DGIWG 104(2)

DGIWG profile of JPEG 2000 for Georeferenced and Referenceable Imagery

Document type:	Standard
Document date:	19 April 2023
Edition:	2.1.3 (DP)
Supersedes:	This document supersedes DGIWG 104(2) Ed. 2.1.2, DGIWG profile of JPEG 2000 for Georeferenced Imagery, dated 17 November 2020.
Responsible Party:	Defence Geospatial Information Working Group
Audience:	This document is approved for public release and is available on the DGIWG website, http://www.dgiwg.org/dgiwg/
Abstract:	This document provides a profile for JPEG 2000 for use as a compression format for raster imagery. JPEG 2000 uses *discrete wavelet transform (DWT)* for compressing raster data, as opposed to the JPEG standard, which uses *discrete cosine transform (DCT).* It is a compression technology which is best suited for continuous raster data, such as satellite imagery and aerial photography. This version adds support for Referenceable imagery.
Copyright:	

NOTICE STATEMENT

This standard, DGIWG 104(2) is an implementation profile of OGC's GML in JPEG 2000 (GMLJP2) Encoding Standard Part 1: Core, version 2.1 and is conformant with OGC's GML 3.2.1 standard and GMLCOV application schema and GMLCOV / Coverage Implementation Schema - ReferenceableGridCoverage Extension.

Users should note that this version of the standard:

- is backwards compatible with version 2, dated 2016-06-21;
- adds support for ReferenceableGridCoverage, via a set of mechanisms including byTransformationModel or bySensorModel (based on the OGC GMLJP2 v2.1 version), including satellite, airborne or in-situ imagery.

Table of Contents

1 Scope 2
2 Conformance 2
3 Normative references 2
3.1 ISO Standards 2
3.1.1 JPEG 2000 2
3.1.2 Geospatial metadata 2
3.2 OGC standards 2
3.3 W3C standards ..... **Erreur ! Signet non défini.**
3.4 DGIWG standards 3
3.5 National standards and specifications 3
4 Terminology 3
4.1 Terms and definitions 3
4.1.1 Bit-rate <JPEG 2000> 3
4.1.2 Channel 4
4.1.3 Code-block <JPEG 2000> 4
4.1.4 Codestream <JPEG 2000> 4
4.1.5 Compression ratio <JPEG 2000> 5
4.1.6 Decomposition levels <JPEG 2000> 5
4.1.7 Discrete Wavelet Transform <JPEG 2000> 5
4.1.8 ICC profile 5
4.1.9 JP2 5
4.1.10 JPEG 2000 6
4.1.11 JPX 6
4.1.12 Packet <JPEG 2000> 6
4.1.13 Precinct <JPEG 2000> 6
4.1.14 Rate Allocation <JPEG 2000> 6
4.1.15 Rectified grid 6
4.1.16 Referenceable grid 6
4.1.17 Resolution level <JPEG 2000> 6
4.1.18 Tile 7
4.1.19 Wavelet Transform 7
4.2 Abbreviated terms 7
5 Overview of the JPEG 2000 standard 8
5.1 JPEG 2000 Part 1 8
5.2 JPEG 2000 Part 2 8
5.3 Codestream 8
5.3.1 Compression of the codestream 8
5.3.2 Partition of the codestream 8
5.3.3 Codestream information 10
5.3.4 Error resilience 11
5.3.5 The EBCOT algorithm 11
5.3.6 Progression order 12
5.3.7 JPEG 2000 Profiles 12
5.3.8 Multiple codestreams 13
5.3.9 JPIP 13
5.4 JPEG 2000 Box 14
5.4.1 Box structure 14
5.4.2 Superbox 14
5.5 Opacity 14
5.5.1 Opacity channels (JPEG 2000 Part 1) 15
5.5.2 Opacity in a separate codestream (JPEG 2000 Part 2) 15
5.5.3 Opacity box (JPEG 2000 Part 2) 15
5.6 Integer values constraint of raster and gridded data 15
6 GML objects 16

6.1 Introduction 16
6.2 Coverage 16
6.3 Coordinate reference system 16
6.4 Axis order 17
6.5 Annotations 17
6.6 Metadata 18
7 Embedding GML in JP2 files 18
7.1 JPX file format 18
7.2 Compatibility list in File type box 18
7.2.1 If opacity of Part 2 type is used 18
7.2.2 Profile 19
7.3 Brand field in File type box 19
7.4 Reader requirement box 19
7.5 File suffix 19
Annex A : Abstract Test Suite 20
A.1 Introduction 20
A.1.1 Test classes 20
A.2 OGC GMLJP2 tests 20
A.3 DGIWG GMLJP2 profile tests 20
A.3.1 Class B tests 20
Annex B : Conformance 26
B.1 Purpose 26
B.2 JPEG 2000 Profiles 26
B.3 Conformance classes 26
B.3.1 Class B – Baseline 26
B.3.2 No other conformance classes 26
B.4 Implementation Conformance Report 27
Annex C : JP2 boxes 33
C.1 Required boxes 33
C.1.1 JPEG 2000 Signature box 33
C.1.2 File type box 33
C.1.3 Reader requirement box 33
C.1.4 JPEG 2000 header box 34
C.1.5 Intellectual property rights box 37
C.1.6 Codestream box 40
Annex D : GML encoding 41
D.1 GML objects 41
D.1.1 GML box structure 41
D.1.2 gml.root-instance 41
D.1.3 GMLJP2 URI 42
D.2 GML Georeferencing 42
D.2.1 GML structure 43
D.2.2 GMLJP2RectifiedGridCoverage 43
D.2.3 GMLJP2ReferenceableGridCoverage 46
D.2.4 EPSG georeference in RectifiedGrid@srsName or ReferenceableGrid@srsName 50
D.2.5 CRSDictionary.gml 51
D.3 Metadata 51
Annex E : Metadata 53
E.1 Metadata location 53
E.2 Metadata requirements 53
E.2.1 Metadata conforms to DMF 53
E.2.2 Coverage information in metadata 53
E.3 Security classification 53
Annex F : Use cases (informative) 54

Annex G : JPEG 2000 profile extensions 55
G.1 Profile Extensions 55
G.1.1 Extensions required by baseline GMLJP2 55
G.1.2 Additional extensions used by DGIWG GMLJP2 55
Annex H : Examples of GML coverages 56
H.1 Examples of GML Rectified Grid coverages in JPEG 2000 files 56
H.1.1 A WGS84 LL example of a GMLJP2RectifiedGridCoverage 56
H.1.2 A UTM example of a GMLJP2RectifiedGridCoverage 58
H.2 Examples of GML Referenceable Grid coverages in JPEG 2000 files 59
H.2.1 A SensorModel example of a GMLJP2ReferenceableGridCoverage with a RPC (Rational Polynomial coefficient) replacement model 59
Annex I : Software used for testing purposes 64
I.1 Software recommendations 64
I.2 Kdu_show 64
I.3 XML Spy 64
I.3.1 XML Schema reference 64
I.4 Microsoft .Net Framework 4.0 64

List of Figures

Figure 1 – Marker segment structure 10
Figure 2 – JPEG 2000 box structure 14
Figure 3 – Left and right hand order 17
Figure D.1 - GML box structure 41
Figure D.2 – GML structure 43
Figure D.3 – Image coordinate system 43

List of Tables

Table 4.1 – Typical bit-rates at different quality levels 4
Table 5.1 – Items in a Coding style default structure 10
Table 5.2 – Items in a Coding style component structure 11
Table 5.3 – List of progression orders 12
Table 6.1 – EPSG codes 17
Table 7.1 – File suffixes 19
Table B.1 – JPEG 2000 Profiles 26
Table C.1 – File type box 33
Table C.2 – Reader requirement box 34
Table C.4 – Colour specification box 36
Table C.5 – Channel definition box 37
Table C.6 – IPR fields 38
Table C.7 – Security classification keywords 39
Table D.1 – Required RectifiedGrid content 44
Table D.2 – GML extent in ISO/TS 19139 metadata 44
Table D.3 – Axis order for different reference systems 45
Table D.4 – Required file type content 46
Table D.5 – Required ReferenceableGridBySensorModel content 47
Table D.6 – Required ReferenceableGridByTransformation content 48
Table D.7 – Required ReferenceableGridByVectors content 49
Table D.8 – Required ReferenceableGridByArrays content 50
Table D.9 – Reference codes from the EPSG namespace 51
Table E.1 – DMF requirement classes 53
Table F.1 – Use cases 54

i. Contributing organizations

Nation	Parent organization
Sweden (Lead Nation)	Swedish Armed Forces Geo SE
European Union	European Satellite Centre (EUSC)
France	Institut Géographique National (IGN)
Germany	Bundeswehr Geoinformation Office (BGIO)
United States	National Geospatial-Intelligence Agency (NGA)
Spain	Cartographic Coordination Unit

ii. Revision history

Date	Release	Primary clauses modified	Description
2014-12-19	1.9	All	Updated for OGC GMLJP2 2.0
2015-06-25	1.9.1	Introduction, Scope, Annex A and Annex D	Resolved comments by FR
2016-02-04	2.0	All	Resolved comments by AUS, US and FR
2016-03-01	2.0	Doc id, 6.1 and annex D	Resolved comments by FR
2016-06-21	2.0.1	All	Resolved comments by FR, US and DEU
2019-04-29	2.1		Addition of ReferenceableGridCoverage support (reviewed / validated by P2 in Vienna)
2020-01-23	2.1.1	Section 3 and 4, and throughout the document	Corrections due to terminology harmonization and DGIWG Architecture of DGIWG standards actions
2020-11-17	2.1.2	6.3 and D.2.4 (Table D.9) 6.6	Accept CRS specified in DGIWG Geodetic Codes and Parameters Registry in addition to Table D.9 Clarification of embedded metadata in GMLJP2 file
2023-02-14	2.1.3	Introduction, 3.1, 5.3.7, B2, 5.5.3, 5.6	Normative references update (ISO IEC 154440 part 1 and 2), GMLCOV reference. JPEG2000 profiles, Opacity box usage and Integer value for elevation data clarifications.
2023-04-19	2.1.3 DP	3, Annexes	Release for publication

iii. Future work

This version doesn’t include annotations. Previous version 1.0 used annotations that are not compatible with annotations in OGC GMLJP2 v2.0 and v2.1.

Annotation extension could be added, pending provision of this capability by underlying OGC GMLJP2 standard, or another standardization effort undergone by another standardization body. Currently the requirement for annotations has not been confirmed in this version 2.

Introduction

JPEG 2000 is a compression format for raster imagery. JPEG 2000 uses *discrete wavelet transform (DWT)* for compressing raster data, as opposed to the JPEG standard, which uses *discrete cosine transform (DCT).* It is a compression technology which is best suited for continuous raster data (i.e. satellite imagery and aerial photography). JPEG 2000 images are commonly compressed in a lossy fashion, but also provides a lossless compression mode.

DWT offers better compression than DCT in several ways. The DWT supports a higher compression ratio over that of the DCT which is usually limited to compression ratio of less than 20 in order to avoid the appearance of artefacts due to pixelisation. DWT limits the loss of high frequencies which results in downgraded transitions / pixelisation in the compressed image. The JPEG 2000 format offers many capabilities allowing derivatives of the image to be retrieved in low bandwidth situations. These options include reduced resolution levels, low bit-rate layers (quality layers), and independently encoded blocks allowing access to regions of interest.

Geography Markup Language (GML) Encoding Standard is used for the georeference of the image and for optional metadata, annotation and geographic features that may be provided with the image.

GMLJP2 is an OGC standard for georeferencing JPEG 2000 files with GML. OGC has developed two versions of the GMLJP2 standard, 1.0, which uses GML 3.1.1, and 2.0 with this newest version 2.1 adding support for ReferenceableGridCoverage, which uses GML 3.2.1. That means that OGC GMLJP2 2.0 and 2.1 use OGC GML 3.2.1 Application Schema for Coverages (GMLCOV). GMLCOV gives the option of referenceable image coverage, which means that the image doesn't have to be described by an equi-spaced grid, by means of the GMLCOV / Coverage Implementation Schema - ReferenceableGridCoverage Extension, whose schema is included in the GMLJP2 v2.1. This gives the possibility to use imagery that isn't georectified.

Annotations, that were included in DGIWG GMLJP2 v1.0, are not included in this profile (DGIWG GMLJP2 v2.1. The reason for excluding annotations from this profile, is that OGC GMLJP2 2.0 shall support annotations in a different way than the DGIWG GMLJP2 1.0.

This profile uses OGC GMLJP2 2.1.

1 Scope

This JPEG 2000 profile is developed for the use of georeferenced JPEG 2000 files on the basis of the OGC GMLJP2 2.1 standard. The standard covers raster data, including orthoimagery, elevation and referenceable imagery (not georectified imagery).

2 Conformance

JPEG 2000 files must follow Annex A, B and C to conform to this profile. Additionally, the GMLJP2 files can follow different conformance classes. The conformance classes were introduced in order to enable the use of annotations. Since the annotations are pending on OGC annotations for GMLJP2, this profile has only one conformance class which is defined below:

> *Class B: Baseline profile – A JPEG 2000 file georeferenced according to the constraints defined by this GMLJP2 profile.*

3 Normative references

The following referenced documents are indispensable for the application of this document. Unless otherwise specified, the latest edition of the referenced document (including any amendments) applies.

While every effort has been made to ensure completeness of this list, document users are cautioned that they must meet all specified requirements of documents cited in this standard.

3.1 ISO Standards

3.1.1 JPEG 2000

- ISO/IEC 15444-1:2019, Information technology - JPEG 2000 Image coding system – Part 1: Core coding System
- ISO/IEC 15444-2:2021, Information technology - JPEG 2000 Image coding system – Part 2: Extensions

Note: The latter standard is required for the *label* and *association* boxes, specified by JPX format (Annex M of Part 2 – M.11.11 and M.11.13 resp.), which are required by GMLJP2.

3.1.2 Geospatial metadata

- ISO 19115-1 – Geographic information: Metadata - Part 1: Fundamentals
- ISO 19115-2:2009 – Geographic information - Metadata - Part 2: Extensions for acquisition and processing
- ISO/TS 19139:2007 Geographic information -- Metadata - XML schema implementation
- ISO/TS 19139-2:2012 Geographic information - Metadata - XML schema implementation -- Part 2: Extensions for imagery and gridded data
- ISO/TS 19115-3:2016 Geographic information - Metadata - Part 3: XML schema implementation for fundamental concepts

3.2 OGC standards

- OGC 08-085r8, GML in JPEG 2000 (GMLJP2) Encoding Standard v2.1 (2018)
- OGC 07-036, Geography Markup Language (GML) Encoding Standard v3.2.1 (2007)
- OGC 12-000, SensorML: Model and XML Encoding Standard v2.0 (2014)

- OGC 09-146r2, Coverage Implementation Schema v1.0.1 (2012) renamed from OGC GML Application Schema - Coverages

3.3 DGIWG standards

- **The ARC System**

  The ARC System as defined in DIGEST Support Document 3 on http://www.dgiwg.org/DGIWG_Geodetic_Codes/

- **DGIWG 114, DGIWG Metadata Foundation (DMF)**

  DGIWG Metadata Foundation, version 2, 12 July 2017

  Geospatial metadata profile for the military community

  Note: DGIWG Metadata Foundation, version 1.0.1, 21 November 2014, is also accepted (for backward compatibility with version 2.0 of this profile).

3.4 National standards and specifications

- **NGA.STND.0036_1.0.0_WGS84**

  US Department of Defense, World Geodetic System 1984 – Its Definition and Relationships with Local Geodetic Systems, 8 July 2014.

  http://earth-info.nga.mil/GandG/publications/NGA_STND_0036_1_0_0_WGS84/NGA.STND.0036_1.0.0_WGS84.pdf

- **NGA.STND.0037_2.0.0_GRIDS**

  Universal grids and grid reference systems, 28 February 2014.

  http://earth-info.nga.mil/GandG/publications/NGA_STND_0037_2_0_0_GRIDS/NGA.STND.0037_2.0.0_GRIDS.pdf

- **NGA.SIG.0012_2.0.0_UTMUPS**

  The Universal Grids and the Transverse Mercator and Polar Stereographic Map Projections, 25 March 2014.

  http://earth-info.nga.mil/GandG/publications/NGA_SIG_0012_2_0_0_UTMUPS/NGA.SIG.0012_2.0.0_UTMUPS.pdf

4 Terminology

4.1 Terms and definitions

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

4.1.1 Bit-rate <JPEG 2000>

a measure of the compression, providing the number of bits per sample

NOTE 1: In literature about JPEG 2000, bit-rate is abbreviated *bps (bits per sample)*[1]. Bps is given by:

Bps = c / (N1 x N2) where c = codestream length, N1 = image height in pixels, N2 = image width in pixels.

NOTE 2: This measure doesn't take the original image's number of bits per sample into account. This means that it is impossible to deduce compression ratio from the bit-rate. But the bit-rate gives a measure for compression performance.

NOTE 3: Typical bit-rates at different quality levels for JPEG 2000.

Table 4.1 – Typical bit-rates at different quality levels

Compression type	Quality	Bit rate (bits per sample)
Lossless		B – 3 bps
Lossy	High	1 bps
	Moderate	0.5 bps
	Usable	0.25 bps

In table 4.1, B represents the bitrate in the uncompressed original. The table in itself is from *JPEG 2000 – Image Compression Fundamentals, Standards and Practice, David S. Taubman, Michael W. Marcellin*. The table gives only rough estimates of the quality. The assumption is that the image is viewed on a 90 ppi (pixels per inch) computer monitor. A higher resolution on the display device, would allow for a higher compression ratio, i.e. lower bit-rate, since it would be more difficult for the eye to discern the details.

[SOURCE: Taubman and Marcellin, 2002, 8, modified]

4.1.2 Channel

one logical component of the image

NOTE: A channel may be a direct representation of one component from the codestream, or may be generated by the application of a palette to a component from the codestream.

[SOURCE: ISO/IEC 15444-1, modified]

4.1.3 Code-block <JPEG 2000>

a rectangular grouping of coefficients from the same sub-band of a tile-component

[SOURCE: ISO/IEC 15444-1]

4.1.4 Codestream <JPEG 2000>

a collection of one or more bit streams, and the main header, tile-part headers, and the EOC required for their decoding and expansion into image data

Note: This is the image data in a compressed form with all of the signaling needed to decode.

[SOURCE: ISO/IEC 15444-1, modified]

[1] *Bps* is a common abbreviation for *bits per second*, which is also a bit-rate. The context (here JPEG 2000) gives the correct meaning.

4.1.5 Compression ratio <JPEG 2000>

a measure of the compression, expressed as the ratio between the uncompressed file and the compressed file

NOTE: If *N1 = image height in pixels, N2 = image width in pixels, B = bits per sample (in the uncompressed image)* and *c = codestream length*, the compression ratio is given by:

compression ratio = *(N1 x N2 x B) / c*

[SOURCE: Taubman and Marcellin, 2002, 8, modifed]

4.1.6 Decomposition levels <JPEG 2000>

a collection of wavelet sub-bands where each coefficient has the same spatial impact or span with respect to the source component samples

NOTE: These include the HL (horizontally high pass and vertically low pass), LH (horizontally low pass and vertically high pass), and HH (horizontally high pass and vertically high pass) sub-bands of the same two dimensional sub-band decomposition. For the last decomposition level the LL (horizontally low pass and vertically low pass) sub-band is also included.

[SOURCE: ISO/IEC 15444-1, modified]

4.1.7 Discrete Wavelet Transform <JPEG 2000>

A transformation that iteratively transforms one signal into two or more filtered and decimated signals corresponding to different frequency bands.

Note: This transformation operates on spatially discrete samples.

[SOURCE: ISO/IEC 15444-1, modified]

4.1.8 ICC profile

defines the data necessary to describe the colour characteristics used to input, display, or output images, and an associated file format for the exchange of this data

[SOURCE: ICC Profile Format Specification Version 3.2, © International Color Consortium]

NOTE 1: An ICC profile is a standard representation of the transformation required to convert one colourspace into another colourspace.

NOTE 2: The JP2 file format also provides for the specification of the colourspace of an image by embedding an ICC profile in the file. That profile shall be of either the Monochrome or Three-Channel Matrix-Based class of input profiles as defined by the ICC Profile Format Specification, version 2.2.0. This allows for the specification of a wide range of greyscale and RGB class colourspaces, as well as a few other spaces that can be represented by those two profiles classes.

[ISO/IEC 15444-1]

4.1.9 JP2

JPEG 2000 file format for storing application specific data along with compressed image codestream and standard metadata as defined in ISO/IEC 15444-1, Part1

[Derived from ISO/IEC 15444-1, Part 1, Annex I]

NOTE: JP2 files can be stored using lossy or lossless compression.

4.1.10 JPEG 2000

image compression standard and coding system created by the Joint Photo Graphic Experts Group named after the year 2000

[Derived from ISO/IEC 15444-1]

NOTE 1: The JPEG 2000 standard is a multi-part standard that was created and published over the course of multiple years, circa 2000.

NOTE 2: JPEG 2000 compression uses Discrete Wavelet Transform (DWT).

4.1.11 JPX

JPEG 2000 file format for extended metadata definition and syntax that is linked with the primary data (the image)

[Derived from ISO/IEC 15444-2, Part 2, Annex M]

4.1.12 Packet <JPEG 2000>

a part of the bit stream comprising a packet header and the compressed image data from one layer of one precinct of one resolution level of one tile-component.

[SOURCE: ISO/IEC 15444-1]

4.1.13 Precinct <JPEG 2000>

a rectangular region of a transformed tile-component within each resolution level used for limiting the size of packets

[SOURCE: ISO/IEC 15444-1]

4.1.14 Rate Allocation <JPEG 2000>

individual compression for a specific code-block in order to achieve a particular bit-rate

[SOURCE: Taubman and Marcellin, 2002, 209]

4.1.15 Rectified grid

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

[SOURCE: ISO 19123]

4.1.16 Referenceable grid

a 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

[SOURCE: ISO 19123]

4.1.17 Resolution level <JPEG 2000>

equivalent to the decomposition level with one exception: the LL (horizontally low pass and vertically low pass) sub-band is also a separate resolution level

[SOURCE: ISO/IEC 15444-1]

NOTE: The spatial mapping of samples to a physical space. In ISO 15444-1, the decomposition levels of the discrete wavelet transform relate to each other with relative resolutions differing by powers of two.

4.1.18 Tile

a rectangular array of points on the reference grid, registered with and offset from the reference grid origin and defined by a width and height

NOTE: The tiles which overlap are used to define tile-components.

[SOURCE: ISO/IEC 15444-1, modified]

4.1.19 Wavelet Transform

a mathematic tool for extracting information from different kinds of data that can be represented as continuous signals

NOTE: The signal is projected on a number of frequency bands. In the raster data context, a frequency could be measured as *lines per mm* (or any other length unit). The original signal can be restored by integration over the frequency components.

[SOURCE: Taubman and Marcellin, 2002, 247, modified]

4.2 Abbreviated terms

bps	Bits per sample
COC	Coding style component
COD	Coding style default
CRS	Coordinate reference system
CSS2	Cascading Style Sheets, level 2
DCT	Discrete Cosine Transform
DMF	DGIWG Metadata Foundation
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimal Truncation
EOC	End Of Codestream
GML	Geography Markup Language
GMLJP2	Geography Markup Language in JPEG 2000
ISO	International Organization for Standardization
JPEG	Joint Photographic Experts Group
OGC	Open Geospatial Consortium
PCRD	Post Compression Rate Distortion
SML	Sensor Model Language
SVG	Scalable Vector Graphics

5 Overview of the JPEG 2000 standard

(Informative)

JPEG 2000 is a raster data compression method which uses DWT. The compression method is best suited for continuous raster data. The compression can be either lossy or lossless, but lossless compression is the less efficient method.

5.1 JPEG 2000 Part 1

JPEG 2000 Part 1 (Core Coding System) is described in ISO/IEC 15444-1. The JPEG Committee has striven to ensure that implementations of Part 1 can be royalty and licence-fee free.

5.2 JPEG 2000 Part 2

ISO/IEC 15444-2 (Part 2) contains extensions to the JPEG 2000 standard. This part is described in ISO/IEC 15444-2. GMLJP2 uses a few extensions included in this part. License and royalties may be required for use of some technologies described in Part 2. The DGIWG profile strives for using extensions that do not require royalty or license-fees. Annex G lists the extensions that are used by the DGIWG profile.

5.3 Codestream

One important property in the JPEG 2000 codestream, which is the compressed raster data, is the possibility to locate and extract data without fully decoding the codestream. This allows extraction of data from the compressed codestream to form a reconstructed image with lower resolution, lower bitrate or a region of the image. A codestream could therefore be efficiently used by even small image devices.

5.3.1 Compression of the codestream

The codestream can be compressed by an arbitrary compression ratio, and even lossless. When encoding elevation data, it is critical to preserve a high precision in the data. Therefore it is recommended to use lossless compression or a low compression ratio for elevation data so that a high precision in the data is maintained.

When encoding imagery, it may not be critical to preserve a high precision in order to extract useful information from the image. Therefore a higher compression ratio may be used in that case.

5.3.2 Partition of the codestream

JPEG 2000 codestreams can be partitioned in several different ways.

The codestream is divided in a number of steps:

a) Decorrelated into components if it is multi-band data. Otherwise it is only one component.

b) Components are divided into tiles.

c) The components of the tiles are wavelet transformed with N_L decomposition levels into sub-bands. Each decomposition level gives a low-pass band representing a low resolution version, and a high-pass band representing the residual between the low resolution version and the original version. By decomposing the low resolution band from the first decomposition, levels with even lower resolutions are given. Thus, each decomposition adds one more resolution level without increasing the amount of information. As a result, each tile is available at $N_L + 1$ distinct resolutions.

d) The resolution levels are divided further into precincts.

e) The sub-bands from step 3 are divided into code-blocks, which are coded independently of each other.

The different resolutions are denoted as *sub band* LL_0, which is the highest spatial frequency, and thus, the highest resolution, continuing this way through the resolution levels LL_1, and onwards, to LL_D, which is the lowest resolution level.

5.3.2.1 Components

A component covers the whole image area. But the component contains only one colour. Each component is separated from the image before any other partitioning.

5.3.2.2 Tiles

Tiles are relatively large partitions of the image. The tiles can be compressed with individual parameters for each tile. It is possible to have an image where some of the tiles are not compressed at all, or some tiles are compressed with relatively low degradation of the image. A lot of image formats use tiling in order to reduce memory usage by only handling small parts of the image. It is possible to do that with the tile mechanism in JPEG 2000, but it isn't necessary, since the organization of the image in precincts enables direct access to parts of the image.

5.3.2.3 Sub-bands

A tile is decomposed into the sub-bands *HH* (horizontally and vertically high pass), *HL* (horizontally high pass and vertically low pass), *LH* and *LL*.

5.3.2.4 Decomposition levels

The LL sub-band from the low pass filtering, gives a resolution level with half the resolution of the image that was filtered. The other sub-bands are discarded.

5.3.2.5 Code-blocks

Code-blocks are coded independently of each other. Common sizes are 32x32 and 64x64 pixels. The fact that the code-blocks are independently encoded, gives some advantages, among them are the fact that many code-blocks can be processed simultaneously. That means JPEG 2000 processing scales well when using multiple CPUs or multi-core CPUs. Since it today seems like the most effective way to increase performance, is to use more cores in the CPUs, this is an important property in JPEG 2000. The code-blocks are also used for *Rate-Allocation*. *Rate-allocation* chooses the contributions of each code-block such that an aimed bit-rate is achieved. Code-blocks are essential for the EBCOT paradigm.

5.3.2.6 Precincts

Precincts organize the compressed data in the codestream. Precincts do not affect the coding of the data. Therefore, the precincts enable direct access to parts of the image. The use of precincts reduces memory use for decompression. Precinct dimensions must always be a power of two. One precinct contributes one *packet* for each *quality layer*.

5.3.2.7 Packets

The compressed data in the codestream is organized in packets. Packets are the smallest partitions of the codestream. Each precinct has one packet for every quality layer. A packet contains contributions up to that quality layer from each code-block within that precinct. Note that the contributions to any particular layer might be empty.

5.3.3 Codestream information

The codestream itself contains information for decoding the codestream.

5.3.3.1 Headers

Headers are collections of markers and marker segments, described in paragraphs. The headers are either *main header*, which has a scope encompassing the complete codestream or *tile-part header*, which has a scope encompassing a separate tile, thereby giving the option of having different compression parameters for individual tiles.

5.3.3.2 Markers and marker segments

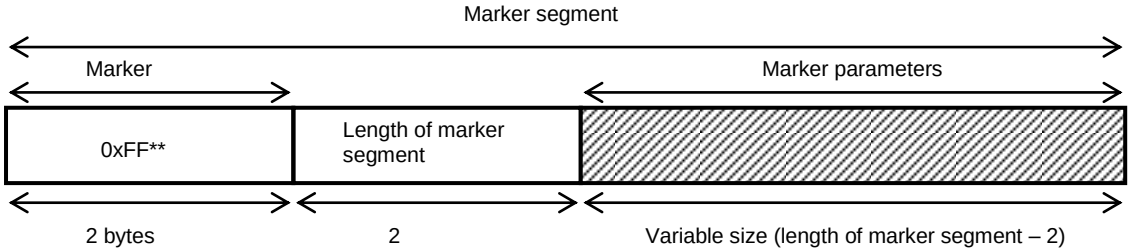

Figure 1 – Marker segment structure

The codestream contains markers for locating information in it. A marker consists of a two-byte word, 0xFF<*marker code*>. The first byte is always *0xFF*. The second byte is the marker code. A marker segment includes a marker and parameters associated with that marker. If there are associated parameters, the marker must be followed by a two-byte word that gives the size of the marker segment, including the size, but excluding the marker. A marker without any parameters, doesn't have a *marker length* either, and is thus only two bytes.

5.3.3.3 COD - Coding style default and COC – Coding style component

The COD marker segment contains information about the encoding of an image. It is possible to override the image COD with a COD for a specific tile or a COC for a specific component or tile.

The COD describes:

Table 5.1 – Items in a Coding style default structure

Items
Number of decomposition levels
Progression order
Number of quality layers
Codeblock size
Codeblock style
Transform
Multiple component transform
Packet partition size

The COC describes:

Table 5.2 – Items in a Coding style component structure

Items
Number of decomposition levels
Codeblock size
Codeblock style
Transform
Packet partition size

If more than one of the markers is used, it's always the most local of the blocks that are used. The precedence is as:

Tile COC > tile COD > main COC > main COD

5.3.4 Error resilience

Error resilience is achieved through the use of two optional markers in the codestream; SOP (Start of packet) and EPH (End of packet header). The idea is that if a codestream is interrupted, the codestream can be resynchronized with the help of the SOP and EPH markers, even though it is impossible to reconstruct everything. The use of SOP or EPH markers is indicated by a COD (codestream default) marker, which can appear both in the main header or the tile header.

5.3.4.1 SOP – Start of packet

The SOP marker carries a two byte counter which starts from 0 in every tile. If it rolls over, i.e. gets larger than 65535, it will start over from 0 again. As the name implies, there should be one marker for each packet. But the codestream is allowed to skip SOP markers for some packets. If some SOP markers are omitted, the next SOP marker must have its counter incremented as if the omitted SOP markers actually did exist for each packet. If a COD indicates the use of SOP markers, the markers won't necessarily appear within the scope of the COD. SOP markers are not allowed when the COD indicates that they are not used.

The use of SOP markers may require as many as six bytes for every packet, but the markers are not necessary for every packet.

5.3.4.2 Payload

The error resilience markers may use as many as eight bytes for each packet. Packet sizes vary. Packet sizes can be expected to be as small as about 100 bytes but most often they will be several KB in size. Therefore it is reasonable to assume that the payload for the error resilience markers will be negligible.

5.3.5 The EBCOT algorithm

The EBCOT (Embedded Block Coding with Optimized Truncation) algorithm is used for fast access of arbitrary regions with arbitrary resolution. The EBCOT algorithm uses the code-blocks to access regions of the image. There are also quality layers, to which the code-blocks contribute up to a specific truncation point.

5.3.5.1 PCRD-opt

Rate-allocation is done with Post-Compression Rate Distortion optimization, which is an algorithm for assigning truncation points in a code-block minimizing the distortion for a specific bit-rate. The truncation points are used for *quality layers*.

5.3.5.2 Quality layers

A *quality layer is* an abstract form of dividing the image. A *quality layer* is made up of the contributions from all code-blocks up to their respective truncation points for each *quality layer*. The PCRD optimization means that the quality layers differ in bit-rates, as opposed to *decomposition levels*, which differs in resolution. Increasing the number of quality layers, means increasing the overhead needed for the truncation points. The overhead consists of length tags inserted into the code-blocks in order to identify its contribution to a *quality layer*.

5.3.6 Progression order

The progression order determines the order of the packets in a specific tile.

Table 5.3 – List of progression orders

Progression order	Description
LRCP	Layer-Resolution-Component-Position progression. Progressive by quality. Quality increases for each layer.
RLCP	Resolution-Layer-Component-Position progression. Progressive by resolution. The outermost loop gets a new resolution level for each stage in the loop.
RPCL	Resolution-Position-Component-Layer progression. Progressive by resolution.
PCRL	Position-Component-Resolution-Layer progression. Progressive by position.
CPRL	Component-Position-Resolution-Layer progression. Progressive by component.

5.3.7 JPEG 2000 Profiles

A JPEG 2000 reader isn't required to be able to recover all of the information in a codestream. But there is a need to ensure that the reader can recover at least some of the information in the codestream. Therefore, the JPEG 2000 standard includes a number of profiles with different restrictions of the parameters that applies to the codestream.

JPEG 2000 codestreams are allowed to follow Profiles 0, 1 and 2 according to the DGIWG profile.

5.3.7.1 Profile-0

Profile-0 is developed for low complexity applications.

- Tiles must be squares with maximum size 128x128 pixels. Only exception is when only one tile for the whole image is used.
- Code-block size is restricted to 32 or 64 in both directions. The value must be same in both directions.
- LL_D must not exceed a spatial resolution above $w = 128$ and $h = 128$. Same restriction applies to *Profile-1.*

Profile-0 excludes features that are necessary for error detection. The restrictions also limit the possibility to optimize the image for rapid access to different image elements or for displaying the image on different devices.

5.3.7.2 Profile-1

Profile-1 is meant to be both easy to implement and allowing high performance at the same time.

- Tiles must be squares with a maximum size 1024x1024. The only exception is when the entire image is composed of only one tile.
- Code-block size is restricted to a maximum value of 64 in both directions.
- LL_D must not exceed a spatial resolution above *w = 128* and *h = 128*. Same restriction applies to *Profile-0.*

Software implementations are strongly encouraged to support at least Profile-1. Therefore, interoperability is maximized by conforming to Profile-1.

5.3.7.3 Profile-2

No restrictions, i.e. all elements defined by ISO/IEC 15444-1 can be used. If there is no indication of profile, the codestream conforms to Profile-2.

5.3.7.4 Implications of the LL_D restriction

The *DWT* is always performed on a tile. Therefore, LL_0, the maximum resolution level, will never exceed the tile size. If the image is divided into only one tile, that is the image size. In that case, the minimum resolution level, LL_D, will be the size of the smallest *"pyramid layer"*. If the image is divided into several tiles, LL_D will be the size of the smallest *"pyramid layer"* of a tile. The whole image will be larger even at that resolution level. If the image conforms to *Profile-1*, the image could be divided into 1024x1024 tiles. In that case, all images would be required to have 5 decomposition levels, since the *h <= 128* restriction would mean that LL_D *== 128x128* isn't sufficiently low (the fourth level). But if the image is divided into only one tile, which is allowed by all profiles, the number of required decomposition levels will vary depending on the image's total size.

5.3.7.5 Recommendation to use Profile-1

Since there is a recommendation that software implementations are able to interpret Profile-1, it is also recommended that a codestream conforms to Profile-1. But certain applications can benefit from parameters outside the restrictions imposed by Profile-1. Therefore a codestream doesn't have to conform to Profile-1.

5.3.8 Multiple codestreams

JP2 format allows multiple codestreams. The use of multiple codestreams requires a JPX capable reader. A DGIWG GMLJP2 compliant reader shall handle multiple codestreams. A DGIWG GMLJP2 compliant writer shall be able to handle multiple codestreams.

5.3.8.1 Compositing layer

JPEG 2000 Part 2 introduces *compositing layers*. A compositing layer is constructed from a number of codestreams in a way defined in the JPEG 2000 file. Components in the codestreams are mapped to specific components in the compositing layer, which is viewed as one separate image. If this option is used, a reader which is compliant with only Part 1 will only be able to read the first codestream.

5.3.9 JPIP

JPIP (ISO/IEC 15444-9) is a protocol for streaming JPEG 2000 data over a network. This protocol allows communication between the client and the server when only small portions of the codestream are transferred. If the client requests a large overview, the server will transfer a low resolution version adapted to the requested resolution. If the client requests a particular area of the image, the server will only transfer that area. Therefore, even gigapixel sized images will be usable over a network even with a relatively low bandwidth. JPIP is also used for progressively display an image while waiting for it to be completely downloaded. JPIP relies on the partition of the image into low resolution decomposition levels and precincts. JPIP can be used over IP, TCP, UDP and HTTP.

5.3.9.1 Data-bin

JPIP organizes the information in *data-bins*. Data-bins can contain precincts, tiles or metadata. It is possible to request only metadata[2]. The data-bins that contain image elements, such as precincts and tiles, contain parts of the codestream box, and not the complete codestream box. The data-bins that contain metadata can contain any collection of boxes. A JPEG 2000 box is the smallest element a metadata-bin can contain. That means it is possible to transmit multiple JPEG 2000 boxes over multiple independent channels, i.e. one JPEG 2000 box per channel. This allows out of band transmission to speed up metadata throughput.

5.4 JPEG 2000 Box

The information in a JPEG 2000 file is organized in boxes. A box can contain the codestream or information about the file.

5.4.1 Box structure

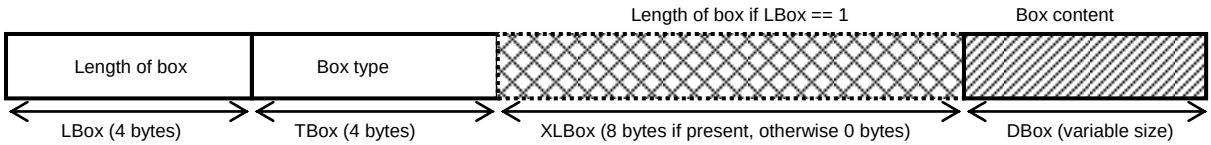

Figure 2 – JPEG 2000 box structure

The box consists of:

LBox: 4-byte unsigned integer. The length of the box. This field and the box type field are included. Possible values that cannot be interpreted as box sizes, since the values are too small:

- 0: box length is unknown, which means that the box contains everything up to the end of file. A typical example of this is the *'jp2c'*-box in many files. If a box contained within a superbox, has this value, when the superbox size is implicitly unknown, and should therefore also have the value 0.
- 1: the box real length is in the *XLBox* field described below.

TBox: Actually defined as a 4-byte unsigned integer, but in practice, they are always referred to by a character string translation of the integer value. A space character is often represented as "\040" (the octal representation of its ordinal number).

XLBox: 8-byte unsigned integer. Extended field for the length. The sizes of the LBox, TBox and XLBox, are included. This field is useful if the length is larger than 4GB.

DBox: Variable length and format. This field contains the actual information in the box.

5.4.2 Superbox

The *superbox* is a box that contains other boxes. This is a mechanism that is used extensively in the *GMLJP2* standard in order to organize different GML instances.

5.5 Opacity

Opacity can be used in different ways.

[2] Here, "metadata" refers to any collection of "boxes" from a JPEG 2000 family file (ISO/IEC 15444-9 A.3.6.1).

5.5.1 Opacity channels (JPEG 2000 Part 1)

The codestream contains an auxiliary channel with opacity information. This is the only Part 1 compatible option.

5.5.2 Opacity in a separate codestream (JPEG 2000 Part 2)

Since the introduction of *compositing layers*, it is possible to store opacity information in a separate codestream. The mechanism is similar to the Part 1 case with opacity channels.

5.5.3 Opacity box (JPEG 2000 Part 2)

The opacity box gives option for specifying that the last channel in a compositing layer is an opacity channel, or giving one or more chroma-key values. A chroma-key value specifies that a certain colour should be considered as transparent. A chroma-key value is useful for GMLJP2 files that have been transformed between different reference systems. Such files usually have large one-colored regions at the edges. This method of opacity, works best with the lossless mode of JPEG 2000, because the compression artifacts would be visible in the regions that should be transparent. The Opacity box is intended for use with lossless coding.

The opacity box requires the use of *compositing layers*.

5.6 Integer values constraint of raster and gridded data

By default, JPEG 2000 allows only integer raster data and provides an extension mechanism to use other types of pixel values that this profile doesn’t use.

In order to store elevation data with a higher precision than one meter, it is necessary to perform an upscaling conversion of elevation values so that they all can be given as integers.

6 GML objects

(Informative)

6.1 Introduction

GMLJP2 allows different kinds of data to be stored as GMLJP2 instances in a JPEG 2000 file (in an XML Box under an ASOC (Association) super-box, as specified by OGC GMLJP2 standard):

- GMLJP2CoverageCollection structure, which is the root element containing the various GMLJP2 objects
- Optional GMLJP2 Coverage collection metadata
- Grid Coverage member(s), allowing the georeference of the image or grid contained in the JPEG 2000 codestream
- Annotations (not covered by this document)
- Annotation styling (not covered by this document).

Each GMLJP2 instance is therefore documented in a GML document, sometimes identified as such (GML document) in this document.

6.2 Coverage

This profile allows for two types of Coverage, either RectifiedGridCoverage when the image or grid is Rectified, or ReferenceableGridCoverage when the image or grid is referenceable by means of any of the mechanisms indicated below.

A GMLCOV *RectifiedGridCoverage* is used to georeference the image or more generally Rectified Grid coverage with the help of a *RectifiedGrid* (D.2.2.1.2) which handles the georeference information with the attribute *srsName* (Table D.5) with the reference systems listed in D.2.4.

A GMLCOV *ReferenceableGridCoverage* is used for referenceable image or grid coverage, with the help of either of the following GMLCOVRGRID mechanisms / elements, and must provide the external reference system of the image with the attribute srsName (Table D.5) of any following element deriving from gmlcov:AbstractReferenceableGrid in the domainSet:

- ReferenceableGridBySensorModel (cf. D.2.3.1.2)
- ReferenceableGridByTransformation (cf. D.2.3.1.3)
- ReferenceableGridByVectors (cf. D.2.3.1.4)
- ReferenceableGridByArrays (cf. D.2.3.1.5)

In addition, the coverage describes the value range of the image or grid, which could include elevation data.

Each RectifiedGridCoverage or ReferenceableGridCoverage is associated with a codestream. The RectifiedGridCoverage or ReferenceableGridCoverage are contained in a GMLJP2CoverageCollection.

6.3 Coordinate reference system

The GML coverage refers to a coordinate reference systems (CRS) used for georeferencing the image or more generally the grid coverage. It could be a predefined code for the CRS, like the EPSG codes. But it is also possible to define a CRS in a *CRSDictionary.gml* box. This box could be named *CRSDictionary.gml*, or any other unique name. This profile shall

use a limited set of EPSG codes defined in Table 6.1, or other CRS specified by DGIWG Geodetic Codes and Parameters Registry, that must be identified by their EPSG Geodetic Parameter registry identifier.

Table 6.1 – EPSG codes

EPSG code	Reference system
4326	GCS_WGS84
326zz and 327zz	UTM for each zone in the northern hemisphere respectively the southern hemisphere
7030	WGS84 ellipsoid as a vertical reference
5773	Earth Gravity Model 1996
3855	Earth Gravity Model 2008
3395	WGS 84 / World Mercator Note: should only be used for raster maps. This reference system is also used in GeoPackage

6.4 Axis order

There are two possible axis orders that are in use, *left hand order* and *right hand order*. The terms for these axis orders, are understood by the figure below:

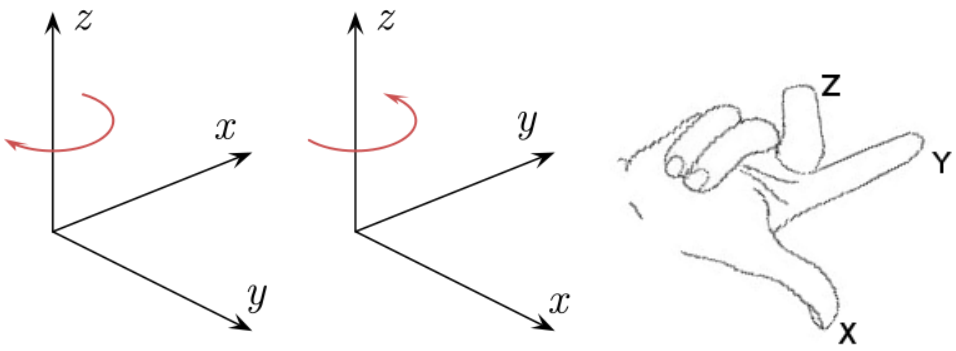

Figure 3 – Left and right hand order

The thumb denotes the first axis (typically X), the index finger denotes the second axis (Y) and the middle finger denotes the last axis (Z). If the right hand fingers are used, the first axis will be the easting axis. If the left hand fingers are used, the first axis will be the northing axis.

Mathematicians traditionally use right hand order. In the geo community, both right hand order and left hand order are used. WGS84 LL is defined as a left hand order system, while UTM is a right hand order system. Both traditions are strong.

6.5 Annotations

Annotations are not handled by this profile. Annotations should be handled by an extension to the profile. Therefore the annotation mechanism isn't described by this document. Annotations may include both text and vector graphics, which can be used to point out different kinds of regions, or extended annotations with embedded images and video.

6.6 Metadata

Any GML element, including the coverage, can include metadata in an arbitrary schema, including ISO/TS 19139, ISO/TS 19139-2 and GML metadata application schema. If a GML information in the GMLJP2 instance (GML document), like geometry or radiometric information in a GML coverage, contradicts the included metadata, this GML information will take precedence over the metadata schema. The data provider should ensure the metadata is consistent with the information provided in the GML.

The metadata should be conformant with DMF in its version 2, or DMF version 1.0.1 (for backward compatibility).

National metadata profiles, based on ISO/TS 19139, ISO/TS 19139-2 or ISO/TS 19115-3 and that can be mapped to DMF are also acceptable.

The metadata document may be either:

- included in the GML document (in the same XML box, labelled as gml.data)
- in an external XML file
- included in another XML box encapsulated under a 2 levels structure of JPEG 2000 asoc boxes, which should be labelled appropriately as metadata. Recommended label for the root (or 1st) level is “iso19115.metadata”. 2nd level will depend on data or product specification or data producer.

Note: In case of collection of datasets in the same GMLJP2 file, resulting in multiple codestreams addressed by multiple coverage instances in the GML, and multiple metadata instances (usually one for each dataset / codestream), the order of the codestreams, corresponding GML coverages and corresponding XML metadata shall be the same.

7 Embedding GML in JP2 files

OGC’s GMLJP2 standard, propose that GML code for georeferencing a JPEG 2000 file, should be stored in xml boxes. There is also a container entity, called *association box*, and a *label box*, which provides the means of using a hierarchical structure within a superbox. This structure of XML boxes with GML code is described in Annex D.1. The box structure supports referring to different boxes in the GML structure, as well as different codestreams outside the GML structure, with the help of a *GMLJP2 URI*, described in Annex D.1.3.

7.1 JPX file format

GMLJP2 uses a number of boxes that are defined in JPEG 2000 Part 2 (JPX specification). These boxes are described in Annex E.

7.2 Compatibility list in File type box

OGC GMLJP2 (08-085r8) recommends that the files are written as JP2 compatible, i.e., the file should be compatible with Part 1, and signaling this by including the string “jp2\040” in the *compatibility list* in the *File type box*. The DGIWG profile has one exception from this rule, which the paragraph about the opacity information mentions (next paragraph, 7.2.1).

7.2.1 If opacity of Part 2 type is used

Opacity information according to one of the Part 2 options (Opacity in a separate codestream (JPEG 2000 Part 2) and Opacity box (JPEG 2000 Part 2)), requires the use of *compositing layers*. A JPEG 2000 Part 1 reader won’t be able to render the codestreams correctly. Therefore the compatibility list should not include the string “jp2\040”, but make use of the string “jpx\040”.

7.2.2 Profile

The compatibility list might also specify which profile (5.3.7) the first codestream should conform to. “J2P0” requires that the first codestream conforms to Profile-0, and “J2P1” requires that the first codestream conforms to Profile-1. If no profile is indicated, the codestream is allowed to follow Profile-2 (no restrictions). The Profile values are set in the Rsiz field in the image and tile size marker of the codestream.

7.3 Brand field in File type box

The brand field shall always be “jpx\040” for JPX files.

7.4 Reader requirement box

The presence of GML data should be signaled with a standard flag with the value 67 in a reader requirement box (C.1.3).

Annex C.1.3 provides guidance on the use of Reader requirement Box.

7.5 File suffix

This profile uses two different file suffixes:

Table 7.1 – File suffixes

File suffix[3]	Explanation
.jp2	JPEG 2000 file that follows ISO/IEC 15444-1 (I.2.1 in that standard). A JPX file that is compatible with ISO/IEC 15444-1 is allowed to use this file suffix (ISO/IEC 15444-2 M.2.1). Generally, a file should use this file suffix, but they are allowed to use “.jpf” as well.
.jpf	JPEG 2000 file that follows ISO/IEC 15444-2 (M.2.1 in that standard). A file that uses one of the Part 2 opacity options (7.2.1) shall use “.jpf”.

ISO/IEC 15444-2 M.2.1 prefer the file suffix “.jpf” for Part 2 files. But the Part 1 file suffix (“.jp2”) is already established for GMLJP2 files. Files that follow this profile can use both “.jp2” and “.jpf”, with the exception of files that uses one of the Part 2 opacity options (7.2.1). These files are required to use “.jpf”, since a Part 1 compliant reader can’t be expected to understand the opacity information. A file that uses the Part 1 opacity option (opacity channels) is allowed to use “.jp2”.

[3] Note: Data consumers can only be sure if the file is a true GMLJP2 file by parsing JP2 boxes.

Annex A: Abstract Test Suite
(normative)

A.1 Introduction

This annex provides an abstract test suite for GMLJP2 data conforming to this profile. Some of these tests require that the tester uses software that gives a good overview of the content in a JPEG 2000 file. Annex H refer to software that can be used for the abstract test suite.

A.1.1 Test classes

The test classes refer to the conformance classes that are described in Annex B.

A.2 OGC GMLJP2 tests

This profile is required to pass all OGC tests that are listed in OGC GML in JPEG 2000 (GMLJP2) (08-085r8).

A.3 DGIWG GMLJP2 profile tests

The DGIWG tests are specific for the DGIWG GMLJP2 profile.

A.3.1 Class B tests

Class B is described in B.3.1.

A.3.1.1 *The RectifiedGrid contains all required content*

a) Test purpose: Verify that the RectifiedGrid contains all mandatory attributes and elements.

b) Test method: Inspect the GML reference. Inspect the *RectifiedGrid* at following XPath: ***/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:domainSet/gml:RectifiedGrid***
Following elements and attributes should exist:

Name	Type
limits	XML element
origin	XML element
offsetVector	XML element. Two elements are required.
axisLabels	XML element. List of names of axis (e.g. Lat Long)
axisName (deprecated in favor of axisLabels)	*XML element. Two elements are required. (may be used if axisLabels is not used)*
srsName	XML attribute

c) Reference: OGC 08-085r8 – OGC GML in JPEG 2000 (GMLJP2) Encoding Standard and this profile (DGIWG GMLJP2: D.2.2.1.3).

d) Test type: Basic.

A.3.1.2 *The external CRS for ReferenceableGridCoverage aligns with specified CRS*

a) Test purpose: Verify that the srsName attribute of any of element deriving from gmlcov:AbstractReferenceableGrid (ReferenceableGridBySensorModel, ReferenceableGridByTransformation, ReferenceableGridByVectors or ReferenceableGridByArrays) is provided and its value is consistent with the CRS defined in table D.9).

b) Test method: Inspect the GML document. Inspect the *domainSet of the ReferenceableGridCoverage* at following XPath: ***/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet***
One of the following elements (ReferenceableGridBySensorModel, ReferenceableGridByTransformation, ReferenceableGridByVectors or ReferenceableGridByArrays) should exist and its srsName attribute should have a value from Table D.9.

c) Reference: OGC 08-085r8 – OGC GML in JPEG 2000 (GMLJP2) Encoding Standard and this profile (DGIWG GMLJP2: D.2.3).

d) Test type: Basic.

A.3.1.3 *Coordinate lists in GML RectifiedGrid have sufficient number of coordinates*

a) Test purpose: Verify that the GML code has sufficient number of coordinates at certain places in the GML document for Rectified Grid. GML uses the *XML Schema Language* type *list* for allowing more than one coordinate in one element. The *XML Schema Language* doesn't allow a schema to put any restrictions on the number of coordinates when using the *list* type. This means an *XML schema validation* will not catch that error. Therefore the number of coordinates has to be verified manually in the GML code. Typically, there should be at least two coordinates in one element, but the *gml:Point* and the *gml:offsetVector* could have three coordinates in the case of 3D data.

b) Test method: Check that there are at least two coordinates in the GML code at following XPaths:
/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:domainSet/gml:RectifiedGrid/gml:limits/gml:GridEnvelope/gml:low
/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:domainSet/gml:RectifiedGrid/gml:limits/gml:GridEnvelope/gml:high
/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:domainSet/gml:RectifiedGrid/gml:origin/gml:Point/gml:pos
/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:domainSet/gml:RectifiedGrid/gml:offsetVector

c) Reference: OGC 08-085r8 – OGC GML in JPEG 2000 (GMLJP2) Encoding Standard

d) Test type: Basic.

A.3.1.4 *Coordinate lists in GML ReferenceableGrid Coverage have sufficient number of coordinates*

a) Test purpose: Verify that the GML code has sufficient number of coordinates at certain places in the GML document for Referenceable Grid. GML uses the XML Schema Language type list for allowing more than one coordinate in one element. The XML Schema Language doesn't allow a schema to put any restrictions on the number of coordinates when using the list type. This means an XML schema validation will not catch that error. Therefore the number of coordinates has to be verified manually in the GML code. Typically, there should be at least two coordinates in one element, but the gml:Point and the gml:offsetVector could have three coordinates in the case of 3D data.

b) Test method: Check that there are at least two coordinates in the GML code at following XPaths:

For ReferenceableGridBySensorModel

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridBySensorModel/gml:limits/gml:GridEnvelope/gml:low

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridBySensorModel/gml:limits/gml:GridEnvelope/gml:high

For ReferenceableGridByTransformation

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByTransformation/gml:limits/gml:GridEnvelope/gml:low

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByTransformation/gml:limits/gml:GridEnvelope/gml:high

For ReferenceableGridByVectors

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByVectors/gml:limits/gml:GridEnvelope/gml:low

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByVectors/gml:limits/gml:GridEnvelope/gml:high

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByVectors/gml:origin/gml:Point/gml:pos

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByVectors/gmlcovrgrid:generalGridAxis

For ReferenceableGridByArray

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByArray/gml:limits/gml:GridEnvelope/gml:low

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByArray/gml:limits/gml:GridEnvelope/gml:high

/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet/gmlcovrgrid:ReferenceableGridByArray/gml:posList

c) Reference: OGC 08-085r8 – OGC GML in JPEG 2000 (GMLJP2) Encoding Standard

d) Test type: Basic.

A.3.1.5 *Left-hand order coordinate system is used for WGS84 LL coordinates*

a) Test purpose: This profile prescribes that the axis order for WGS84 LL should follow standard practice, i.e., use *left-hand order* axis (latitude, longitude). This test is not applicable for UTM/WGS84, which uses *right-hand order* axis.

Test method: It is not advisable to test this by inspecting the image in GIS software, because that might instead highlight the axis handling in the software. One important part of the problem is that both types of axis orders are common in the software. This profile tries to achieve unification in the handling of axis orders, which is in line with the recommendation of GML 3.2.1. Therefore inspect the coordinates directly in the GML code at following XPaths:

b) ***/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:domainSet/gml:RectifiedGrid/gml:origin/gml:Point/gml:pos***
While using WGS84-LL, the axis order shall be left-handed.
/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:domainSet/gml:RectifiedGrid/gml:offsetVector
While using WGS84-LL, the axis order shall be left-handed.

c) Reference: DGIWG GMLJP2 Profile – D.2.2.1.3 and 6.3 for an illustration

d) Test type: Basic

A.3.1.6 *Image coordinate system gives the coordinates in the correct order*

a) Test purpose: The image coordinates does not follow the same rule as the ground coordinates. The image coordinates shall always be given in following order (width, height).

Test method: Inspect the coordinates directly in the GML code at following XPath: ***/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:domainSet/gml:RectifiedGrid /gml:limits/gml:GridEnvelope/gml:high*** for Rectified grid Coverage

b) or
/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2ReferenceableGridCoverage/gml:domainSet /gml:RectifiedGrid/gml:limits/gml:GridEnvelope/gml:high sub-element for any the corresponding ReferenceableGridBySensorModel, ReferenceableGridByTransformation, ReferenceableGridByVectors or ReferenceableGridByArrays, for Referenceable grid Coverage.
The axis order shall be given as width and height in that order. This can only be inspected in a test case there the image **is not square**.

c) Reference: DGIWG GMLJP2 Profile – D.2.2.1.3 and D.2.3.1.1

d) Test type: Basic

A.3.1.7 *Security classification in metadata*

a) Test purpose: Verify that the security classification is recorded in the metadata.

b) Test method: Inspect the metadata to see if there is a security classification.
The metadata is embedded at the following XPath:
/gmljp2:GMLJP2CoverageCollection/gmljp2:featureMember/gmljp2:GMLJP2RectifiedGridCoverage/gml:metadataProperty
In the metadata, the security classification is found at following XPath:
/gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:resourceConstraints/gmd:MD_SecurityConstraints/gmd:classification

c) Reference: ISO 19115-1

d) Test type: Basic.

A.3.1.8 *Security classification in Intellectual property rights box*

a) Test purpose: If the file is classified in a security class more restricted than *unclassified*, the classification shall be recorded in an *Intellectual property rights* box in order to secure that information in case the metadata is lost in some way.

b) Test method: Inspect the *'jp2i'*-box (the *Intellectual property rights* box). The box shall contain an IPR element. The security classification shall be stored at following XPath:
/jp:IPR/jp:IPR_EXPLOITATION/jp:IPR_USE_RESTRICTION

c) Reference: DGIWG GMLJP2 Profile – C.1.5

d) Test type: Basic.

A.3.1.9 *Precinct partition size*

a) Test purpose: The precinct partition size is restricted to be a power of two.

b) Test method: Inspect the codestreams markers COD and COC. The COC information overrides the COD information. COD and COC for a tile, overrides the COD and COC for the whole codestream.

c) Reference: ISO/IEC 15444-1 B.5, A.6.1, A.6.2 and DGIWG GMLJP2 Profile: 5.3.2.6.

d) Test type: Basic.

A.3.1.10 *File suffix*

a) Test purpose: A GMLJP2 file must use a file suffix which reflects the file type that is used.

b) Test method: Inspect the compatibility list in the file type box and the filename. If the compatibility list contains ‘jp2\040’, the file suffixes “.jp2” and “.jpf” are allowed. Otherwise only “.jpf” is allowed.

c) Reference: ISO/IEC 15444-2 M.2.1 and DGIWG GMLJP2 Profile: 7.5 and C.1.2.

d) Test type: Basic.

Annex B: Conformance
(normative)

This annex describes the conformance requirements for DGIWG GMLJP2 files as well as software that read or process a DGIWG GMLJP2 file.

B.1 Purpose

The purpose of this annex is to define different sets of requirements for a reader or a processor of GMLJP2 files as well as files.

B.2 JPEG 2000 Profiles

The JPEG 2000 profiles are different sets of requirements that apply to the JPEG 2000 files.. Profile-2 means *no restrictions*, i.e. all elements defined by ISO/IEC 15444-1 can be used. If there is no indication of profile, the file conforms to Profile-2. Profile-2 is not listed in the table of the restrictions:

Table B.1 – JPEG 2000 Profiles

Restrictions	Profile-0	Profile-1
Tile size	Either smaller or equal to 128x128, or one tile for the whole image	Either smaller or equal to 1024x1024, or one tile for the whole image
Code block size	32x32 or 64x64	Smaller or equal to 64x64. Non-square codeblocks are allowed
LL_D *(lowest resolution element)*	Not larger than 128x128	Not larger than 128x128

DGIWG GMLJP2 profile only allows profiles 0, 1 and 2 that are within its scope.

B.3 Conformance classes

The conformance classes are different sets of requirements on a reader or a processor depending on the complexity of the file. The file belongs to a certain conformance class, and can thus be said to have certain requirements on the reader or processor.

B.3.1 Class B – Baseline

A Class B file is a JPEG 2000 file georeferenced with GML according to this profile.

B.3.2 No other conformance classes

In DGIWG GMLJP2 v2, *class B* is the only conformance class. The previous version had two other conformance classes in addition to *class B*. These used annotations. OGC GMLJP2 v2 uses annotations in a different way than DGIWG GMLJP2 v1. Therefore DGIWG GMLJP2 v2 doesn't use annotations. That means the other conformance classes are irrelevant for now. There are plans for an extension of the profile for including annotations.

B.4 Implementation Conformance Report

This chapter provides a report form for testing software conformance to the DGIWG GMLJP2 Profile. The following code values are used in Table B.2 below.

R = Required C = Conditional O = Optional S = Supported P = Partial support N = No support

Table B.2 – Implementation Conformance Report

Conformance test	**Produce**				**Use**				**Comment**	**Conf. class**
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
JPEG 2000 boxes										
JPEG 2000 Signature box See C.1.1	R									B
File type box ('ftyp'), see C.1.2	R									B
BR field always **'jpx\040'**	R									B
MinV field always **0**	R									B
CL field	R									B
'jp2\040' or **'jpx\040'** in the CL field	R									B
Reader requirements box ('rreq'), see C.1.3	R									B
One of the standard flags (SF^i) must have value **67**, which signals the presence of GML data.	R									B
JPEG 2000 header box ('jp2h'), see C.1.4	R				R					B

Table B.2 – Implementation Conformance Report-Continued

Conformance test	**Produce**				**Use**				**Comment**	**Conf. class**
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
Image header box ('ihdr'), see C.1.4.1	R				R					B
Height and Width	R				[4]					B
Number of components	R									B
Bits per component	R									B
Compression type always **7**	R									B
Unknown colourspace: **0** or **1**	R				R				Ignoring the feature might lead to erroneous color reproduction	B
Intellectual property rights This post signals that an Intellectual Property Rights box is present or absent: **0:** No Intellectual Property Rights box. **1:** An Intellectual Property Rights box exists.	R				R				Ignoring the feature might lead to copyright violation	B
Colour specification box ('colr'), see C.1.4.2	R				R				Ignoring the feature might lead to erroneous color reproduction	B
METH – **1** or **2**	R				R				Ignoring the feature might lead to erroneous color reproduction	B
PREC - **0**	R									B
APPROX - **0**	R									B
EnumCS – **16**, **17** or **18**	C				R				Ignoring the feature might lead to erroneous color reproduction	B

[4] Height, width, number of components and number of bits per component, are information available from the codestream, so they are not a reader requirement in the context of an *Image header box*.

Table B.2 – Implementation Conformance Report-Continued

Conformance test	**Produce**				**Use**				**Comment**	**Conf. class**
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
PROFILE	C				R				Ignoring the feature might lead to erroneous color reproduction	B
Intellectual property rights box ('jp2i'), see C.1.5	C[5]				R				Ignoring the feature might lead to copyright violation	B
IPR_USE_RESTRICTION One of following keywords must be used: ***unclassified*** ***restricted*** ***confidential*** ***secret*** ***topSecret*** ***sensitiveButUnclassified*** ***forOfficialUseOnly*** ***protected*** ***limitedDistribution***	C[6]				R				Ignoring the feature might lead to copyright violation	B
IPR_MGMT_TYPE	O				R				Ignoring the feature might lead to copyright violation	B
Codestream box ('jp2c'), see C.1.6	R				R					B
Codestream-related properties										
Multiple codestreams	R				R				Ignoring some of the codestreams means that the content might be interpreted wrong	B

[5] Required when the Image Header box signals the presence of an Intellectual Property Rights box. Additionally, the IPR box is required by this profile if the image data is classified as any of the options *restricted*, *confidential*, *secret* and *topSecret*.
[6] Required when classified as *restricted*, *confidential*, *secret* or *topSecret*.

Table B.2 – Implementation Conformance Report-Continued

Conformance test	**Produce**				**Use**				**Comment**	**Conf. class**
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
Compositing layers	O				R				Ignoring some of the codestreams means that the content might be interpreted wrong	B
Opacity channel	O				R					B
Chroma-key opacity	O				R					B
GML content										
WGS84 LL (EPSG::4326)	O[7]				R					B
UTM/WGS84 (EPSG::32hzz)	O				R					B
WGS84 vertical reference (EPSG::7030)	O				R					B
Earth Gravity Model 1996 (EPSG::5773)	O				R					B
Earth Gravity Model 2008 (EPSG::3855)	O				R					B
One and only one GML RectifiedGridCoverage for each codestream	R				R					B
The RectifiedGrid uses the attribute *srsName* to refer to the reference system by its EPSG code	R				R					B
The origin of the RectifiedGrid must have at least two coordinates in the coordinates list	R				R					B
The coordinate order of the origin must follow the coordinate order required by D.2.2.1.4	R				R					B

[7] At least one of the horizontal reference systems WGS84 LL or UTM/WGS84 must be used.

Table B.2 – Implementation Conformance Report-Continued

Conformance test	**Produce**				**Use**				**Comment**	**Conf. class**
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
The image coordinates for the *limits* element must be valid. The minimum coordinates must be <gml:low>0 0</gml:low> The maximum coordinates must not exceed the image size expressed in number of pixels in each dimension, i.e., following must apply: <gml:high>(width – 1) (height – 1)</gml:high>	R				R					B
The RectifiedGrid must have one instance of axisLabels (or two instances of *axisName)*.	R				R					B
The RectifiedGrid must have two instances of *offsetVector*	R				R					B
The instances of *offsetVector* must have two coordinates each	R				R					B
The coordinates in the offsetVectors must follow the coordinate order required by D.2.2.1.4	R				R					B
The metadata extent must comply with the extent given by the RectifiedGrid	R				R					B
Requirements for Referenceable Grid Coverage										
One and only one GML ReferenceableGridCoverage for each codestream	R				R					B

Table B.2 – Implementation Conformance Report-Continued

Conformance test	**Produce**				**Use**				**Comment**	**Conf. class**
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
The srsName attribute of any of element deriving from gmlcov:AbstractReferenceableGrid (ReferenceableGridBySensorModel, ReferenceableGridByTransformation, ReferenceableGridByVectors or ReferenceableGridByArrays) is provided and its value is consistent with the CRS defined in table D.9)	R				R					B
Coordinate lists in GML ReferenceableGrid Coverage have sufficient number of coordinates (2)	R				R					B
The axis order of the /gml:limits/gml:GridEnvelope/gml:high sub-element for any the corresponding ReferenceableGridBySensorModel, ReferenceableGridByTransformation, ReferenceableGridByVectors or ReferenceableGridByArrays, for Referenceable grid Coverage shall be given as width and height in that order.	R				R					B
Other requirements										
Metadata	R				R					B

Annex C: JP2 boxes
(normative)

Note: This annex documents required boxes. However use of other boxes is not excluded. For example uuid box may be used for specific purpose.

C.1 Required boxes

C.1.1 JPEG 2000 Signature box

Box type is 'jP\040\040'. The box size is always 12 bytes including the box length and type. The content is always '<CR><LF><0x87><LF>'. This box is always first in a JPEG 2000 file. This box could be used as a *magic number* for identifying a JPEG 2000 file without a correct file suffix. The inclusion of <CR> and <LF>, serves the purpose of catching file transfer errors. It is common that these two characters are translated while transferring a file between a Windows and a UNIX system if one forgets to use *binary transfer*.

C.1.2 File type box

Box type is 'ftyp'. The box has following fields:

BR: 4-byte string for the *brand field*. Shall always be 'jpx\040' in this profile, since ISO/IEC 15444-2 (JPEG 2000 with extensions) is needed for the GMLJP2 standard.

MinV: Minor version of the ISO/IEC 15444-1 specification. The value is an integer that shall always be 0.

CL: Compatibility list. This is a list of four-byte strings denoting different standards and profiles that the file conforms to. It shall contain at least the string 'jpx\040' (ISO/IEC 15444-2). In addition the compatibility list shall contain the string 'jp2\040' (ISO/IEC 15444-1), with the exception of files that uses one of the Part 2 variants of opacity.

Table C.1 – File type box

Field name	Size and type	Value
BR	4 byte string	'jpx\040'
MinV	4 byte unsigned integer	0
CL	List of 4 byte strings	'jpx\040' and possibly 'jp2\040' if the file uses one of the Part 2 options for opacity.

C.1.3 Reader requirement box

Box type is 'rreq'. The *reader requirement* box specifies both which features the JPEG 2000 file has used and which features that must be supported by a reader to fully use the file. This box must follow immediately after the *File type* box. This profile requires that the presence of GML should be signaled by this box.

ML: Mask length. The total size of the compatibility masks.

FUAM: Fully Understand Aspects Mask. Describes requirements for fully understand the image.

DCM: Decode Completely Mask. Describes requirements for displaying the image correctly.

NSF: Number of Standard Flags.

SF^{i}: Standard Flag. Standard features the file uses. This profile requires that one of them shall signal the presence of GML data by having one standard flag with the value **67**.

SM^{i}: Standard Mask. Compatibility mask with respect to the features specified by **SF^{i}**.

NVF: Number of Vendor Features.

VF^{i}: Vendor Feature.

VM^{i}: Vendor Mask.

Table C.2 – Reader requirement box

Field name	Size and type	Value
ML	1 byte unsigned integer	
FUAM	signed integer - size specified by ML	
DCM	signed integer - size specified by ML	
NSF	2 byte unsigned integer	
SF^{i}	2 byte unsigned integer for each flag	One of the flags must have the value **67**
SM^{i}	signed integer – size specified by ML	
NVF	2 byte unsigned integer	
VF^{i}	8 byte (128-bit UUID) per vendor feature	
VM^{i}	signed integer - size specified by ML	

Note that the DGIWG GMLJP2 Profile doesn't have any requirements for the mask fields (FUAM, DCM, SM and VM (the vendor mask is outside the scope anyway)). There is a description about how to calculate the masks in ISO 15444-2 M.6. Since this description is quite complicated and there are hardly no good examples on how to use these masks, this profile don't require the use of the masks. However, a producer that wishes to populate these masks is recommended to follow ISO 15444-2 M.6. In that case, a suggestion is that the FUAM requires that the GML Coverage is understood, and that the DCM does not require that the GML Coverage is understood.

C.1.4 JPEG 2000 header box

Box type is 'jp2h'. This is a superbox, that is, it contains other boxes. The box shall at least contain:

C.1.4.1 *Image header box*

Box type is 'ihdr'. This box shall always be the first in a JPEG 2000 header box. The box has following fields:

Height: 4-byte unsigned integer for the image height.

Width: 4-byte unsigned integer for the image width.

NC: 2-byte unsigned integer for the number of components.

BPC: 1-byte field for the number of bits per component minus 1. The value is in the 7 lowest bits. The highest bit gives if the components contain signed or unsigned values:

- 0: unsigned values
- 1: signed values

C: 1-byte unsigned integer for the compression type. Shall always be 7.

UnkC: 1-byte unsigned integer which says if the colourspace is unknown. Allowed values:

- 0: known colourspace specified in the *Colourspace Specification* boxes
- 1: unknown colourspace (in that case, the provided *Colourspace Specification* box shall be treated as it does accurately reproduce the image even though that is not known. There is always at least one *Colourspace Specification* box present no matter if the colourspace is known or not, since it is a requirement in the JPEG 2000 standard.)

IPR: 1-byte field indicating whether an IPR box with information about Intellectual Property Rights exists.

- 0: no information about intellectual property right information, thus no IPR box
- 1: there is information about intellectual property rights, so there is an IPR box

Table C.3 – Image header box

Field name	Size and type	Value
Height	4 byte unsigned integer	
Width	4 byte unsigned integer	
NC	2 byte unsigned integer	
BPC	1 byte	Number of bits per component minus 1
C	1 byte unsigned integer	7
UnkC	1 byte unsigned integer	0: known colourspace 1: unknown colourspace
IPR	1 byte	0: no information about intellectual property rights 1: there are information about intellectual property rights

C.1.4.2 *Colour specification box*

Box type is ‘colr’. A colour specification box defines a method to interpret the colourspace of the decompressed image data. It is possible to have multiple colour specification boxes, but at least one is required. The box has following fields:

METH: 1-byte unsigned integer specifying the method used. Allowed values:

- 1: enumerated colourspace
- 2: restricted ICC profile

PREC: 1-byte signed integer. Precedence. Reserved for ISO use. Value shall be set to zero.

APPROX: 1-byte unsigned integer for colourspace approximation. Value shall be set to zero.

EnumCS: 4-byte unsigned integer for enumerated colourspaces. EnumCS shall only exist if METH == 1. Allowed values:

- 16: sRGB
- 17: greyscale
- 18: sYCC

PROFILE: field with variable size containing an ICC profile. This field shall only exist if METH == 2.

Table C.4 – Colour specification box

Field name	Size and type	Value
METH	1 byte unsigned integer	1: enumerated colourspace 2: restricted ICC profile
PREC	1 byte signed integer	0
APPROX	1 byte unsigned integer	0
EnumCS	4 byte unsigned integer	16: sRGB 17: greyscale 18: sYCC
PROFILE	Variable size	

C.1.4.3 *Channel definition box*

Box type is ‘cdef’. The channel definition box specifies the meaning of each sample in each channel in the image. The box has following fields:

N: 2-byte unsigned integer. Number of channel descriptions in the image.

Cn^{i}: 2-byte unsigned integer. Channel index for channel number i.

Typ^{i}: 2-byte unsigned integer. Channel type for channel number i. Allowed values:

- 0: Colour image data for the associated colour.
- 1: Opacity. A sample value of 0 indicates that the sample is completely transparent, i.e. 0% opaque. A maximum sample value, i.e. as high as the bit depth allows, indicates 100% opaque.

Asoc^{i}: 2-byte unsigned integer. Channel association. Specifies the index of the colour for which the channel is associated. Allowed values:

- 0: This channel is associated as the image as a whole (for example, an opacity channel that should be applied to all colour channels).
- 1 to (2^{16}-2): This channel is associated with the colour indicated by this value.

Table C.5 – Channel definition box

Field name	Size and type	Value
N	2 byte unsigned integer	
Cn^{i}	2 byte unsigned integer	
Typ^{i}	2 byte unsigned integer	0: Colour image data for the associated colour. 1: Opacity. Sample value of 0 indicates completely transparent (0% opaque). Sample with maximum value (related to the bit depth) indicates 100% opaque.
Asoc^{i}	2 byte unsigned integer	0: This channel is associated as the image as a whole. 1 to (2^{16}-2): This channel is associated with the colour indicated by this value.

C.1.5 Intellectual property rights box

Box type is ‘jp2i’. The original purpose of this box is to store information about intellectual property rights. In addition, this profile requires that security constraints should be stored in this box. This is a way to ensure that security constraints won’t get lost in case the metadata gets lost. The only exception is if the *GMLJP2* file is *unclassified*. ISO/IEC 15444-2 provides a schema definition for metadata in XML format (ISO/IEC 15444-2 N.4.4), from which it is recommended to use the IPR element (ISO/IEC 15444-2 N.5.4 and N.6.4). This profile restricts the content further to require security classification in the IPR content.

Table C.6 – IPR fields

Field name	Description
IPR_USE_RESTRICTION XPath: /jp:IPR/jp:IPR_EXPLOITATION/jp:IPR_USE_RESTRICTION	Required unless *unclassified.* IPR_USE_RESTRICTION shall contain the classification code: *unclassified* *restricted* *confidential* *secret* *topSecret* This classification code should be aligned with the Classification level of the Resource Security constraint in DMF metadata. If different items in the file have different security levels, the highest security level should be used in this field.
IPR_MGMT_TYPE XPath: /jp:IPR/jp:IPR_EXPLOITATION/jp:IPR_MGMT_SYS/jp:IPR_MGMT_TYPE	Optional. IPR_MGMT_TYPE in this profile can be used to indicate the country or body responsible for the security classification.
IPR_PROTECTION XPath: /jp:IPR/jp:IPR_EXPLOITATION/jp:IPR_PROTECTION	Optional. IPR_PROTECTION in this profile should be used as a releasability list. Countries or bodies that the file is releasable to are listed separated by semicolon. This releasability information should be aligned with the Resource Releasability in DMF metadata.

C.1.5.1 *IPR_USE_RESTRICTION*

The security classification shall be stored in such way that it becomes clear while reading the XML content without a parser. That means that the security classification should be signaled by the use of one of following words in the IPR_USE_RESTRICTION element:

Table C.7 – Security classification keywords

Keyword
unclassified
restricted
confidential
secret
topSecret

The keywords are used in DMF 2.0. If the dataset is *unclassified*, this element is optional. Any other classification code makes this element mandatory. Releasability / Distribution limitations are handled under the IPR_PROTECTION (cf. table C.6).

Note: Any other codelist can be defined to fit to other classification systems.

C.1.5.2 *IPR_MGMT_TYPE*

This profile uses *IPR_MGMT_TYPE* to indicate the country responsible for the security classification.

C.1.5.3 *IPR_PROTECTION*

This profile uses *IPR_PROTECTION* for a releasability list. The list contains country names that the file is releasable to. The country names are separated by semicolon.

C.1.5.4 *Country codes*

IPR_MGMT_TYPE and *IPR_PROTECTION* shall use trigram codes from *ISO 3166-1 alpha-3* for the countries. For coalitions and other types of regions (like NATO or Asia), use trigrams from STANAG 1059 when available.

C.1.5.5 *Example*

Example of the content in an IPR box: unclassified, limited distribution to a list of countries.

```
<?xml version="1.0"?>
<jp:IPR
    xmlns:jp=http://www.jpeg.org/jpx/1.0/xml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <jp:IPR_EXPLOITATION>
        <jp:IPR_USE_RESTRICTION>unclassified</jp:IPR_USE_RESTRICTION>
        <jp:IPR_MGMT_SYS>
            <jp:IPR_MGMT_TYPE>SWE</jp:IPR_MGMT_TYPE>
        </jp:IPR_MGMT_SYS>
        <jp:IPR_PROTECTION>SWE;FRA;USA;GBR;ARE;ZAF;DEU;ITA;CZE
```

```
            </jp:IPR_PROTECTION>
        </jp:IPR_EXPLOITATION>
</jp:IPR>
```

Example of the content in an IPR box: confidential, limited distribution to NATO.

```
<?xml version="1.0"?>
<jp:IPR
    xmlns:jp=http://www.jpeg.org/jpx/1.0/xml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <jp:IPR_EXPLOITATION>
        <jp:IPR_USE_RESTRICTION>confidential</jp:IPR_USE_RESTRICTION>
        <jp:IPR_MGMT_SYS>
            <jp:IPR_MGMT_TYPE>NTT</jp:IPR_MGMT_TYPE>
        </jp:IPR_MGMT_SYS>
        <jp:IPR_PROTECTION>NTT
        </jp:IPR_PROTECTION>
    </jp:IPR_EXPLOITATION>
</jp:IPR>
```

C.1.6 Codestream box

Box type is ‘jp2c’. This box contains the actual raster data, i.e. the codestream. It is common to set the *LBox* field to zero, which means the codestream box contains the rest of the file content. But if we have multiple codestreams, we have to give the true box length for at least every codestream except the last.

Annex D: GML encoding
(normative)

D.1 GML objects

GML is used for different geographic properties in the image or the grid. OGC GMLJP2 standard v2.1 (08-085r8) specifies that GML 3.2.1 shall be used in GMLJP2. This profile uses GMLJP2RectifiedGridCoverage and GMLJP2ReferenceableGridCoverage elements.

D.1.1 GML box structure

The GML code is embedded in JP2 boxes:

Figure D.1 - GML box structure

The *asoc* boxes associates the labels with their GML instances, which are contained in *xml* boxes. The outer *asoc* box is labeled *"gml.data"*. The inner *asoc* box labeled *"gml.root-instance"* is required. All boxes must have unique names as their labels because these names are used in URIs referencing GML instances in the file.

D.1.2 gml.root-instance

This box shall contain a coverage description. It may also contain:

- Metadata instances
- Annotation instances

It may not contain anything else.

This instance can contain coverages and annotations for multiple codestreams.

D.1.3 GMLJP2 URI

The *GMLJP2 URI* is a Universal Resource Identifier for locating objects within a JPEG 2000 file. In a *rangeSet* element, it is used for locating the range of values that the *GMLJP2RectifiedGridCoverage* consists of. The range of values is the codestream.

A GMLJP2 URI is structured as:

gmljp2://[*resource type*]/[*resource id*][#fragment-identifier]

where *resource type* is *xml* or *codestream*.

D.1.3.1 *Codestream resource type*

This type of URI references a particular *codestream* within the JPEG 2000 file. The URI is written as:

gmljp2://codestream/[*codestream number*]

where [codestream number] is an integer greater than or equal to 0, identifying a particular codestream within the file.

If one intends to have multiple codestream files, this is the only way to know which codestream a specific GML instance applies to.

D.1.3.2 *XML resource type*

This type of URI references a particular *xml* box in the file. The URI has the following form:

gmljp2://xml/[label][#id]

where [label] identifies a particular *xml* box with its label. That means that the labeling of the *xml* boxes serves the purpose of addressing specific boxes in the file. In order for this to work, it is mandatory to use unique names in the labels. The last part, *[#id]*, which is optional, identifies a particular XML fragment in the xml instance by its *id* attribute. This mechanism can reference any type of *xml* box inside the GML box structure. This profile uses same mechanism to reference parts of an SVG document containing symbols for the annotations. That makes it possible to store SVG symbols embedded in the file as a library inside one single XML instance.

D.2 GML Georeferencing

For georeferencing an image or a grid, GMLJP2 uses *GMLCOV coverage*, which may be either *RectifiedGridCoverage* or *ReferenceableGridCoverage* (GMLCOVRGRID extension) via the GMLJP2RectifiedGridCoverage and/or GMLJP2ReferenceableGridCoverage elements. One specific codestream can and must have only one coverage. The coverage is in the *gml.root-instance* box.

D.2.1 GML structure

The GML is structured as:

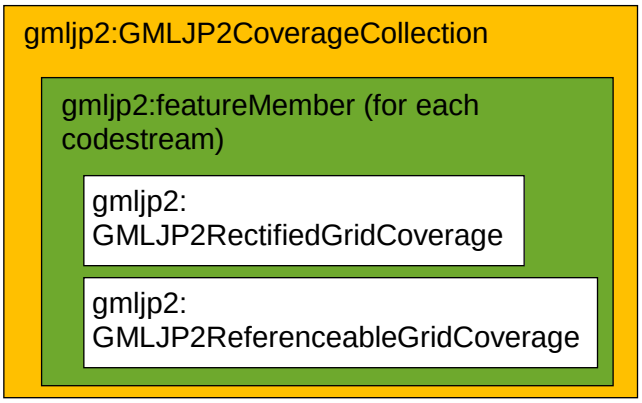

Figure D.2 – GML structure

The GML structure, providing the georeferencing information for the grid Coverage, uses as a root element, a *GMLJP2CoverageCollection* for the whole GMLJP2 file. That *GMLJP2CoverageCollection* has one feature member for each codestream. A feature member can be a *GMLJP2RectifiedGridCoverage* or a *GMLJP2ReferenceableGridCoverage*.

The required *gml.root-instance* box shall contain a *GMLJP2CoverageCollection* with a *GMLJP2RectifiedGridCoverage or/and a GMLJP2ReferenceableGridCoverage*.

D.2.2 GMLJP2RectifiedGridCoverage

This element gives the extent of the image or grid file. The rectified grid has an origin in a given coordinate reference system. The *origin's Point* element, references the CRS by either an EPSG code or a GMLJP2 URI to a CRS described in the JPEG 2000 file.

D.2.2.1 *domainSet*

The *domainSet* contains a *RectifiedGrid*, which takes care of the orientation parameters for the image or grid.

D.2.2.1.1 *Image coordinates*

The image coordinates uses a coordinate system with the dimensions *width* and *height* in that order, where (0, 0) is located in the upper left corner of the image.

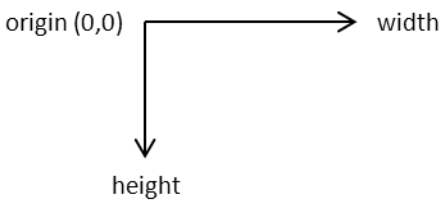

Figure D.3 – Image coordinate system

D.2.2.1.2 *RectifiedGrid*

The *domainSet* can contain a *gml:RectifiedGrid*:

```
<complexType name="RectifiedGridType">
    <complexContent>
        <extension base="gml:GridType">
            <sequence>
                <element name="origin" type="gml:PointPropertyType"/>
                <element name="offsetVector" type="gml:VectorType"
                maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Table D.1 – Required RectifiedGrid content

Element/attribute	Description
limits	An envelope for the grid in image coordinates (D.2.2.1.1).
origin	The origin in geo-coordinates. The origin is in the upper left corner of the grid, at the **center of the upper left pixel**. The axis order is defined by D.2.2.1.5.
offsetVector	The offsetVector gives the offset each pixel contributes to in each direction. Two offsetVector elements are required. The axis order is defined by D.2.2.1.5.
axisLabels	axisLabels provides a list of labels of the axes of the grid (e.g. Lat Long)
axisName (deprecated)	axisName gives the name of the axis. Two axisName elements are required. axisName may be used if axisLabels in not used.
srsName	This attribute is used for referring to the reference system that is used. It is required in the *RectifiedGrid* by this profile. Allowed values are listed in Table D.9.

D.2.2.1.3 *Extent in metadata (ISO/TS 19139)*

This profile requires that the geographical extent information does not contradict the extent information given by the GMLJP2 coverage. ISO/TS 19139 metadata can use GML for defining the geographical extent for a dataset. That means this requirement can be fulfilled by using the RectifiedGrid from previous paragraph in the metadata extent information.

Table D.2 – GML extent in ISO/TS 19139 metadata

Xpath to extent information in ISO/TS 19139
/gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:extent /gmd:EX_Extent/gmd:geographicElement/gmd:EX_BoundingPolygon /gmd:polygon/gml:RectifiedGrid

D.2.2.1.4 *Axis order*

The origin and the *offsetVector*, uses the specific axis order that is defined for the used reference system.

Table D.3 – Axis order for different reference systems

Reference system	Axis order
WGS84 LatLong	Left hand order (<Latitude> <Longitude>)
UTM	Right hand order (<Easting> <Northing>)

D.2.2.2 *rangeSet*

The rangeSet is defined as:

```
<complexType name="RangeSetType">
    <choice>
        <element ref="gml:File"/>
    </choice>
</complexType>
```

The *rangeSet* is the collection of values that the coverage covers. The *gml:File* element is the only relevant element in GMLJP2. This element is defined as:

```
<complexType name="FileType">
    <sequence>
        <element ref="gml:rangeParameters"/>
        <element name="fileName" type="anyURI"/>
        <element name="fileStructure" type="gml:FileValueModelType"/>
    </sequence>
</complexType>
```

Table D.4 – Required file type content

Element	Description
gml:rangeParameters	This element describes the values of the file with GML elements. This element is mandatory in GML 3.1.1, but it is not mandatory to have any content in it, so this profile don't require any content in this element. The content is optional and can be freely chosen within the constraints imposed by GML 3.1.1.
fileName	A GMLJP2 URI pointing to the codestream that the coverage describes
fileStructure	This element is irrelevant for this profile, but it is mandatory in the gml:File element, so it must be used anyway. The only supported value in GML 3.1.1 is "Record Interleaved", so just use that string no matter what the real file structure is.

The *rangeSet* element is mandatory in both *gmljp2:GMLJP2RectifiedGridCoverage* and *gmljp2:GMLJP2CoverageCollection*. In GMLJP2, this element refers to the codestream that a *gmljp2:GMLJP2RectifiedGridCoverage* is describing. Therefore it is only the *rangeSet* in the *gmljp2:GMLJP2RectifiedGridCoverage* that is relevant. The *rangeSet* for the *gmljp2:GMLJP2CoverageCollection* is inapplicable. The examples in *Annex H* clarify that.

D.2.3 GMLJP2ReferenceableGridCoverage

D.2.3.1 *domainSet*

The *domainSet* contains a *ReferenceableGrid*, which provides the parameters allowing the georeferencing of the image or grid by a set of mechanisms as provided by the *ReferenceableGridBySensorModel* or *ReferenceableGridByTransformation* or *ReferenceableGridByArray* or *ReferenceableGridByVectors*, as provided by rhe *GMLCOVRGRID* extension.

D.2.3.1.1 *Image / grid coordinates*

The image (or grid) coordinates uses an internal (to the image) coordinate system with the dimensions *width* and *height* in that order, where (0, 0) is located in the upper left corner of the image or grid, as described in D.2.2.1.1.

An optional attribute gridCRS is available for specifying a sequence of CRS definitions used by sensorModel (for ReferenceableGridBySensorModel) or the transformation or the concatenatedOperation (for ReferenceableGridByTransformation). It is recommended that for such ReferenceableGrid types, the first CRS definition of the gridCRS be specified as an ImageCRS that represents the internal CRS, while any subsequent CRS definitions represent any additional CRSs that may be required by the specific referenceable grid transformation. It is also recommended that none of these additional CRSs after the first be an ImageCRS.

D.2.3.1.2 *ReferenceableGridBySensorModel*

The domainSet of the *ReferenceableGridCoverage* can contain a gmlcovrgrid:ReferenceableGridBySensorModel:

```
<complexType name="ReferenceableGridBySensorModelType">
    <complexContent>
        <extension base="gmlcov:AbstractReferenceableGridType">
            <sequence>
                <element ref="gmlcovrgrid:gridCRS" minOccurs="0"/>
                <element name="sensorModel
            type="sml:AbstractProcessPropertyType" maxOccurs="unbounded"/>
                <element name="sensorInstance"
            type="sml:AbstractProcessPropertyType" minOccurs="0"
            maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Table D.5 – Required ReferenceableGridBySensorModel content

Element/attribute	Description
srsName	This attribute is used for referring to the reference system that is used. It is required in the *ReferenceableGrid* by this profile. Allowed values are listed in Table D.9.
limits	An envelope for the grid in image coordinates (D.2.3.1.1).
axisLabels	This attribute gives the names of the 2 axis. Two names are required.
gridCRS	An optional sequence of CRS definitions used by sensorModel
sensorModel	SensorML model allowing to yield the direct positions of the grid (via the geopositioning model) – cf. GMLCOVRGRID specification (SML::AbstractProcessPropertyType)
sensorInstance	Parameter values for the sensor model (SML::AbstractProcessPropertyType)

ReferenceableGridBySensorModel fully defines a sensor model (via SensorML 2.0) that is used to geolocate the referenceable grid. Such a sensor model involves two inputs: one or more sensor model descriptions containing free variables (using SML::sensorModel) plus a respective set of variable instantiations (using SML::sensorInstance).

A sequence of CRS is optionally defined in gridCRS. The definition of the grid CRS, intermediate CRS (if any) and external CRS is usually defined in the sensorModel, by the referenceFrame attribute specified for the Image Coordinate system (as SML input) and the external CRS (as SML output) and the intermediate CRS (if any) for each SML process describing the sensor model. The gridCRS sequence (if present) must therefore be consistent with this sensorModel suite of referenceFrame attributes.

D.2.3.1.3 *ReferenceableGridByTransformation*

The domainSet of the *ReferenceableGridCoverage* can contain a gmlcovrgrid:ReferenceableGridByTransformation:

```
<complexType name="ReferenceableGridByTransformationType">
    <complexContent>
        <extension base="gmlcov:AbstractReferenceableGridType">
            <sequence>
                <element ref="gmlcovrgrid:gridCRS" minOccurs="0"/>
                <choice>
                    <element name="transformation"
                    type="gml:TransformationPropertyType"/>
                    <element name="concatenatedOperation"
                    type="gml:ConcatenatedOperationPropertyType"/>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Table D.6 – Required ReferenceableGridByTransformation content

Element/attribute	Description
srsName	This attribute is used for referring to the reference system that is used. It is required in the *ReferenceableGrid* by this profile. Allowed values are listed in Table D.9.
limits	An envelope for the grid in image coordinates (D.2.3.1.1).
axisLabels	This attribute gives the names of the 2 axis. Two names are required.
gridCRS	An optional sequence of CRS definitions used by sensorModel
transformation	A general coordinate transformation using a sequence of operations based on GML::method that have an unbounded set of GML::parameterValue (GML:: Transformation PropertyType)
concatenatedOperation	An ordered sequence of two or more coordinate operations (GML:: ConcatenatedOperation PropertyType)

ReferenceableGridByTransformation specifies either a GML::Transformation or a GML::ConcatenatedOperation to specify the relationship between positions in the source CRS (image CRS) and corresponding positions in the target CRS (external CRS).

A sequence of CRS is optionally defined in gridCRS.

D.2.3.1.4 *ReferenceableGridByVectors*

The domainSet of the *ReferenceableGridCoverage* can contain a gmlcovrgrid:ReferenceableGridByVectors:

```
<complexType name="ReferenceableGridByVectorsType">
    <complexContent>
        <extension base="gmlcov:AbstractReferenceableGridType">
            <sequence>
                <element name=”origin” type=”gml:PointPropertyType"/>
                <element name=”generalGridAxis”
            type=”gmlcovrgrid:GeneralGridAxisPropertyType" maxOccurs=”unbounded”/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Table D.7 – Required ReferenceableGridByVectors content

Element/attribute	Description
srsName	This attribute is used for referring to the reference system that is used. It is required in the *ReferenceableGrid* by this profile. Allowed values are listed in Table D.9.
limits	An envelope for the grid in image coordinates (D.2.3.1.1).
axisLabels	This attribute gives the names of the 2 axis. Two names are required.
origin	The origin of the referenceable grid in the external CRS (GML:: PointPropertyType)
generalGridAxis	Used to define an offset vector and support parameters (GeneralGridAxisPropertyType)

ReferenceableGridByVectors defines a referenceable grid by specifying an origin and a set of offset vectors, with multiplicative coefficients that scale the offset vectors to generate a (potentially) irregularly-spaced grid.

D.2.3.1.5 *ReferenceableGridByArray*

The domainSet of the *ReferenceableGridCoverage* can contain a gmlcovrgrid:ReferenceableGridByArray:

```
<complexType name="ReferenceableGridByArrayType">
    <complexContent>
        <extension base="gmlcov:AbstractReferenceableGridType">
            <sequence>
                <group ref=”gml:geometricPositionListGroup"/>
                <element name=”sequenceRule” type=”gml:SequenceRuleType”/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Table D.8 – Required ReferenceableGridByArrays content

Element/attribute	Description
srsName	This attribute is used for referring to the reference system that is used. It is required in the *ReferenceableGrid* by this profile. Allowed values are listed in Table D.9.
limits	An envelope for the grid in image coordinates (D.2.3.1.1).
axisLabels	This attribute gives the names of the 2 axis. Two names are required.
GML::posList (for example)	Specifies the array of grid point locations in the external CRS, via either a GML::posList or a sequence of GML::pos or GML::Point objects (GML:: geometricPositionList Group)
sequenceRule	Specifies the sequence order of the grid point locations over the grid. (GML:: SequenceRuleType)

ReferenceableGridByArrays defines a referenceable grid by listing an array of grid point locations explicitly, as a sequence of direct positions in a defined sequence order over the grid.

D.2.4 EPSG georeference in RectifiedGrid@srsName or ReferenceableGrid@srsName

GML features can refer to coordinate reference systems with the attribute *srsName*. This profile mandates that the attribute *srsName* in the *gml:boundedBy/gml:Envelope* element shall be used for declaring the reference system. This declaration shall apply globally to all GML features in the file.

The reference system used, is referred to by an OGC namespace with EPSG codes: *urn:ogc:def:<object type>:EPSG:<EPSG version>:<EPSG code>*

where the following values are allowed by this profile:

Table D.9 – Reference codes from the EPSG namespace

EPSG code	Reference system
urn:ogc:def:crs:EPSG::4326 *or* http://www.opengis.net/def/crs/EPSG/0/4326	GCS_WGS84
urn:ogc:def:crs:EPSG::326zz and urn:ogc:def:crs:EPSG::327zz *or* http://www.opengis.net/def/crs/EPSG/0/326zz and http://www.opengis.net/def/crs/EPSG/0/327zz	UTM for each zone in the northern hemisphere respectively the southern hemisphere
urn:ogc:def:ellipsoid:EPSG::7030	WGS84 ellipsoid as a vertical reference
urn:ogc:def:crs:EPSG::5773	Earth Gravity Model 1996
urn:ogc:def:crs:EPSG::3855	Earth Gravity Model 2008

Note: other CRS specified by DGIWG Geodetic Codes and Parameters Registry are allowed and must be identified by their EPSG Geodetic Parameter registry identifier

The table above doesn't use version numbers in the namespace references, but the profile allows version numbers. The profile does not mandate any particular version number. The EPSG codes used in the table will be used as above given reference systems no matter which version number is used.

D.2.5 CRSDictionary.gml

This is an optional box which can be used for defining a CRS in GML. With this box, it is possible to use a CRS which is not defined in any CRS namespace.

D.3 Metadata

Metadata can be connected to any kind of GML entity with a *metadataProperty* element. At least, the *RectifiedGridCoverage* should have its *metadataProperty* element populated. This element is defined as:

```
<complexType name="GenericMetaDataType" mixed="true">
    <complexContent mixed="true">
        <extension base="gml:AbstractMetaDataType">
            <sequence>
                <any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

This element can contain metadata according to external schemas.

DMF (in its version 2) is the DGIWG standard to be used for documenting metadata. The schemas from ISO/TS 19139 or the newest schemas from ISO 19115-3 are both handled by DMF.

Annex E: Metadata
(normative)

E.1 Metadata location

This profile requires that the metadata may be either embedded in the GML document in the JPEG 2000 file (connected to the *RectifiedGridCoverage* (D.2.2) or to the *ReferenceableGridCoverage* (D.2.3)) or supplied as an external file. There are good reasons for both practices. Metadata that is embedded in the file will not be lost even though a user forgets the external metadata file. Metadata in an external file enables a GIS system to handle metadata in the same way regardless of file format. Not all file formats allows metadata embedded in the file. If there is a conflict between the embedded metadata and the metadata in the external file, the embedded metadata will take precedence.

E.2 Metadata requirements

This profile has two requirements on the metadata.

E.2.1 Metadata conforms to DMF

Instead, the profile only requires that the metadata shall conform to DMF (DGIWG Metadata Foundation). For detailed requirements and conformance tests, this profile refers to the DMF work, in particular following parts:

Table E.1 – DMF requirement classes

Requirement class	Description
DMF/Core	Minimum set of metadata elements
DMF/Common	Additional set of metadata elements
DMF/Data	Metadata for datasets, series and tiles
DMF/Data+	Metadata for coverage results and a number of elements from ISO 19115-2 (extensions for imagery and gridded data)
DMF/Sensor	Metadata for Sensor acquisition, Spectral mode, or acquisition or processing modes for imagery or gridded data

E.2.2 Coverage information in metadata

Metadata that follows ISO/TS 19139 usually contains information about the geographical extent of the dataset. The extent defined by the metadata, shall not contradict the extent given by the GML coverage in the GMLJP2 georeference. ISO/TS 19139 has the possibility to use GML objects, such as the *RectifiedGrid*, which is also used in the *GMLJP2RectifiedGridCoverage* in GMLJP2.

E.3 Security classification

The part of the metadata that describes the security classification shall be stored in an *Intellectual Property Rights* box in addition to the security classification information in the ISO/TS 19139 metadata. This is described more in C.1.5. The reason for this additional location is to ensure that the security classification information doesn't get lost from the GMLJP2 file. The only exception from this requirement is if the file is *unclassified*.

Annex F: Use cases (informative)

This appendix lists the use cases for this profile.

Table F.1 – Use cases

UC#	Use case	Initiating actor	Receiving actor	Description
1	**Raster types**			
1.1	Gray tone	Raster image processor	File system	Raster image with a single band is compressed.
1.2	RGB	Raster image processor	File system	Colour raster image with three bands denoting Red, Green and Blue respectively, is compressed.
1.3	Multispectral	Raster image processor	File system	A multispectral image is compressed.
1.4	Discrete raster	Raster image processor	File system	An image with large solid colour areas is compressed.
1.5	Palletized image	Raster image processor	File system	An image which uses a palette for the colours, i.e. indexed colours, is compressed.
1.6	Elevation grid	Raster image processor	File system	An elevation grid is compressed.
1.7	Raster map	Raster image processor	File system	A thematic map where the pixel values denotes different objects in the map.
1.8	Sensor Image	Sensor or acquisition instrument	File system	A sensor or scanned image where the pixel values denotes different objects captured from acquisition instrument
2	**GML**			
2.1	Geo-rectified	Photogrammetric expert	User	A geo-rectified image has its geo-rectification parameters encoded in GML.
2.2	Ortho-rectified	Photogrammetric expert	User	An ortho-rectified image has its ortho-rectification parameters encoded in GML.
2.3	Referenceable	Collection system or ground station	User	A sensor or scanned image as collected by instrument or delivered by ground station. Not geometrically processed (except from instrument geometric distortion from calibration)
3	**Metadata**			
3.1	Embed metadata	Geodata handler	User	The image's metadata is embedded in the image file.
3.2	Extract metadata	User	User	The image's metadata is read from the image file.
4	**Data access**			
4.1	Query WMS or WCS and getting a GMLJP2 file as a response	WMS- or WCS-client	WMS- or WCS-server	An OWS client sends a GetMap request to a WMS-server or a GetCoverage request to a WCS-server, and gets a GMLJP2 file as a response.
4.2	Query metadata from a JPIP-server	JPIP-client	JPIP-server	A JPIP-client request metadata from a JPIP-server, and gets a JPIP data-bin containing only metadata.

Annex G: JPEG 2000 profile extensions (informative)

G.1 Profile Extensions

Both GMLJP2 and this profile of GMLJP2 use extensions from ISO/IEC 15444-2.

G.1.1 Extensions required by baseline GMLJP2

The extensions listed in this paragraph, are used for georeferencing the image or the grid with GML. A JPEG 2000 reader which conforms to ISO/IEC 15444-1 is able to render the codestream. It is only the GML document with georeferencing information that would be ignored.

G.1.1.1 *Association box*

The purpose of the association box is to associate certain boxes with other boxes into logical groups. It is possible to consider the association box as a *folder* in a file system. GMLJP2 uses this mechanism to group together different boxes with GML data.

G.1.1.2 *Label box*

The label box contains only text. GMLJP2 uses the label box to identify certain GML instances in a JPEG 2000 file.

G.1.1.3 *Reader requirement box*

The reader requirement box specifies which features that have been used in a file and which features the reader must support to fully use the file.

G.1.2 Additional extensions used by DGIWG GMLJP2

The extensions listed in this paragraph, are used for *compositing layers*, which is a feature for combining different codestreams. A JPEG 2000 reader which conforms to ISO/IEC 15444-1 is unable to render the compositing layer. The reader will only read the first codestream. That means that if a file uses any of the Part 2 options for opacity (5.5.2 and 5.5.3), the reader wouldn't render the opacity.

G.1.2.1 *Opacity box*

The opacity box gives the option to use chroma-key based transparency (ISO/IEC 15444-2, M.11.7.6).

G.1.2.2 *Compositing layer header box*

The compositing layer box is required for combining different codestreams (ISO/IEC 15444-2, M.11.7).

G.1.2.3 *Codestream registration box*

Codestreams that shall be included in a compositing layer must be registered in the codestream registration box (ISO/IEC 15444-2, M.11.7.7).

Annex H: Examples of GML coverages (informative)

H.1 Examples of GML Rectified Grid coverages in JPEG 2000 files

This annex provides examples of typical GML coverages that can be expected in a GMLJP2 file compliant with this profile.

H.1.1 A WGS84 LL example of a GMLJP2RectifiedGridCoverage

```
<?xml version="1.0" encoding="UTF-8"?>
<gmljp2:GMLJP2CoverageCollection gml:id="JPEG 2000_0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0"
xmlns:gmljp2="http://www.opengis.net/gmljp2/2.0" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gmljp2/2.0
http://schemas.opengis.net/gmljp2/2.0/gmljp2.xsd">
    <gml:domainSet nilReason="inapplicable"/>
    <gml:rangeSet>
        <gml:DataBlock>
        <gml:rangeParameters nilReason="inapplicable"/>
        <gml:doubleOrNilReasonTupleList>inapplicable</gml:doubleOrNilReasonTupleList>
        </gml:DataBlock>
    </gml:rangeSet>
    <gmlcov:rangeType>
        <swe:DataRecord>
            <swe:field name="Collection"/>
        </swe:DataRecord>
    </gmlcov:rangeType>
    <gmljp2:featureMember>
        <gmljp2:GMLJP2RectifiedGridCoverage gml:id="JP2_Coverage1">
            <gml:boundedBy>
                <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326"
            srsDimension="2" axisLabels="Lat Long" uomLabels="deg deg">
                    <gml:lowerCorner>0 0</gml:lowerCorner >
                    <gml:upperCorner>4999 4999</gml:upperCorner>
                </gml:Envelope>
            </gml:boundedBy>
            <gml:domainSet>
                <gml:RectifiedGrid gml:id="rg0001" dimension="2">
                    <gml:limits>
                        <gml:GridEnvelope>
                            <gml:low>0 0</gml:low>
                            <gml:high>4999 4999</gml:high>
                        </gml:GridEnvelope>
                    </gml:limits>
                    <gml:axisLabels>Lat Long</gml: axisLabels>
                    <gml:origin>
                        <gml:Point gml:id="P0001">
                            <!-- "Upper-left" image origin -->
                            <gml:pos>19.1234567   37.1234567</gml:pos>
```

```
                                    </gml:Point>
                                </gml:origin>
                                <gml:offsetVector>0.0   0.00001234</gml:offsetVector>
                                <gml:offsetVector> -0.00001234   0.0</gml:offsetVector>
                            </gml:RectifiedGrid>
                        </gml:domainSet>
                        <gml:rangeSet>
                            <gml:File>
                                <gml:rangeParameters/>
                                <gml:fileName>gmljp2://codestream</gml:fileName>
                                <gml:fileStructure>inapplicable</gml:fileStructure>
                            </gml:File>
                        </gml:rangeSet>
                        <gmlcov:rangeType>
                            <swe:DataRecord>
                                <swe:field name="Panchromatic">
                        <swe:Quantity definition="http://www.opengis.net/def/ogc-
    eo/opt/SpectralMode/Panchromatic">
                                    <swe:description>Panchromatic</swe:description>
                                    <swe:uom code="unity"/>
                                    </swe:Quantity>
                                </swe:field>
                            </swe:DataRecord>
                        </gmlcov:rangeType>
                    </gmljp2:GMLJP2RectifiedGridCoverage>
                </gmljp2:featureMember>
            </gmljp2:GMLJP2CoverageCollection>
```

This example is a *GMLJP2RectifiedGridCoverage* describing the coverage of a rectangular raster with 10 000 pixel in the northing direction and 5000 pixels in the easting direction. The coverage is in Sudan, close to the coast. The coverage uses *WGS84 LL* coordinates, as indicated by the EPSG code (4326). The origin and the offset vectors uses left hand ordered coordinates. That means the latitude is 19.1234567 decimal degrees and the longitude is 37.1234567 decimal degrees.

H.1.2 A UTM example of a GMLJP2RectifiedGridCoverage

```
<?xml version="1.0" encoding="UTF-8"?>
<gmljp2:GMLJP2CoverageCollection gml:id="JPEG 2000_0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0"
xmlns:gmljp2="http://www.opengis.net/gmljp2/2.0" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gmljp2/2.0
http://schemas.opengis.net/gmljp2/2.0/gmljp2.xsd">
    <gml:domainSet nilReason="inapplicable"/>
    <gml:rangeSet>
        <gml:DataBlock>
        <gml:rangeParameters nilReason="inapplicable"/>
        <gml:doubleOrNilReasonTupleList>inapplicable</gml:doubleOrNilReasonTupleList>
        </gml:DataBlock>
    </gml:rangeSet>
    <gmlcov:rangeType>
        <swe:DataRecord>
            <swe:field name="Collection"/>
        </swe:DataRecord>
    </gmlcov:rangeType>
    <gmljp2:featureMember>
        <gmljp2:GMLJP2RectifiedGridCoverage gml:id="JP2_Coverage1">
            <gml:boundedBy>
            <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/32637"
        srsDimension="2" axisLabels="X Y" uomLabels="m m">
                        <gml:lowerCorner>0 0</gml:lowerCorner >
                        <gml:upperCorner>4999 4999</gml:upperCorner>
                    </gml:Envelope>
                </gml:boundedBy>
                <gml:domainSet>
                    <gml:RectifiedGrid gml:id="rg0001" dimension="2">
                        <gml:limits>
                            <gml:GridEnvelope>
                                <gml:low>0 0</gml:low>
                                <gml:high>4999 4999</gml:high>
                            </gml:GridEnvelope>
                        </gml:limits>
                        <gml: axisLabels>X Y</gml: axisLabels>
                        <gml:origin>
                            <gml:Point gml:id="P0001">
                                <!-- "Upper-left" image origin -->
                                <gml:pos>302466.089   2115546.978</gml:pos>
                            </gml:Point>
                        </gml:origin>
                        <gml:offsetVector>1.327588  0.0</gml:offsetVector>
                        <gml:offsetVector>0.0  -1.372799</gml:offsetVector>
                    </gml:RectifiedGrid>
                </gml:domainSet>
```

```
                <gml:rangeSet>
                    <gml:File>
                        <gml:rangeParameters/>
                        <gml:fileName>gmljp2://codestream</gml:fileName>
                        <gml:fileStructure>inapplicable</gml:fileStructure>
                    </gml:File>
                </gml:rangeSet>
                <gmlcov:rangeType>
                    <swe:DataRecord>
                        <swe:field name="Panchromatic">
                <swe:Quantity definition="http://www.opengis.net/def/ogc-
eo/opt/SpectralMode/Panchromatic">
                                    <swe:description>Panchromatic</swe:description>
                                    <swe:uom code="unity"/>
                                </swe:Quantity>
                        </swe:field>
                    </swe:DataRecord>
                </gmlcov:rangeType>
            </gmljp2:GMLJP2RectifiedGridCoverage>
        </gmljp2:featureMember>
</gmljp2:GMLJP2CoverageCollection>
```

This example uses UTM coordinates. The area is in Sudan, close to the coast, like in the WGS84 example. The EPSG code (32637) indicates UTM zone 37 on the northern hemisphere. The origin and the offset vectors use right hand coordinates. That means the easting coordinate is 302466.089 m and the northing coordinate is 2115546.978 m.

H.2 Examples of GML Referenceable Grid coverages in JPEG 2000 files

H.2.1 A SensorModel example of a GMLJP2ReferenceableGridCoverage with a RPC (Rational Polynomial coefficient) replacement model

```
<?xml version="1.0" encoding="UTF-8"?>
<gmljp2:GMLJP2CoverageCollection gml:id="Rio1" xmlns="http://www.opengis.net/gml/3.2"
xmlns:gmlcovrgrid="http://www.opengis.net/gmlcov/gmlcovrgrid/1.0"
xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0"
xmlns:gmljp2="http://www.opengis.net/gmljp2/2.1" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gmljp2/2.1
D:\Documents\OGC\GMLJP2\v2.1\gmljp2_v2_1.xsd">
    <gml:domainSet nilReason="inapplicable"/>
    <gml:rangeSet>
        <gml:DataBlock>
            <gml:rangeParameters nilReason="inapplicable"/>

    <gml:doubleOrNilReasonTupleList>inapplicable</gml:doubleOrNilReasonTupleList>
        </gml:DataBlock>
    </gml:rangeSet>
    <gmlcov:rangeType>
        <swe:DataRecord>
```

```
                    <swe:field name="Collection"> </swe:field>
                </swe:DataRecord>
            </gmlcov:rangeType>
            <gmljp2:featureMember>
                <gmljp2:GMLJP2ReferenceableGridCoverage gml:id="RefGridCov_0">
                    <gml:boundedBy>
                <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326"
axisLabels="Lat Long" uomLabels="deg deg" srsDimension="2">
                    <gml:lowerCorner>-43.2003684     -23.0263692</gml:lowerCorner>
                    <gml:upperCorner>-43.0428245     -22.8894003</gml:upperCorner>
                </gml:Envelope>
            </gml:boundedBy>
                    <domainSet>
                        <gmlcovrgrid:ReferenceableGridBySensorModel gml:id="RefGrid_0"
dimension="2" srsDimension="3" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
                            <limits>
                                <GridEnvelope>
                                    <low>0 0</low>
                                    <high>7336 7986</high>
                                </GridEnvelope>
                            </limits>
                            <axisLabels>Row Column</axisLabels>
                            <gmlcovrgrid:sensorModel
xlink:title="OGC:sensorML:CSM_RPC00B"
xlink:href="D:\Documents\DGIWG\DATP\A13\SensorModels\BLOCKA\CSM_RPC00B.xml"/>
                            <gmlcovrgrid:sensorInstance>
                                <sml:SimpleProcess gml:id="SensorML_0"
xmlns="http://www.opengis.net/gml/3.2" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:sml="http://www.opengis.net/sensorml/2.0" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/sensorml/2.0
      http://schemas.opengis.net/sensorml/2.0/sensorML.xsd">
        <gml:description>
    An example of an instance of the Community Sensor Model for RPC00B, as proposed by
DGIWG in SML2.0
  </gml:description>
        <gml:identifier
codeSpace="uid">http://www.example.org/def/DGIWG/0/sensor/1.0/CSM_RPC00B</gml:identifi
er>
        <sml:typeOf xlink:title="http://www.example.org/def/DGIWG/0/sensor/CSM_RPC00B"
xlink:href="D:\Documents\DGIWG\DATP\A13\SensorModels\RPC00B\CSM_RPC00B.xml"/>
        <sml:configuration>
            <sml:Settings>
                <sml:setValue
ref="parameters/RPC00B_Params/ERR_BIAS">1.480000000000000e+00</sml:setValue>
                <sml:setValue
ref="parameters/RPC00B_Params/ERR_RAND">1.290000000000000e+00</sml:setValue>
                <sml:setValue
ref="parameters/RPC00B_Params/LINE_OFF">3992</sml:setValue>
                <sml:setValue
ref="parameters/RPC00B_Params/SAMP_OFF">3667</sml:setValue>
                <sml:setValue ref="parameters/RPC00B_Params/LAT_OFF">-
2.295800000000000e+01</sml:setValue>
                <sml:setValue ref="parameters/RPC00B_Params/LONG_OFF">-
4.312140000000000e+01</sml:setValue>
                <sml:setValue
ref="parameters/RPC00B_Params/HEIGHT_OFF">132</sml:setValue>
```

<sml:setValue ref="parameters/RPC00B_Params/LINE_SCALE">3993</sml:setValue>
<sml:setValue ref="parameters/RPC00B_Params/SAMP_SCALE">3668</sml:setValue>
<sml:setValue ref="parameters/RPC00B_Params/LAT_SCALE">7.660000000000000e-02</sml:setValue>
<sml:setValue ref="parameters/RPC00B_Params/LONG_SCALE">8.110000000000001e-02</sml:setValue>
<sml:setValue ref="parameters/RPC00B_Params/HEIGHT_SCALE">501</sml:setValue>
<sml:setArrayValues ref="parameters/RPC00B_Params/LINE_NUM_COEFF">
<sml:ArrayValues>
<sml:encoding>
<swe:TextEncoding tokenSeparator=" " blockSeparator="
"/>
</sml:encoding>
<sml:value>
5.997692000000000e-04 5.300710000000000e-02 1.060405000000000e+00 6.781648000000000e-03 1.304341000000000e-04 6.837408000000000e-07 -1.039336000000000e-05 -3.534793000000000e-04 -9.213983000000000e-04 -5.224494000000000e-07 -1.249479000000000e-06 -1.496295000000000e-06 -1.425152000000000e-05 -2.031819000000000e-06 -2.895077000000000e-05 -1.573550000000000e-04 -4.047168000000000e-05 -2.438193000000000e-07 -1.354090000000000e-06 -2.594373000000000e-07
</sml:value>
</sml:ArrayValues>
</sml:setArrayValues>
<sml:setArrayValues ref="parameters/RPC00B_Params/LINE_DEN_COEFF">
<sml:ArrayValues>
<sml:encoding>
<swe:TextEncoding tokenSeparator=" " blockSeparator="
"/>
</sml:encoding>
<sml:value>
1.000000000000000e+00 -1.458582000000000e-04 5.512421000000000e-04 8.040210000000000e-06 1.830241000000000e-05 -9.966218000000000e-07 2.431857000000000e-06 -2.748566000000000e-05 1.090844000000000e-04 -3.816117000000000e-05 -1.621419000000000e-06 -1.142476000000000e-07 -1.235324000000000e-04 0.000000000000000e+00 -6.619828000000000e-06 -8.272211000000000e-04 -1.318910000000000e-07 -3.961112000000000e-08 -1.624079000000000e-05 0.000000000000000e+00
</sml:value>
</sml:ArrayValues>
</sml:setArrayValues>
<sml:setArrayValues ref="parameters/RPC00B_Params/SAMP_NUM_COEFF">
<sml:ArrayValues>
<sml:encoding>
<swe:TextEncoding tokenSeparator=" " blockSeparator="
"/>
</sml:encoding>
<sml:value>
6.860120000000000e-03 -1.032202000000000e+00 9.002337000000000e-04 2.926985000000000e-02 9.897145000000000e-04 -1.770975000000000e-04 1.710241000000000e-04 -6.233348000000000e-03 -3.066917000000000e-04

```
9.441388000000000e-06 2.800347000000000e-06 -3.548515000000000e-05 -
4.343459000000000e-05 -2.916795000000000e-06 -4.145524000000000e-05 -
6.301348999999999e-04 -6.818026000000000e-08 -3.303297000000000e-06 -
1.380852000000000e-05 8.169272000000001e-08
                </sml:value>
                        </sml:ArrayValues>
                    </sml:setArrayValues>
                    <sml:setArrayValues
ref="parameters/RPC00B_Params/SAMP_DEN_COEFF">
                        <sml:ArrayValues>
                            <sml:encoding>
                                <swe:TextEncoding tokenSeparator=" "
blockSeparator="&#x0A;"/>
                            </sml:encoding>
                            <sml:value>
1.000000000000000e+00 6.509521000000000e-04 1.005558000000000e-03 -
3.837196000000000e-04 1.256965000000000e-05 -2.208074000000000e-06 -
1.916325000000000e-08 8.157947000000001e-06 -4.275273000000000e-05
2.746759000000000e-06 0.000000000000000e+00 -1.341338000000000e-07
8.229087000000000e-07 -1.788627000000000e-08 -3.235013000000000e-07 -
1.528876000000000e-06 0.000000000000000e+00 -4.502659000000000e-08
3.425873000000000e-08 0.000000000000000e+00
                </sml:value>
                        </sml:ArrayValues>
                    </sml:setArrayValues>
                </sml:Settings>
            </sml:configuration>
</sml:SimpleProcess>

                            </gmlcovrgrid:sensorInstance>
                        </gmlcovrgrid:ReferenceableGridBySensorModel>
                    </domainSet>
                    <gml:rangeSet>
                        <gml:File>
                            <gml:rangeParameters/>
                            <gml:fileName>gmljp2://codestream/0</gml:fileName>
                            <gml:fileStructure>Record Interleaved</gml:fileStructure>
                        </gml:File>
                    </gml:rangeSet>
                    <gmlcov:rangeType>
                        <swe:DataRecord>
                            <swe:field name="Color">
                                <swe:CountRange
definition="http://www.opengis.net/def/ogc-eo/opt/SpectralMode/COLOR">
                                    <swe:label>Red, Green, Blue NIR bands for pixel
values</swe:label>
                                    <swe:description>4 bands (RGBNIR) - Dynamic per
band (16 bits)</swe:description>
                                    <swe:value>0 65535</swe:value>
                                </swe:CountRange>
                            </swe:field>
                        </swe:DataRecord>
                    </gmlcov:rangeType>
                </gmljp2:GMLJP2ReferenceableGridCoverage>
            </gmljp2:featureMember>
</gmljp2:GMLJP2CoverageCollection>
```

This example is a *GMLJP2ReferenceableGridCoverage* describing the coverage of a rectangular raster with 7987 pixels in the northing direction and 7337 pixels in the easting direction, based on the ReferenceableGridBySensorModel, and on the RPC00B specification for Rational Polynomial Coefficent replacement model (cf. STDI-0002). The coverage uses *WGS84 LL* coordinates, as indicated by the EPSG code (4326).

The sensor model (SML 2.0) for this RPC00B sensor model is contained on a dedicated document/file, for reusability (cf. https://portal.dgiwg.org/files/?artifact_id=68558).

Note: Please note that the sensor model and parameters are presently supposed to be under DGIWG domain. It might in some close future move to OGC domain, according to OGC Name Type Specification for Sensor Models and Parameters.

Annex I: Software used for testing purposes (informative)

I.1 Software recommendations

Examination of GMLJP2 files can be done with the help of a few different software packages.

I.2 Kdu_show

Kdu_show let the user inspect a JPEG 2000 file. The program is included in the demonstration package from *Kakadu Software* which can be downloaded from:

http://www.kakadusoftware.com/index.php?option=com_content&task=view&id=26&Itemid=22

This program has a metadata viewer (opened by *Metadata -> Open Metashow*), which can be used for inspecting the metadata that is stored in the JPEG 2000 boxes, which includes the GML Coverage.

The program can also display the codestream properties (*File – Properties*) for viewing information about the codestream.

I.3 XML Spy

XML Spy is a tool from one of the leading XML tool developers, Altova. This tool is used for validating the GML Coverage. *XML Spy* can't validate the GML coverage in a GMLJP2 file directly. In order to validate the GML Coverage, the user have to use a tool, such as *kdu_show*, which lets the user copy and paste the GML code to an appropriate location for the validation, i.e. an xml file.

I.3.1 XML Schema reference

Note that in order to validate an XML file with *XML Spy*, it is necessary to add a reference to the XML schema that the file shall follow. Therefore the tester has to add the following attribute to the root element in the GML Coverage:

xsi:schemaLocation="http://www.opengis.net/gml **file:///D:/dgiwg/jp2/GML-3.2.1/profiles/DGIWGgmlJP2Profile/1.0.0/DGIWGgmlJP2Profile.xsd**"

(the file path in bold, shall be changed to point to the files location on the particular system where the validation takes place).

I.4 Microsoft .Net Framework 4.0

This is not a particular application ready to run, but rather a framework for building applications. The XML library in this framework can validate XML code. In this case, it is not necessary to add a schema reference to the root element. In fact, it is not even necessary to copy the GML code into an XML file. It is sufficient to let the application built with *.Net Framework 4.0* read the GML code into a string, and perform an XML validation on that string. It requires a great deal of software development from the user, but once done, the user could have a tool which lets the user batch validate a large number of GMLJP2 files.

Bibliography

David S. Taubman, Michael W. Marcellin. 2002. *JPEG 2000 – Image Compression Fundamentals, Standards and Practice*, third printing, Norwell, Massachusetts: Kluwer Academic Publishers.

Peter Schelkens, Athanassios Skodras, Touradj Ebrahimi. 2009. *The JPEG 2000 Suite*, John Wiley & Sons Ltd.

STDI-0002 Appendix E: ASDE 2.1 - The Compendium of Controlled Extensions (CE) for the National Imagery Transmission Format (NITFS) 16 November 2000