastingGeoKey		(as GeoTIFF v1.0)
3083	Double	ProjFalseNorthingGeoKey		(as GeoTIFF v1.0)
3084	Double	ProjFalseOriginLongGeoKey		(as GeoTIFF v1.0)
3085	Double	ProjFalseOriginLatGeoKey		(as GeoTIFF v1.0)
3086	Double	ProjFalseOriginEastingGeoKey		(as GeoTIFF v1.0)
3087	Double	ProjFalseOriginNorthingGeoKey		(as GeoTIFF v1.0)
3088	Double	ProjCenterLongGeoKey		(as GeoTIFF v1.0)
3089	Double	ProjCenterLatGeoKey		(as GeoTIFF v1.0)
3090	Double	ProjCenterEastingGeoKey		(as GeoTIFF v1.0)
3091	Double	ProjCenterNorthingGeoKey		(as GeoTIFF v1.0)
3092	Double	ProjScaleAtNatOriginGeoKey	ProjScaleAtOriginGeoKey	(as GeoTIFF v1.0)

Key ID	Type	GeoTIFF v1.0 key name	GeoTIFF v1.0 key alias	This document key name
3093	Double	ProjScaleAtCenterGeoKey		(as GeoTIFF v1.0)
3094	Double	ProjAzimuthAngleGeoKey		(as GeoTIFF v1.0)
3095	Double	ProjStraightVertPoleLongGeoKey		(as GeoTIFF v1.0)
Vertical CRS Parameter Keys (4096-5119)				
4096	Short	VerticalCSTypeGeoKey		VerticalGeoKey
4097	Ascii	VerticalCitationGeoKey		(as GeoTIFF v1.0)
4098	Short	VerticalDatumGeoKey		(as GeoTIFF v1.0)
4099	Short	VerticalUnitsGeoKey		(as GeoTIFF v1.0)

Annex F: Annex: Examples

F.1. Introduction

This annex provides examples of how GeoTIFF may be implemented at the Tag and GeoKey level, following the general "Cookbook" approach presented in Cookbook: common examples, less common ones, including a Lunar example.

F.2. Common Examples

F.2.1. *UTM Projected Aerial Photo*

We have an aerial photo which has been orthorectified and resampled to a UTM grid, zone 60, using WGS 84 coordinate reference system; the coordinates of the upper-left corner of the image is are given in easting/northing, as 350807.4m, 5316081.3m. The scanned map pixel scale is 100 meters/pixels (the actual dpi scanning ratio is irrelevant).

```
ModelTiepointTag = (0, 0, 0, 350807.4, 5316081.3, 0.0)
ModelPixelScaleTag = (100.0, 100.0, 0.0)
GeoKeyDirectoryTag:
     GTModelTypeGeoKey = 1 (ModelTypeProjected 2D)
     GTRasterTypeGeoKey = 1 (RasterPixelIsArea)
     ProjectedCRSGeoKey = 32660 (Projected 2D CRS WGS 84 / UTM zone 60N)
     ProjectedCitationGeoKey = "UTM Zone 60 N with WGS84"
```

Notes:

1. We did not need to specify CitationGeoKey and indicate the base geographic CRS and the projection, since the 32660 code implies particular geographic CRS, projection and units already (WGS 84 Geographic 2D, UTM in zone 60N and meters). The citation was added just for documentation.
2. The "GeoKeyDirectoryTag" is expressed using the "GeoKey" structure defined above. At the TIFF level the tags look like this:

```
GeoKeyDirectoryTag=( 1,     0,  2,     4,
                  1024,     0,  1,     1,
                  1025,     0,  1,     1,
                  3072,     0,  1, 32660,
                  3073, 34737, 25,     0 )
GeoAsciiParamsTag(34737)=("UTM Zone 60 N with WGS84|")
```

For the rest of these examples we will only show the GeoKey-level dump, with the understanding that the actual TIFF-level tag representation can be determined from the documentation.

F.2.2. Standard State Plane

We have a USGS State Plane Map of Texas, Central Zone, using NAD83, correctly oriented. The map resolution is 1000 meters/pixel, at origin. There is a grid intersection line in the image at pixel location (50,100), and corresponds to the projected coordinate reference system easting/northing of (949465.0, 3070309.1).

```
ModelTiepointTag = ( 50, 100, 0, 949465.0, 3070309.1, 0)
ModelPixelScaleTag = (1000, 1000, 0)
GeoKeyDirectoryTag:
     GTModelTypeGeoKey = 1 (ModelTypeProjected 2D)
     GTRasterTypeGeoKey = 1 (RasterPixelIsArea)
     ProjectedCRSGeoKey = 32139 (Projected 2D CRS NAD83 / Texas Central)
```

Notice that in this case, since the PCS is a standard code, we do not need to define the GCS, datum, etc, since those are implied by the PCS code. Also, since this is NAD83, meters are used rather than US Survey feet (as in NAD 27).

F.2.3. Lambert Conformal Conic Aeronautical Chart

We have a 500 x 500 scanned aeronautical chart of Seattle, WA, using Lambert Conformal Conic projection, correctly oriented. The central meridian is at 120 degrees west. The map resolution is 1000 meters/pixel, at origin, and uses NAD27 datum. The standard parallels of the projection are at 41d20m N and 48d40m N. The latitude of the origin is at 45 degrees North, and occurs in the image at the raster coordinates (80,100). The origin is given a false easting and northing of 200000m, 1500000m.

```
ModelTiepointTag = ( 80, 100, 0, 200000, 1500000, 0)
ModelPixelScaleTag = (1000, 1000, 0)
GeoKeyDirectoryTag:
     GTModelTypeGeoKey = 1 (ModelTypeProjected 2D)
     GTRasterTypeGeoKey = 1 (RasterPixelIsArea)
     GeographicTypeGeoKey = 4267 (GCS_NAD27)
     ProjectedCRSGeoKey = 32767 (user-defined)
     ProjectionGeoKey = 32767 (user-defined)
     ProjLinearUnitsGeoKey = 9001 (Linear_Meter)
     ProjCoordTransGeoKey = 8 (CT_LambertConfConic_2SP)
          ProjStdParallel1GeoKey = 41.333
          ProjStdParallel2GeoKey = 48.666
          ProjCenterLongGeoKey =-120.0
          ProjNatOriginLatGeoKey = 45.0
          ProjFalseEastingGeoKey, = 200000.0
          ProjFalseNorthingGeoKey, = 1500000.0
```

Notice that the Tiepoint takes the false easting and northing into account when tying the raster point (50,100) to the projection origin.

F.2.4. DMA ADRG Raster Graphic Map

The U.S. Defense Mapping Agency produces ARC digitized raster graphics datasets by scanning maps and geometrically resampling them into an equirectangular projection, so that they may be directly indexed with WGS84 geographic coordinates. The scale for one map is 0.2 degrees per pixel horizontally, 0.1 degrees per pixel vertically. If stored in a GeoTIFF file it contains the following information:

```
ModelTiepointTag=(0.0, 0.0, 0.0, -120.0, 32.0, 0.0)
ModelPixelScale = (0.2, 0.1, 0.0)
GeoKeyDirectoryTag:
     GTModelTypeGeoKey = 2 (ModelTypeGeographic 2D)
     GTRasterTypeGeoKey = 1 (RasterPixelIsArea)
     GeographicTypeGeoKey = 4326 (Geographic 2D WGS 84)
```

F.3. Less Common Examples

F.3.1. Unrectified Aerial photo, known tiepoints, in degrees.

We have an aerial photo, and know only the WGS 84 GPS location of several points in the scene: the upper left corner is 120 degrees West, 32 degrees North, the lower-left corner is at 120 degrees West, 30 degrees 20 minutes North, and the lower-right hand corner of the image is at 116 degrees 40 minutes West, 30 degrees 20 minutes North. The photo is not geometrically corrected, however, and the complete projection is therefore not known.

```
ModelTiepointTag=( 0.0,    0.0, 0.0,       -120.0,     32.0, 0.0,
                   0.0, 1000.0, 0.0,       -120.0, 30.33333, 0.0,
                1000.0, 1000.0, 0.0, -116.6666667, 30.33333, 0.0)
    GeoKeyDirectoryTag:
         GTModelTypeGeoKey = 1 (ModelTypeGeographic 2D)
         GTRasterTypeGeoKey = 1 (RasterPixelIsArea)
         GeographicTypeGeoKey = 4326 (Geographic 2D WGS 84)
```

Remark: Since we have not specified the ModelPixelScaleTag, clients reading this GeoTIFF file are not permitted to infer that there is a simple linear relationship between the raster data and the geographic model coordinate space. The only points that are know to be exact are the ones specified in the tiepoint tag.

F.3.2. Rotated Scanned Map

We have a scanned standard British National Grid, covering the 100km grid zone NZ. Consulting documentation for BNG we find that the southwest corner of the NZ zone has an easting,northing of 400000m, 500000m, relative to the BNG standard false origin. This scanned map has a resolution of 100 meter pixels, and was rotated 90 degrees to fit onto the scanner, so that the southwest corner is now the northwest corner. In this case we must use the ModelTransformation tag rather than the tiepoint/scale pair to map the raster data into model space:

```
ModelTransformationTag = ( 0, 100.0, 0, 400000.0,
                       100.0,     0, 0, 500000.0,
                           0,     0, 0,        0,
                           0,     0, 0,        1)
  GeoKeyDirectoryTag:
       GTModelTypeGeoKey = 1 ( ModelTypeProjected 2D)
       GTRasterTypeGeoKey = 1 (RasterPixelIsArea)
       ProjectedCRSGeoKey = 27700 (ProjectedCRS OSGB 1936 / British National Grid)
       ProjectedCitationGeoKey = "British National Grid, Zone NZ"
```

Remark: the matrix has 100.0 in the off-diagonals due to the 90 degree rotation; increasing I points north, and increasing J points east.

F.3.3. Digital Elevation Model

The DMA stores digital elevation models using an equirectangular projection, so that it may be indexed with WGS 84 geographic coordinates. Since elevation postings are point-values, the pixels should not be considered as filling areas, but as point-values at grid vertices. To accommodate the base elevation of the Angeles Crest forest, the pixel value of 0 corresponds to an elevation of 1000 meters relative to WGS 84 reference ellipsoid. The upper left corner is at 120 degrees West, 32 degrees North, and has a pixel scale of 0.2 degrees/pixel longitude, 0.1 degrees/pixel latitude.

```
ModelTiepointTag=(0.0, 0.0, 0.0, -120.0, 32.0, 1000.0)
ModelPixelScale = (0.2, 0.1, 1.0)
GeoKeyDirectoryTag:
     GTModelTypeGeoKey = 2 (ModelTypeGeographic 2D)
     GTRasterTypeGeoKey = 2 (RasterPixelIsPoint)
     GeographicTypeGeoKey = 4326 (Geographic 2D WGS 84)
     VerticalGeoKey = 4979 (Geographic 3D WGS 84, used here to document use of
ellipsoidal height)
     VerticalCitationGeoKey = "Geographic 3D WGS 84, Ellipsoidal height"
     VerticalUnitsGeoKey = 9001 (Linear_Meter)
```

Remarks:

1. Note the "RasterPixelIsPoint" raster space, indicating that the DEM posting of the first pixel is at the raster point (0,0,0), and therefore corresponds to 120W,32N exactly.
2. The third value of the "PixelScale" is 1.0 to indicate that a single pixel-value unit corresponds to 1 meter, and the last tiepoint value indicates that base value zero indicates 1000m above the reference surface.

F.3.4. Spherical Moon Example

Introduction

The GeoTIFF Standard can be used for images from extraterrestrial bodies as well as the Earth. This Annex illustrates a simple example for a spherical Moon. This example also shows how more custom Earth-base examples could also be defined, highlighting the flexibility of the GeoTiff standard.

Example

Note this example (using listgeo), is showing the header values as mapped strings instead of the original short Integer. e.g. GTModelTypeGeoKey = ModelTypeProjected (which is really mapped from value 1) see: https://github.com/ufz/geotiff/blob/473ab941f80592ada6a226ec666d7e6e8f79c21b/geonames.h#L67

$ listgeo Lunar_LRO_LOLA_Global_LDEM_118m_Mar2014.tif

```
Geotiff_Information:
   Version: 1
   Key_Revision: 1.0
   Tagged_Information:
      ModelTiepointTag (2,3):
         0                 0                 0
         -5458203.076608   2729101.538304    0
      ModelPixelScaleTag (1,3):
         118.4505876       118.4505876       0
      End_Of_Tags.
   Keyed_Information:
      GTModelTypeGeoKey (Short,1): ModelTypeProjected
      GTRasterTypeGeoKey (Short,1): RasterPixelIsArea
      GTCitationGeoKey (Ascii,29): "SimpleCylindrical Moon"
      GeographicTypeGeoKey (Short,1): User-Defined
      GeogCitationGeoKey (Ascii,124): "GCS Name = Moon 2000|Datum =
D_Moon_2000|Ellipsoid =
          Moon_2000_IAU_IAG|Primem = Reference_Meridian|AUnits = Decimal_Degree|"
      GeogGeodeticDatumGeoKey (Short,1): User-Defined
      GeogAngularUnitSizeGeoKey (Double,1): 0.0174532925199433
      GeogEllipsoidGeoKey (Short,1): User-Defined
      GeogSemiMajorAxisGeoKey (Double,1): 1737400
      GeogSemiMinorAxisGeoKey (Double,1): 1737400
      GeogPrimeMeridianLongGeoKey (Double,1): 0
      ProjectedCSTypeGeoKey (Short,1): User-Defined
      ProjectionGeoKey (Short,1): User-Defined
      ProjCoordTransGeoKey (Short,1): CT_Equirectangular
      ProjLinearUnitsGeoKey (Short,1): Linear_Meter
      ProjStdParallel1GeoKey (Double,1): 0
      ProjFalseEastingGeoKey (Double,1): 0
      ProjFalseNorthingGeoKey (Double,1): 0
      ProjCenterLongGeoKey (Double,1): 0
      ProjCenterLatGeoKey (Double,1): 0
      End_Of_Keys.
   End_Of_Geotiff.
```

Annex G: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version
2018-06-07	0.2	C. Heazel	all	AsciiDoc version
2019-01-07	0.3	E. Devys, R. Lott, E. Rouault	update of all geographic requirements & content	revised AsciiDoc version

Annex H: Bibliography

GeoTIFF profile for Georeferenced Imagery, DGIWG 108, 2017-12-08, Retrieved from https://portal.dgiwg.org/files/?artifact_id=68102&format=pdf

Ritter, N., & Ruth, M. (1997). The GeoTiff data interchange standard for raster geographic images. *International Journal of Remote Sensing, 18*(7), 1637–1647. doi:10.1080/014311697218340Ritter

Wiggins, R. H., Davidson, H. C., Harnsberger, H. R., Lauman, J. R., & Goede, P. a. (2001). Image file formats: past, present, and future. *Radiographics : a review publication of the Radiological Society of North America, Inc, 21*(3), 789–98. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/11353125