**Open Geospatial Consortium**

Submission Date: <yyyy-dd-mm>

Approval Date:   <yyyy-dd-mm>

Publication Date:   <yyyy-dd-mm>

External identifier of this OGC® document: <<http://www.opengis.net/def/doc-type/standard/1.0>>

Internal reference number of this OGC® document:    YY-nnnrx

Version: n.n.n

Category: OGC® <Implementation / Abstract Specification / Best Practice>

Editor:   Ted Habermann

OGC GeoTIFF Standard

**Copyright notice**

Copyright © 2015 Open Geospatial Consortium  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

**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.

Document type:    OGC® Standard

Document subtype:    if applicable

Document stage:    Draft

Document language:  English

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.

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER’S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR’s sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize 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.

Contents

1. Scope 9

2. Conformance 9

3. References 10

4. Terms and Definitions 10

5. Conventions 15

6. Clauses not Containing Normative Material 15

6.1 Clauses not containing normative material sub-clause 1 16

6.1.1 Clauses not containing normative material sub-clause 2 16

7. Requirements 16

7.1 Underlying TIFF Requirements 16

7.1.1 Requirements Class TIFF 16

7.2 GeoTIFF Requirements 16

7.2.1 Requirements Class GeoKeyDirectoryTag 16

7.2.2 Requirements Class GeoKeyCode 17

7.2.3 Requirements Class GeoAsciiParamsTag 18

7.2.4 Requirements Class GeodeticDatumGeoKey 18

7.2.5 Requirements Class GeoDoubleParamsTag 19

7.2.6 Requirements Class GeogAngularUnitsGeoKey 19

7.2.7 Requirements Class GeogAngularUnitSizeGeoKey 20

7.2.8 Requirements Class GeogAzimuthUnitsGeoKey 20

7.2.9 Requirements Class GeogCitationGeoKey 21

7.2.10 Requirements Class GeogEllipsoidGeoKey 21

7.2.11 Requirements Class GeogGeodeticDatumGeoKey 21

7.2.12 Requirements Class GeogInvFlatteningGeoKey 22

7.2.13 Requirements Class GeogLinearUnitsGeoKey 22

7.2.14 Requirements Class GeogLinearUnitSizeGeoKey 23

7.2.15 Requirements Class GeogPrimeMeridianGeoKey 23

7.2.16 Requirements Class GeogPrimeMeridianLongGeoKey 24

7.2.17 Requirements Class GeographicTypeGeoKey 24

7.2.18 Requirements Class GeogSemiMajorAxisGeoKey 25

7.2.19 Requirements Class GeogSemiMinorAxisGeoKey 25

7.2.20 Requirements Class GeoKeyRange 26

7.2.21 Requirements Class GTCitationGeoKey 26

7.2.22 Requirements Class GTModelTypeGeoKey 27

7.2.23 Requirements Class GTRasterTypeGeoKey 28

7.2.24 Requirements Class IntergraphMatrixTag 28

7.2.25 Requirements Class KeyEntrySet 29

7.2.26 Requirements Class ModelPixelScaleTag 30

7.2.27 Requirements Class ModelTag 30

7.2.28 Requirements Class ModelTiePointTag 31

7.2.29 Requirements Class ModelTransformationTag 31

7.2.30 Requirements Class PCSCitationGeoKey 32

7.2.31 Requirements Class ProjectedCSTypeGeoKey 32

7.2.32 Requirements Class ProjectionGeoKey 32

7.2.33 Requirements Class ProjCoordTransGeoKey 33

7.2.34 Requirements Class ProjLinearUnitsGeoKey 33

7.2.35 Requirements Class ProjLinearUnitSizeGeoKey 33

7.2.36 Requirements Class ProjStdParallel1GeoKey 34

7.2.37 Requirements Class ProjStdParallel2GeoKey 34

7.2.38 Requirements Class ProjNatOriginLongGeoKe 34

7.2.39 Requirements Class ProjNatOriginLatGeoKey 35

7.2.40 Requirements Class ProjFalseEastingGeoKey 35

7.2.41 Requirements Class ProjFalseNorthingGeoKey 35

7.2.42 Requirements Class ProjFalseOriginLongGeoKey 36

7.2.43 Requirements Class ProjFalseOriginLatGeoKey 36

7.2.44 Requirements Class ProjFalseOriginEastingGeoKey 37

7.2.45 Requirements Class ProjFalseOriginNorthingGeoKey 37

7.2.46 Requirements Class ProjCenterLongGeoKey 38

7.2.47 Requirements Class ProjCenterLatGeoKey 38

7.2.48 Requirements Class ProjCenterEastingGeoKey 39

7.2.49 Requirements Class ProjScaleAtNatOriginGeoKey 39

7.2.50 Requirements Class ProjScaleAtCenterGeoKey 40

7.2.51 Requirements Class ProjAzimuthAngleGeoKey 40

7.2.52 Requirements Class ProjStraightVertPoleLongGeoKey 40

7.2.53 Requirements Class VerticalCitationGeoKey 41

7.2.54 Requirements Class VerticalCSTypeGeoKey 41

7.2.55 Requirements Class VerticalDatumGeoKey 42

7.2.56 Requirements Class VerticalUnitsGeoKey 42

8. Media Types for any data encoding(s) 43

D.5.1 Device Space and GeoTIFF 53

D.5.2 Raster Coordinate Systems 53

D.5.2.1 Raster Data 53

D.5.2.2 Raster Space 53

D.5.3 Model Coordinate Systems 54

D.5.3.1 Geographic Coordinate Systems 55

D.5.3.2 Geocentric Coordinate Systems 57

D.5.3.3 Projected Coordinate Systems 57

D.5.3.4 Vertical Coordinate Systems 59

D.5.4 Reference Parameters 59

D.6.1 GeoTIFF Tags for Coordinate Transformations 61

D.6.2 Coordinate Transformation Data Flow 65

D.6.3 Cookbook for Defining Transformations 66

D.7.1 General Approach 67

D.7.2 Cookbook for Geocoding Data 67

D.8.1 Common Examples 69

D.8.1.1 UTM Projected Aerial Photo 69

D.8.1.2 Standard State Plane 70

D.8.1.3 Lambert Conformal Conic Aeronautical Chart 70

D.8.1.4 DMA ADRG Raster Graphic Map 71

D.8.2 Less Common Examples 71

D.8.2.1 Unrectified Aerial photo, known tiepoints, in degrees. 71

D.8.2.2 Rotated Scanned Map 71

D.8.2.3 Digital Elevation Model 72

F.2.1 Requirements Class EPSG-GeodeticDatumGeoKey 76

F.2.2 Requirements Class EPSG-GeogEllipsoidGeoKey 76

F.2.3 Requirements Class EPSG-GeogLinearUnitsGeoKey 77

F.2.4 Requirements Class EPSG-GeogPrimeMeridianGeoKey 77

F.2.5 Requirements Class EPSG-GeographicTypeGeoKey 78

F.2.6 Requirements Class EPSG-ProjectedCSTypeGeoKey 78

F.2.7 Requirements Class EPSG-VerticalCSTypeGeoKey 79

F.2.8 Requirements Class EPSG-VerticalUnitsGeoKey 79

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.

Keywords

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

ogcdoc, OGC document, GeoTIFF, remote-sensing, image, data format

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 to share geographic image data throughout the geospatial and earth science communities. 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.*

Submitting organizations

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

The HDF Group

Submitters

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

|  |  |
| --- | --- |
| Name | Affiliation |
| Ted Habermann | The HDF Group |
|  |  |
|  |  |

# Scope

This OGC**®** 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 map projection, and for describing those projections. This OGC**®** Standard defines the Geographic Tagged Image File Format (GeoTIFF) file format and the requirements to which every GeoTIFF file must adhere. It focuses on updating the current GeoTIFF community specification and aligning it with current OGC standardization practice.

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.

# Conformance

This standard defines XXXX.

Requirements for N standardization target types are considered:

* AAAA
* BBBB

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[[1]](#footnote-1).

In order to conform to this OGC™interface standard, a software implementation shall choose to implement:

1. Any one of the conformance levels specified in Annex B (normative).
2. Any one of the Distributed Computing Platform profiles specified in Annexes TBD through TBD (normative).

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

# 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.

There are no normative references.

# Terms and Definitions

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.

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

1. absolute accuracy

Closeness of coordinate value to the true or accepted valuein a specified reference system (in this profile, the reference system is the World Geodetic System 1984 (WGS84))

1. ASCII

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

1. band

A well-defined range of wavelengths, frequencies or energies of optical, electric or acoustic radiation. 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 ith sample values from the pixel vectors.

1. cell

A rectangular area in Raster space, in which a single pixel value is filled.

1. code

representation of a label according to a specified scheme

1. coordinate

One of a sequence of numbers designating the position of a point in N-dimensional space

1. coordinate reference system

Coordinate system that is related to an object (of the real world) by a datum.

1. coordinate system

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

1. coverage

Feature that acts as a function to return values from its range for any direct position within its spatial, temporal, or spatiotemporal domain. Examples include a digital image, raster map, and digital elevation matrix.

1. coverage geometry

Configuration of the domain of a coverage described in terms of coordinates.

1. data compression

Reducing the amount of storage space required to store a given amount of data, or reducing the length of message required to transfer a given amount of reduction in the number of bits used to represent source image data [ISO 10918-1] (JPEG Part 1) information. (data / image) compression

1. dataset

Identifiable collection of data.

1. datum

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

1. device space

A coordinate space referencing scanner, printers and display devices.

1. direct position

Position described by a single set of coordinates within a coordinate reference system.

1. domain

Well-defined set. Note, Domains are used to define the domain set and range set of operators and functions.

1. double

8-byte IEEE double precision floating point.

1. ellipsoid

A surface formed by the rotation of an ellipse about a main axis

1. EPSG

European Petroleum Survey Group.

1. evaluation

Determination of the values of a coverage at a direct position within the domain of the coverage.

1. flattening

A 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

1. geocoding

A translation of one form of location into another

1. geographic coordinate system

A Geographic CS consists of a well-defined ellipsoidal datum, a Prime Meridian, and an angular unit, allowing the assignment of a Latitude-Longitude (and optionally, geodetic height) vector to a location on earth.

1. geokey

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

1. georectified grid

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

1. georeferencing

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

1. geoTIFF

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

1. grid

A 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

1. grid

gridded data Network composed of two or more sets of curves in which the members of each set intersect the members of the other sets in a algorithmic way.

1. IEEE

Institute of Electrical and Electronics Engineers, Inc.

1. 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).

1. imagery

Representation of phenomena as images produced electronically and/or optical techniques.

1. meridian

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

1. metadata

Data about data.

1. model space

A flat geometrical space used to model a portion of the earth.

1. mosaic

For purposes of this profile, a mosaic image is an image composed of two or more separately collected (sensed) images. Additional XML metadata may be used to identify the cut-lines (boundaries and parameters for the images used to compose the mosaic.

1. null value

Value having no value or existence.

1. orthorectified grid

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

1. parallel

Lines of constant latitude, parallel to the equator.

1. pixel

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

1. 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 The intensity of each pixel is variable; in color systems, each pixel has typically three or four dimensions of variability such as red, green and blue, or cyan, magenta, yellow and black.

1. POSC

Petrotechnical Open Software Corporation.

1. prime meridian

A meridian from which the longitudes of other meridians are quantified

1. projected coordinate system

A coordinate reference system derived from a two-dimensional geodetic coordinate reference system by applying a map projection

1. projection

projected coordinate reference system: coordinate reference system derived from a two-dimensional geodetic coordinate reference system by applying a map projection

1. qualification layer

A qualification layer is a coverage consisting of graphics information associated to geospatial data together with associated metadata (these metadata mostly identify the meaning of colour codes used in graphics).

1. range

Set of feature attribute values associated by a function with the elements of the domain of a coverage.

1. raster space

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

1. rational

In TIFF format, a RATIONAL value is a fractional value represented by the ratio of two unsigned 4-byte integers.

1. rectified grid

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

1. referenceable 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

1. relative accuracy / relative positional accuracy

Evaluation of the random errors in determining the position of one point or feature with respect to another / closeness of coordinate difference value to the true or accepted value in a specified reference system

1. SDTS

The USGS Spatial Data Transmission Standard.

1. tag

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

1. tessellation / tiling

Partitioning of a space into a set of conterminous subspaces having the same dimension as the space being partitioned

1. TIFF

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

1. transparency mask

A Transparency Mask defines visible pixels of another image in the same TIFF file (that may be organised as an irregularly shaped region of visible pixels). The 1-bits define the visible pixels; the 0-bits define transparent pixels. (fdefinition based on TIFF specification)

1. USGS

US Geological Survey

# Conventions

None

# Clauses not Containing Normative Material

Paragraph

## Clauses not containing normative material sub-clause 1

Paragraph

### Clauses not containing normative material sub-clause 2

# Requirements

## Underlying TIFF Requirements

### Requirements Class TIFF

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/TIFF | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/TIFF.double *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.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/TIFF.fileStructure *A GeoTIFF file is a TIFF 6.0 file, and inherits the file structure as described in the corresponding portion of the TIFF spec.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/TIFF.noPrivateInformation *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* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/TIFF.tagOrder *GeoKey entries SHALL be written within the CoordSystemInfoTag in tag-ID sorted order.* |

## GeoTIFF Requirements

### Requirements Class GeoKeyDirectoryTag

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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* |
| 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.keyRevision *The second unsigned short integer in the GeoKeyDirectoryTag SHALL hold the KeyRevision.* |
| 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.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* |

### Requirements Class GeoKeyCode

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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* |

### Requirements Class GeoAsciiParamsTag

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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.type *The GeoAsciiParamsTag SHALL have type = ASCII* |
| 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* |

### Requirements Class GeodeticDatumGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.reserved *GeodeticDatumGeoKey values in the range 1001-5999 and 7000-32766 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.WGSDatum *GeodeticDatumGeoKey values in the range 6322-6327 SHALL be WGS Datum* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.ArchaicDatum *GeodeticDatumGeoKey values in the range 6900-6999 SHALL be Archaic Datum* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeodeticDatumGeoKey.private *GeodeticDatumGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class GeoDoubleParamsTag

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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.type *The GeoDoubleParamsTag SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeoDoubleParamsTag.count *The GeoDoubleParamsTag MAY hold any number of key parameters with type = double. (May not be necessary, the same as keyentry.count)* |

### Requirements Class GeogAngularUnitsGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogAngularUnitsGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogAngularUnitsGeoKey.ID *The GeogAngularUnitsGeoKey SHALL have ID = 2054* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogAngularUnitsGeoKey.type *The GeogAngularUnitsGeoKey SHALL have type = SHORT* |

### Requirements Class GeogAngularUnitSizeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogAngularUnitSizeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogAngularUnitSizeGeoKey.ID *The GeogAngularUnitSizeGeoKey SHALL have ID = 2055* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogAngularUnitSizeGeoKey.type *The GeogAngularUnitSizeGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogAngularUnitSizeGeoKey.units *The units of the GeogAngularUnitSizeGeoKey SHALL be radians* |

### Requirements Class GeogAzimuthUnitsGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogAzimuthUnitsGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogAzimuthUnitsGeoKey.ID *The GeogAzimuthUnitsGeoKey SHALL have ID = 2060* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogAzimuthUnitsGeoKey.type *The GeogAzimuthUnitsGeoKey SHALL have type = SHORT* |

### Requirements Class GeogCitationGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogCitationGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogCitationGeoKey.ID *The GeogCitationGeoKey SHALL have ID = 2049* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogCitationGeoKey.type *The GeogCitationGeoKey SHALL have type = ASCII* |

### Requirements Class GeogEllipsoidGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogEllipsoidGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogEllipsoidGeoKey.ID *The GeogEllipsoidGeoKey SHALL have ID = 2056* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogEllipsoidGeoKey.type *The GeogEllipsoidGeoKey SHALL have type = SHORT* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogEllipsoidGeoKey.reserved *GeogEllipsoidGeoKey values in the range 1001-6999 and 8000-32766 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogEllipsoidGeoKey.private *GeogEllipsoidGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class GeogGeodeticDatumGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogGeodeticDatumGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogGeodeticDatumGeoKey.ID *The GeogCitationGeoKey SHALL have ID = 2050* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogGeodeticDatumGeoKey.type *The GeogCitationGeoKey SHALL have type = SHORT* |

### Requirements Class GeogInvFlatteningGeoKey

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

### Requirements Class GeogLinearUnitsGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitsGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitsGeoKey.ID *The GeogLinearUnitsGeoKey SHALL have ID = 2052* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitsGeoKey.type *The GeogLinearUnitsGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitsGeoKey.obsolete *GeogLinearUnitsGeoKey values in the range 1-2000 SHALL be obsolete GeoTIFF Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitsGeoKey.reserved *GeogLinearUnitsGeoKey values in the range 2001-8999 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitsGeoKey.private *GeogLinearUnitsGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class GeogLinearUnitSizeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitSizeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitSizeGeoKey.ID *The GeogLinearUnitSizeGeoKey SHALL have ID = 2053* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitSizeGeoKey.type *The GeogLinearUnitSizeGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogLinearUnitSizeGeoKey.units *The units of the GeogLinearUnitSizeGeoKey SHALL be meters* |

### Requirements Class GeogPrimeMeridianGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianGeoKey.ID *The GeogPrimeMeridianGeoKey SHALL have ID = 2051* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianGeoKey.type *The GeogPrimeMeridianGeoKey SHALL have type = SHORT* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianGeoKey.reserved *GeogPrimeMeridianGeoKey values in the range 101-7999 and 9000-32766 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianGeoKey.private *GeogPrimeMeridianGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class GeogPrimeMeridianLongGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianLongGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianLongGeoKey.ID *The GeogPrimeMeridianLongGeoKey SHALL have ID = 2061* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianLongGeoKey.type *The GeogPrimeMeridianLongGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogPrimeMeridianLongGeoKey.units *The GeogPrimeMeridianLongGeoKey SHALL have units = GeogAngularUnits* |

### Requirements Class GeographicTypeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeographicTypeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeographicTypeGeoKey.ID *The GeographicTypeGeoKey SHALL have ID = 2048* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeographicTypeGeoKey.type *The GeographicTypeGeoKey SHALL have type = SHORT* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeographicTypeGeoKey.reserved *GeographicTypeGeoKey values in the range 1001-3999 and 5000-32766 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeographicTypeGeoKey.private *GeographicTypeGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class GeogSemiMajorAxisGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogSemiMajorAxisGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogSemiMajorAxisGeoKey.ID *The GeogSemiMajorAxisGeoKey SHALL have ID = 2057* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogSemiMajorAxisGeoKey.type *The GeogSemiMajorAxisGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogSemiMajorAxisGeoKey.units *The units of the GeogSemiMajorAxisGeoKey SHALL be Geocentric CS linear Units* |

### Requirements Class GeogSemiMinorAxisGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeogSemiMinorAxisGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogSemiMinorAxisGeoKey.ID *The GeogSemiMinorAxisGeoKey SHALL have ID = 2058* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogSemiMinorAxisGeoKey.type *The GeogSemiMinorAxisGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeogSemiMinorAxisGeoKey.units *The units of the GeogSemiMinorAxisGeoKey HALL be Geocentric CS linear Units* |

### Requirements Class GeoKeyRange

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyRange | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyRange.reserved *GeoKey ID's in the range 0-1023 and 5120-32767 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyRange.configuration *GeoKey ID's in the range 1024-2047 SHALL be GeoTIFF Configuration Keys* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyRange.geographic *GeoKey ID's in the range 2048-3071 SHALL be Geographic/Geocentric Coordinate System Parameter Keys* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyRange.projected *GeoKey ID's in the range 3072-4095 SHALL be Projected Coordinate System Parameter Keys* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyRange.reserved *GeoKey ID's in the range 4096-5119 SHALL be Vertical Coordinate System Parameter Keys* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GeoKeyRange.reserved *GeoKey ID's in the range 32768-65535 SHALL be for private use* |

### Requirements Class GTCitationGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GTCitationGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTCitationGeoKey.ID *The GTCitationGeoKey SHALL have ID = 1026* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTCitationGeoKey.type *The GTCitationGeoKey SHALL have type = ASCII* |

### Requirements Class GTModelTypeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/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.projected *A value of 1 for the GTModelTypeGeoKey SHALL indicate a projected coordinate system* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.geographic *A value of 2 for the GTModelTypeGeoKey SHALL indicate a geographic latitude-longitude coordinate system* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.geocentric *A value of 3 for the GTModelTypeGeoKey SHALL indicate a geocentric(X,Y,Z) coordinate system* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.reserved *GTModelTypeGeoKey values in the range 1-32766 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTModelTypeGeoKey.private *GTModelTypeGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class GTRasterTypeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.ID *The GTModelTypeGeoKey SHALL have ID = 1025* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.type *The GTModelTypeGeoKey SHALL have type = SHORT* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.rasterPixelIsArea *A value of 1 for the GTRasterTypeGeoKey SHALL indicate that this raster pixel is an area (for DGIWG profile, this is used by imagery products).* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.rasterPixelIsPoint *A value of 2 for the GTRasterTypeGeoKey SHALL indicate that this raster pixel is a point (for DGIWG profile, this is used for discrete coverage data including elevation data ).* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.private *GTRasterTypeGeoKey values in the range 32768-65535 SHALL be private* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/GTRasterTypeGeoKey.reserved *GTRasterTypeGeoKey values in the range 1-32766 SHALL be reserved* |

### Requirements Class IntergraphMatrixTag

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/IntergraphMatrixTag | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/IntergraphMatrixTag.ID *The IntergraphMatrixTag SHALL have ID = 33920* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/IntergraphMatrixTag.type *The IntergraphMatrixTag SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/IntergraphMatrixTag.count *The IntergraphMatrixTag SHALL hold sixteen or seventeen values* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/IntergraphMatrixTag.deprecated *The IntergraphMatrixTag SHALL not be used* |

### Requirements Class KeyEntrySet

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/KeyEntrySet | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/KeyEntrySet.size *Each Key Entry Set SHALL hold four unsigned short integers* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/KeyEntrySet.keyID *The first unsigned short integer in each Key Entry Set SHALL hold the KeyID.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/KeyEntrySet.TIFFTagLocation *The second unsigned short integer in each Key Entry Set SHALL hold the TIFFTagLocation. If TIFFTagLocation is 0, then the value is SHORT, and is contained in the "Value\_Offset" entry. Otherwise, the type of the value is implied by the TIFF-Type of the tag containing the value.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/KeyEntrySet.count *The third unsigned short integer in each Key Entry Set SHALL hold the number of values in the key (the Count). If TIFFTagLocation=0, Count=1 is implied.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/KeyEntrySet.valueOffset *The fourth unsigned short integer in each Key Entry Set SHALL hold the Value\_Offset. Value\_Offset indicates the index-offset \*into\* the TagArray indicated by TIFFTagLocation, if it is nonzero. If TIFFTagLocation is 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.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/KeyEntrySet.shortKeyValues *Following the KeyEntry definitions, the GeoKeyDirectoryTag MAY hold values for keys that are short integers. (NOT CLEAR, maybe the following: If TIFFTagLocation=0, then Value\_Offset contains the actual (SHORT) value of the Key, and Count=1 is implied. shortKeyValues specifies that value. )* |

### Requirements Class ModelPixelScaleTag

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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 hold three values (ScaleX, ScaleY, ScaleZ)* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ModelPixelScaleTag.independent *The ModelPixelScaleTag SHALL be independent of the Xposition, Yposition, and Orientation TIFF tags* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ModelPixelScaleTag.reversal *Negative values of the ModelPixelScaleTag components SHALL indicate simple reversals of orientation between raster and model space (e.g. horizontal or vertical flips)* |

### Requirements Class ModelTag

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ModelTag | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ModelTag.count *If the ModelTiePointTag and ModelPixelScaleTags are defined the ModelTransformationTag SHALL not be used* |

### Requirements Class ModelTiePointTag

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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 MAY hold any number of tiepoints* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ModelTiePointTag.size *The ModelTiePointTag SHALL include six values for each tiepoint* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ModelTiePointTag.independent *The ModelTiePointTag SHALL be independent of the Xposition, Yposition, and Orientation TIFF tags* |

### Requirements Class ModelTransformationTag

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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 hold sixteen values* |

### Requirements Class PCSCitationGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/PCSCitationGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/PCSCitationGeoKey.ID *The PCSCitationGeoKey SHALL have ID = 3073* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/PCSCitationGeoKey.type *The PCSCitationGeoKey SHALL have type = ASCII* |

### Requirements Class ProjectedCSTypeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCSTypeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCSTypeGeoKey.ID *The ProjectedCSTypeGeoKey SHALL have ID = 3072* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCSTypeGeoKey.type *The ProjectedCSTypeGeoKey SHALL have type = SHORT* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjectedCSTypeGeoKey.private *ProjectedCSTypeGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class ProjectionGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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* |

### Requirements Class ProjCoordTransGeoKey

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

### Requirements Class ProjLinearUnitsGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearUnitsGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearUnitsGeoKey.ID *The ProjLinearUnitsGeoKey SHALL have ID = 3076* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearUnitsGeoKey.type *The ProjLinearUnitsGeoKey SHALL have type = SHORT* |

### Requirements Class ProjLinearUnitSizeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearUnitSizeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearUnitSizeGeoKey.ID *The ProjLinearUnitSizeGeoKey SHALL have ID = 3077* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearUnitSizeGeoKey.type *The ProjLinearUnitSizeGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjLinearUnitSizeGeoKey.units *The ProjLinearUnitSizeGeoKey SHALL have units = meters* |

### Requirements Class ProjStdParallel1GeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjStdParallel1GeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStdParallel1GeoKey.ID *The ProjStdParallel1GeoKey SHALL have ID = 3078* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStdParallel1GeoKey.type *The ProjStdParallel1GeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStdParallel1GeoKey.units *The ProjStdParallel1GeoKey SHALL have units = GeogAngularUnit* |

### Requirements Class ProjStdParallel2GeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjStdParallel2GeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStdParallel2GeoKey.ID *The ProjStdParallel2GeoKey SHALL have ID = 3079* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStdParallel2GeoKey.type *The ProjStdParallel2GeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStdParallel2GeoKey.units *The ProjStdParallel2GeoKey SHALL have units = GeogAngularUnit* |

### Requirements Class ProjNatOriginLongGeoKe

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjNatOriginLongGeoKe | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjNatOriginLongGeoKe.ID *The ProjNatOriginLongGeoKe SHALL have ID = 3080* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjNatOriginLongGeoKe.type *The ProjNatOriginLongGeoKe SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjNatOriginLongGeoKe.units *The ProjNatOriginLongGeoKe SHALL have units = GeogAngularUnit* |

### Requirements Class ProjNatOriginLatGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjNatOriginLatGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjNatOriginLatGeoKey.ID *The ProjNatOriginLatGeoKey SHALL have ID = 3081* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjNatOriginLatGeoKey.type *The ProjNatOriginLatGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjNatOriginLatGeoKey.units *The ProjNatOriginLatGeoKey SHALL have units = GeogAngularUnit* |

### Requirements Class ProjFalseEastingGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseEastingGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseEastingGeoKey.ID *The ProjFalseEastingGeoKey SHALL have ID = 3082* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseEastingGeoKey.type *The ProjFalseEastingGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseEastingGeoKey.units *The ProjFalseEastingGeoKey SHALL have units = ProjLinearUnit* |

### Requirements Class ProjFalseNorthingGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseNorthingGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseNorthingGeoKey.ID *The ProjFalseNorthingGeoKey SHALL have ID = 3083* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseNorthingGeoKey.type *The ProjFalseNorthingGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseNorthingGeoKey.units *The ProjFalseNorthingGeoKey SHALL have units = ProjLinearUnit* |

### Requirements Class ProjFalseOriginLongGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginLongGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginLongGeoKey.ID *The ProjFalseOriginLongGeoKey SHALL have ID = 3084* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginLongGeoKey.type *The ProjFalseOriginLongGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginLongGeoKey.units *The ProjFalseOriginLongGeoKey SHALL have units = GeogAngularUnit* |

### Requirements Class ProjFalseOriginLatGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginLatGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginLatGeoKey.ID *The ProjFalseOriginLatGeoKey SHALL have ID = 3085* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginLatGeoKey.type *The ProjFalseOriginLatGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginLatGeoKey.units *The ProjFalseOriginLatGeoKey SHALL have units = GeogAngularUnit* |

### Requirements Class ProjFalseOriginEastingGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginEastingGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginEastingGeoKey.ID *The ProjFalseOriginEastingGeoKey SHALL have ID = 3086* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginEastingGeoKey.type *The ProjFalseOriginEastingGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginEastingGeoKey.units *The ProjFalseOriginEastingGeoKey SHALL have units = ProjLinearUnit* |

### Requirements Class ProjFalseOriginNorthingGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginNorthingGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginNorthingGeoKey.ID *The ProjFalseOriginNorthingGeoKey SHALL have ID = 3087* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginNorthingGeoKey.ID *The ProjFalseOriginNorthingGeoKey SHALL have ID = 3091* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginNorthingGeoKey.type *The ProjFalseOriginNorthingGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginNorthingGeoKey.type *The ProjFalseOriginNorthingGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginNorthingGeoKey.units *The ProjFalseOriginNorthingGeoKey SHALL have units = ProjLinearUnit* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjFalseOriginNorthingGeoKey.units *The ProjFalseOriginNorthingGeoKey SHALL have units = ProjLinearUnit* |

### Requirements Class ProjCenterLongGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterLongGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterLongGeoKey.ID *The ProjCenterLongGeoKey SHALL have ID = 3088* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterLongGeoKey.type *The ProjCenterLongGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterLongGeoKey.units *The ProjCenterLongGeoKey SHALL have units = GeogAngularUnit* |

### Requirements Class ProjCenterLatGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterLatGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterLatGeoKey.ID *The ProjCenterLatGeoKey SHALL have ID = 3089* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterLatGeoKey.type *The ProjCenterLatGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterLatGeoKey.units *The ProjCenterLatGeoKey SHALL have units = GeogAngularUnit* |

### Requirements Class ProjCenterEastingGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterEastingGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterEastingGeoKey.ID *The ProjCenterEastingGeoKey SHALL have ID = 3090* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterEastingGeoKey.type *The ProjCenterEastingGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjCenterEastingGeoKey.units *The ProjCenterEastingGeoKey SHALL have units = ProjLinearUnit* |

### Requirements Class ProjScaleAtNatOriginGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjScaleAtNatOriginGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjScaleAtNatOriginGeoKey.ID *The ProjScaleAtNatOriginGeoKey SHALL have ID = 3092* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjScaleAtNatOriginGeoKey.type *The ProjScaleAtNatOriginGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjScaleAtNatOriginGeoKey.units *The ProjScaleAtNatOriginGeoKey SHALL have units = none* |

### Requirements Class ProjScaleAtCenterGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjScaleAtCenterGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjScaleAtCenterGeoKey.ID *The ProjScaleAtCenterGeoKey SHALL have ID = 3093* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjScaleAtCenterGeoKey.type *The ProjScaleAtCenterGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjScaleAtCenterGeoKey.units *The ProjScaleAtCenterGeoKey SHALL have units = none* |

### Requirements Class ProjAzimuthAngleGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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 = GeogAzimuthUnit* |

### Requirements Class ProjStraightVertPoleLongGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/ProjStraightVertPoleLongGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStraightVertPoleLongGeoKey .ID *The ProjStraightVertPoleLongGeoKey SHALL have ID = 3095* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStraightVertPoleLongGeoKey .type *The ProjStraightVertPoleLongGeoKey SHALL have type = DOUBLE* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/ProjStraightVertPoleLongGeoKey .units *The ProjStraightVertPoleLongGeoKey SHALL have units = GeogAngularUnit* |

### Requirements Class VerticalCitationGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/VerticalCitationGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalCitationGeoKey.ID *The VerticalCitationGeoKey SHALL have ID = 4097* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalCitationGeoKey.type *The VerticalCitationGeoKey SHALL have type = ASCII* |

### Requirements Class VerticalCSTypeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/VerticalCSTypeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalCSTypeGeoKey.ID *The VerticalCSTypeGeoKey SHALL have ID = 4096* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalCSTypeGeoKey.type *The VerticalCSTypeGeoKey SHALL have type = SHORT* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalCSTypeGeoKey.reserved *VerticalCSTypeGeoKey values in the range 1-4999 and 6000-32766 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalCSTypeGeoKey.private *VerticalCSTypeGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class VerticalDatumGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| 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.VertDatum *VerticalDatumGeoKey values in the range 1-16383 SHALL be Vertical Datum Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey.reserved *VerticalDatumGeoKey values in the range 16384-32766 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalDatumGeoKey.private *VerticalDatumGeoKey values in the range 32768-65535 SHALL be private* |

### Requirements Class VerticalUnitsGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/VerticalUnitsGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalUnitsGeoKey.ID *The VerticalUnitsGeoKey SHALL have ID = 4099* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalUnitsGeoKey.type *The VerticalUnitsGeoKey SHALL have type = SHORT* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalUnitsGeoKey.obsolete *VerticalUnitsGeoKey values in the range 1-2000 SHALL be obsolete GeoTIFF Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalUnitsGeoKey.reserved *VerticalUnitsGeoKey values in the range 2001-8999 SHALL be reserved* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/VerticalUnitsGeoKey.private *VerticalUnitsGeoKey values in the range 32768-65535 SHALL be private* |

# Media Types for any data encoding(s)

The MIME type for GeoTIFF is image/tiff.

1. Conformance Class Abstract Test Suite (Normative)
   1. Conformance class: GeoTIFF
2. Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Release | Author | Paragraph modified | Description |
| 2014-11-30 | 0.0 | Ted Habermann | Entire Document | Initial document |
| 2016-10-06 | 0.1 | Ted Habermann | Entire Document | Many edits |
|  |  |  |  |  |
|  |  |  |  |  |

1. Bibliography

Association Adobe Developers. (1992). *Revision 6.0*.

GeoTIFF profile for Georeferenced Imagery, DGIWG 108, 2014-03-17, https://portal.dgiwg.org/files/?artifact\_id=5440&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:10.1080/014311697218340

Ritter, Niles, & Ruth, M. (1995). GeoTIFF Format Specification

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

1. The GeoTIFF File Structure (Informative)
   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 geographic as well as projected coordinate systems needs. Projections include UTM, US State Plane and National Grids, as well as the underlying projection types such as Transverse Mercator, Lambert Conformal Conic, etc. 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 projection types, coordinate systems, datums, ellipsoids, etc. The projection, datums and ellipsoid codes are derived from the EPSG list compiled by the Petrotechnical Open Software Corporation (POSC), and mechanisms for adding further international projections, datums and ellipsoids has 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 Projected coordinate 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.

* 1. 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 should be a SHORT type, rather than an ASCII string. This avoids common 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.
  1. 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 now available; see the "References" (section 5 of Ritter and Ruth, 1995) for the location.

* 1. 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).

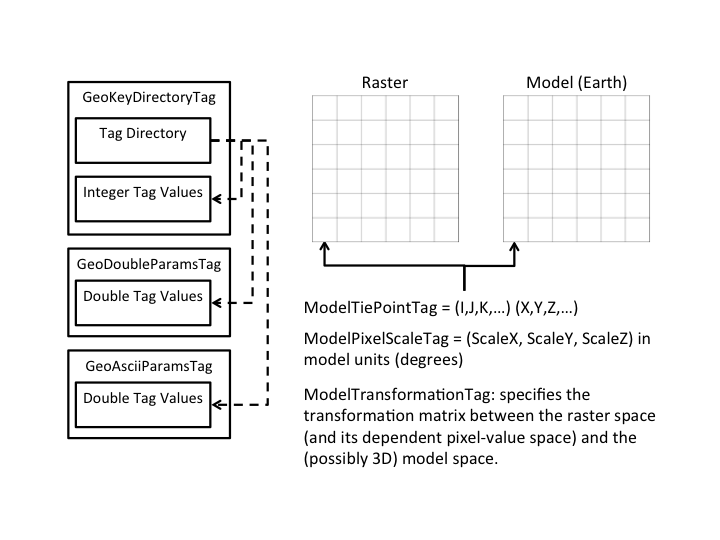
To avoid these problems, a GeoTIFF file stores projection parameters in a set of "Keys" which are virtually identical in function to a "Tag", but has one more level of abstraction above TIFF. Effectively, it is a sort of "Meta-Tag". A Key works with formatted tag-values of a TIFF file the way that a 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.

Figure 1. Schematic structure of GeoTIFF file and tags.

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

Owner: SPOT Image, Inc.

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 is of 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

Owner: SPOT Image, Inc.

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

Owner: SPOT Image, Inc.

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), 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 and section 8.2.3 of Ritter and Ruth, 1995 where the parameters for defining Coordinate Systems and their underlying projections are defined.

* 1. Coordinate Systems in GeoTIFF

GeoTIFF has been designed so that standard map coordinate system definitions can be readily stored in a single registered TIFF tag. It has also been designed to allow the description of coordinate system definitions that are non-standard, and for the description of transformations between coordinate systems, through the use of three or four additional TIFF tags.

However, in order for the information to be correctly exchanged between various clients and providers of GeoTIFF, it is important to establish a common system for describing map projections.

In the TIFF/GeoTIFF framework, there are essentially three different spaces upon which coordinate systems 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.

* + 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 2.5.2.2.

* + 1. Raster Coordinate Systems
       1. 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.

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.

* + - 1. 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.

* + - * 1. "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

* + - * 1. "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).

* + 1. Model Coordinate Systems

The following methods of describing spatial model locations (as opposed to raster) are recognized in GeoTIFF:

Geographic coordinates

Geocentric coordinates

Projected coordinates

Vertical coordinates

Geographic, geocentric and projected coordinates are all imposed on models of the earth. To describe a location uniquely, a coordinate set must be referenced to an adequately defined coordinate system. If a coordinate system is from the GeoTIFF standard definitions, the only reference required is the standard coordinate system code/name. If the coordinate system is non-standard, it must be defined. The required definitions are described below.

Projected coordinates, local grid coordinates, and (usually) geographical coordinates, 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 it is necessary to consider height as a second one-dimensional vertical coordinate system.

To georeference an image in GeoTIFF, you must specify a Raster Space coordinate system, choose a horizontal model coordinate system, and a transformation between these two, as will be described in section D.6.

* + - 1. Geographic Coordinate Systems

Geographic Coordinate Systems are those that relate angular latitude and longitude (and optionally geodetic height) to an actual point on the earth. The process by which this is accomplished is rather complex, and so we describe the components of the process in detail here.

* + - * 1. Ellipsoidal Models of the Earth

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 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 mapping. The size and shape of these 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).

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.

Other ellipsoid parameters needed for mapping applications, for example the square of the eccentricity, can easily be calculated by an application 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 (triaxial) ellipsoid might be required, where (b) would represent the semi-median axis and (c) the semi-minor axis.

Numeric codes for ellipsoids regularly used for earth mapping are included in the GeoTIFF reference lists (see also section 7.2.10).

* + - * 1. Latitude and Longitude

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 mapping.

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.

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.

* + - * 1. Geodetic Datums

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 geographical coordinates of a point will change.

Conversely, for geographical 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. An exact geodetic definition of geodetic datums is beyond the current scope of GeoTIFF. However the GeoTIFF standard requires that the geodetic datum being utilized be identified by numerical code. If required, defining parameters for the geodetic datum can be included as a citation.

* + - * 1. Defining Geographic Coordinate Systems

In summary, geographic coordinates are only unique if qualified by the code of the geographic coordinate system to which they belong. A geographic coordinate system has two axes, latitude and longitude, which are only unambiguous when both of the related prime meridian and geodetic datum are given, and in turn the geodetic datum definition includes the definition of an ellipsoid. The GeoTIFF standard includes a list of frequently used geographic coordinate systems and their component ellipsoids, geodetic datums and prime meridians. Within the GeoTIFF standard a geographic coordinate system can be identified either by

the code of a standard geographic coordinate system

or by

a user-defined system.

The user is expected to provide geographic coordinate system code/name, geodetic datum code/name, ellipsoid code (if in standard) or ellipsoid name and two defining parameters (a) and either (1/f) or (b), and prime meridian code (if in standard) or name and longitude relative to Greenwich.

* + - 1. Geocentric Coordinate Systems

A geocentric coordinate system is a 3-dimensional coordinate system with its origin at or near the center of the earth and with 3 orthogonal axes. 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 or parallel to the plane of the equator and passes through its intersection with the Greenwich meridian, and the Y-axis is in the plane of the equator forming a right-handed coordinate system with the X and Z axes.

Geocentric coordinate systems are not frequently used for describing locations, but they are often utilized as an intermediate step when transforming between geographic coordinate systems. (Coordinate system transformations are described in section D.6 below).

In the GeoTIFF standard, a geocentric coordinate system can be identified, either

through the geographic code (which in turn implies a datum),

or

through a user-defined name.

* + - 1. Projected Coordinate Systems

Although a geographical coordinate system is mathematically two-dimensional, it describes a three-dimensional object and cannot be represented on a plane surface without distortion. Map projections are transformations of geographical coordinates to plane coordinates in which the characteristics of the distortions are controlled. A map projection consists of a coordinate system transformation method and a set of defining parameters. A projected coordinate system (PCS) is a two dimensional (horizontal) coordinate set which, for a specific map projection, has a single and unambiguous transformation to a geographic coordinate system.

In GeoTIFF PCS's are defined using the POSC/EPSG system, in which the PCS planar coordinate system, the Geographic coordinate system, and the transformation between them, are broken down into simpler logical components. Here are schematic formulas showing how the Projected Coordinate Systems and Geographic Coordinates Systems are encoded:

Projected\_CS = Geographic\_CS + Projection

Geographic\_CS = Angular\_Unit + Geodetic\_Datum + Prime\_Meridian

Projection = Linear Unit + Coord\_Transf\_Method + CT\_Parameters

Coord\_Transf\_Method = { TransverseMercator | LambertCC | ...}

CT\_Parameters = {OriginLatitude + StandardParallel+...}

(See also the Reference Parameters documentation in section D.5.4).

Notice that "Transverse Mercator" is not referred to as a "Projection", but rather as a "Coordinate Transformation Method"; in GeoTIFF, as in EPSG/POSC, the word "Projection" is reserved for particular, well-defined systems in which both the coordinate transformation method, its defining parameters, and their linear units are established.

Several tens of coordinate transformation methods have been developed. Many are very similar and for practical purposes can be considered to give identical results. For example in the GeoTIFF standard Gauss-Kruger and Gauss-Boaga projection types are considered to be of the type Transverse Mercator. GeoTIFF includes a listing of commonly used projection defining parameters.

Different algorithms require different defining parameters. A future version of GeoTIFF will include formulas for specific map projection algorithms recommended for use with listed projection parameters.

To limit the magnitude of distortions of projected coordinate systems, the boundaries of usage are sometimes restricted. To cover more extensive areas, two or more projected coordinate systems may be required. In some cases many of the defining parameters of a set of projected coordinate systems will be held constant.

The GeoTIFF standard does not impose a strict hierarchy onto such zoned systems such as US State Plane or UTM, but considers each zone to be a discrete projected coordinate system; the ProjectedCSTypeGeoKey code value alone is sufficient to identify the standard coordinate systems.

Within the GeoTIFF standard a projected coordinate system can be identified either by

the code of a standard projected coordinate system

or by

a user-defined system.

User-define projected coordinate systems may be defined by defining the Geographic Coordinate System, the coordinate transformation method and its associated parameters, as well as the planar system's linear units.

* + - 1. Vertical Coordinate Systems

Many uses of GeoTIFF will be limited to a two-dimensional, horizontal, description of location for which geographic coordinate systems and projected coordinate systems are adequate. If a three-dimensional description of location is required, GeoTIFF allows this either through the use of a geocentric coordinate system or by defining a vertical coordinate system and using this together with a geographic or projected coordinate system.

In general usage, elevations and depths are referenced to a surface at or close to the geoid. Through increasing use of satellite positioning systems the ellipsoid is increasingly being used as a vertical reference surface. The relationship between the geoid and an ellipsoid is in general not well known, but is required when coordinate system transformations are to be executed.

* + 1. Reference Parameters

Most of the numerical coding systems and coordinate system definitions are based on the hierarchical system developed by EPSG/POSC. The complete set of EPSG tables used in GeoTIFF is available at:

ftp://ftpmcmc.cr.usgs.gov/release/geotiff/jpl-mirror/tables

or:

<ftp://mtritter.jpl.nasa.gov/pub/tiff/geotiff/tables>

Appended below is the README.TXT file that accompanies the tables of defining parameters for those codes:

+-----------------------------------+

| EPSG Geodesy Parameters |

| version 2.1, 2nd June 1995. |

+-----------------------------------+

The European Petroleum Survey Group (EPSG) has compiled and is

distributing this set of parameters defining various geodetic

and cartographic coordinate systems to encourage

standardisation across the Exploration and Production segment

of the oil industry. The data is included as reference data

in the GeoTIFF data exchange specification, in Iris21 the

Petroconsultants data model, and in Epicentre, the POSC data

model. Parameters map directly to the POSC Epicentre model

v2.0, except for data item codes which are included in the

files for data management purposes. Geodetic datum parameters

are embedded within the geographic coordinate system file.

This has been done to ease parameter maintenance as there is a

high correlation between geodetic datum names and geographic

coordinate system names. The Projected Coordinate System v2.0

tabulation consists of systems associated with locally used

projections. Systems utilising the popular UTM grid system

have also been included.

Criteria used for material in these lists include:

- information must be in the public domain: "private" data

is not included.

- data must be in current use.

- parameters are given to a precision consistent with

coordinates being to a precision of one centimetre.

The user assumes the entire risk as to the accuracy and the

use of this data. The data may be copied and distributed

subject to the following conditions:

1) All data must then be copied without modification

and all pages must be included;

2) All components of this data set must be distributed

together;

3) The data may not be distributed for profit by any

third party; and

4) Acknowledgement to the original source must be

given.

INFORMATION PROVIDED IN THIS DOCUMENT IS PROVIDED "AS IS"

WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.

Data is distributed on MS-DOS formatted diskette in comma-

separated record format. Additional copies may be obtained

from Jean-Patrick Girbig at the address below at a cost of

US$100 to cover media and shipping, payment to be made in

favour of Petroconsultants S.A at Union Banque Suisses,

1211 Geneve 11, Switzerland (compte number 403 458 60 K).

The data is to be made available on a bulletin board shortly.

Shipping List

-------------

This data set consists of 8 files:

PROJCS.CSV Tabulation of Projected Coordinate Systems to

which map grid coordinates may be referenced.

GEOGCS.CSV Tabulation of Geographic Coordinate Systems to

which latitude and longitude coordinates may be

referenced. This table includes the equivalent

geocentric coordinate systems and also the

geodetic datum, reference to which allows latitude

and longitude or geocentric XYZ to uniquely

describe a location on the earth.

VERTCS.CSV Tabulation of Vertical Coordinate Systems to

which heights or depths may be referenced. This

table is currently in an early form.

PROJ.CSV Tabulation of transformation methods and

parameters through which Projected Coordinate

Systems are defined and related to Geographic

Coordinate Systems.

ELLIPS.CSV Tabulation of reference ellipsoids upon which

geodetic datums are based.

PMERID.CSV Tabulation of prime meridians upon which geodetic

datums are based.

UNITS.CSV Tabulation of length units used in Projected and

Vertical Coordinate Systems and angle units used

in Geographic Coordinate Systems.

README.TXT This file.

* 1. 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.

* + 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

Owner: Intergraph

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 future 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

Owner: SoftDesk

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

Owner: JPL Cartographic Applications Group

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

| | Tz = Z - K/Sz (if not 0)

| 0.0 0.0 Sz Tz |

| |

| 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.

Note: In Revision 0.2 and earlier, another tag was used for this matrix, which has been renamed as follows:

IntergraphMatrixTag

Tag = 33920 (8480.H)

Type = DOUBLE

N = 17 (Intergraph implementation) or 16 (GeoTIFF 0.2 impl.)

Owner: Intergraph

This tag conflicts with an internal software implementation at Intergraph, and so its use is no longer encouraged. A GeoTIFF reader should look first for the new tag, and only if it is not found should it check for this older tag. If found, it should only consider it to be contain valid GeoTIFF matrix information if the tag-count is 16; the Intergraph version uses 17 values.

* + 1. Coordinate Transformation Data Flow

The dataflow of the various GeoTIFF parameter datasets is based upon the EPSG/POSC configuration. Here is the text of the description accompanying the EPSG parameter tables:

The data files (.CSV) have a hierarchical structure:

+---------------------------+ +----------------------------+

| VERTCS | | PROJCS |

+---------------------------+ +----------------------------+

|Vertical Coordinate Systems| |Projected Coordinate Systems|

+-------------+-------------+ +------------+---------------+

| |

+--------+ |

| |

| +--------------------------+

| | |

| | +-------------+---------------+

| | | GEOGCS |

| | +-----------------------------+

| | |Geographic Coordinate Systems|

| | |Geocentric Coordinate Systems|

| | +-----------------------------+

| | | Geodetic Datums |

| | +-------------+---------------+

| | |

| | +--------+-------+

| | | |

| +------+-----+ +------+-----+ +------+-------+

| | PROJ | | ELLIPS | | PMERID |

| +------------+ +------------+ +--------------+

| | Projection | | Ellipsoid | |Prime Meridian|

| | Parameters | | Parameters | | Parameters |

| +------+-----+ +------+-----+ +------+-------+

| | | |

+------------+-----------+-----+----------------+

|

+-------------+------------+

| UNITS |

+--------------------------+

| Linear and Angular Units |

+--------------------------+

The parameter listings are "living documents" and will be

updated by the EPSG from time to time. Any comment or

suggestions for improvements should be directed to:

Jean-Patrick Girbig, or Roger Lott,

Manager Cartography, Head of Survey,

Petroconsultants S.A., BP Exploration,

PO Box 152, Uxbridge One,

24 Chemin de la Marie, Harefield Road,

1258 Perly-Geneva, Uxbridge,

Switzerland. Middlesex UB8 1PD,

England.

Internet:

lottrj@txpcap.hou.xwh.bp.com

Requests for the inclusion of new data should include supporting

documentation. Requests for changing existing data should include

reference to both the name and code of the item.

* + 1. 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 system (PCS). If you are geocoding this data

set, then the model space is defined to be the corresponding

geographic, geocentric or Projected coordinate system (skip

to the "Cookbook" section 2.7.3 first to do determine this).

Step 3: Identify the nature of the transformations needed to tie

the raster data down to the model space coordinate 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 system.

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

* 1. Geocoding Raster Data
     1. General Approach

A geocoded image is a georeferenced image as described in section D.6, which also specifies a model space coordinate system (CS) 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 CS

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.

* + 1. Cookbook for Geocoding Data

Step 1: Determine the Coordinate system type of the raster data, based on the nature of the data: pixels derived from scanners or other optical devices represent areas, and most commonly will use the RasterPixelIsArea coordinate system. Pixel data such as digital elevation models represent points, and will probably use RasterPixelIsPoint coordinates.

Store in: GTRasterTypeGeoKey

Step 2: Determine which class of model space coordinates are most natural for this dataset:Geographic, Geocentric, or Projected Coordinate System. Usually this will be PCS.

Store in: GTModelTypeGeoKey

Step 3: This step depends on the GTModelType:

case PCS: Determine the PCS projection system. Most of the PCS's used in standard State Plane and national grid systems are defined, so check this list first; the EPSG index in section 6.4 may be useful for this purpose.

Store in: ProjectedCSTypeGeoKey, ProjectedCSTypeGeoKey

If coded, it will not be necessary to specify the Projection datum, etc for this case, since all of those parameters are determined by the ProjectedCSTypeGeoKey code. Skip to step 4 from here.

If none of the coded PCS's match your system, then this is a user-defined PCS. Use the Projection code list to check for standard projection systems.

Store in: ProjectionGeoKey and skip to Geographic CS case.

If none of the Projection codes match your system, then this is a user-defined projection. Use the ProjCoordTransGeoKey to specify the coordinate transformation method (e.g. Transverse Mercator), and all of the associated parameters of that method. Also define the linear units used in the planar coordinate system.

Store in: ProjCoordTransGeoKey, ProjLinearUnitsGeoKey <and other CT related parameter keys>

Now continue on to define the Geographic CS, below.

case GEOCENTRIC:

case GEOGRAPHIC: Check the list of standard GCS's and use the corresponding code. To use a code both the Datum, Prime Meridian, and angular units must match those of the code.

Store in: GeographicTypeGeoKey and skip to Step 4.

If none of the coded GCS's match exactly, then this is a user-defined GCS. Check the list of standard datums, Prime Meridians, and angular units to define your system.

Store in: GeogGeodeticDatumGeoKey, GeogAngularUnitsGeoKey, GeogPrimeMeridianGeoKey and skip to Step 4.

If none of the datums match your system, you have a user-defined datum, which is an odd system, indeed. Use the GeogEllipsoidGeoKey to select the appropriate ellipsoid or use the GeogSemiMajorAxisGeoKey, GeogInvFlatteningGeoKey to define, and give a reference using the GeogCitationGeoKey.

Store in: GeogEllipsoidGeoKey, etc. and go to Step 4.

Step 4: Install the GeoKeys/codes into the GeoKeyDirectoryTag, and the DOUBLE and ASCII key values into the corresponding value-tags.

Step 5: Having completely defined the Raster & Model coordinate system, go to Cookbook section 2.6.2 and use the Georeferencing Tags to tie the raster image down onto the Model space.

* 1. Examples

Examples of how GeoTIFF may be implemented at the Tag and GeoKey level, following the general "Cookbook" approach above.

* + 1. Common Examples
       1. UTM Projected Aerial Photo

We have an aerial photo which has been orthorectified and resampled to a UTM grid, zone 60, using WGS84 datum; 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)

GTRasterTypeGeoKey = 1 (RasterPixelIsArea)

ProjectedCSTypeGeoKey = 32660 (PCS\_WGS84\_UTM\_zone\_60N)

PCSCitationGeoKey = "UTM Zone 60 N with WGS84"

Notes:

1) We did not need to specify the GCS lat-long, since the

PCS\_WGS84\_UTM\_zone\_60N codes implies particular

GCS and units already (WGS\_84 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.

* + - 1. 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 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)

GTRasterTypeGeoKey = 1 (RasterPixelIsArea)

ProjectedCSTypeGeoKey = 32139 (PCS\_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).

* + - 1. 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)

GTRasterTypeGeoKey = 1 (RasterPixelIsArea)

GeographicTypeGeoKey = 4267 (GCS\_NAD27)

ProjectedCSTypeGeoKey = 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.

* + - 1. 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)

GTRasterTypeGeoKey = 1 (RasterPixelIsArea)

GeographicTypeGeoKey = 4326 (GCS\_WGS\_84)

* + 1. Less Common Examples
       1. Unrectified Aerial photo, known tiepoints, in degrees.

We have an aerial photo, and know only the WGS84 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)

GTRasterTypeGeoKey = 1 (RasterPixelIsArea)

GeographicTypeGeoKey = 4326 (GCS\_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.

* + - 1. 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)

GTRasterTypeGeoKey = 1 (RasterPixelIsArea)

ProjectedCSTypeGeoKey = 27700 (PCS\_British\_National\_Grid)

PCSCitationGeoKey = "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.

* + - 1. Digital Elevation Model

The DMA stores digital elevation models using an equirectangular projection, so that it may be indexed with WGS84 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 WGS84 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)

GTRasterTypeGeoKey = 2 (RasterPixelIsPoint)

GeographicTypeGeoKey = 4326 (GCS\_WGS\_84)

VerticalCSTypeGeoKey = 5030 (VertCS\_WGS\_84\_ellipsoid)

VerticalCitationGeoKey = "WGS 84 Ellipsoid"

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.

1. DGIWG Requirements (Informative)
   1. Introduction

The Defence Geospatial Information Working Group (DGIWG) published a profile of GeoTIFF (). Requirements from that profile are included here for information.

* 1. Requirements Class DGIWG

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.bitsPerSample  *The BitsPerSample field in the TIFF Image File Directory defines the number of bits per component* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.byteOrder  *The first two bytes of the GeoTIFF file SHALL be equal to "I" (ASCII) (49 in hexadecimal) for TIFF files encoded using â€˜Little-Endianâ€™ and SHALL be equal to "M" (ASCII) (4D in hexadecimal) for TIFF files encoded using â€˜Big-Endianâ€™* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.griddedValueDataTypes  *For gridded data (e.g. elevation data, matrices of lat/lon values, etc.), the range (data) values MAY be stored in additional representations to include 8-bit and 16-bit signed integer and 32-bit floating point.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.IFD  *There must be at least 1 IFD in a TIFF file and each IFD must have at least one entry.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.IFDCount  *The maximum nuber of IFDs in a GeoTIFF is two, with the second IFD only used to support a transparency mask.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.imageryValueDataTypes  *For imagery, the range (data) values SHALL be unsigned integer data, 8 or 16-bits-per-band.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.DateTime  *The format for the field in ASCII type is â€œYYYY:MM:DD HH:MM:SSâ€ with 24 hour time used for the hours and one space character between the date and time, and one terminating NUL character. The length of the string, including the terminating NUL, is 20 bytes. All dates and times shall be expressed in Coordinated Universal Time (UTC).* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.DGIWGReference  *WGS84 + may include Reference document citation (EPSG, DGIWG Registry or [DMA TR 8350.2])* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.DGIWGValues  *SHALL be 4326 (GCS\_WGS84) or 4030 (GCSE\_WGS84, not recommended by GeoTIFF)* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.cardinality  *The private TIFF tag for holding XML metadata MAY be used more than once in a single GeoTIFF file.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.ID  *The private TIFF tag for holding XML metadata SHALL have ID=50909* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.Name  *The private TIFF tag for holding XML metadata SHALL be named GEO\_METADATA* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.count  *The ModelTiePointTag SHALL only hold one tiepoint that corresponds to the upper left corner of the image.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.notAllowed  *The ModelTransformationTag SHALL not be used.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.KeyValues  *326zz â€“ UTM Northern Hemisphere, 327zz â€“ UTM Southern Hemisphere (Where zz is the UTM zone number), Other PCS allowed by this standard (in conformance with DGIWG Geodetic Codes and Parameters Registry) 12 Present only for cartographic data. In this case, GTModelTypeGeoKey = 1 and GeographicTypeGeoKey is absent* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.KeyValues  *WGS84 Ellipsoid, EGM84, EGM96, EGM2008, MSL height, MSL depth , or the name of the Sounding datum identified in DGIWG Geodetic registry (S-1 to S-40), or description os user-defined vertical CRS* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.KeyValues  *4979 (WGS84 3D ellipsoid), 5773 (EGM96), 3855 (EGM08), 5798 (EGM84), 5714 (MSL height), 5715 (MSL depth), 32767 for other Sounding datums identified in DGIWG Geodetic registry, or user defined Vertical CRS* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.KeyValues  *9001 (meaning Linear\_Meter)* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.ID  *The private TIFF tag for defining void areas SHALL have ID = 42113.* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/DGIWG.Name  *The private TIFF tag for defining void areas SHALL be named GDAL\_NODATA. (GDAL\_NODATA is not defined in the GeoTIFF standard, however, this is an private TIFF tag that may be used for the purpose of declaring these values.)* |

1. European Petroleum Survey Group (EPSG) Requirements (Informative)
   1. Introduction

The current GeoTIFF Specification (Ritter and Ruth, 1995) defined ranges for a number of GeoTIFF keys in terms of ESPG codes. Since that specification was written, the number of ESPG codes has expanded significantly and the original ranges are no longer valid. GeoTIFF requirements that refer to those ranges are included here for completeness with the caveat that they may no longer be meaningful.

* 1. Requirements Related to EPSG Code Ranges
     1. Requirements Class EPSG-GeodeticDatumGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeodeticDatumGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-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/EPSG-GeodeticDatumGeoKey.EPSGEllipsoid *GeodeticDatumGeoKey values in the range 6000-6199 SHALL be EPSG Datum Based on Ellipsoid only* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeodeticDatumGeoKey.EPSGDatum *GeodeticDatumGeoKey values in the range 6200-6999 SHALL be EPSG Datum Based on EPSG Datum* |

* + 1. Requirements Class EPSG-GeogEllipsoidGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogEllipsoidGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogEllipsoidGeoKey.obsolete *GeogEllipsoidGeoKey values in the range 1-1000 SHALL be obsolete EPSG/POSC Datum Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogEllipsoidGeoKey.EPSGEllipsoid *GeogEllipsoidGeoKey values in the range 7000-7999 SHALL be EPSG Ellipsoid Codes* |

* + 1. Requirements Class EPSG-GeogLinearUnitsGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogLinearUnitsGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogLinearUnitsGeoKey.linear *GeogLinearUnitsGeoKey values in the range 9000-9099 SHALL be EPSG linear units* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogLinearUnitsGeoKey.angular *GeogLinearUnitsGeoKey values in the range 9100-9199 SHALL be EPSG angular units* |

* + 1. Requirements Class EPSG-GeogPrimeMeridianGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogPrimeMeridianGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogPrimeMeridianGeoKey.obsolete *GeogPrimeMeridianGeoKey values in the range 1-100 SHALL be obsolete EPSG/POSC Datum Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeogPrimeMeridianGeoKey.EPSGEllipsoid *GeogPrimeMeridianGeoKey values in the range 8000-8999 SHALL be EPSG Prime Meridian Codes* |

* + 1. Requirements Class EPSG-GeographicTypeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeographicTypeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeographicTypeGeoKey.obsolete *GeographicTypeGeoKey values in the range 1-1000 SHALL be obsolete EPSG/POSC Geographic Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeographicTypeGeoKey.EPSGEllipsoid *GeographicTypeGeoKey values in the range 4000-4199 SHALL be EPSG GCS Based on Ellipsoid only* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-GeographicTypeGeoKey.EPSGDatum *GeographicTypeGeoKey values in the range 4200-4999 SHALL be EPSG GCS Based on EPSG Datum* |

* + 1. Requirements Class EPSG-ProjectedCSTypeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-ProjectedCSTypeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-ProjectedCSTypeGeoKey.obsolete *ProjectedCSTypeGeoKey values in the range 1-1000 SHALL be obsolete EPSG/POSC Datum Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-ProjectedCSTypeGeoKey.EPSGProjection *ProjectedCSTypeGeoKey values in the range 20000-32760 SHALL be EPSG Projection System Codes* |

* + 1. Requirements Class EPSG-VerticalCSTypeGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-VerticalCSTypeGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-VerticalCSTypeGeoKey.EPSGEllipsoid *VerticalCSTypeGeoKey values in the range 5000-5099 SHALL be EPSG Ellipsoid Vertical CS Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-VerticalCSTypeGeoKey.EPSGOrthometric *VerticalCSTypeGeoKey values in the range 5100-5199 SHALL be EPSG Orthometric Vertical CS Codes* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-VerticalCSTypeGeoKey.reservedEPSG *VerticalCSTypeGeoKey values in the range 5200-5999 SHALL be reserved EPSG* |

* + 1. Requirements Class EPSG-VerticalUnitsGeoKey

|  |  |
| --- | --- |
| **Requirements Class** | |
| http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-VerticalUnitsGeoKey | |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-VerticalUnitsGeoKey.linear *VerticalUnitsGeoKey values in the range 9000-9099 SHALL be EPSG linear units* |
| Requirement | http://www.opengis.net/spec/GeoTIFF/0.0/EPSG-VerticalUnitsGeoKey.angular *VerticalUnitsGeoKey values in the range 9100-9199 SHALL be EPSG angular units* |

1. [www.opengeospatial.org/cite](http://www.opengeospatial.org/cite) [↑](#footnote-ref-1)