
Geographic information — Well-known text representation of coordinate reference systems

Information géographique — Représentation textuelle bien lisible de systèmes de référence par coordonnées

STANDARDSISO.COM : Click to view the full PDF of ISO 19162:2019



STANDARDSISO.COM : Click to view the full PDF of ISO 19162:2019



COPYRIGHT PROTECTED DOCUMENT

© ISO 2019

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

	Page
Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms, definitions and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Abbreviated terms	10
4 Conformance requirements	10
5 Backus-Naur Form notation and syntax	11
6 WKT string form	12
6.1 Overview	12
6.2 Encoding	12
6.3 Characters used in WKT	12
6.3.1 Basic characters	12
6.3.2 Numbers	13
6.3.3 Date and time	14
6.3.4 CRS WKT characters	15
6.3.5 Double quote	15
6.4 Delimiter	15
6.5 Case sensitivity	16
6.6 Reserved keywords	16
6.7 Backward compatibility	18
7 WKT representation of common attributes	18
7.1 General	18
7.2 Name	18
7.3 Scope, extent, identifier and remark	19
7.3.1 General	19
7.3.2 Usage (scope and extent)	19
7.3.3 Identifier	22
7.3.4 Remark	23
7.4 Unit and unit conversion factor	24
7.4.1 Unit description	24
7.4.2 Conversion factor — Spatial and parametric units	25
7.4.3 Conversion factor — Temporal quantities	25
7.4.4 Default unit	26
7.5 Coordinate system	26
7.5.1 Syntax	26
7.5.2 Coordinate system type, dimension and coordinate data type	29
7.5.3 Axis name and abbreviation	29
7.5.4 Axis direction	30
7.5.5 Axis order	31
7.5.6 Axis unit and coordinate system unit	32
7.5.7 Examples of WKT describing coordinate systems	33
7.6 Datum ensemble	34
7.7 Dynamic coordinate reference systems	36
8 WKT representation of geodetic and geographic coordinate reference systems	37
8.1 Overview	37
8.2 Geodetic reference frame (geodetic datum)	38
8.2.1 Ellipsoid	38
8.2.2 Prime meridian	39
8.2.3 Geodetic reference frame (datum)	40

8.3	Coordinate systems for geodetic and geographic CRSs	41
8.4	Examples of WKT describing a geodetic or geographic CRS	42
9	WKT representation of projected CRSs	43
9.1	Overview	43
9.2	Base CRS	43
9.2.1	General	43
9.2.2	Ellipsoidal CS unit	44
9.3	Map projection	45
9.3.1	Introduction	45
9.3.2	Map projection name and identifier	46
9.3.3	Map projection method	46
9.3.4	Map projection parameter	47
9.4	Coordinate systems for projected CRSs	47
9.5	Examples of WKT describing a projected CRS	47
10	WKT representation of vertical CRSs	49
10.1	Overview	49
10.2	Vertical reference frame (vertical datum)	50
10.3	Vertical coordinate system	50
10.4	Example of WKT describing a vertical CRS	51
11	WKT representation of engineering CRSs	51
11.1	Overview	51
11.2	Engineering datum	51
11.3	Coordinate systems for engineering CRSs	52
11.4	Examples of WKT describing an engineering CRS	52
12	WKT representation of parametric CRSs	53
12.1	Overview	53
12.2	Parametric datum	53
12.3	Parametric coordinate system	53
12.4	Example of WKT describing a parametric CRS	53
13	WKT representation of temporal CRSs	54
13.1	Temporal CRS	54
13.2	Temporal datum	54
13.3	Temporal coordinate system	55
13.3.1	General	55
13.3.2	Axis unit for temporalDateTime coordinate systems	55
13.3.3	Axis unit for temporalCount and temporalMeasure coordinate systems	55
13.4	Examples of WKT describing a temporal CRS	55
14	WKT representation of derived CRSs	56
14.1	Overview	56
14.2	Deriving conversion	57
14.2.1	General	57
14.2.2	Derived CRS conversion method	57
14.2.3	Derived CRS conversion parameter	58
14.2.4	Derived CRS conversion parameter file	58
14.2.5	Derived CRS conversion example	59
14.3	Derived geodetic CRS and derived geographic CRS	59
14.3.1	Representation	59
14.3.2	Example of WKT describing a derived geographic CRS	61
14.4	Derived projected CRS	61
14.4.1	Representation	61
14.4.2	Example of WKT describing a derived projected CRS	62
14.5	Derived vertical CRS	63
14.6	Derived engineering CRS	63
14.7	Derived parametric CRS	64
14.8	Derived temporal CRS	64

15	WKT representation of compound coordinate reference systems	65
15.1	Overview	65
15.2	Examples of WKT describing a compound CRS	65
16	WKT representation of coordinate epoch and coordinate metadata	66
16.1	Coordinate epoch	66
16.2	Coordinate metadata	67
17	WKT representation of coordinate transformations and coordinate conversions excluding map projections	68
17.1	Coordinate operations	68
17.2	Transformation and conversion components	68
17.2.1	Operation name and version	68
17.2.2	Source and target CRS	68
17.2.3	Transformation and conversion name and identifier	69
17.2.4	Coordinate operation method	69
17.2.5	Coordinate operation parameter	69
17.2.6	Coordinate operation parameter file	70
17.2.7	Interpolation CRS	70
17.2.8	Coordinate operation accuracy	70
17.2.9	Other coordinate operation attributes	71
17.3	Examples of WKT describing a coordinate transformation	71
18	WKT representation of point motion operations	72
19	WKT representation of concatenated coordinate operations	73
19.1	General	73
19.2	Examples of WKT describing a concatenated coordinate operation	75
20	WKT representation of CRS and coordinate operation couplets	76
20.1	Bound CRS	76
20.2	Bound CRS components	77
20.2.1	Abridged coordinate transformation	77
20.2.2	Coordinate operation method in abridged coordinate transformations	77
20.2.3	Abridged coordinate transformation parameter	77
20.2.4	Coordinate operation parameter file	78
20.3	Examples of WKT describing a bound CRS	78
	Annex A (normative) Abstract test suite	80
	Annex B (informative) Recommended practice for implementation	89
	Annex C (informative) Mapping of concepts from previous versions of CRS WKT	92
	Annex D (informative) Backward compatibility with ISO 19162:2015	103
	Annex E (normative) Triaxial ellipsoid	107
	Annex F (informative) Identifiers for coordinate operation methods and parameters	108
	Bibliography	113

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics* in close collaboration with the Open Geospatial Consortium (OGC).

This second edition cancels and replaces the first edition (ISO 19162:2015), which has been technically revised.

The main changes compared to the previous edition are as follows:

- updates to reflect the changes made in ISO 19111:2019 from its previous edition ISO 19111:2007 to describe dynamic geodetic reference frames, three-dimensional projected coordinate reference systems, datum ensembles and coordinate metadata;
- remodelling of the descriptions of temporal coordinate reference systems, to reflect the changes made in ISO 19111:2019;
- the correction of minor errors.

Further details are given in [Annex D](#).

In accordance with the ISO/IEC Directives, Part 2, 2018, *Rules for the structure and drafting of International Standards*, in International Standards the decimal sign is a comma on the line. However, the General Conference on Weights and Measures (*Conférence Générale des Poids et Mesures*) at its meeting in 2003 passed unanimously the following resolution:

“The decimal marker shall be either a point on the line or a comma on the line.”

In practice, the choice between these alternatives depends on customary use in the language concerned. In the technical areas of geodesy and geographic information it is customary for the decimal point always to be used, for all languages. That practice is used throughout this document.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Well-known text (WKT) offers a compact machine- and human-readable representation of geometric objects. WKT may also be used for succinctly describing the critical elements of coordinate reference system (CRS) definitions.

WKT was described in the Open Geospatial Consortium implementation specifications 99-036 through 06-103r4 and ISO 19125-1:2004. The WKT representation of coordinate reference systems was subsequently extended in Open Geospatial Consortium implementation specification 01-009 "Coordinate Transformation Services" and this extension was later adopted in the Open Geospatial Consortium GeoAPI 3.0 implementation standard 09-083r3 and GeoPackage 1.0 implementation standard 12-128r10. The WKT representation of coordinate reference systems as defined in ISO 19125-1:2004 and OGC specification 01-009 is inconsistent with the terminology and technical provisions of ISO 19111:2007 and OGC Abstract Specification topic 2 (08-015r2), "Geographic information — Spatial referencing by coordinates".

The 2015 version of this document provided an updated version of WKT representation of coordinate reference systems that follows the provisions of ISO 19111:2007 and ISO 19111-2:2009. It extended earlier WKT to allow for the description of coordinate operations.

This document updates WKT for the extensions to ISO 19111 made through its 2019 revision:

- the description of dynamic geodetic and vertical coordinate reference systems;
- the change of coordinate values within a coordinate reference system due to point motion caused by tectonic deformation;
- the description of geoid-based vertical coordinate reference systems;
- the description of datum ensembles, groups of realizations of one terrestrial or vertical reference system that for low accuracy purposes may be merged ignoring coordinate transformation;
- a rigorous description of temporal coordinate reference systems;
- the removal (deprecation) of image coordinate reference systems; and
- the remodelling of scope and extent information.

This document defines the structure and content of well-known text strings. It does not prescribe how implementations should read or write these strings.

[STANDARDSISO.COM](https://standardsiso.com) : Click to view the full PDF of ISO 19162:2019

Geographic information — Well-known text representation of coordinate reference systems

1 Scope

This document defines the structure and content of a text string implementation of the abstract model for coordinate reference systems described in ISO 19111. The string defines frequently needed types of coordinate reference systems and coordinate operations in a self-contained form that is easily readable by machines and by humans. The essence is its simplicity; as a consequence there are some constraints upon the more open content allowed in ISO 19111. To retain simplicity in the well-known text (WKT) description of coordinate reference systems and coordinate operations, the scope of this document excludes parameter grouping and pass-through coordinate operations. The text string provides a means for humans and machines to correctly and unambiguously interpret and utilise a coordinate reference system definition with look-ups or cross references only to define coordinate operation mathematics. A WKT string is not suitable for the storage of definitions of coordinate reference systems or coordinate operations because it omits metadata about the source of the data and may omit metadata about the applicability of the information.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

ISO 19111:2019, *Geographic information — Referencing by coordinates*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

3.1.1

affine coordinate system

coordinate system in Euclidean space with straight axes that are not necessarily mutually perpendicular

[SOURCE: ISO 19111:2019, 3.1.1]

3.1.2

bearing

<geodesy> horizontal angle at a point relative to a specified direction

Note 1 to entry: The direction is usually specified to be north. In some communities the term bearing refers specifically to grid north and directions relative to true north are then termed 'azimuth'; in other communities a bearing refers specifically to true north. In this document bearing is used for any specified reference direction. The angle may be reckoned positive clockwise or positive counter-clockwise depending upon the application.

3.1.3

Cartesian coordinate system

coordinate system in Euclidean space which gives the position of points relative to n mutually perpendicular straight axes all having the same unit of measure

Note 1 to entry: n is 2 or 3 for the purposes of this document.

Note 2 to entry: A Cartesian coordinate system is a specialisation of an affine coordinate system.

[SOURCE: ISO 19111:2019, 3.1.2]

3.1.4

compound coordinate reference system

coordinate reference system using at least two independent coordinate reference systems

Note 1 to entry: Coordinate reference systems are independent of each other if coordinate values in one cannot be converted or transformed into coordinate values in the other.

[SOURCE: ISO 19111:2019, 3.1.3]

3.1.5

coordinate conversion

coordinate operation that changes coordinates in a source coordinate reference system to coordinates in a target coordinate reference system in which both coordinate reference systems are based on the same datum

Note 1 to entry: A coordinate conversion uses parameters which have specified values.

EXAMPLE 1 A mapping of ellipsoidal coordinates to Cartesian coordinates using a map projection.

EXAMPLE 2 Change of units such as from radians to degrees or from feet to metres.

[SOURCE: ISO 19111:2019, 3.1.6]

3.1.6

coordinate epoch

epoch to which coordinates in a dynamic coordinate reference system are referenced

[SOURCE: ISO 19111:2019, 3.1.7]

3.1.7

coordinate operation

process using a mathematical model, based on a one-to-one relationship, that changes coordinates in a source coordinate reference system to coordinates in a target coordinate reference system, or that changes coordinates at a source coordinate epoch to coordinates at a target coordinate epoch within the same coordinate reference system

[SOURCE: ISO 19111:2019, 3.1.8]

3.1.8

coordinate reference system

coordinate system that is related to an object by a datum

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

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

[SOURCE: ISO 19111:2019, 3.1.9]

3.1.9

coordinate system

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

[SOURCE: ISO 19111:2019, 3.1.11]

3.1.10

coordinate transformation

coordinate operation that changes coordinates in a source coordinate reference system to coordinates in a target coordinate reference system in which the source and target coordinate reference systems are based on different datums

Note 1 to entry: A coordinate transformation uses parameters which are derived empirically. Any error in those coordinates will be embedded in the coordinate transformation and when the coordinate transformation is applied the embedded errors are transmitted to output coordinates.

Note 2 to entry: A coordinate transformation is colloquially sometimes referred to as a 'datum transformation'. This is erroneous. A coordinate transformation changes coordinate values. It does not change the definition of the datum. In this document coordinates are referenced to a coordinate reference system. A coordinate transformation operates between two coordinate reference systems, not between two datums.

[SOURCE: ISO 19111:2019, 3.1.12]

3.1.11

cylindrical coordinate system

three-dimensional coordinate system in Euclidean space in which position is specified by two linear coordinates and one angular coordinate

[SOURCE: ISO 19111:2019, 3.1.14]

3.1.12

datum

reference frame

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

[SOURCE: ISO 19111:2019, 3.1.15]

3.1.13

datum ensemble

group of multiple realizations of the same terrestrial or vertical reference system that, for approximate spatial referencing purposes, are not significantly different

Note 1 to entry: Datasets referenced to the different realizations within a datum ensemble may be merged without coordinate transformation.

Note 2 to entry: 'Approximate' is for users to define but typically is in the order of under 1 decimetre but may be up to 2 metres.

EXAMPLE "WGS 84" as an undifferentiated group of realizations including WGS 84 (TRANSIT), WGS 84 (G730), WGS 84 (G873), WGS 84 (G1150), WGS 84 (G1674) and WGS 84 (G1762). At the surface of the Earth these have changed on average by 0.7 m between the TRANSIT and G730 realizations, a further 0.2 m between G730 and G873, 0.06 m between G873 and G1150, 0.2 m between G1150 and G1674 and 0.02 m between G1674 and G1762.

[SOURCE: ISO 19111:2019, 3.1.16]

3.1.14

derived coordinate reference system

coordinate reference system that is defined through the application of a specified coordinate conversion to the coordinates within a previously established coordinate reference system

Note 1 to entry: The previously established coordinate reference system is referred to as the base coordinate reference system.

Note 2 to entry: A derived coordinate reference system inherits its datum or reference frame from its base coordinate reference system.

Note 3 to entry: The coordinate conversion between the base and derived coordinate reference system is implemented using the parameters and formula(s) specified in the definition of the coordinate conversion

[SOURCE: ISO 19111:2019, 3.1.8]

3.1.15

dynamic coordinate reference system

coordinate reference system that has a dynamic reference frame

Note 1 to entry: Coordinates of points on or near the crust of the Earth that are referenced to a dynamic coordinate reference system may change with time, usually due to crustal deformations such as tectonic motion and glacial isostatic adjustment.

Note 2 to entry: Metadata for a dataset referenced to a dynamic coordinate reference system should include coordinate epoch information.

[SOURCE: ISO 19111:2019, 3.1.9]

3.1.16

dynamic reference frame

dynamic datum reference frame in which the defining parameters include time evolution

Note 1 to entry: The defining parameters that have time evolution are usually a coordinate set.

[SOURCE: ISO 19111:2019, 3.1.20]

3.1.17

ellipsoid

reference ellipsoid

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

Note 1 to entry: For the Earth the ellipsoid is bi-axial with rotation about the polar axis. This results in an oblate ellipsoid with the midpoint of the foci located at the nominal centre of the Earth.

[SOURCE: ISO 19111:2019, 3.1.22]

3.1.18

ellipsoidal coordinate system

geodetic coordinate system

coordinate system in which position is specified by geodetic latitude, geodetic longitude and (in the three-dimensional case) ellipsoidal height

[SOURCE: ISO 19111:2019, 3.1.23]

3.1.19**ellipsoidal height**

geodetic height

h

distance of a point from the reference ellipsoid along the perpendicular from the reference ellipsoid to this point, positive if upwards or outside of the reference ellipsoid

Note 1 to entry: Only used as part of a three-dimensional ellipsoidal coordinate system or as part of a three-dimensional Cartesian coordinate system in a three-dimensional projected coordinate reference system, but never on its own.

[SOURCE: ISO 19111:2019, 3.1.24]

3.1.20**engineering coordinate reference system**

coordinate reference system based on an engineering datum

EXAMPLE 1 System for identifying relative positions within a few kilometres of the reference point, such as a building or construction site.

EXAMPLE 2 Coordinate reference system local to a moving object such as a ship or an orbiting spacecraft.

EXAMPLE 3 Internal coordinate reference system for an image. This has continuous axes. It may be the foundation for a grid.

[SOURCE: ISO 19111:2019, 3.1.25]

3.1.21**engineering datum**

local datum

datum describing the relationship of a coordinate system to a local reference

Note 1 to entry: Engineering datum excludes both geodetic and vertical reference frames.

[SOURCE: ISO 19111:2019, 3.1.26]

3.1.22**epoch**

<geodesy> point in time

Note 1 to entry: In this document an epoch is expressed in the Gregorian calendar as a decimal year.

EXAMPLE 2017-03-25 in the Gregorian calendar is epoch 2017,23.

[SOURCE: ISO 19111:2019, 3.1.27]

3.1.23**flattening***f*

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

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

[SOURCE: ISO 19111:2019, 3.1.28]

3.1.24**frame reference epoch**

epoch of coordinates that define a dynamic reference frame

[SOURCE: ISO 19111:2019, 3.1.29]

3.1.25

geodetic coordinate reference system

three-dimensional coordinate reference system based on a geodetic reference frame and having either a three-dimensional Cartesian or a spherical coordinate system

Note 1 to entry: In this document a coordinate reference system based on a geodetic reference frame and having an ellipsoidal coordinate system is geographic.

[SOURCE: ISO 19111:2019, 3.1.31]

3.1.26

geodetic latitude

ellipsoidal latitude

φ

angle from the equatorial plane to the perpendicular to the ellipsoid through a given point, northwards treated as positive

[SOURCE: ISO 19111:2019, 3.1.32]

3.1.27

geodetic longitude

ellipsoidal longitude

λ

angle from the prime meridian plane to the meridian plane of a given point, eastward treated as positive

[SOURCE: ISO 19111:2019, 3.1.33]

3.1.28

geodetic reference frame

reference frame or datum describing the relationship of a two- or three-dimensional coordinate system to the Earth

Note 1 to entry: In the data model described in this document, the UML class GeodeticReferenceFrame includes both modern terrestrial reference frames and classical geodetic datums.

[SOURCE: ISO 19111:2019, 3.1.34]

3.1.29

geographic coordinate reference system

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

[SOURCE: ISO 19111:2019, 3.1.35]

3.1.30

linear coordinate system

one-dimensional coordinate system in which a linear feature forms the axis

EXAMPLE 1 Distances along a pipeline.

EXAMPLE 2 Depths down a deviated oil well bore.

[SOURCE: ISO 19111:2019, 3.1.39]

3.1.31

map projection

coordinate conversion from an ellipsoidal coordinate system to a plane

[SOURCE: ISO 19111:2019, 3.1.40]

3.1.32**parametric coordinate reference system**

coordinate reference system based on a parametric datum

[SOURCE: ISO 19111:2019, 3.1.45]

3.1.33**parametric coordinate system**

one-dimensional coordinate system where the axis units are parameter values which are not inherently spatial

[SOURCE: ISO 19111:2019, 3.1.46]

3.1.34**parametric datum**

datum describing the relationship of a parametric coordinate system to an object

Note 1 to entry: The object is normally the Earth.

[SOURCE: ISO 19111:2019, 3.1.47]

3.1.35**point motion operation**

coordinate operation that changes coordinates within one coordinate reference system due to the motion of the point

Note 1 to entry: The change of coordinates is from those at an initial epoch to those at another epoch.

Note 2 to entry: In this document the point motion is due to tectonic motion or crustal deformation.

[SOURCE: ISO 19111:2019, 3.1.48]

3.1.36**polar coordinate system**

two-dimensional coordinate system in Euclidean space in which position is specified by one distance coordinate and one angular coordinate

Note 1 to entry: For the three-dimensional case, see spherical coordinate system.

[SOURCE: ISO 19111:2019, 3.1.49]

3.1.37**prime meridian**

meridian from which the longitudes of other meridians are quantified

[SOURCE: ISO 19111:2019, 3.1.50]

3.1.38**projected coordinate reference system**

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

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

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

[SOURCE: ISO 19111:2019, 3.1.51]

**3.1.39
reference frame**

datum

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

[SOURCE: ISO 19111:2019, 3.1.52]

**3.1.40
semi-major axis**

a

semi-diameter of the longest axis of an ellipsoid

[SOURCE: ISO 19111:2019, 3.1.53]

**3.1.41
semi-minor axis**

b

semi-diameter of the shortest axis of an ellipsoid

[SOURCE: ISO 19111:2019, 3.1.54]

**3.1.42
spatio-parametric coordinate reference system**

compound coordinate reference system in which one constituent coordinate reference system is a spatial coordinate reference system and one is a parametric coordinate reference system

Note 1 to entry: Normally the spatial component is “horizontal” and the parametric component is “vertical”.

[SOURCE: ISO 19111:2019, 3.1.57]

**3.1.43
spatio-parametric-temporal coordinate reference system**

compound coordinate reference system comprised of spatial, parametric and temporal coordinate reference systems

[SOURCE: ISO 19111:2019, 3.1.58]

**3.1.44
spatio-temporal coordinate reference system**

compound coordinate reference system in which one constituent coordinate reference system is a spatial coordinate reference system and one is a temporal coordinate reference system

[SOURCE: ISO 19111:2019, 3.1.59]

**3.1.45
spherical coordinate system**

three-dimensional coordinate system in Euclidean space in which position is specified by one distance coordinate and two angular coordinates

Note 1 to entry: Not to be confused with an ellipsoidal coordinate system based on an ellipsoid ‘degenerated’ into a sphere.

[SOURCE: ISO 19111:2019, 3.1.60]

**3.1.46
spheroid**

closed surface that differs only slightly from that of a sphere

3.1.47**static coordinate reference system**

coordinate reference system that has a static reference frame

Note 1 to entry: Coordinates of points on or near the crust of the Earth that are referenced to a static coordinate reference system do not change with time.

Note 2 to entry: Metadata for a dataset referenced to a static coordinate reference system does not require coordinate epoch information.

[SOURCE: ISO 19111:2019, 3.1.61]

3.1.48**static reference frame**

static datum

reference frame in which the defining parameters exclude time evolution

[SOURCE: ISO 19111:2019, 3.1.62]

3.1.49**temporal coordinate reference system**

coordinate reference system based on a temporal datum

[SOURCE: ISO 19111:2019, 3.1.63]

3.1.50**temporal coordinate system**

<geodesy> one-dimensional coordinate system where the axis is time

[SOURCE: ISO 19111:2019, 3.1.64]

3.1.51**temporal datum**

datum describing the relationship of a temporal coordinate system to an object

Note 1 to entry: The object is normally time on the Earth.

[SOURCE: ISO 19111:2019, 3.1.65]

3.1.52**vertical coordinate reference system**

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

[SOURCE: ISO 19111:2019, 3.1.70]

3.1.53**vertical coordinate system**

one-dimensional coordinate system used for gravity-related height or depth measurements

[SOURCE: ISO 19111:2019, 3.1.71]

3.1.54**vertical reference frame**

vertical datum

reference frame describing the relation of gravity-related heights or depths to the Earth

Note 1 to entry: In most cases, the vertical reference frame will be related to mean sea level. Vertical datums include sounding datums (used for hydrographic purposes), in which case the heights may be negative heights or depths.

Note 2 to entry: Ellipsoidal heights are related to a three-dimensional ellipsoidal coordinate system referenced to a geodetic reference frame.

[SOURCE: ISO 19111:2019, 3.1.72]

3.1.55

white space

sequence of one or more characters that have no glyphs

[SOURCE: ISO/IEC 9075-2:2016, 3.1.6.77]

3.2 Abbreviated terms

BNF Backus-Naur form

CRS coordinate reference system

CS coordinate system

EPSG European Petroleum Survey Group (geodetic parameter dataset now maintained at www.epsg-registry.org by the International Association of Oil and Gas Producers)

IRM international reference meridian

OGC Open Geospatial Consortium, www.opengeospatial.org

TRF Terrestrial Reference Frame

UTC Coordinated Universal Time

VRF Vertical Reference Frame

WKT Well-known text

4 Conformance requirements

This document defines eighteen classes of conformance (see [Annex A](#)):

- a) Any WKT string claiming conformance of a coordinate reference system definition shall satisfy the requirements in [Annex A](#) as shown in [Table 1](#).

Table 1 — Conformance requirements for coordinate reference systems

Coordinate reference system type	Conformance requirements given in
geodetic geographic	A.1
projected	A.2
vertical	A.3
engineering	A.4
parametric	A.5
temporal	A.6
derived geodetic derived geographic	A.7
derived projected	A.8
derived vertical	A.9
derived engineering	A.10
derived parametric	A.11
derived temporal	A.12
compound	A.13

- b) Any WKT string claiming conformance of coordinate metadata shall satisfy the requirements given in [A.14](#).
- c) Any WKT string claiming conformance of a coordinate operation definition shall satisfy the requirements in [Annex A](#) as shown in [Table 2](#).

Table 2 — Conformance requirements for coordinate operations

Coordinate operation type	Conformance requirements given in
coordinate transformation	A.15
coordinate conversion other than a deriving conversion	
deriving conversion	Within derived CRS in A.2 and A.7 to A.12
point motion operation	A.16
concatenated coordinate operation	A.17

- d) Any WKT string claiming conformance of coordinate transformation bound to a coordinate reference system definition shall satisfy the requirements given in [A.18](#).

Conformance is applicable to the WKT string. Recommended practices for implementations writing or reading coordinate reference system WKT strings are given in [Annex B](#).

5 Backus-Naur Form notation and syntax

The WKT representation of coordinate reference systems and coordinate operations is defined in this document using the extended version of Backus-Naur form (BNF) notation that is defined in ISO/IEC 9075-1:2016, 6.2. The BNF provides the mechanism for generating a WKT string. The production rules in ISO/IEC 9075-1:2016, 6.2 apply.

In this extended version of BNF the characters have the meaning:

- A character string enclosed in angle brackets "< >" is a syntactic element.
- A vertical bar "|" indicates alternatives.
- Braces "{" }" define a group of elements.
- Square brackets "[]" denote optional elements. This use of square brackets within BNF notation should not be confused with the use of square brackets as delimiters in WKT strings.
- Ellipsis after an element "< >..." allows the use of one or multiple instances of that element.
- Ellipsis after braces "{ }..." allows the use of one or multiple instances of of the group of elements.
- Ellipsis after square brackets "[]..." means that the content inside the square brackets may occur zero to many times.
- Double exclamation marks "!!" introduce normal English text. This is used when the definition of a syntactic element is supplemented by constraints not in the BNF definition but given later in the text.

In the BNF notation spaces are used to separate syntactic elements. Multiple spaces and line breaks are treated as a single space. These spaces do not form part of the resulting WKT string. All other characters in the BNF stand for themselves. The order of syntactic elements in the BNF is significant; it defines the format of the WKT string.

6 WKT string form

6.1 Overview

The WKT string is a representation of the definition of a CRS or coordinate operation. A string describes one CRS or coordinate operation object. Each object is represented by a token comprised of a keyword followed by a set of attributes of the object, the set enclosed by delimiters. Some objects are composed of other objects so the result may be a nested structure. Nesting may continue to any depth.

EXAMPLE KEYWORD1[attribute1,KEYWORD2[attribute2,attribute3]]

Keywords are case-insensitive. Where human readability of the string is important, as in this document, keywords are normally in upper case.

The delimiters are normally <left bracket> and <right bracket>. Implementations are free to substitute parentheses for brackets.

Attributes may be from an enumeration, may be numbers or may be text. Text is enclosed in double quotes. Two forms of text are defined, one restricted to the Latin1 character set and the other permitting any Unicode character set. Attributes are separated by a comma.

A WKT string contains no white space outside of double quotes. However, padding with white space to improve human readability is permitted; the examples of WKT that are included in this document have spaces and line feeds inserted to improve clarity. Any padding is stripped out or ignored by parsers, refer to [Annex B](#).

6.2 Encoding

All WKT strings are realized as a sequence of characters, or a character string. It is not the goal of this document to specify any encoding used in a given implementation. The only restriction is that the same encoding shall be used throughout the entire WKT definition.

Requirements:

- a) A WKT string shall use one encoding throughout the entire string. UTF-8 shall be used if no encoding is specified in the carrier.
- b) The characters used in a WKT string shall be wholly contained within the domain of a specific character set. This character set shall exist as a subset of the repertoire of the Universal Character Set specified by ISO/IEC 10646.

6.3 Characters used in WKT

6.3.1 Basic characters

The basic characters in a CRS WKT string are taken directly from ISO/IEC 9075-2:2016, 5.1 and 5.3.

<simple Latin upper case letter>	::=	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z <i>!! ISO/IEC 10646:2012 character identifiers U+0041 through U+005A</i>
<simple Latin lower case letter>	::=	a b c d e f g h i j k l m n o p q r s t u v w x y z <i>!! ISO/IEC 10646:2012 character identifiers U+0061 through U+007A</i>
<digit>	::=	0 1 2 3 4 5 6 7 8 9 <i>!! ISO/IEC 10646:2012 character identifiers U+0030 through U+0039</i>

<space>	::=	SP	!! ISO/IEC 10646:2012 character identifier U+0020
<double quote>	::=	"	!! ISO/IEC 10646:2012 character identifier U+0022
<number sign>	::=	#	!! ISO/IEC 10646:2012 character identifier U+0023
<percent>	::=	%	!! ISO/IEC 10646:2012 character identifier U+0025
<ampersand>	::=	&	!! ISO/IEC 10646:2012 character identifier U+0026
<quote>	::=	'	!! ISO/IEC 10646:2012 character identifier U+0027
<left paren>	::=	(!! ISO/IEC 10646:2012 character identifier U+0028
<right paren>	::=)	!! ISO/IEC 10646:2012 character identifier U+0029
<asterisk>	::=	*	!! ISO/IEC 10646:2012 character identifier U+002A
<plus sign>	::=	+	!! ISO/IEC 10646:2012 character identifier U+002B
<comma>	::=	,	!! ISO/IEC 10646:2012 character identifier U+002C
<minus sign> ::- <hyphen>	::=	-	!! ISO/IEC 10646:2012 character identifier U+002D
<period>	::=	.	!! ISO/IEC 10646:2012 character identifier U+002E
<solidus>	::=	/	!! ISO/IEC 10646:2012 character identifier U+002F
<colon>	::=	:	!! ISO/IEC 10646:2012 character identifier U+003A
<semicolon>	::=	;	!! ISO/IEC 10646:2012 character identifier U+003B
<less than operator>	::=	<	!! ISO/IEC 10646:2012 character identifier U+003C
<equals operator>	::=	=	!! ISO/IEC 10646:2012 character identifier U+003D
<greater than operator>	::=	>	!! ISO/IEC 10646:2012 character identifier U+003E
<question mark>	::=	?	!! ISO/IEC 10646:2012 character identifier U+003F
<left bracket>	::=	[!! ISO/IEC 10646:2012 character identifier U+005B
<reverse solidus>	::=	\	!! ISO/IEC 10646:2012 character identifier U+005C
<right bracket>	::=]	!! ISO/IEC 10646:2012 character identifier U+005D
<circumflex>	::=	^	!! ISO/IEC 10646:2012 character identifier U+005E
<underscore>	::=	_	!! ISO/IEC 10646:2012 character identifier U+005F
<left brace>	::=	{	!! ISO/IEC 10646:2012 character identifier U+007B
<vertical bar>	::=		!! ISO/IEC 10646:2012 character identifier U+007C
<right brace>	::=	}	!! ISO/IEC 10646:2012 character identifier U+007D
<degree symbol>	::=	°	!! ISO/IEC 10646:2012 character identifier U+00B0

6.3.2 Numbers

<number>	::=	<signed numeric literal> <unsigned numeric literal>
<signed numeric literal>	::=	[<sign>] <unsigned numeric literal>
<unsigned numeric literal>	::=	<exact numeric literal> <approximate numeric literal>
<approximate numeric literal>	::=	<mantissa> E <exponent>
<mantissa>	::=	<exact numeric literal>
<exponent>	::=	<signed integer>
<signed integer>	::=	[<sign>] <unsigned integer>
<exact numeric literal>	::=	<unsigned integer> [<period> [<unsigned integer>]] <period> <unsigned integer>
<unsigned integer>	::=	<digit>...
<sign>	::=	<plus sign> <minus sign>

The integer and fractional parts of a number are separated by a <period>; a comma is not permitted. No other separator (e.g. for thousands or multiples thereof) is allowed.

6.3.3 Date and time

In this document calendar dates and times are restricted to the Gregorian calendar and the 24-hour clock as defined in ISO 8601-1. Only the ISO 8601-1 extended format (separators between date units and between sexagesimal time units) is permitted. Any precision is allowed. When time is included a UTC or local time zone designator is required. Other date formats such as geological eras or calendars other than Gregorian may be stated through a free format quoted text string.

<datetime>	::=	<Gregorian calendar datetime> <Gregorian ordinal datetime>
<Gregorian calendar datetime>	::=	<Gregorian calendar date> [<24 hour clock>]
<Gregorian calendar date>	::=	<year> [<hyphen> <month> [<hyphen> <day>]]
<year>	::=	<unsigned integer> <i>!! four digits</i>
<month>	::=	<unsigned integer> <i>!! two digits including leading zero if less than 10</i>
<day>	::=	<unsigned integer> <i>!! two digits including leading zero if less than 10</i>
<Gregorian ordinal datetime>	::=	<Gregorian ordinal date> [<24 hour clock>]
<Gregorian ordinal date>	::=	<year> [<hyphen> <ordinal day>]
<ordinal day>	::=	<unsigned integer> <i>!! three digits including leading zeroes if less than 100</i>
<24 hour clock>	::=	<time designator> <hour> [<colon> <minute> [<colon> <second>]] <time zone designator>
<time designator>	::=	T
<hour>	::=	<unsigned integer> <i>!! two digits including leading zero if less than 10</i>
<minute>	::=	<unsigned integer> <i>!! two digits including leading zero if less than 10</i>
<second>	::=	<seconds integer> [<period> [<seconds fraction>]] <i>!! In this document the separator between the integer and fractional parts of a second value shall be a period. The ISO 8601 preference for comma is not permitted.</i>
<seconds integer>	::=	<unsigned integer> <i>!! two digits including leading zero if less than 10</i>
<seconds fraction>	::=	<unsigned integer>
<time zone designator>	::=	<utc designator> <local time zone designator>
<utc designator>	::=	Z
<local time zone designator>	::=	{ <plus sign> <minus sign> } <hour> [<colon> <minute>]

In ISO 8601-1 time zone designator may be omitted, in which case local time is assumed but unspecified. In this document a time zone designator is mandatory.

EXAMPLE 1	2014	(Precision = year)
EXAMPLE 2	2014-01	(Calendar date, precision = month)
EXAMPLE 3	2014-03-01	(Calendar date, precision = day)
EXAMPLE 4	2014-060	(Ordinal date, precision = day)
EXAMPLE 5	2014-05-06T23Z	(Calendar date, precision = hour, UTC)
EXAMPLE 6	2014-157T23Z	(Ordinal date, precision = hour, UTC)
EXAMPLE 7	2014-07-12T16:00Z	(Precision = minute, UTC)
EXAMPLE 8	2014-07-12T17:00+01	(Precision = minute, local time one hour ahead of UTC)

EXAMPLE 9	2014-09-18T08:17:56-08	(Precision = second, local time eight hours behind UTC)
EXAMPLE 10	2014-11-23T00:34:56.789Z	(Precision = 1/1 000 second, UTC)

6.3.4 CRS WKT characters

The characters permitted in elements of a CRS WKT string formed from those in [6.3.1](#), [6.3.2](#) and [6.3.3](#) are:

<left delimiter>	::=	<left bracket> <left paren> <i>!! In this document the preferred left delimiter is <left bracket>. <left paren> is permitted for backward compatibility. Implementations shall be able to read both forms.</i>
<right delimiter>	::=	<right bracket> <right paren> <i>!! See 6.4.</i>
<wkt separator>	::=	<comma>
<quoted Latin text>	::=	<double quote> <wkt Latin text character>... <double quote>
<wkt Latin text character>	::=	<simple Latin upper case letter> <simple Latin lower case letter> <digit> <underscore> <left bracket> <right bracket> <left paren> <right paren> <left brace> <right brace> <less than operator> <equals operator> <greater than operator> <period> <comma> <colon> <semicolon> <plus sign> <minus sign> <space> <number sign> <percent> <ampersand> <quote> <asterisk> <circumflex> <solidus> <reverse solidus> <question mark> <vertical bar> <degree symbol> <doublequote symbol>
<quoted Unicode text>	::=	<double quote> { <wkt Unicode text character> }... <double quote>
<wkt Unicode text character>	::=	<nondoublequote character> <doublequote symbol>
<nondoublequote character>	::=	<i>!! A <nondoublequote character> is any character of the source language character set other than a <double quote>.</i>
<doublequote symbol>	::=	<i>"" !! two consecutive <double quote> characters</i>

6.3.5 Double quote

Requirement: If a double quote is required within a <quoted Latin text> or <quoted Unicode text> string, a <doublequote symbol> shall be used.

EXAMPLE "Datum origin is 30°25'20""N, 130°25'20""E."

6.4 Delimiter

In WKT strings the attributes for an item are included within a pair of left and right delimiters. The delimiters should normally be <left bracket> and <right bracket>. Implementations are free to substitute parentheses for brackets. If <left bracket> is used as a <left delimiter> then <right bracket> is required to be used as the corresponding <right delimiter>; if <left paren> is used as a <left delimiter> then <right paren> is required to be used as the corresponding <right delimiter>. A nested token is required to use the same type of delimiter as the token in which it is nested.

Requirement: A CRS WKT string shall include only one of the two forms of delimiter, brackets or parentheses, allowed in [6.3.4](#).

6.5 Case sensitivity

WKT keywords are case insensitive: KEYWORD, keyword, KeyWord and KEYwORd are all equivalent. Where human readability is important (as in the examples in this document) keywords should be written in only the <simple Latin upper case letter> set. KEYWORD is not equivalent to KEY_WORD; the underscore character is significant.

WKT enumerations are case insensitive. CARTESIAN is equivalent to cartesian is equivalent to Cartesian; NORTH is equivalent to north is equivalent to North.

Within quoted text, case is significant. “H” is not equivalent to “h”.

Requirement: Outside of quoted text, characters in a WKT string shall not be case sensitive.

6.6 Reserved keywords

The keywords defined in [Clauses 7](#) to [20](#) and [Annex E](#) of this document for coordinate reference systems, coordinate operations, their component elements and their attributes are summarised here. Those formed from multiple words or abbreviations are given in camel case to aid understanding; see [6.5](#) for implementation in a WKT string.

Keyword	BNF element	Clause in which defined
abridgedTransformation	<abridged transformation keyword>	20
anchor	<datum anchor keyword>	8 , 10 , 11 and 12
angleUnit	<angle unit keyword>	7
area	<area description keyword>	7
axis	<axis keyword>	7
baseEngCRS	<base engineering crs keyword>	14
baseGeodCRS	<base geodetic crs keyword>	9 and 14
baseGeogCRS	<base geographic crs keyword>	9 and 14
baseParamCRS	<base parametric crs keyword>	14
baseProjCRS	<base projected crs keyword>	14
baseTimeCRS	<base temporal crs keyword>	14
baseVertCRS	<base vertical crs keyword>	14
bBox	<geographic bounding box keyword>	7
bearing	<bearing keyword>	7
boundCRS	<bound crs keyword>	20
calendar	<calendar keyword>	13
citation	<citation keyword>	7
compoundCRS	<compound crs keyword>	15
concatenatedOperation	<concatenated operation keyword>	19
conversion	<map projection keyword>	9
coordEpoch	Alternative keyword for coordinate epoch	16
coordinateMetadata	<coordinate metadata keyword>	16
coordinateOperation	<operation keyword>	17
cs	<cs keyword> (coordinate system)	7
datum	<geodetic reference frame keyword>	8
derivedProjCRS	<derived projected crs keyword>	14
derivingConversion	<deriving conversion keyword>	14
dynamic	<dynamic crs keyword>	7
eDatum	<engineering datum keyword>	11

Keyword	BNF element	Clause in which defined
ellipsoid	<ellipsoid keyword>	8
engCRS	<engineering crs keyword>	11 and 14
engineeringCRS	Alternative keyword for engineering crs	11 and 14
engineeringDatum	Alternative keyword for engineering datum	11
ensemble	<datum ensemble keyword>	7
ensembleAccuracy	<datum ensemble accuracy keyword>	7
epoch	<coordinate epoch keyword>	16
frameEpoch	<frame reference epoch keyword>	7
geodCRS	<geodetic crs keyword>	8 and 14
geodeticCRS	Alternative keyword for geodetic crs	8 and 14
geodeticDatum	Alternative keyword for geodetic reference frame	8
geogCRS	<geographic crs keyword>	8 and 14
geographicCRS	Alternative keyword for geographic crs	8 and 14
geoidModel	<geoid model ID keyword>	10
id	<identifier keyword>	7
interpolationCRS	<interpolation crs keyword>	17
lengthUnit	<length unit keyword>	7
member	<datum ensemble member keyword>	7
meridian	<meridian keyword>	7
method	(i) <map projection method keyword> (ii) <operation method keyword>	9 14 , 17 and 18
model	<deformation model ID keyword>	7
operationAccuracy	<operation accuracy keyword>	17 and 18
order	<axis order keyword>	7
parameter	<parameter keyword>	9 , 14 , 17 and 18
parameterFile	<parameter file keyword>	14 , 17 and 18
parametricCRS	<parametric crs keyword>	12 and 14
parametricDatum	Alternative keyword for parametric datum	12
parametricUnit	<parametric unit keyword>	7
pDatum	<parametric datum keyword>	12
pointMotionOperation	<point motion operation keyword>	18
primeM	<prime meridian keyword>	8
primeMeridian	Alternative keyword for prime meridian	8
projCRS	<projected crs keyword>	9
projectedCRS	Alternative keyword for projected crs	9
projection	Alternative keyword for map projection method	9
remark	<remark keyword>	7
scaleUnit	<scale unit keyword>	7
scope	<scope keyword>	7
sourceCRS	<source crs keyword>	17 and 18
spheroid	Alternative keyword for ellipsoid	8
step	concatenated operation <step keyword>	19
targetCRS	<target crs keyword>	17
tDatum	<temporal datum keyword>	13
TRF	Alternative keyword for geodetic reference frame	8

Keyword	BNF element	Clause in which defined
temporalQuantity	Alternative keyword for time unit	7
timeCRS	<temporal crs keyword>	13 and 14
timeDatum	Alternative keyword for temporal datum	13
timeExtent	<temporal extent keyword>	7
timeOrigin	<temporal origin keyword>	13
timeUnit	<temporal unit keyword>	7
triaxial	<triaxial ellipsoid keyword>	Annex E
unit	Alternative keyword for all units	7
uri	<uri keyword>	7
usage	<usage keyword>	7
vDatum	<vertical reference frame keyword>	10
velocityGrid	Alternative keyword for deformation model ID	7
version	<operation version keyword>	17-20
vertCRS	<vertical crs keyword>	10 and 14
verticalCRS	Alternative keyword for vertical crs	10 and 14
verticalDatum	Alternative keyword for vertical reference frame	10
verticalExtent	<vertical extent keyword>	7
VRF	Alternative keyword for vertical reference frame	10

6.7 Backward compatibility

This document makes several references to backward compatibility. Backward compatibility means that an implementation of the text strings in this document would be able to read CRS WKT strings conforming to the old (ISO 19125-1:2004) syntax. It does not mean that a parser of a string compliant to ISO 19125-1:2004 could read WKT strings written in conformance with this document. It also does not require an implementation of the text strings in this document to be able to output an object according to the old syntax. [B.8](#) gives guidance on determining the version of a CRS WKT string. A mapping of older syntaxes to this document is given in [Annex C](#). Revisions to ISO 19162:2015 made in this document are described in [Annex D](#).

7 WKT representation of common attributes

7.1 General

The WKT representation of attributes that are common to coordinate reference systems and coordinate operations — name, scope, extent, identifier and remarks — is described in [7.2](#) to [7.4](#). Coordinate system (a component of coordinate reference systems) is described in [7.5](#). Elements for a datum ensemble and a dynamic coordinate reference system are described in [7.6](#) and [7.7](#) respectively. Attributes specific to coordinate reference systems and to coordinate operations are described in [Clauses 8](#) to [19](#).

7.2 Name

From a computational perspective name is redundant information provided for human readability. For example, an ellipsoid is defined by the values of the semi-major axis and inverse flattening, and the ellipsoid name is redundant information. However, the name may be used to verify the defining values against those from an authoritative source. Comment on the comparison of text strings is given in [Annex B](#).

NOTE Name is a required attribute for the CRS object as well as for some component objects. Depending upon the convention adopted for CRS naming, this can result in duplication of name within the CRS WKT string.

7.3 Scope, extent, identifier and remark

7.3.1 General

The BNF element <scope extent identifier remark> is a collection of four optional attributes which may be applied to a coordinate reference system, a coordinate operation or a bound CRS. The <scope extent identifier remark> collection is to simplify the BNF through grouping. Usage (<scope> and <extent>) is described in [7.3.2](#), <identifier> in [7.3.3](#) and <remark> in [7.3.4](#).

Requirement: The WKT representation of a <scope extent identifier remark> element shall be:

```
<scope extent identifier remark> ::= [ { <wkt separator> <usage> } ]...
                                   [ { <wkt separator> <identifier> } ]...
                                   [ { <wkt separator> <remark> } ]
```

7.3.2 Usage (scope and extent)

7.3.2.1 Usage

Usage is an optional attribute which if included in a WKT string shall include both <scope> and <extent>. Multiple pairs of scope/extent may be used to describe the usage for different purposes over different extents. In this document the <scope> and <extent> elements may not be given alone but only as a pairing. Within each pairing, extent may consist of one or more of area textual description, area bounding box, vertical extent and/or temporal extent, see [7.3.2.3](#).

Requirement: The WKT representation of a <usage> element shall be:

```
<usage> ::= <usage keyword> <left delimiter>
           <scope> <wkt separator> <extent>
           <right delimiter>
<usage keyword> ::= USAGE
```

EXAMPLE 1 Structure of one scope-extent pairing

```
USAGE [<scope>, <extent>]
```

EXAMPLE 2 Structure of two scope-extent pairings

```
USAGE [scope1, extent1], USAGE [scope2, extent2]
```

The individual <scope> and <extent> elements are described in the following subclauses. See [7.3.2.4](#) for further examples of scope and extent pairings.

7.3.2.2 Scope

Scope describes the purpose or purposes for which a CRS, datum, datum ensemble, coordinate operation or bound CRS is applied.

Requirement: The WKT representation of a <scope> shall be:

```
<scope> ::= <scope keyword> <left delimiter>
           <scope text description> <right delimiter>
<scope keyword> ::= SCOPE
<scope text description> ::= <quoted Latin text>
```

EXAMPLE SCOPE["Large scale topographic mapping and cadastre."]

7.3.2.3 Extent

7.3.2.3.1 General

Extent describes the spatial and/or temporal applicability of a CRS, datum, datum ensemble, coordinate operation or bound CRS. Extent in this document uses the concepts described in ISO 19115-1. However, this document permits horizontal extent to be described by description and/or by geographic bounding box, but not by polygon because of string length considerations. It also allows for vertical and temporal extent. These extent attributes are all optional, although at least one shall be included in any Usage. Multiple extent attributes may be provided, but there is a constraint that within any one scope-extent pairing these shall be of different types.

Requirement: The WKT representation of an <extent> shall be:

```
<extent> ::= <area description> | <geographic bounding box> | <vertical extent> | <temporal extent>
| { <area description> <wkt separator> <geographic bounding box> }
| { <area description> <wkt separator> <vertical extent> }
| { <area description> <wkt separator> <temporal extent> }
| { <geographic bounding box> <wkt separator> <vertical extent> }
| { <geographic bounding box> <wkt separator> <temporal extent> }
| { <vertical extent> <wkt separator> <temporal extent> }
| { <area description> <wkt separator> <geographic bounding box> <wkt separator>
  <vertical extent> }
| { <area description> <wkt separator> <geographic bounding box> <wkt separator>
  <temporal extent> }
| { <area description> <wkt separator> <vertical extent> <wkt separator>
  <temporal extent> }
| { <geographic bounding box> <wkt separator> <vertical extent> <wkt separator>
  <temporal extent> }
| { <area description> <wkt separator> <geographic bounding box> <wkt separator>
  <vertical extent> <wkt separator> <temporal extent> }
```

7.3.2.3.2 Area description

Area description is an optional attribute which describes a geographic area over which a CRS or coordinate operation is applicable.

Requirement: The WKT representation of an <area description> shall be:

```
<area description> ::= <area description keyword> <left delimiter>
  <area text description> <right delimiter>
<area description keyword> ::= AREA
<area text description> ::= <quoted Latin text>
```

EXAMPLE AREA["Netherlands offshore."]

7.3.2.3.3 Geographic bounding box

The geographic bounding box is an optional attribute which describes a "north up" area. Upper right latitude will be greater than the lower left latitude. Generally, the upper right longitude will be greater than the lower left longitude. However, when the area crosses the 180° meridian, the value of the lower left longitude will be greater than the value of the upper right longitude.

Requirement: The WKT representation of a <temporal extent> shall be:

```

<temporal extent> ::= <temporal extent keyword> <left delimiter>
                    <temporal extent start> <wkt separator>
                    <temporal extent end> <right delimiter>

<temporal extent keyword> ::= TIMEEXTENT
<temporal extent start> ::= <datetime> | <quoted Latin text>
<temporal extent end> ::= <datetime> | <quoted Latin text>

!! <temporal extent end> should have the same data
type (dateTime or quoted Latin text) as <temporal
extent start>.
    
```

EXAMPLE 1 TIMEEXTENT[2013-01-01,2013-12-31]

EXAMPLE 2 TIMEEXTENT["Jurassic","Quaternary"]

7.3.2.4 Examples of WKT describing usage

EXAMPLE 1 One scope-extent pairing (the <extent> including two extent attributes, area description and temporal extent):

```

USAGE[SCOPE["Spatial referencing."],
AREA["Netherlands offshore."],TIMEEXTENT[1976-01,2001-04]]
    
```

EXAMPLE 2 Two scope-extent pairings, the second of which has multiple extent attributes:

```

USAGE[SCOPE["Small scale topographic mapping."],
AREA["Finland - onshore and offshore."],
USAGE[SCOPE["Cadastre."],
AREA["Finland - onshore between 26°30'E and 27°30'E."],
BBOX[60.36,26.5,70.05,27.5]]
    
```

7.3.3 Identifier

Identifier is an optional attribute which references an external description of the object and which may be applied to a coordinate reference system, a coordinate operation or a bound CRS. It may also be utilised for components of these objects although this is recommended only for the following circumstances:

- coordinate operation methods and parameters;
- source and target CRSs when embedded within a coordinate transformation or a concatenated coordinate operation;
- source CRS when embedded within a point motion operation;
- individual coordinate operations embedded within a concatenated coordinate operation;
- base CRS when embedded within a derived CRS (including projected CRS);
- source CRS, target CRS and abridged transformation when embedded within a bound CRS;
- individual members of a datum ensemble.

Multiple identifiers may be given for any object.

When an identifier is given for a coordinate reference system, coordinate operation or bound CRS, it applies to the whole object including all of its components.

Should any attributes or values given in the cited identifier be in conflict with attributes or values given explicitly in the WKT description, the WKT values shall prevail.

Requirement: The WKT representation of an <identifier> shall be:

<identifier>	::=	<identifier keyword> <left delimiter> <authority name> <wkt separator> <authority unique identifier> [<wkt separator> <version>] [<wkt separator> <authority citation>] [<wkt separator> <id uri>] <right delimiter>
<identifier keyword>	::=	ID
<authority name>	::=	<quoted Latin text>
<authority unique identifier>	::=	<number> <quoted Latin text>
<version>	::=	<number> <quoted Latin text>
<authority citation>	::=	<citation keyword> <left delimiter> <citation> <right delimiter>
<citation keyword>	::=	CITATION
<citation>	::=	<quoted Latin text>
<id uri>	::=	<uri keyword> <left delimiter> <uri> <right delimiter>
<uri keyword>	::=	URI
<uri>	::=	<quoted Latin text>

Version is an optional attribute indicating the version of the repository or object that is cited. Citation is an optional attribute that may be used to give further details of the authority. URI is an optional attribute that may be used to give reference to an online resource.

NOTE In previous specifications the authority object was defined more narrowly than is the identifier object in this document. See [Annex C](#).

EXAMPLE 1 ID["Authority name", "Abcd_Ef", 21]

EXAMPLE 2 ID["EPSG", 4326]

EXAMPLE 3 ID["EPSG", 4326, URI["urn:ogc:def:crs:EPSG::4326"]]

EXAMPLE 4 ID["EuroGeographics", "ES_ED50 (BAL99) to ETRS89", "2001-04-20"]

Further examples are included in [Clauses 8](#) to [17](#).

7.3.4 Remark

<remark> is an optional attribute. Any information contained in a <remark> is informative. It does not modify the defining parameters of an object.

A <remark> may be applied to a coordinate reference system, coordinate operation or bound CRS as a whole. A <remark> should not be included in the WKT for components of a coordinate reference system or coordinate operation, but a remark in the coordinate reference system or coordinate operation object may include information about these components.

NOTE A <remark> can be included within the descriptions of source and target CRS embedded within a coordinate transformation as well as within the coordinate transformation itself.

Any character other than a <wkt Latin text character> that is to be contained in a CRS WKT string may be included only as part of <quoted Unicode text> within a <remark>.

Requirement: The WKT representation of a <remark> shall be:

<remark>	::=	<remark keyword> <left delimiter> <quoted Unicode text> <right delimiter>
----------	-----	--

<remark keyword> ::= REMARK

EXAMPLE 1 REMARK["A remark in ASCII"]

EXAMPLE 2 REMARK["Замечание на русском языке"]

EXAMPLE 3 GEOGCRS["S-95",
 DATUM["Pulkovo 1995",
 ELLIPSOID["Krassowsky 1940",6378245,298.3,
 LENGTHUNIT["metre",1.0]]
],
 CS[ellipsoidal,2],
 AXIS["latitude",north,ORDER[1]],
 AXIS["longitude",east,ORDER[2]],
 ANGLEUNIT["degree",0.0174532925199433],
 REMARK["Система Геодезических Координат года 1995 (СК-95)"]
]

Further examples including remarks are given in [Clauses 8 to 19](#).

7.4 Unit and unit conversion factor

7.4.1 Unit description

Some attributes of coordinate system axes and coordinate operation parameters are numbers which require the unit to be specified. General aspects are described here. For additional aspects that are specific to ordinal and temporal coordinate system axes see [2.5.6](#).

Requirement: The WKT representation of a <unit> description shall be:

<unit>	::=	<spatial unit> <time unit>
<spatial unit>	::=	<angle unit> <length unit> <parametric unit> <scale unit>
<angle unit>	::=	<angle unit keyword> <left delimiter> <unit name> <wkt separator> <conversion factor> [{ <wkt separator> <identifier> }]... <right delimiter>
<length unit>	::=	<length unit keyword> <left delimiter> <unit name> <wkt separator> <conversion factor> [{ <wkt separator> <identifier> }]... <right delimiter>
<parametric unit>	::=	<parametric unit keyword> <left delimiter> <unit name> <wkt separator> <conversion factor> [{ <wkt separator> <identifier> }]... <right delimiter>
<scale unit>	::=	<scale unit keyword> <left delimiter> <unit name> <wkt separator> <conversion factor> [{ <wkt separator> <identifier> }]... <right delimiter>
<time unit>	::=	<time unit keyword> <left delimiter> <unit name> <wkt separator> <conversion factor> [{ <wkt separator> <identifier> }]... <right delimiter>
<angle unit keyword>	::=	ANGLEUNIT UNIT <i>!! In this document when the <time unit> is applied to a temporal coordinate system axis the <conversion factor> is conditional: see 7.4.3.</i>
		<i>!! In this document the preferred keyword is ANGLEUNIT. UNIT is permitted for backward compatibility. Implementations shall be able to read both forms.</i>

<length unit keyword>	::=	LENGTHUNIT UNIT <i>!! In this document the preferred keyword is LENGTHUNIT. UNIT is permitted for backward compatibility. Implementations shall be able to read both forms.</i>
<parametric unit keyword>	::=	PARAMETRICUNIT
<scale unit keyword>	::=	SCALEUNIT UNIT <i>!! In this document the preferred keyword is SCALEUNIT. UNIT is permitted for backward compatibility. Implementations shall be able to read both forms.</i>
<time unit keyword>	::=	TIMEUNIT TEMPORALQUANTITY <i>!! In this document the preferred keyword is TIMEUNIT. TEMPORALQUANTITY is permitted. Implementations shall be able to read both forms.</i>
<unit name>	::=	<quoted Latin text>
<conversion factor>	::=	<unsigned numeric literal> <i>!! <conversion factor> is the number of SI standard units per unit. See 7.4.2 and 7.4.3.</i>

NOTE <identifier> is described in [7.3.4](#).

7.4.2 Conversion factor — Spatial and parametric units

Requirements: If the unit is linear its conversion factor shall be to metres and is the number of metres per unit. If the unit is angular its conversion factor shall be to radians and is the number of radians per unit. For parametric units the appropriate SI standard unit shall be used. For scale, unity shall be used.

EXAMPLE 1 LENGTHUNIT["metre",1]
 EXAMPLE 2 LENGTHUNIT["German legal metre",1.0000135965]
 EXAMPLE 3 ANGLEUNIT["degree",0.0174532925199433]
 EXAMPLE 4 SCALEUNIT["parts per million",1E-06]
 EXAMPLE 5 PARAMETRICUNIT["hectopascal",100]

Further examples are included in [7.5](#) and [Clauses 8](#) to [17](#).

7.4.3 Conversion factor — Temporal quantities

Unit uses the datatype of UnitOfMeasure. This is defined in ISO 19103. The class includes a note "conversion ToISOstandardUnit is not null only if the conversion is a simple scale". For many temporal cases, the unit is not a simple scale: the size of a month, a day or an hour vary at different locations in the calendar due to correction factors and alterations such as leap seconds, leap years, and seasonal time zone changes. Conversion of a temporal quantity (time unit) to the SI base unit for time, the second, therefore may or may not be ambiguous when compared to a calendar definition of that quantity. Examples are given in ISO 19111:2019, D.4. In these cases, UnitOfMeasure instances of temporalCount and temporalMeasure are described with no relation to the second.

NOTE In ISO 8601-1 the terms 'calendar day', 'calendar month' and 'calendar year' are used, with the note: often referred to as 'day', 'month' and 'year' respectively.

POSIX time is commonly used in software. It is dimensioned in seconds, but leap seconds are ignored (not applied)^[6]. A unit of measure "second" may be used to represent this, but it shall be defined independent of the SI second, not as a specific number of SI seconds. It may be thought of as a "calendar second".

Requirement: For temporal quantities, in this document called time units, the unit conversion factor shall be to seconds and is the number of seconds per unit, but the conversion factor is not to be given when it is not a simple scaling.

EXAMPLE 1 Simple scaling, so conversion factor is required.

`TIMEUNIT["millisecond",0.001]`

EXAMPLE 2 Scaling not simple (number of seconds in a month varies), so conversion factor not required.

`TIMEUNIT["calendar month"]`

EXAMPLE 3 Using the POSIX formula^[6] which ignores leap seconds, so conversion factor not required.

`TIMEUNIT["calendar second"]`

NOTE The example given in ISO 19162:2015:

`TIMEUNIT["day",86400.0]`

is deprecated. The number of seconds in a calendar day is not a simple scaling (because some days contain leap seconds) so conversion factor is not to be given.

7.4.4 Default unit

Recommended practice is for units to be explicitly described. However, for backward compatibility reasons, this document in places permits unit description to be implied. This may result in incomplete specification.

EXAMPLE Angular units cannot be inferred from a coordinate system with linear units, as in the angle unit for longitude of a prime meridian which cannot be inferred from a geodetic coordinate reference system having a Cartesian coordinate system.

Requirement: Where no implied unit can be inferred then in this document the default implied linear unit shall be metre, the default implied angular unit shall be degree.

7.5 Coordinate system

7.5.1 Syntax

Most coordinate system attributes are common to all subtypes of spatial and temporal coordinate systems. Exceptions are associated with the coordinate system axis unit attribute and its qualifier, the conversion factor to an SI base unit:

- a) When the coordinate system type is 'temporalCount' or 'temporalMeasure', the inclusion of the axis unit conversion factor in WKT is conditional, see [7.4.3](#).
- b) When the coordinate system type is 'ordinal' or 'temporalDateTime', the axis unit attribute and its conversion factor are not required in WKT, see [7.5.6](#) and [13.3](#).

The syntax for all coordinate systems is described here.

Requirement: The WKT representation of a coordinate system shall be:

```

<coordinate system> ::= <spatial cs>
                        | <temporalCountMeasure cs> | <ordinal-dateTime cs>
<spatial cs> ::= <cs keyword> <left delimiter> <spatial cs type>
                <wkt separator> <dimension>
                [ { <wkt separator> <identifier> } ]... <right delimiter>
                { <wkt separator> <spatial axis> }...
                [ <wkt separator> <cs unit> ]
                !! Requires axis unit attributes including
                conversion factor.
    
```

<temporalCountMeasure cs>	::=	<cs keyword> <left delimiter> <temporalCountMeasure cs type> <wkt separator> <dimension> [{ <wkt separator> <identifier> }]... <right delimiter> <wkt separator> <temporalCountMeasure axis> <i>!! Requires axis unit attributes, conversion factor is conditional. See 7.5.2.</i>
<ordinal-dateTime cs>	::=	<cs keyword> <left delimiter> <ordinal-dateTime cs type> <wkt separator> <dimension> [{ <wkt separator> <identifier> }]... <right delimiter> { <wkt separator> <ordinal-dateTime axis> }... <i>!! Axis unit is not required.</i>
<cs keyword>	::=	CS
<spatial cs type>	::=	affine Cartesian cylindrical ellipsoidal linear parametric polar spherical vertical <i>!! See 7.5.2 for constraints.</i>
<temporalCountMeasure cs type>	::=	temporalCount temporalMeasure <i>!! See 7.5.2 for constraints.</i>
<ordinal-dateTime cs type>	::=	ordinal temporalDateTime <i>!! See 7.5.2 for constraints.</i>
<dimension>	::=	1 2 3 <i>!! Unsigned integer. See 7.5.2 for constraints.</i>
<spatial axis>	::=	<axis keyword> <left delimiter> <axis nameAbbrev> <wkt separator> <axis direction> [<wkt separator> <axis order>] [<wkt separator> <spatial unit>] [{ <wkt separator> <identifier> }]... <right delimiter> <i>!! Axis unit and conversion factor mandatory. See 7.5.6.2.</i>
<temporalCountMeasure axis>	::=	<axis keyword> <left delimiter> <axis nameAbbrev> <wkt separator> <axis direction> [<wkt separator> <axis order>] [<wkt separator> <time unit>] [{ <wkt separator> <identifier> }]... <right delimiter> <i>!! Axis unit mandatory, conversion factor conditional. See 7.5.6.4.</i>
<ordinal-dateTime axis>	::=	<axis keyword> <left delimiter> <axis nameAbbrev> <wkt separator> <axis direction> [<wkt separator> <axis order>] [{ <wkt separator> <identifier> }]... <right delimiter> <i>!! The attribute 'axis unit' is not required for an ordinal coordinate system or for a temporal dateTime coordinate system. See 7.5.6.3.</i>
<axis keyword>	::=	AXIS
<axis nameAbbrev>	::=	<double quote> { <axis name> <axis abbreviation> <axis name and abbrev> } <double quote>
<axis name>	::=	<wkt Latin text character>...
<axis abbreviation>	::=	<left paren> <simple Latin letter>... <right paren>

<axis name and abbrev>	::=	<axis name> <space> <axis abbreviation> <i>!! See 7.5.3 for requirements for axis name and abbreviation.</i>
<axis direction>	::=	north [{ <wkt separator> <meridian> }] northNorthEast northEast eastNorthEast east eastSouthEast southEast southSouthEast south [{ <wkt separator> <meridian> }] southSouthWest southWest westSouthWest west westNorthWest northWest northNorthWest geocentricX geocentricY geocentricZ up down forward aft port starboard clockwise <wkt separator> <bearing> counterClockwise <wkt separator> <bearing> columnPositive columnNegative rowPositive rowNegative displayRight displayLeft displayUp displayDown future past towards awayFrom unspecified <i>!! See 7.5.4 for constraints.</i>
<meridian>	::=	<meridian keyword> <left delimiter> <number> <wkt separator> <angle unit> <right delimiter> <i>!! See 7.5.4 for constraints.</i>
<meridian keyword>	::=	MERIDIAN
<bearing>	::=	<bearing keyword> <left delimiter> <number> <right delimiter> <i>!! See 7.5.4 for constraints.</i>
<bearing keyword>	::=	BEARING
<axis order>	::=	<axis order keyword> <left delimiter> <unsigned integer> <right delimiter> <i>!! See 7.5.5 for constraints.</i>
<axis order keyword>	::=	ORDER
<cs unit>	::=	<unit> <i>!! See 7.4 and 7.5.6 for constraints.</i>
<spatial unit>		See 7.4 and 7.5.6 .
<time unit>		See 7.4 for time unit definition and 13.3 for application to coordinate systems.

An <ellipsoidal 2D coordinate system> is a special case of <coordinate system> required in the construct of the <geographic2D crs> element of a compound CRS ([Clause 15](#)):

<ellipsoidal 2D coordinate system>	::=	<cs keyword> <left delimiter> <ellipsoidal 2D cs type> <wkt separator> <ellipsoidal 2D dimension> [{ <wkt separator> <identifier> }]... <right delimiter> { <wkt separator> <spatial axis> }... [<wkt separator> <cs unit>]
<ellipsoidal 2D cs type>	::=	ellipsoidal
<ellipsoidal 2D dimension>	::=	2

7.5.2 Coordinate system type, dimension and coordinate data type

For various types of CRS the type of coordinate system that may be used is constrained, as is the permissible number of axes. Additionally, the data type for coordinates in an ordinal coordinate system and in a temporal coordinate system is constrained. These constraints are summarised in [Table 3](#).

Table 3 — Permitted coordinate system type, dimension and coordinate data type by CRS

CRS type	Permitted CS type(s)	Dimension (number of axes)	Coordinate data type
geodetic	Cartesian	3	
derived geodetic	spherical (ellipsoidal — read only: see 8.3)	3 (2 or 3)	
geographic	ellipsoidal	2 or 3	
derived geographic			
projected	Cartesian	2 or 3	
derived projected	affine Cartesian cylindrical ordinal polar spherical	2 or 3 2 or 3 3 1 or 2 or 3 2 3	(no constraint) (no constraint) (no constraint) integer (no constraint) (no constraint)
vertical	vertical	1	
derived vertical			
engineering	affine Cartesian cylindrical linear ordinal polar spherical	2 or 3 2 or 3 3 1 1 or 2 or 3 2 3	(no constraint) (no constraint) (no constraint) (no constraint) integer (no constraint) (no constraint)
derived engineering			
parametric	parametric	1	
derived parametric			
temporal	temporalDateTime	1	dateTime
derived temporal	temporalCount temporalMeasure	1 1	integer real

See [6.5](#) for comment on case sensitivity.

The constraints on the data type of coordinates in an ordinal or temporal coordinate system applies to the coordinate data set. The data type is not explicitly described in the CRS definition. However, it is implied. If a CRS's coordinate system is ordinal or is temporalCount, it is implied that the coordinates referenced to the CRS shall be integer. If a CRS's coordinate system is temporalMeasure, it is implied that the coordinates referenced to the CRS shall be real; if a CRS's coordinate system is temporalDateTime, it is implied that the coordinates referenced to the CRS shall be dateTime.

7.5.3 Axis name and abbreviation

ISO 19111 requires the name and abbreviation for each axis. In this document, name and/or abbreviation is permitted. They are contained in a single quoted text string. If abbreviation is included in the text string it is given in parentheses.

EXAMPLE 1 "easting"

EXAMPLE 2 "(X)"

EXAMPLE 3 "easting (X)"

Requirement: The following constraints shall apply:

- a) In WKT strings all axis abbreviations shall be from the <wkt Latin text character> set.
- b) For geodetic CRSs having a geocentric Cartesian coordinate system, in WKT strings the axis name should be omitted as it is given through the mandatory axis direction, see 7.5.4 c), but the axis abbreviation, respectively 'X', 'Y' and 'Z', shall be given.
- c) For geographic CRSs having a two-dimensional ellipsoidal coordinate system, the two-dimensional ellipsoidal coordinate system axes are geodetic latitude and geodetic longitude. In WKT strings the values of axis name shall be 'latitude' and 'longitude' respectively.

ISO 19111 specifies the lower-case Greek letters φ and λ as symbols for geodetic latitude and geodetic longitude. In this document the abbreviations to be used in WKT strings shall be from the Latin character set. P and L are the transliterations of the Greek letters phi and lambda. B for Breite and L for Länge are the standard German abbreviations and used in academic texts worldwide. 'lat' and 'lon' are in common usage.

- d) For geographic CRSs having a three-dimensional ellipsoidal coordinate system, the name and abbreviation of the horizontal axes in a WKT string shall follow the requirements in c). The vertical axis name shall be 'ellipsoidal height'; the vertical axis abbreviation shall be 'h' and shall be included when abbreviations for the horizontal axes are included.
- e) For projected CRSs, the two-dimensional Cartesian coordinate system axes names shall be 'northing' or 'southing' and 'easting' or 'westing' and shall be given when the axis direction and order are not east first, north second. Axis name may be omitted from WKT strings when the axis direction and order are east first, north second with abbreviations 'E' and 'N' respectively. Axis abbreviation shall be given.
- f) For projected CRSs having a three-dimensional Cartesian coordinate system, the name and abbreviation of the horizontal axes in a WKT string shall follow the requirements in e). The vertical axis name shall be 'ellipsoidal height'; the vertical axis abbreviation shall be 'h' and shall be included.
- g) For vertical CRSs, the axis direction is up or down, see 7.5.4. In WKT strings the value of axis name shall be 'gravity-related height' and 'depth' respectively. Axis abbreviation may be omitted but if given for height it shall not be 'h' (which is used for the ellipsoidal height component of an ellipsoidal 3D coordinate system, see 7.5.3 d) above); the abbreviation for gravity-related height should normally be 'H'.

Recommendations:

- For engineering CRSs using a polar coordinate system where the lower-case Greek letter θ is conventionally used as the symbol for direction, the letter 'U' from the Latin character set should be used as a one-character abbreviation in WKT strings.
- For geodetic and engineering CRSs using a spherical coordinate system where the Greek letters φ and θ are conventionally used as the symbols for direction, the letters 'U' and 'V' respectively from the Latin character set should be used as a one-character abbreviations in WKT strings.

7.5.4 Axis direction

Axis direction indicates the positive increment along an axis. The handedness of a 2- or 3-dimensional coordinate system may be derived from the directions.

Requirement: the following constraints shall apply:

- a) For geographic CRSs having an ellipsoidal 2-D coordinate system, the two-dimensional ellipsoidal coordinate system axes are geodetic latitude, positive northwards, and geodetic longitude, positive eastwards. Axis direction shall be 'north' and 'east' respectively.

- b) For geographic CRSs having an ellipsoidal 3-D coordinate system, the three-dimensional ellipsoidal coordinate system axes are geodetic latitude, positive northwards, geodetic longitude, positive eastwards, and ellipsoidal height, positive upwards. Axis direction shall be 'north', 'east' and 'up' respectively.
- c) For geodetic CRSs having a geocentric Cartesian coordinate system, in WKT strings the axis directions shall be 'geocentricX', 'geocentricY' and 'geocentricZ' respectively. The first axis of the earth-centred 3D Cartesian coordinate system lies in the equatorial plane such that a vector pointing in the positive direction passes through the intersection of the equator and the prime meridian. The second axis is in the equatorial plane such that a vector pointing in the positive direction passes through the intersection of the equator and the meridian of 90°E. The third axis is perpendicular to the first two such that it completes a right-handed coordinate system; it is approximately parallel to the earth's rotation axis, positive towards the north pole.
- d) For projected CRSs, except for coordinate systems centred on a pole, the horizontal axis direction shall be 'north' or 'south' and 'east' or 'west'. For the 3D case the vertical axis direction shall be 'up'.

For coordinate systems centred on a pole the direction for both axes will be 'south' (for the north pole case) or 'north' (for the south pole case); the axes direction shall be supplemented with a <meridian> description. This is the value of the meridian that the axis follows from the pole. The prime meridian from which the meridian value is reckoned is given through the <prime meridian> object; if no <prime meridian> object is in the WKT string then the IRM or Greenwich meridian shall be assumed.

- e) For vertical CRSs, the axis direction shall be 'up' or 'down'.
- f) For temporal CRSs, the axis direction shall be 'future' or 'past'.
- g) In engineering CRSs the horizontal directions are only approximate, the set of directions indicating whether the coordinate system is left-handed or right-handed. (In the 2D case, the handedness is when viewed from above the plane of the system). For engineering CRSs with polar coordinate systems the direction of the rotational axis shall be 'clockwise' or 'counterClockwise'. The specified direction from which the rotation is measured shall be given through the supplementary object <bearing>; the bearing value shall be given in the unit defined through <axis unit>.

See [6.5](#) for comment on case sensitivity.

7.5.5 Axis order

Axis is repeated in a sequence. The number of axes in the sequence is the same as the dimensions of the coordinate system.

<axis order> identifies the order in which the coordinates of a point in a dataset are given and therefore is significant. In this document it is defined in the BNF as an optional attribute to allow backward compatibility with OGC 01-009, however it is recommended that it should be explicitly included in a CRS WKT string.

Requirement: The following constraints shall apply:

- a) For coordinate systems with more than one axis, either every axis description shall include an <axis order> or none of the axes descriptions shall include an <axis order>. If <order> is included a sequence value shall not be repeated.
- b) When <axis order> is present in the WKT string the sequence of <axis> descriptions shall start at 1 and increment in steps of 1.
- c) If <axis order> is omitted from the WKT string the sequence of <axis> descriptions shall imply the order of the axes and of coordinates referenced to the CRS.
- d) For compound CRSs, the axes are described through each component CRS description and the order values shall apply to each component system, not to the compound system. The order of the axes in

the compound system shall be inferred from firstly the order of the component CRSs then secondly the order of axes within each component CRS.

EXAMPLE A compound CRS consists of a projected CRS with a vertical CRS, the component CRSs described in that order. The axes of the projected CRS are northing first, easting second. The only and therefore first axis of the vertical CRS is gravity-related height. The axis order for the compound CRS is northing first, easting second and gravity-related height third.

7.5.6 Axis unit and coordinate system unit

7.5.6.1 General

This document provides two methods for specifying the coordinate system axis units. <axis unit> is an optional attribute which may be applied to each axis description and if applied shall describe the unit for that axis. <cs unit> is an optional attribute which if applied shall specify the unit for all axes of the coordinate system. <axis unit> and <cs unit> are subsets of <unit> which is described in 7.4.

For coordinate systems in which all axes have the same units, the use of <axis unit> leads to duplication of unit name and conversion factor. Duplication is avoided through the use of <cs unit>.

<axis unit> or <cs unit> may also specify the unit for implied map projection parameter values, as described in 9.3.

7.5.6.2 Axis unit for spatial and parametric coordinate systems

Requirement: The following constraints shall apply:

- a) A CRS WKT string for all types of coordinate system except ordinal, temporalDateTime, temporalCount and temporalMeasure shall include either <spatial axis unit> for each axis or <cs unit> applying to all axes.
- b) <cs unit> shall not be used if the unit does not apply to all axes. In these cases <axis unit> shall be used.

EXAMPLE 1 A polar coordinate system requires one axis to be an angle and the other axis to be a length: <spatial axis unit> shall be used.

EXAMPLE 2 A three-dimensional ellipsoidal coordinate system requires two axes (geodetic latitude and geodetic longitude) to be angles and the third axis (ellipsoidal height) to be a length: <spatial axis unit> shall be used.

7.5.6.3 Axis unit for ordinal coordinate systems

The coordinates in ordinal coordinate systems are integers. The WKT syntax for an ordinal coordinate system is given in 7.5.1; no axis unit shall be given.

Requirement: For an ordinal coordinate system, neither <axis unit> nor <cs unit> shall be given.

EXAMPLE

```
CS[ordinal,2],
  AXIS["inline (I)",southeast,ORDER[1]]
  AXIS["crossline (J)",northeast,ORDER[2]]
```

NOTE This requirement also applies to temporalDateTime coordinate systems, see 13.3.2.

Per 7.5.1 and 7.5.3, for an ordinal CS both axis name/abbreviation and axis direction shall be given but axis order may be given or be implied.

7.5.6.4 Axis unit for temporal coordinate systems

Temporal coordinate systems have constraints on their treatment of <axisUnit> and its <conversion factor> which differ from the requirements for other types of coordinate system. Unit for temporalCount

and temporalMeasure coordinate system axis is described in 7.4.3. Requirements for temporalDateTime coordinate system axis unit are described in 13.3.

7.5.7 Examples of WKT describing coordinate systems

7.5.7.1 Coordinate systems for geodetic CRSs

EXAMPLE 1 Earth centred earth fixed Cartesian CS. Axis order is implied, <cs unit> is used.

```
CS[Cartesian,3],
  AXIS["(X)",geocentricX],
  AXIS["(Y)",geocentricY],
  AXIS["(Z)",geocentricZ],
  LENGTHUNIT["metre",1.0]
```

EXAMPLE 2 Topocentric Cartesian CS. Axis order is implied, <cs unit> is used.

```
CS[Cartesian,3],
  AXIS["(X)",east],AXIS["(Y)",north],AXIS["(Z)",up],
  LENGTHUNIT["metre",1.0]
```

EXAMPLE 3 Spherical CS. Axis order is explicit, <axis unit> is used, axes have name and abbreviation.

```
CS[spherical,3],
  AXIS["distance (r)",awayFrom,ORDER[1],LENGTHUNIT["kilometre",1000]],
  AXIS["longitude (U)",counterClockwise,BEARING[0],ORDER[2],
  ANGLEUNIT["degree",0.0174532925199433]],
  AXIS["elevation (V)",up,ORDER[3],
  ANGLEUNIT["degree",0.0174532925199433]]
```

7.5.7.2 Coordinate systems for geographic CRSs

EXAMPLE 1 3D ellipsoidal CS. Axis order is explicit, <axis unit> is used, first two axes have name but no abbreviation.

```
CS[ellipsoidal,3],
  AXIS["latitude",north,ORDER[1],ANGLEUNIT["degree", 0.0174532925199433]],
  AXIS["longitude",east,ORDER[2],ANGLEUNIT["degree", 0.0174532925199433]],
  AXIS["ellipsoidal height (h)",up,ORDER[3],LENGTHUNIT["metre",1.0]]
```

EXAMPLE 2 2D ellipsoidal CS. Axis order is implied, <cs unit> is used, axes have abbreviation but no name.

```
CS[ellipsoidal,2],
  AXIS["(lat)",north],
  AXIS["(lon)",east],
  ANGLEUNIT["degree",0.0174532925199433]
```

7.5.7.3 Coordinate systems for projected CRSs

EXAMPLE 1 CS[Cartesian,2],
 AXIS["(E)",east,ORDER[1],LENGTHUNIT["metre",1.0]],
 AXIS["(N)",north,ORDER[2],LENGTHUNIT["metre",1.0]] (using <axis unit>)

EXAMPLE 2 CS[Cartesian,2],
 AXIS["(E)",east],
 AXIS["(N)",north],
 LENGTHUNIT["metre",1.0] (using <cs unit>)

EXAMPLE 3 CS[Cartesian,2],
 AXIS["northing (X)",north,ORDER[1]],
 AXIS["easting (Y)",east,ORDER[2]],
 LENGTHUNIT["German legal metre",1.0000135965]

EXAMPLE 4 For an azimuthal projection centred on the north pole:

```
CS[Cartesian,2],
  AXIS["easting (X)",south,
    MERIDIAN[90,ANGLEUNIT["degree",0.0174532925199433]],ORDER[1]
  ],
  AXIS["northing (Y)",south,
    MERIDIAN[180,ANGLEUNIT["degree",0.0174532925199433]],ORDER[2]
  ],
  LENGTHUNIT["metre",1.0]
```

EXAMPLE 5 For a projected 3D CRS (using <cs unit>)

```
CS[Cartesian,3],
  AXIS["(E)",east],
  AXIS["(N)",north],
  AXIS["ellipsoid height (h)",up],
  LENGTHUNIT["metre",1.0]
```

7.5.7.4 Coordinate systems for vertical CRSs

EXAMPLE 1 CS[vertical,1],
 AXIS["gravity-related height (H)",up],
 LENGTHUNIT["metre",1.0]

(using <cs unit>)

EXAMPLE 2 CS[vertical,1],
 AXIS["depth (D)",down],
 LENGTHUNIT["metre",1.0]

(using <axis unit>)

7.5.7.5 Coordinate systems for engineering CRSs

EXAMPLE 1 CS[Cartesian,2],
 AXIS["site north (x)",southeast,ORDER[1]],
 AXIS["site east (y)",southwest,ORDER[2]],
 LENGTHUNIT["metre",1.0]

EXAMPLE 2 CS[polar,2],
 AXIS["distance (r)",awayFrom,ORDER[1],LENGTHUNIT["metre",1.0]],
 AXIS["bearing (U)",clockwise,BEARING[234],ORDER[2],
 ANGLEUNIT["degree",0.0174532925199433]
]

EXAMPLE 3 CS[Cartesian,3],
 AXIS["ahead (x)",forward,ORDER[1]],
 AXIS["right (y)",starboard,ORDER[2]],
 AXIS["down (z)",down,ORDER[3]],
 LENGTHUNIT["metre",1.0]

EXAMPLE 4 CS[ordinal,2],
 AXIS["Inline (I)",northEast,ORDER[1]],
 AXIS["Crossline (J)",northwest,ORDER[2]]

7.6 Datum ensemble

Geodetic and vertical CRSs are associated with either a reference frame (datum) or a datum ensemble. The members of a datum ensemble are given as a list of reference frames. The list may contain reference frame name and/or identifier. All members of a datum ensemble are realizations of one shared terrestrial or vertical reference system.

For an ensemble of geodetic reference frames (datums), the WKT string includes the description of the ellipsoid used by the members. This information is available from any and all of the definitions of each member. It is included in the ensemble WKT to facilitate direct access to the information. The WKT string for a datum ensemble may also include the description of the prime meridian applying to all members of the ensemble.

For both geodetic and vertical datum ensembles, the ensemble description includes its 'accuracy', an indication of the difference in coordinate values of a point between different members of the datum ensemble. It may be regarded as a measure of the inaccuracy introduced through the assumption that ensemble members are approximately equivalent.

Use of the datum ensemble concept comes with a health warning. If data is associated with a CRS having a datum ensemble, it will not be possible to identify which of the datum ensemble members the data might more accurately be referenced to. In high accuracy applications, datum ensembles should not be used; individual reference frames should be identified.

Requirement: The WKT representation of a datum ensemble shall be:

<datum ensemble>	::=	<geodetic datum ensemble> <vertical datum ensemble>
<geodetic datum ensemble>	::=	<datum ensemble keyword> <left delimiter> <datum ensemble name> { <wkt separator> <datum ensemble member> }... <wkt separator> <ellipsoid> <wkt separator> <datum ensemble accuracy> [<wkt separator> <datum ensemble identifier>]... <right delimiter> { <wkt separator> <prime meridian> }
<vertical datum ensemble>	::=	<datum ensemble keyword> <left delimiter> <datum ensemble name> { <wkt separator> <datum ensemble member> }... <wkt separator> <datum ensemble accuracy> [{ <wkt separator> <datum ensemble identifier> }]... <right delimiter>
<datum ensemble keyword>	::=	ENSEMBLE
<datum ensemble name>	::=	<quoted Latin text> !! See 7.2
<datum ensemble member>	::=	<datum ensemble member keyword> <left delimiter> <datum ensemble member name> [{ <wkt separator> <datum ensemble member identifier> }]... <right delimiter>
<datum ensemble member keyword>	::=	MEMBER
<datum ensemble member name>	::=	<quoted Latin text> !! See 7.2
<datum ensemble member identifier>	::=	<identifier> !! See 7.3.4
<datum ensemble accuracy>	::=	<datum ensemble accuracy keyword> <left delimiter> <accuracy> <right delimiter>
<datum ensemble accuracy keyword>	::=	ENSEMBLEACCURACY
<accuracy>	::=	<number> !! <accuracy> is in metres
<datum ensemble identifier>	::=	<identifier> !! See 7.3.4

<ellipsoid> is described in [8.2.1](#).

<prime meridian> is described in [8.2.2](#).

EXAMPLE 1 Ensemble of geodetic reference frames with ensemble member name only

```
ENSEMBLE ["WGS 84 ensemble",
  MEMBER ["WGS 84 (TRANSIT)"],
  MEMBER ["WGS 84 (G730)"],
  MEMBER ["WGS 84 (G834)"],
  MEMBER ["WGS 84 (G1150)"],
  MEMBER ["WGS 84 (G1674)"],
  MEMBER ["WGS 84 (G1762)"],
  ELLIPSOID ["WGS 84", 6378137, 298.2572236, LENGTHUNIT ["metre", 1.0]],
  ENSEMBLEACCURACY [2]
]
```

EXAMPLE 2 Ensemble of geodetic reference frames with ensemble member name and ID

```
ENSEMBLE ["WGS 84 ensemble",
  MEMBER ["WGS 84 (TRANSIT)", ID ["EPSG", 1166]],
  MEMBER ["WGS 84 (G730)", ID ["EPSG", 1152]],
  MEMBER ["WGS 84 (G834)", ID ["EPSG", 1153]],
  MEMBER ["WGS 84 (G1150)", ID ["EPSG", 1154]],
  MEMBER ["WGS 84 (G1674)", ID ["EPSG", 1155]],
  MEMBER ["WGS 84 (G1762)", ID ["EPSG", 1156]],
  ELLIPSOID ["WGS 84", 6378137, 298.2572236, LENGTHUNIT ["metre", 1.0]],
  ENSEMBLEACCURACY [2]
]
```

EXAMPLE 3 Ensemble of vertical reference frames with ensemble member name only

```
ENSEMBLE ["EVRS ensemble",
  MEMBER ["EVRF2000"], MEMBER ["EVRF2007"],
  ENSEMBLEACCURACY [0.01]
]
```

7.7 Dynamic coordinate reference systems

Some types of coordinate reference system may be dynamic. A CRS is dynamic if its reference frame is dynamic. For a dynamic CRS the WKT for the dynamic attributes as described below is embedded within the CRS definition ([Clauses 8](#) and [10](#)).

In a dynamic CRS the coordinate values of a point on or near the surface of the Earth change with time. When a coordinate set is referenced to a dynamic CRS, to be unambiguous the coordinate set needs to additionally be referenced to a coordinate epoch. The WKT for coordinate epoch is described in [Clause 16](#).

Requirement: The WKT representation of the dynamic attributes of a dynamic coordinate reference system shall be:

```
<dynamic crs> ::= <dynamic crs keyword> <left delimiter>
  <frame reference epoch>
  [ <wkt separator> <deformation model ID> ]
  <right delimiter>

<dynamic crs keyword> ::= DYNAMIC
<frame reference epoch> ::= <frame reference epoch keyword> <left delimiter>
  <reference epoch> <right delimiter>
  !! Used when the CRS has a dynamic reference frame.

<frame reference epoch keyword> ::= FRAMEEPOCH
<reference epoch> ::= <unsigned integer> [ <period> [ <unsigned integer> ] ]
  !! See 6.3.2.
```

<deformation model ID>	::=	<deformation model ID keyword> <left delimiter> <deformation model name> [{ <wkt separator> <identifier> }]... <right delimiter> <i>!! Used when the CRS is associated with a deformation model or velocity grid. A full description may be given separately - see Clause 18. A full description of the deformation model shall not be embedded within the CRS WKT.</i>
<deformation model ID keyword>	::=	MODEL VELOCITYGRID <i>!! In this document for brevity the preferred keyword is MODEL. VELOCITYGRID is permitted. Implementations should be prepared to read both forms.</i> <i>Note that MODEL when used for a deformation model should not be confused with the reference to a geoid model for a geoid-based vertical CRS, for which the keyword is GEOIDMODEL: see Clause 10.</i>
<deformation model name>	::=	<quoted Latin text> <i>!! See 7.2.</i>

EXAMPLE 1 CRS with dynamic reference frame.

```
DYNAMIC [FRAMEEPOCH[2010.0]]
```

EXAMPLE 2 CRS with dynamic reference frame and deformation model.

```
DYNAMIC [FRAMEEPOCH[2010.0],MODEL["NAD83(CSR) v6 velocity grid"]]
```

8 WKT representation of geodetic and geographic coordinate reference systems

8.1 Overview

Requirement: The WKT representation of a geodetic or a geographic coordinate reference system shall be:

<geodetic crs>	::=	<static geodetic crs> <dynamic geodetic crs> <geographic crs>
<geographic crs>	::=	<static geographic crs> <dynamic geographic crs>
<static geodetic crs>	::=	<geodetic crs keyword> <left delimiter> <crs name> <wkt separator> { <geodetic reference frame> <geodetic datum ensemble> } <wkt separator> <coordinate system> <scope extent identifier remark> <right delimiter>
<static geographic crs>	::=	<geographic crs keyword> <left delimiter> <crs name> <wkt separator> { <geodetic reference frame> <geodetic datum ensemble> } <wkt separator> <coordinate system> <scope extent identifier remark> <right delimiter>
<dynamic geodetic crs>	::=	<geodetic crs keyword> <left delimiter> <crs name> <wkt separator> <dynamic crs> <wkt separator> <geodetic reference frame> <wkt separator> <coordinate system> <scope extent identifier remark> <right delimiter>

<code><dynamic geographic crs></code>	<code>::=</code>	<code><geographic crs keyword></code> <code><left delimiter></code> <code><crs name></code> <code><wkt separator></code> <code><dynamic crs></code> <code><wkt separator></code> <code><geodetic reference frame></code> <code><wkt separator></code> <code><coordinate system></code> <code><scope extent identifier remark></code> <code><right delimiter></code>
<code><geodetic crs keyword></code>	<code>::=</code>	GEODCRS GEODETTICRS <i>!! The <geodetic crs keyword> should be used when the CRS's coordinate system type is either Cartesian or spherical: see 8.3 and 7.5. In this document for brevity the preferred keyword is GEODCRS. GEODETTICRS is permitted. Implementations should be prepared to read both forms.</i>
<code><geographic crs keyword></code>	<code>::=</code>	GEOGCRS GEOGRAPHICCRS <i>!! The <geographic crs keyword> should be used when the CRS's coordinate system type is ellipsoidal: see 8.3 and 7.5. In this document for brevity the preferred keyword is GEOGCRS. GEOGRAPHICCRS is permitted. Implementations should be prepared to read both forms.</i>
<code><crs name></code>	<code>::=</code>	<code><quoted Latin text></code> <i>!! See 7.2.</i>

`<dynamic crs>` is described in 7.7. It is mandatory when the geodetic CRS or geographic CRS is dynamic, and shall not be given when the geodetic CRS or geographic CRS is static.

`<geodetic datum ensemble>` is described in 7.6, `<scope extent identifier remark>` is described in 7.3.

8.2 Geodetic reference frame (geodetic datum)

8.2.1 Ellipsoid

The `<ellipsoid>` object is an attribute of `<geodetic reference frame>`. It is not used with other types of datum.

Requirement: The WKT representation of an oblate ellipsoid shall be:

<code><ellipsoid></code>	<code>::=</code>	<code><ellipsoid keyword></code> <code><left delimiter></code> <code><ellipsoid name></code> <code><wkt separator></code> <code><semi-major axis></code> <code><wkt separator></code> <code><inverse flattening></code> [<code><wkt separator></code> <code><length unit></code>] [{ <code><wkt separator></code> <code><identifier></code> }]... <code><right delimiter></code>
<code><ellipsoid keyword></code>	<code>::=</code>	ELLIPSOID SPHEROID <i>!! In this document the preferred keyword is ELLIPSOID. SPHEROID is permitted for backward compatibility. Implementations should be prepared to read both forms.</i>
<code><ellipsoid name></code>	<code>::=</code>	<code><quoted Latin text></code> <i>!! See 7.2.</i>
<code><semi-major axis></code>	<code>::=</code>	<code><unsigned numeric literal></code> <i>!! See below for constraint.</i>
<code><inverse flattening></code>	<code>::=</code>	<code><unsigned numeric literal></code> <i>!! See below for constraint.</i>

ISO 19111 allows an oblate ellipsoid to be defined through semi-major axis (a) and either semi-minor axis (b) or inverse flattening ($1/f$). If semi-minor axis is used as the second defining parameter the value for inverse flattening to be shown in the WKT string should be calculated from $1/f = a/(a - b)$.

ISO 19111 also allows for the earth model to be a sphere, for which $1/f$ is infinite. In this document if the earth model is a sphere `<inverse flattening>` shall be given an artificial value of zero.

Requirements:

- a) The WKT representation of a sphere shall have an `<inverse flattening>` value of 0.

- b) <length unit> is an optional attribute, optional for reasons of backward compatibility, but it is recommended that it is explicitly included in WKT strings. Its <conversion factor> shall be to metres and is the number of metres per unit. <length unit> is described in 7.4. If it is omitted then the value for the length of the semi-axis or -axes shall be given in metres. Conversely, if it is omitted then the value for the semi-major axis shall be assumed to be in metres.

NOTE In the WKT for a geodetic, geographic or projected CRS, the length unit for the ellipsoid can differ from the length unit for the coordinate system. The units in which coordinates are expressed are given by the CS element.

Examples of WKT describing an ellipsoid:

EXAMPLE 1 ELLIPSOID["GRS 1980",6378137,298.257222101,LENGTHUNIT["metre",1.0]]

EXAMPLE 2 (unit = metre is implied)
SPHEROID["GRS 1980",6378137.0,298.257222101]

EXAMPLE 3 ELLIPSOID["Clark 1866",20925832.164,294.97869821,
LENGTHUNIT["US survey foot",0.304800609601219]]

EXAMPLE 4 ELLIPSOID["Sphere",6371000,0,LENGTHUNIT["metre",1.0]]

The definition of WKT for a triaxial ellipsoid required for planetary mapping is given in [Annex E](#).

8.2.2 Prime meridian

The prime meridian is the meridian with a value of zero longitude. The prime meridian is usually defined to be the international reference meridian which for the Earth passes near Greenwich, but this may not always be the case. The <prime meridian> object is conditional.

Requirements:

- a) Prime meridian shall not be given for any type of datum and CRS other than geodetic reference frame.

NOTE For projected CRSs the prime meridian is inherited through the base geodetic CRS.

- b) Prime meridian shall be given if the CRS type is geodetic or geographic and the prime meridian is not the international reference meridian. It may be given if the CRS type is geodetic or geographic and the prime meridian is the international reference meridian.
- c) Conversely if the CRS type is geodetic or geographic and prime meridian is not given, the prime meridian shall be assumed to be the international reference meridian.
- d) If the prime meridian's angle unit is omitted, the IRM longitude value shall be in the angular unit of the CRS containing the prime meridian when the CRS has an ellipsoidal CS, else it shall be in decimal degrees.
- e) The WKT representation of a prime meridian shall be:

```
<prime meridian> ::= <prime meridian keyword> <left delimiter>
<prime meridian name> <wkt separator> <irm longitude>
[ { <wkt separator> <identifier> } ]... <right delimiter>

<prime meridian keyword> ::= PRIMEM | PRIMEMERIDIAN

!! In this document the preferred keyword is PRIMEM
for backward compatibility. PRIMEMERIDIAN is permitted.
Implementations should be prepared to read both forms.

<prime meridian name> ::= <quoted Latin text> !! See 7.2.

<irm longitude> ::= <signed numeric literal> [ <wkt separator> <angle unit> ]

!! If <angle unit> is omitted, the <signed numeric
literal> value is to be in the CRS's CS angular unit if
available, else in decimal degrees.
```

<irm longitude> is the longitude of the prime meridian measured from the international reference meridian, positive eastward.

<angle unit> is an optional attribute, optional for reasons of backward compatibility, but best practice is that it is included in WKT strings. If it is omitted then the value for <irm longitude> shall be given in the CRS's <cs unit> where this is angular, else in decimal degrees. If the subtype of the geodetic CRS to which the prime meridian is an attribute is geographic, the prime meridian's <irm longitude> value shall be given in the same angular units as those for the horizontal axes of the geographic CRS; if the geodetic CRS subtype is geocentric the prime meridian's <irm longitude> value shall be given in degrees. Its <conversion factor> shall be to radians and is the number of radians per unit. <angle unit> is described in 7.4.

Examples of WKT describing a prime meridian:

EXAMPLE 1 PRIMEM["Paris",2.5969213,ANGLEUNIT["grad",0.015707963267949]]

EXAMPLE 2 PRIMEM["Ferro",-17.6666667]
(unit unspecified so it takes <cs unit> if angular, else decimal degrees)

EXAMPLE 3 PRIMEM["Greenwich",0.0, ANGLEUNIT["degree",0.0174532925199433]]

NOTE When the prime meridian name is "Greenwich" or the longitude of the international reference meridian in the geodetic CRS is zero, the inclusion of the prime meridian object in a WKT string is optional, so this example string should not normally be present. It is included here for backward compatibility reasons as the prime meridian object was mandatory in earlier CRS WKT specifications.

8.2.3 Geodetic reference frame (datum)

In this document a geodetic reference frame is either a modern *terrestrial reference frame* or a classical *geodetic datum*. For backward compatibility and to assist modular programming of different types of CRS, in this document 'datum' is retained in the WKT

Geodetic reference frames are either static or dynamic. If the reference frame is dynamic all CRSs based on it are dynamic CRSs. The *frame reference epoch* attribute is then required. However, a geodetic or geographic CRS may also be associated with a deformation model. In this document the *frame reference epoch* attribute of a dynamic reference frame is treated with the deformation model at the CRS level as described in 7.7 and 8.1.

Requirement: The WKT representation of a geodetic reference frame (geodetic datum) shall be:

```

<geodetic reference frame> ::= <geodetic reference frame keyword>
                             <left delimiter> <datum name> <wkt separator> <ellipsoid>
                             [ <wkt separator> <datum anchor> ]
                             [ { <wkt separator> <identifier> } ]... <right delimiter>
                             [ { <wkt separator> <prime meridian> } ]

<geodetic reference frame keyword> ::= DATUM | TRF | GEODETICDATUM

!! In this document for compatibility with
previous versions of WKT the preferred keyword
is DATUM, TRF is also permitted (see B.2.2 for
additional comment). For consistency with the
other datum types described in Clauses 10 to
13 either GEODETICDATUM or TRF is permitted.
Implementations should be prepared to read all
three forms.

<datum name> ::= <quoted Latin text> !! See 7.2.
<datum anchor> ::= <datum anchor keyword> <left delimiter>
                  <datum anchor description> <right delimiter>
<datum anchor keyword> ::= ANCHOR
    
```

<datum anchor description> ::= <quoted Latin text>

<geodetic reference frame> is used when the CRS type is geodetic or geographic. For a projected CRS, the geodetic reference frame (datum) is included in the Base Geodetic CRS, see [Clause 9](#).

<ellipsoid> is described in [8.2.1](#).

<prime meridian> is a conditional attribute; see [8.2.2](#). Following the data model of ISO 19111 it would be nested within the geodetic reference frame object. In this document its nesting follows previous versions of CRS WKT.

```
EXAMPLE 1 DATUM["North American Datum 1983",
            ELLIPSOID["GRS 1980",6378137,298.257222101,LENGTHUNIT["metre",1.0]]
          ]
```

```
EXAMPLE 2 TRF["World Geodetic System 1984",
            ELLIPSOID["WGS 84",6378388.0,298.257223563,LENGTHUNIT["metre",1.0]]
          ],
          PRIMEM["Greenwich",0.0]
```

NOTE When the prime meridian name is "Greenwich" or the longitude of the IRM meridian in the geodetic CRS is zero, the inclusion of the prime meridian object in a WKT string is optional, so this example string can omit the prime meridian object, as shown in Example 1.

```
EXAMPLE 3 GEODETICDATUM["Tananarive 1925",
                        ELLIPSOID["International 1924",6378388.0,297.0,LENGTHUNIT["metre",1.0]]
                      ],
          ANCHOR["Tananarive observatory:21.0191667gS, 50.23849537gE (of Paris)"]
          ],
          PRIMEM["Paris",2.5969213,ANGLEUNIT["grad",0.015707963267949]]
```

8.3 Coordinate systems for geodetic and geographic CRSs

<coordinate system> is described in [7.5](#). Several constraints and recommendations for coordinate systems used in geodetic and geographic CRSs are described there. These include the requirement for

- a geodetic CRS to contain either a 3D Cartesian or a 3D spherical coordinate system, and
- a geographic CRS to contain an ellipsoidal coordinate system, either 2D or 3D.

The previous version of this document did not distinguish between geodetic CRS and geographic CRS, and a geodetic CRS could have an ellipsoidal coordinate system. Implementations should be prepared to read a WKT string for a geodetic CRS which includes an ellipsoidal coordinate system. Implementations in conformance with this document should not write geodetic CRSs containing an ellipsoidal coordinate system.

8.4 Examples of WKT describing a geodetic or geographic CRS

EXAMPLE 1 Static geodetic CRS with Cartesian coordinate system and including scope, extent, ID and remarks:

```
GEODCRS["JGD2000",
  DATUM["Japanese Geodetic Datum 2000",
    ELLIPSOID["GRS 1980",6378137,298.257222101]
  ],
  CS[Cartesian,3],
  AXIS["(X)",geocentricX],
  AXIS["(Y)",geocentricY],
  AXIS["(Z)",geocentricZ],
  LENGTHUNIT["metre",1.0],
  USAGE[SCOPE["Geodesy, topographic mapping and cadastre"],
    AREA["Japan"],
    BBOX[17.09,122.38,46.05,157.64],
    TIMEEXTENT[2002-04-01,2011-10-21]],
  ID["EPSG",4946,URI["urn:ogc:def:crs:EPSG::4946"]],
  REMARK["注:JGD2000ジオセントリックは現在JGD2011に代わりました。"]
]
```

Non-Latin characters may only be included within remarks (see [7.3.4](#)).

EXAMPLE 2 Dynamic geographic CRS with ellipsoidal 3D coordinate system, no optional attributes:

```
GEOGCRS["WGS 84 (G1762)",
  DYNAMIC[FRAMEEPOCH[2005.0]],
  TRF["World Geodetic System 1984 (G1762)",
    ELLIPSOID["WGS 84",6378137,298.257223563,LENGTHUNIT["metre",1.0]
  ]
],
  CS[ellipsoidal,3],
  AXIS["(lat)",north,ANGLEUNIT["degree",0.0174532925199433]],
  AXIS["(lon)",east,ANGLEUNIT["degree",0.0174532925199433]],
  AXIS["ellipsoidal height (h)",up,LENGTHUNIT["metre",1.0]]
]
```

EXAMPLE 3 Static geographic CRS with IRM as prime meridian and ellipsoidal 2D coordinate system in degrees:

```
GEOGRAPHICCRS["NAD83",
  DATUM["North American Datum 1983",
    ELLIPSOID["GRS 1980",6378137,298.257222101,LENGTHUNIT["metre",1.0]]
  ],
  CS[ellipsoidal,2],
  AXIS["latitude",north],
  AXIS["longitude",east],
  ANGLEUNIT["degree",0.017453292519943],
  ID["EPSG",4269],
  REMARK["1986 realisation"]
]
```

EXAMPLE 4 Static geographic CRS with prime meridian other than IRM and ellipsoidal 2D coordinate system in grads:

```
GEOGCRS["NTF (Paris)",
  DATUM["Nouvelle Triangulation Francaise",
    ELLIPSOID["Clarke 1880 (IGN)",6378249.2,293.4660213]
  ],
  PRIMEM["Paris",2.5969213],
  CS[ellipsoidal,2],
  AXIS["latitude",north,ORDER[1]],
  AXIS["longitude",east,ORDER[2]],
  ANGLEUNIT["grad",0.015707963267949],
  REMARK["Nouvelle Triangulation Française"]
]
```

Non-Latin1 characters may only be included within remarks (see [7.3.4](#)).

9 WKT representation of projected CRSs

9.1 Overview

A projected CRS is a special case of a derived CRS. Because of its importance to geographic information and particularly for backward compatibility reasons necessitating a different string structure it is treated separately from the general case described in [Clause 14](#). The structures of WKT strings for principal and derived CRSs are compared in [14.1](#), to which the reader is referred.

Requirement: The WKT representation of a projected coordinate reference system shall be:

```
<projected crs> ::= <projected crs keyword> <left delimiter> <crs name>
                  <wkt separator> <base geodetic crs>
                  <wkt separator> <map projection>
                  <wkt separator> <coordinate system>
                  <scope extent identifier remark> <right delimiter>

<projected crs keyword> ::= PROJCRS | PROJECTEDCRS
                          !! In this document for brevity the preferred keyword is
                          PROJCRS. PROJECTEDCRS is permitted. Implementations should
                          be prepared to read both forms.

<crs name> ::= <quoted Latin text> !! See 7.2.
```

9.2 Base CRS

9.2.1 General

The base CRS for a projected CRS may be either a geodetic CRS or (more usually) a geographic CRS. A geographic 2D CRS may act as the base CRS for only another 2D CRS.

Requirement: The dimension of a derived CRS shall be equal to or less than the dimension of its base CRS.

Requirement: The WKT representation of a projected coordinate reference system's base CRS shall be:

```
<base geodetic crs> ::= <base static geodetic crs> | <base dynamic geodetic crs>
                    <base static geographic crs> | <base dynamic geographic crs>
                    !! The projected CRS takes its static/dynamic status
                    from its base CRS.
```

<code><base static geodetic crs></code>	<code>::=</code>	<code><base geodetic crs keyword> <left delimiter> <base crs name></code> <code><wkt separator></code> <code>{ <geodetic reference frame> <geodetic datum ensemble> }</code> <code>[<wkt separator> <ellipsoidal cs unit>]</code> <code>[{ <wkt separator> <identifier> }]... <right delimiter></code>
<code><base dynamic geodetic crs></code>	<code>::=</code>	<code><base geodetic crs keyword> <left delimiter> <base crs name></code> <code><wkt separator> <dynamic crs></code> <code><wkt separator> <geodetic reference frame></code> <code>[<wkt separator> <ellipsoidal cs unit>]</code> <code>[{ <wkt separator> <identifier> }]... <right delimiter></code>
<code><base static geographic crs></code>	<code>::=</code>	<code><base geographic crs keyword> <left delimiter></code> <code><base crs name> <wkt separator></code> <code>{ <geodetic reference frame> <geodetic datum ensemble> }</code> <code>[{ <wkt separator> <ellipsoidal cs unit> }]</code> <code>[{ <wkt separator> <identifier> }]... <right delimiter></code>
<code><base dynamic geographic crs></code>	<code>::=</code>	<code><base geographic crs keyword> <left delimiter></code> <code><base crs name> <wkt separator> <dynamic crs></code> <code><wkt separator> <geodetic reference frame></code> <code>[{ <wkt separator> <ellipsoidal cs unit> }]</code> <code>[{ <wkt separator> <identifier> }]... <right delimiter></code>
<code><base geodetic crs keyword></code>	<code>::=</code>	BASEGEODCRS
<code><base geographic crs keyword></code>	<code>::=</code>	BASEGEOGCRS
		<p><i>!! In this document the base CRS keyword should reflect the type of CRS (geodetic or geographic) upon which the projected CRS is referenced. Usually this will be BASEGEOGCRS. The previous version of this document used only BASEGEODCRS. Implementations should be prepared to read both forms.</i></p>
<code><base crs name></code>	<code>::=</code>	<code><quoted Latin text></code> <i>!! See 7.2.</i>
<code><ellipsoidal cs unit></code>	<code>::=</code>	<code><angle unit></code>

`<dynamic crs>` is described in 7.7. It is mandatory when the base CRS is dynamic and should not be given when the base CRS is static. The projected CRS has the same static/dynamic status as its base CRS.

`<geodetic reference frame>` is described in 8.2. It includes ellipsoid and prime meridian descriptions. `<geodetic datum ensemble>` is described in 7.6.

NOTE Constructs similar to `<base static geodetic crs>`, `<base dynamic geodetic crs>`, `<base static geographic crs>` and `<base dynamic geographic crs>` but excluding the `<ellipsoidal cs unit>` element are also used for derived geodetic and derived geographic CRSs, see 14.3.

9.2.2 Ellipsoidal CS unit

The base geographic CRS `<ellipsoidal cs unit>` is that in which geodetic latitude and longitude would be quoted in the geographic CRS. It is defined as an optional attribute for backward compatibility reasons (see 9.3).

`<angle unit>` is described in 7.4.

Requirement: The `<ellipsoidal cs unit>` of the base geographic CRS shall be included in projected CRS WKT strings when the units of the map projection angular parameters are not explicitly given within those parameters.

Recommendation: The units of the map projection angular parameters should be explicitly included in the projected CRS WKT string and when this is done the <ellipsoidal cs unit> of the base geographic CRS should not be included in the string.

9.3 Map projection

9.3.1 Introduction

Map projection is a deriving conversion (see 14.2). By definition the source CRS is the base geodetic or geographic CRS and the target CRS is projected CRS; these are therefore implied rather than explicitly stated in the WKT string. Additionally, for reasons of backward compatibility, in map projections the parameter units may be implied; map projection is therefore a specialised subset of deriving conversion.

Requirement: The WKT representation of a map projection shall be:

<map projection>	::=	<map projection keyword> <left delimiter> <map projection name> <wkt separator> <map projection method> [{ <wkt separator> <map projection parameter> }]... [{ <wkt separator> <identifier> }]... <right delimiter>
<map projection keyword>	::=	CONVERSION
<map projection name>	::=	<quoted Latin text> <i>!! See 7.2.</i>
<map projection method>	::=	<map projection method keyword> <left delimiter> <map projection method name> [{ <wkt separator> <identifier> }]... <right delimiter>
<map projection method keyword>	::=	METHOD PROJECTION <i>In this document the preferred keyword is METHOD. PROJECTION is permitted for backward compatibility. Implementations should be prepared to read both forms.</i>
<map projection method name>	::=	<quoted Latin text> <i>!! See 7.2.</i>
<map projection parameter>	::=	<parameter keyword> <left delimiter> <parameter name> <wkt separator> <parameter value> [<wkt separator> <map projection parameter unit>] [{ <wkt separator> <identifier> }]... <right delimiter>
<parameter keyword>	::=	PARAMETER
<parameter name>	::=	<quoted Latin text> <i>!! See 7.2.</i>
<parameter value>	::=	<signed numeric literal>

<map projection parameter unit> ::= <length unit> | <angle unit> | <scale unit>
 !! See 9.3.4 for constraints.

EXAMPLE 1 Identifier given for the method and each parameter.

```
CONVERSION["UTM zone 10N",
  METHOD["Transverse Mercator", ID["EPSG", 9807]],
  PARAMETER["Latitude of natural origin", 0,
    ANGLEUNIT["degree", 0.0174532925199433],
    ID["EPSG", 8801]],
  PARAMETER["Longitude of natural origin", -123,
    ANGLEUNIT["degree", 0.0174532925199433], ID["EPSG", 8802]],
  PARAMETER["Scale factor at natural origin", 0.9996,
    SCALEUNIT["unity", 1.0], ID["EPSG", 8805]],
  PARAMETER["False easting", 500000,
    LENGTHUNIT["metre", 1.0], ID["EPSG", 8806]],
  PARAMETER["False northing", 0, LENGTHUNIT["metre", 1.0], ID["EPSG", 8807]]
]
```

EXAMPLE 2 Identifier given for conversion as a whole.

```
CONVERSION["UTM zone 10N",
  METHOD["Transverse Mercator"],
  PARAMETER["Latitude of natural origin", 0,
    ANGLEUNIT["degree", 0.0174532925199433]],
  PARAMETER["Longitude of natural origin", -123,
    ANGLEUNIT["degree", 0.0174532925199433]],
  PARAMETER["Scale factor at natural origin", 0.9996,
    SCALEUNIT["unity", 1.0]],
  PARAMETER["False easting", 500000, LENGTHUNIT["metre", 1.0]],
  PARAMETER["False northing", 0, LENGTHUNIT["metre", 1.0]],
  ID["EPSG", 16010]
]
```

Further examples are included in [9.5](#).

9.3.2 Map projection name and identifier

Map projection (zone) encompasses the collection of method and parameter values. Its name is for human readability. Depending upon the naming convention in use it may also be included as part of the projected CRS name.

<identifier> is described in [7.3.4](#). It is an optional attribute. If an identifier is provided as an attribute within the <map projection conversion> object, because it is expected to describe a complete collection of zone name, method, parameters and parameter values, it shall override any identifiers given within the map projection method and map projection parameter objects.

9.3.3 Map projection method

Method name is for human readability. For interoperability it is the method formula that is critical in determining the equivalence of methods; this may be given through a map projection method <identifier>.

<identifier> is described in [7.3.4](#). It is an optional attribute. However, if an identifier is included as an attribute within the <map projection> object ([9.3.1](#)), it will take precedence over any identifier within the map projection method object.

If a map projection method <identifier> is not given, the WKT description is potentially ambiguous, relying on interpretation of method name. It is recommended that a map projection identifier or a map projection method identifier is included in WKT strings. Identifiers for commonly encountered map projection methods are given in [F.2](#).

9.3.4 Map projection parameter

Parameter name is for human readability. For interoperability it is the method formula and its parameters that are critical in determining the equivalence of methods. See [Annex F](#). Identifiers for commonly encountered map projection methods are given in [F.2](#); their parameters are listed in [F.3](#).

The map projection parameters required are specific to the map projection method and will be listed sequentially. The order within the sequence is not significant but should be logical.

<map projection parameter unit> is an optional attribute, for reasons of backward compatibility. Best practice is that it is included explicitly in WKT strings.

Requirements: If <map projection parameter unit> is omitted from <map projection parameter> then:

- a) Map parameter values that are lengths shall be given in metres.
- b) Map projection parameter values that are angles shall be given in decimal degrees.
- c) Map projection parameters that are unitless (for example scale factor) shall be given as a number which is close to or is unity (1.0).

NOTE The original parameter values can be in units other than metres, degrees and unity. For this reason, best practice is that units are not omitted.

The parameter unit type is included in [F.3](#).

<identifier> is described in [7.3.4](#). It is an optional attribute. If an identifier is included as an attribute within the *map projection conversion* object ([9.3.1](#)) or *map projection method* object ([9.3.3](#)), it will take precedence over any identifier within the map projection parameter object.

9.4 Coordinate systems for projected CRSs

<coordinate system> is described in [7.5](#). Several constraints and recommendations for coordinate systems used in projected CRSs are described there. These include the requirement for a projected CRS to contain a Cartesian coordinate system.

9.5 Examples of WKT describing a projected CRS

EXAMPLE 1

```
PROJCRS["ETRS89 Lambert Azimuthal Equal Area CRS",
  BASEGEOGCRS["ETRS89",
    DATUM["ETRS89",
      ELLIPSOID["GRS 80",6378137,298.257222101,LENGTHUNIT["metre",1.0]]
    ],ID["EuroGeographics","ETRS89-LatLon"]
  ],
  CONVERSION["LAEA",
    METHOD["Lambert Azimuthal Equal Area",ID["EPSG",9820]],
    PARAMETER["Latitude of origin",52.0,
      ANGLEUNIT["degree",0.0174532925199433]],
    PARAMETER["Longitude of origin",10.0,
      ANGLEUNIT["degree",0.0174532925199433]],
    PARAMETER["False easting",4321000.0,LENGTHUNIT["metre",1.0]],
    PARAMETER["False northing",3210000.0,LENGTHUNIT["metre",1.0]]
  ],
  CS[Cartesian,2],
  AXIS["(Y)",north,ORDER[1]],
  AXIS["(X)",east,ORDER[2]],
  LENGTHUNIT["metre",1.0],
  USAGE[SCOPE["Description of a purpose"],AREA["An area description"]],
  ID["EuroGeographics","ETRS-LAEA"]
]
```

EXAMPLE 2

```

PROJCRS["NAD27 / Texas South Central",
  BASEGEOGCRS["NAD27",
    DATUM["North American Datum 1927",
      ELLIPSOID["Clarke 1866",20925832.164,294.97869821,
        LENGTHUNIT["US survey foot",0.304800609601219]]
    ],
  ],
  CONVERSION["Texas South Central SPCS27",
    METHOD["Lambert Conic Conformal (2SP)",ID["EPSG",9802]],
    PARAMETER["Latitude of false origin",27.8333333333333,
      ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8821]],
    PARAMETER["Longitude of false origin",-99.0,
      ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8822]],
    PARAMETER["Latitude of 1st standard parallel",28.3833333333333,
      ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8823]],
    PARAMETER["Latitude of 2nd standard parallel",30.2833333333333,
      ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8824]],
    PARAMETER["Easting at false origin",2000000.0,
      LENGTHUNIT["US survey foot",0.304800609601219],ID["EPSG",8826]],
    PARAMETER["Northing at false origin",0.0,
      LENGTHUNIT["US survey foot",0.304800609601219],ID["EPSG",8827]]
  ],
  CS[Cartesian,2],
  AXIS["(X)",east],
  AXIS["(Y)",north],
  LENGTHUNIT["US survey foot",0.304800609601219],
  REMARK["Fundamental point: Meade's Ranch KS, latitude 39°13'26.686"N,
    longitude 98°32'30.506"W."]]
]

```

EXAMPLE 3 This example is included for backwards compatibility to show the implicit description of parameter units. It is recommended that units are explicitly given in the string, as in the previous two examples.

```

PROJCRS["NAD83 UTM 10",
  BASEGEOGCRS["NAD83(86)",
    DATUM["North American Datum 1983",
      ELLIPSOID["GRS 1980",6378137,298.257222101]], (default length unit is metre)
    PRIMEM["Greenwich",0], (implied angle unit is degree)
  ],
  CONVERSION["UTM zone 10N",
    METHOD["Transverse Mercator"],
    PARAMETER["Latitude of natural origin",0.0], (implied angle unit is degree)
    PARAMETER["Longitude of natural origin",-123.0], (implied angle unit is degree)
    PARAMETER["Scale factor",0.9996], (implied scale unit is unity)
    PARAMETER["False easting",500000.0], (implied length unit is degree)
    PARAMETER["False northing",0.0], (implied length unit is degree)
  ],
  ID["EPSG",16010]],
  CS[Cartesian,2],
  AXIS["(E)",east,ORDER[1]],
  AXIS["(N)",north,ORDER[2]],
  LENGTHUNIT["metre",1.0],
  REMARK["In this example parameter value units are not given. This is
    allowed for backward compatibility. However it is strongly recommended
    that units are explicitly given in the string, as in the previous two
    examples."]]

```

EXAMPLE 4 3D Projected CRS

```

PROJCRS["WGS 84 (G1762) / UTM zone 31N 3D",
  BASEGEOGCRS["WGS 84",
    DATUM["World Geodetic System of 1984 (G1762)",
      ELLIPSOID["WGS 84",6378137, 298.257223563,LENGTHUNIT["metre",1.0]]
    ]
  ],
  CONVERSION["UTM zone 31N 3D",
    METHOD["Transverse Mercator (3D)"],
    PARAMETER["Latitude of origin",0.0,
      ANGLEUNIT["degree",0.0174532925199433]],
    PARAMETER["Longitude of origin",3.0,
      ANGLEUNIT["degree",0.0174532925199433]],
    PARAMETER["Scale factor",0.9996,SCALEUNIT["unity",1.0]],
    PARAMETER["False easting",500000.0,LENGTHUNIT["metre",1.0]],
    PARAMETER["False northing",0.0,LENGTHUNIT["metre",1.0]]
  ],
  CS[Cartesian,3],
    AXIS["(E)",east,ORDER[1]],
    AXIS["(N)",north,ORDER[2]],
    AXIS["ellipsoidal height (h)",up,ORDER[3]],
    LENGTHUNIT["metre",1.0]
  ]

```

10 WKT representation of vertical CRSs

10.1 Overview

Requirement: The WKT representation of a vertical coordinate reference system shall be:

<vertical crs>	::=	<static vertical crs> <dynamic vertical crs>
<static vertical crs>	::=	<vertical crs keyword> <left delimiter> <crs name> <wkt separator> { <vertical reference frame> <vertical datum ensemble> } <wkt separator> <coordinate system> [<wkt separator> <geoid model ID>] <scope extent identifier remark> <right delimiter>
<dynamic vertical crs>	::=	<vertical crs keyword> <left delimiter> <crs name> <wkt separator> <dynamic crs> <wkt separator> <vertical reference frame> <wkt separator> <coordinate system> [<wkt separator> <geoid model ID>] <scope extent identifier remark> <right delimiter>
<geoid model ID>	::=	<geoid model keyword> <left delimiter> <geoid model name> [<wkt separator> <identifier>] <right delimiter>
<geoid model ID keyword>	::=	GEOIDMODEL
<geoid model name>	::=	<quoted Latin text> <i>!! See 7.2.</i>

!! This information identifies the geoid model for a geoid-based vertical CRS. A full description may be given separately- see Clause 17. A full description of the geoid model shall not be embedded within the vertical CRS WKT.

<vertical crs keyword> ::= VERTCRS | VERTICALCRS
!! In this document for brevity the preferred keyword is VERTCRS. VERTICALCRS is permitted. Implementations should be prepared to read both forms.

<crs name> ::= <quoted Latin text> *!! See 7.2.*

<dynamic crs> is described in 7.7. It is mandatory when the vertical CRS is dynamic, and shall not be given when the vertical CRS is static.

<vertical datum ensemble> is described in 7.6, <scope extent identifier remark> is described in 7.3.

10.2 Vertical reference frame (vertical datum)

Modern geodetic terminology is to use the term *vertical reference frame*. Previous versions of this document used *vertical datum*. For backward compatibility and to assist modular programming of different types of CRS, in this document 'datum' is retained in the WKT.

Vertical reference frames are either static or dynamic. If the reference frame is dynamic all CRSs based on it are dynamic CRSs. In this document the *frame reference epoch* attribute of a dynamic vertical reference frame is treated at the CRS level as described in 7.7 and 10.1.

Requirement: The WKT representation of a vertical reference frame (vertical datum) shall be:

<vertical reference frame> ::= <vertical reference frame keyword> <left delimiter>
 <datum name> [<wkt separator> <datum anchor>]
 [{ <wkt separator> <identifier> }]... <right delimiter>

<vertical reference frame keyword> ::= VDATUM | VRF | VERTICALDATUM
!! In this document for consistency with other datum types described in this document the preferred keyword is VDATUM, but either VRF or VERTICALDATUM is permitted. Implementations should be prepared to read all three forms. See B.2.2 for additional comment.

<datum name> ::= <quoted Latin text> *!! See 7.2.*

<datum anchor> ::= <datum anchor keyword> <left delimiter>
 <datum anchor description> <right delimiter>

<datum anchor keyword> ::= ANCHOR

<datum anchor description> ::= <quoted Latin text>

EXAMPLE 1 VDATUM["Newlyn"]

EXAMPLE 2 VERTICALDATUM["Newlyn",ANCHOR["Mean Sea Level 1915 to 1921."]]

10.3 Vertical coordinate system

<coordinate system> is described in 7.5. Several constraints and recommendations for coordinate systems used in vertical CRSs are described there. These include the requirement for a vertical CRS to contain a vertical coordinate system.

10.4 Example of WKT describing a vertical CRS

EXAMPLE 1 Static vertical CRS

```
VERTCRS["NAVD88",
  VDATUM["North American Vertical Datum 1988"],
  CS[vertical,1],
  AXIS["gravity-related height (H)",up],
  LENGTHUNIT["metre",1.0]
]
```

EXAMPLE 2 Static geoid-based vertical CRS

```
VERTCRS["CGVD2013",
  VRF["Canadian Geodetic Vertical Datum of 2013"],
  CS[vertical,1],
  AXIS["gravity-related height (H)",up],
  LENGTHUNIT["metre",1.0],
  GEOIDMODEL["CGG2013",ID["EPSG",6648]]
]
```

EXAMPLE 3 Dynamic vertical CRS

```
VERTCRS["RH2000",
  DYNAMIC[FRAMEEPOCH[2000.0],MODEL["NKG2016LU"]],
  VDATUM["Rikets Hojdsystem 2000"],
  CS[vertical,1],
  AXIS["gravity-related height (H)",up],
  LENGTHUNIT["metre",1.0]
]
```

11 WKT representation of engineering CRSs

11.1 Overview

Requirement: The WKT representation of an engineering coordinate reference system shall be:

```
<engineering crs> ::= <engineering crs keyword> <left delimiter> <crs name>
                    <wkt separator> <engineering datum>
                    <wkt separator> <coordinate system>
                    <scope extent identifier remark> <right delimiter>

<engineering crs keyword> ::= ENGCRS | ENGINEERINGCRS

!! In this document for brevity the preferred keyword
is ENGCRS. ENGINEERINGCRS is permitted. Implementations
should be prepared to read both forms.

<crs name> ::= <quoted Latin text> !! See 7.2.
```

<scope extent identifier remark> is described in [7.3](#).

11.2 Engineering datum

Requirement: The WKT representation of an engineering datum shall be:

```
<engineering datum> ::= <engineering datum keyword> <left delimiter> <datum name>
                        [ <wkt separator> <datum anchor> ]
                        [ { <wkt separator> <identifier> } ]... <right delimiter>
```

<engineering datum keyword>	::=	EDATUM ENGINEERINGDATUM	
			<i>!! In this document for brevity the preferred keyword is EDATUM, but ENGINEERINGDATUM is permitted. Implementations should be prepared to read both forms.</i>
<datum name>	::=	<quoted Latin text>	<i>!! See 7.2.</i>
<datum anchor>	::=	<datum anchor keyword> <left delimiter>	
		<datum anchor description> <right delimiter>	
<datum anchor keyword>	::=	ANCHOR	
<datum anchor description>	::=	<quoted Latin text>	

11.3 Coordinate systems for engineering CRSs

<coordinate system> is described in 7.5. Several constraints and recommendations for coordinate systems used in engineering CRSs are described there. These include the requirement for an engineering CRS to contain contain an affine, Cartesian, cylindrical, linear, ordinal, polar or spherical coordinate system.

11.4 Examples of WKT describing an engineering CRS

EXAMPLE 1 `ENGCRS["A construction site CRS",
EDATUM["P1",ANCHOR["Peg in south corner"]],
CS[Cartesian,2],
 AXIS["site east",southWest,ORDER[1]],
 AXIS["site north",southEast,ORDER[2]],
 LENGTHUNIT["metre",1.0],
USAGE[SCOPE["Construction"],TIMEEXTENT["date/time t1","date/time t2"]]
]`

EXAMPLE 2 `ENGINEERINGCRS["Astra Minas Grid",
ENGINEERINGDATUM["Astra Minas"],
CS[Cartesian,2],
 AXIS["northing (X)",north,ORDER[1]],
 AXIS["westing (Y)",west,ORDER[2]],
 LENGTHUNIT["metre",1.0],
ID["EPSG",5800]
]`

EXAMPLE 3 `ENGCRS["A ship-centred CRS",
EDATUM["Ship reference point",ANCHOR["Centre of buoyancy"]],
CS[Cartesian,3],
 AXIS["(x)",forward],
 AXIS["(y)",starboard],
 AXIS["(z)",down],
 LENGTHUNIT["metre",1.0]
]`

EXAMPLE 4 **A continuous (not discrete) CRS for an image.**
`ENGCRS["An analogue image CRS",
EDATUM["Image reference point",ANCHOR["Top left corner of
image = 0,0"]],
CS[Cartesian,2],
 AXIS["Column (x)",columnPositive],
 AXIS["Row (y)",rowPositive],
 LENGTHUNIT["micrometre",1E-6]
]`

EXAMPLE 5 A discrete CRS for an image.

```
ENGCRS["A digital image CRS",
  EDATUM["Image reference point",ANCHOR["Top left corner of
  image = 0,0"]],
  CS[ordinal,2],
  AXIS["Column pixel (x)",columnPositive,ORDER[1]],
  AXIS["Row pixel (y)",rowPositive,ORDER[2]]
]
```

12 WKT representation of parametric CRSs

12.1 Overview

Requirement: The WKT representation of a parametric coordinate reference system shall be:

```
<parametric crs> ::= <parametric crs keyword> <left delimiter> <crs name>
                  <wkt separator> <parametric datum>
                  <wkt separator> <coordinate system>
                  <scope extent identifier remark> <right delimiter>

<parametric crs keyword> ::= PARAMETRICCRS
<crs name> ::= <quoted Latin text> !! See 7.2.
```

<scope extent identifier remark> is described in [7.3](#).

12.2 Parametric datum

Requirement: The WKT representation of a parametric datum shall be:

```
<parametric datum> ::= <parametric datum keyword> <left delimiter>
                    <datum name>
                    [ <wkt separator> <datum anchor> ]
                    [ { <wkt separator> <identifier> } ]... <right delimiter>

<parametric datum keyword> ::= PDATUM | PARAMETRICDATUM
                               !! In this document for brevity the preferred keyword
                               is PDATUM, but PARAMETRICLDATUM is permitted.
                               Implementations should be prepared to read both forms.

<datum name> ::= <quoted Latin text> !! See 7.2.
<datum anchor> ::= <datum anchor keyword> <left delimiter>
                  <datum anchor description> <right delimiter>

<datum anchor keyword> ::= ANCHOR
<datum anchor description> ::= <quoted Latin text>
```

12.3 Parametric coordinate system

<coordinate system> is described in [7.5](#). Several constraints and recommendations for coordinate systems used in parametric CRSs are described there. These include the requirement for a parametric CRS to contain a parametric coordinate system.

12.4 Example of WKT describing a parametric CRS

```
EXAMPLE PARAMETRICCRS["WMO standard atmosphere layer 0",
  PDATUM["Mean Sea Level",ANCHOR["1013.25 hPa at 15°C"]],
  CS[parametric,1],
  AXIS["pressure (hPa)",up],PARAMETRICUNIT["HectoPascal",100.0]
]
```

13 WKT representation of temporal CRSs

13.1 Temporal CRS

The document supports three subtypes of temporal CRS: `temporalDateTime` (date and time using the representation defined in ISO 8601-1), `temporalCount` (when the temporal quantities are integers) and `temporalMeasure` (when the temporal quantities are real numbers). In WKT strings this subtyping is exposed through the coordinate system element.

Requirement: The WKT representation of a temporal coordinate reference system shall be:

```
<temporal crs> ::= <temporal crs keyword> <left delimiter> <crs name>
                <wkt separator> <temporal datum>
                <wkt separator> <coordinate system>
                <scope extent identifier remark> <right delimiter>
<temporal crs keyword> ::= TIMECRS
<crs name> ::= <quoted Latin text> !! See 7.2.
```

<scope extent identifier remark> is described in [7.3](#).

13.2 Temporal datum

Requirement: The WKT representation of a temporal datum shall be:

```
<temporal datum> ::= <temporal datum keyword> <left delimiter> <datum name>
                    [ <wkt separator> <calendar> ]
                    [ <wkt separator> <temporal origin> ]
                    [ { <wkt separator> <identifier> } ]... <right delimiter>
<temporal datum keyword> ::= TDATUM | TIMEDATUM
                            !! In this document for brevity the preferred keyword
                            is TDATUM, but TIMEDATUM is permitted. Implementations
                            should be prepared to read both forms.
<datum name> ::= <quoted Latin text> !! See 7.2.
<temporal origin > ::= <temporal origin keyword> <left delimiter>
                    <temporal origin description> <right delimiter>
<temporal origin keyword> ::= TIMEORIGIN
<temporal origin description> ::= <datetime> | <quoted Latin text>
<calendar> ::= <calendar keyword> <left delimiter>
                <calendar identifier> <right delimiter>
<calendar keyword> ::= CALENDAR
<calendar identifier> ::= <quoted Latin text>
```

In this document the “proleptic Gregorian” calendar as defined in ISO 8601-1 is the only standardised entry for the calendar identifier. If the calendar element is omitted, “proleptic Gregorian” is the assumed value. If the temporal origin element is omitted, then “1875-05-20” is the assumed default value.

NOTE The default value for temporal origin of 1875-05-20 is the date of signing of the Convention du Mètre, an international treaty that was signed in Paris on 20 May 1875 by representatives of 17 nations (Argentina, Austria-Hungary, Belgium, Brazil, Denmark, France, Germany, Italy, Peru, Portugal, Russia, Spain, Sweden and Norway, Switzerland, Turkey, United States of America, and Venezuela). The treaty created the International Bureau of Weights and Measures (BIPM), an intergovernmental organization under the authority of the General Conference on Weights and Measures (CGPM) and under the supervision of the International Committee for Weights and Measures (CIPM), which coordinates international metrology and the development of the metric system. The Gregorian calendar was introduced in October 1582. Adoption of this calendar has occurred in various countries over a 400-year period.

EXAMPLE 1

Temporal datum in which Calendar included.

```
TIMEDATUM["Gregorian calendar",CALENDAR["proleptic Gregorian"],
TIMEORIGIN[0000-01-01]]
```

EXAMPLE 2

Temporal datum with Calendar omitted so should be assumed to be "proleptic Gregorian".

```
TDATUM["Gregorian calendar",TIMEORIGIN["0001 January 1st"]]
```

EXAMPLE 3

Temporal datum with Calendar omitted so should be assumed to be "proleptic Gregorian", and with TimeOrigin omitted so should be assumed to be 1875-05-20. This will normally be used with a temporalDateTime coordinate system.

```
TDATUM["Gregorian calendar"]
```

13.3 Temporal coordinate system

13.3.1 General

<coordinate system> is described in 7.5. Several general constraints and recommendations for coordinate systems are described there. Those in 7.5.1 to 7.5.5 apply to temporal coordinate systems. Temporal coordinate systems treat <axisUnit> and <conversion factor> differently from other types of coordinate system and their additional constraints are described here.

13.3.2 Axis unit for temporalDateTime coordinate systems

The dateTime syntax is a representation of a compound string including multiple temporal quantities. Its components are defined in ISO 8601-1. The syntax for a dateTime value in WKT is given in 6.3. The syntax for a temporalDateTime coordinate system in WKT is given in 7.5.1; no axis unit is included.

Requirement: For a temporalDateTime coordinate system, neither <axis unit> nor <cs unit> shall be given.

NOTE This requirement also applies to ordinal coordinate systems, see 7.5.6.3.

Per 7.5.1 to 7.5.5, for a temporalDateTime CS axis name/abbreviation and axis direction shall be given, axis order may be given or be implied. Only axis unit is omitted.

13.3.3 Axis unit for temporalCount and temporalMeasure coordinate systems

For temporalCount and temporalMeasure coordinate systems, the requirements for coordinate system and coordinate system axis are the same as those for a spatial coordinate system and spatial coordinate system axis except that the axis unit conversion factor is conditional, to be given only when the temporal quantity (time unit) has a simple scalar conversion ratio to the SI base unit of a second. Requirements and examples are in 7.4.1 and 7.4.3.

13.4 Examples of WKT describing a temporal CRS

EXAMPLE 1 Temporal CRS with a temporalDateTime coordinate system (axis time unit shall not be given). Calendar attribute is omitted and time origin attribute is omitted (defaults inferred in both cases):

```
TIMECRS["DateTime",
TDATUM["Gregorian Calendar"],
CS[TemporalDateTime,1],AXIS["Time (T)",future]
]
```

NOTE Example coordinate data suitable for use with this CRS:
(2018-04-25T16:30:00Z, 2018-05-01T09:35:24Z, ...).

EXAMPLE 2 Temporal CRS with a temporalCount CS (coordinates are integers) in which axis unit includes a scalar conversion factor:

```
TIMECRS["GPS milliseconds",
  TDATUM["GPS time origin", TIMEORIGIN[1980-01-01T00:00:00.0Z]],
  CS[TemporalCount, 1], AXIS["(T)", future, TIMEUNIT["millisecond
  (ms)", 0.001]]
]
```

NOTE Example coordinate data suitable for use with this CRS:
(5351236450450, 5351236450950, ...).

EXAMPLE 3 Temporal CRS with a temporalCount CS in which axis quantity is integer count. Axis unit is used but conversion factor is omitted because not scalar.

```
TIMECRS["Calendar hours from 1979-12-29",
  TDATUM["29 December 1979", TIMEORIGIN[1979-12-29T00]],
  CS[TemporalCount, 1], AXIS["Time", future, TIMEUNIT["hour"]]
]
```

NOTE Example coordinate data suitable for use with this CRS: (429, 453, ...).

EXAMPLE 4 Temporal CRS with a temporalMeasure CS (coordinates are real numbers) in which <axis unit> is used but <conversion factor> is omitted because not scalar:

```
TIMECRS["Decimal Years CE",
  TDATUM["Common Era", TIMEORIGIN[0000]],
  CS[TemporalMeasure, 1], AXIS["Decimal years
  (a)", future, TIMEUNIT["year"]]
]
```

NOTE Example coordinate data suitable for use with this CRS: (2000.4, 2002.8, ...).

EXAMPLE 5 Temporal CRS with a temporalCount CS (coordinates are integers). <axis unit> is used but <conversion factor> is omitted because not scalar:

```
TIMECRS["Unix time",
  TDATUM["Unix epoch", TIMEORIGIN[1970-01-01T00:00:00Z]],
  CS[TemporalCount, 1], AXIS["Time", future, TIMEUNIT["second"]]
]
```

NOTE The timeunit second is a calendar second, not an SI second; leap seconds are not applied^[Z]. Example coordinate data suitable for use with this CRS:
(1528368006, 1528368219, ...).

14 WKT representation of derived CRSs

14.1 Overview

ISO 19111 includes a modelling concept of a derived CRS. A derived CRS is a CRS which cannot exist in its own right but is defined through a coordinate conversion from another coordinate reference system.

A projected CRS is a special case of a derived CRS. Because of its importance to geographic information and particularly for backward compatibility reasons necessitating a different string structure it is treated separately from the general case and is described in [Clause 9](#).

NOTE Georeferenced image coordinates are not referenced to a derived CRS but to an engineering CRS. The georeferencing is accomplished through a coordinate transformation between the image's engineering CRS and the (usually geodetic or projected) CRS to which the image is georeferenced.

This document implements this modelling concept through the inclusion of a 'base CRS' description within the description of the CRS. [Table 4](#) compares the general structure of WKT for the principal CRS types described in [Clauses 8](#) and [10](#) to [14](#) with the general structure of WKT for a derived CRS.

Table 4 — Comparison of general structure of WKT for principal and derived CRSs

Structure for principal CRSs	Structure for derived CRSs
a) keyword	a) keyword
b) name	b) name
c) datum or datum ensemble	c) base CRS including its datum or datum ensemble
	d) deriving conversion
d) coordinate system	e) coordinate system
e) optional metadata	f) optional metadata

In derived CRSs the base CRS detail includes the name and datum of the base CRS from which the derived CRS is derived. The base CRS detail does not include optional metadata or the coordinate system information for that base CRS, but the dimension of the base CRS's coordinate system constrains the dimension of the derived CRS.

Requirement: The dimension of a derived CRS shall be equal to or less than the dimension of its base CRS.

14.2 Deriving conversion

14.2.1 General

The deriving conversion in a derived CRS is a specialised case of coordinate conversion described in [Clause 17](#) in which the source and target CRSs are by definition the base CRS and the derived CRS respectively; these are therefore implied rather than explicitly stated in the WKT string. Because coordinate conversions are exact, the attribute operation accuracy is not relevant and excluded from the deriving conversion WKT string. A map projection is a special case of a deriving conversion (see [9.3](#)).

Requirement: The WKT representation of a <deriving conversion> shall be:

```

<deriving conversion > ::= <deriving conversion keyword> <left delimiter>
                           <deriving conversion name> <wkt separator>
                           <operation method> [ { <wkt separator>
                           { <operation parameter> | <operation parameter file> } } ]...
                           [ { <wkt separator> <identifier> } ]... <right delimiter>

<deriving conversion keyword> ::= DERIVINGCONVERSION
<deriving conversion name> ::= <quoted Latin text>           !! See 7.2.

```

Operation method, operation parameter and operation parameter file are described in [17.2.4](#) to [17.2.6](#).

Identifier is described in [7.3.4](#).

14.2.2 Derived CRS conversion method

Requirement: The WKT representation of a derived CRS conversion method shall be:

```

<operation method> ::= <operation method keyword> <left delimiter>
                       <operation method name>
                       [ { <wkt separator> <identifier> } ]... <right delimiter>

<operation method keyword> ::= METHOD
<operation method name> ::= <quoted Latin text>           !! See 7.2.

```

Method name is for human readability. For interoperability it is the method formula that is critical in determining the equivalence of methods; this may be given through an operation method <identifier>.

<identifier> is described in 7.3.4. It is an optional attribute. If an identifier is included as an attribute within the derived CRS conversion object (14.2.1), it shall take precedence over any identifier within the operation method object.

If an <identifier> is omitted for both derived CRS conversion and derived CRS conversion method, the WKT description is potentially ambiguous, relying on interpretation of method name. It is recommended that either a derived CRS conversion identifier or a derived CRS conversion method identifier is included in WKT strings. Identifiers for commonly encountered coordinate transformation methods are given in E.4.

14.2.3 Derived CRS conversion parameter

Requirement: The WKT representation of a derived CRS conversion parameter shall be:

```

<operation parameter> ::= <parameter keyword> <left delimiter> <parameter name>
                        <wkt separator><parameter value> <wkt separator>
                        <parameter unit>
                        [ { <wkt separator> <identifier> } ]... <right delimiter>
<parameter keyword> ::= PARAMETER
<parameter name>    ::= <quoted Latin text>           !! See 7.2.
<parameter value>   ::= <signed numeric literal>
<parameter unit>    ::= <length unit> | <angle unit> | <scale unit> | <time unit>
                        | <parametric unit>
    
```

Units are described in 7.4.

Parameter name is for human readability. For interoperability it is the method formula and its parameters that are critical in determining the equivalence of methods. See Annex F. Identifiers for commonly encountered coordinate operation methods and their parameters are given in E.4; the parameters are listed in E.5. The coordinate operation parameters required are specific to the coordinate operation method and are listed sequentially. The order within the sequence is not significant but should be logical. Implementations should be prepared to read any order. For those methods included in Annex F the parameter order given in E.4 is recommended.

Requirements:

- a) In derived CRS conversion WKT strings <parameter value> shall be given in the sense base CRS to derived CRS. If the parameter unit is linear its conversion factor shall be to metres and is the number of metres per unit. If the parameter unit is angular its conversion factor shall be to radians and is the number of radians per unit. If the parameter is a scaling unit the conversion factor shall be to unity, for example parts per million (ppm) shall be given as 10⁻⁶.

NOTE For commonly-encountered parameters the parameter type is included in E.5.

- b) <identifier> is described in 7.3.4. It is an optional attribute. If an identifier is included as an attribute within the *derived CRS conversion* object (14.2.1) or *coordinate operation method* object (14.2.2), it shall take precedence over any identifier within the derived CRS conversion parameter object.

14.2.4 Derived CRS conversion parameter file

Requirement: The WKT representation of a derived CRS conversion parameter file shall be:

```

<operation parameter file> ::= <parameter file keyword> <left delimiter> <parameter name>
                              <wkt separator> <parameter file name>
                              [ { <wkt separator> <identifier> } ]... <right delimiter>
<parameter file keyword> ::= PARAMETERFILE
    
```

<parameter name> ::= <quoted Latin text> !! See 7.2.
 <parameter file name> ::= <quoted Latin text> !! See 7.2.

For <parameter name> and <identifier> the requirements given in [17.2.5](#) shall apply.

14.2.5 Derived CRS conversion example

EXAMPLE DERIVINGCONVERSION["conversion name",
 METHOD["method name", ID["authority", 123]],
 PARAMETER["parameter 1 name", 0,
 ANGLEUNIT["degree", 0.0174532925199433],
 ID["authority", 456]
],
 PARAMETER["parameter 2 name", -123,
 ANGLEUNIT["degree", 0.0174532925199433],
 ID["authority", 789]
]
]

14.3 Derived geodetic CRS and derived geographic CRS

14.3.1 Representation

Requirement: The WKT representation of a derived geodetic or a derived geographic coordinate reference system shall be:

<derived geodetic crs> ::= <derived static geod crs> | <derived dynamic geod crs>
 | <derived geographic crs>
 <derived geographic crs> ::= <derived static geog crs> | <derived dynamic geog crs>
 <derived static geod crs> ::= <geodetic crs keyword> <left delimiter>
 <derived crs name> <wkt separator>
 { <base static geod crs> | <base static geog crs> }
 <wkt separator> <deriving conversion>
 <wkt separator> <coordinate system>
 <scope extent identifier remark> <right delimiter>
 <derived dynamic geod crs> ::= <geodetic crs keyword>
 <left delimiter> <derived crs name> <wkt separator>
 { <base dynamic geod crs> | <base dynamic geog crs> }
 <wkt separator> <deriving conversion>
 <wkt separator> <coordinate system>
 <scope extent identifier remark> <right delimiter>
 <derived static geog crs> ::= <geographic crs keyword >
 <left delimiter> <derived crs name> <wkt separator>
 { <base static geod crs> | <base static geog crs> }
 <wkt separator> <deriving conversion>
 <wkt separator> <coordinate system>
 <scope extent identifier remark> <right delimiter>

<code><derived dynamic geog crs></code>	::=	<code><geographic crs keyword ></code> <code><left delimiter> <derived crs name> <wkt separator></code> <code>{ <base dynamic geod crs> <base dynamic geog crs> }</code> <code><wkt separator> <deriving conversion></code> <code><wkt separator> <coordinate system></code> <code><scope extent identifier remark> <right delimiter></code>
<code><geodetic crs keyword></code>	::=	<code>GEODCRS GEODETICCRS</code> <i>!! See 8.1.</i>
<code><geographic crs keyword></code>	::=	<code>GEOGCRS GEOGRAPHICCRS</code> <i>!! See 8.1.</i>
<code><derived crs name></code>	::=	<code><quoted Latin text></code> <i>!! See 7.2.</i>
<code><base static geod crs></code>	::=	<code><base geodetic crs keyword> <left delimiter></code> <code><base crs name> <wkt separator></code> <code>{ <geodetic reference frame> <geodetic datum ensemble> }</code> <code>[{ <wkt separator> <identifier> }]... <right delimiter></code>
<code><base dynamic geod crs></code>	::=	<code><base geodetic crs keyword> <left delimiter></code> <code><base crs name> <wkt separator> <dynamic crs></code> <code><wkt separator> <geodetic reference frame></code> <code>[{ <wkt separator> <identifier> }]... <right delimiter></code>
<code><base static geog crs></code>	::=	<code><base geographic crs keyword> <left delimiter></code> <code><base crs name> <wkt separator></code> <code>{ <geodetic reference frame> <geodetic datum ensemble> }</code> <code>[{ <wkt separator> <identifier> }]... <right delimiter></code>
<code><base dynamic geog crs></code>	::=	<code><base geographic crs keyword> <left delimiter></code> <code><base crs name> <wkt separator> <dynamic crs></code> <code><wkt separator> <geodetic reference frame></code> <code>[{ <wkt separator> <identifier> }]... <right delimiter></code>
<code><base geodetic crs keyword></code>	::=	<code>BASEGEODCRS</code>
<code><base geographic crs keyword></code>	::=	<code>BASEGEOGCRS</code>
		<i>!! The base CRS keyword should reflect the type of CRS (geodetic or geographic) upon which the derived CRS is referenced. The previous version of this document used only BASEGEODCRS. Implementations should be prepared to read both forms.</i>
<code><base crs name></code>	::=	<code><quoted Latin text></code> <i>!! See 7.2.</i>

`<coordinate system>` is described in 7.5; the constraints for geodetic CRSs and geographic CRSs apply to derived geodetic CRSs and derived geographic CRSs.

`<dynamic crs>` is described in 7.7. It is mandatory when the base CRS is dynamic and should not be given when the base CRS is static. The derived CRS has the same static/dynamic status as its base CRS.

`<geodetic reference frame>` is described in 8.2; it includes ellipsoid and prime meridian descriptions. `<geodetic datum ensemble>` is described in 7.6.

`<deriving conversion>` is described in 14.2, `<scope extent identifier remark>` in 7.3.

NOTE Constructs similar to `<base static geod crs>`, `<base dynamic geod crs>`, `<base static geog crs>` and `<base dynamic geog crs>`, but with an extra element, are also used for projected CRSs, see 9.2.

14.3.2 Example of WKT describing a derived geographic CRS

EXAMPLE Derived geographic CRS with rotated pole, base CRS is dynamic:

```
GEOGCRS["WMO Atlantic Pole",
  BASEGEOGCRS["WGS 84 (G1762)",
    DYNAMIC[FRAMEEPOCH[2005.0]],
    TRF["World Geodetic System 1984 (G1762)",
      ELLIPSOID["WGS 84",6378137,298.257223563,LENGTHUNIT["metre",1.0]]]
  ],
  DERIVINGCONVERSION["Atlantic pole",
    METHOD["Pole rotation",ID["Authority",1234]],
    PARAMETER["Latitude of rotated pole",52.0,
      ANGLEUNIT["degree",0.0174532925199433]],
    PARAMETER["Longitude of rotated pole",-30.0,
      ANGLEUNIT["degree",0.0174532925199433]],
    PARAMETER["Axis rotation",-25.0,
      ANGLEUNIT["degree",0.0174532925199433]]
  ],
  CS[ellipsoidal,2],
  AXIS["latitude",north,ORDER[1]],
  AXIS["longitude",east,ORDER[2]],
  ANGLEUNIT["degree",0.0174532925199433]
]
```

14.4 Derived projected CRS

14.4.1 Representation

The term 'derived projected CRS' is used for consistency in the ISO 19111 UML modelling. A derived projected CRS is not a projected CRS; 'derived from projected CRS' would be a more accurate description. However, in addition to inheriting its datum or reference frame from its base projected CRS, a derived projected CRS also inherits the projection distortions of its base projected CRS.

Requirement: The WKT representation of a derived projected coordinate reference system shall be:

```
<derived projected crs> ::= <derived projected crs keyword> <left delimiter>
  <derived crs name> <wkt separator> <base projected crs>
  <wkt separator> <deriving conversion>
  <wkt separator> <coordinate system>
  <scope extent identifier remark> <right delimiter>

<derived projected crs keyword> ::= DERIVEDPROJCRS
<derived crs name> ::= <quoted Latin text>           !! See 7.2.
<base projected crs> ::= <base projected crs keyword>
  <left delimiter> <base crs name>
  <wkt separator> <base geodetic geographic crs>
  <wkt separator> <map projection>
  [ { <wkt separator> <identifier> } ]... <right delimiter>

<base projected crs keyword> ::= BASEPROJCRS
<base crs name> ::= <quoted Latin text>           !! See 7.2.
<base geodetic geographic crs> ::= <base static geod crs> | <base dynamic geod crs>
  | <base static geog crs> | <base dynamic geog crs>

!! These four elements are described in 14.3.1.
The derived projected CRS takes its dynamic
characteristics from its base geodetic or
geographic CRS.
```

<coordinate system> is described in 7.5; the constraint on projected CRS having CS type of Cartesian does not apply to derived projected CRSs.

<deriving conversion> is described in 14.2, <map projection> in 9.3, and <scope extent identifier remark> in 7.3.

14.4.2 Example of WKT describing a derived projected CRS

```

DERIVEDPROJCRS["Gulf of Mexico speculative seismic survey bin grid",
  BASEPROJCRS["NAD27 / Texas South Central",
    BASEGEOGCRS["NAD27",
      DATUM["North American Datum 1927",
        ELLIPSOID["Clarke 1866",20925832.164,294.97869821,
          LENGTHUNIT["US survey foot",0.304800609601219]
        ]
      ]
    ],
    CONVERSION["Texas South Central SPCS27",
      METHOD["Lambert Conic Conformal (2SP)",ID["EPSG",9802]],
      PARAMETER["Latitude of false origin",27.83333333333333,
        ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8821]],
      PARAMETER["Longitude of false origin",-99.0,
        ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8822]],
      PARAMETER["Latitude of 1st standard parallel",28.38333333333333,
        ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8823]],
      PARAMETER["Latitude of 2nd standard parallel",30.28333333333333,
        ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8824]],
      PARAMETER["Easting at false origin",2000000.0,
        LENGTHUNIT["US survey foot",0.304800609601219],ID["EPSG",8826]],
      PARAMETER["Northing at false origin",0.0,
        LENGTHUNIT["US survey foot",0.304800609601219],ID["EPSG",8827]]
    ]
  ],
  DERIVINGCONVERSION["Gulf of Mexico speculative survey bin grid",
    METHOD["P6 (I = J-90°) seismic bin grid transformation",ID["EPSG",1049]],
    PARAMETER["Bin grid origin I",5000,SCALEUNIT["Bin",1.0],ID["EPSG",8733]],
    PARAMETER["Bin grid origin J",0,SCALEUNIT["Bin",1.0],ID["EPSG",8734]],
    PARAMETER["Bin grid origin Easting",871200,
      LENGTHUNIT["US survey foot",0.304800609601219],ID["EPSG",8735]],
    PARAMETER["Bin grid origin Northing",10280160,
      LENGTHUNIT["US survey foot",0.304800609601219],ID["EPSG",8736]],
    PARAMETER["Scale factor of bin grid",1.0,
      SCALEUNIT["Unity",1.0],ID["EPSG",8737]],
    PARAMETER["Bin width on I-axis",82.5,
      LENGTHUNIT["US survey foot",0.304800609601219],ID["EPSG",8738]],
    PARAMETER["Bin width on J-axis",41.25,
      LENGTHUNIT["US survey foot",0.304800609601219],ID["EPSG",8739]],
    PARAMETER["Map grid bearing of bin grid J-axis",340,
      ANGLEUNIT["degree",0.0174532925199433],ID["EPSG",8740]],
    PARAMETER["Bin node increment on I-axis",1.0,
      SCALEUNIT["Bin",1.0],ID["EPSG",8741]],
    PARAMETER["Bin node increment on J-axis",1.0,
      SCALEUNIT["Bin",1.0],ID["EPSG",8742]]
  ],
  CS[ordinal,2],
  AXIS["Inline (I)",northNorthWest],
  AXIS["Crossline (J)",westSouthWest]
]

```

NOTE A similar approach can be used to describe any rectangular grid referencing system based on a projected CRS. The grid cell size and placement are defined through the deriving conversion.

14.5 Derived vertical CRS

Requirement: The WKT representation of a derived vertical coordinate reference system shall be:

<derived vertical crs>	::=	<vertical crs keyword> <left delimiter> <derived crs name> <wkt separator> <base vertical crs> <wkt separator> <deriving conversion> <wkt separator> <coordinate system> <scope extent identifier remark> <right delimiter>
<vertical crs keyword>	::=	VERTCRS VERTICALCRS <i>!! See 10.1.</i>
<derived crs name>	::=	<quoted Latin text> <i>!! See 7.2.</i>
<base vertical crs>	::=	<base static vertical crs> <base dynamic vertical crs> <i>!! The derived CRS takes its dynamic characteristics from its base CRS.</i>
<base static vertical crs>	::=	<base vertical crs keyword> <left delimiter> <base crs name> <wkt separator> { <vertical reference frame> <vertical datum ensemble> } [{ <wkt separator> <identifier> }]... <right delimiter>
<base dynamic vertical crs>	::=	<base vertical crs keyword> <left delimiter> <base crs name> <wkt separator> <dynamic crs> <wkt separator> <vertical reference frame> [{ <wkt separator> <identifier> }]... <right delimiter>
<base vertical crs keyword>	::=	BASEVERTCRS
<base crs name>	::=	<quoted Latin text> <i>!! See 7.2.</i>

<coordinate system> is described in 7.5; the constraints for vertical CRSs apply to vertical derived CRSs.

<dynamic crs> is described in 7.7. It is mandatory when the base CRS is dynamic and should not be given when the base CRS is static. The derived CRS has the same static/dynamic status as its base CRS.

<vertical reference frame> is described in 10.2, <vertical datum ensemble> in 7.6, <deriving conversion> in 14.2 and <scope extent identifier remark> in 7.3.

14.6 Derived engineering CRS

Requirement: The WKT representation of a derived engineering coordinate reference system shall be:

<derived engineering crs>	::=	<engineering crs keyword> <left delimiter> <derived crs name> <wkt separator> <base engineering crs> <wkt separator> <deriving conversion> <wkt separator> <coordinate system> <scope extent identifier remark> <right delimiter>
<engineering crs keyword>	::=	ENGCRS ENGINEERINGCRS <i>!! See 11.1.</i>
<derived crs name>	::=	<quoted Latin text> <i>!! See 7.2.</i>
<base engineering crs>	::=	<base engineering crs keyword> <left delimiter> <base crs name> <wkt separator> <engineering datum> [{ <wkt separator> <identifier> }]... <right delimiter>
<base engineering crs keyword>	::=	BASEENGCRS
<base crs name>	::=	<quoted Latin text> <i>!! See 7.2.</i>

<coordinate system> is described in 7.5; the constraints for engineering CRSs apply to derived engineering CRSs.

<engineering datum> is described in 11.2, <deriving conversion> is described in 14.2 and <scope extent identifier remark> in 7.3.

14.7 Derived parametric CRS

Requirement: The WKT representation of a derived parametric coordinate reference system shall be:

```

<derived parametric crs> ::= <parametric crs keyword> <left delimiter> <derived crs name>
                             <wkt separator> <base parametric crs>
                             <wkt separator> <deriving conversion>
                             <wkt separator> <coordinate system>
                             <scope extent identifier remark> <right delimiter>

<parametric crs keyword> ::= PARAMETRICCRS           !! See 12.1.1.
<derived crs name>       ::= <quoted Latin text>      !! See 7.2.
<base parametric crs>    ::= <base parametric crs keyword> <left delimiter>
                             <base crs name> <wkt separator>
                             <parametric datum>
                             [ { <wkt separator> <identifier> } ]... <right delimiter>

<base parametric crs keyword> ::= BASEPARAMCRS
<base crs name>               ::= <quoted Latin text>           !! See 7.2.
    
```

<coordinate system> is described in 7.5; the constraints for parametric CRSs apply to derived parametric CRSs.

<parametric datum> is described in 12.2, <deriving conversion> in 14.2 and <scope extent identifier remark> in 7.3.

14.8 Derived temporal CRS

Requirement: The WKT representation of a derived temporal coordinate reference system shall be:

```

<derived temporal crs> ::= <temporal crs keyword> <left delimiter> <derived crs name>
                             <wkt separator> <base temporal crs>
                             <wkt separator> <deriving conversion>
                             <wkt separator> <coordinate system>
                             <scope extent identifier remark> <right delimiter>

<temporal crs keyword> ::= TIMECRS           !! See 13.1.
<derived crs name>     ::= <quoted Latin text>      !! See 7.2.
<base temporal crs>    ::= <base temporal crs keyword> <left delimiter> <base crs name>
                             <wkt separator> <temporal datum>
                             [ { <wkt separator> <identifier> } ]... <right delimiter>

<base temporal crs keyword> ::= BASETIMECRS
<base crs name>             ::= <quoted Latin text>           !! See 7.2.
    
```

<coordinate system> for temporal CRSs is described in 7.5 excluding 7.5.6 and 13.3; the constraints for temporal CRSs apply to temporal derived CRSs.

<temporal datum> is described in 13.2, <deriving conversion> in 14.2 and <scope extent identifier remark> in 7.3.

15 WKT representation of compound coordinate reference systems

15.1 Overview

A compound CRS is a non-repeating sequence of two or more independent coordinate reference systems none of which can be compound.

Requirement: The WKT representation of a compound coordinate reference system shall be:

```

<compound crs> ::= <compound crs keyword> <left delimiter> <compound crs name>
                  <wkt separator> <single crs>
                  <wkt separator> <single crs>
                  [ { <wkt separator> <single crs> } ]...
                  <scope extent identifier remark> <right delimiter>
                  !! ISO 19111 defines valid combinations of single CRS that
                  may form a compound CRS.

<compound crs keyword> ::= COMPOUNDCRS
<compound crs name>   ::= <quoted Latin text>           !! See 7.2.
<single crs>          ::= <geodetic crs> | <derived geodetic crs>
                        | <projected crs> | <derived projected crs>
                        | <vertical crs> | <derived vertical crs>
                        | <engineering crs> | <derived engineering crs>
                        | <parametric crs> | <derived parametric crs>
                        | <temporal crs> | <derived temporal crs>

```

The representation of constituent single CRSs is elaborated in [Clauses 8](#) to [14](#).

Constraints on axis order for compound CRSs are described in [7.5.5](#).

<scope extent identifier remark> is described in [7.3](#).

15.2 Examples of WKT describing a compound CRS

EXAMPLE 1 Spatial compound CRS:

```

COMPOUNDCRS["NAD83 + NAVD88",
  GEOGCRS["NAD83",
    DATUM["North American Datum 1983",
      ELLIPSOID["GRS 1980",6378137,298.257222101,
        LENGTHUNIT["metre",1.0]]
    ],
    PRIMEMERIDIAN["Greenwich",0],
    CS[ellipsoidal,2],
    AXIS["latitude",north,ORDER[1]],
    AXIS["longitude",east,ORDER[2]],
    ANGLEUNIT["degree",0.0174532925199433]
  ],
  VERTCRS["NAVD88",
    VDATUM["North American Vertical Datum 1983"],
    CS[vertical,1],
    AXIS["gravity-related height (H)",up],
    LENGTHUNIT["metre",1]
  ]
]

```

EXAMPLE 2 Spatio-parametric compound CRS:

```
COMPOUNDCRS["ICAO layer 0",
  GEOGRAPHICCRS["WGS 84",
    DYNAMIC[FRAMEEPOCH[2005]],
    DATUM["World Geodetic System 1984",
      ELLIPSOID["WGS 84",6378137,298.257223563,
        LENGTHUNIT["metre",1.0]],
      CS[ellipsoidal,2],
        AXIS["latitude",north,ORDER[1]],
        AXIS["longitude",east,ORDER[2]],
        ANGLEUNIT["degree",0.0174532925199433]
      ],
    PARAMETRICCRS["WMO standard atmosphere",
      PARAMETRICDATUM["Mean Sea Level",
        ANCHOR["Mean Sea Level = 1013.25 hPa"]],
      CS[parametric,1],
        AXIS["pressure (P)",unspecified],
        PARAMETRICUNIT["HectoPascal",100]
      ]
    ]
  ]
```

EXAMPLE 3 Spatio-temporal compound CRS (ellipsoid axis unit is metres as <lengthunit> is omitted):

```
COMPOUNDCRS["2D GPS position with civil time in ISO 8601 format",
  GEOGCRS["WGS 84 (G1762)",
    DYNAMIC[FRAMEEPOCH[2005]],
    DATUM["World Geodetic System 1984 (G1762)",
      ELLIPSOID["WGS 84",6378137,298.257223563]],
      CS[ellipsoidal,2],
        AXIS["(lat)",north,ORDER[1]],
        AXIS["(lon)",east,ORDER[2]],
        ANGLEUNIT["degree",0.0174532925199433]
      ],
    TIMECRS["DateTime"
      TDATUM["Gregorian Calendar"],
      CS[TemporalDateTime,1],AXIS["Time (T)",future]
    ]
  ]
```

16 WKT representation of coordinate epoch and coordinate metadata

16.1 Coordinate epoch

Coordinate epoch is a mandatory attribute for a coordinate set that is referenced to a dynamic CRS. Coordinate epoch is not part of a CRS definition, it is additional metadata for the coordinates which is required to ensure that they are unambiguous when referenced to a dynamic CRS.

Requirement: The WKT representation of a coordinate epoch shall be:

```
<metadata coordinate epoch> ::= <coordinate epoch keyword> <left delimiter>
                                <coordinate epoch> <right delimiter>
<coordinate epoch keyword> ::= EPOCH | COORDEPOCH
```

!! In this document for brevity the preferred keyword is EPOCH, but COORDEPOCH is permitted. Implementations should be prepared to read both forms.

!! EPOCH shall not be used as the keyword for frame reference epoch, for which FRAMEEPOCH should be used, or as the name of a coordinate operation parameter, for which "Parameter Epoch" or "Transformation Epoch" as appropriate should be used.

<coordinate epoch> ::= <unsigned integer> [<period> [<unsigned integer>]]
 !! See 6.3.2.

EXAMPLE EPOCH[2016.47]

16.2 Coordinate metadata

Coordinate metadata is the information required to make coordinates unambiguous. For a coordinate set referenced to a static CRS it is the CRS definition. For a coordinate set referenced to a dynamic CRS it is the CRS definition together with the coordinate epoch of the coordinates in the coordinate set.

Requirement: The WKT representation of coordinate metadata shall be:

<coordinate metadata> ::= <coordinate metadata keyword> <left delimiter>
 { <static crs coordinate metadata>
 | { <dynamic crs coordinate metadata> <wkt separator>
 <metadata coordinate epoch> } }
 <right delimiter>

<coordinate metadata keyword> ::= COORDINATEMETADATA

<static crs coordinate metadata> ::= <static geodetic crs> | <static geographic crs>
 | <projected crs> | <static vertical crs>
 | <engineering crs> | <parametric crs>
 | <temporal crs> | <derived geodetic crs>
 | <derived projected crs> | <derived vertical crs>
 | <derived engineering crs> | <derived parametric crs>
 | <derived temporal crs> | <compound crs>

!! A projected CRS is static if its base CRS is static.

<dynamic crs coordinate metadata> ::= <dynamic geodetic crs> | <dynamic geographic crs>
 | <projected crs> | <dynamic vertical crs>
 | <derived geodetic crs>
 | <derived projected crs> | <derived vertical crs>

!! The inclusion of the keyword DYNAMIC within the CRS description indicates that the CRS is a dynamic CRS. See 7.7.

A derived CRS (including a projected CRS) is dynamic if its base CRS is dynamic.

The WKT for these types of coordinate reference system (CRS) is described in [Clauses 8](#) to [15](#).

EXAMPLE Coordinate metadata for a dataset referenced to a dynamic CRS:

```
COORDINATEMETADATA[
  GEOGCRS["WGS 84 (G1762)",
    DYNAMIC[FRAMEEPOCH[2005.0]],
    DATUM["World Geodetic System 1984 (G1762)",
      ELLIPSOID["WGS 84",6378137,298.257223563,LENGTHUNIT["metre",1.0]]
    ],
  CS[ellipsoidal,3],
  AXIS["(lat)",north,ANGLEUNIT["degree",0.0174532925199433]],
  AXIS["(lon)",east,ANGLEUNIT["degree",0.0174532925199433]],
  AXIS["ellipsoidal height (h)",up,LENGTHUNIT["metre",1.0]]
],
EPOCH[2016.47]
]
```

17 WKT representation of coordinate transformations and coordinate conversions excluding map projections

17.1 Coordinate operations

The ISO 19111 data model describes five subtypes of coordinate operation: transformation, conversion, point motion operation, concatenated operation and pass-through operation. ISO 19111 also defines the terms *coordinate transformation* and *coordinate conversion*; these differ through whether the source and target CRSs are referenced to the same or different reference frames (datums). For reasons of backward compatibility, in this document the keyword <coordinate operation> is used specifically for coordinate transformations and coordinate conversions other than deriving conversions including map projections. Deriving conversions are described in [Clause 14](#); map projections are a special case of deriving conversion and part of a projected CRS definition and are described in [9.3](#). The WKT for point motion operations is described in [Clause 18](#). The WKT for concatenated operations is described in [Clause 19](#). This document does not define WKT for pass-through operations.

Requirement: The WKT representation of a coordinate transformation and of a coordinate conversion other than a deriving conversion (including map projection) shall be:

```
<coordinate operation> ::= <operation keyword> <left delimiter> <operation name>
                           [ <wkt separator> <operation version> ]
                           <wkt separator> <source crs> <wkt separator> <target crs>
                           <wkt separator> <operation method>
                           [ { <wkt separator>
                             { <operation parameter> | <operation parameter file> } } ]...
                           [ <wkt separator> <interpolation crs> ]
                           [ <wkt separator> <operation accuracy> ]
                           <scope extent identifier remark> <right delimiter>

<operation keyword> ::= COORDINATEOPERATION
```

17.2 Transformation and conversion components

17.2.1 Operation name and version

```
<operation name> ::= <quoted Latin text> !! See 7.2.
<operation version> ::= <operation version keyword> <left delimiter>
                       <operation version text> <right delimiter>
<operation version keyword> ::= VERSION
<operation version text> ::= <quoted Latin text> !! See 7.2.
```

17.2.2 Source and target CRS

Requirement: The WKT representation of the source and target CRSs of a <coordinate operation> shall be:

```
<source crs> ::= <source crs keyword> <left delimiter>
               <coordinate reference system> <right delimiter>
<source crs keyword> ::= SOURCECRS
<target crs> ::= <target crs keyword> <left delimiter>
               <coordinate reference system> <right delimiter>
<target crs keyword> ::= TARGETCRS
<coordinate reference system> ::= <single crs> | <compound crs>
```

Coordinate reference systems are defined in [Clauses 8](#) to [15](#). <single crs> and <compound crs> are elaborated in [15.1](#).

17.2.3 Transformation and conversion name and identifier

Coordinate operation encompasses the collection of method and parameter values. Its name is for human readability.

<identifier> is described in [7.3.4](#). It is an optional attribute. If an identifier is provided as an attribute within the <coordinate operation> object ([17.1](#)), because it is expected to describe a complete collection of zone name, method, parameters and parameter values, it shall override any identifiers given within the coordinate operation method and coordinate operation parameter objects.

Requirement: If an identifier is provided as an attribute of a coordinate operation object, it shall override any identifiers given within the component method and parameter objects.

17.2.4 Coordinate operation method

Requirement: The WKT representation of a coordinate operation method shall be:

```
<operation method> ::= <operation method keyword> <left delimiter>
                    <operation method name>
                    [ { <wkt separator> <identifier> } ]... <right delimiter>

<operation method keyword> ::= METHOD

<operation method name> ::= <quoted Latin text>           !! See 7.2.
```

Method name is for human readability. For interoperability it is the method formula that is critical in determining the equivalence of methods; this may be given through an operation method <identifier>.

<identifier> is described in [7.3.4](#). If an <identifier> is omitted for both coordinate operation ([17.1](#)) and operation method, the WKT description is potentially ambiguous, relying on interpretation of method name. It is recommended that either a coordinate operation identifier or a coordinate operation method identifier is included in WKT strings. Identifiers for commonly encountered coordinate transformation methods are given in [F.4](#).

17.2.5 Coordinate operation parameter

Requirement: The WKT representation of a coordinate operation parameter shall be:

```
<operation parameter> ::= <parameter keyword> <left delimiter> <parameter name>
                        <wkt separator><parameter value> <wkt separator> <parameter unit>
                        [ { <wkt separator> <identifier> } ]... <right delimiter>

<parameter keyword> ::= PARAMETER

<parameter name> ::= <quoted Latin text>           !! See 7.2.

<parameter value> ::= <signed numeric literal>

<parameter unit> ::= <length unit> | <angle unit> | <scale unit> | <parametric unit>
                   | <time unit>
```

Units are described in [7.4](#).

Parameter name is for human readability. For interoperability it is the method formula and its parameters that are critical in determining the equivalence of methods. See [Annex F](#). Identifiers for commonly encountered coordinate operation methods and their parameters are given in [F.4](#); the parameters are listed in [F.5](#). The coordinate operation parameters required are specific to the coordinate operation method and shall be listed sequentially. The order within the sequence is not significant but should be logical. Implementations should be prepared to read any order. For those methods included in [Annex F](#) the parameter order given in [F.4](#) is recommended.

Requirements:

- a) In coordinate operation WKT strings <parameter value> shall be given in the sense <source crs> to <target crs>. If the transformation parameter unit is linear its conversion factor shall be to metres and is the number of metres per unit. If the unit is angular its conversion factor shall be to radians and is the number of radians per unit. If the parameter is a scaling unit the conversion factor shall be to unity, for example parts per million (ppm) shall be given as 10⁻⁶.

NOTE For commonly-encountered transformation parameters the parameter type is included in [E.5](#).

- b) <identifier> is described in [7.3.4](#). It is an optional attribute. If an identifier is included as an attribute within the *coordinate operation* object ([17.1](#)) or *coordinate operation method* object ([17.2.4](#)), it shall take precedence over any identifier within the coordinate operation parameter object.

17.2.6 Coordinate operation parameter file

Requirement: The WKT representation of a coordinate operation parameter file shall be:

```

<operation parameter file> ::= <parameter file keyword> <left delimiter> <parameter name> <wkt separator>
                             <parameter file name>
                             [ { <wkt separator> <identifier> } ]... <right delimiter>
<parameter file keyword> ::= PARAMETERFILE
<parameter name>         ::= <quoted Latin text>           !! See 7.2.
<parameter file name>    ::= <quoted Latin text>           !! See 7.2.
    
```

For <parameter name> and <identifier> the requirements given in [17.2.5](#) shall apply.

17.2.7 Interpolation CRS

Some coordinate operation methods require coordinates referenced to a CRS which is neither the source CRS nor the target CRS. For example, in a coordinate operation applying a vertical offset between two vertical CRSs using either the vertical offset and slope method or a grid interpolation method such as VERTCON, horizontal coordinates are required. <interpolation crs> provides the mechanism for defining the CRS to which these coordinates are referenced.

Requirement: The WKT representation of an <interpolation crs> shall be:

```

<interpolation crs> ::= <interpolation crs keyword> <left delimiter>
                       <coordinate reference system> <right delimiter>
<interpolation crs keyword> ::= INTERPOLATIONCRS
    
```

<coordinate reference system> is defined in [17.2.2](#).

17.2.8 Coordinate operation accuracy

Operation accuracy is an optional attribute which indicates the typical error the application of a coordinate operation will introduce into transformed target CRS coordinates assuming input of errorless source CRS coordinates. It is an approximate figure for the area of applicability of the coordinate operation as a whole, given in metres.

Requirement: The WKT representation of an <operation accuracy> shall be:

```

<operation accuracy> ::= <operation accuracy keyword> <left delimiter>
                       <accuracy> <right delimiter>
<operation accuracy keyword> ::= OPERATIONACCURACY
<accuracy> ::= <number>           !! <accuracy> is in metres
    
```

17.2.9 Other coordinate operation attributes

<identifier> is described in 7.3.4. It is an optional attribute.

<scope> is described in 7.3.2 and <extent> in 7.3.3. These are optional attributes describing the applicability of the coordinate operation.

<remark> is described in 7.3.4.

17.3 Examples of WKT describing a coordinate transformation

Line feeds are included in these examples to aid clarity. In several of these examples the full source and target CRS definitions are omitted from the example so that the coordinate transformation elements are more clearly identified.

EXAMPLE 1

```
COORDINATEOPERATION["Tokyo to JGD2000",VERSION["GSI"],
SOURCECRS [
  GEODCRS ["Tokyo",
    DATUM ["Tokyo 1918",
      ELLIPSOID ["Bessel 1841",6377397.155,299.1528128,
        LENGTHUNIT ["metre",1.0]]],
    CS [Cartesian,3],
    AXIS ["(X)",geocentricX,ORDER[1]],
    AXIS ["(Y)",geocentricY,ORDER[2]],
    AXIS ["(Z)",geocentricZ,ORDER[3]],
    LENGTHUNIT ["metre",1.0]
  ]
],
TARGETCRS [
  GEODCRS ["JGD2000",
    DATUM ["Japanese Geodetic Datum 2000",
      ELLIPSOID ["GRS 1980",6378137.0,298.257222101,
        LENGTHUNIT ["metre",1.0]]],
    CS [Cartesian,3],
    AXIS ["(X)",geocentricX],
    AXIS ["(Y)",geocentricY],
    AXIS ["(Z)",geocentricZ],
    LENGTHUNIT ["metre",1.0]
  ]
],
METHOD ["Geocentric translations",ID["EPSG",1031]],
PARAMETER ["X-axis translation",-146.414,
  LENGTHUNIT ["metre",1.0],ID["EPSG",8605]],
PARAMETER ["Y-axis translation",507.337,
  LENGTHUNIT ["metre",1.0],ID["EPSG",8606]],
PARAMETER ["Z-axis translation",680.507,
  LENGTHUNIT ["metre",1.0],ID["EPSG",8607]]
]
```

EXAMPLE 2

```
COORDINATEOPERATION["AGD84 to GDA94",
SOURCECRS [...full CRS definition required here but omitted for brevity...],
TARGETCRS [...full CRS definition required here but omitted for brevity...],
METHOD ["Geocentric translations",ID["EPSG",1031]],
PARAMETER ["X-axis translation",-128.5,LENGTHUNIT ["metre",1.0]],
PARAMETER ["Y-axis translation",-53.0,LENGTHUNIT ["metre",1.0]],
PARAMETER ["Z-axis translation",153.4,LENGTHUNIT ["metre",1.0]],
OPERATIONACCURACY [5],
USAGE [SCOPE ["Low accuracy applications."],
  AREA ["Australia onshore"],BBOX [-43.7,112.85,-9.87,153.68]],
REMARK ["Use NTV2 file for better accuracy"]
]
```

EXAMPLE 3

```
COORDINATEOPERATION["NZGD49 to NZGD2000",
  SOURCECRS[...full WKT definition required here but omitted for brevity...],
  TARGETCRS[...full WKT definition required here but omitted for brevity...],
  METHOD["NTv2",ID["EPSG",9615]],
  PARAMETERFILE["Latitude and longitude difference file",
    "nzgd2kgrid0005.gsb"],
  ID["EPSG",1568,CITATION["LINZS25000"]],
  URI["http://www.linz.govt.nz/geodetic/software-downloads/"],
  REMARK["Coordinate transformation accuracy 0.1-1.0m"]
]
```

EXAMPLE 4

```
COORDINATEOPERATION["Amersfoort to ETRS89 (3)",
  SOURCECRS[...full CRS definition required here but omitted for brevity...],
  TARGETCRS[...full CRS definition required here but omitted for brevity...],
  METHOD["Coordinate Frame"],
  PARAMETER["X-axis translation",565.2369,LENGTHUNIT["metre",1.0]],
  PARAMETER["Y-axis translation",50.0087,LENGTHUNIT["metre",1.0]],
  PARAMETER["Z-axis translation",465.658,LENGTHUNIT["metre",1.0]],
  PARAMETER["X-axis rotation",1.9725,ANGLEUNIT["microradian",1E-06]],
  PARAMETER["Y-axis rotation",-1.7004,ANGLEUNIT["microradian",1E-06]],
  PARAMETER["Z-axis rotation",9.0677,ANGLEUNIT["microradian",1E-06]],
  PARAMETER["Scale difference",4.0812,SCALEUNIT["parts per million",1E-06]],
  ID["EPSG",15739]
]
```

EXAMPLE 5

```
COORDINATEOPERATION["DHHN92 height to EVRF2007 height",
  SOURCECRS[...full WKT definition of DHHN92 required here but omitted for brevity...],
  TARGETCRS[...full WKT definition of EVRF2007 required here but omitted for brevity...],
  METHOD["Vertical Offset and Slope",ID["EPSG",1046]],
  PARAMETER["Inclination in latitude",-0.010,
    ANGLEUNIT["arc-second",4.84813681109535E-06]],
  PARAMETER["Inclination in longitude",0.002,
    ANGLEUNIT["arc-second",4.84813681109535E-06]],
  PARAMETER["Vertical offset",0.015,LENGTHUNIT["metre",1.0]],
  PARAMETER["Ordinate 1 of evaluation point",51.05,
    ANGLEUNIT["degree",0.0174532925199433]],
  PARAMETER["Ordinate 2 of evaluation point",10.2166666666667,
    ANGLEUNIT["degree",0.0174532925199433]],
  INTERPOLATIONCRS["ETRS89"...full WKT definition of ETRS89 required here but omitted for brevity...],
  OPERATIONACCURACY[0.1],
  REMARK["Determined at 427 points. RMS residual 0.002m, maximum 0.007m"]
]
```

18 WKT representation of point motion operations

A point motion operation describes the change in coordinate values due to motion of the point between two epochs.

Requirement: The WKT representation of a <point motion operation> shall be:

```

<point motion operation> ::= <point motion keyword> <left delimiter> <operation name>
                             [ <wkt separator> <operation version> ]
                             <wkt separator> <source crs>
                             <wkt separator> <operation method>
                             [ { <wkt separator> { <operation parameter>
                               | <operation parameter file> } } ]...
                             [ <wkt separator> <operation accuracy> ]
                             <scope extent identifier remark> <right delimiter>

                             !! This BNF string is very similar to a subset of that for
                             coordinate transformations described in 17.1. Differences
                             are the omission of <target crs> and <interpolation crs>
                             and a change in keyword.

<point motion keyword> ::= POINTMOTIONOPERATION

```

The attributes <operation name>, <operation version>, <source crs>, <operation method>, <operation parameter>, <operation parameter file>, <operation accuracy>] and <scope extent identifier remark> are the same as for coordinate transformations and are described in [17.2](#).

A point motion operation changes coordinate values within a CRS; as such it does not have source or target CRSs. This document uses <source crs> in the BNF for convenience of similarity with coordinate transformations described in [Clause 17](#). Although this document allows any type of CRS, in practice a point motion operation will almost always be in a geodetic CRS, geographic CRS or vertical CRS.

EXAMPLE

```

POINTMOTIONOPERATION["Canada velocity grid v6",
  SOURCECRS[...full WKT definition of NAD83(CRS)v6 required here but omitted for brevity...],
  METHOD["Point motion by grid (Canada NTv2_Vel)"],
  PARAMETERFILE["Point motion velocity grid file","cvg60.cvb"],
  OPERATIONACCURACY[0.01]
]

```

19 WKT representation of concatenated coordinate operations

19.1 General

A concatenated coordinate operation is two or more coordinate transformations or coordinate conversions or point motion operations, or combinations of these coordinate operation types, which are applied sequentially. The target CRS for the step n is the source CRS for step $n + 1$. The source CRS for the concatenated coordinate operation is the source CRS for step 1. The target CRS for the concatenated coordinate operation is the target CRS for the last step.

Requirement: The WKT representation of a <concatenated operation> shall be:

```

<concatenated operation> ::= <concatenated operation keyword> <left delimiter>
                             <operation name> [ <wkt separator> <operation version> ]
                             <wkt separator> <source crs>
                             <wkt separator> <target crs>
                             { <wkt separator> <step keyword> <left delimiter>
                               { <coordinate operation> | <point motion operation>
                                 | <map projection> | <deriving conversion> }
                               <right delimiter> }...
                             [ <wkt separator> <operation accuracy> ]
                             <scope extent identifier remark> <right delimiter>

<concatenated operation keyword> ::= CONCATENATEDOPERATION
<step keyword> ::= STEP
<operation name> ::= <quoted Latin text> !! See 7.2.
    
```

<source crs>, <target crs>, <coordinate operation>, <operation version> and <operation accuracy> are defined in 17.1 and 17.2, <point motion operation> is defined in Clause 18.

EXAMPLE

```

CONCATENATEDOPERATION["xxxx to zzzz",
  SOURCECRS [...full WKT definition of CRS xxxx required here but omitted for brevity...],
  TARGETCRS [...full WKT definition of CRS zzzz required here but omitted for brevity...],
  STEP [...full WKT definition of operation xxxx to yyyy required here but omitted for brevity...],
  STEP [...full WKT definition of operation yyyy to zzzz required here but omitted for brevity...],
  OPERATIONACCURACY[5]
  USAGE[SCOPE["Concatenated operation scope description."],
    AREA["Concatenated operation area description."]]
]
    
```

For a concatenated coordinate operation sequence of *n* coordinate operations:

- source CRS (concatenated coordinate operation) = source CRS (coordinate operation step 1);
- target CRS (coordinate operation step *i*) = source CRS (coordinate operation step *i* + 1), where *i* = 1 ...(*n* - 1);
- target CRS (concatenated coordinate operation) = target CRS (coordinate operation step *n*).

Instead of a forward coordinate operation, an inverse coordinate operation may be used for one or more of the coordinate operation steps mentioned above, if the inverse coordinate operation is uniquely defined by the forward coordinate operation method.

EXAMPLE 1 Changing coordinates from being referenced to CRS A to being referenced to CRS B through coordinate transformation CRS A to CRS C followed by coordinate transformation CRS C to CRS B where the second transformation is documented as CRS B to CRS C but is reversible. In application of the concatenated operation, the second transformation is applied in the direction from CRS C to CRS B.

In these circumstances, it is recommended that:

- a) if the step description includes an identifier from a geodetic registry, the WKT for the step description should reflect the registry entry. In the example above:
 - the source CRS and target CRS for the concatenated operation are A and B respectively;
 - the source CRS and target CRS for step 1 are A and C and the transformation parameter values are consistent with this direction;

- the source CRS and target CRS for step 2 are B and C and the transformation parameter values are consistent with this direction.

This ensures that the parameter values which are documented in the WKT reflect those given in the geodetic registry.

- b) if the step description does not include an identifier from a geodetic registry, the WKT for the step description should reflect the direction in which the coordinate operation is applied, and the parameter values should be appropriate for this direction.

In both cases, for each step the parameter values shall be consistent with the documented direction from source to target.

19.2 Examples of WKT describing a concatenated coordinate operation

EXAMPLE 1 Concatenated operation with two steps, the last of which is documented in the reverse direction to how it is applied in the concatenated operation because a geodetic registry identifier for the step is included in the WKT. The WKT documents the step in the direction given in the registry:

```

CONCATENATEDOPERATION["RT90 to KKJ",
  SOURCECRS[GEOGCRS["RT90",
    ...full WKT definition of concatenated operation source CRS required here but omitted for brevity...]],
  TARGETCRS[GEOGCRS["KKJ",
    ...full WKT definition of concatenated operation target CRS required here but omitted for brevity...]],
  STEP[COORDINATEOPERATION["RT90 to ETRS89",
    SOURCECRS[GEOGCRS["RT90",
      ...full WKT definition of step 1 source CRS required here but omitted for brevity...]],
    TARGETCRS[GEOGCRS["ETRS89",
      ...full WKT definition of step 1 target CRS required here but omitted for brevity...]],
    METHOD[
      ...full WKT definition of operation RT90 to ETRS89 required here but omitted for brevity...
      ID["EPSG",1437]
    ],
  STEP[COORDINATEOPERATION["KKJ to ETRS89",
    SOURCECRS[GEOGCRS["KKJ",
      ...full WKT definition of step 2 source CRS required here but omitted for brevity...]],
    TARGETCRS[GEOGCRS["ETRS89",
      ...full WKT definition of step 2 target CRS required here but omitted for brevity...]],
    METHOD["Coordinate Frame rotation",ID["EPSG",9607]],
    PARAMETER["X-axis translation",-96.062,LENGTHUNIT["metre",1.0]],
    ...full WKT definition of operation KKJ to ETRS89 required here but omitted for brevity...
    ID["EPSG",10098]
  ],
  USAGE[SCOPE["Concatenated operation scope description."],
    AREA["Concatenated operation area description."]],
  REMARK["Step 2 is applied in reverse direction"]
]

```

EXAMPLE 2 The same concatenated operation as in Example 1, with no geodetic registry ID for the individual step transformations so both steps documented in the direction as applied in the concatenated operation. Note reversal of sign of parameter in second step compared to those in Example 1:

```

CONCATENATEDOPERATION["RT90 to KKJ",
  SOURCECRS [GEOGCRS ["RT90",
    ...full WKT definition of concatenated operation source CRS required here but omitted for brevity...]],
  TARGETCRS [GEOGCRS ["KKJ",
    ...full WKT definition of concatenated operation target CRS required here but omitted for brevity...]],
  STEP [COORDINATEOPERATION["RT90 to ETRS89",
    SOURCECRS [GEOGCRS ["RT90",
      ...full WKT definition of step 1 source CRS required here but omitted for brevity...]],
    TARGETCRS [GEOGCRS ["ETRS89",
      ...full WKT definition of step 1 target CRS required here but omitted for brevity...]],
    METHOD [
      ...full WKT definition of operation RT90 to ETRS89 required here but omitted for brevity...
    ],
  ],
  STEP [COORDINATEOPERATION["ETRS89 to KKJ",
    SOURCECRS [GEOGCRS ["ETRS89",
      ...full WKT definition of step 2 source CRS required here but omitted for brevity...]],
    TARGETCRS [GEOGCRS ["KKJ",
      ...full WKT definition of step 2 target CRS required here but omitted for brevity...]],
    METHOD ["Coordinate Frame rotation", ID["EPSG", 9607]],
    PARAMETER ["X-axis translation", 96.062, LENGTHUNIT ["metre", 1.0]],
    ...full WKT definition of operation ETRS89 to KKJ required here but omitted for brevity...
  ],
],
]

```

20 WKT representation of CRS and coordinate operation couplets

20.1 Bound CRS

The definition of a CRS is not dependent upon any relationship to an independent CRS. However, in an implementation that merges datasets referenced to differing CRSs, it is sometimes useful to associate the definition of the transformation that has been used with the CRS definition. This facilitates the interrelationship of CRS by concatenating transformations via a common or hub CRS. This document permits the association of an abridged coordinate transformation description with a coordinate reference system description in a single text string. In a bound CRS the abridged coordinate transformation is applied to the source CRS with the target CRS being the common or hub system.

Requirement: The WKT representation of a bound CRS shall be:

```

<bound crs> ::= <bound crs keyword> <left delimiter>
               <source crs> <wkt separator> <target crs>
               <wkt separator> <abridged coordinate transformation>
               <scope extent identifier remark>
               <right delimiter>

<bound crs keyword> ::= BOUNDCRS

```

Examples are given in [20.3](#).

The source and target CRSs shall be defined through a full CRS description as described in [Clause 8](#). If the source CRS type is projected and the abridged coordinate transformation operates in the geodetic CRS domain, the transformation is deemed to operate on the projected CRS's base CRS but the projected CRS shall be described.

20.2 Bound CRS components

20.2.1 Abridged coordinate transformation

In the WKT representation of projected coordinate reference systems the units of map projection parameters may be implied (see 9.3). In a similar way the abridged coordinate transformation also omits explicit identification of coordinate transformation parameter unit from the text string. It has several constraints.

Requirements:

- a) The abridged form of coordinate transformation shall only be used as part of a bound CRS.
- b) The abridged transformation shall be described in the sense from <source CRS> to <target CRS>.
- c) The WKT representation of an abridged coordinate transformation shall be:

```
<abridged coordinate transformation> ::= <abridged transformation keyword> <left delimiter> <operation name>
                                         [ <wkt separator> <operation version> ]
                                         <wkt separator> <operation method>
                                         [ { <wkt separator> { <abridged transformation parameter>
                                         | <operation parameter file> } } ]...
                                         <scope extent identifier remark> <right delimiter>

<abridged transformation keyword> ::= ABRIDGEDTRANSFORMATION
```

20.2.2 Coordinate operation method in abridged coordinate transformations

In an abridged coordinate transformation description the format for operation method is identical to that for coordinate operation method defined in [Clause 17](#).

20.2.3 Abridged coordinate transformation parameter

Requirement: The WKT representation of an abridged transformation parameter shall be:

```
<abridged transformation parameter> ::= <parameter keyword> <left delimiter>
                                         <parameter name> <wkt separator>
                                         <parameter value>
                                         [ { <wkt separator> <identifier> } ]... <right delimiter>
```

The format for abridged transformation parameter is similar to that for coordinate operation parameter defined in [17.2.5](#) but with the following constraints on parameter values:

Requirements:

- a) The value of parameters which are linear shall be given in metres.
- b) The value of parameters which are angular shall be given in arc-seconds (4.848 136 811 095 E-06 radian).
- c) The value of parameters which are scale units shall be given as a number with respect to unity, for example 3.5 parts per million (ppm) shall be given as 1.000 003 5 and -3.5 ppm shall be given as 0.999 996 5.
- d) <unit> shall not be given.
- e) Implementations are expected to identify the parameter value unit type from the parameter name.
- f) The parameter values shall be described in the sense from source CRS to target CRS.

The parameters required are specific to the coordinate operation method and are listed sequentially. The order within the sequence is not significant but should be logical.

```
EXAMPLE  ABRIDGEDTRANSFORMATION["Tokyo to JGD2000 (GSI)",
        METHOD["Geocentric translations",ID["EPSG",1031]],
        PARAMETER["X-axis translation",-146.414],
        PARAMETER["Y-axis translation",507.337],
        PARAMETER["Z-axis translation",680.507]
    ]
```

20.2.4 Coordinate operation parameter file

Operation parameter file is defined in [17.2.6](#).

20.3 Examples of WKT describing a bound CRS

EXAMPLE 1

```
BOUNDCRS [
    SOURCECRS [
        GEODCRS ["NAD27",
            DATUM ["North American Datum 1927",
                ELLIPSOID ["Clarke 1866", 6378206.4, 294.978698213]
            ],
            CS [ellipsoidal, 2],
            AXIS ["latitude", north], AXIS ["longitude", east],
            ANGLEUNIT ["degree", 0.0174532925199433]
        ]
    ],
    TARGETCRS [
        GEODCRS ["NAD83",
            DATUM ["North American Datum 1983",
                ELLIPSOID ["GRS 1980", 6378137, 298.2572221]
            ],
            CS [ellipsoidal, 2],
            AXIS ["latitude", north], AXIS ["longitude", east],
            ANGLEUNIT ["degree", 0.0174532925199433]
        ]
    ],
    ABRIDGEDTRANSFORMATION ["NAD27 to NAD83 Alaska",
        METHOD ["NADCON", ID ["EPSG", 9613]],
        PARAMETERFILE ["Latitude difference file", "alaska.las"],
        PARAMETERFILE ["Longitude difference file", "alaska.los"]
    ]
]
```

EXAMPLE 2

```
BOUNDCRS [
    SOURCECRS [...full WKT definition required here but omitted for brevity...],
    TARGETCRS [...full WKT definition required here but omitted for brevity...],
    ABRIDGEDTRANSFORMATION ["NAD27 to NAD83 (86) National",
        METHOD ["NTv2", ID ["EPSG", 9615]],
        PARAMETERFILE ["Latitude and longitude difference file", "NTv2_0.gsb"]
    ]
]
```

EXAMPLE 3

```

BOUNDCRS [
  SOURCECRS [...full WKT definition required here but omitted for brevity...],
  TARGETCRS [...full WKT definition required here but omitted for brevity...],
  ABRIDGEDTRANSFORMATION["Amersfoort to ETRS89 (3)",
    METHOD["Coordinate Frame", ID["EPSG", 1032]],
    PARAMETER["X-axis translation", 565.2369, ID["EPSG", 8605]],
    PARAMETER["Y-axis translation", 50.0087, ID["EPSG", 8606]],
    PARAMETER["Z-axis translation", 465.658, ID["EPSG", 8607]],
    PARAMETER["X-axis rotation", 0.407, ID["EPSG", 8608]],
    PARAMETER["Y-axis rotation", -0.351, ID["EPSG", 8609]],
    PARAMETER["Z-axis rotation", 1.870, ID["EPSG", 8610]],
    PARAMETER["Scale difference", 1.000004812, ID["EPSG", 8611]]
  ]
]

```

NOTE Compare the rotation values (here in arc-seconds because they are in an abridged coordinate transformation) with those in the fourth example in [17.3](#).

STANDARDSISO.COM : Click to view the full PDF of ISO 19162:2019

Annex A (normative)

Abstract test suite

A.1 Conformance of a WKT string describing a geodetic or geographic CRS

A.1.1 Structure

- a) Test purpose: to determine whether a text string representation of a geodetic coordinate reference system or of a geographic coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.1.2 Content

- a) Test purpose: to determine whether a text string representation of a geodetic coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a geodetic CRS or geographic CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#) and [8](#).
- d) Test type: capability.

A.2 Conformance of a WKT string describing a projected CRS

A.2.1 Structure

- a) Test purpose: to determine whether a text string representation of a projected coordinate reference system conforms to the characters and syntax required by this document
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.2.2 Content

- a) Test purpose: to determine whether a text string representation of a projected coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a projected CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7, 8](#) and [9](#).

- d) Test type: capability.

A.3 Conformance of a WKT string describing a vertical CRS

A.3.1 Structure

- a) Test purpose: to determine whether a text string representation of a vertical coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.3.2 Content

- a) Test purpose: to determine whether a text string representation of a vertical coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a vertical CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#) and [10](#).
- d) Test type: capability.

A.4 Conformance of a WKT string describing an engineering CRS

A.4.1 Structure

- a) Test purpose: to determine whether a text string representation of an engineering coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.4.2 Content

- a) Test purpose: to determine whether a text string representation of an engineering coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for an engineering CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#) and [11](#).
- d) Test type: capability.

A.5 Conformance of a WKT string describing a parametric CRS

A.5.1 Structure

- a) Test purpose: to determine whether a text string representation of a parametric coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.5.2 Content

- a) Test purpose: to determine whether a text string representation of a parametric coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a parametric CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#) and [12](#).
- d) Test type: capability.

A.6 Conformance of a WKT string describing a temporal CRS

A.6.1 Structure

- a) Test purpose: to determine whether a text string representation of a temporal coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.6.2 Content

- a) Test purpose: to determine whether a text string representation of a temporal coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a temporal CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#) and [13](#).
- d) Test type: capability.

A.7 Conformance of a WKT string describing a derived geodetic or derived geographic CRS

A.7.1 Structure

- a) Test purpose: to determine whether a text string representation of a derived geodetic coordinate reference system of a derived geographic coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.7.2 Content

- a) Test purpose: to determine whether a text string representation of a derived geodetic CRS or of a derived geographic CRS conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a derived geodetic CRS or for a derived geographic CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#), [8.2](#), [14.2](#) and [14.3](#).
- d) Test type: capability.

A.8 Conformance of a WKT string describing a derived projected CRS

A.8.1 Structure

- a) Test purpose: to determine whether a text string representation of a derived projected coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.8.2 Content

- a) Test purpose: to determine whether a text string representation of a derived projected coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a derived projected CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#), [8.2](#), [9.2](#), [9.3](#), [14.2](#) and [14.4](#).
- d) Test type: capability.

A.9 Conformance of a WKT string describing a derived vertical CRS

A.9.1 Structure

- a) Test purpose: to determine whether a text string representation of a derived vertical coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.9.2 Content

- a) Test purpose: to determine whether a text string representation of a derived vertical coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a derived vertical CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#), [10.2](#), [14.2](#) and [14.5](#).
- d) Test type: capability.

A.10 Conformance of a WKT string describing a derived engineering CRS

A.10.1 Structure

- a) Test purpose: to determine whether a text string representation of a derived engineering coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.10.2 Content

- a) Test purpose: to determine whether a text string representation of a derived engineering coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a derived engineering CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#), [11.2](#), [14.2](#) and [14.6](#).
- d) Test type: capability.

A.11 Conformance of a WKT string describing a derived parametric CRS

A.11.1 Structure

- a) Test purpose: to determine whether a text string representation of a derived parametric coordinate reference system conforms to the characters and syntax required by this document.

- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.11.2 Content

- a) Test purpose: to determine whether a text string representation of a derived parametric coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a derived parametric CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#), [12.2](#), [14.2](#) and [14.7](#).
- d) Test type: capability.

A.12 Conformance of a WKT string describing a derived temporal CRS

A.12.1 Structure

- a) Test purpose: to determine whether a text string representation of a derived temporal coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.12.2 Content

- a) Test purpose: to determine whether a text string representation of a derived temporal coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a derived temporal CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clauses 7](#), [13.2](#), [14.2](#) and [14.8](#).
- d) Test type: capability.

A.13 Conformance of a WKT string describing a compound CRS

A.13.1 Structure

- a) Test purpose: to determine whether a text string representation of a compound coordinate reference system conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.13.2 Content

- a) Test purpose: to determine whether a text string representation of a compound coordinate reference system conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a compound CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clause 15](#).
- d) Test type: capability.

A.14 Conformance of a WKT string describing coordinate metadata

A.14.1 Structure

- a) Test purpose: to determine whether a text string representation of coordinate metadata conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.14.2 Content

- a) Test purpose: to determine whether a text string representation of coordinate metadata conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for coordinate metadata
- c) Reference: [Clause 16](#).
- d) Test type: capability.

A.15 Conformance of a WKT string describing a coordinate transformation

A.15.1 Structure

- a) Test purpose: to determine whether a text string representation of a coordinate transformation conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.15.2 Content

- a) Test purpose: to determine whether a text string representation of a coordinate transformation conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a coordinate transformation, and that these use the correct terminology and syntax.

- c) Reference: [Clause 17](#).
- d) Test type: capability.

A.16 Conformance of a WKT string describing a point motion operation

A.16.1 Structure

- a) Test purpose: to determine whether a text string representation of a point motion operation conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.16.2 Content

- a) Test purpose: to determine whether a text string representation of a point motion operation conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a point motion operation, and that these use the correct terminology and syntax.
- c) Reference: [Clause 18](#).
- d) Test type: capability.

A.17 Conformance of a WKT string describing a concatenated coordinate operation

A.17.1 Structure

- a) Test purpose: to determine whether a text string representation of a concatenated coordinate operation conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.17.2 Content

- a) Test purpose: to determine whether a text string representation of a concatenated coordinate operation conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a concatenated coordinate operation, and that these use the correct terminology and syntax.
- c) Reference: [Clause 19](#).
- d) Test type: capability.

A.18 Conformance of a WKT string describing a bound CRS

A.18.1 Structure

- a) Test purpose: to determine whether a text string representation of a bound CRS conforms to the characters and syntax required by this document.
- b) Test method: verify that the structure of the WKT string conforms to the requirements of [Clause 5](#) and that text string syntax conforms to the requirements of [Clause 6](#).
- c) Reference: [Clauses 5](#) and [6](#).
- d) Test type: capability.

A.18.2 Content

- a) Test purpose: to determine whether a text string representation of a bound CRS conforms to the content required by this document.
- b) Test method: verify that the text string includes all of the elements specified for a bound CRS, and that these use the correct terminology and syntax.
- c) Reference: [Clause 20](#).
- d) Test type: capability.

STANDARDSISO.COM : Click to view the full PDF of ISO 19162:2019

Annex B (informative)

Recommended practice for implementation

B.1 General

This document defines the structure and content of WKT strings. It does not specifically address the requirements for implementations that write or read these strings. Some recommendations for such implementations are given in [B.2](#) to [B.8](#).

B.2 Keywords

B.2.1 Keyword case sensitivity

WKT keywords by definition are case insensitive. KEYWORD is equivalent to keyword is equivalent to KeyWord and to KEYwORD. Where human readability is important (as in the examples in this document) keywords should be written in only the <simple Latin upper case letter> set.

KEYWORD is not equivalent to KEY_WORD. The underscore character is significant.

B.2.2 Alternative keywords

Where alternative keywords are defined, as a minimum, implementations should be able to write the preferred and to read all alternatives.

For geodetic and vertical CRSs two alternative keywords (geodeticDatum/verticalDatum and TRF/VRF) are permitted. For brevity and backward compatibility, datum and vdatum are recommended. Should one of the alternatives be preferred then it is recommended to use geodeticDatum/verticalDatum only for classical datums and TRF/VRF only for modern frames.

B.2.3 Handling of unrecognised keywords

It is recognised that prior to the publication of this document some tokens that are not included in this document have been produced. Parsers reading CRS WKT strings and encountering unrecognised keywords should ignore the token and its content. They should be able to continue parsing the string without throwing an exception.

B.3 Characters

B.3.1 Handling of unrecognised characters

Implementations parsing WKT strings conformant with this document are not required to be able to read non-LATIN1 character sets. But if they are unable to do so they should be able to continue reading the WKT string by ignoring the <remark> token (keyword and content) without throwing an exception.

B.3.2 String length

The length of a WKT string or of its components is not prescribed. However, the following maximum lengths are recommended for implementations writing CRS WKT strings:

- The total length of a keyword should not exceed 24 characters.