



XML Schema of the Geodatabase

An ESRI® Technical Paper • June 2008

Copyright © 2008 ESRI
All rights reserved.
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts and Legal Services Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100 USA.

The information contained in this document is subject to change without notice.

ESRI, the ESRI globe logo, ArcGIS, ArcObjects, ArcSDE, ArcMap, Spatial Database Engine, SDE, www.esri.com, and @esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein may be trademarks or registered trademarks of their respective trademark owners.

XML Schema of the Geodatabase

An ESRI Technical Paper

Contents	Page
XML Schema	1
Domain, RangeDomain, and CodedValueDomain	3
Geodatabase XML Document Types	4
Workspace Document	4
RecordSet Document	6
Data Changes Document	7
XML Types in the Schema	9
Data Elements	9
DataElement	10
Arrays	11
DEDataset	11
DEGeoDataset	11
DEFeatureDataset	11
DETable, DEFeatureClass, and DERasterCatalog	12
Fields and Field	14
Subtype	15
DETopology	16
DEGeometricNetwork	16
DENetworkDataset	17
NetworkDirections	19
EdgeFeatureSources	19
NetworkSourceDirections	20
Shields	21
StreetNameFields	21
JunctionFeatureSources	22
SystemJunctionSources	23
TurnFeatureSources	24
EvaluatedNetworkAttributes	25
AttributeParameters	26
NetworkAttributes	27
NetworkAssignments	27
DECadastralFabric	28
DERepresentationClass	29

Contents	Page
DETerrain	30
TerrainDataSources	32
TerrainPyramidLevelZTo1	33
TerrainPyramidLevelWindowSize	34
Controllers and Memberships	34
DERelationshipClass	36
Index and Indexes	38
DERasterDataset and DERasterBand	39
DERasterCatalog	40
DEToolbox	41
RecordSet	41
RasterData	42
Geometry Types	44
Point and Multipoint	46
Envelope	46
Segment, Line, CircularArc, and EllipticArc	47
Path and Ring	49
Polycurve, Polyline, and Polygon	49
3D Geometry	50
XMLPersistedObject	52
XmlPropertySet	52
Appendixes	
Appendix A: Binary versus Normalized Serialization	53
Appendix B: Sample Instance Document	56

XML Schema of the Geodatabase

Since version 9, ArcGIS® has provided the ability to transfer geodatabase data using XML. With the ArcGIS XML format, you can import and export all items and data in a geodatabase such as domains, rules, feature datasets, and topologies. XML containing geodatabase data can be validated against the XML schema using automated utilities.

This document describes the XML schema for the geodatabase. Basic concepts of XML schema are discussed first, followed by the different XML document types that can be generated. This document also discusses some geodatabase XML types. The details of all geodatabase XML types can be found in the XML schema document GdbExchange.xsd.

XML Schema

There are multiple possible representations that could be used to describe a geodatabase in XML. You could, for example, represent a field using one XML attribute for each property of a geodatabase field:

```
<Field xsi:type="esri:Field">
  <Name>OBJECTID</Name>
  <Type>esriFieldTypeOID</Type>
  <IsNullable>false</IsNullable>
```

Alternatively, each property of the field could be represented using separate XML elements instead:

```
<Field xsi:type="esri:Field">
  <Name>OBJECTID</Name>
  <Type>esriFieldTypeOID</Type>
  <IsNullable>false</IsNullable>
  <Length>4</Length>
  <Precision>0</Precision>
  <Scale>0</Scale>
  <Required>true</Required>
  <Editable>false</Editable>
  <AliasName>OBJECTID</AliasName>
  <ModelName>OBJECTID</ModelName>
</Field>
```

XML schema can be used to specify the format that the XML document should follow. In the latter case above, the schema would specify that a field XML element should have a sequence of children elements, the first called Name (a string), the second called Type (an enumeration value), and so on. The geodatabase exchange format uses only XML elements (not attributes).

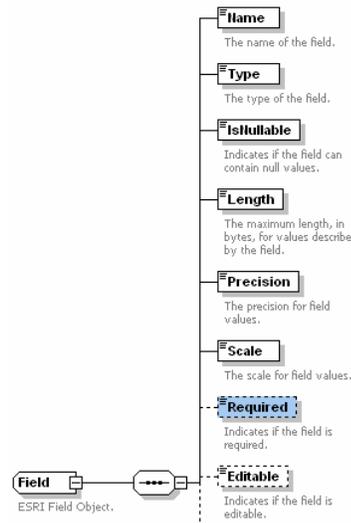
* A reduced set of properties for a field is shown here. See the XML schema for the full definition of the field XML type.

XML schema itself uses XML. Here is the schema definition for field:

```
<xs:complexType name="Field">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Type" type="esriFieldType"/>
    <xs:element name="IsNullabled" type="xs:boolean"/>
    <xs:element name="Length" type="esriFieldType"/>
    <xs:element name="Precision" type="esriFieldType"/>
    <xs:element name="Scale" type="esriFieldType"/>
    <xs:element name="Required" type="xs:boolean"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Notice that the Required element is optional because it can occur a minimum of zero times. Likewise, there is a maxOccurs attribute that can be used to specify the maximum number of times an element may appear. "Unbounded" is its value when there is no limit.

This document uses a graphic notation to represent the XML types. The following figure is the graphic that corresponds to the Field type. Notice how the optional element Required is represented.



An XML document can be validated against its XML schema. In the field case, the validating tool can compare the instance* document versus the schema and determine if it complies with the schema rules. A document without a Name element for a field would be considered schema invalid, but it could still be valid just from an XML standpoint.

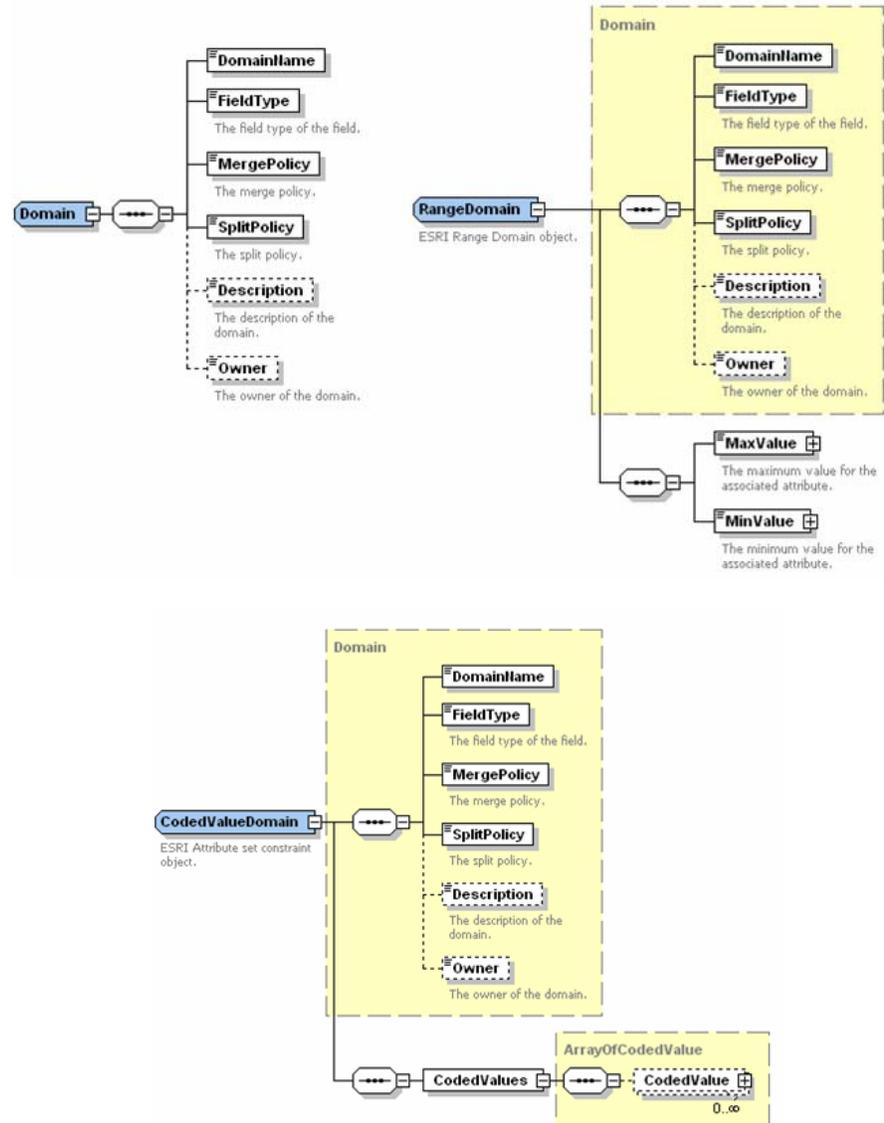
The XML schema snippet above defines an XML type for field. An element in an instance document has a type that can be explicitly stated by using the type attribute (as in xsi:type="esri:Field").

* XML documents that are compliant with an XML schema are commonly referred to as *instance* documents. Exporting one geodatabase to XML will generate an instance document.

J-9620

Domain, RangeDomain, and CodedValueDomain

Types can be derived from other types, most commonly by adding extra attributes to the base type. Domains are a good example of type inheritance. Domain has a name, type, and field type. A range domain is derived from it and adds two more elements, MinValue and MaxValue.



It doesn't make sense for an XML document to contain elements of the base type Domain. Rather, the document should have either range or coded-value domains. The Domain type can be marked as abstract to indicate this and prevent XML document authors from including domains in their documents, although such a document would be considered schema invalid. In general, instance documents use concrete derived types instead of their parent abstract type.

An XML schema can be seen as a collection of named types that can be used as templates when creating XML instance documents. However, two schemas could potentially define

two types with the same name. To resolve these conflicts, each schema is tied to a namespace, and all types defined in the schema belong to that namespace. For instance, an XML schema with types to describe petroleum data could have a type named "Field" to represent an oil field. However, this field would not be confused with the ArcGIS field type because they belong to different namespaces. The namespace for ArcGIS is <http://www.esri.com/schemas/ArcGIS/9.3>.

Field is an ArcObjects™ class that has a corresponding peer type in the XML schema. Mapping ArcObjects classes to XML types is common in ArcGIS and used extensively when exporting and importing geodatabase information to XML.

The Open Geospatial Consortium, Inc[®], has published a standard for the exchange of geographic data using XML called GML.* The format of the XML documents generated by ArcGIS is different from GML. The objective of the ArcGIS XML exchange tools is to represent the semantics of geodatabase constructs completely and unambiguously. It is possible to write utilities that perform the translation between the two formats.

The above discussion of XML schema is basic and intended to let you understand the basic constructs in the XML schema for XML exchange in ArcGIS. More information about this topic can be found in XML Schema Part 0: Primer.**

Geodatabase XML Document Types

There are three types of XML documents that can be created in ArcGIS: a workspace document, a record set document, and a data changes document. They are covered in detail below.

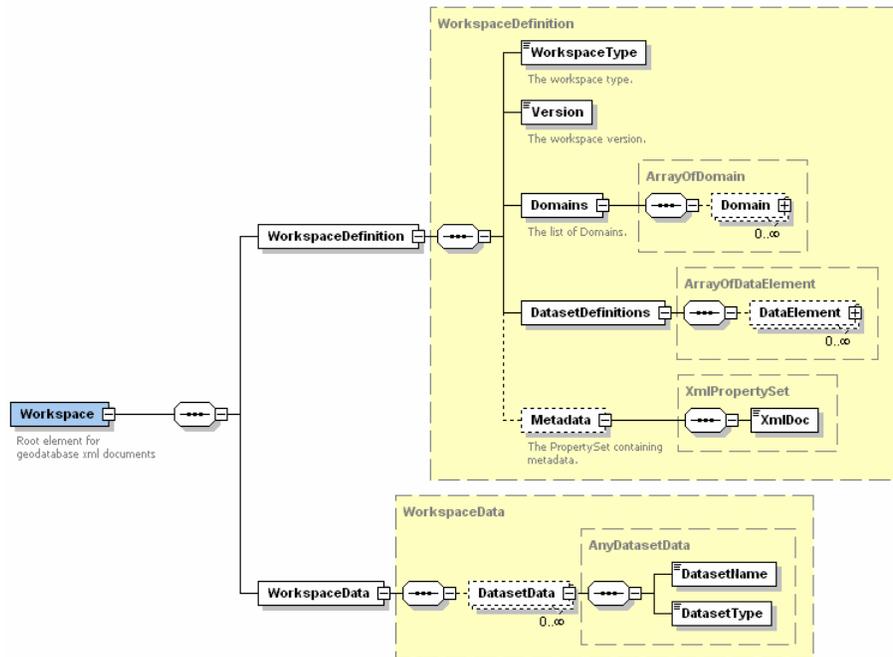
Workspace Document

Exporting a geodatabase to XML generates a workspace document. The XML element Workspace in the XML schema contains two child elements: WorkspaceDefinition and WorkspaceData.

* OpenGIS[®] Geography Markup Language (GML) Implementation Specification. January 29, 2003.

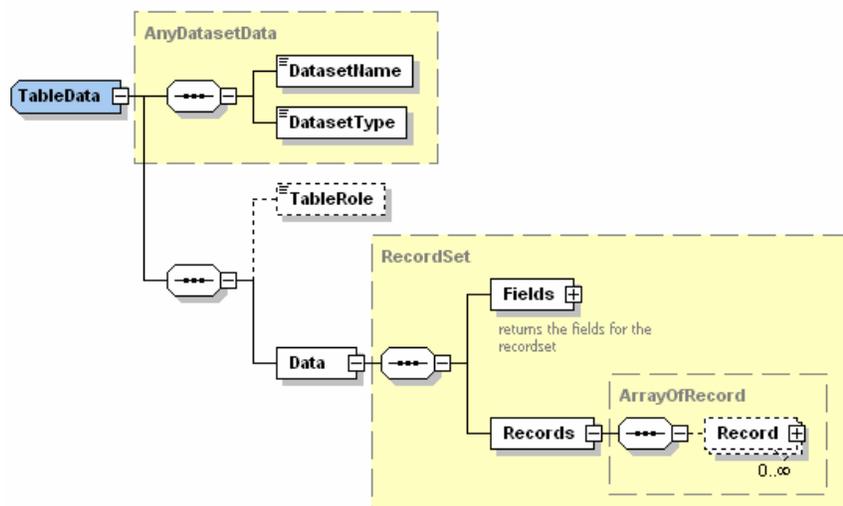
<http://www.opengis.org/techno/implementation.htm>

** XML Schema Part 0: Primer. W3C Recommendation. May 2, 2001. <http://www.w3.org/TR/xmlschema-0/>

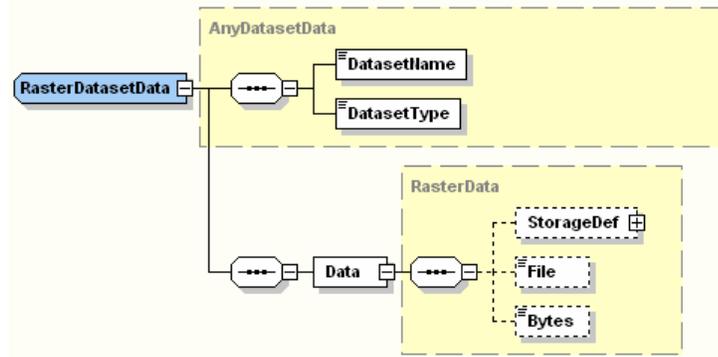


WorkspaceDefinition has the workspace type, the version being exported (if the exported database is based on ArcSDE®), an array of domains (notice there can be zero to many domains present), and an array of data elements. Domain is an abstract type. Instances of the derived types range and coded-value domains are inserted here instead. DataElement is also an abstract type whose derived types are used to describe the schema of geodatabase constructs such as tables.

The WorkspaceData element may have zero to many child DatasetData elements of type AnyDatasetData. Two XML types are derived from AnyDatasetData: TableData and RasterDatasetData.



TableData is derived from AnyDatasetData, adding an element named Data, of type RecordSet. In turn, record set contains the actual records from a table. RecordSet is discussed in detail later in this document.



RasterDatasetData is also derived from AnyDatasetData, adding the Data element (of type RasterData), which contains the file name of the raster file where the band has been exported. RasterData is explained later.

Using WorkspaceDefinition and WorkspaceData, workspace documents will contain the schema of a geodatabase and the data of its tables. Having the definition separated from the actual data offers a few benefits. A software agent can read only the definition section of the document to discover what it contains, and by making the data section optional, a workspace document can be used to transfer only schema information.

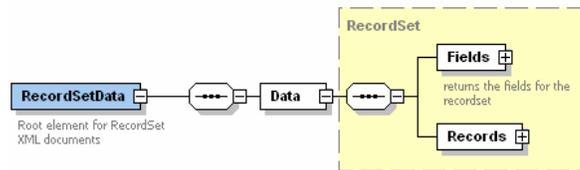
In the case of tables and feature classes, the name of the data element in the definition section corresponds to the name of the table data in the data section. Other datasets, such as topologies, geometric networks, or feature datasets, appear only in the definition section (no data is exported for them).

The behavior exported as part of the WorkspaceDefinition section includes all simple and custom features data, participation in networks and topologies, network connectivity and topology rules, simple and composite relationships, and any other information associated with the geodatabase datasets. Therefore, all behavior associated with the geodatabase is preserved and can be re-created when the XML document is imported.

RecordSet Document

The rows from a single feature class or table can be exported as simple features to a record set document. The features are exported as simple features, and no additional geodatabase-related information is written to the output file. Exporting to a record set document is analogous to exporting to a shapefile. For example, complex edge features in a geometric network will be exported as simple features, and relationships to features in other tables will not be exported.

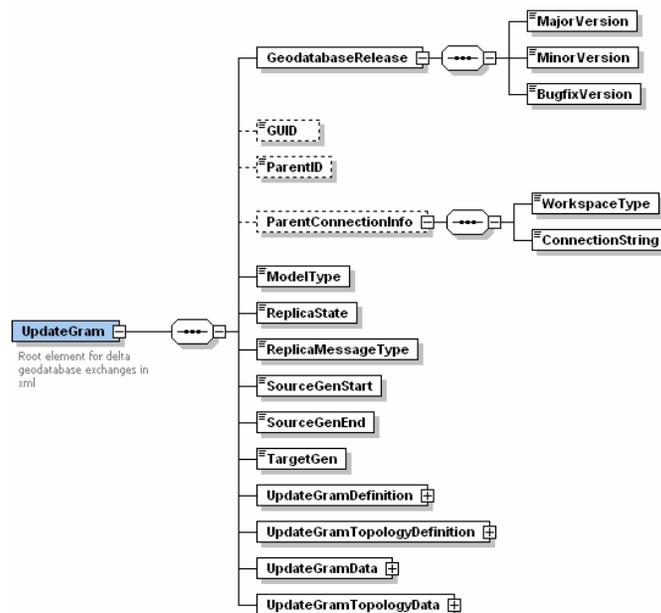
The root element of such a document is RecordSetData, consisting of just one mandatory child element called Data. The XML type of the data element is RecordSet. RecordSet contains a set of fields and the records of the exported table. RecordSet is covered in detail later.



Data Changes Document

The ArcGIS disconnected editing framework allows you to check out data from a database into a separate geodatabase, then edit the data without having a live connection to the parent database. Once editing is done, it is possible to export only the changes (not all the data) to an XML file. This file can be used to subsequently check in the changes to the parent database.

The root element of a data changes document is UpdateGram. GeodatabaseRelease contains version information about the parent database (Major Version, Minor Version, and Bugfix Version). The GUID subelement contains the unique identifier for the checkout database (assigned at checkout time). ParentID and ParentConnectionInfo contain additional information identifying the parent database. ModelType describes whether the checkout was done as simple features or with full geodatabase semantics. ReplicaState returns the replica state, which is the source of the data changes. The replica can be in a sending data state, sending acknowledgement state, or a waiting for data state. The ReplicaMessageType identifies the replica message type of the data changes object and returns a value of the esriReplicaMessageType enumeration. SourceGenStart, SourceGenEnd, and TargetGen return the start, end, and target generation of the source of the data changes message.



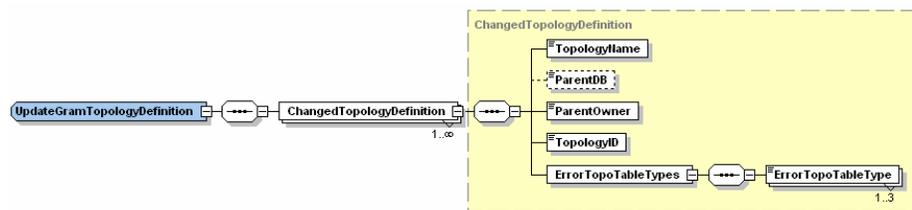
UpdateGramDefinition identifies the datasets that were modified in the checkout database. Conversely, the UpdateGramData element contains the actual data that was

changed. This separation allows software programs to find the data that was changed without needing to search the entire geodatabase for changed features.

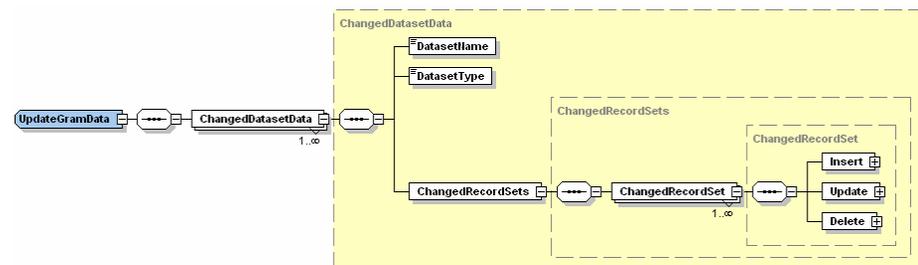
UpdateGramDefinition is made of one or more ChangedDatasetDefinition elements, each containing the elements that identify a changed dataset (name and type of the dataset, the parent database, and the owner).



UpdateGramTopologyDefinition is made of one or more ChangedTopologyDefinition elements, each containing the elements that identify a changed topology (name, parent and owner of the topology, the ID and the error types).



The next element of UpdateGram is UpdateGramData. As its name implies, it registers the data that changed in the checkout database. Since many datasets could have been changed, UpdateGramData can have one or many ChangedDatasetData elements, one for each changed dataset.

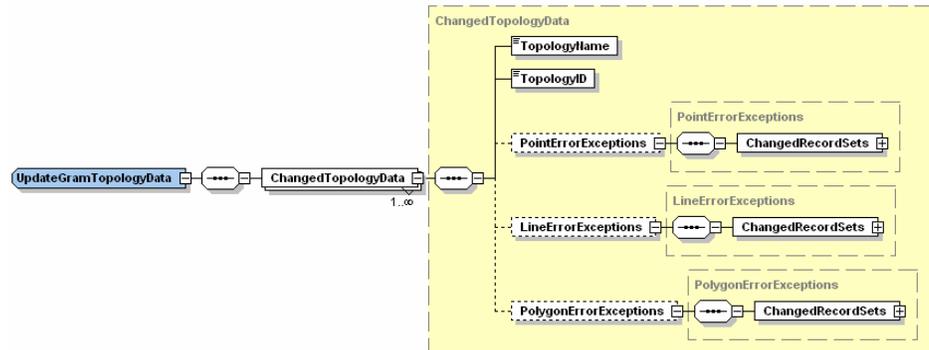


The purpose of the ChangedDatasetData elements is to record what changes were made to the original features in a dataset. The possible changes are the creation of new features (inserts) and updates to or deletion of existing features. To record these actions, each ChangedDatasetData element contains RecordSets of the features inserted, updated, or deleted.

Delete record set only has one field, the ObjectID or GlobalID, because that is the information required to delete the features from the parent database when the check-in is performed.

Finally, DatasetName and DatasetType appear in both UpdateGramDefinition and UpdateGramData. Each ChangedDatasetDefinition element that appears in the UpdateGramDefinition section must have a corresponding ChangedDatasetData element in the UpdateGramData section.

The last element of UpdateGram is UpdateGramTopologyData and is made of one or more ChangedTopologyData elements. The purpose of the ChangedTopologyData elements is to record what changes were made to a topology. The possible changes are the point error exceptions, line error exceptions, and polygon error exceptions. To record these actions, each ChangedTopologyData element contains RecordSets of the features inserted, updated, or deleted.



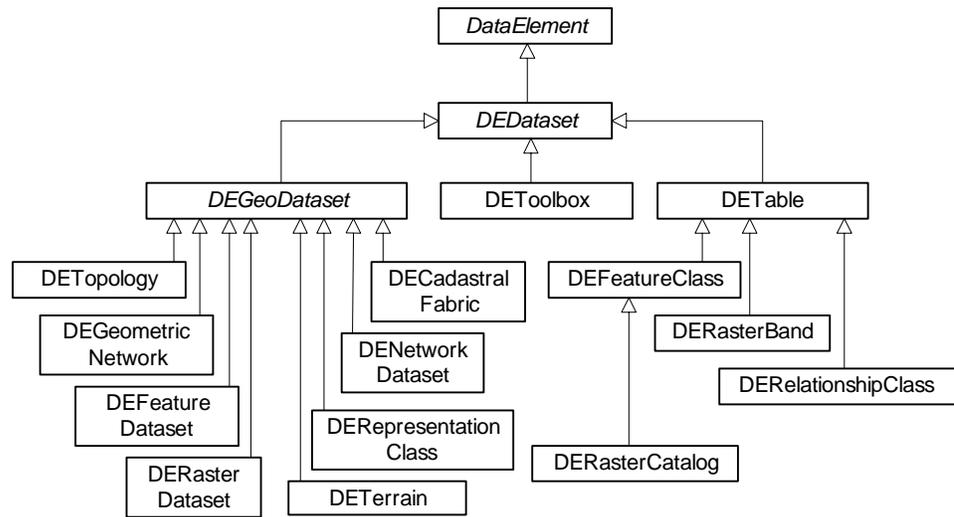
XML Types in the Schema

Data Elements

This section describes in detail some XML types that are used to create the three kinds of XML documents described in the preceding section.

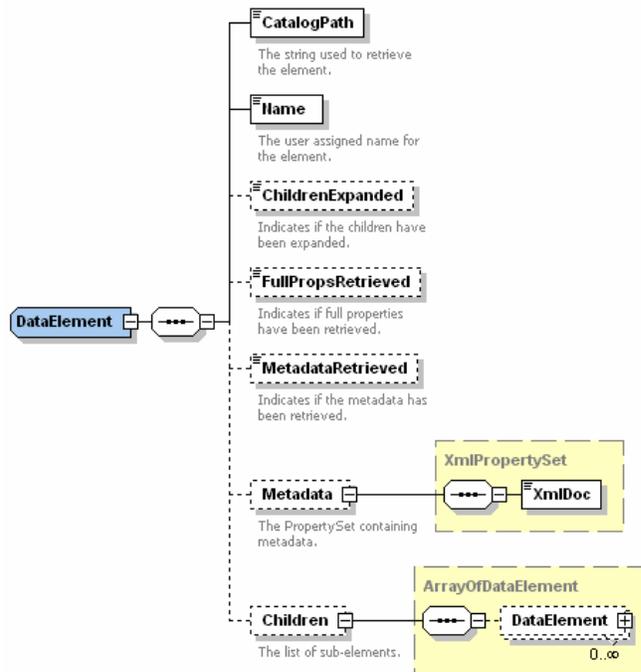
Data elements are a set of ArcObjects classes added in ArcGIS 9. These objects describe all aspects of a dataset in a geodatabase. For example, the data element for a feature class captures the name, feature type, fields, indexes, subtypes, and other characteristics of a feature class.

There are data elements for most geodatabase items. All information that relates to database schema is exported to XML using these data elements. Shown below is part of the hierarchy of XML types that correspond to the data element ArcObjects. The arrows point to the parent types, and the names in *italics* denote abstract types.



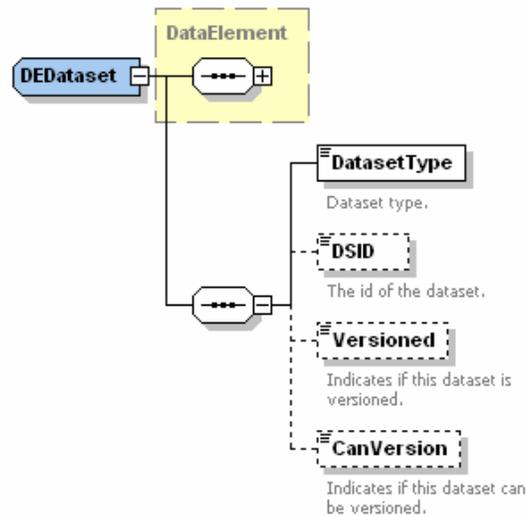
DataElement

DataElement is an abstract type used as the base type for all data elements; therefore, all data elements inherit from DataElement. The value of the CatalogPath element is a string that describes the location of the dataset. The Name element contains the name of the data element. The Children element is an ArrayOfDataElement type, which allows some data elements to have child data elements (e.g., FeatureDataset). Last, if metadata is exported, the element MetadataRetrieved will be present and have a value of True, and the Metadata element will contain the actual metadata XML document.

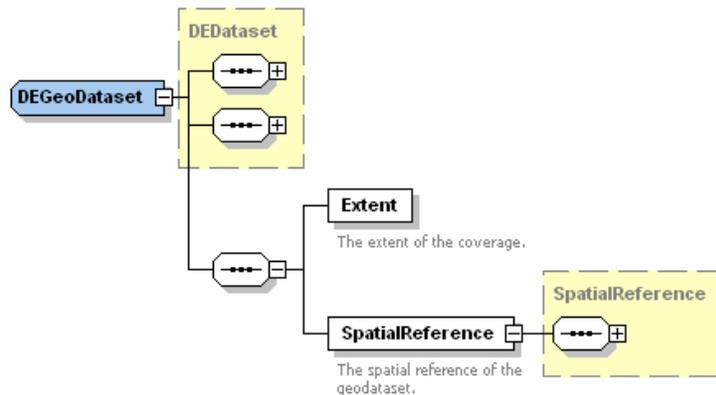


Arrays Arrays are used in many other data elements to convey "has a" associations between geodatabase items. A table data element has an array of subtypes, geometric networks and topologies have an array of rules, and so on. The names of array types are usually made of the *ArrayOf* prefix followed by the name of the type of elements it is made of (e.g., *ArrayOfDataElement*).

DEDataset DEDataset is used as an abstract base type to represent characteristics common to all datasets. DEDataset extends DataElement, adding elements to hold the ID of the dataset, its type, whether it can be versioned, and if so, if it has been versioned.

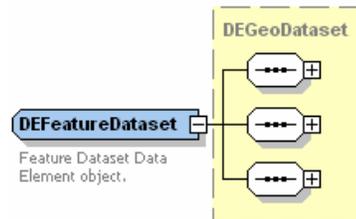


DEGeoDataset DEGeoDataset is another abstract base type used to group characteristics common to datasets that have a geographic extent such as feature datasets or topologies. DEGeoDataset extends DEDataset, adding the Extent and SpatialReference elements.



DEFeatureDataset DEFeatureDataset is derived from DEGeoDataset but doesn't add any more elements, which means the diagram above also represents a feature dataset data element. DEFeatureDataset is ultimately derived from DataElement. This means DEFeatureDataset has a Children element that can be used to contain an array of other

data elements. Since geometric networks, topologies, feature classes, and relationship classes are also derived from DataElement, they can appear in the Children array of a DEFeatureDataset element.



***DETable,
DEFeatureClass, and
DERasterCatalog***

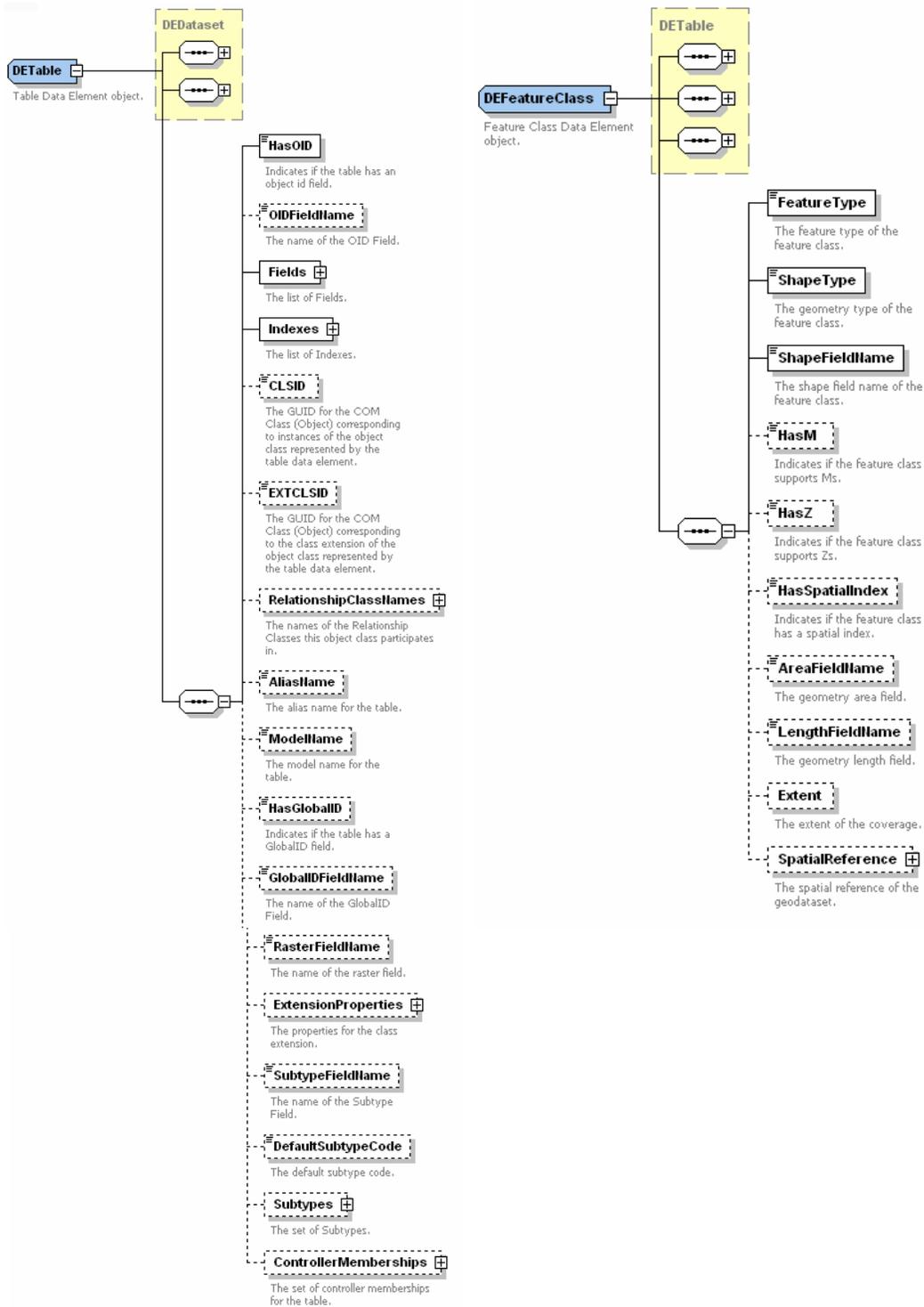
DETable is derived from DEDataset, adding a number of extra elements to the base type including the name of the ObjectID field; arrays for fields, indexes, and subtypes; and the names of the relationship classes the table is involved in. In addition, the DETable type has optional elements to describe the subtypes of the table if they are defined (SubtypeFieldName, DefaultSubtypeCode, and the array of Subtypes). The controller memberships array is explained later.

The CLSID element contains the identifier of the behavior class associated with the object class. If the object class has a feature class extension, the EXTCLSID and ClassExtensionProperties elements describe the feature class extension behavior class and its properties, respectively.

DEFeatureClass has elements to describe the feature type, shape (geometry) type, name of the field that contains the geometries, extent, and spatial reference of the feature class. Note that because it is derived from DETable, DEFeatureClass can also reference the relationship classes it is involved in or have subtypes.*

* Annotation and dimension feature classes are treated as normal feature classes by the geodatabase XML import/export utilities. No separate data element or XML type exists for these datasets.

J-9620



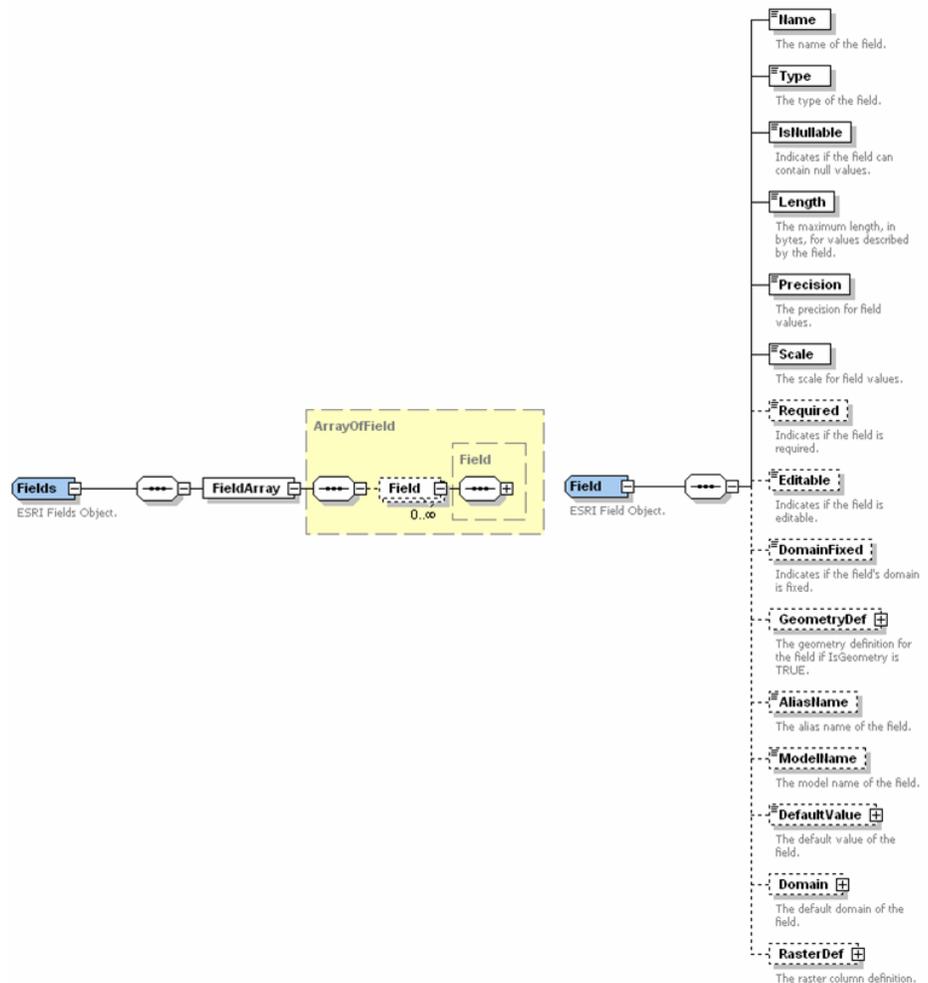
DERasterCatalog XML type is derived from DEFeatureClass but doesn't add any extra elements. The RasterFieldName element of DETable should have the name of the field that contains the rasters. The actual rasters in the catalog are exported to XML as values of a record in the record set of the raster catalog.

Fields and Field

Tables, feature classes, relationship classes, and other datasets have collections of fields and indexes. The Field XML type defines elements to describe characteristics of a field such as name, type, whether it allows null values, alias and model names, default value, and domain.

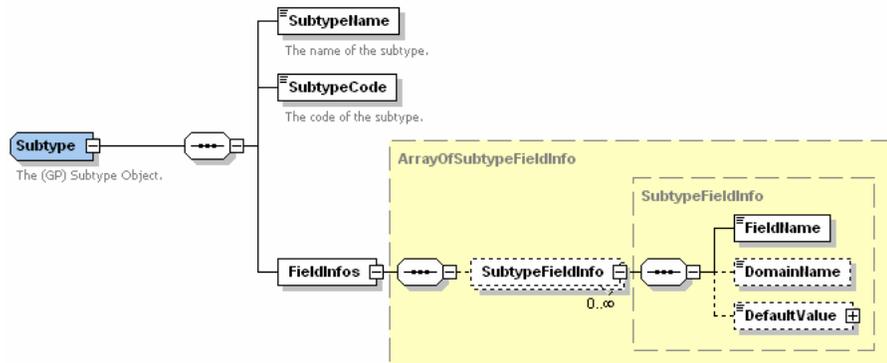
If the field type is geometry, then the element GeometryDef should be present and describe the average number of points, geometry type, grid sizes, spatial reference, and whether vertices of the feature will contain measures or elevations. If the field type is raster, then the element RasterDef should be present and describe the default spatial reference for the rasters stored in the table.

As its name indicates, the FieldArray XML type is an array of fields that may have zero to any number of fields. The Field XML type has one element whose name is FieldArray (of type FieldArray).



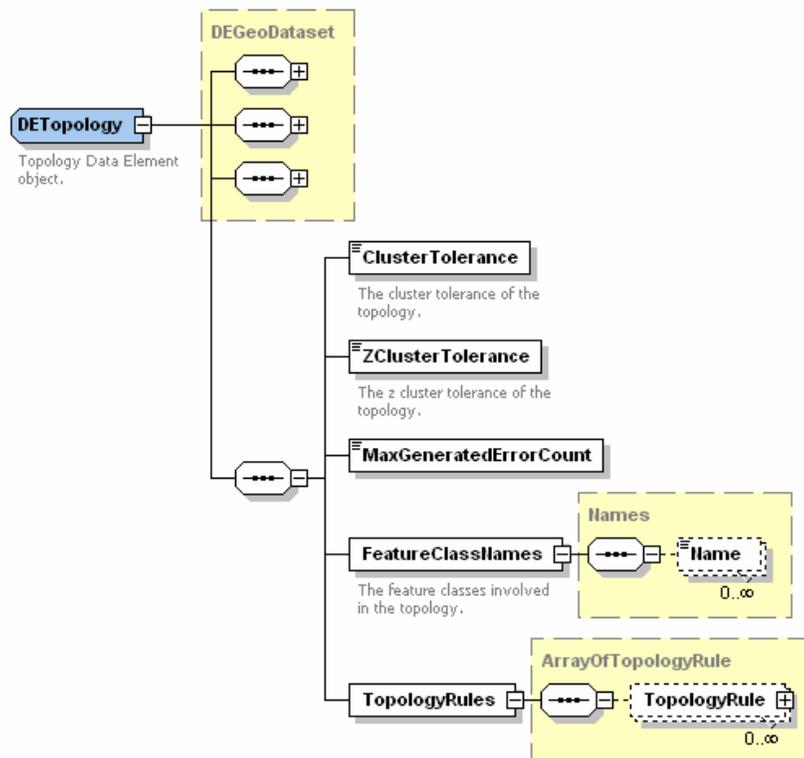
Subtype The array of subtypes in the table data element may have zero to many elements of type Subtype. This type has elements for the name and code of the subtype and an array of SubtypeFieldInfo elements.

In turn, the SubtypeFieldInfo type has elements for FieldName, DefaultValue, and DomainName. As an example, a SmallTransformer subtype could state that the default value for the field named Capacity is 100 but not set a domain for the field. Other subtypes could do the opposite, setting the domain but not the default value. For this reason, the elements DefaultValue and DomainName are optional.



DETopology

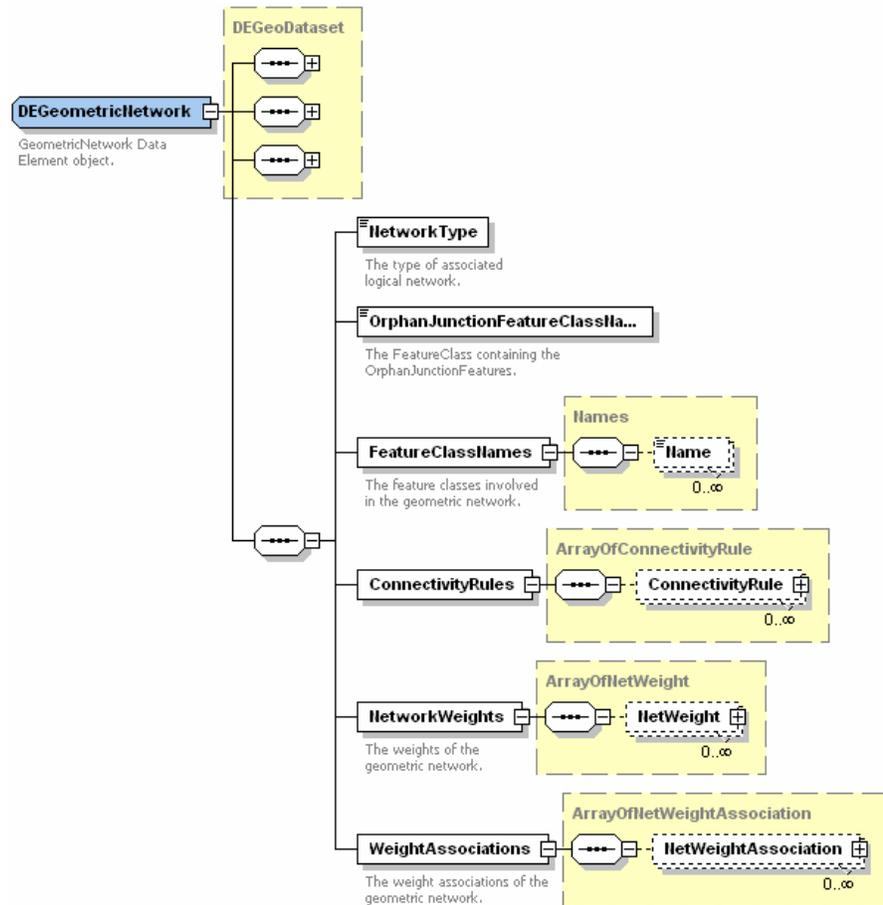
DETopology XML type is derived from DEGeoDataset. It has elements for the cluster tolerance, the maximum number of errors to generate the names of the feature classes involved in the topology, and an array of topology rules. The topology rule XML type (not shown) has elements to identify the type of topology rule and the classes and subtypes affected by the rule.



DEGeometricNetwork

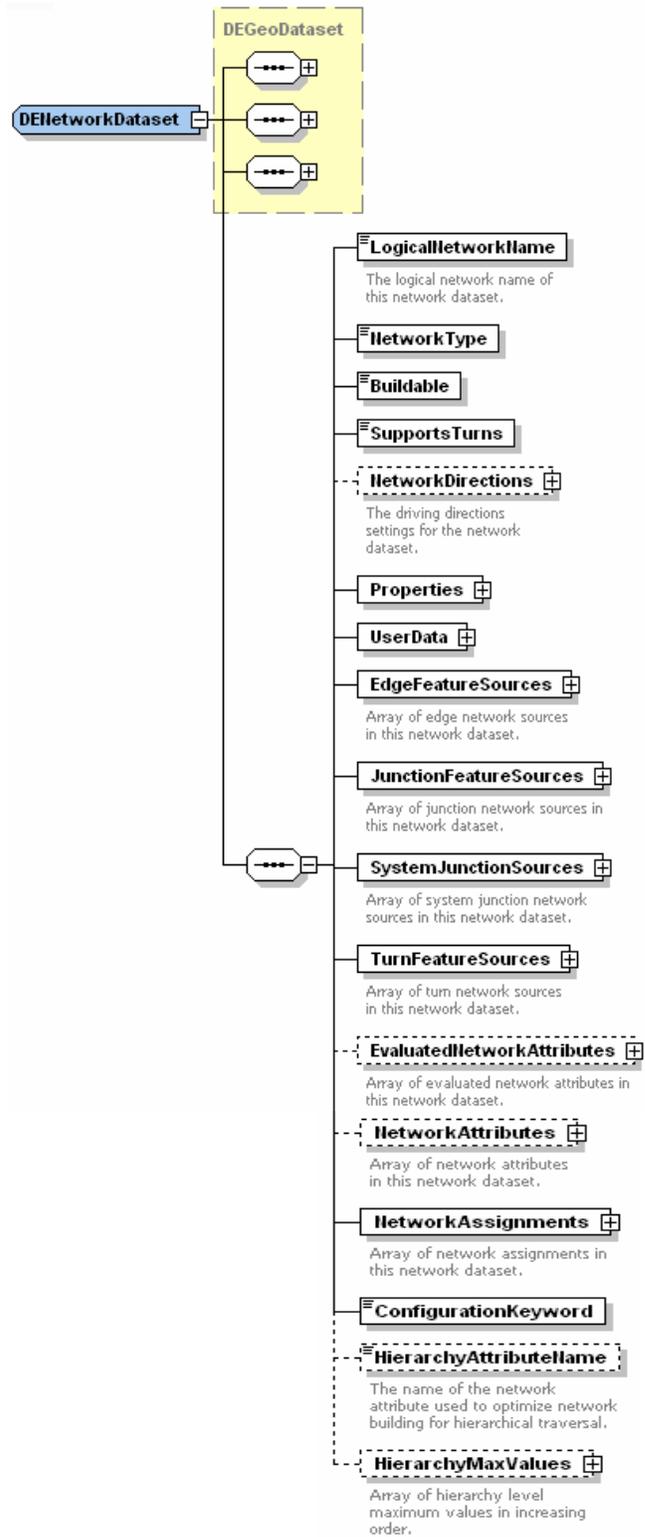
DEGeometricNetwork XML type is derived from DEGeoDataset type. Besides network type and the names of the feature classes in the network, the geometric network data element has the name of the orphan feature class, network weights and weight associations, and connectivity rules.

J-9620



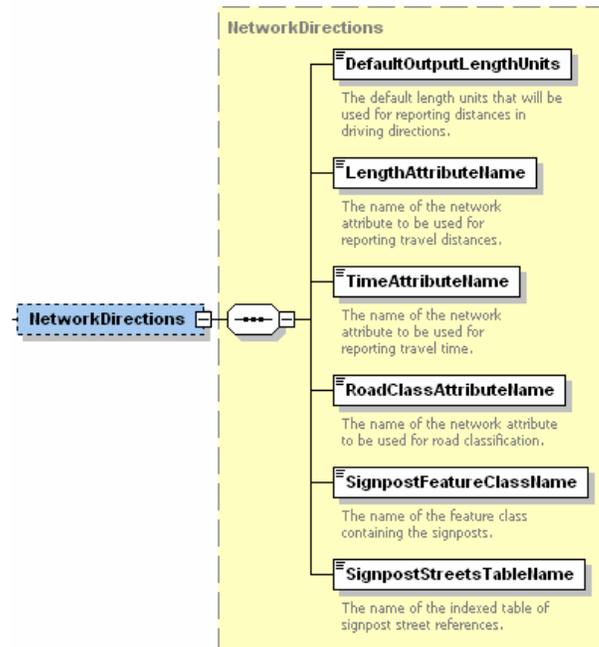
DENetworkDataset

DENetworkDataset XML type is derived from DEGeoDataset. It has elements for the name of the logical network, the type of network dataset, whether or not the network dataset is buildable, whether or not the network dataset supports turns, the settings for generating driving directions, network properties, and user-defined data. It also has array elements for various sources, attributes, and assignments in the network dataset. For network datasets in an ArcSDE geodatabase, there is also a configuration keyword.



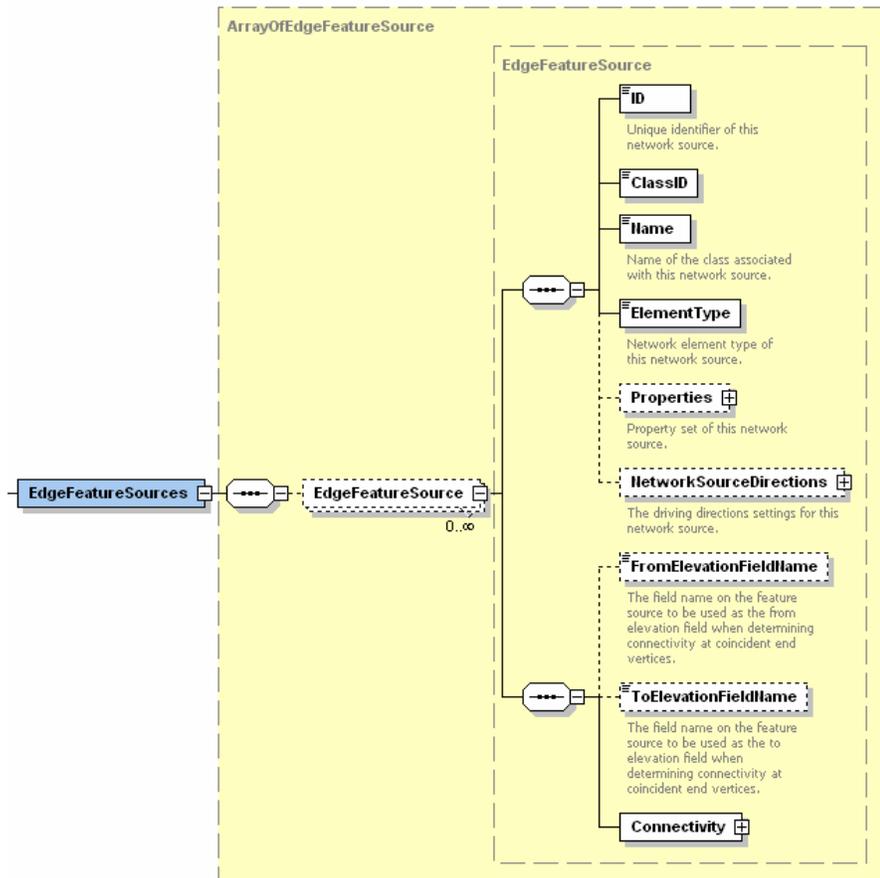
NetworkDirections

NetworkDirections defines the general settings to be used when generating driving directions for this network dataset. The NetworkDirections XML type consists of the default units to use when generating the driving directions text; the names of the network attributes used to determine length, travel time, and road classification; and the names of the signpost feature class and signpost streets table.

***EdgeFeatureSources***

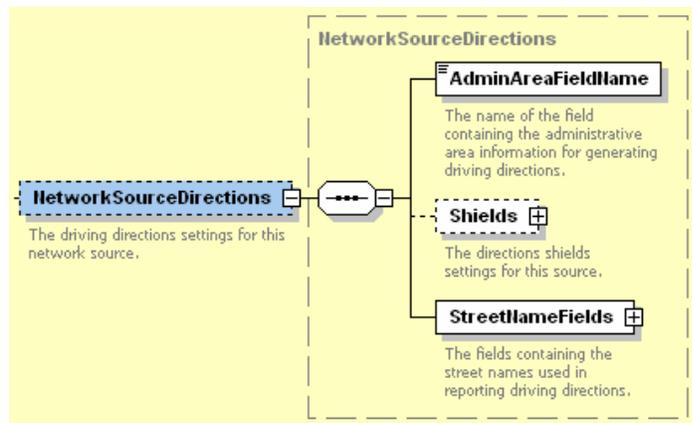
Edge feature sources are sources in the network dataset in which edge elements are generated from source line features.

The EdgeFeatureSource XML type is derived from the NetworkSource abstract XML type, which consists of the unique ID of the network source, the class ID and name of the source feature class, the type of elements generated by this source, and any properties and source-specific directions settings on the network source. In addition, the EdgeFeatureSource XML type also consists of fields indicating the from and to endpoint elevation values and connectivity group and connectivity policy information.



NetworkSource Directions

NetworkSourceDirections defines the source-specific settings for generating driving directions. The NetworkSourceDirections XML type consists of the field name for the administrative area to which a street belongs, Shields fields used in describing highways, and StreetNameFields used to describe each street.

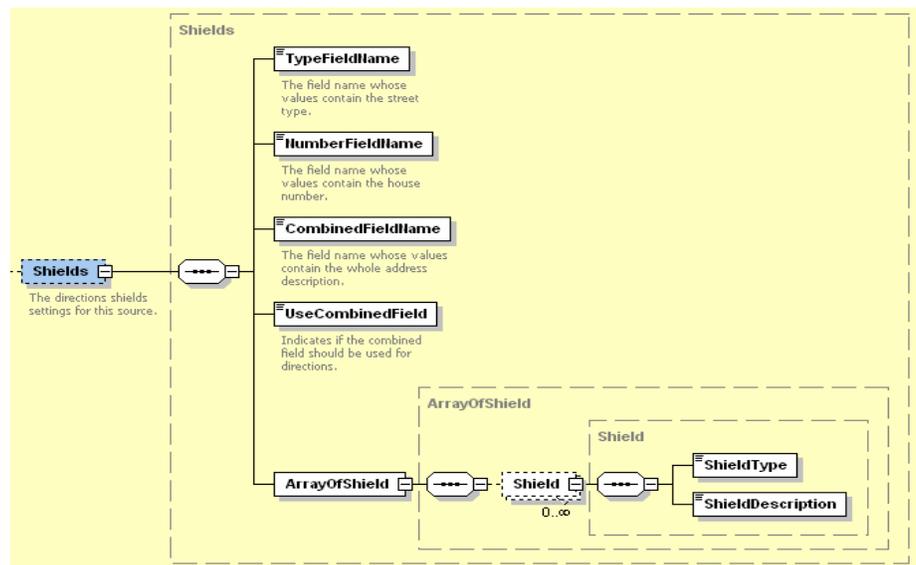


Shields

Shields indicates the highway number and highway type of federally, locally, or state-designated highways. For example, for highway I-10, I is the type of highway, and 10 is the highway number. This designation is used in the driving directions text.

The Shields XML type consists of the field names for the highway type and highway number or the name of the single combined field used to indicate both highway type and highway number. It also consists of an element to indicate whether to use the combined field instead of the individual highway type and highway number fields when generating the driving directions text.

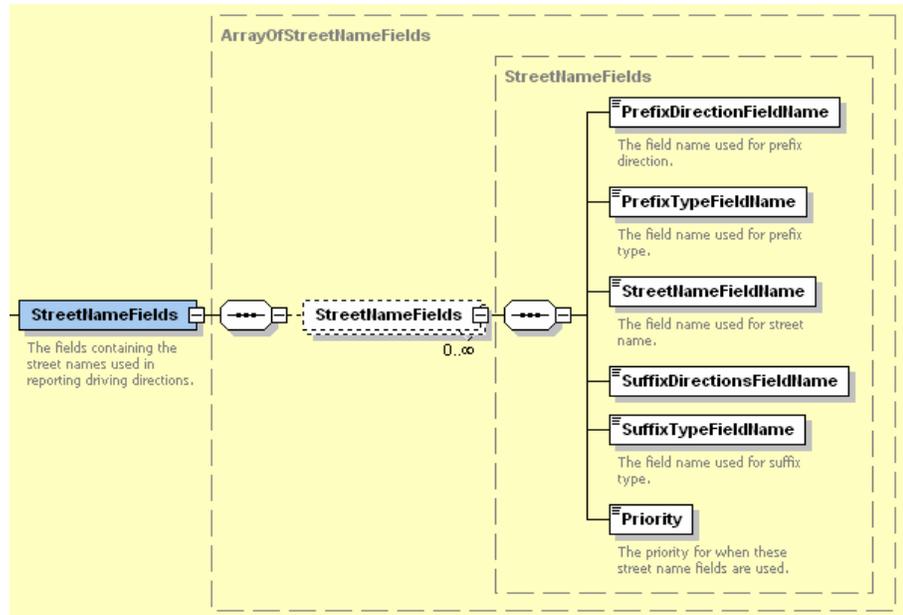
The Shields XML type also contains an array of Shield XML types, which are used to describe the various highway types. In the example of I-10 above, the I highway type can be given a shield description of Interstate so that it is reported in the driving directions text as Interstate 10 instead of I-10.



StreetNameFields

StreetNameFields indicates which fields contain the name of the street. These fields are combined to form the street name that is used in the driving directions text.

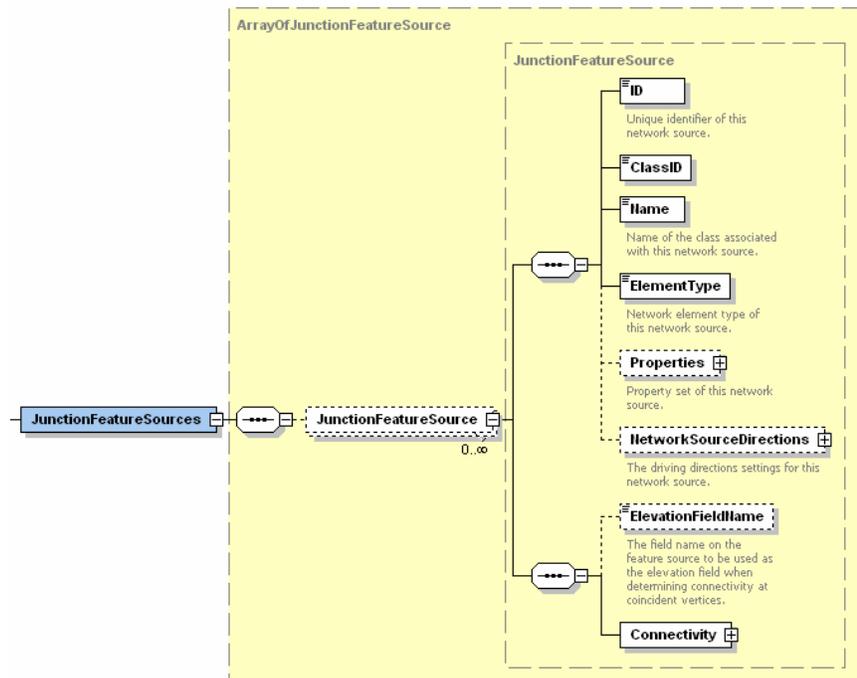
The StreetNameFields XML type consists of the names of the fields for the directional prefix of the street name, the street type prefix of the street name, the base name of the street name, the directional suffix of the street name, and the street type suffix of the street name. It also contains an element for reporting the priority level of this street name.



JunctionFeature Sources

Junction feature sources are sources in the network dataset in which junction elements are generated from source point features.

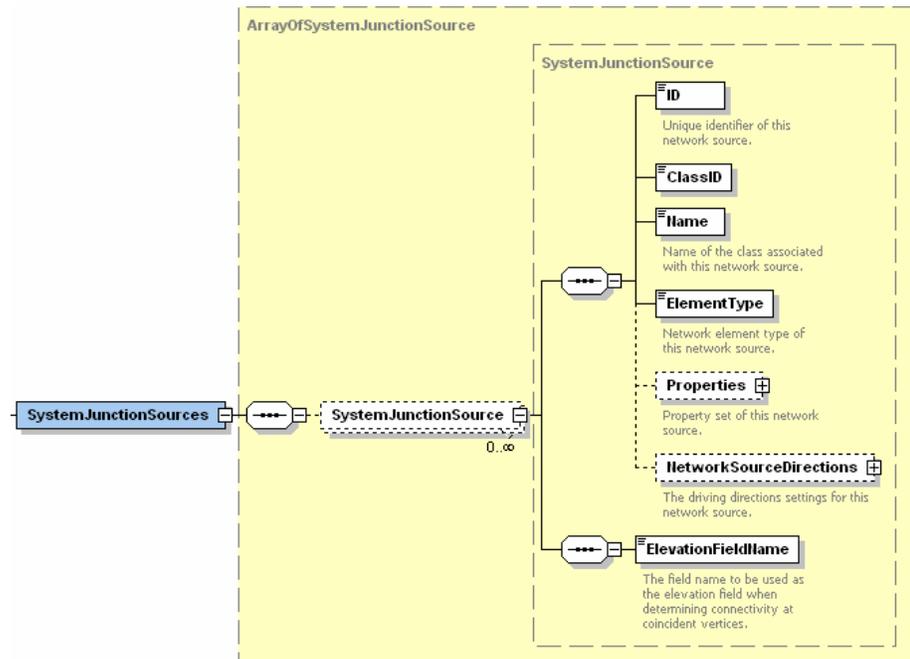
The JunctionFeatureSource XML type is derived from the NetworkSource abstract XML type, which consists of the unique ID of the network source, the class ID and name of the source feature class, the type of elements generated by this source, and any properties and source-specific directions settings on the network source. In addition, the JunctionFeatureSource XML type also consists of the field name indicating the elevation values of each junction feature, connectivity group, and connectivity policy information.



SystemJunction Sources

The system junction source is a system-managed feature class with point geometry that was created when the network dataset was created. This feature class contains point features for every junction element that was created where there was no corresponding feature in any of the junction feature sources in the network dataset.

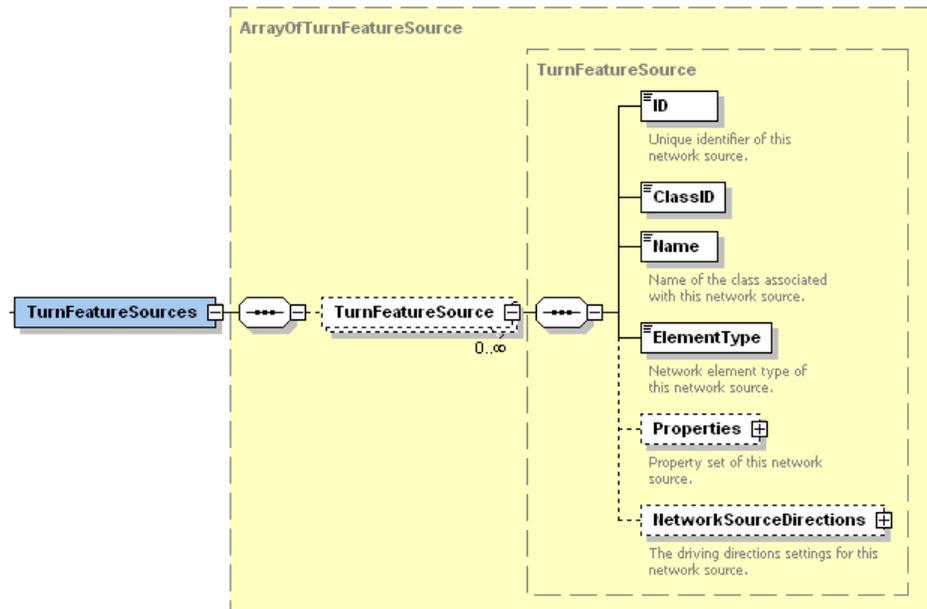
The SystemJunctionSource XML type is derived from the NetworkSource abstract XML type, which consists of the unique ID of the network source, the class ID and name of the source feature class, the type of elements generated by this source, and any properties and source-specific directions settings on the network source. In addition, the SystemJunctionSource XML type also consists of a field name indicating the elevation values of each system junction feature.



TurnFeatureSources

Turn feature sources are sources in the network dataset in which turn elements are generated from source turn features.

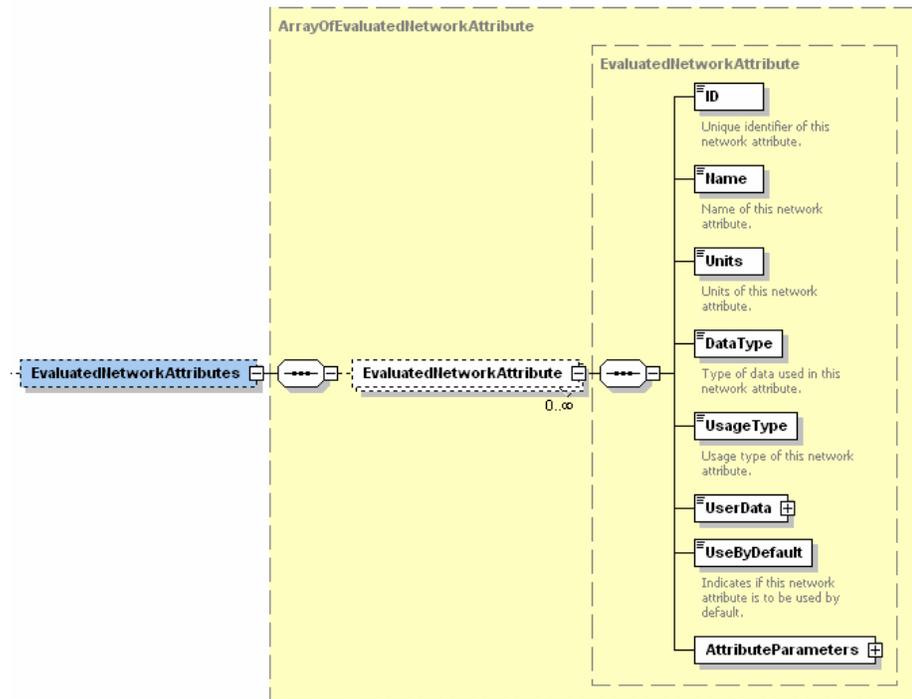
The TurnFeatureSource XML type is derived from the NetworkSource abstract XML type, which consists of the unique ID of the network source, the class ID and name of the source feature class, the type of elements generated by this source, and any properties and source-specific directions settings on the network source. The TurnFeatureSource XML type does not contain any additional elements beyond the NetworkSource abstract XML type.



EvaluatedNetwork Attributes

A network attribute consists of data values stored on the elements in a network dataset for use by the network analysis solvers. Network attributes define impedance values, such as travel time or distance, for each network element. An evaluated network attribute is a network attribute whose data values are calculated by network evaluators.

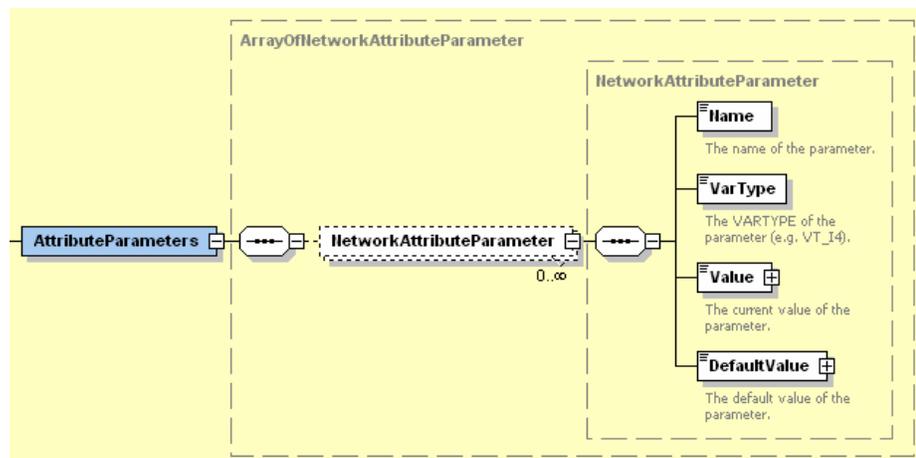
In addition to the unique ID and name of the evaluated network attribute, the **EvaluatedNetworkAttribute** XML type consists of the units of distance or time that the attribute is in, the type of data in which the network attribute values are stored, how the network attribute will be used during network analysis, any user-defined data associated with the network attribute, whether the network attribute will be used by the solver by default, and any attribute parameters associated with the network attribute.



AttributeParameters

Attribute parameters on a network attribute are changeable data values used by network evaluators when computing attribute values. An example of an attribute parameter is the weight of a vehicle. Whether or not a bridge is traversable depends on the current weight of the vehicle.

In addition to the name of the attribute parameter, the `NetworkAttributeParameter` XML type consists of the data type of the parameter, the current value of the parameter to be used during analysis, and the default value of the parameter to be initially presented to the user.

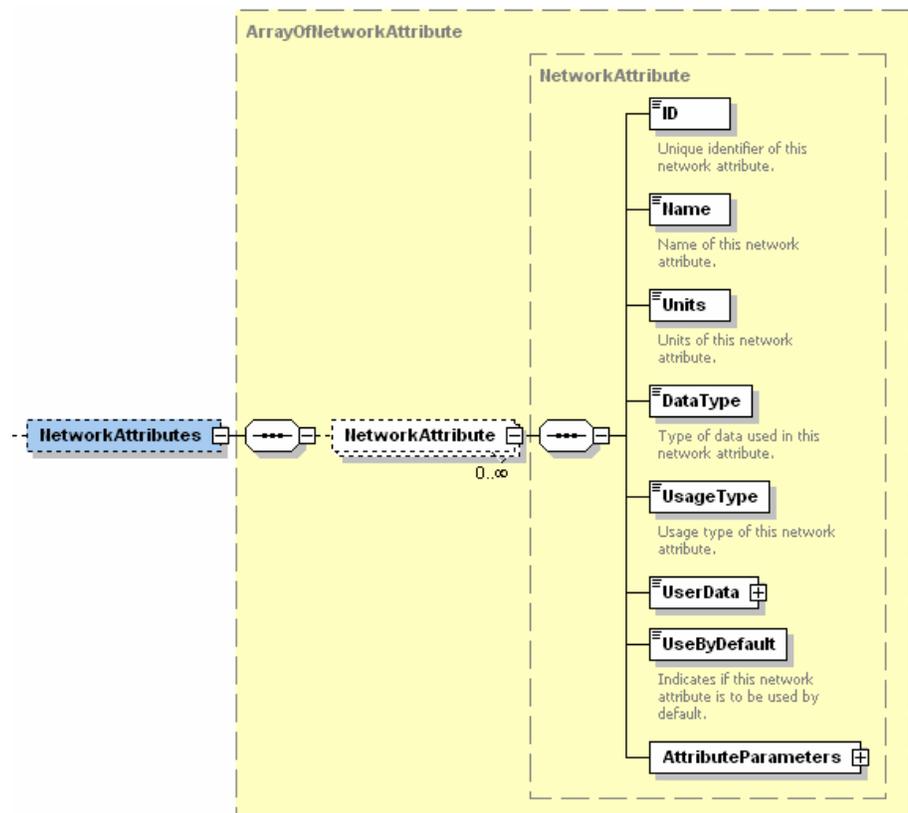


NetworkAttributes

A network attribute consists of data values stored on the elements in a network dataset for use by the network analysis solvers. Network attributes define impedance values, such as travel time or distance, for each network element.

In addition to the unique ID and name of the network attribute, the NetworkAttribute XML type consists of the units of distance or time that the attribute is in, the type of data in which the network attribute values are stored, how the network attribute will be used during network analysis, any user-defined data associated with the network attribute, whether the network attribute will be used by the solver by default, and any attribute parameters associated with the network attribute.

Unlike the EvaluatedNetworkAttribute XML type, the NetworkAttribute XML type represents network attributes that do not have any evaluators assigned to them.

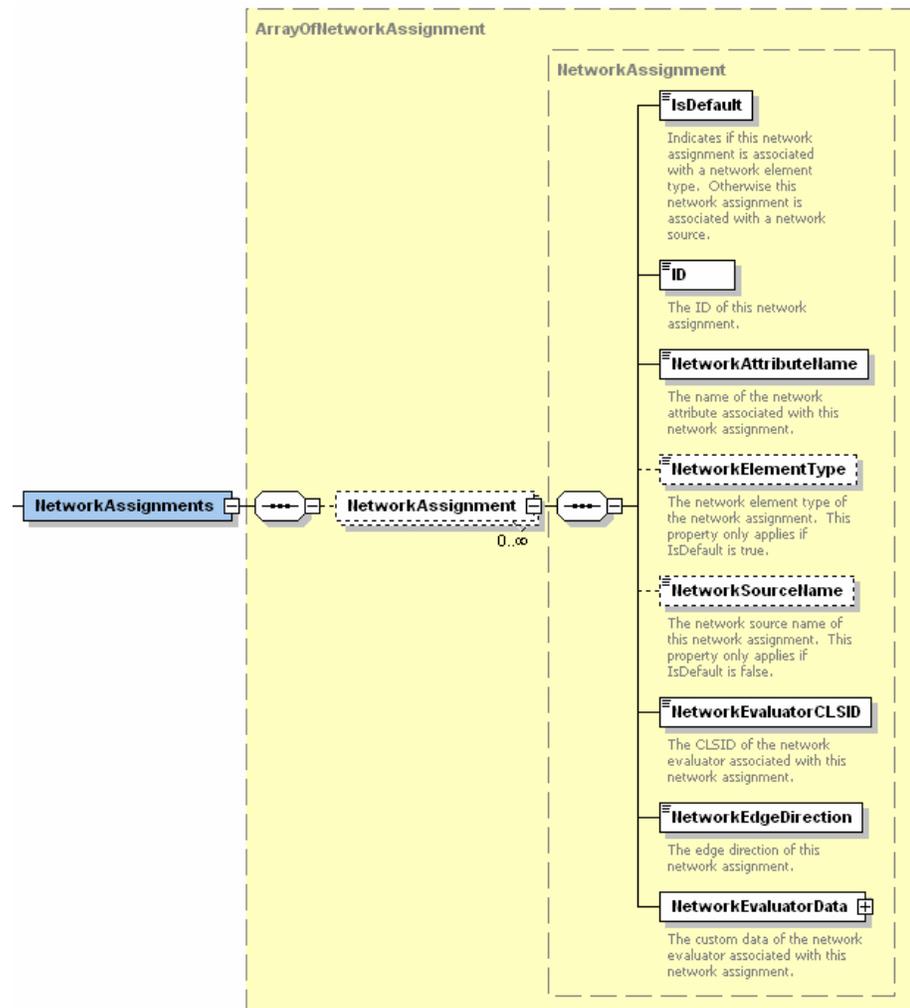


NetworkAssignments

A network assignment is an assignment of a network evaluator to an evaluated network attribute. An assignment can be either a default assignment that applies to all network elements of a given type or a network source-specific assignment that applies to all network elements generated from a given source.

The NetworkAssignment XML type consists of an element that indicates whether or not the assignment is a default assignment, the unique ID value of the assignment, the name of the evaluated network attribute to which this assignment applies, the type of network elements to which this assignment applies (for default assignments), the network source

to which this assignment applies (for source-specific assignments), the class identifier (CLSID) of the network evaluator being assigned, the edge direction to which this assignment applies (for assignments to an edge source), and the data associated with the assigned network evaluator.



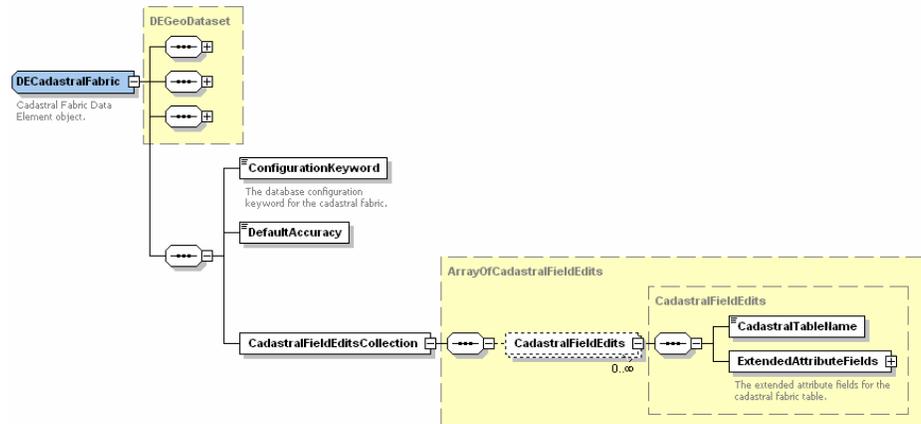
DECadastralFabric

The DECadastralFabric dataset is an extendable data model for the management of parcel data. The serialization of the cadastral fabric schema will include any extended attribute fields of the cadastral fabric tables and the DefaultAccuracy value used by the DECadastralFabric.

The DECadastralFabric data element is derived from DEGeoDataset. It has elements for the database ConfigurationKeyword, DefaultAccuracy, and an array of CadastralFieldEdits.

The CadastralFieldEditsCollection is a collection of CadastralFieldEdits that is used to edit the user-defined fields of the cadastral fabric tables. These are custom fields that are in addition to the system fields that make up a cadastral fabric table.

The DefaultAccuracy element that will be used for cadastral features may be specified on the creation of a cadastral fabric. This value will be applied to new features that do not have an accuracy value explicitly defined. The accuracy values are defined in the cadastral fabric accuracy table. The values represent the quality of instruments used at the time the data was collected.



DERepresentation Class

DERepresentationClass is the data element for a feature class representation and is derived from DEDataset. A feature class representation stores the symbology in the geodatabase for a given feature class. A feature class representation is considered a property of a feature class; that's why these appear as children of a feature class. The symbology is referred to as rules. Two fields from that feature class identify which rule and which container are used for overrides (a binary large object [BLOB] field).

DERepresentationClass has elements for the FeatureClassName, RuleIDFieldName, OverrideFieldName, RequireShapeOverride, Rules, and GeneralAttributes elements.

FeatureClassName is a string property defining the name of the attached feature class in the feature representation.

RuleIDFieldName is a string property defining the integer field from the feature class that determines which rule is used by a feature representation.

OverrideFieldName is a string property defining the BLOB field from the feature class that contains the symbology and geometry overrides for a feature representation.

RequireShapeOverride is a Boolean property determining whether geometry modifications should be stored as geometry override when editing the feature representation.

Rules is a collection of RepresentationRule items that instruct how feature representation within a Representation Class will be symbolized within ArcMap™.

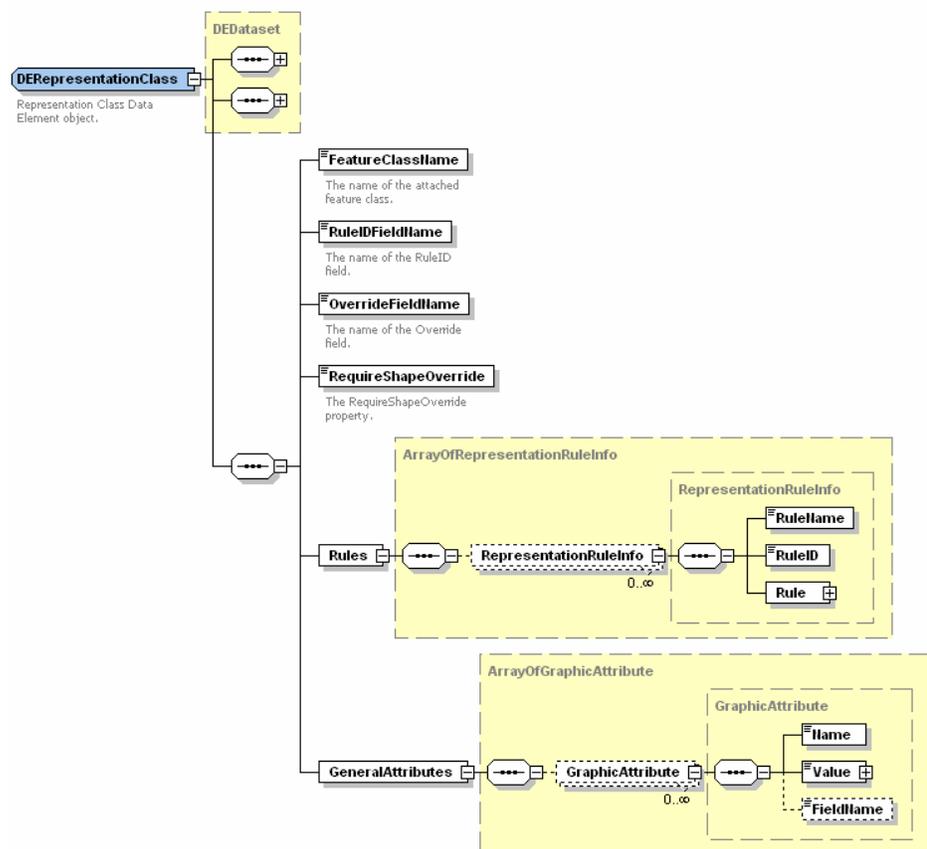
RepresentationRuleInfo consists of elements that indicate the RuleName, RuleID, and Rule information for each RepresentationRule.

RuleName is a string property defining the name of the RepresentationRule.

RuleID is an integer property defining a unique ID value assigned for each RepresentationRule.

Rule is a property defining a RepresentationRule that instructs how feature representations within a RepresentationClass will be symbolized within ArcMap.

GeneralAttributes is a collection of GraphicAttribute items defined at a RepresentationClass level such as Visibility. GraphicAttribute consists of elements that indicate the Name, Value, and FieldName of a graphic or symbol.



DETerrain

DETerrain XML type is derived from DEGeoDataset. Terrain datasets are surfaces that represent three-dimensional space. They use measurements (stored as feature classes) and rules to generate TIN pyramids to represent elevation. The source feature classes can either be referenced by the terrain dataset or be embedded in the terrain dataset.

DETerrain has elements for the FeatureDatasetName, PyramidType, WindowSizeMethod, WindowSizeZThreshold, WindowSizeZThresholdStrategy, TileSize, MaxShapeSize, MaxOverviewSize, ConfigurationKeyword, TerrainDataSources, TerrainPyramidLevelZtols, and TerrainPyramidLevelWindowSizes.

FeatureDatsetName is a string property defining the name of the feature dataset containing the terrain.

PyramidType is an integer property defined by the kind of the pyramid as defined by the type of filter it uses to thin points.

WindowSizeMethod is an integer property defined by the method used by the window size filter to select points.

WindowSizeZThreshold is an integer property defined by the vertical displacement allowed for secondary thinning of points.

WindowSizeZThresholdStrategy is an integer property that controls the potential of secondary thinning to remove points.

TileSize is an integer property defined by the horizontal distance used to spatially index and partition terrain data.

MaxShapeSize is an integer property defined by the maximum number of vertices per multipoint stored in the terrain pyramid.

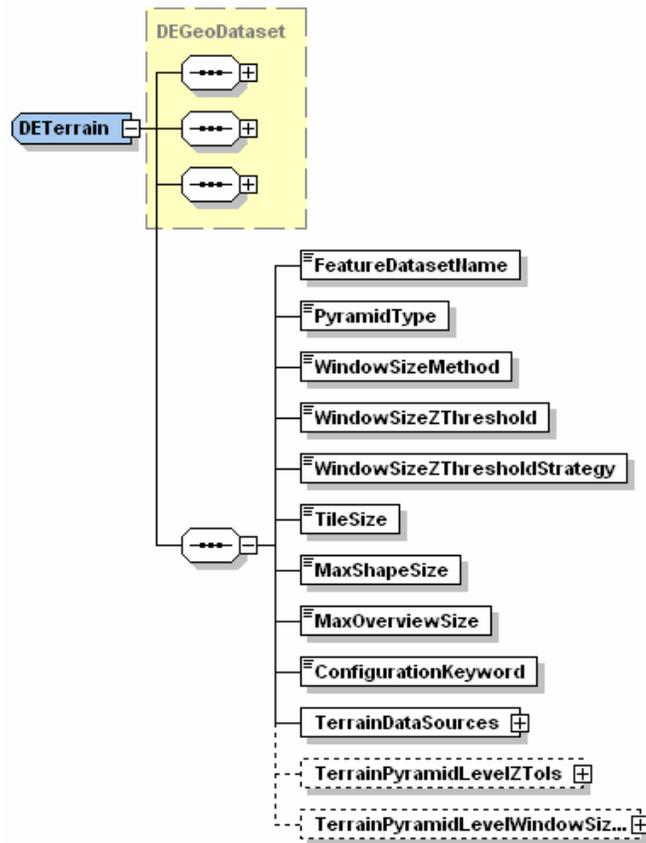
MaxOverviewSize is an integer property defined by the maximum number of points in the most generalized representation of a terrain.

ConfigurationKeyword is a string property defining a Spatial Database Engine™ (SDE™) storage parameter.

TerrainDataSources is defined by type ArrayofTerrainDataSources.

TerrainPyramidLevelZtols is defined by type ArrayTerrainPyramidLevelZtols.

TerrainPyramidLevelWindowSizes is defined by type ArrayTerrainPyramidLevelWindowSize.



TerrainDataSources

TerrainDataSources represents the feature classes participating in the terrain.

FeatureClassID is the database identifier associated with the feature class.

FeatureClassName is the name of the feature class.

GroupID is the terrain group that the feature class belongs to.

SourceStatus represents the flags indicating whether the feature class has just been added, is built into, or has just been removed from the terrain.

SourceType indicates whether the feature class is referenced, embedded, or pending being embedded.

SurfaceFeatureType indicates the way features are used to define the surface.

IsBase indicates whether the feature class participates in the full-resolution terrain.

ApplyToOverview indicates whether the feature class is used in the terrain's overview.

AutoGeneralize indicates whether the features are automatically generalized by the software.

ResolutionLowerBound is the resolution of the finest detail pyramid level in which the feature class participates.

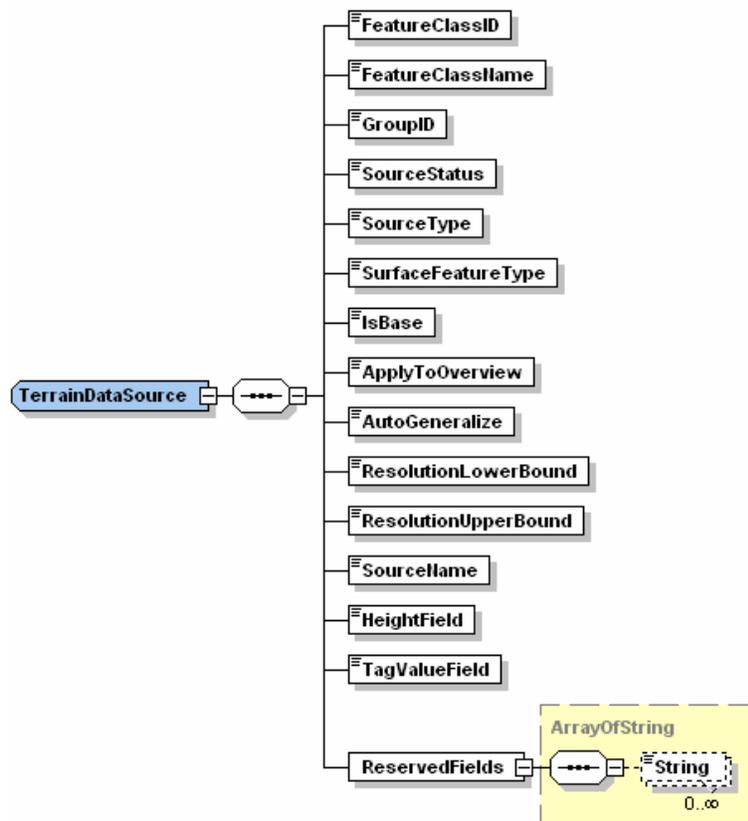
ResolutionUpperBound is the resolution of the coarsest detail pyramid level in which the feature class participates.

SourceName is the name used for an embedded feature class.

HeightField is the name of the field containing z-values for features.

TagValueField is the name of the field containing tag values for features.

ReservedFields indicates the names of any embedded attributes.



TerrainPyramid LevelZTo1

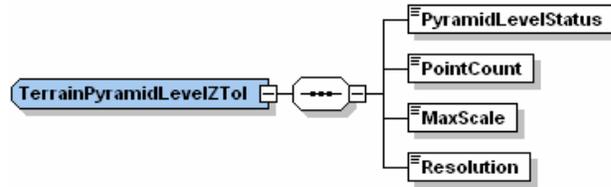
TerrainPyramidLevelZTo1 is represented by the z-tolerance values for all pyramid levels.

PyramidLevelStatus represents the flags indicating whether the pyramid level has just been added, is built into, or has just been removed from the terrain.

PointCount is the number of points in the pyramid level (currently unused).

MaxScale is the largest display scale for the pyramid level.

Resolution is the z-tolerance associated with the pyramid level.



TerrainPyramid LevelWindowSize

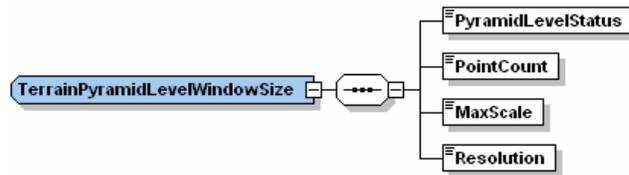
TerrainPyramidLevelWindowSize represents the window size for all pyramid levels.

PyramidLevelStatus represents the flags indicating whether the pyramid level has just been added, is built into, or has just been removed from the terrain.

PointCount is the number of points in the pyramid level (currently unused).

MaxScale is the largest display scale for the pyramid level.

Resolution is the z-tolerance associated with the pyramid level.



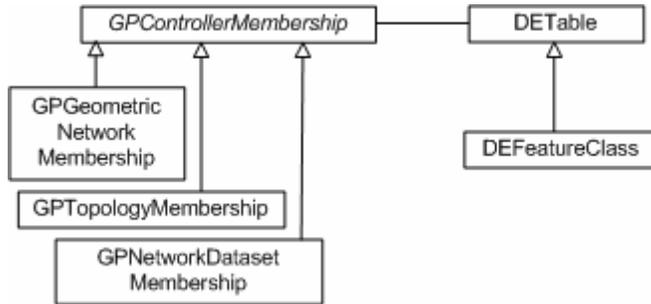
Controllers and Memberships

The association between a feature class and a topology has extra attributes, for example, the rank of the feature class. The association of a feature class and a geometric network also has extra attributes, for example, the enabled field. These are attributes of the feature class's membership in a controlling entity—whether it is the topology or the geometric network. They are attributes of the membership because they wouldn't exist if the feature class was not a member of the controller.

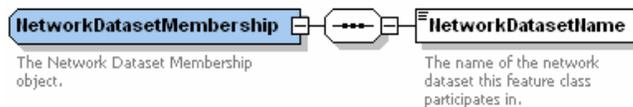
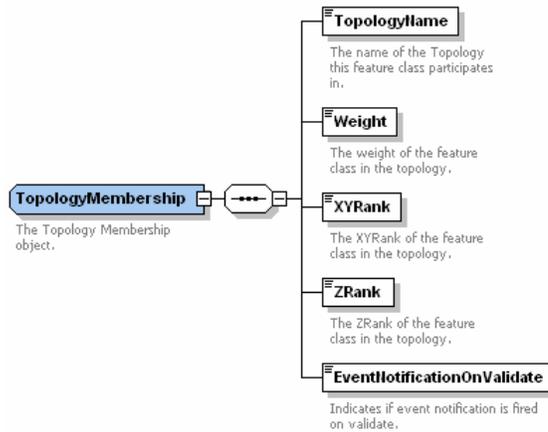
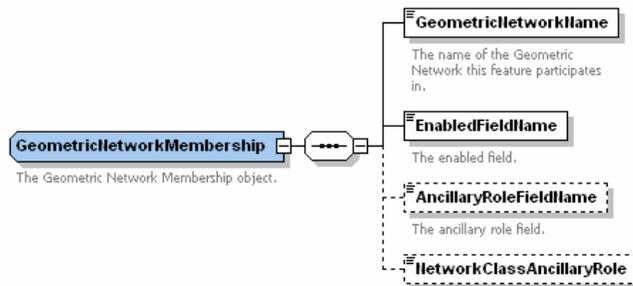
The table data element has an array of controller memberships. Each membership in the array contains data about the relationship between the feature class and the controlling entity (e.g., rank or enabled field).^{*} The following diagram shows the logical association between table and controller memberships.

^{*} Even though tables can't participate in topologies or geometric networks, the controller memberships array is defined in the table to make the solution generic. In the future, tables may be able to participate in new controllers.

J-9620



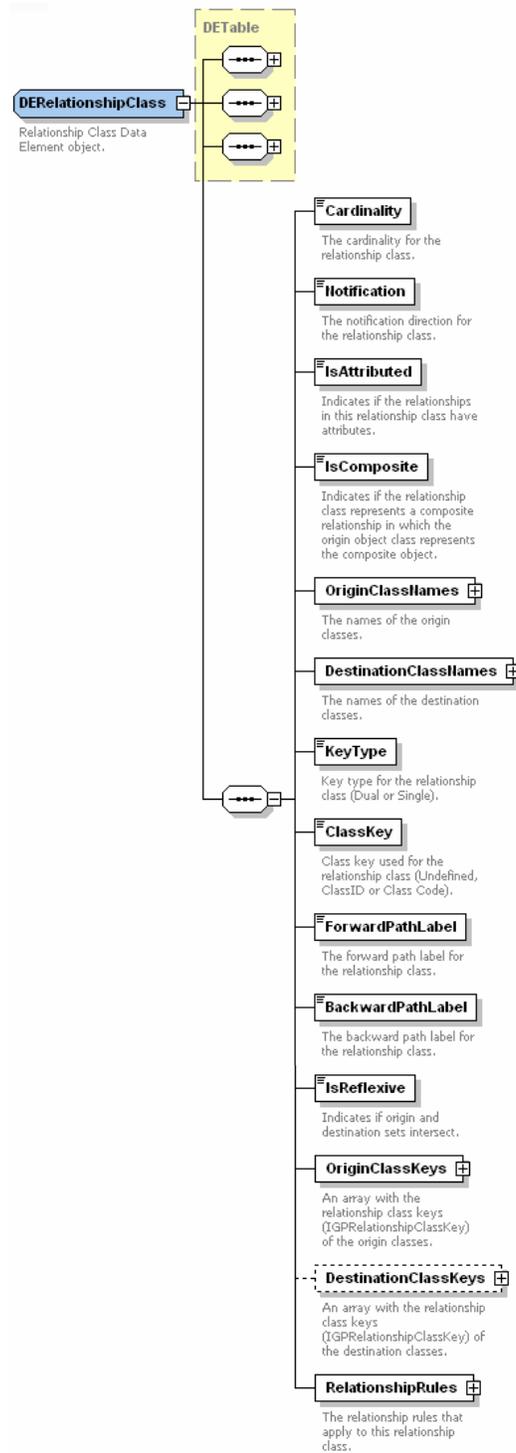
GPControllerMembership is an abstract type that allows different controllers to be placed in the table's controller memberships array. At the time of this writing, controller membership types have been defined for a geometric network, topology, and network dataset. Their layout is shown below.



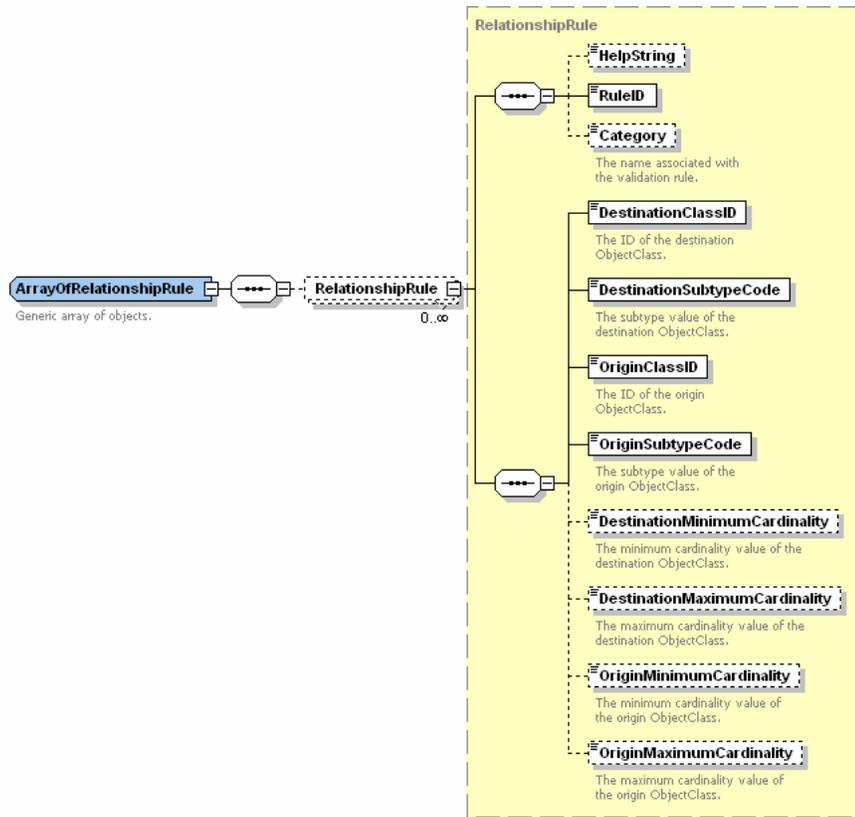
DERelationshipClass The elements in the DERelationshipClass data element type allow the description of geodatabase relationship classes. This type can describe the name of the origin and destination feature classes, cardinality, notification, fields, and so on.

In future releases of ArcGIS, the geodatabase will support relationship classes that involve multiple origin and destination classes as well as relationship classes that are reflexive; that is, whose origin and destination classes are the same. The relationship class data element has been designed to describe current and future relationship classes.

When describing current relationship classes, the data element will have only one origin class name and one destination class name, and IsReflexive will always be false. The array of origin class keys will contain the origin primary and origin foreign keys. If the relationship class is attributed, the destination class keys will contain the destination primary and foreign keys.

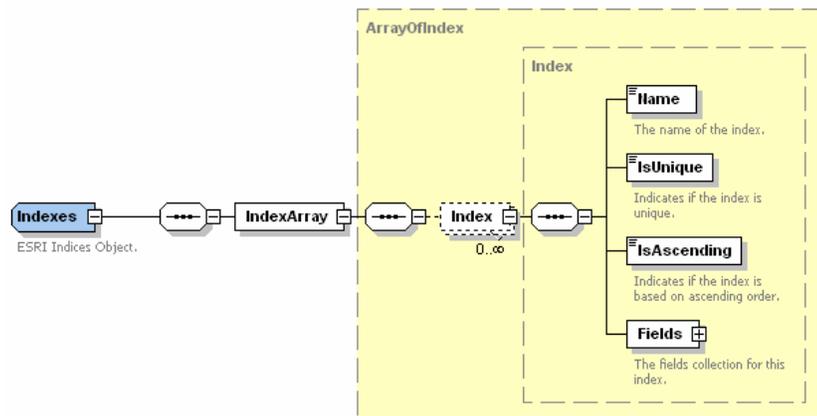


RelationshipRules contains an ArrayOfRelationshipRule that has a RelationshipRule element containing a RelationshipRule. RelationshipRule constrains the cardinality between two subtypes that participate in a RelationshipClass.



Index and Indexes

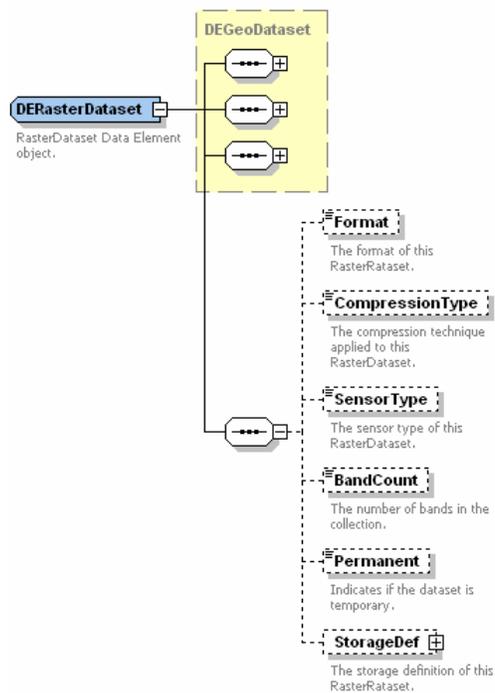
The Index XML type describes a geodatabase index. It has elements for the index name, whether the index is unique or ascending, and the set of fields that makes up the index. Similar to the Fields case, there are types for **IndexArray** and **Indexes**.



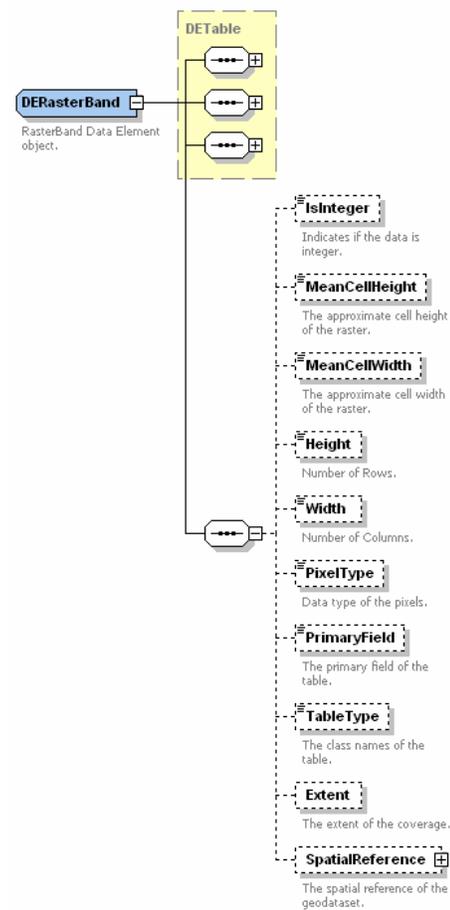
DERasterDataset and DERasterBand

The DERasterDataset XML type is derived from DEGeoDataset. Serialized properties for a raster dataset include its format, compression type, sensor type, number of bands, and whether it is a temporary or permanent dataset. The StorageDef element will hold information about how a raster dataset should be stored in ArcSDE. It defines parameters such as the compression type and quality, pyramid characteristics, tile dimension, origin, and cell size.

A raster dataset may have one or more raster bands. To express this relationship, the Children element of DERasterDataset will contain a collection of DERasterBand properties.



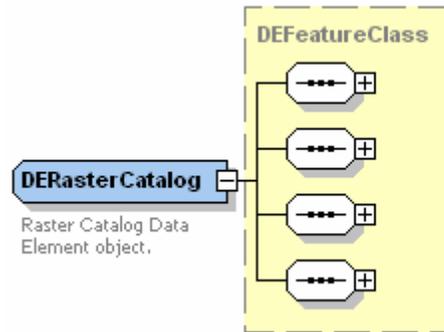
Because a raster band may have an associated table, the DERasterBand XML type is derived from DETable. It holds information such as the type of pixel (e.g., integer or float); number of rows and columns (width and height); cell size; and field in the attribute table whose values are stored in the pixels, extent, and spatial reference.



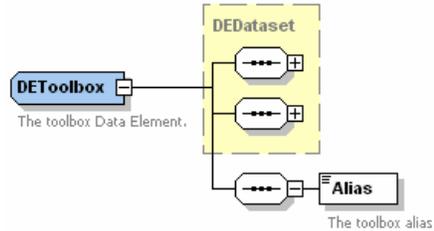
DERasterCatalog

The **DERasterCatalog** XML type is derived from **DEFeatureClass**. The raster catalog manages a set of related raster datasets; in addition, it also manages their footprints, names, metadata, and any other user attributes. Specifically, a raster catalog is a custom feature class containing a field of type **Raster** and a predefined **Shape** field of type **Geometry**; each raster dataset corresponds to a feature.

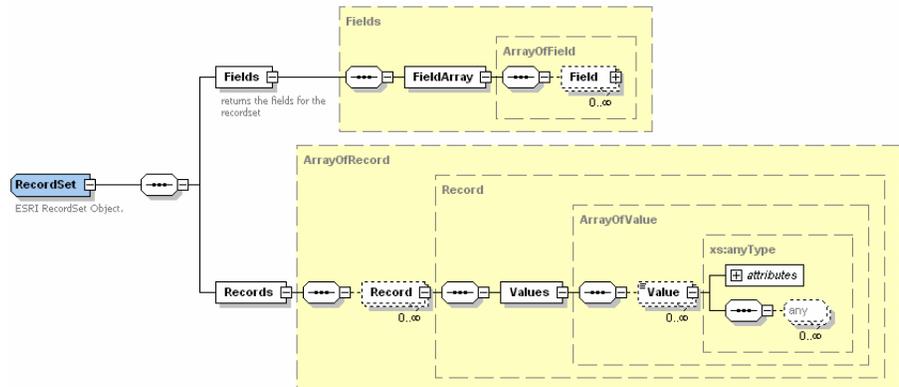
The raster field can have a spatial reference that is applied to all raster values, and it is defined using **RasterDef**. The **Shape** field stores the footprint of a raster value in a given target spatial reference (equal to the spatial reference of the **Shape** field). The footprint is automatically managed, populated, and spatially indexed by the geodatabase.



DEToolbox Toolboxes are represented using the DEToolbox type. It is derived from DEDataset, adding the Alias element.



RecordSet RecordSet type represents the tabular data in a database. It consists of a set of fields and a set of records. A record consists of a set of values, one for each field in the record set. The set of fields and the set of values for a record are ordered and must correspond; that is, the first value in each record must correspond to the first field, and so on.



The following is an XML fragment that shows what a record set looks like. In this case, the record set has three fields, and there are only two records. The type of the third field, POPULATION, is double. The type of the values in each record is also double. This is a restriction for record sets that can't be expressed in XML schema.

```
<Data xsi:type="esri:RecordSet">
  <Fields xsi:type="esri:Fields">
    <FieldArray xsi:type="esri:ArrayOfField">
```

```

<Field xsi:type="esri:Field">
  <Name>OBJECTID</Name>
  <Type>esriFieldTypeOID</Type>
  <IsNullable>>false</IsNullable>
</Field>
<Field xsi:type="esri:Field">
  <Name>TRACT_ID</Name>
  <Type>esriFieldTypeDouble</Type>
  <IsNullable>>true</IsNullable>
</Field>
<Field xsi:type="esri:Field">
  <Name>POPULATION</Name>
  <Type>esriFieldTypeDouble</Type>
  <IsNullable>>true</IsNullable>
</Field>
</FieldArray>
</Fields>
<Records xsi:type="esri:ArrayOfRecord">
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">1</Value>
      <Value xsi:type="xs:double">100</Value>
      <Value xsi:type="xs:double">4231</Value>
    </Values>
  </Record>
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">2</Value>
      <Value xsi:type="xs:double">200</Value>
      <Value xsi:type="xs:double">1683</Value>
    </Values>
  </Record>
</Records>
</Data>

```

If the table has a field of type BLOB, the record set will serialize its values using the type XMLPersistedObject, which is explained later.

RasterData

To prevent the creation of very large XML files when raster columns are present, ArcGIS exports such rasters into separate files. All files will be placed in a directory with the same name as the original XML document but with the extension .images. For example, if the export document name is c:\temp\usa.xml, the directory with the raster files will be called c:\temp\usa.images.

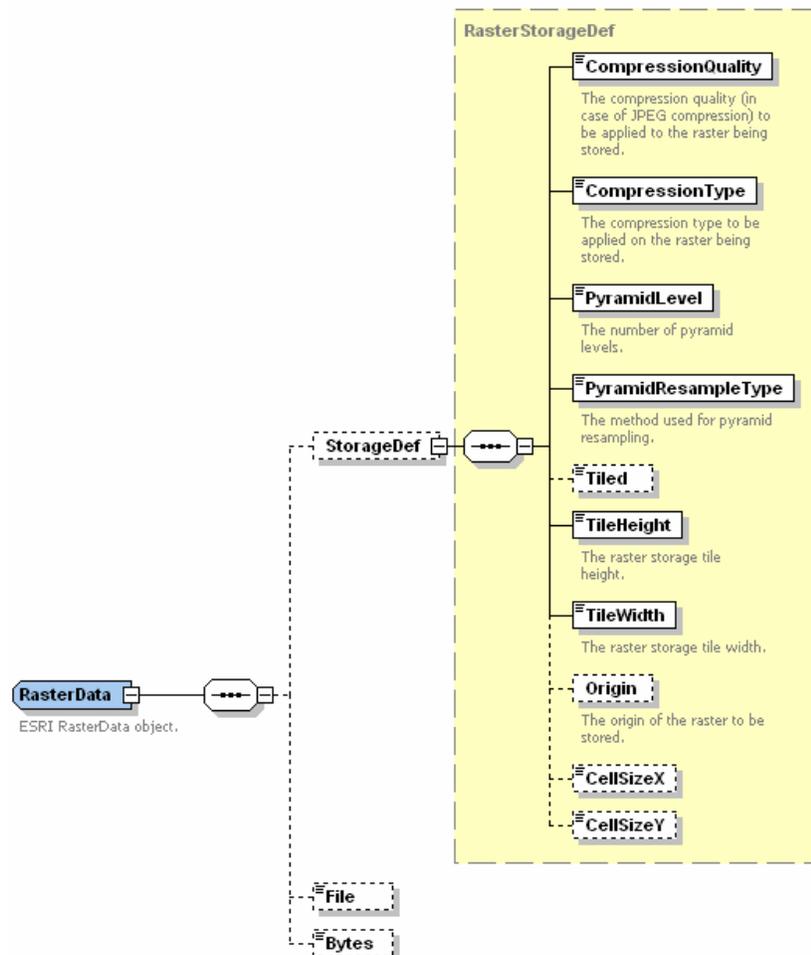
Each record in the record set will store a reference to the corresponding raster file using a value of type RasterData. Here is an example of a record with three values, the second of which references a raster file. The element StorageDef has been removed for brevity:

```

<Record xsi:type="esri:Record">
  <Values xsi:type="esri:ArrayOfValue">
    <Value xsi:type="xs:int">1</Value>
    <Value xsi:type="esri:RasterData">
      <StorageDef xsi:type="esri:RasterStorageDef">
        </StorageDef>
        <File>rd_386997465.img</File>
      </Value>
    <Value xsi:nil="true"/>
  </Values>
</Record>

```

The RasterData XML type is defined as shown in the diagram below. The StorageDef element contains elements with storage parameters for ArcSDE, the File element has the name of the raster file, and the Bytes element can be used to embed the raster information as base64-encoded binary data. As stated before, the XML exchange tools of ArcGIS will not embed the raster information in the generated documents.



Geometry Types

In feature classes, the Shape field contains the geometry for each feature. These geometries are serialized to XML just like any other field value. Geometry serialization can be done in one of two ways: normalized or binary.

In normalized form, geometry is exported to XML by mapping individual components of the geometry to child elements. For example, a point has the following child elements: X, Y, M, Z, and ID. This example shows a point in normalized form:

```
<Value xsi:type="esri:Point">
  <X>405.28579</X>
  <Y>1496.42008</Y>
  <M>10</M>
  <Z>1</Z>
</Value>
```

In its binary form, geometry is exported using base64 encoding. This is the same point in binary form:

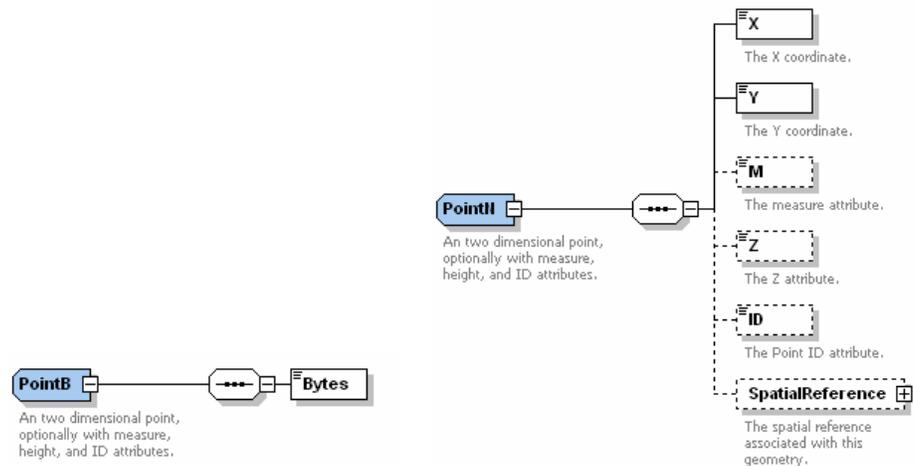
```
<Value xsi:type="esri:Point">
  <Bytes>AQAAAAACMkCtiVAZHWr5/R7oEA=</Bytes>
</Value>
```

Internally, the exporter will invoke the `ExportToESRIShape` method on the `ArcObjects` geometry, then encode the resulting byte stream in base64 format. Conversely, the importer will decode the base64 string into a byte stream, then invoke `ImportFromESRIShape` to re-create the geometry.

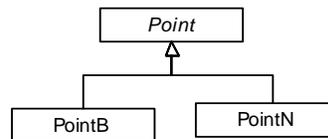
In normalized form, polygons and polylines are represented by much longer XML strings. Polygons, for example, are made of an array of rings, each containing an array of segments and each segment containing an array of points. See appendix A for an example of polygon serialization in both forms.

Binary serialization produces smaller documents and performs better than normalized serialization. On the other hand, a third-party tool using an XML parser can interpret normalized geometries easily without requiring `ArcObjects`.

Because a geometry may be represented in binary or normalized form, there are two XML types for each geometry. They are identified with the suffixes N for normalized and B for binary, for example, `PointN` and `PointB`. Below is a diagram that represents both XML point types.



To enable these representations to be used interchangeably in instance documents, a base abstract type was defined, and the concrete types derive from it. The diagram below shows the relationship among Point (abstract), PointN, and PointB.

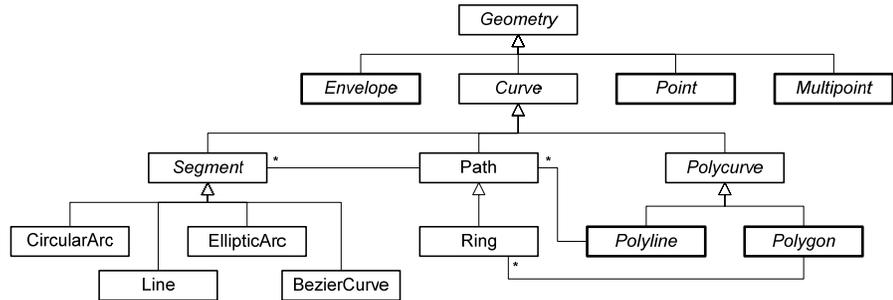


This separation of XML types is followed by all "top-level" geometries; that is, point, polyline, polygon, multipoint, and multipatch. Envelope also follows the same pattern. For all of them, the binary XML type only contains the Bytes element.

EllipticArc, Line, BezierCurve, Path, and Ring types are parts of polylines or polygons. When a polyline or polygon is serialized in normalized form, its parts are also expanded into XML. On the other hand, when serialized in binary form, all its parts are embedded in the base64-encoded string inside the Bytes element. Therefore, only the normalized XML type is required for these geometries.

All geometry XML types have an element for the spatial reference. In a feature class, all features share the same spatial reference. To avoid serializing the same information repeatedly, the spatial reference is not written when a geometry is serialized inside a record set. However, when used in other contexts, such as an argument for a Web service method, an individual geometry will contain the spatial reference.

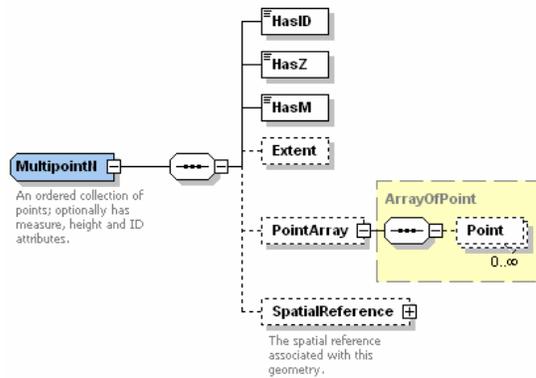
The XML types for geometry form a hierarchy that mirrors that of the geometry classes in ArcObjects. Geometry, the root type, is an abstract XML type with no elements. For simplicity, normalized and binary types have been excluded from the diagram, but they exist for the highlighted geometries.



Point and Multipoint

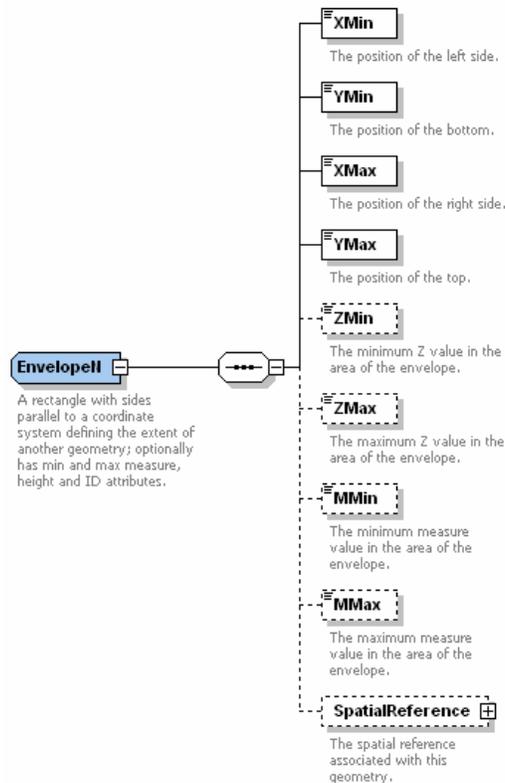
A point is a zero-dimensional geometry. It is defined with an x,y coordinate pair and, optionally, an elevation, measure, and point ID.

A multipoint is a collection of points. In addition to the spatial reference, point array, and bounding envelope, the XML type has elements that indicate whether the geometry has point IDs, elevation, or measures.



Envelope

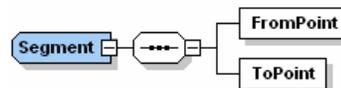
An envelope is a rectangular bounding box. The XML type has elements for the coordinates defining the envelope.



Segment, Line, CircularArc, and EllipticArc

A curve is an abstract type with no elements. All segment-based geometries are derived from a curve.

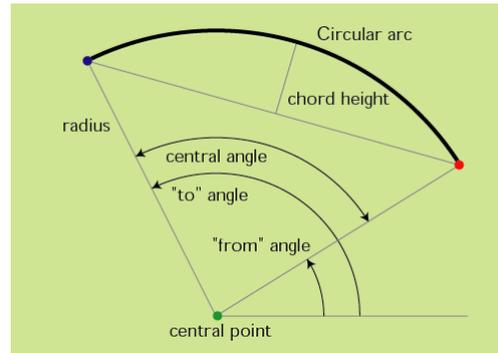
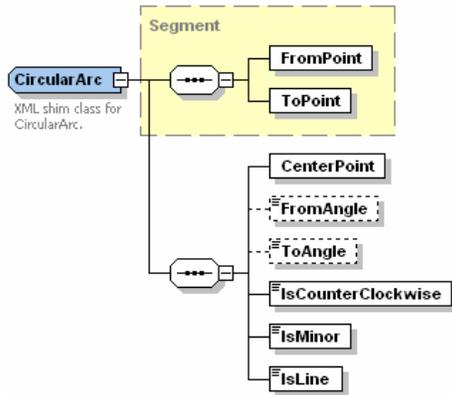
A segment is a parametric curve or function between two endpoints. For example, a line is a linear segment between the points, while a circular arc is a part of a circle between the points. The abstract XML type for a segment has elements for the two endpoints.



A line is a straight segment between two points. The XML type for a line is derived from a segment and adds no extra elements.

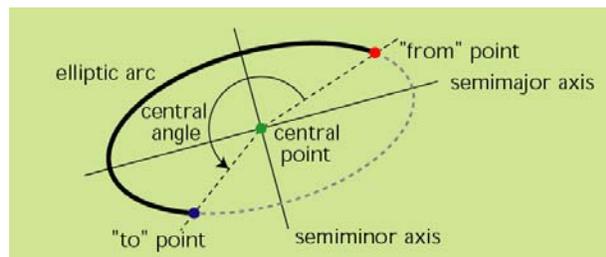
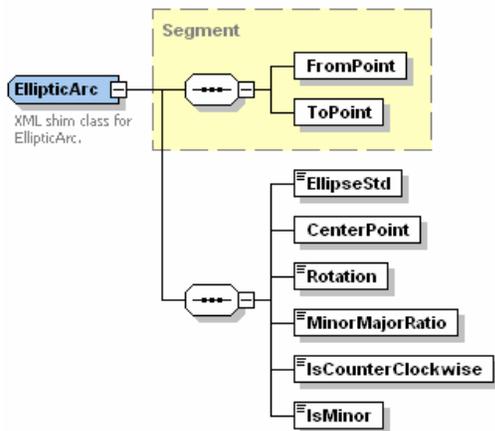
A circular arc is a segment that describes a portion of a circle or a complete circle. FromPoint, ToPoint, and CenterPoint define a circular arc. The IsCounterClockwise element describes how to rotate about the arc's center, getting from the from point to the to point of an arc. The IsLine element indicates when the arc has degenerated to a line (infinite radius). The IsMinor element indicates whether the arc is less than one half of a circle.

When three points defining the arc share the same coordinates (i.e., the arc has degenerated to a point), the elements FromAngle and ToAngle must be present in the arc's serialization.*



An elliptic arc is a segment that describes an ellipse or a part of it. The XML type describes the ellipse using the endpoints and a central point. If EllipseStd is false, the coordinates are measured in a Cartesian coordinate system. If, on the other hand, EllipseStd is true, then FromPoint and ToPoint are offsets relative to CenterPoint.

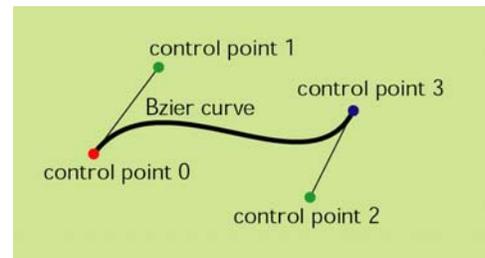
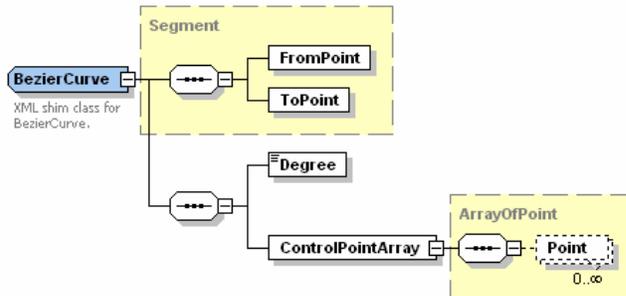
The rotation element contains the angle of the major axis measured counterclockwise from the x-axis. MinorMajorRatio contains the ratio between the axes, and IsCounterClockwise describes how to rotate about the arc's center from the from point to the to point of an arc. Finally, the IsMinor element indicates whether the ellipse is less than one half of an ellipse.



* In a degenerated circular arc where the to, from, and center points coincide, the only way to un-degenerate the circular arc is by providing a radius and a central angle to which its FromAngle and ToAngle have been previously set.

J-9620

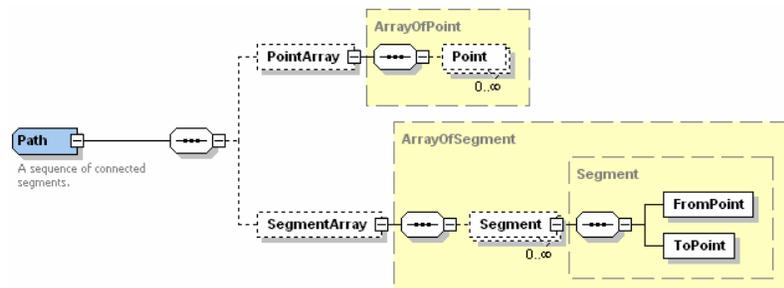
The last parametric segment is the Bezier curve. Such a curve is defined with two endpoints and two extra control points. The XML type has an element for Degree and a second element for the array of control points. In ArcGIS, Degree must be a value of 3 and the array of points must have exactly two points.



Path and Ring

A sequence of connected segments forms a path. A path may also be defined by a collection of points. The XML type for path has an element for each case, but only one is allowed in instance documents.

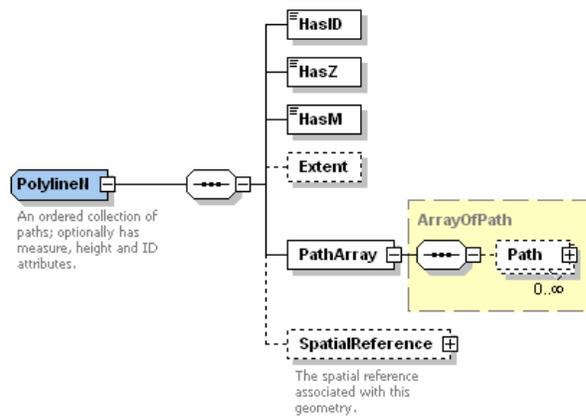
A ring is a closed path. The Ring XML type is derived from a path but doesn't add any extra elements; therefore, paths and rings are represented by the same diagram.



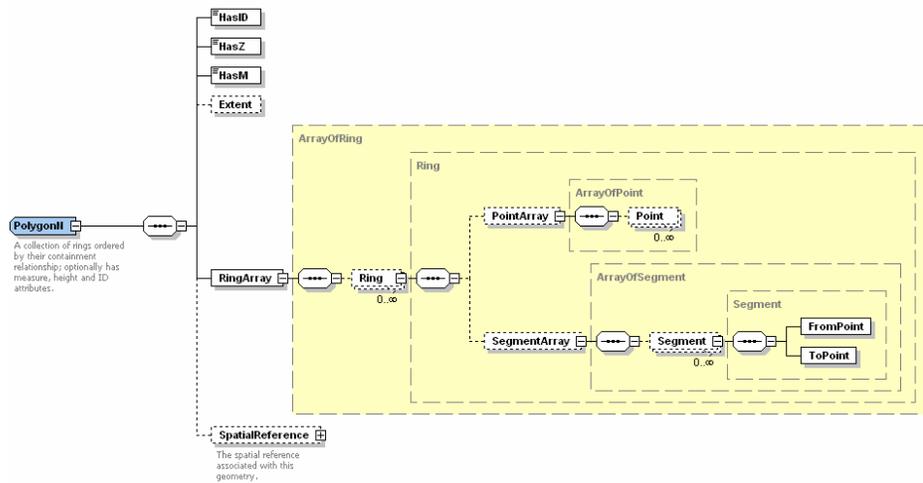
Polycurve, Polyline, and Polygon

Polycurve is an abstract XML type with no elements. It is derived from Curve and Geometry, which also do not have any elements.

Polyline and Polygon types are derived from Polycurve. A polyline is a sequence of paths. The Polyline XML type has elements for HasID, HasZ, HasM, Extent, and SpatialReference. The PathArray element contains an array of paths. As is the case with Point and Multipoint, the Bytes element is only used when the polygon is serialized in binary form.



A polygon is a collection of rings. The XML type for Polygon is identical to the Polyline type above, except PathArray is replaced by a RingArray element. RingArray contains rings that form the polygon.



More information about the geometry classes in ArcObjects can be found in *Exploring ArcObjects*.*

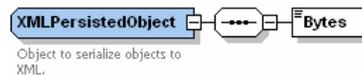
3D Geometry

Multipatches are 3D geometries used to represent the outer surface, or shell, of features that occupy a discrete area or volume in three-dimensional space. Multipatches comprise planar 3D rings and triangles that are used in combination to model a three-dimensional shell. Multipatches can be used to represent anything including simple objects such as iso-surfaces and buildings.

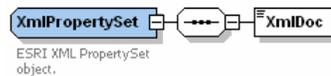
* Zeiler, Michael. *Exploring ArcObjects*, (ESRI Press). Redlands, California, July 2001.

XMLPersistedObject Almost all XML types described so far have a corresponding class in ArcObjects. In fact, XML documents generated by ArcGIS are the result of asking each object to serialize itself into XML. Some objects, however, may not have this capability, or the data itself may be difficult to represent using XML elements, as is the case with BLOB fields that contain images.

In such cases, the object, or BLOB, is written to XML using the XMLPersistedObject type, which has only one element, Bytes. The value of this element is base64-encoded binary data.



XmlPropertySet The XmlPropertySet type is used to serialize metadata. It has a single subelement, XmlDoc, which will contain the XML document representing the metadata. Metadata of tables, feature classes, topologies, and other geodatabase datasets will be represented in this manner.



Appendix A: Binary versus Normalized Serialization

This sample shows a polygon being serialized in binary form.

```
<Value xsi:type="esri:Polygon">
  <Bytes>
DwAAAADjGTT0121AQH5v05+5kkAgufyHtCSCQNCWcymuYZdAAQAAACAAAAAAAAAYP
iImJJUeUDQ
lnMprmGXQCC5/Ie0JIJAMI/8wWCylkDAndLB+n+AQAiPNo54pJRAIEHxY0z5ekBQhA
1Pr9ySQKDT
SEt1ZnRAQH5v05+5kkAA4xk09NdtQAiPNo54pJRAYPiImJJUeUDQlnMprmGXQAAAA
AAPA/AAAA
AAAAGEAAAAAAAAADwPwAAAAAAAAABAAAAAAAAACEAAAAAAAAAQAAAAAAAAABRAAAAAA
AAGEAAAAAA
AADwPwAAAAAACRAAAAAAAATkAAAAAAAkQAAAAAAAAADRAAAAAAAAPkAAAAAAA
BEQAAAAAA
AE1AAAAAAAAATkAAAAAAAkQA==
  </Bytes>
</Value>
```

Below is the same polygon serialized in normalized form. The polygon has two rings.

```
<Value xsi:type="esri:Polygon">
  <HasID>>false</HasID>
  <HasZ>>true</HasZ>
  <HasM>>true</HasM>
  <Extent xsi:type="esri:Envelope">
    <XMin>45.91597</XMin>
    <YMin>207.947770000001</YMin>
    <XMax>1115.26034</XMax>
    <YMax>1145.81537</YMax>
    <ZMin>0</ZMin>
    <ZMax>0</ZMax>
    <MMin>0</MMin>
    <MMax>0</MMax>
  </Extent>
  <RingArray xsi:type="esri:ArrayOfRing">
    <Ring xsi:type="esri:Ring">
      <SegmentArray xsi:type="esri:ArrayOfSegment">
        <Segment xsi:type="esri:Line">
          <FromPoint xsi:type="esri:Point">
            <X>440.34627</X>
            <Y>891.62696</Y>
            <M>NaN</M>
            <Z>0</Z>
          </FromPoint>
          <ToPoint xsi:type="esri:Point">
            <X>580.58815</X>
            <Y>716.3246</Y>
            <M>NaN</M>
            <Z>0</Z>
          </ToPoint>
        </Segment>
```

```

<Segment xsi:type="esri:CircularArc">
  <FromPoint xsi:type="esri:Point">
    <X>580.58815</X>
    <Y>716.3246</Y>
    <M>NaN</M>
    <Z>0</Z>
  </FromPoint>
  <ToPoint xsi:type="esri:Point">
    <X>159.862499999999</X>
    <Y>444.605949999999</Y>
    <M>NaN</M>
    <Z>0</Z>
  </ToPoint>
  <CenterPoint xsi:type="esri:Point">
    <X>387.984009699999</X>
    <Y>552.967956860254</Y>
  </CenterPoint>
  <IsCounterClockwise>>false</IsCounterClockwise>
  <IsMinor>>false</IsMinor>
  <IsLine>>false</IsLine>
</Segment>
<Segment xsi:type="esri:Line">
  <FromPoint xsi:type="esri:Point">
    <X>159.862499999999</X>
    <Y>444.605949999999</Y>
    <M>NaN</M>
    <Z>0</Z>
  </FromPoint>
  <ToPoint xsi:type="esri:Point">
    <X>45.91597</X>
    <Y>602.378070000001</Y>
    <M>NaN</M>
    <Z>0</Z>
  </ToPoint>
</Segment>
<Segment xsi:type="esri:Line">
  <FromPoint xsi:type="esri:Point">
    <X>45.91597</X>
    <Y>602.378070000001</Y>
    <M>NaN</M>
    <Z>0</Z>
  </FromPoint>
  <ToPoint xsi:type="esri:Point">
    <X>440.34627</X>
    <Y>891.62696</Y>
    <M>NaN</M>
    <Z>0</Z>
  </ToPoint>
</Segment>
</SegmentArray>
</Ring>
<Ring xsi:type="esri:Ring">
  <PointArray xsi:type="esri:ArrayOfPoint">
    <Point xsi:type="esri:Point">
      <X>878.602150000001</X>
      <Y>602.378070000001</Y>
      <M>NaN</M>
      <Z>0</Z>
    </Point>
    <Point xsi:type="esri:Point">

```

```
<X>927.128280000001</X>
<Y>585.12434</Y>
<M>NaN</M>
<Z>0</Z>
</Point>
<Point xsi:type="esri:Point">
  <X>1115.26034</X>
  <Y>470.9013</Y>
  <M>NaN</M>
  <Z>0</Z>
</Point>
<Point xsi:type="esri:Point">
  <X>966.25333</X>
  <Y>225.477999999999</Y>
  <M>NaN</M>
  <Z>0</Z>
</Point>
<Point xsi:type="esri:Point">
  <X>738.360269999999</X>
  <Y>207.947770000001</Y>
  <M>NaN</M>
  <Z>0</Z>
</Point>
<Point xsi:type="esri:Point">
  <X>878.602150000001</X>
  <Y>602.378070000001</Y>
  <M>NaN</M>
  <Z>0</Z>
</Point>
</PointArray>
</Ring>
</RingArray>
</Value>
```

Appendix B: Sample Instance Document

The following instance document represents a workspace consisting of a feature class named Schools and a table named Tract_Pop.

```
<esri:Workspace
xmlns:esri="http://www.esri.com/schemas/ArcGIS/9.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <WorkspaceDefinition xsi:type="esri:WorkspaceDefinition">
    <WorkspaceType>esriLocalDatabaseWorkspace</WorkspaceType>
    <Version/>
    <Domains xsi:type="esri:ArrayOfDomain"/>
    <DatasetDefinitions xsi:type="esri:ArrayOfDataElement">
      <DataElement xsi:type="esri:DEFeatureClass">
        <CatalogPath>/FC=Schools</CatalogPath>
        <Name>Schools</Name>
        <DatasetType>esriDTFeatureClass</DatasetType>
        <DSID>15</DSID>
        <Versioned>>false</Versioned>
        <CanVersion>>false</CanVersion>
        <HasOID>>true</HasOID>
        <OIDFieldName>OBJECTID</OIDFieldName>
        <Fields xsi:type="esri:Fields">
          <FieldArray xsi:type="esri:ArrayOfField">
            <Field xsi:type="esri:Field">
              <Name>OBJECTID</Name>
              <Type>esriFieldTypeOID</Type>
              <IsNullable>>false</IsNullable>
              <Length>4</Length>
              <Precision>0</Precision>
              <Scale>0</Scale>
              <Required>>true</Required>
              <Editable>>false</Editable>
              <AliasName>FID</AliasName>
              <ModelName>FID</ModelName>
            </Field>
            <Field xsi:type="esri:Field">
              <Name>Shape</Name>
              <Type>esriFieldTypeGeometry</Type>
              <IsNullable>>true</IsNullable>
              <Length>0</Length>
              <Precision>0</Precision>
              <Scale>0</Scale>
              <Required>>true</Required>
              <GeometryDef xsi:type="esri:GeometryDef">
                <AvgNumPoints>0</AvgNumPoints>
                <GeometryType>esriGeometryPoint</GeometryType>
                <HasM>>false</HasM>
                <HasZ>>false</HasZ>
                <SpatialReference
                  xsi:type="esri:UnknownCoordinateSystem">
                  <XOrigin>2303532.130808</XOrigin>

```

```

        <YOrigin>711403.568308</YOrigin>
        <XYScale>62499.9999417923</XYScale>
    </SpatialReference>
    <GridSize0>5812.44695459666</GridSize0>
</GeometryDef>
<AliasName>Shape</AliasName>
<ModelName>Shape</ModelName>
</Field>
<Field xsi:type="esri:Field">
    <Name>NAME</Name>
    <Type>esriFieldTypeString</Type>
    <IsNullable>true</IsNullable>
    <Length>32</Length>
    <Precision>0</Precision>
    <Scale>0</Scale>
</Field>
<Field xsi:type="esri:Field">
    <Name>SCHOOL_ID</Name>
    <Type>esriFieldTypeInteger</Type>
    <IsNullable>true</IsNullable>
    <Length>4</Length>
    <Precision>0</Precision>
    <Scale>0</Scale>
</Field>
</FieldArray>
</Fields>
<Indexes xsi:type="esri:Indexes">
    <IndexArray xsi:type="esri:ArrayOfIndex">
        <Index xsi:type="esri:Index">
            <Name>FDO_FID</Name>
            <IsUnique>true</IsUnique>
            <IsAscending>true</IsAscending>
            <Fields xsi:type="esri:Fields">
                <FieldArray xsi:type="esri:ArrayOfField">
                    <Field xsi:type="esri:Field">
                        <Name>OBJECTID</Name>
                        <Type>esriFieldTypeOID</Type>
                        <IsNullable>>false</IsNullable>
                        <Length>4</Length>
                        <Precision>0</Precision>
                        <Scale>0</Scale>
                        <Required>true</Required>
                        <Editable>>false</Editable>
                        <AliasName>FID</AliasName>
                        <ModelName>FID</ModelName>
                    </Field>
                </FieldArray>
            </Fields>
        </Index>
        <Index xsi:type="esri:Index">
            <Name>Shape_INDEX</Name>
            <IsUnique>true</IsUnique>
            <IsAscending>true</IsAscending>
            <Fields xsi:type="esri:Fields">
                <FieldArray xsi:type="esri:ArrayOfField">
                    <Field xsi:type="esri:Field">
                        <Name>Shape</Name>
                        <Type>esriFieldTypeGeometry</Type>
                        <IsNullable>true</IsNullable>

```

```

        <Length>0</Length>
        <Precision>0</Precision>
        <Scale>0</Scale>
        <Required>>true</Required>
        <GeometryDef xsi:type="esri:GeometryDef">
            <AvgNumPoints>0</AvgNumPoints>

<GeometryType>esriGeometryPoint</GeometryType>
        <HasM>>false</HasM>
        <HasZ>>false</HasZ>
        <SpatialReference
xsi:type="esri:UnknownCoordinateSystem">
            <XOrigin>2303532.130808</XOrigin>
            <YOrigin>711403.568308</YOrigin>
            <XYScale>62499.9999417923</XYScale>
        </SpatialReference>
            <GridSize0>5812.44695459666</GridSize0>
        </GeometryDef>
        <AliasName>Shape</AliasName>
        <ModelName>Shape</ModelName>
    </Field>
</FieldArray>
</Fields>
</Index>
</IndexArray>
</Indexes>
<CLSID>{52353152-891A-11D0-BEC6-00805F7C4268}</CLSID>
<EXTCLSID/>
<RelationshipClassNames xsi:type="esri:Names"/>
<AliasName>Schools</AliasName>
<ModelName/>
<HasGlobalID>>false</HasGlobalID>
<GlobalIDFieldName/>
<RasterFieldName/>
<ExtensionProperties xsi:type="esri:PropertySet">
    <PropertyArray
xsi:type="esri:ArrayOfPropertySetProperty"/>
</ExtensionProperties>
<ControllerMemberships
xsi:type="esri:ArrayOfControllerMembership"/>
<FeatureType>esriFTSimple</FeatureType>
<ShapeType>esriGeometryPoint</ShapeType>
<ShapeFieldName>Shape</ShapeFieldName>
<HasM>>false</HasM>
<HasZ>>false</HasZ>
<HasSpatialIndex>>true</HasSpatialIndex>
<AreaFieldName/>
<LengthFieldName/>
<Extent xsi:type="esri:Envelope">
    <XMin>2310757.99999873</XMin>
    <YMin>715855.687496146</YMin>
    <XMax>2330666.00000127</XMax>
    <YMax>741311.187503854</YMax>
</SpatialReference
xsi:type="esri:UnknownCoordinateSystem">
    <XOrigin>2303532.130808</XOrigin>
    <YOrigin>711403.568308</YOrigin>
    <XYScale>62499.9999417923</XYScale>
</SpatialReference>

```

```

</Extent>
<SpatialReference xsi:type="esri:UnknownCoordinateSystem">
  <XOrigin>2303532.130808</XOrigin>
  <YOrigin>711403.568308</YOrigin>
  <XYScale>62499.9999417923</XYScale>
</SpatialReference>
</DataElement>
<DataElement xsi:type="esri:DETable">
  <CatalogPath>/OC=Tract_Pop</CatalogPath>
  <Name>Tract_Pop</Name>
  <DatasetType>esriDTable</DatasetType>
  <DSID>18</DSID>
  <Versioned>>false</Versioned>
  <CanVersion>>false</CanVersion>
  <HasOID>>true</HasOID>
  <OIDFieldName>OBJECTID</OIDFieldName>
  <Fields xsi:type="esri:Fields">
    <FieldArray xsi:type="esri:ArrayOfField">
      <Field xsi:type="esri:Field">
        <Name>OBJECTID</Name>
        <Type>esriFieldTypeOID</Type>
        <IsNullable>>false</IsNullable>
        <Length>4</Length>
        <Precision>0</Precision>
        <Scale>0</Scale>
        <Required>>true</Required>
        <Editable>>false</Editable>
        <AliasName>Rowid</AliasName>
        <ModelName>Rowid</ModelName>
      </Field>
      <Field xsi:type="esri:Field">
        <Name>TRACT_ID</Name>
        <Type>esriFieldTypeDouble</Type>
        <IsNullable>>true</IsNullable>
        <Length>8</Length>
        <Precision>0</Precision>
        <Scale>0</Scale>
      </Field>
      <Field xsi:type="esri:Field">
        <Name>POPULATION</Name>
        <Type>esriFieldTypeDouble</Type>
        <IsNullable>>true</IsNullable>
        <Length>8</Length>
        <Precision>0</Precision>
        <Scale>0</Scale>
      </Field>
    </FieldArray>
  </Fields>
  <Indexes xsi:type="esri:Indexes">
    <IndexArray xsi:type="esri:ArrayOfIndex">
      <Index xsi:type="esri:Index">
        <Name>FDO_Rowid</Name>
        <IsUnique>>true</IsUnique>
        <IsAscending>>true</IsAscending>
        <Fields xsi:type="esri:Fields">
          <FieldArray xsi:type="esri:ArrayOfField">
            <Field xsi:type="esri:Field">
              <Name>OBJECTID</Name>
              <Type>esriFieldTypeOID</Type>

```

```

        <IsNullable>>false</IsNullable>
        <Length>4</Length>
        <Precision>0</Precision>
        <Scale>0</Scale>
        <Required>>true</Required>
        <Editable>>false</Editable>
        <AliasName>Rowid</AliasName>
        <ModelName>Rowid</ModelName>
    </Field>
</FieldArray>
</Fields>
</Index>
</IndexArray>
</Indexes>
<CLSID>{7A566981-C114-11D2-8A28-006097AFF44E}</CLSID>
<EXTCLSID/>
<RelationshipClassNames xsi:type="esri:Names"/>
<AliasName>Tract_Pop</AliasName>
<ModelName/>
<HasGlobalID>>false</HasGlobalID>
<GlobalIDFieldName/>
<RasterFieldName/>
<ExtensionProperties xsi:type="esri:PropertySet">
    <PropertyArray
xsi:type="esri:ArrayOfPropertySetProperty"/>
</ExtensionProperties>
<ControllerMemberships
xsi:type="esri:ArrayOfControllerMembership"/>
</DataElement>
</DatasetDefinitions>
</WorkspaceDefinition>
<WorkspaceData xsi:type="esri:WorkspaceData">
    <DatasetData xsi:type="esri:TableData">
        <DatasetName>Schools</DatasetName>
        <DatasetType>esriDTFeatureClass</DatasetType>
        <Data xsi:type="esri:RecordSet">
            <Fields xsi:type="esri:Fields">
                <FieldArray xsi:type="esri:ArrayOfField">
                    <Field xsi:type="esri:Field">
                        <Name>OBJECTID</Name>
                        <Type>esriFieldTypeOID</Type>
                        <IsNullable>>false</IsNullable>
                        <Length>4</Length>
                        <Precision>0</Precision>
                        <Scale>0</Scale>
                        <Required>>true</Required>
                        <Editable>>false</Editable>
                        <AliasName>FID</AliasName>
                        <ModelName>FID</ModelName>
                    </Field>
                    <Field xsi:type="esri:Field">
                        <Name>Shape</Name>
                        <Type>esriFieldTypeGeometry</Type>
                        <IsNullable>>true</IsNullable>
                        <Length>0</Length>
                        <Precision>0</Precision>
                        <Scale>0</Scale>
                        <Required>>true</Required>
                        <GeometryDef xsi:type="esri:GeometryDef">

```

```

        <AvgNumPoints>0</AvgNumPoints>
        <GeometryType>esriGeometryPoint</GeometryType>
        <HasM>>false</HasM>
        <HasZ>>false</HasZ>
        <SpatialReference
xsi:type="esri:UnknownCoordinateSystem">
            <XOrigin>2303532.130808</XOrigin>
            <YOrigin>711403.568308</YOrigin>
            <XYScale>62499.9999417923</XYScale>
        </SpatialReference>
        <GridSize0>5812.44695459666</GridSize0>
    </GeometryDef>
    <AliasName>Shape</AliasName>
    <ModelName>Shape</ModelName>
</Field>
<Field xsi:type="esri:Field">
    <Name>NAME</Name>
    <Type>esriFieldTypeString</Type>
    <IsNullable>>true</IsNullable>
    <Length>32</Length>
    <Precision>0</Precision>
    <Scale>0</Scale>
</Field>
<Field xsi:type="esri:Field">
    <Name>SCHOOL_ID</Name>
    <Type>esriFieldTypeInteger</Type>
    <IsNullable>>true</IsNullable>
    <Length>4</Length>
    <Precision>0</Precision>
    <Scale>0</Scale>
</Field>
</FieldArray>
</Fields>
<Records xsi:type="esri:ArrayOfRecord">
    <Record xsi:type="esri:Record">
        <Values xsi:type="esri:ArrayOfValue">
            <Value xsi:type="xs:int">1</Value>
            <Value xsi:type="esri:Point">
                <Bytes>AQAAAFj1//8yoUFBT4EAYH6fJkE=</Bytes>
            </Value>
            <Value xsi:type="xs:string">Northwestern
Prep</Value>
            <Value xsi:type="xs:int">1</Value>
        </Values>
    </Record>
    <Record xsi:type="esri:Record">
        <Values xsi:type="esri:ArrayOfValue">
            <Value xsi:type="xs:int">2</Value>
            <Value xsi:type="esri:Point">
                <Bytes>AQAAAIn5/z8/okFBx7wA4CQoJkE=</Bytes>
            </Value>
            <Value xsi:type="xs:string">Elm Elementary</Value>
            <Value xsi:type="xs:int">2</Value>
        </Values>
    </Record>
    <Record xsi:type="esri:Record">
        <Values xsi:type="esri:ArrayOfValue">
            <Value xsi:type="xs:int">3</Value>
            <Value xsi:type="esri:Point">

```

```

        <Bytes>AQAAAI1AAGAAEFB4zoAoMdKJkE=</Bytes>
    </Value>
    <Value xsi:type="xs:string">Stowe Elementary</Value>
    <Value xsi:type="xs:int">3</Value>
</Values>
</Record>
<Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
        <Value xsi:type="xs:int">4</Value>
        <Value xsi:type="esri:Point">
            <Bytes>AQAAAL++/78atUFB3GgAALg0JkE=</Bytes>
        </Value>
        <Value xsi:type="xs:string">Jefferson Middle</Value>
        <Value xsi:type="xs:int">4</Value>
    </Values>
</Record>
<Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
        <Value xsi:type="xs:int">5</Value>
        <Value xsi:type="esri:Point">
            <Bytes>AQAAABPP/58vuUFB1lH/PxIyJkE=</Bytes>
        </Value>
        <Value xsi:type="xs:string">Roosevelt
Elementary</Value>
        <Value xsi:type="xs:int">5</Value>
    </Values>
</Record>
<Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
        <Value xsi:type="xs:int">6</Value>
        <Value xsi:type="esri:Point">
            <Bytes>AQAAAIAnACC9rUFBmvv+f2fZJUE=</Bytes>
        </Value>
        <Value xsi:type="xs:string">The Green Valley
School</Value>
        <Value xsi:type="xs:int">6</Value>
    </Values>
</Record>
</Records>
</Data>
</DatasetData>
<DatasetData xsi:type="esri:TableData">
    <DatasetName>Tract_Pop</DatasetName>
    <DatasetType>esriDTTable</DatasetType>
    <Data xsi:type="esri:RecordSet">
        <Fields xsi:type="esri:Fields">
            <FieldArray xsi:type="esri:ArrayOfField">
                <Field xsi:type="esri:Field">
                    <Name>OBJECTID</Name>
                    <Type>esriFieldTypeOID</Type>
                    <IsNullable>>false</IsNullable>
                    <Length>4</Length>
                    <Precision>0</Precision>
                    <Scale>0</Scale>
                    <Required>>true</Required>
                    <Editable>>false</Editable>
                    <AliasName>Rowid</AliasName>
                    <ModelName>Rowid</ModelName>
                </Field>
            </FieldArray>
        </Fields>
    </Data>
</DatasetData>

```

```

<Field xsi:type="esri:Field">
  <Name>TRACT_ID</Name>
  <Type>esriFieldTypeDouble</Type>
  <IsNullable>true</IsNullable>
  <Length>8</Length>
  <Precision>0</Precision>
  <Scale>0</Scale>
</Field>
<Field xsi:type="esri:Field">
  <Name>POPULATION</Name>
  <Type>esriFieldTypeDouble</Type>
  <IsNullable>true</IsNullable>
  <Length>8</Length>
  <Precision>0</Precision>
  <Scale>0</Scale>
</Field>
</FieldArray>
</Fields>
<Records xsi:type="esri:ArrayOfRecord">
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">1</Value>
      <Value xsi:type="xs:double">100</Value>
      <Value xsi:type="xs:double">4231</Value>
    </Values>
  </Record>
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">2</Value>
      <Value xsi:type="xs:double">200</Value>
      <Value xsi:type="xs:double">1683</Value>
    </Values>
  </Record>
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">3</Value>
      <Value xsi:type="xs:double">300</Value>
      <Value xsi:type="xs:double">2580</Value>
    </Values>
  </Record>
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">4</Value>
      <Value xsi:type="xs:double">400</Value>
      <Value xsi:type="xs:double">6012</Value>
    </Values>
  </Record>
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">5</Value>
      <Value xsi:type="xs:double">500</Value>
      <Value xsi:type="xs:double">7046</Value>
    </Values>
  </Record>
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">6</Value>
      <Value xsi:type="xs:double">600</Value>
      <Value xsi:type="xs:double">5170</Value>
    </Values>
  </Record>
</Records>

```

```
    </Values>
  </Record>
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">7</Value>
      <Value xsi:type="xs:double">700</Value>
      <Value xsi:type="xs:double">6203</Value>
    </Values>
  </Record>
  <Record xsi:type="esri:Record">
    <Values xsi:type="esri:ArrayOfValue">
      <Value xsi:type="xs:int">8</Value>
      <Value xsi:type="xs:double">801</Value>
      <Value xsi:type="xs:double">2914</Value>
    </Values>
  </Record>
</Records>
</Data>
</DatasetData>
</WorkspaceData>
</esri:Workspace>
```



ESRI

380 New York Street
Redlands, California
92373-8100 USA

Phone: 909-793-2853
Fax: 909-793-5953
E-mail: info@esri.com

For more than 35 years, ESRI has been helping people make better decisions through management and analysis of geographic information. A full-service GIS company, ESRI offers a framework for implementing GIS technology and business logic in any organization from personal GIS on the desktop to enterprise-wide GIS servers (including the Web) and mobile devices. ESRI GIS solutions are flexible and can be customized to meet the needs of our users.

For More Information

1-800-GIS-XPRT (1-800-447-9778)

www.esri.com

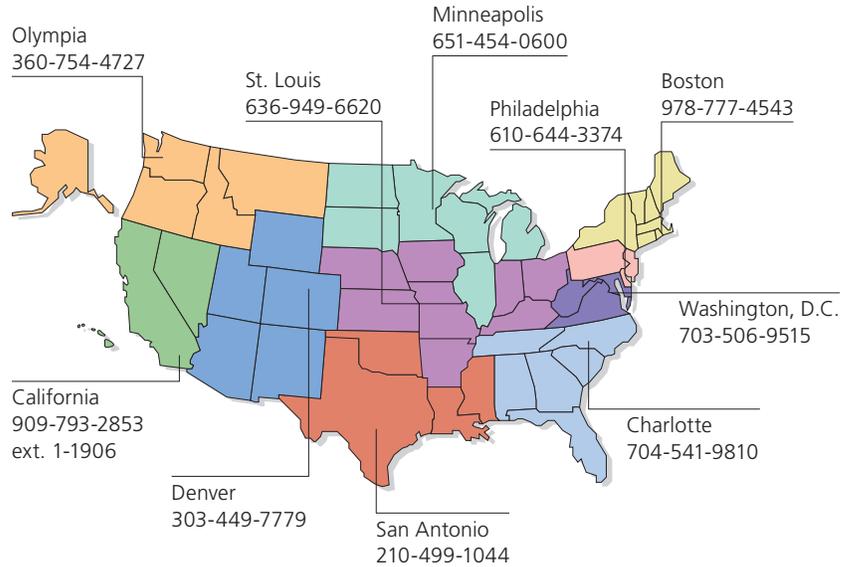
Locate an ESRI value-added reseller near you at

www.esri.com/resellers

Outside the United States, contact your local ESRI distributor. For the number of your distributor, call ESRI at 909-793-2853, ext. 1-1235, or visit our Web site at

www.esri.com/distributors

ESRI Regional Offices



ESRI International Offices

Australia
www.esriaustralia.com.au

Belgium/Luxembourg
www.esribelux.com

Bulgaria
www.esribulgaria.com

Canada
www.esricanada.com

Chile
www.esri-chile.com

China (Beijing)
www.esrichina-bj.cn

China (Hong Kong)
www.esrichina-hk.com

Eastern Africa
www.esri-ke.com

Finland
www.esri-finland.com

France
www.esrifrance.fr

Germany/Switzerland
www.esri-germany.de
www.esri-suisse.ch

Hungary
www.esrihu.hu

India
www.esriindia.com

Indonesia
www.esrisa.com.my

Italy
www.esriitalia.it

Japan
www.esrij.com

Korea
www.esrikr.co.kr

Malaysia
www.esrisa.com.my

Netherlands
www.esri.nl

Northeast Africa
www.esri-nea.com

Poland
www.esripolska.com.pl

Portugal
www.esri-portugal.pt

Romania
www.esriro.com

Singapore
www.esrisa.com

Spain
www.esri-es.com

Sweden
www.esri-sgroup.se

Thailand
www.esri-th.com

Turkey
www.esriturkey.com.tr

United Kingdom
www.esriuk.com

Venezuela
www.esri-ven.com



No. GS-35F-5086H