

Item E - Geometrical and geographical data

Introduction

AIXM 5.1 uses the Geographical Markup Language (GML) for the encoding of geometrical and geographical data. This is done by directly including the GML 3.2.1 Schema into the AIXM XML Schema. In order to automatically generate the text of the NOTAM messages, it is also necessary to translate back the GML encodings into human readable information. Therefore, the purpose of this section is to provide guidelines for the decoding from GML of the geographical and geometrical data that is in the scope of the Event Scenarios covered by the specification.

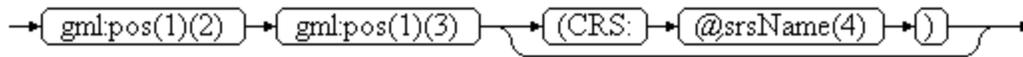
As explained in the data encoding section, all geometrical/geographical data is encoded in AIXM/GMI as one of the following elements:

- aixm:Point or aixm:ElevatedPoint;
- aixm:Curve or aixm:ElevatedCurve;
- aixm:Surface or aixm:ElevatedSurface;

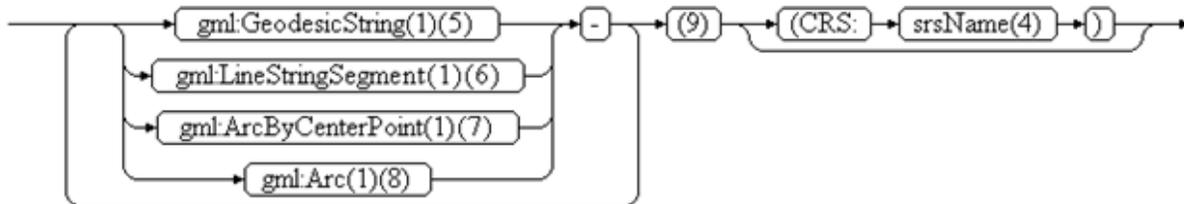
Decoding rules

This section provides rules for converting these geometries into NOTAM text.

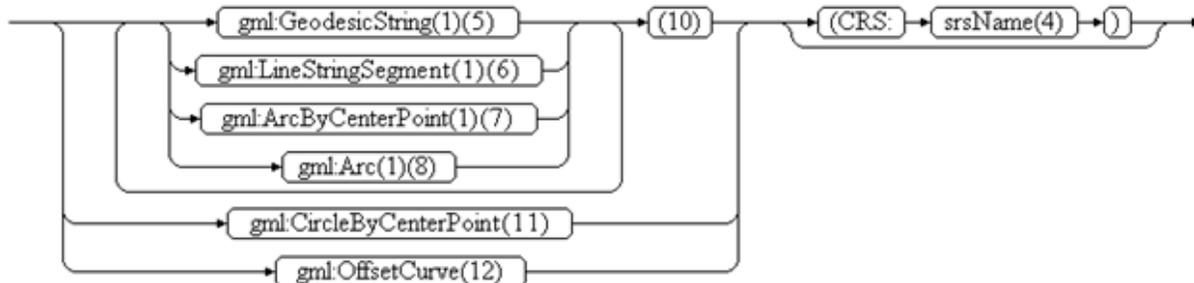
template_aixmPoint



template_aixmCurve



template_aixmSurface

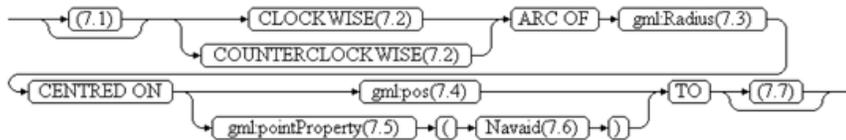


Reference	Rules
(1)	<p>Identify the srsName that applies (see section 3.2 of the "Use of GML for aviation data" guidance document). If this specification was followed for the data encoding, then it is an attribute of the ancestor aixm:Point, aixm:ElevatedPoint, aixm:Curve, aixm:ElevatedCurve, aixm:Surface or aixm:ElevatedSurface element.</p> <ul style="list-style-type: none"> • If srsName="urn:ogc:def:crs:EPSG::4326", then the first value in the gml:pos element is the latitude and the second one is the longitude, both in WGS-84; • If another srsName is found, then specific decoding rules must be available for it.

(2)	Convert the latitude value into the format "DDMMSSH" or "DDMMSS.ssH" as appropriate; trim to 2 decimals if necessary, but do not add trailing zeros after the decimal point!
(3)	Convert the longitude value into the format "DDMMSSH" or "DDMMSS.ssH" as appropriate; trim to 2 decimals if necessary, but do not add trailing zeros after the decimal point!
(4)	If srsName="urn:ogc:def:crs:EPSG::4326", then leave empty this branch. Otherwise, decode the srsName as appropriate.
(5)	Convert each pair of latitude/longitude values of the child gml:posList into the format "DDMMSSH DDDMMSSH"; insert " - " as separator between consecutive "lat long" pairs.
(6)	Convert each pair of latitude/longitude values of the child gml:posList into the format "DDMMSSH DDDMMSSH"; insert " - " as separator between consecutive "lat long" pairs.

(7) Decode the gml:ArcByCenterPoint as detailed in the diagram below

template_gmlArcByCenterPoint



Important note: It is assumed that the srsName applicable to the gml:ArcByCentrePoint has the value "urn:ogc:def:crs:EPSG::4326". Otherwise, the result of this decoding might be incorrect.

Reference	Rule
(7.1)	If the gml:ArcByCenterPoint is the first child element of gml:segments, then calculate the position of the start of the arc using the gml:pos, gml:radius and gml:startAngle data and insert here in format "DDMMSSH DDDMMSSH - ". Otherwise leave empty (the start of the arc is the end of the previous curve segment).
(7.2)	If gml:startAngle is lower than gml:endAngle, then insert "CLOCKWISE", otherwise insert "COUNTERCLOCKWISE".
(7.3)	Insert the value of the gml:radius element followed by the value of its uom attribute, decoded as follows: <ul style="list-style-type: none"> m -> "M" km -> "KM" [nmi_i] -> "NM"
(7.4)	If a gml:pos is provided for the centre of the arc, then decode here the lat/long values of that element in the format "DDMMSSH DDDMMSSH";
(7.5)	If a gml:pointProperty is used in order to refer to the position of another gml:Point element, then decode here the lat /long values of its gml:pos child element in the format "DDMMSSH DDDMMSSH".
(7.6)	If a gml:pointProperty is used, then insert here the type and the designator of the Navaid that owns the referenced gml: Point.
(7.7)	If the gml:ArcByCenterPoint is the last child element of gml:segments or it is followed by another gml: ArcByCenterPoint, then calculate the position of the end of the arc using the gml:pos, gml:radius and gml:endAngle data and insert here in format "DDMMSSH DDDMMSSH". Otherwise leave empty (the end of the arc is the start of the next curve segment).

(8) Decode the gml:Arc as detailed in the diagram below:

template_gmlArc



Reference	Rules
(8.1)	Decode the lat/long data of the first gml:pos child element of the gml:Arc in the format "DDMMSSH DDDMMSSH"
(8.2)	Decode the lat/long data of the second gml:pos child element of the gml:Arc in the format "DDMMSSH DDDMMSSH"
(8.3)	Decode the lat/long data of the third gml:pos child element of the gml:Arc in the format "DDMMSSH DDDMMSSH"

(9) If there exist more than one child element of gml:segments, it is possible that the individual decoding of each gml:GeodesicString, gml:LineStringSegment, gml:ArcByCenterPoint or gml:Arc results into duplicate pairs of lat/long values (because the start of one curve may be the end of the previous one). To prevent this situation, inspect check the complete decoding of the geometry and eliminate any duplicate consecutive lat/long values.

(10) If there exist more than one child element of gml:segments, it is possible that the individual decoding of each gml:GeodesicString, gml:LineStringSegment, gml:ArcByCenterPoint or gml:Arc results into duplicate pairs of lat/long values (because the start of one curve may be the end of the previous one). To prevent this situation, inspect check the complete decoding of the geometry and eliminate any duplicate consecutive lat/long values. Pay attention that the first and the last position should be identical. If this is not the case, a warning message shall be raised.

(11) Decode the gml:CircleByCenterPoint as detailed in the diagram below:

template_gmlCircleByCenterPoint



(12) If an [AirspaceVolume](#) with a [centreline](#) is encountered, then it is assumed that it is the encoding of a corridor type of geometry.