



ENTERPRISE ARCHITECT

User Guide Series

Geography Markup Language (GML)

Author: Sparx Systems

Date: 2022-11-16

Version: 16.1

CREATED WITH  **ENTERPRISE
ARCHITECT**

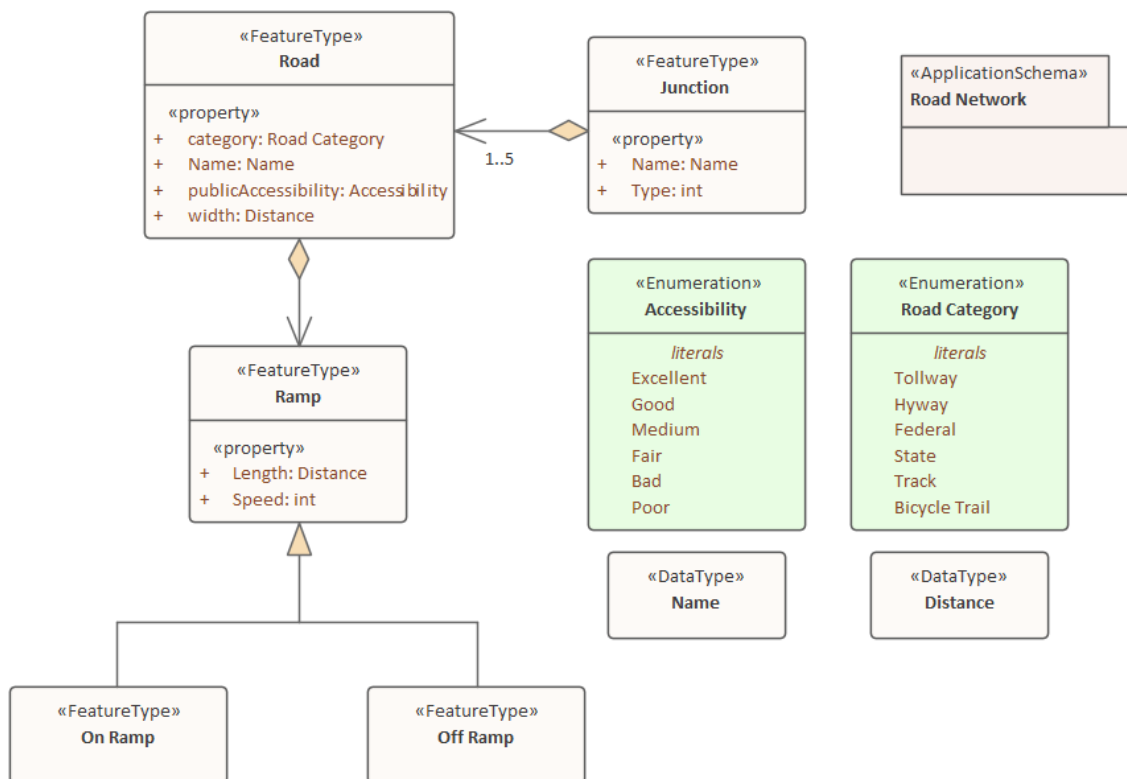
Table of Contents

Geography Markup Language (GML)	3
More Information	4
Example Diagram	5
Getting Started	6
Generate GML Application Schema	7
Modeling with GML	9
GML Toolbox Pages	11

Geography Markup Language (GML)

Model Geographic Features and Generate Application Schemas

You can create expressive and collaborative models of the important features in your domain and use these to generate Geography Markup Language (GML) compliant application schemas that can be consumed by other applications. Many of the disruptive forces and technologies that have changed the way we interact with each other and the world we inhabit, involve geographic locations and features. We drive along roads and stop at lookouts to view coastal features or cityscapes, we travel abroad to view monuments and buildings such as churches and museums, we rely on wind farms for energy and we take off and land at airports to name a few. You can model any geographic features of interest using Enterprise Architect's implementation of the Geography Markup Language which is fundamental for geographic information systems as well as its use as an open interchange format for geographic transactions on the Internet.



GML Model of roads showing two Features with properties that access two Data Types and Enumerations

GML for Enterprise Architect is an implementation of the Open Geospatial Consortium's Geography Markup Language (GML) 3.3 , which provides an XML grammar for geographical feature modeling capabilities within Enterprise Architect at or later than Release 10.

Through GML, you can:

- Apply a UML Profile for the Geography Markup Language (GML) 3.3
- Make use of customized diagram types and toolbox pages, for convenient access to elements and relationships to model geographical features effectively
- Generate GML Application Schema files

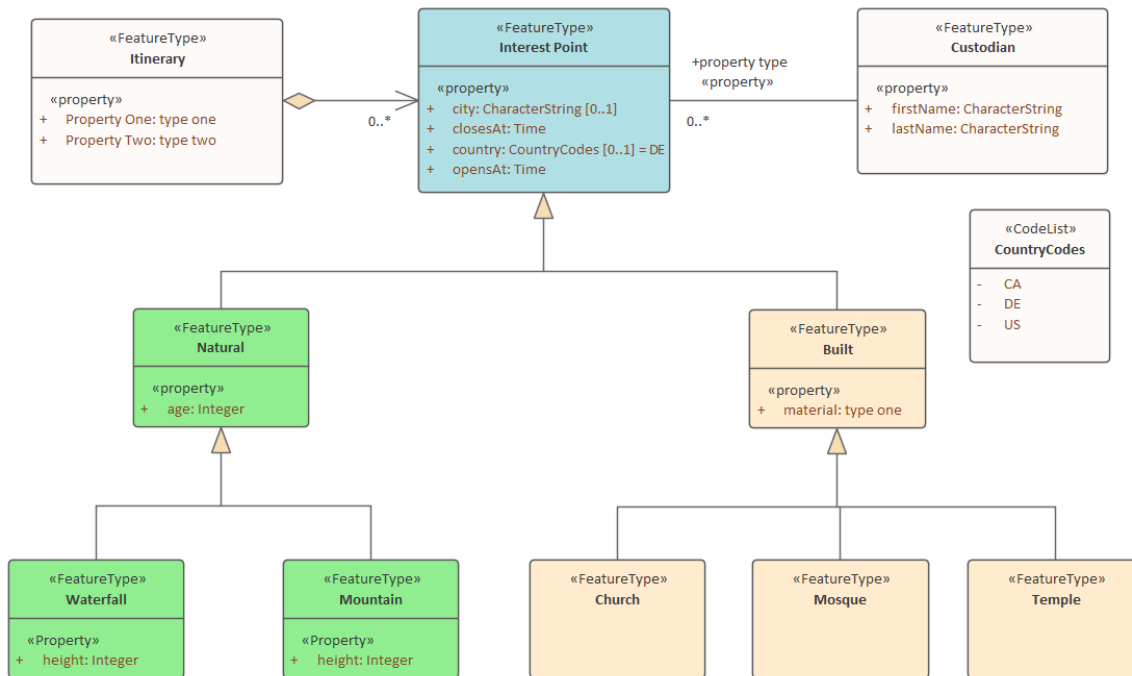
More Information

Edition Information

- Integration with GML is available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect

Example Diagram

Using the Geography Markup Language (GML) facility you can model organization or community based application schemas. The models can be used to show the relationship between Features Types that contain any number of Properties that qualify their characteristics. These can be based on defined Types, Data Types, Code Lists or Enumerations. You can collaborate with other geospatial colleagues or with people working in traditional systems implementations in disciplines that manage artifacts including: Strategies, Goals, Requirements, Data Models, Software Models, Deployment Descriptions and more.



Getting Started


Using the Geography Markup Language (GML) facility you can model organization- or community-based application schemas. The models can be used to show the relationship between Features and these elements in turn can contain any number of Properties that qualify their characteristics. These can be based on defined Types, DataTypes, CodeLists or Enumerations. You can collaborate with other geospatial colleagues or with people working in traditional system implementations in disciplines including Strategy, Goals, Requirements, Data Models, Software, Deployment and more.

The Geography Markup Language is the XML grammar defined by the Open Geospatial Consortium (OGC) and is used to express geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet.

In this topic you will learn how to work with the features that support the GML outlined in the sections.

Selecting the Perspective

Enterprise Architect partitions the tool's extensive features into Perspectives, which ensures that you can focus on a specific task and work with the tools you need without the distraction of other features. To work with the Geography Markup Language features you first need to select this Perspective:

 <perspective name> > Information Exchange > Geographic GML

Setting the Perspective ensures that the Geography Markup Language diagrams, their tool boxes and other features of the Perspective will be available by default.

Example Diagram

An example diagram provides a visual introduction to the topic and allows you to see some of the important elements and connectors that are created in specifying or describing a GML model including Features and Data Types.

Modeling with GML

This topic introduces the GML profile, which provides the diagrams, Toolbox pages and elements that you will work with, including Application Schemas, Leafs, Feature Types, Data Types Code Lists, Properties and more. You are able to select the GML perspective from the Geospatial group, which will set the tool up for modeling Application Schemas and Geographical features.

Generate GML Application Schema

In this topic you will learn how to generate a GML compliant Application Schema from your model of Geographic Features, Types, Code List and other elements. The resulting schema can be used with a range of other geographic information systems that know how to ingest the schema.

More Information

This section provides useful links to other topics and resources that you might find useful when working with the Geography Markup Language tool features.

Generate GML Application Schema

An Application schema is typically created by geographic informaticians in an organization, industry or community of modelers. The collaboration facilities in Enterprise Architect allow you and your colleagues to develop the model collaboratively. Once you have created a model using the GML diagram and Diagram Toolbox, and have created the required elements and connectors, you can automatically generate a GML-compliant Application Schema directly from the model. This Application Schema describes the Feature Types that a community of users is interested in. For example, an application for the travel industry might define Features including churches, lookouts, coastlines, camping grounds, monuments, places of interest, museums, and road exit ramps. These Feature Types in turn reference the primitive object types defined in the GML standard.


After you have created a GML model you can generate GML Application Schema from the model Package.

Access

Ribbon	Specialize > Technologies > GML > Generate GML Application Schema
Context Menu	Right-click on the Package Specialize GML Generate GML Application Schema

Generate GML Application Schema

Option	Action
Source Package	Displays the name of the selected «Application Schema» stereotyped Package.
Directory	Type in or browse for the directory into which the application schema file is to be generated.
GML Version	Click on the drop-down arrow and select the appropriate GML version for the GML Application Schema. Currently versions 3.3 and 3.2.1 are supported.
XML Encoding	Click on the drop-down arrow and select the XML encoding scheme to be used.
Ignore Element and Property Visibility	Click on this option to include all Classes, Attributes and Association End roles in the GML Application Schema, irrespective of their visibility (or Scope). Otherwise, only those with Public visibility will be considered for generation.
Stylesheet	(Optional) Select an XSL stylesheet to post-process the Schema before saving it to file. Import the required stylesheet into the project through the 'Resources' tab of the Browser window before selecting it in this field.
Application Schema(s)	Lists all the Application Schema Packages under the selected source Package. Click on a Package to display its details, as well as any of its child «Leaf» stereotyped Package details, in the Package Details panel. GML Application Schema will not be generated for any Package in the list that is unchecked.

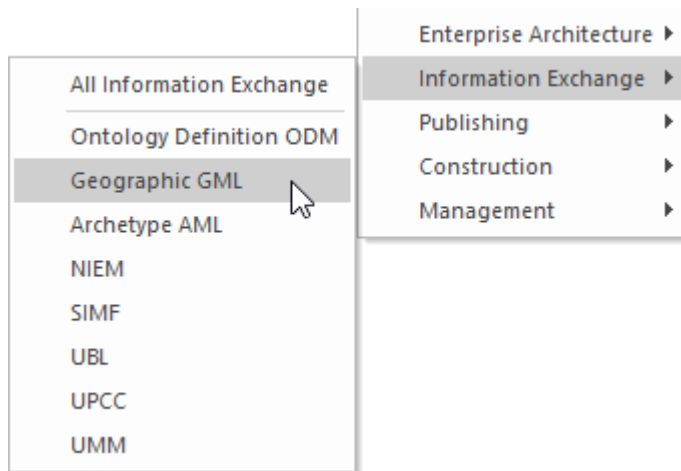
<p>Package Details</p>	<p>Against each of the child «Application Schema» and (optionally) «Leaf» stereotyped Packages (if any), enter a filename. To do this, double-click on the Package name in the list and, on the 'Package Details' dialog, review or complete these fields:</p> <ul style="list-style-type: none"> • 'Stereotype' - The stereotype of the selected Package • 'Filename' - (Optional for a «Leaf» stereotyped Package) Click on the  button and type the name of the .xsd file into which the GML Application Schema is to be generated; the filename must be in the same file path as selected in the 'Directory' field on the 'Generate GML Application Schema' dialog • 'Namespace' - (Not required for a «Leaf» stereotyped Package) Type in the namespace for the selected Package • 'Namespace Prefix' - (Not required for a «Leaf» stereotyped Package) Type the abbreviated value that represents the Namespace <p>Note that:</p> <ul style="list-style-type: none"> • The value of the 'Filename' field will be saved as the value of the tag xsdDocument on the Package • The value of the 'Namespace' field will be saved as the value of the tag targetNamespace on the Package • The value of the 'Namespace Prefix' field will be saved as the value of the tag xmlns on the Package
<p>Generate</p>	<p>Click on this button to generate the GML Application Schema for each of the checked Packages in the 'Package Details' panel.</p>
<p>View Schema</p>	<p>Click on this button to display the generated GML Application Schema for the selected Package in the 'Package Details' panel.</p>
<p>Close</p>	<p>Click on this button to close the 'Generate GML Application Schema' dialog.</p>
<p>Help</p>	<p>Click on this button to display this Help topic.</p>

Notes

- GML is available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect
- A «CodeList» stereotyped Class with the tag 'asDictionary' set to True will be exported as a separate XML file to the directory specified in the 'Directory' field, with the filename CodeList_<classname>.xml (where <classname> is the name of the «CodeList» stereotyped Class)
- Using the configurable file GMLStereotypes.xml in the Sparx Systems > EA > Config > GML folder, you can specify aliases for the standard GML stereotypes; the GML Application Schema Generator will also consider these aliases during Schema generation
- Association Class is specific to GML 3.3; when generating GML 3.2.1 Application Schema, the Association and Class of an Association Class are treated as separate entities

Modeling with GML

You can create GML models using the comprehensive diagramming and modeling facilities in Enterprise Architect. First you need to select the GML or Information Exchange Perspective. Perspectives are a useful focusing tool facility that ensure you remain focused and can concentrate on GML modeling.



Perspective menu - GML Perspective Selection

This activates the UML Profile for GML, allowing you to create models with elements and connectors that describe your organization or community domains.

Access

Ribbon	Specialize > Technologies > GML
Context Menu	Right-click on Package Specialize GML

Features

Feature	Detail
Profile Support	<p>You can develop GML constructs quickly and simply, through use of the built-in GML facilities provided in the form of:</p> <ul style="list-style-type: none"> • A GML diagram type, accessed through the 'New Diagram' dialog • GML pages in the Diagram Toolbox that map GML concepts to appropriately stereotyped UML elements • GML element and relationship entries in the 'Toolbox Shortcut Menu' and 'Quick Linker'
GML Toolbox Page	The GML Toolbox pages contain elements and connectors to model geographical features effectively.
UML Classes from ISO	(Optional) You can download the UML Classes implemented in ISO/TC 211 as an

	<p>XMI file, then import the XMI file into Enterprise Architect as a Package containing diagrams and standard UML Classes, which you can reuse in your model.</p> <ul style="list-style-type: none"> • Not all UML Classes implemented in ISO/TC 211 have a corresponding mapping in GML; the Classes that have a mapping (as specified in the GML 3.2.1 specification) are specified in the configurable file GMLClassMapping.xml in the 'Sparx Systems > EA > Config > GML' folder • The Namespace information for these Classes is specified in the configurable file GMLNamespaces.xml in the 'Sparx Systems > EA > Config > GML folder'
<p>GML Application Schema Generation</p>	<p>Any model you create using GML in Enterprise Architect can be exported as a GML Application Schema.</p> <p>Using the configurable file GMLSterotypes.xml in the 'Sparx Systems > EA > Config > GML' folder, you can specify aliases for the standard GML stereotypes. The GML Application Schema Generator will also consider these aliases during Schema generation.</p>


Notes

- GML is available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect

GML Toolbox Pages

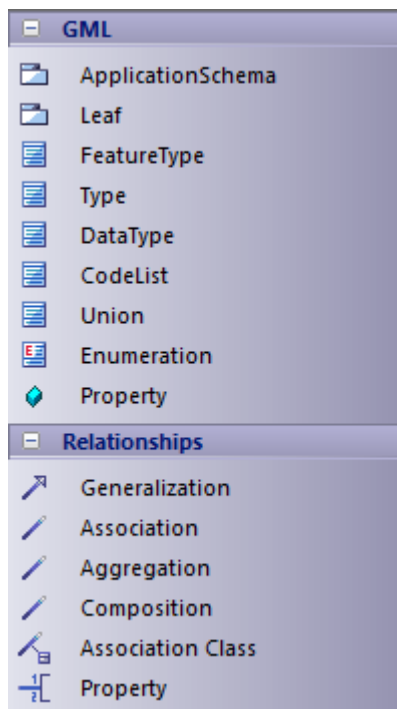
When you begin to model GML concepts and relationships, you can represent them on a GML diagram using the elements and connectors provided in the GML Toolbox pages.

Access

On the Diagram Toolbox, click on  to display the 'Find Toolbox Item' dialog and specify 'GML'.

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Ctrl+Shift+3

GML Toolbox pages



GML Element Toolbox

Item	Description
FeatureType	An extended UML Class that maps to an XSD Global element, whose content model is a globally scoped XML Schema type derived by direct/indirect extension of gml:AbstractFeatureType.
Type	An extended UML Class that maps to an XSD element, whose content model is a

	globally scoped XML Schema type derived by direct/indirect extension of gml:AbstractGMLType.
Data Type	An extended UML Class that maps to an XSD Global element, whose content model is a globally scoped XML Schema complexType.
Code List	An extended UML Class, that maps to: <ul style="list-style-type: none"> • A union of an enumeration and a Pattern, or • an external Dictionary
Union	An extended UML Class that maps to an XSD Choice group, whose members are GML objects or features, or objects corresponding to DataTypes.
Enumeration	An extended UML Class that is a restriction of XSD string with enumeration values.

GML Package Toolbox

Item	Description
ApplicationSchema	The GML workspace Package, which holds all the GML modeling elements. You can export the contents of this Package to produce the GML Application Schema.
Leaf	A stereotyped Package that can be created under an Application Schema. Leaf is a convenient Package for grouping elements within an Application Schema. If the value of the tag xsdDocument on this Package is empty, the contents of this Package will be exported as a part of the parent Package whose tag xsdDocument is not empty.

GML Relationship Toolbox

Item	Description
Generalization	Identifies an element as a specialized descendant of another element, containing additional properties. This relationship is allowed between: <ul style="list-style-type: none"> • Feature Types • DataTypes and Unions • Types <p>An element can have a maximum of one Generalization connector.</p>
Association	Identifies the connection between two elements. Only navigable Association Ends will be mapped to GML Application Schema.

Aggregation	An extended UML Association that shows that an element is composed of other elements.
Composition	An extended UML Association that is a stronger form of Aggregation, indicating ownership of the whole over its parts.
Association Class	Identifies the attributed connection between two elements. Only navigable Association Ends can be mapped to GML Application Schema.
Attribute	
Property	An extended UML Attribute, whose type is either a property type (if the type is a complex type) or a simple type.
Association Role	
Property	An extended UML Association Role, whose type is always a property type.

Notes

- In GML, Association Class is specific to GML 3.3; when generating GML 3.2.1 Application Schema, the Association and Class of an Association Class are treated as separate entities
- The values for tags on the «Application Schema» and «Leaf» stereotyped Packages can be entered using either:
 - The 'Tags' tab of the Properties window or
 - The 'Generate GML Application Schema' dialog

