
**Geographic information — Geography
Markup Language (GML) —**

**Part 2:
Extended schemas and encoding rules**

*Information géographique — Langage de balisage en géographie
(GML) —*

Partie 2: Schémas étendus et règles d'encodage





COPYRIGHT PROTECTED DOCUMENT

© ISO 2015, Published in Switzerland

All rights reserved. Unless otherwise specified, 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
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword	vi
Introduction	vii
1 Scope	1
2 Conformance	1
3 Normative references	2
4 Terms, definitions, symbols and abbreviated terms	2
4.1 General	2
4.2 Terms and definitions	2
4.3 Symbols and abbreviated terms	3
5 Conventions	3
5.1 MIME media types	3
5.2 XML namespaces	3
5.3 Deprecated parts of previous versions of GML	4
6 Additional base types	5
6.1 Target namespace	5
6.2 Localisable strings	5
6.2.1 LanguageStringType	5
6.2.2 Additional types based on LanguageStringType	5
6.3 TimePositionUnion	5
6.4 Requirements class	7
6.5 Conformance	7
7 Compact Encodings of Commonly Used GML Geometries	8
7.1 Target namespace	8
7.2 Introduction	8
7.3 SimplePolygon	8
7.4 SimpleRectangle	9
7.5 SimpleTriangle	10
7.6 SimpleArcString	10
7.7 SimpleArc	11
7.8 SimpleArcByCenterPoint	11
7.9 SimpleArcStringByBulge	12
7.10 SimpleArcByBulge	12
7.11 SimpleCircle	13
7.12 SimpleCircleByCenterPoint	13
7.13 SimpleMultiPoint	14
7.14 MultiPointPropertyType	14
7.15 Requirements class	14
7.16 Conformance	14
8 Triangulated Irregular Networks	15
8.1 Target namespace	15
8.2 Introduction	15
8.3 TriangulatedSurface	15
8.4 SimpleTrianglePatch	15
8.5 TIN	16
8.6 TINElement	16
8.7 TINElementPropertyType	17
8.8 TINElementType	17
8.9 Requirements class	19
8.10 Conformance	20
9 Linear Referencing	20
9.1 Target namespaces	20

9.2	Introduction	20
9.3	Basic Linear Referencing	21
9.3.1	Target namespace	21
9.3.2	Introduction	21
9.3.3	PositionExpression	21
9.3.4	PositionExpressionPropertyType	21
9.3.5	LinearElement	22
9.3.6	LinearElementPropertyType	23
9.3.7	StartValueType	23
9.3.8	LinearReferencingMethod	23
9.3.9	LinearReferencingMethodPropertyType	24
9.3.10	DistanceExpressionType	24
9.3.11	DistanceExpressionPropertyType	25
9.3.12	AlongReferent	25
9.3.13	AlongReferentPropertyType	25
9.3.14	Referent	26
9.3.15	ReferentPropertyType	27
9.3.16	MeasureType	27
9.3.17	LRMNameType	27
9.3.18	LRMTypeType	31
9.3.19	ReferentTypeType	32
9.3.20	LinearSRS	33
9.3.21	LinearSRSPPropertyType	33
9.4	Linear Referencing Towards Referent	34
9.4.1	Target namespace	34
9.4.2	Introduction	34
9.4.3	DualAlongReferent	34
9.4.4	DualAlongReferentPropertyType	34
9.5	Linear Referencing Offset	35
9.5.1	Target namespace	35
9.5.2	Introduction	35
9.5.3	LRMWithOffset	35
9.5.4	LRMWithOffsetPropertyType	35
9.5.5	LateralOffsetDistanceExpressionType	36
9.5.6	LateralOffsetExpressionType	36
9.5.7	VerticalOffsetExpressionType	37
9.5.8	LateralOffsetDirectionType	38
9.5.9	VerticalOffsetDirectionType	39
9.5.10	LateralOffsetLinearSRS	39
9.5.11	LateralOffsetLinearSRSPPropertyType	40
9.6	Linear Referencing Offset Vectors	41
9.6.1	Target namespace	41
9.6.2	Introduction	41
9.6.3	VectorOffsetDistanceExpressionType	41
9.6.4	VectorOffsetExpressionType	41
9.6.5	VectorOffsetLinearSRS	42
9.6.6	VectorOffsetLinearSRSPPropertyType	45
9.7	Requirements classes	45
9.8	Conformance	47
10	ReferenceableGrid	48
10.1	Target namespace	48
10.2	Clarifications to ISO 19123:2005	48
10.3	AbstractReferenceableGrid	49
10.4	ReferenceableGridByArray	50
10.5	ReferenceableGridByVectorsType, ReferenceableGridByVectors	51
10.6	ReferenceableGridByTransformation	57
10.7	gridCRS	58
10.8	Coverages using ReferenceableGrid	58

10.9	Requirements classes.....	58
10.10	Conformance.....	59
11	Code lists, dictionaries and definitions.....	60
11.1	Overview	60
11.2	Requirements class	61
11.3	Conformance.....	62
12	Encoding rule.....	62
12.1	Target namespace.....	62
12.2	Improved conversion rule.....	62
12.2.1	Conversion rule changes.....	62
12.2.2	Requirements class.....	64
12.2.3	Conformance.....	65
12.3	Association class conversion rule.....	67
12.3.1	Overview.....	67
12.3.2	Requirements class.....	68
12.3.3	Conformance.....	69
12.4	Encoding rule extensions.....	70
12.4.1	Overview.....	70
12.4.2	gmlexr:extendedEncodingRule	70
12.4.3	Requirements class.....	70
12.4.4	Conformance class.....	71
	Annex A (informative) Linear referencing method examples.....	73
	Bibliography	78

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. 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. 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 on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: [Foreword - Supplementary information](#).

The Geography Markup Language (GML) was originally developed within the Open Geospatial Consortium (OGC). The Committee responsible for this document is ISO/TC 211, *Geographic information/Geomatics*.

ISO 19136 consists of the following parts, under the general title *Geographic Information — Geography Markup Language*:

— *Part 2: Extended schemas and encoding rules*

A future Part 1 will revise and replace the currently available ISO 19136:2007.

Introduction

Geography Markup Language is an XML grammar written in XML Schema for the description of application schemas as well as the transport and storage of geographic information.

The key concepts used by Geography Markup Language (GML) to model the world are drawn from the ISO 19100- series of International Standards and the OpenGIS Abstract Specification.

A feature is an “abstraction of real world phenomena” (ISO 19101); it is a geographic feature if it is associated with a location relative to the Earth. So a digital representation of the real world may be thought of as a set of features. The state of a feature is defined by a set of properties, where each property may be thought of as a {name, type, value} triple.

The number of properties a feature may have, together with their names and types, is determined by its type definition. Geographic features with geometry are those with properties that may be geometry-valued. A feature collection is a collection of features that may itself be regarded as a feature; as a consequence a feature collection has a feature type and thus may have distinct properties of its own, in addition to the features it contains.

Following ISO 19109, the feature types of an application or application domain is usually captured in an application schema. A GML application schema is specified in XML Schema and can be constructed in two different and alternative ways:

- by adhering to the rules specified in ISO 19109 for application schemas in UML, and conforming to both the constraints on such schemas and the rules for mapping them to GML application schemas specified in this part of ISO 19136;
- by adhering to the rules for GML application schemas specified in this part of ISO 19136 for creating a GML application schema directly in XML Schema.

Both ways are supported by this part of ISO 19136. To ensure proper use of the conceptual modelling framework of the ISO 19100- series of International Standards, all application schemas are expected to be modelled in accordance with the General Feature Model as specified in ISO 19109. Within the ISO 19100- series, UML is the preferred language by which to model conceptual schemas.

GML specifies XML encodings, conformant with ISO 19118, of several of the conceptual classes defined in the ISO 19100- series of International Standards and the OpenGIS Abstract Specification. These conceptual models include those defined in:

- ISO/TS 19103, *Geographic information — Conceptual schema language* (units of measure, basic types);
- ISO 19107, *Geographic information — Spatial schema* (geometry and topology objects);
- ISO 19108, *Geographic information — Temporal schema* (temporal geometry and topology objects, temporal reference systems);
- ISO 19109, *Geographic information — Rules for application schemas* (features);
- ISO 19111, *Geographic information — Spatial referencing by coordinates* (coordinate reference systems);
- ISO 19123, *Geographic information — Schema for coverage geometry and functions*;
- ISO 19148, *Geographic information — Linear referencing*.

The aim is to provide a standardized encoding (i.e. a standardized implementation in XML) of types specified in the conceptual models specified by the International Standards listed above. If every application schema were encoded independently and the encoding process included the types from, for example, ISO 19108, then, without unambiguous and completely fixed encoding rules, the XML encodings would be different. Also, since every implementation platform has specific strengths and weaknesses, it is helpful to standardize XML encodings for core geographic information concepts modelled in the ISO 19100- series of International Standards and commonly used in application schemas.

In many cases, the mapping from the conceptual classes is straightforward, while in some cases the mapping is more complex (a detailed description of the mapping is part of this part of ISO 19136).

In addition, GML provides XML encodings for additional concepts not yet modelled in the ISO 19100-series of International Standards or the OpenGIS Abstract Specification, for example, dynamic features, simple observations or value objects.

Predefined types of geographic feature in GML include coverages and simple observations.

A coverage is a subtype of feature that has a coverage function with a spatiotemporal domain and a value set range of homogeneous 1- to n -dimensional tuples. A coverage may represent one feature or a collection of features “to model and make visible spatial relationships between, and the spatial distribution of, Earth phenomena” (OGC Abstract Specification Topic 6) and a coverage “acts as a function to return values from its range for any direct position within its spatiotemporal domain” (ISO 19123).

An observation models the act of observing, often with a camera or some other procedure, a person or some form of instrument (Merriam-Webster Dictionary: “an act of recognizing and noting a fact or occurrence often involving measurement with instruments”). An observation is considered to be a GML feature with a time at which the observation took place, and with a value for the observation.

A reference system provides a scale of measurement for assigning values to a position, time or other descriptive quantity or quality.

A coordinate reference system consists of a set of coordinate system axes that is related to the Earth through a datum that defines the size and shape of the Earth.

A temporal reference system provides standard units for measuring time and describing temporal length or duration.

A reference system dictionary provides definitions of reference systems used in spatial or temporal geometries.

Spatial geometries are the values of spatial feature properties. They indicate the coordinate reference system in which their measurements have been made. The “parent” geometry element of a geometric complex or geometric aggregate makes this indication for its constituent geometries.

Temporal geometries are the values of temporal feature properties. Like their spatial counterparts, temporal geometries indicate the temporal reference system in which their measurements have been made.

Spatial or temporal topologies are used to express the different topological relationships between features.

A units-of-measure dictionary provides definitions of numerical measures of physical quantities, such as length, temperature and pressure, and of conversions between units.

Geographic information — Geography Markup Language (GML) —

Part 2: Extended schemas and encoding rules

1 Scope

The Geography Markup Language (GML) is an XML encoding in compliance with ISO 19118 for the transport and storage of geographic information modelled in accordance with the conceptual modelling framework used in the ISO 19100- series of International Standards and including both the spatial and non-spatial properties of geographic features.

This part of ISO 19136 defines the XML Schema syntax, mechanisms and conventions that:

- provide an open, vendor-neutral framework for the description of geospatial application schemas for the transport and storage of geographic information in XML;
- allow profiles that support proper subsets of GML framework descriptive capabilities;
- support the description of geospatial application schemas for specialized domains and information communities;
- enable the creation and maintenance of linked geographic application schemas and datasets;
- support the storage and transport of application schemas and datasets;
- increase the ability of organizations to share geographic application schemas and the information they describe.

Implementers may decide to store geographic application schemas and information in GML, or they may decide to convert from some other storage format on demand and use GML only for schema and data transport.

This part of ISO 19136 builds on ISO 19136:2007 (GML 3.2), and extends it with additional schema components and requirements.

NOTE If an ISO 19109 conformant application schema described in UML is used as the basis for the storage and transportation of geographic information, this part of ISO 19136 provides normative rules for the mapping of such an application schema to a GML application schema in XML Schema and, as such, to an XML encoding for data with a logical structure in accordance with the ISO 19109 conformant application schema.

2 Conformance

This part of ISO 19136 defines XML implementations of concepts used in spatiotemporal datasets. It extends the XML implementations specified in ISO 19136:2007 (GML 3.2). Requirements and conformance classes specified in ISO 19136:2007 also apply for this part of ISO 19136.

XML instances that encode geographic information using one or more of the schemas specified in this part of ISO 19136 are the standardization target of the requirements stated in this part of ISO 19136.

The implementation is described using the XML Schema language and Schematron.

Conformance classes are specified in [Clauses 6](#) to [12](#) of this part of ISO 19136.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 19136:2007, *Geographic information — Geography Markup Language (GML)*

ISO 19148:2012, *Geographic information — Linear referencing*

OGC Technical Committee Policies and Procedures: *MIME Media Types for GML*¹⁾

ISO 8601:2004, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 13249-3:2011, *Information technology — Database languages — SQL multimedia and application packages — Part 3: Spatial*

4 Terms, definitions, symbols and abbreviated terms

4.1 General

For the purposes of this document, the terms, definitions, symbols and abbreviated terms listed in ISO 19136:2007 (GML 3.2), [Clause 4](#), apply.

4.2 Terms and definitions

In addition to the terms listed in ISO 19136:2007 (GML 3.2), the following terms and definitions apply.

4.2.1

grid coordinate reference system

grid CRS

coordinate reference system for the positions in a grid that uses a defined coordinate system congruent with the coordinate system described by the GridEnvelope and axisLabels of gml:GridType

Note 1 to entry: A grid CRS uses a defined coordinate system with the same grid point positions and origin as the GridEnvelope, with the same axisLabels, but need not define any limits on the grid size. This coordinate system is sometimes called the internal grid coordinate system.

4.2.2

referenceable grid

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

Note 1 to entry: If the coordinate reference system is related to the Earth by a datum, the grid is a georeferenceable grid.

[SOURCE: ISO 19123:2005, 4.1.33]

1) The latest version at the publication of this part of ISO 19136 is OGC document 09-144r1. The MIME type is currently in the registration process at IETF / IANA. The reference has intentionally been undated so that the reference is to the latest version in case changes to the MIME media type specification are required as part of the registration process.

4.3 Symbols and abbreviated terms

In addition to the symbols and abbreviated terms listed in ISO 19136:2007 (GML 3.2), the following are used in this document:

LRS	Linear Referencing System
OWL	Web Ontology Language
OWS	OGC Web Services
SKOS	Simple Knowledge Organization System

5 Conventions

5.1 MIME media types

For exchanging GML instance documents over the internet, the media type “application/gml+xml” is used as specified by the OGC Technical Committee Policies and Procedures: MIME Media Types for GML.

5.2 XML namespaces

The XML namespaces used within this part of ISO 19136 are listed in [Table 1](#). For each namespace, the namespace prefix used within this document and the canonical location of the all-components schema document are provided, too.

Table 1 — XML Namespaces

XML Namespace	Name-space prefix	Canonical location of all-components schema document
http://www.opengis.net/gml/3.2	gml	http://schemas.opengis.net/gml/3.2.1/gml.xsd
http://www.opengis.net/gml/3.3/xbt	gmlxbt	http://schemas.opengis.net/gml/3.3/extdBaseTypes.xsd
http://www.opengis.net/gml/3.3/ce	gmlce	http://schemas.opengis.net/gml/3.3/geometryCompact.xsd
http://www.opengis.net/gml/3.3/tin	gmltin	http://schemas.opengis.net/gml/3.3/tin.xsd
http://www.opengis.net/gml/3.3/lr	gmllr	http://schemas.opengis.net/gml/3.3/linearRef.xsd
http://www.opengis.net/gml/3.3/lrtr	gmlrtr	http://schemas.opengis.net/gml/3.3/linearRefTowardsReferent.xsd
http://www.opengis.net/gml/3.3/lro	gmlro	http://schemas.opengis.net/gml/3.3/linearRefOffset.xsd
http://www.opengis.net/gml/3.3/lrov	gmlrov	http://schemas.opengis.net/gml/3.3/linearRefOffsetVector.xsd
http://www.opengis.net/gml/3.3/rgrid	gmlr-grid	http://schemas.opengis.net/gml/3.3/referenceableGrid.xsd
http://www.opengis.net/gml/3.3/xer	gmlexr	http://schemas.opengis.net/gml/3.3/extdEncRule.xsd
http://www.w3.org/1999/xlink	xlink	http://www.w3.org/1999/xlink.xsd
http://www.w3.org/XML/1998/namespace	xml	http://www.w3.org/2001/xml.xsd
http://www.w3.org/2001/XMLSchema	xs (or default)	n/a

NOTE 1 A GML application schema conforming to this part of ISO 19136 will import the GML 3.2 schema plus zero or more additional GML 3.3 schemas as needed.

EXAMPLE 1 The following GML application schema imports both the GML 3.2 schema and the GML 3.3 compact geometry encoding.

```
<schema
  targetNamespace="http://www.example.org/app"
  xmlns:app="http://www.example.org/app"
  xmlns:gmlce="http://www.opengis.net/gml/3.3/ce"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  elementFormDefault="qualified"
  version="1.0.0">
  <import namespace="http://www.opengis.net/gml/3.2"
    schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.3/ce"
    schemaLocation="http://schemas.opengis.net/gml/3.3/geometryCompact.xsd"/>
  <!-- ... -->
  <element name="Parcel" substitutionGroup="gml:AbstractFeature"
    type="app:ParcelType"/>
  <complexType name="ParcelType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="geometry" type="gml:SurfacePropertyType"/>
          <!-- ... -->
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

NOTE 2 A GML instance conforming to this part of ISO 19136 will directly or indirectly reference the GML 3.2 schema and zero or more additional GML 3.3 schemas as needed. A schema location attribute in the root element of the document has to include the schema of the namespace of the root element.

EXAMPLE 2 A sample instance for the GML application schema from example 1 as returned from a Web Feature Service. The GML schemas are imported by the GML application schema that is referenced from the instance document:

```
<wfs:FeatureCollection
  timeStamp="2011-04-03T05:40:00Z"
  numberMatched="12"
  numberReturned="12"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:app="http://www.example.org/app"
  xmlns:gmlce="http://www.opengis.net/gml/3.3/ce"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=" http://www.example.org/app http://www.example.org/app.xsd
    http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0/wfs.xsd">
  <wfs:member>
    <app:Parcel gml:id="o1">
      <app:geometry>
        <gmlce:SimplePolygon gml:id="g1"
          srsName="http://www.opengis.net/def/crs/EPSG/0/4258">
          <gml:posList>50 6 50 7 51 7 51 6</gml:posList>
        </gmlce:SimplePolygon>
      </app:geometry>
      <!-- ... -->
    </app:Parcel>
  </wfs:member>
  <!-- ... -->
</wfs:FeatureCollection>
```

5.3 Deprecated parts of previous versions of GML

The verb **“deprecate”** provides notice that the referenced portion is being retained for backwards compatibility with earlier versions but may be removed from or superseded in this or a future version.

6 Additional base types

6.1 Target namespace

All schema components specified in [Clause 6](#) are in the target namespace:

<http://www.opengis.net/gml/3.3/xbt>

6.2 Localisable strings

6.2.1 LanguageStringType

The type `gmlxht:LanguageStringType` adds an optional `xml:lang` attribute to `xs:string`. It is provided as a base type for linguistic text for use within the GML schemas and in GML application schemas.

```
<complexType name="LanguageStringType">
  <simpleContent>
    <extension base="xs:string">
      <attribute ref="xml:lang"/>
    </extension>
  </simpleContent>
</complexType>
```

NOTE The name "LanguageStringType" was selected to align the naming with the equivalent type in the OWS Common standard.

6.2.2 Additional types based on LanguageStringType

The following types from GML 3.2 are defined in the <http://www.opengis.net/gml/3.3/xbt> schema with the same content model as in GML 3.2 with the exception that `gmlxht:LanguageStringType` is used in place of `xs:string`:

- `gml:CodeType`,
- `gml:CodeWithAuthorityType`

In addition, the global property elements `gmlxht:description` and `gmlxht:remarks` have an optional `xml:lang` attribute in their content model. These elements are in the substitution group of the GML 3.2 property element with the same local name.

- `gmlxht:description`
- `gmlxht:remarks`.

6.3 TimePositionUnion

In GML 3.2 the simple type `gml:TimePositionUnion` is a union of XML Schema simple types which instantiate the subtypes for temporal position described in ISO 19108. These are:

- `gml:CalDate` (union of `xs:date`, `xs:gYearMonth` and `xs:gYear`)
- `xs:time`
- `xs:dateTime`
- `xs:anyURI`
- `xs:decimal`

ISO 8601:2004, 4.1.2 specifies the Calendar Date and its representations with reduced accuracy (i.e., YYYY-MM and YYYY). `gml:CalDate` is designed to support their encoding.

ISO 8601:2004, 4.1.3 specifies the Ordinal Date, which is composed from the calendar year and the calendar day of the year (YYYY-DDD).

ISO 8601:2004, 4.1.4 specifies the Week Date, which is composed from the calendar year, the calendar week and the calendar day of the week (YYYY-Www-D). ISO 8601:2004, 4.1.4.3 specifies a Week Date representation with reduced accuracy that omits the day of the week component (YYYY-Www).

The ISO 8601:2004 Ordinal Date and Week Date with reduced accuracy are commonly used in some communities (for example, aviation) but are not supported by `gml:TimePositionUnion`. `gmlxht:TimePositionUnion` is provided to accommodate representations of these sibling date-representations from ISO 8601:2004 by adding `gmlxht:OrdDate` and `gmlxht:WeekDate` to the union.

```
<simpleType name="TimePositionUnion">
  <union memberTypes="gml:CalDate gmlxht:OrdDate gmlxht:WeekDate time dateTime anyURI
decimal"/>
</simpleType>

<simpleType name="OrdDate">
  <restriction base="string">
    <pattern value="-?[0-9]{4}-[0-9]{3}"/>
  </restriction>
</simpleType>

<simpleType name="WeekDate">
  <restriction base="string">
    <pattern value="-?[0-9]{4}-W(0[1-9]|[1-4][0-9]|5[0-3])(-[1-7])?"/>
  </restriction>
</simpleType>
```

The lexical space of `gmlxht:OrdDate` consists of finite-length sequences of characters of the form

'?'yyyy' ddd

where:

- '?'yyyy is a four-or-more digit optionally negative-signed numeral that represents the year; if more than four digits, leading zeros are prohibited, and '0000' is prohibited (note that a plus sign is not permitted);
- the '-' is a separator between parts of the date;
- the ddd is a three-digit numeral that represents the calendar day of the year where the first calendar day of any calendar year is represented by '001' and subsequent calendar days are numbered in ascending sequence.

The lexical space of `gmlxht:WeekDate` consists of finite-length sequences of characters of the form

'?'yyyy'-W' ww ('-' d)?

where:

- '?'yyyy is a four-or-more digit optionally negative-signed numeral that represents the year; if more than four digits, leading zeros are prohibited, and '0000' is prohibited (note that a plus sign is not permitted);
- the '-W' is a separator indicating that week-of-year follows;
- the ww is a two-digit numeral that represents the calendar week of the year where the first calendar week of any calendar year is represented by '01' and subsequent calendar weeks are numbered in ascending sequence;
- the '-' d (if present) is a one-digit numeral that represents the calendar day of the week where Monday shall be identified as calendar day '1' of any calendar week, and subsequent calendar days of the same calendar week shall be numbered in ascending sequence to Sunday (calendar day '7').

6.4 Requirements class

Table 2 specifies the requirement class “Basic type (extensions)”.

Table 2 — Requirements class “Basic types (extensions)”

Requirements Class	
http://www.opengis.net/spec/GML/3.3/req/xbt	
Target type	Data instance
Name	Basic types (extensions)
Dependency	http://www.opengis.net/doc/IS/GML/3.2/clause/2.4
Requirement	http://www.opengis.net/spec/GML/3.3/req/xbt/valid Any XML node whose content model is specified using schema components in the http://www.opengis.net/gml/3.3/xbt schema SHALL be well-formed and valid.
Requirement	http://www.opengis.net/spec/GML/3.3/req/xbt/iso8601 Any XML node whose content model is specified using <code>gml:TimePositionUnit</code> , <code>gml:OrdDate</code> or <code>gml:WeekDate</code> SHALL conform to ISO 8601:2004.

6.5 Conformance

Table 3 specifies the conformance class “Basic types (extensions)”.

Table 3 — Conformance class “Basic types (extensions)”

Conformance Class	
http://www.opengis.net/spec/GML/3.3/conf/xbt	
Requirements	http://www.opengis.net/spec/GML/3.3/req/xbt
Dependency	http://www.opengis.net/doc/IS/GML/3.2/clause/2.4
Test	http://www.opengis.net/spec/GML/3.3/conf/xbt/valid
	Requirement http://www.opengis.net/spec/GML/3.3/req/xbt/valid Test purpose Verify that any XML node whose content model is specified using schema components in the http://www.opengis.net/gml/3.3/xbt schema is well-formed and valid.
	Test method Validate the XML document using the XML schema document http://schemas.opengis.net/gml/3.3/extdBasicTypes.xsd . Pass if no errors reported. Fail otherwise.
	Test type Basic
Test	http://www.opengis.net/spec/GML/3.3/conf/xbt/iso8601
	Requirement http://www.opengis.net/spec/GML/3.3/req/xbt/iso8601 Test purpose Verify that any XML node whose content model is specified using <code>gml:TimePositionUnit</code> conforms to ISO 8601:2004.
	Test method Validate the XML document using the Schematron document http://schemas.opengis.net/gml/3.3/iso8601.sch . Replace “{nodesOfTypeTimePositionUnion}” in the file with an Xpath expression to all nodes with a content model of <code>gml:TimePositionUnit</code> . Pass if no errors reported. Fail otherwise.
	Test type Capability

7 Compact Encodings of Commonly Used GML Geometries

7.1 Target namespace

All schema components specified in [Clause 7](#) are in the target namespace:

<http://www.opengis.net/gml/3.3/ce>

7.2 Introduction

The simplified geometry encodings presented in [Clause 7](#) conform to the spatial schema (ISO 19107), but are offered as compact encoding alternatives to some of the existing GML geometries specified in GML 3.2. The compact encodings follow the idea of the compact `gml:LineString` encoding, which represents a curve that is equivalent to the `gml:Curve` encoding restricted to the use of `gml:LineStringSegment` elements as depicted in [Figure 1](#).

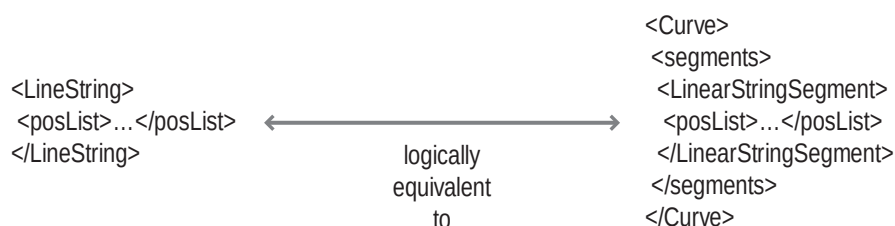


Figure 1 — LineString encoding

7.3 SimplePolygon

A `gmlce:SimplePolygon` is a specialized polygon that has a simplified encoding of the logically equivalent `gml:Surface` with a single `gml:PolygonPatch` as its surface patch consisting of a single `gml:LinearRing` as its exterior boundary and does not have any interior boundary. See [Figure 2](#).

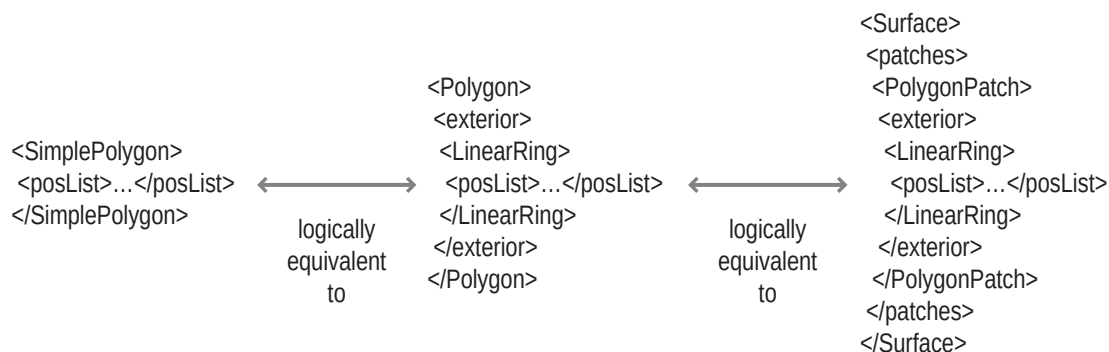


Figure 2 — SimplePolygon encoding

NOTE The boundary of a `gmlce:SimplePolygon` is coplanar and the polygon uses planar interpolation in its interior.

The usage of the term 'simple' here refers to a specialized polygon with a simplified encoding, which is *simply* connected (no interior rings) and uses a simple closed curve (no self-crossings) to represent its single boundary ring. The schema element declaration and corresponding type definition is as follows:

```

<element name="SimplePolygon"
  type="gmlce:SimplePolygonType"
  substitutionGroup="gmlce:AbstractSimplePolygon"/>
  
```

```

<element name="AbstractSimplePolygon" type="gml:AbstractSurfaceType"
  substitutionGroup="gml:AbstractSurface"/>

<complexType name="SimplePolygonType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <choice>
          <choice minOccurs="3" maxOccurs="unbounded">
            <element ref="gml:pos"/>
            <element ref="gml:pointProperty"/>
          </choice>
          <element ref="gml:posList"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

The last coordinate does not have to repeat the first coordinate in this simplified encoding, so only three control points are required to specify a simple polygon. For this reason, the inner choice declaration above has the corresponding occurrence constraint `minOccurs="3"`.

7.4 SimpleRectangle

A `gmlce:SimpleRectangle` is the special case of a simple polygon, which has exactly 4 control points in its boundary encoding representing the 4 corners of the rectangle. See [Figure 3](#).

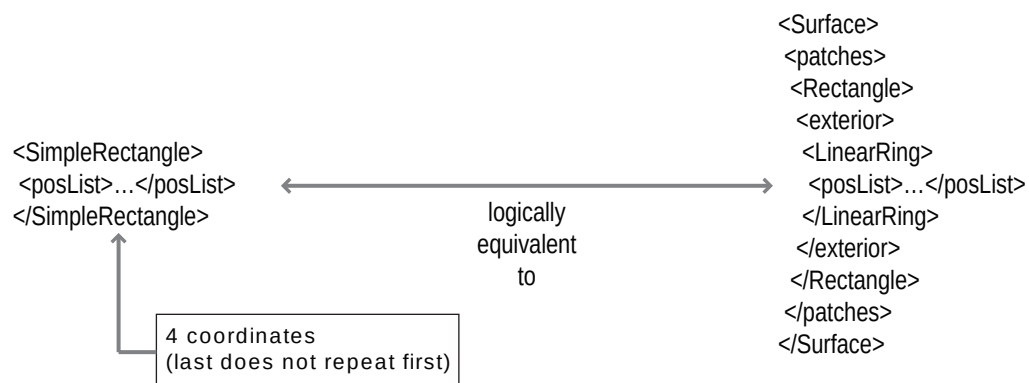


Figure 3 — SimpleRectangle encoding

NOTE The boundary of a `gmlce:SimpleRectangle` is coplanar and the polygon uses planar interpolation in its interior.

The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleRectangle" type="gmlce:SimpleRectangleType"
  substitutionGroup="gmlce:AbstractSimplePolygon"/>

<complexType name="SimpleRectangleType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <choice>
          <choice minOccurs="4" maxOccurs="4">
            <element ref="gml:pos"/>
            <element ref="gml:pointProperty"/>
          </choice>
          <element ref="gml:posList"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

</complexContent>
</complexType>

```

7.5 SimpleTriangle

A `gmlce:SimpleTriangle` is the special case of a simple polygon, which has exactly 3 control points in its boundary encoding representing the 3 corners of the triangle. See [Figure 4](#).

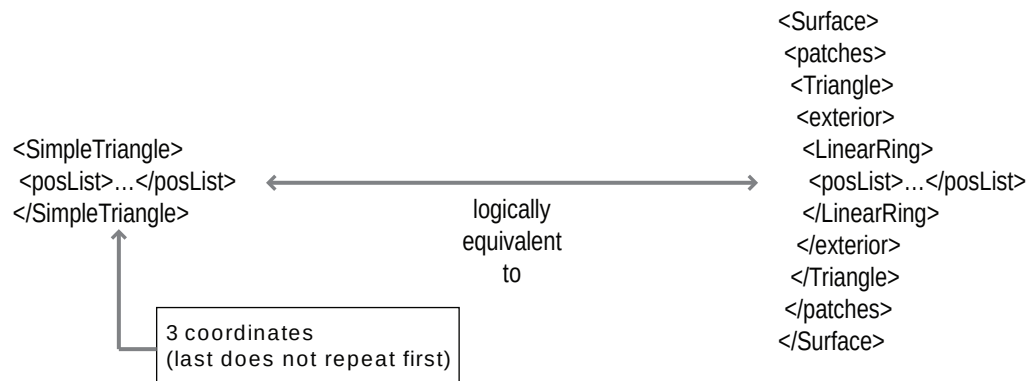


Figure 4 — SimpleTriangle encoding

The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleTriangle" type="gmlce:SimpleTriangleType"
substitutionGroup="gmlce:AbstractSimplePolygon"/>

```

```

<complexType name="SimpleTriangleType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <choice>
          <choice minOccurs="3" maxOccurs="3">
            <element ref="gml:pos"/>
            <element ref="gml:pointProperty"/>
          </choice>
          <element ref="gml:posList"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

EXAMPLE

```

<gmlce:SimpleTriangle gml:id="ID000"
  srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
  <gml:posList>50 10 49 10 49.5 11</gml:posList>
</gmlce:SimpleTriangle>

```

7.6 SimpleArcString

A `gmlce:SimpleArcString` is a specialized curve that has a simplified encoding of the logically equivalent `gml:Curve` with one or more `gml:Arc` segments. The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleArcString" type="gmlce:SimpleArcStringType"
substitutionGroup="gmlce:AbstractSimpleArcString"/>

<element name="AbstractSimpleArcString" type="gml:AbstractCurveType"
substitutionGroup="gml:AbstractCurve"/>

<complexType name="SimpleArcStringType">
  <complexContent>
    <extension base="gml:AbstractCurveType">

```

```

<sequence>
  <choice>
    <choice minOccurs="3" maxOccurs="unbounded">
      <element ref="gml:pos"/>
      <element ref="gml:pointProperty"/>
    </choice>
    <element ref="gml:posList"/>
  </choice>
</sequence>
<attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="circularArc3Points"/>
<attribute name="numArc" type="integer"/>
</extension>
</complexContent>
</complexType>

```

7.7 SimpleArc

A `gmlce:SimpleArc` is a specialized curve that has a simplified encoding of the logically equivalent `gml:Curve` with a single `gml:ArcString` segment. See [Figure 5](#).

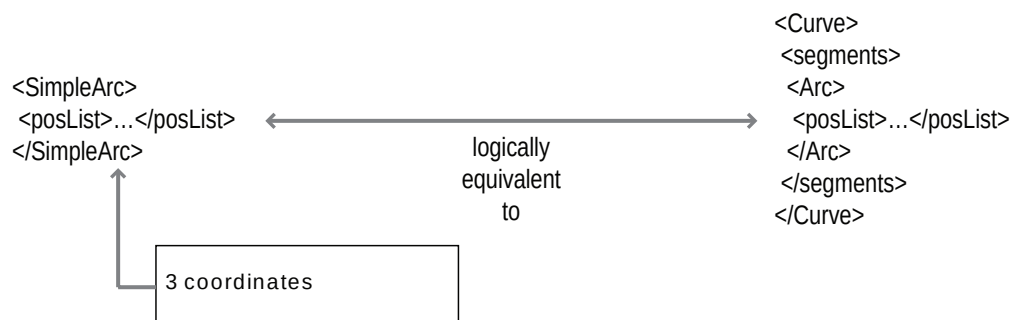


Figure 5 — SimpleArc encoding

The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleArc" type="gmlce:SimpleArcType"
  substitutionGroup="gmlce:AbstractSimpleArcString"/>

<complexType name="SimpleArcType">
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <choice>
          <choice minOccurs="3" maxOccurs="3">
            <element ref="gml:pos"/>
            <element ref="gml:pointProperty"/>
          </choice>
          <element ref="gml:posList"/>
        </choice>
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="circularArc3Points"/>
      <attribute name="numArc" type="integer" fixed="1"/>
    </extension>
  </complexContent>
</complexType>

```

7.8 SimpleArcByCenterPoint

A `gml:SimpleArcByCenterPoint` is a specialized curve that has a simplified encoding of the logically equivalent `gml:Curve` with a single `gml:ArcByCenterPoint` segment. As with the `gml:ArcByCenterPoint`, this representation can be used only in 2D. The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleArcByCenterPoint" type="gmlce:SimpleArcByCenterPointType"
  substitutionGroup="gmlce:AbstractSimpleArcString"/>

<complexType name="SimpleArcByCenterPointType">
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <choice>
          <choice>
            <element ref="gml:pos"/>
            <element ref="gml:pointProperty"/>
          </choice>
          <element ref="gml:posList"/>
        </choice>
        <element name="radius" type="gml:LengthType"/>
        <element name="startAngle" type="gml:AngleType"/>
        <element name="endAngle" type="gml:AngleType"/>
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="circularArcCenterPointWithRadius"/>
      <attribute name="numArc" type="integer" use="required" fixed="1"/>
    </extension>
  </complexContent>
</complexType>

```

NOTE Start and end angle are mandatory in this encoding unlike `gml:ArcByCentrePoint`.

7.9 SimpleArcStringByBulge

A `gmlce:SimpleArcStringByBulge` is a specialized curve that has a simplified encoding of the logically equivalent `gml:Curve` with one or more `gml:ArcByBulge` segments. The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleArcStringByBulge" type="gmlce:SimpleArcStringByBulgeType"
  substitutionGroup="gmlce:AbstractSimpleArcString"/>

<complexType name="SimpleArcStringByBulgeType">
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <choice>
          <choice minOccurs="2" maxOccurs="unbounded">
            <element ref="gml:pos"/>
            <element ref="gml:pointProperty"/>
          </choice>
          <element ref="gml:posList"/>
        </choice>
        <element name="bulge" type="double" maxOccurs="unbounded"/>
        <element name="normal" type="gml:VectorType" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="circularArc2PointWithBulge"/>
      <attribute name="numArc" type="integer"/>
    </extension>
  </complexContent>
</complexType>

```

7.10 SimpleArcByBulge

A `gml:SimpleArcByBulge` is a specialized curve that has a simplified encoding of the logically equivalent `gml:Curve` with one `gml:ArcByBulge` segment. The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleArcByBulge" type="gmlce:SimpleArcByBulgeType"
  substitutionGroup="gmlce:AbstractSimpleArcString"/>

<complexType name="SimpleArcByBulgeType">
  <complexContent>

```

```

<extension base="gml:AbstractCurveType">
  <sequence>
    <choice>
      <choice minOccurs="2" maxOccurs="2">
        <element ref="gml:pos"/>
        <element ref="gml:pointProperty"/>
      </choice>
      <element ref="gml:posList"/>
    </choice>
    <element name="bulge" type="double" maxOccurs="unbounded"/>
    <element name="normal" type="gml:VectorType" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="circularArc2PointWithBulge"/>
  <attribute name="numArc" type="integer" fixed="1"/>
</extension>
</complexContent>
</complexType>

```

7.11 SimpleCircle

A `gmlce:SimpleCircle` is a specialized curve that has a simplified encoding of the logically equivalent `gml:Curve` with a single `gml:Circle` segment. See [Figure 6](#).

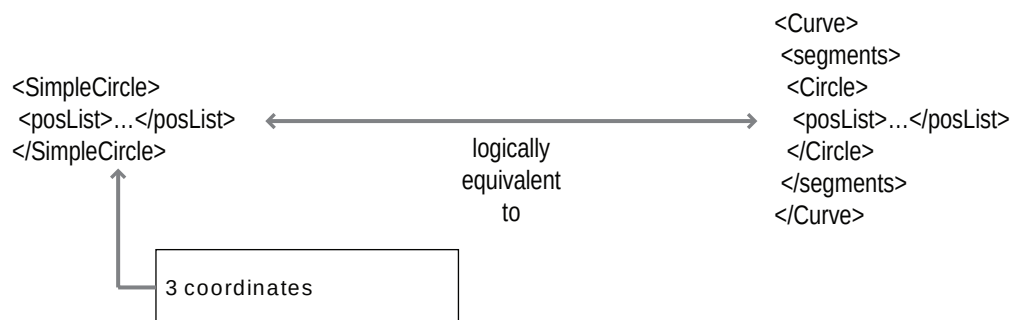


Figure 6 — SimpleCircle encoding

Consistent with `gml:Circle`, a `gmlce:SimpleCircle` is an arc whose ends coincide to form a simple closed loop. The three control points shall be distinct non-co-linear points for the circle to be unambiguously defined. The arc is simply extended past the third control point until the first control point is encountered.

The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleCircle" type="gmlce:SimpleArcType"
  substitutionGroup="gmlce:AbstractSimpleArcString"/>

```

7.12 SimpleCircleByCenterPoint

A `gmlce:SimpleCircleByCenterPoint` is a specialized curve that has a simplified encoding of the logically equivalent `gml:Curve` with a single `gml:CircleByCenterPoint` segment. As with the `gml:CircleByCenterPoint`, this representation can be used only in 2D. The schema element declaration and corresponding type definition is as follows:

```

<element name="SimpleCircleByCenterPoint"
  type="gmlce:SimpleArcByCenterPointType"
  substitutionGroup="gmlce:AbstractSimpleArcString"/>

```

Start and end angles are mandatory and should not be identical; they should differ by 360 degrees.

7.13 SimpleMultiPoint

`gmlce:SimpleMultiPoint` implements, and provides a simplified encoding for, ISO 19107 `GM_MultiPoint` (see ISO 19107:2003, 6.5.4). A `gmlce:SimpleMultiPoint` consists of a list of `DirectPositions`.

```
<complexType name="SimpleMultiPointType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:posList"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="SimpleMultiPoint" type="gmlce:SimpleMultiPointType"
substitutionGroup="gml:AbstractGeometricAggregate" />
```

7.14 MultiPointPropertyType

A property that has a collection of points as its value domain may either be an appropriate geometry element encapsulated in an element of this type or an XLink reference to a remote geometry element (where remote includes geometry elements located elsewhere in the same document). The rules stated in ISO 19136:2007, 7.2.3.4 apply.

```
<complexType name="MultiPointPropertyType">
  <choice minOccurs="0">
    <element ref="gml:MultiPoint"/>
    <element ref="gmlce:SimpleMultiPoint"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

7.15 Requirements class

[Table 4](#) specifies the requirements class “Compact Geometry Encoding”.

Table 4 — Requirements class “Compact Geometry Encoding”

Requirements Class	
http://www.opengis.net/spec/GML/3.3/req/ce	
Target type	Data instance
Name	Compact Geometry Encoding
Dependency	http://www.opengis.net/doc/IS/GML/3.2/clause/2.4
Requirement	http://www.opengis.net/spec/GML/3.3/req/ce/valid Any XML node whose content model is specified using schema components in the http://www.opengis.net/gml/3.3/ce schema SHALL be well-formed and valid.

7.16 Conformance

[Table 5](#) specifies the conformance class “Compact Geometry Encoding”.

Table 5 — Conformance class “Compact Geometry Encoding”

Conformance Class	
http://www.opengis.net/spec/GML/3.3/conf/ce	
Requirements	http://www.opengis.net/spec/GML/3.3/req/ce
Dependency	http://www.opengis.net/doc/IS/GML/3.2/clause/2.4
Test	http://www.opengis.net/spec/GML/3.3/conf/ce/valid
	Requirement http://www.opengis.net/spec/GML/3.3/req/ce/valid
	Test purpose Verify that any XML node whose content model is specified using schema components in the http://www.opengis.net/gml/3.3/ce schema is well-formed and valid.
	Test method Validate the XML document using the XML schema document http://schemas.opengis.net/gml/3.3/geometryCompact.xsd . Pass if no errors reported. Fail otherwise.
	Test type Basic

8 Triangulated Irregular Networks

8.1 Target namespace

All schema components specified in [Clause 8](#) are in the target namespace:

<http://www.opengis.net/gml/3.3/tin>

8.2 Introduction

The support for Triangulated Irregular Networks (TIN) in ISO 19136:2007 (GML 3.2) is consistent with ISO 19107:2003 and ISO 19125-1:2004. However, three areas of concern of the TIN model in these standards are addressed by the new schema components to support typical civil engineering practice:

- Need for more robust set of “TIN elements”
- Need for a simpler triangle encoding
- Need for a by-reference encoding of the triangulated surface

8.3 TriangulatedSurface

`gmltin:TriangulatedSurface` implements ISO 19107 GM_TriangulatedSurface (see ISO 19136:2007, D.2.3.4 and ISO 19107:2003, 6.4.37).

A triangulated surface is a polyhedral surface that is composed only of triangles. There is no restriction on how the triangulation is derived.

`gml:patches` encapsulates the triangles of the triangulated surface. All patches shall be simple triangle patches.

```
<element name="TriangulatedSurface" type="gml:SurfaceType"
substitutionGroup="gml:Surface"/>
```

8.4 SimpleTrianglePatch

`gmltin:SimpleTrianglePatch` is based upon the pattern used in the GML 3.3 SimpleTriangle, rather than the GML 3.2 Triangle.

A `gmltin:SimpleTrianglePatch` is a surface patch that is defined by a set of three points. The points shall be coplanar and the resultant simple triangle uses planar interpolation.

boundary of simple triangle patch can be obtained by constructing a `gml:LinearRing`, defined by four coordinate tuples representing the three points such that the first point is repeated as the last point, with linear interpolation between the points.

`gml:interpolation` is fixed to “planar”, i.e. an interpolation shall return points on a single plane. The boundary of the patch shall be contained within that plane.

```
<complexType name="SimpleTrianglePatchType">
  <complexContent>
    <extension base="gml:AbstractSurfacePatchType">
      <sequence>
        <choice>
          <choice minOccurs="3" maxOccurs="3">
            <element ref="gml:pos"/>
            <element ref="gml:pointProperty"/>
          </choice>
          <element ref="gml:posList"/>
        </choice>
      </sequence>
      <attribute name="interpolation"
        type="gml:SurfaceInterpolationType"
        fixed="planar"/>
    </extension>
  </complexContent>
</complexType>

<element name="SimpleTrianglePatch" type="gmltin:SimpleTrianglePatchType"
  substitutionGroup="gml:AbstractSurfacePatch"/>
```

8.5 TIN

`gmltin:TIN` is a triangulated surface that uses the Delaunay algorithm or a similar algorithm complemented with consideration of constraints defined by TIN Elements (`gmltin:tinElements`) and possibly a maximum length of triangle sides (`gmltin:maxLength`).

```
<complexType name="TINType">
  <complexContent>
    <extension base="gml:SurfaceType">
      <sequence>
        <element name="tinElement" type="gmltin:TINElementPropertyType" minOccurs="0"
          maxOccurs="unbounded"/>
        <element name="maxLength" type="gml:LengthType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="TIN" type="gmltin:TINType"
  substitutionGroup="gmltin:TriangulatedSurface"/>
```

The set of the positions (three or more) used as posts for this TIN (corners of the triangles in the TIN) are specified either in TIN Element control points or group spots.

`gmltin:TIN` implements ISO/IEC 13249-3 ST_TIN (see ISO/IEC 13249-3:2011, 8.6.1 ST_TIN Type) which extends ISO 19107 (see ISO 19136:2007, D.2.3.4 and ISO 19107:2003, 6.4.37).

8.6 TINElement

`gmltin:TINElement` implements ISO/IEC 13249-3 ST_TINElement (see ISO/IEC 13249-3:2011, 15.1.1 ST_TINElement Type). It specifies elements associated with a TIN which can be:

- 1) constraints (boundary, break line, soft break, control contour, break void, drape void, void, hole, stop line) applied to a TIN surface after the initial Delaunay triangulation which modify the surface,

- 2) points (random points, group spots) which are the vertices of simple triangles that define the TIN surface or vertices of constraints, or
- 3) user-defined.

```
<complexType name="TINElementType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="elementType"
          type="gmltin:TINElementTypeType"/>
        <element name="elementID" type="integer" minOccurs="0"/>
        <element name="elementTag" type="string" minOccurs="0"/>
        <element name="elementGeometry"
          type="gml:GeometryPropertyType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="TINElement" type="gmltin:TINElementType"
  substitutionGroup="gml:AbstractObject"/>
```

The `gmltin:elementType` property specifies the type of TIN element. Allowable values are specified by the `gmltin:TINElementTypeType` codelist.

The `gmltin:elementID` property allows for an optional integer value to numerically identify the TIN element.

The `gmltin:elementTag` property allows for an optional string value to alphanumerically identify the TIN element.

The `gmltin:elementGeometry` property specifies the geometry of the TIN element. The allowable geometry type is dependent upon the value of the `gmltin:elementType`:

- random points and group spot TIN elements shall have a 3D `gml:MultiPoint` geometry
- boundary, break void and void TIN elements shall have a 3D `gml:Polygon` geometry
- drape void and hole TIN elements shall have a 2D or 3D `gml:Polygon` geometry
- break line, soft break and control contour TIN elements shall have a 3D `gml:LineString` geometry
- stop line TIN elements shall have a 2D or 3D `gml:LineString` geometry
- for a user-defined TIN element, the choice of `gml:GeometryPropertyType` shall be user-defined.

8.7 TINElementPropertyType

A property that has a TIN element as its value domain may either be an appropriate TIN element encapsulated in an element of this type or an XLink reference to a remote TIN element (where remote includes TIN elements located elsewhere in the same document). Either the reference or the contained element shall be given, but neither both nor none.

```
<complexType name="TINElementPropertyType">
  <sequence minOccurs="0">
    <element ref="gmltin:TINElement"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

8.8 TINElementTypeType

`gmltin:TINElementTypeType` is a codelist which is the union of an enumeration of pre-defined TIN element types and a pattern for specifying a user-defined TIN type.

```
<simpleType name="TINElementTypeType">
  <union>
    <simpleType>
      <restriction base="string">
        <enumeration value="randomPoints"/>
        <enumeration value="groupSpot"/>
        <enumeration value="boundary"/>
        <enumeration value="breakline"/>
        <enumeration value="softBreak"/>
        <enumeration value="controlContour"/>
        <enumeration value="breakVoid"/>
        <enumeration value="drapeVoid"/>
        <enumeration value="void"/>
        <enumeration value="hole"/>
        <enumeration value="stopLine"/>
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="string">
        <pattern value="other:\w{2,}" />
      </restriction>
    </simpleType>
  </union>
</simpleType>
```

A TIN element of type 'random points' (`gmltin:randomPoints`) represents points on the surface of known elevation from which triangles can be generated or which represent linestring points contained in other TIN element geometries.

A TIN element of type 'group spot' (`gmltin:groupSpot`) represents a collection of related points on the surface of known elevation from which triangles can be generated or which represent linestring points contained in other TIN element geometries.

A TIN element of type 'boundary' (`gmltin:boundary`) is used to specify the boundary of the TIN surface. As a constraint applied to a TIN surface after the initial Delaunay triangulation, it causes the surface to be clipped to the boundary TIN element's `gml:Polygon` geometry value. This may result in the elimination or addition of points contained in the initial random points or group spots TIN elements. It may also result in localized re-triangulation at the boundary. It is implementation-defined whether interior boundaries are supported.

A TIN element of type 'breakline' (`gmltin:breakline`) is used to represent a local ridge or depression in the TIN surface. When a breakline is specified for a TIN surface, simple triangle patches must be adjusted so that no triangle is crossed by the breakline. Part or all of the breakline becomes an edge of two or more triangles. The elevation along the breakline takes precedence over the elevation of the original triangulated surface for the entire length of the breakline.

A TIN element of type 'soft break' (`gmltin:softBreak`) behaves as a 'breakline' (see above) except that contour lines generated for the surface can be smoothed where they cross soft breaks.

A TIN element of type 'control contour' (`gmltin:controlContour`) behaves as a 'breakline' (see above). The z coordinate values must be identical for all points in the 'control contour' `gml:LineString` geometry. Triangles within the vicinity of a control contour need to be assessed during re-triangulation to insure that they are not zero slope triangles (all three vertices fall on the same control contour).

Voids enclose a voided area of the TIN surface. They can be represented by either break, drape or (regular) void types of TIN element. Void geometry is of type `gml:Polygon`, which might include interior boundaries. Triangles within a void still exist in the TIN surface but are considered to be void. Triangles inside of the `gml:Polygon` interior boundary are in the exterior of the `gml:Polygon` and therefore not in the void: these triangles are therefore not considered to be void.

For a void TIN element of type 'break void' (`gmltin:breakVoid`) the boundary linestrings (`gml:LineString`) of the 3D `gml:Polygon` behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the break void boundary. Part or all of the break void boundary becomes an edge of two or more triangles. The elevation of this

break void boundary takes precedence over the elevation of the original triangulated surface for the entire length of the boundary.

For a void TIN element of type 'drape void' (`gmltin:drapeVoid`) the boundary linestrings (`gml:LineString`) of the 2D or 3D `gml:Polygon` behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the drape void boundary. Part or all of the drape void boundary becomes an edge of two or more triangles. However, for drape voids, the elevation of the original triangulated surface takes precedence over the elevation of the drape void boundary.

For a void TIN element of type (regular) 'void' (`gmltin:void`) the boundary linestrings (`gml:LineString`) of the 3D `gml:Polygon` behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the void boundary. Part or all of the void boundary becomes an edge of two or more triangles. However, for regular voids, only the elevations of the void boundary vertices take precedence over the elevation of corresponding points on the original surface. That is, these vertices are treated as points for triangulating. The regular void boundaries between these vertices are handled as drape void boundaries – elevations from the original surface take precedence.

A hole is an area of a TIN surface, defined by a 2D or 3D `gml:Polygon`, which is to be treated as a hole in the surface in that the triangles within this area still exist but are considered to be part of the hole. The difference between a void and a hole is realized when two TIN surfaces are merged. When merging TIN surface A with TIN surface B, a void in surface A will take precedence over what is in the same area in B, the result being retention of the voided A triangles. A hole in surface A results in that part of surface B showing through the hole and becoming part of the resultant merged surface. Though GML does not support merging of TINs, it is necessary to persist the distinction between voids and holes to make this information available to applications which do.

Hole boundaries are different than an interior boundary of a surface. The hole is still part of the interior of the surface since the triangles still exist. They are merely considered as being in the hole so applications know how to deal with them in a visibility context. This also allows for "islands" to exist inside the holes (if the hole polygon has interior rings): here the triangles exist and are not hole triangles. Had the triangles in the hole been eliminated, islands could not be supported in a single surface.

A TIN element of type 'hole' (`gmltin:hole`) encloses an area of the TIN surface designated as a hole. When a hole is specified for `gmltin:TINElement`, the boundary linestrings (`gml:LineString`) of the 2D or 3D `gml:Polygon` behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the hole boundary. Part or all of the hole boundary becomes an edge of two or more triangles. Hole boundaries are treated like drape void boundaries in that the elevation of the original triangulated surface takes precedence over the elevation of the hole boundary, if present.

A TIN element of type 'stopLine' (`gmltin:stopLine`) is used to specify areas where the local continuity or regularity of the TIN surface is questionable. It is implementation-defined whether triangles in the simple triangle patch collection whose boundaries are (2D) crossed by a stop line are removed from the collection of simple triangle patches or are retained but enclosed within a 'drape void' (`gmltin:drapeVoid`) type of `gmltin:TINElement`.

8.9 Requirements class

[Table 6](#) specifies the requirements class "Triangulated Irregular Networks".