# XML Schema of the Geodatabase

*An ESRI ® Technical Paper • February 2004*

**ESRI** 380 New York St., Redlands, CA 92373-8100, USA • TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB www.esri.com

# XML Schema of the Geodatabase

## An ESRI Technical Paper

### Contents

# XML Schema of the Geodatabase

Since version 9.0, ArcGIS® has provided the ability to transfer geodatabase data using XML.With the ArcGIS XML format, you can import and export all the 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 of the geodatabase XML types. The details of all the 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[1]:

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

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>
</Field>
```

XML schema can be used to specify the format that the XML document should follow. For 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).

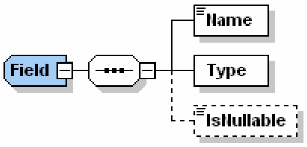
[1] A reduced set of properties for a field is shown here. See the XML schema for the full definition of the field XML type.

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="IsNullable" type="xs:boolean"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

Notice that the IsNullable 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 graphical notation to represent the XML types. The following figure is the graphic that corresponds to the Field type. Notice how the optional element IsNullable is represented.

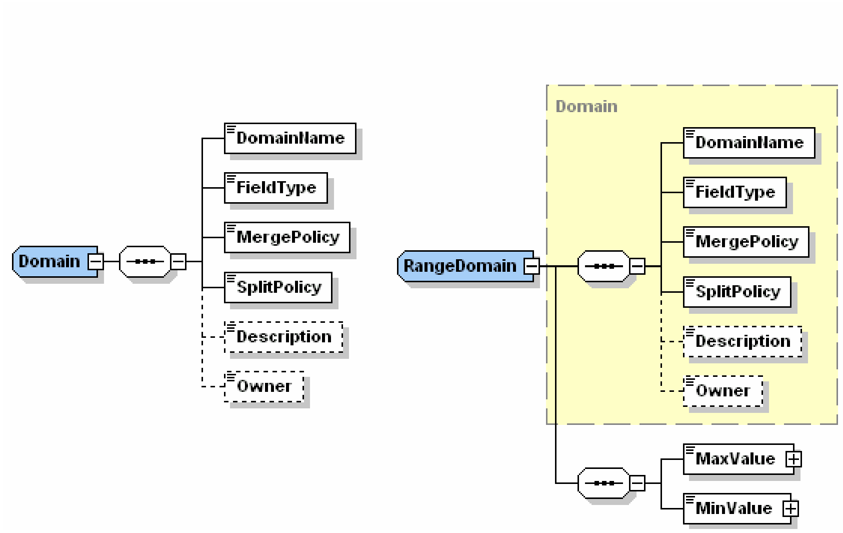


An XML document can be validated against its XML Schema. In the field case, the validating tool can compare the instance[1] document versus the schema and determine if it complies with the schema rules. A document without a Name element for a field would be considered schema invalid. Note it could still 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”).

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 a field type. A range domain is derived from it and adds two more elements, MinValue and MaxValue.

[1] XML documents that are compliant with an XML schema are commonly referred to as *instance* documents. Exporting one geodatabase to XML will generate one of such instance documents.



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, 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 the types defined in the schema belong to that namespace. 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/9.0.

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

The Open GIS Consortium has published a standard for the exchange of geographic data using XML called GML [1]. 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.

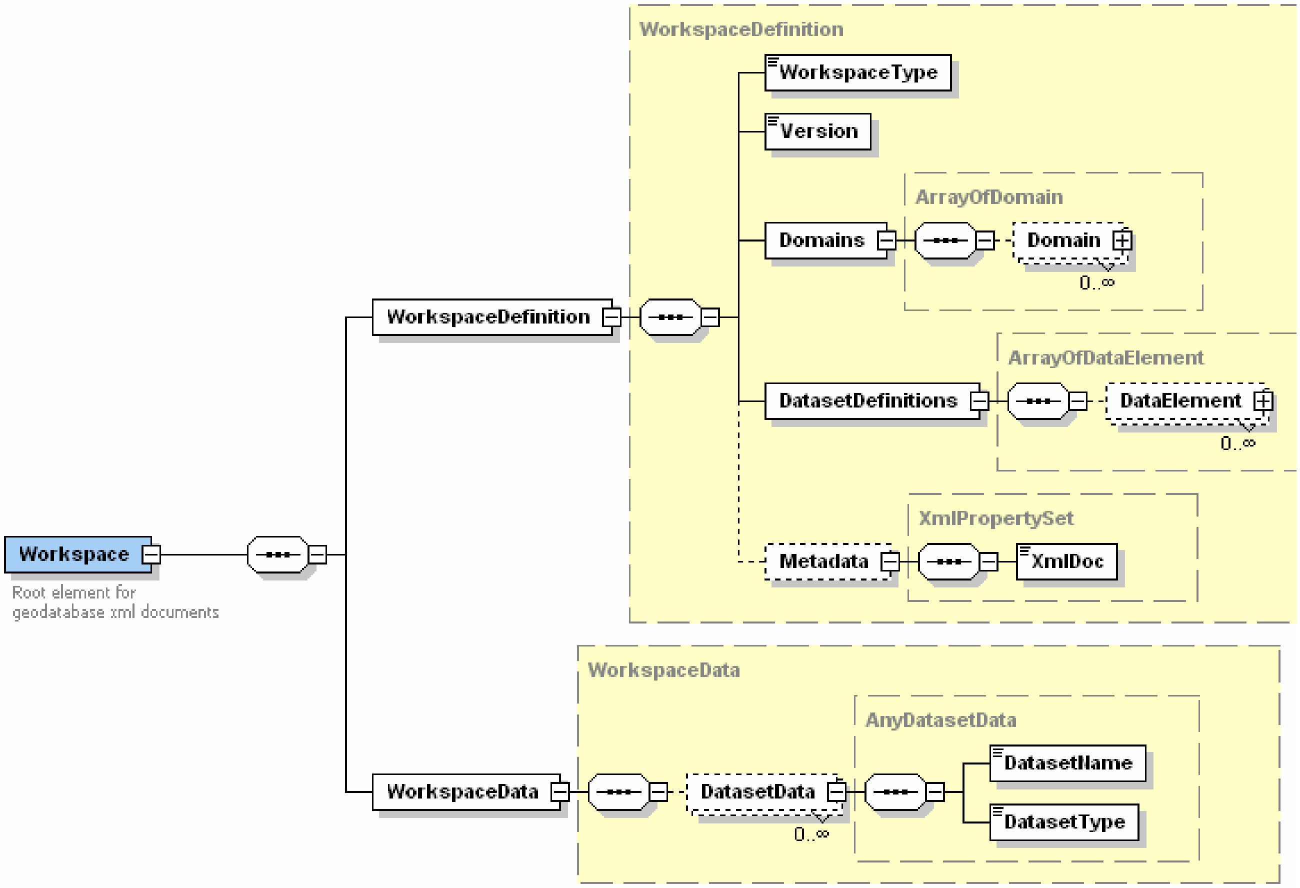
The above discussion of XML Schema is basic and intended to let you understand the basic constructs in the XML Schema for XML exchange in ArcGIS. More information about this topic can be found in XML Schema Part 0: Primer [2].

## Geodatabase XML document types

There are three types of XML documents that can be created in ArcGIS: a Workspace document, a RecordSet document, and a Data Changes document. They are covered in detail below.

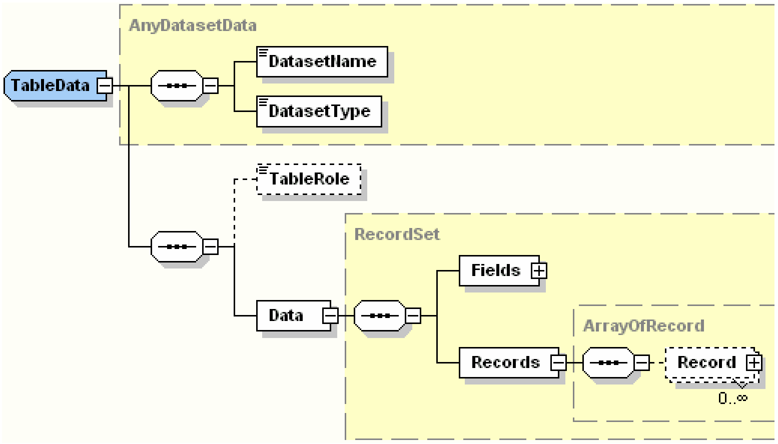
### Workspace document

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

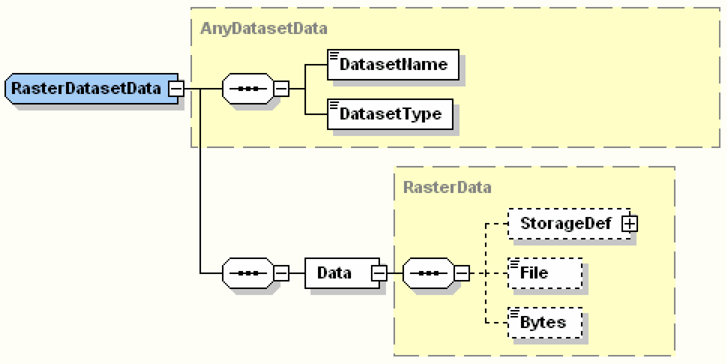


WorkspaceDefinition has the workspace type, the version being exported (if the exported database is based on ArcSDE®), an array of domains (notice there can be zero to many domains present), and an array of data elements. Domain is an abstract type. Instances of the derived types range and coded-value 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, the RecordSet contains the actual records from a table. RecordSets are discussed in detail later in this document.



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

Using the 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.

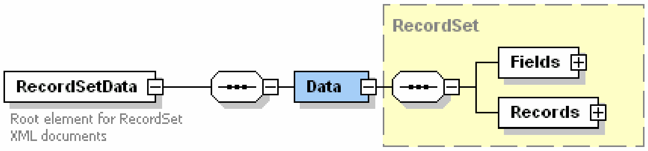
In the case of tables and feature classes, the name of the data element in the definition section corresponds to the name of the table data in the data section. Other datasets, such as topologies, geometric networks, or feature datasets, appear only in the definition section (no data is exported for 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 RecordSet document. The features are exported as simple features, and no additional geodatabase-related information is written to the output file. Exporting to a RecordSet 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.

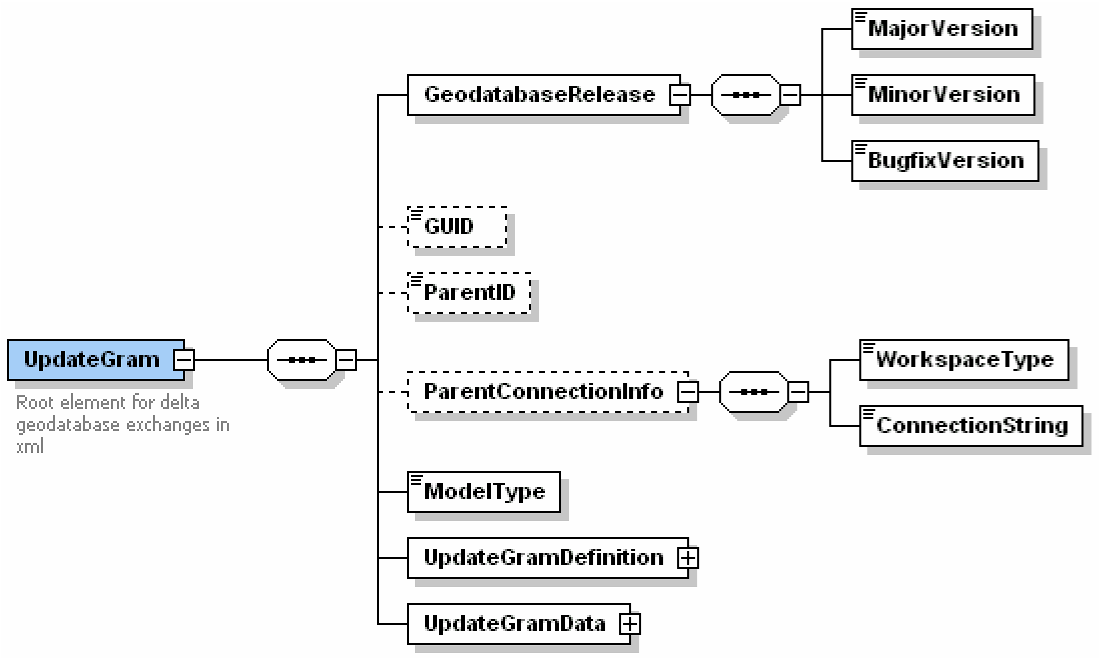
The root element of such a document is RecordSetData, consisting of just one mandatory child element called Data. The XML type of the data element is RecordSet. RecordSet contains a set of fields and the records of the exported table. RecordSet is covered in detail later on.



## Data Changes document

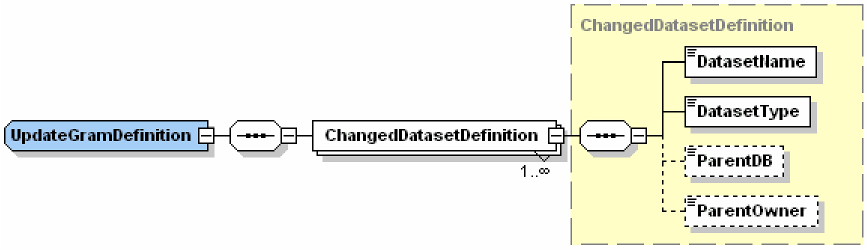
The ArcGIS disconnected editing framework allows you to check data out of a database into a separate geodatabase, then edit the data without having a live connection to the parent database. Once the editing is done, it is possible to export only the changes (not all the data) to an XML file. This file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

The root element of a Data Changes document is an UpdateGram. The GeodatabaseRelease contains version information about the parent database (Major, Minor, and bug versions). The GUID subelement contains the unique identifier for the checkout database (assigned at checkout time). The 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.

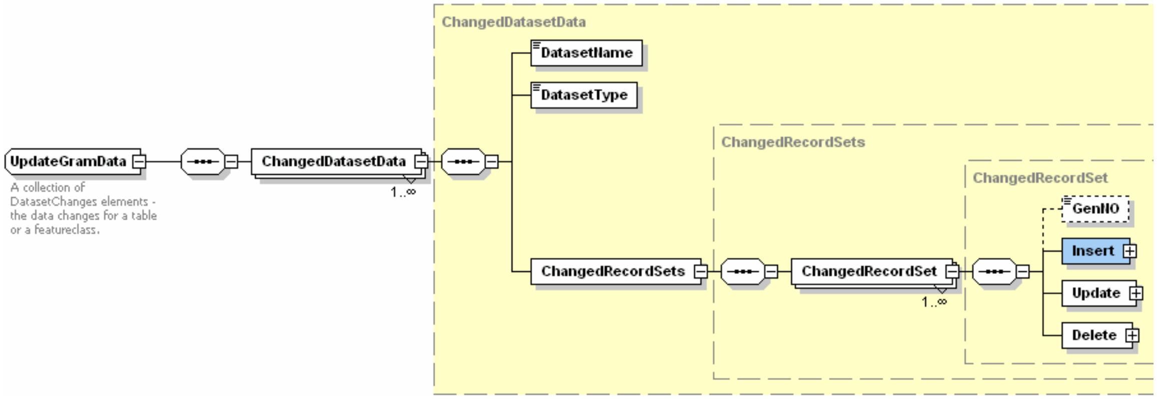


UpdateGramDefinition identifies the datasets that were modified in the checkout database. Conversely, the UpdateGramData element contains the actual data that was changed. This separation allows software programs to find the data that was changed without needing to search the entire geodatabase for changed features.

The 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).



The last element of the UpdateGram is the 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), updates to, or deletion of existing features. To record these actions each ChangedDatasetData element contains RecordSets with the features inserted, updated, or deleted.

The Delete RecordSet only has one field, the object ID or global ID, 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 that appears in the UpdateGramDefinition section must have a corresponding ChangedDatasetData element in the UpdateGramData section.

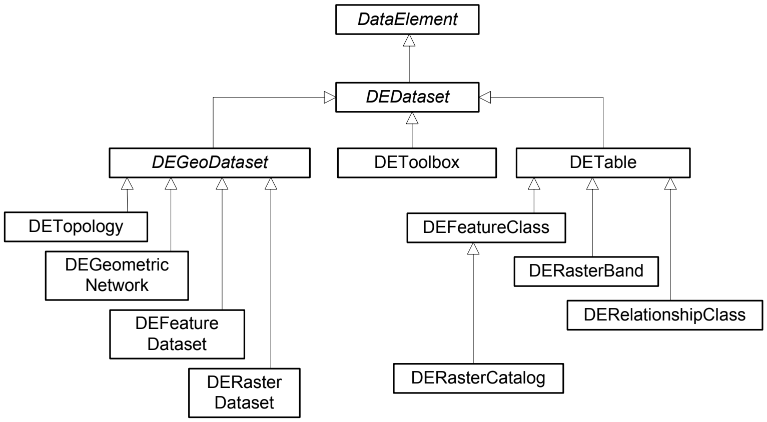
## XML types in the schema

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

### Data elements

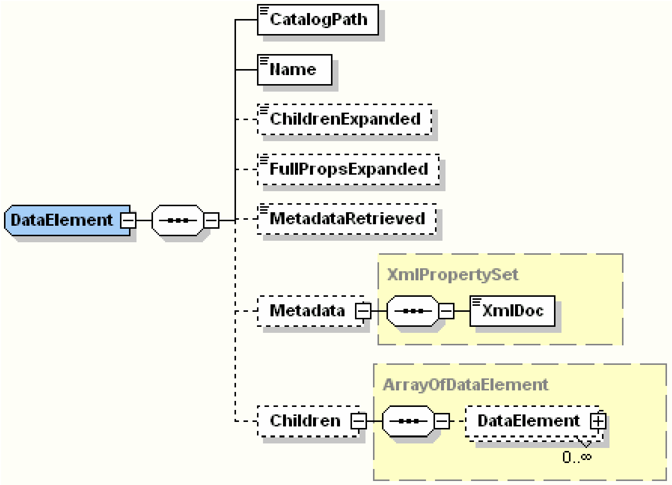
Data elements are a set of ArcObjects classes added in ArcGIS 9. These objects describe all aspects of a dataset in a geodatabase. For example, the data element for a feature class captures name, feature type, fields, indexes, subtypes, and other characteristics of a feature class.

There are data elements for most of the geodatabase items. All the information that relates to database schema is exported to XML using these data elements. Here is part of the hierarchy of XML types that corresponds to the data element ArcObjects. The arrow points to the parent type, and the names in italics denote abstract types.



## DataElement

DataElement is an abstract type used as base type for all data elements; therefore, all data elements have the elements of 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 array of DataElements, which allows for some data elements to have child data elements (FeatureDataset, for instance). Lastly, 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.

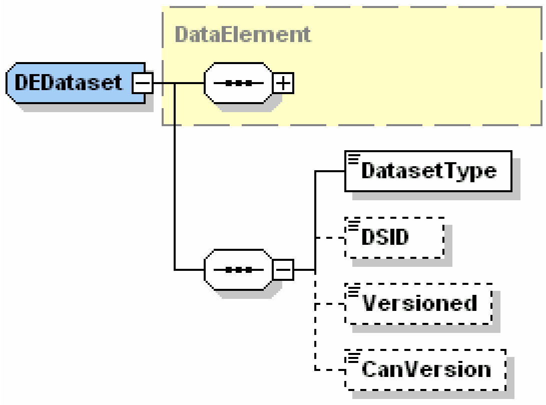


## Arrays

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 arrays 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).

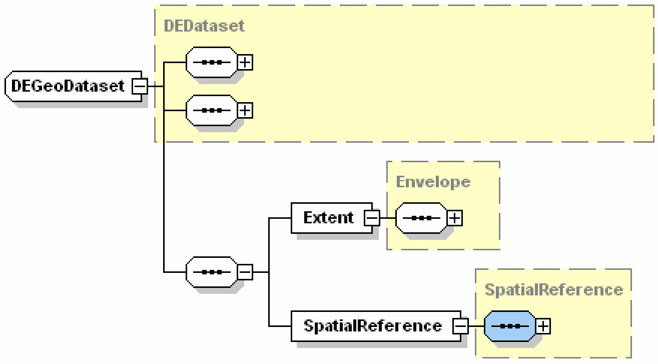
## DEDataset

A 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

A DEGeoDataset is yet 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.

Recall that a DEFeatureDataset is ultimately derived from DataElement. This means DEFeatureDataset has a Children element that can be used to contain an array of other DataElements. Since geometric networks, topologies, feature classes, and relationship classes are also derived from DataElement, they can appear in the Children array of a DEFeatureDataset.

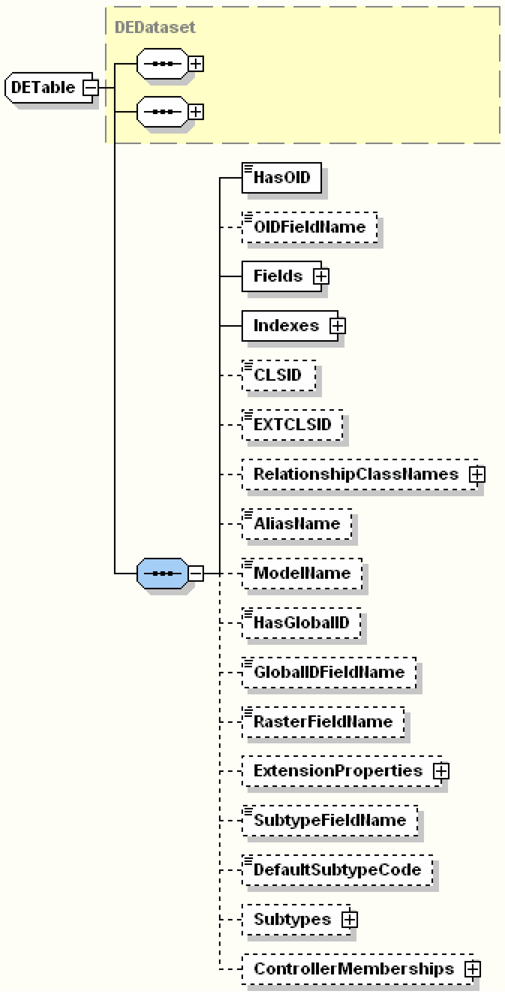
## DETable, DEFeatureClass, and DERasterCatalog

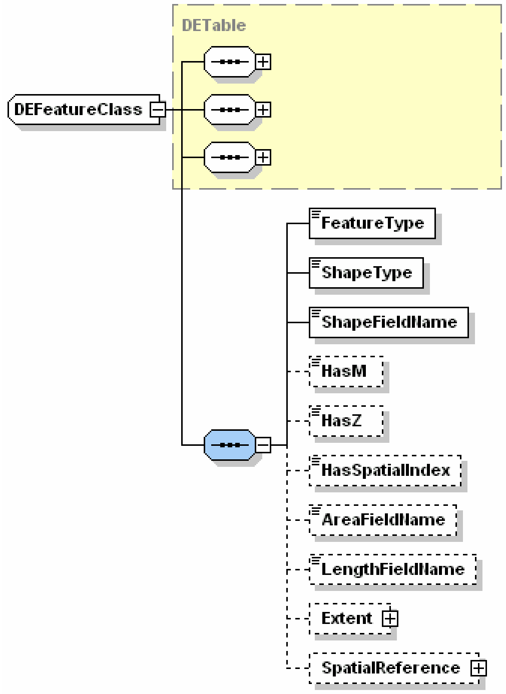
DETable is derived from DEDataset, adding a number of extra elements to the base type, including the name of the object ID field; arrays for fields, indexes, and subtypes; and the names of the relationship classes the table is involved in. In addition, the DETable type has optional elements to describe 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, the shape (geometry) type, the name of the field that contains the geometries, the extent, and the spatial reference of the feature class. Note that because it is derived from DETable, a DEFeatureClass can also reference the relationship classes it is involved in, or have subtypes.[1]





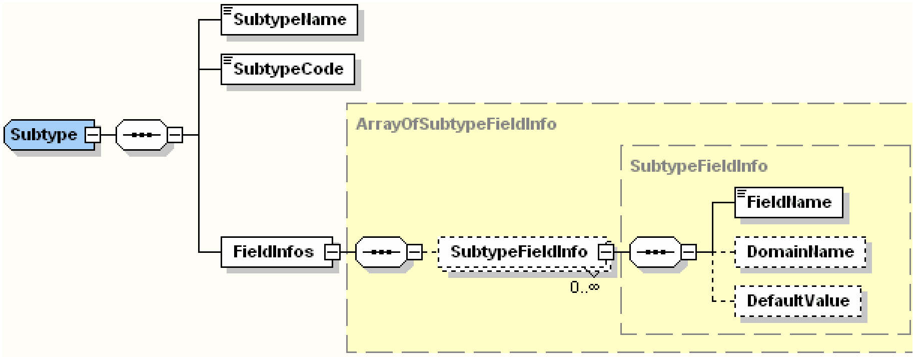
The DERasterCatalog XML type is derived from DEFeatureClass but doesn't add any extra elements. The RasterFieldName element of DETable should have the name of the field that contains the rasters. The actual rasters in the catalog are exported to XML as values of a record in the RecordSet of the raster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

[1] 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.

## Subtype

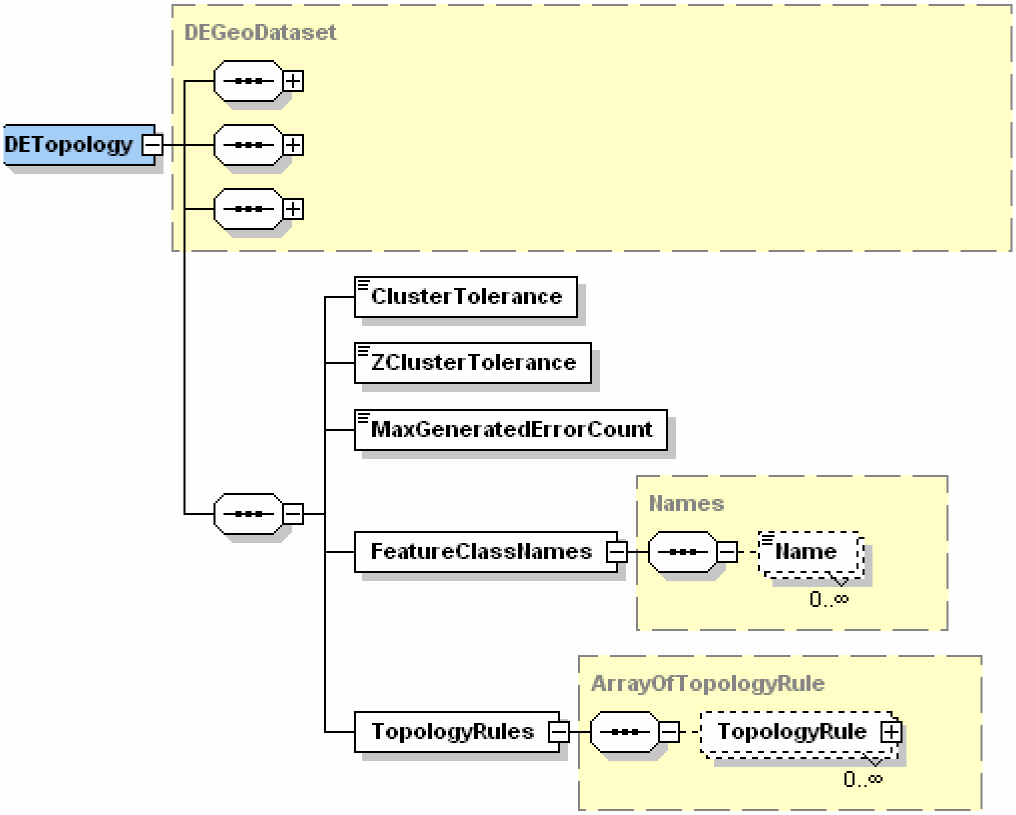
The array of subtypes in the table data element may have zero to many elements of type Subtype. This type has elements for the name and code of the subtype and an array of SubtypeFieldInfo elements.

In turn, the SubtypeFieldInfo type has elements for the FieldName, the DefaultValue, and the 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 Domain are optional.



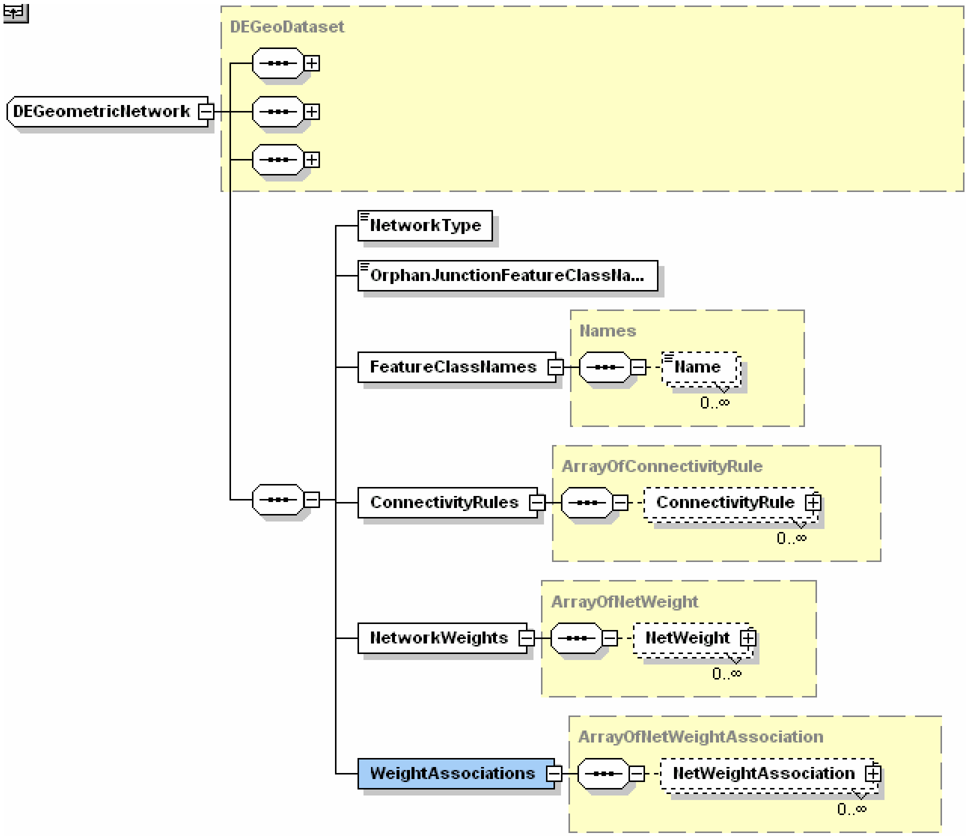
## DETopology

The topology data element 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

Like the topology data element, the geometric network data element is a DEGeoDataset. 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, the network weights and weight associations, and the connectivity rules.

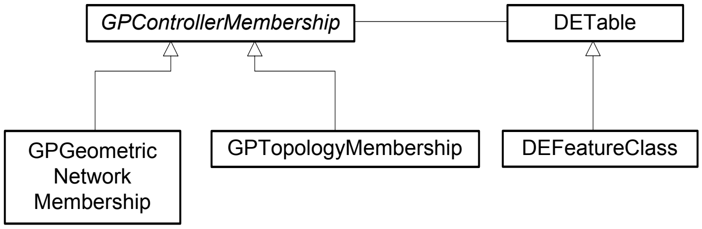


## Controllers and memberships

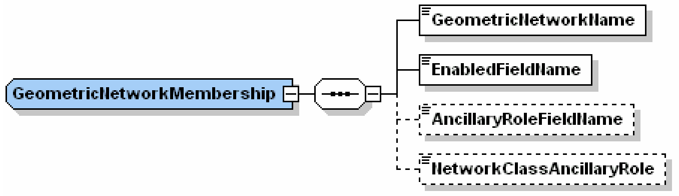
The association between a feature class and a topology has extra attributes—for example, the rank of the feature class. The association of a feature class and a geometric network also has extra attributes—for example, the enabled field. These are attributes of the feature class's membership in a controlling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

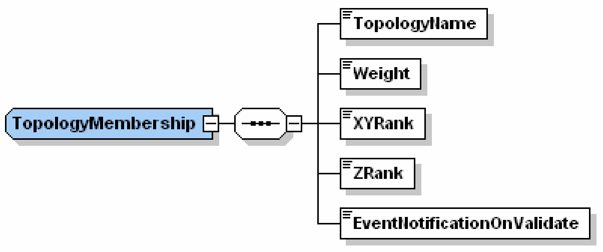
The table data element has an array of controller memberships. Each membership in the array contains the data about the relationship between the feature class and the controlling entity (rank or enabled field, for example)[1]. The following diagram shows the logical association between table and controller memberships.

[1] 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.



The 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 geometric network and topology. Their layout is shown below.



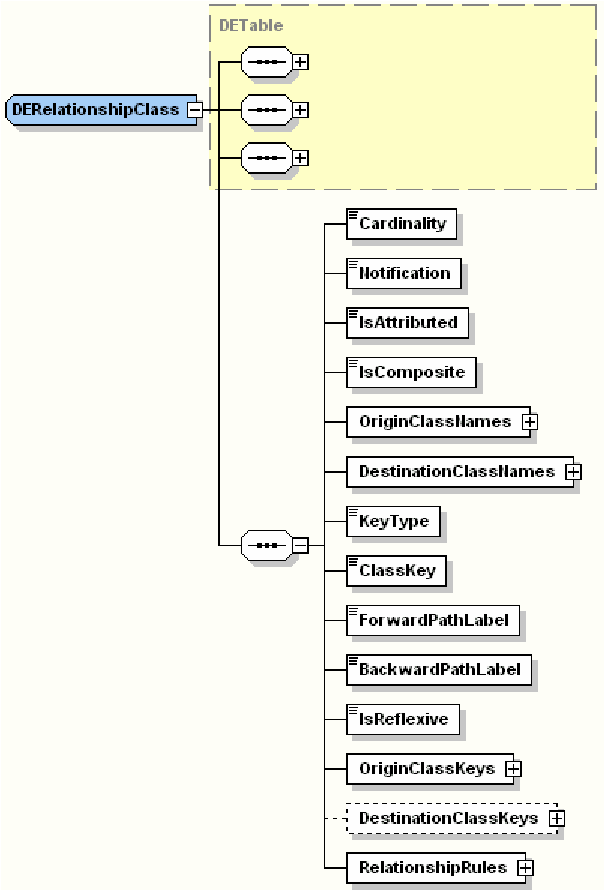


## DERelationshipClass

The elements in the relationship class data element type allow for 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.

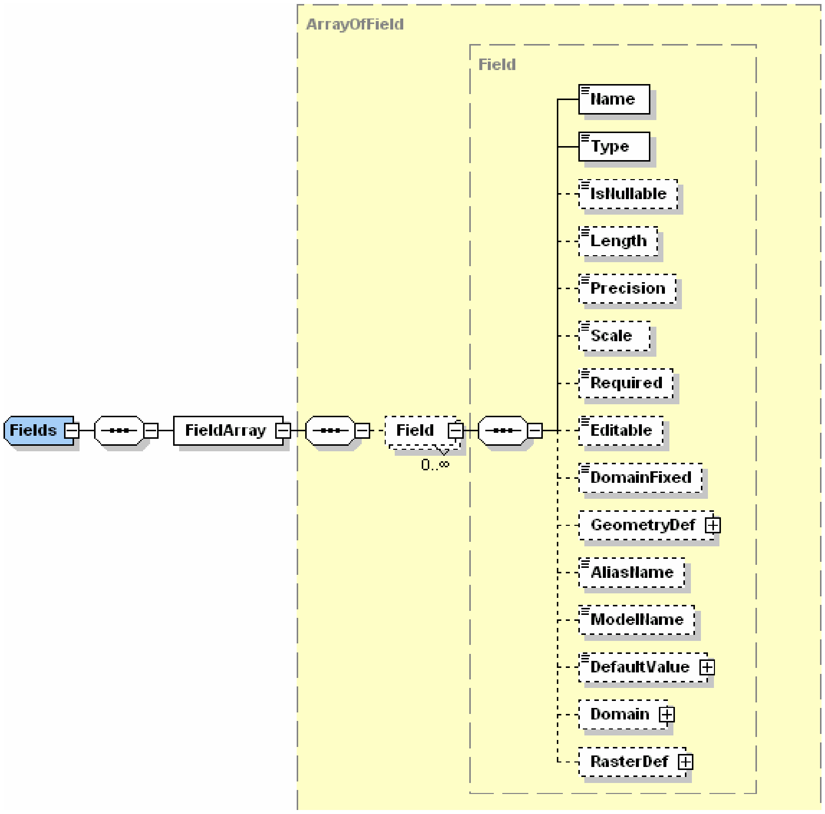


## Field and Fields

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.

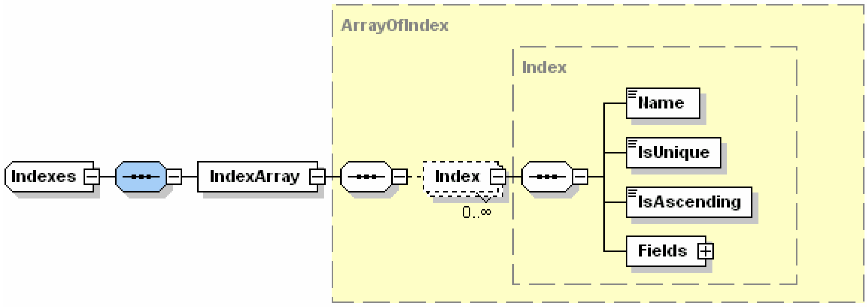
If the field type is geometry then the element GeometryDef should be present and describe the average number of points, geometry type, grid sizes, spatial reference, and whether vertices of the feature will contain measures or elevations. If the field type is raster then the element RasterDef should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

As the 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).



## Index and Indexes

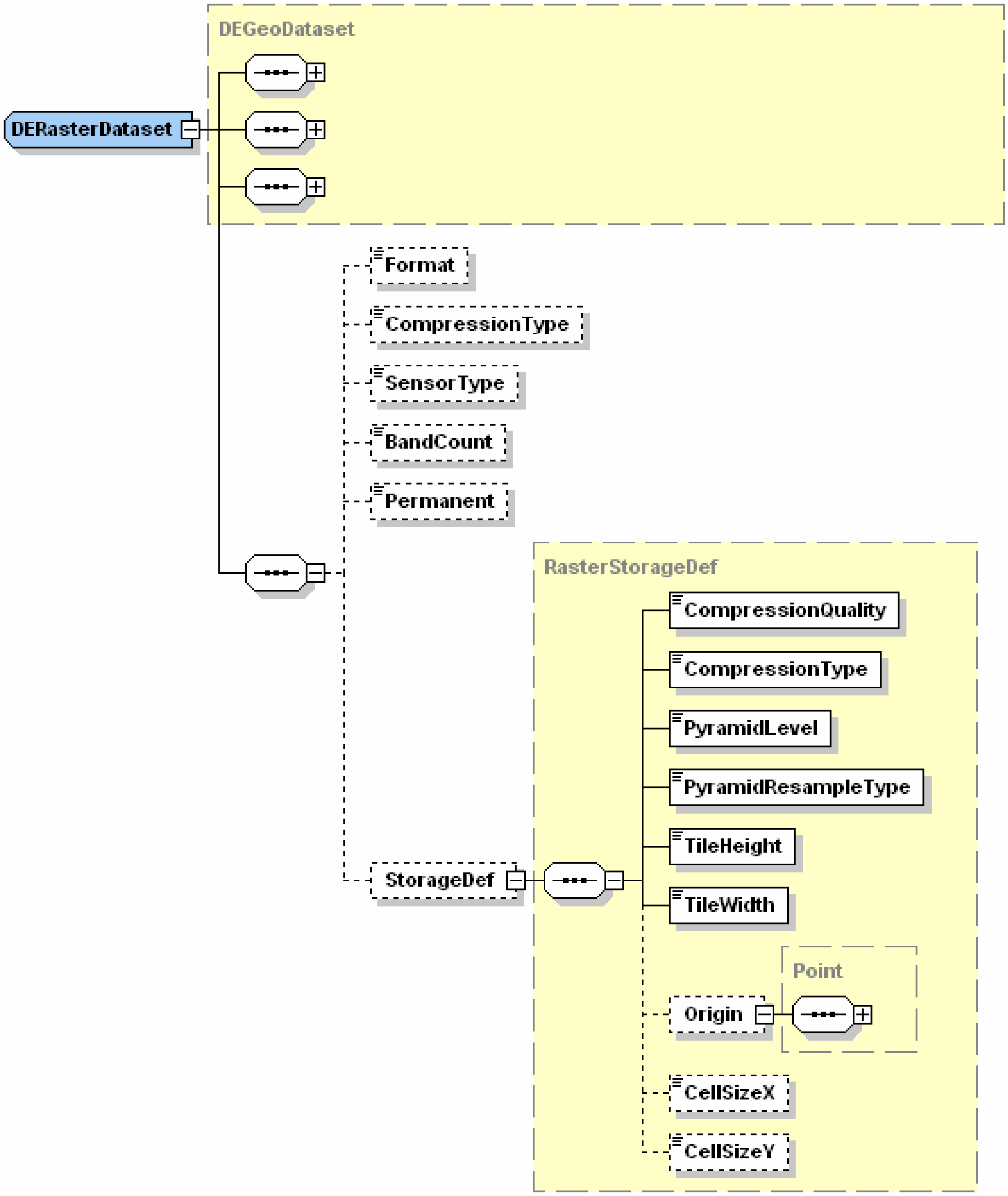
The Index XML type describes a geodatabase index. It has elements for the index name, whether the index is unique or ascending, and the set of fields that makes up the index. Similarly to the Fields case, there are types for IndexesArray and Indexes.



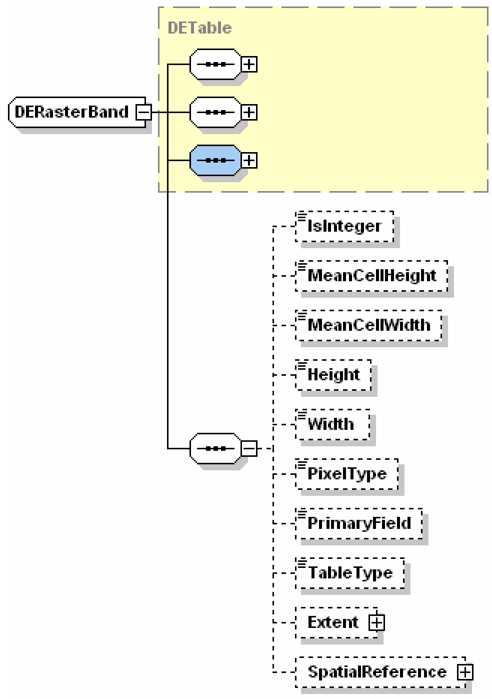
## DERasterDataset and DERasterBand

The DERasterDataset type is derived from DEGeoDataset. Serialized properties for a rater dataset include its Format, CompresionType, SensorType, 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 DERasterDataset's Children element will contain a collection of DERasterBands.

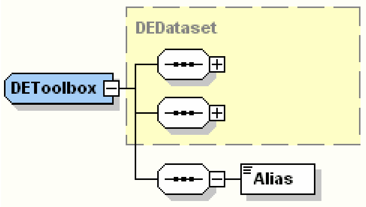


Because a raster band may have an associated table, the DERasterBand XML type is derived from DETable. It holds information such as the type of pixel (for example, integer or float), number of rows and columns (width and height), cell size, field in the attribute table whose values are stored in the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



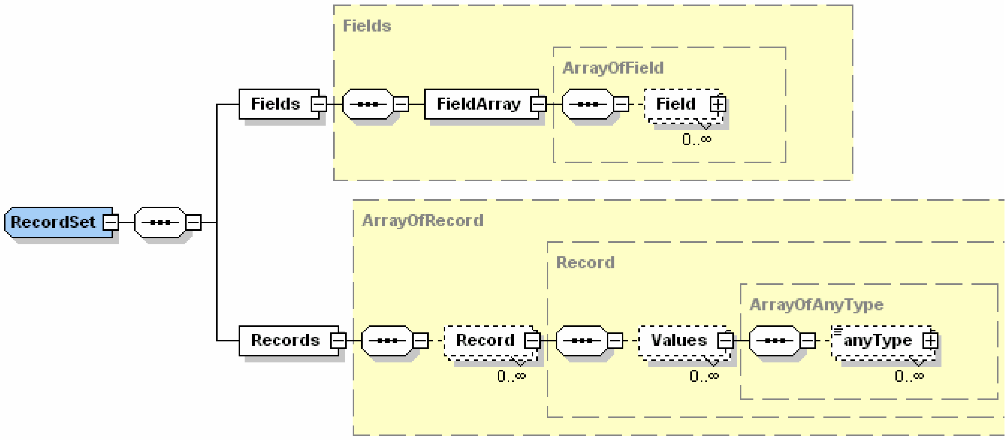
## DEToolbox

Toolboxes are represented using the DEToolbox type. It derives from DEDataset adding the Alias element.



## RecordSet

A RecordSet represents 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 RecordSet. 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 RecordSet looks like. In this case, the RecordSet 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 RecordSets 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 RecordSet will serialize its values using the type XMLPersistedObject, which is explained later.

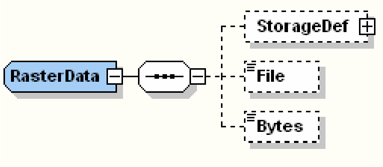
## RasterData

To prevent the creation of very large XML files when raster columns are present, ArcGIS exports such rasters into separate files. All the 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 RecordSet 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 picture 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 their binary form, geometry is exported using base64 encoding. This is the same point in binary form:

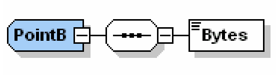
```
<Value xsi:type="esri:Point">
   <Bytes>AQAAAAAcCMkCtiVAZHWr5/R7oEA=</Bytes>
</Value>
```

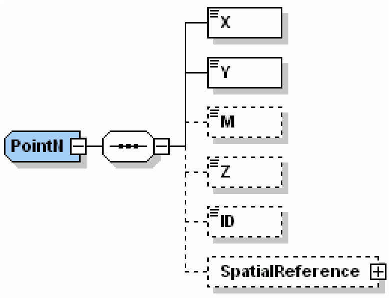
Internally the exporter will invoke the ExportToESRIShape method on the ArcObjects geometry, then encode the resulting byte stream in base64 format. Conversely, the importer will decode the base64 string into a byte stream, then invoke ImportFromESRIShape to re-create the geometry.

In normalized form, polygons and polylines are represented by much longer XML. 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.

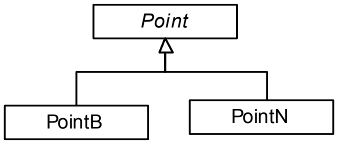
Binary serialization produces smaller documents and performs better than normalized serialization. On the other hand, a third party tool using an XML parser can interpret normalized geometries easily without requiring ArcObjects.

Because a geometry may be represented in binary or normalized form, there are two XML types for each geometry. They are identified with the suffix 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, we defined a base abstract type and have the concrete types derive from it. The diagram below shows the relationship among Point (abstract), PointN, and PointB.

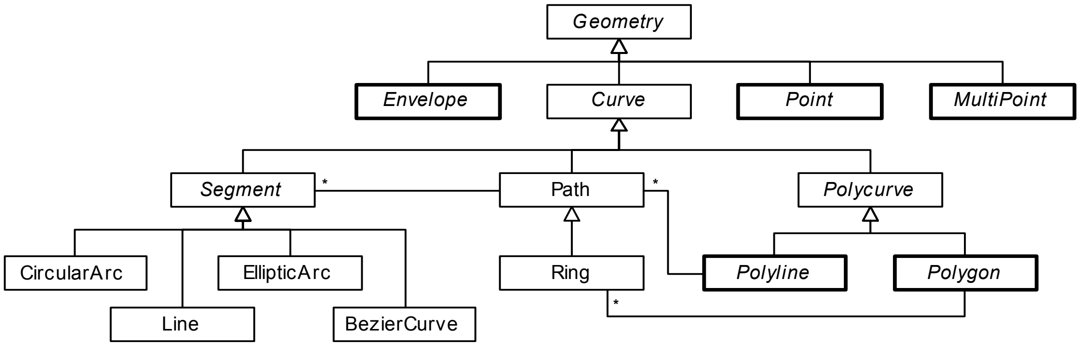


This separation of XML types is followed by all “top-level” geometries, that is Point, Polyline, Polygon, Multipoint, and Multipatch. Envelope also follows the same pattern. For all of them the binary XML type only contains the Bytes element.

EllipticArc, Line, BezierCurve, Path, and Ring 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 the features share the same spatial reference. To avoid serializing the same information over and over again, the spatial reference is not written when a geometry is serialized inside a RecordSet. However, when used in other contexts, such as an argument for a Web service method, an individual geometry will contain the spatial reference.

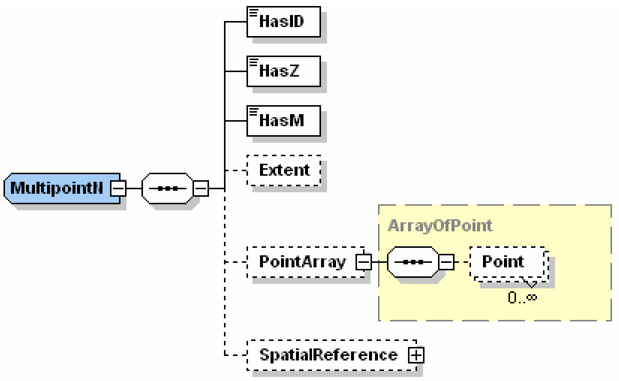
The XML types for geometry form a hierarchy that mirrors that of the geometry classes in ArcObjects. Geometry, the root type, is an abstract XML type with no elements. For simplicity, normalized and binary types have been excluded from the diagram, but they exist for the highlighted geometries.



## Point and Multipoint

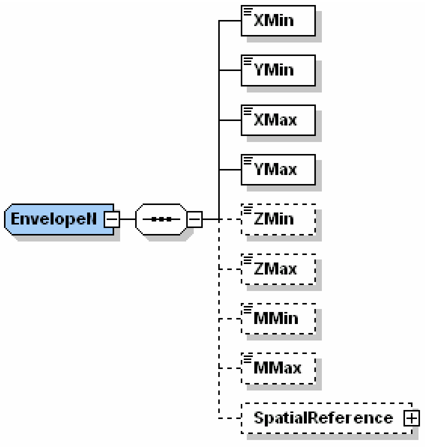
A point is a zero-dimensional geometry. It is defined with an x,y coordinate pair and, optionally, an elevation, measure, and point ID.

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



## Envelope

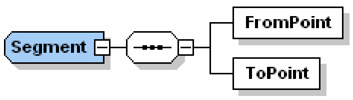
An envelope is a rectangular bounding box. The XML type has elements for the coordinates defining the envelope.



## Segment, Line, CircularArc, and EllipticArc

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

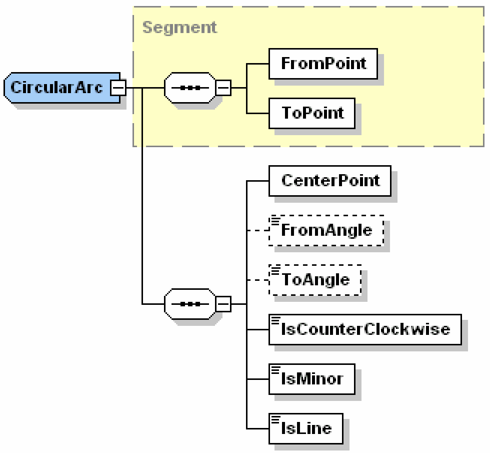
A segment is a parametric curve or function between two endpoints. For example, a line is a linear segment between the points, while a circular arc is a part of a circle between the points. The abstract XML type for segment has elements for the two endpoints.

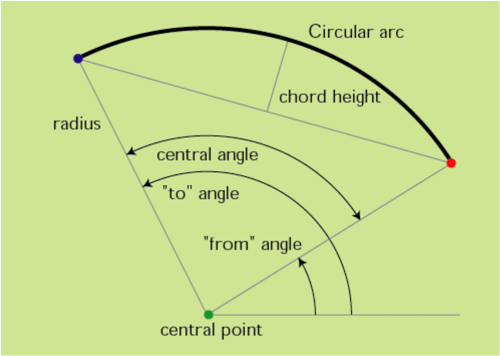


A line is a straight segment between two points. The XML type for a line is derived from a segment and adds no extra elements.

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 half 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.[1]

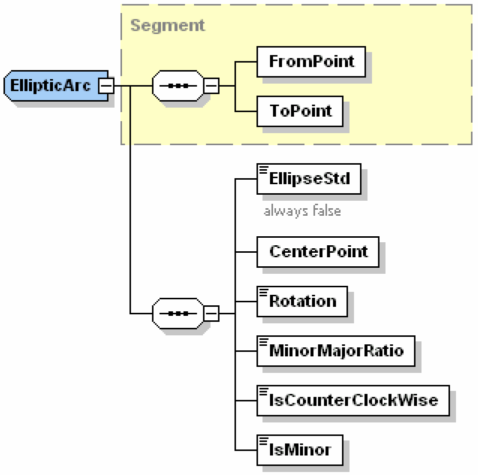


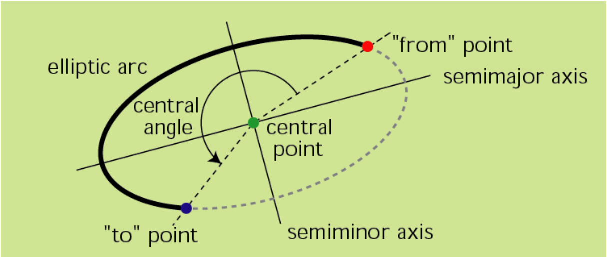


An elliptic arc is a segment that describes an ellipse or a part of it. The XML type describes the ellipse using the endpoints and a central point. If EllipseStd is false the coordinates are measured in a Cartesian coordinate system. If, on the other hand, EllipseStd is true, then the FromPoint and ToPoint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

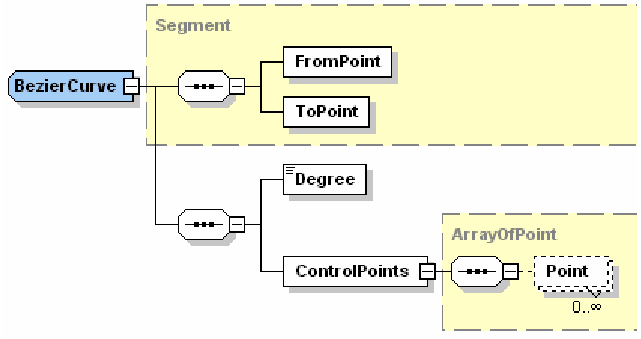
The rotation element contains the angle of the major axis measured counterclockwise from the x-axis. The MinorMajorRatio contains the ratio between the axes, and IsCounterClockwise describes how to rotate about the arc's center from the 'from' point to the 'to' point of an arc. Finally, the IsMinor element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

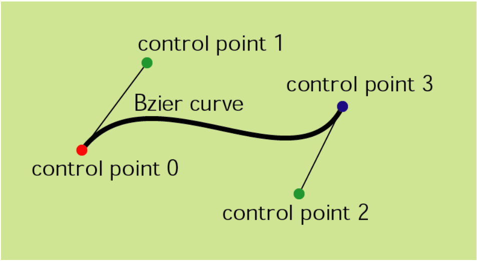
[1] Assume you have a degenerated circular where the to, from, and center points coincide. The only way to "un-degenerate" it by providing a radius and a central angle is to have previously set the FromAngle and ToAngle.





The last parametric segment is the Bezier curve. Such a curve is defined with two endpoints and an extra two 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 3 and the array of points must have exactly two points.

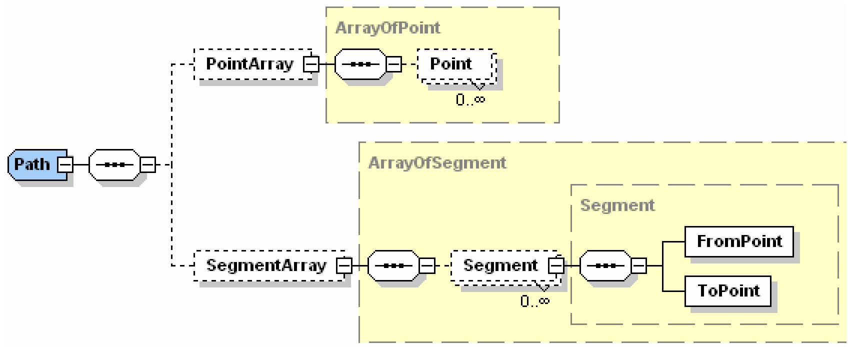




## Path and Ring

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.

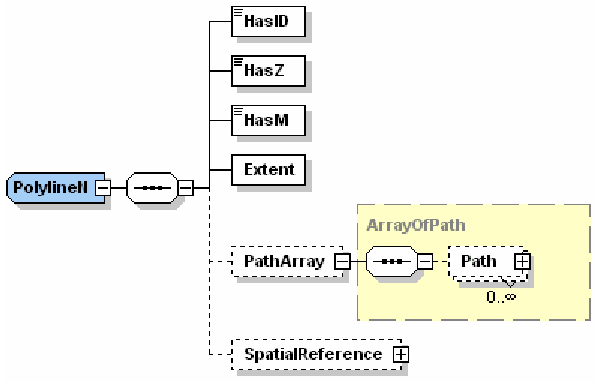
A ring is a closed path. The Ring XML type is derived from a path but doesn’t add any extra elements; therefore, paths and rings are represented by the same diagram.



## 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 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.



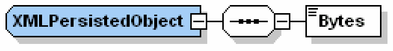
A polygon is a collection of rings. The XML type for polygon is identical to the polyline type above, except the PathArray is replaced by a RingArray element. RingArray contains rings that form the polygon.

More information about the geometry classes in ArcObjects can be found in *Exploring ArcObjects* [3].

## XMLPersistedObject

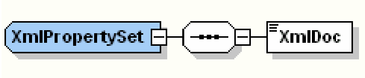
Almost all the XML types described so far have a corresponding class in ArcObjects. In fact, the XML documents generated by ArcGIS are the result of asking each object to serialize itself into XML. Some objects, however, may not have this capability, or the data itself may be difficult to represent using 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, that will contain the XML document representing the metadata. Metadata of tables, feature classes, topologies, and other geodatabase datasets will be represented in this manner.



## References

[1] OpenGIS® Geography Markup Language (GML) Implementation Specification. 29 January 2003. http://www.opengis.org/techno/implementation.htm
[2] XML Schema Part 0: Primer. W3C Recommendation, 2 May 2001. http://www.w3.org/TR/xmlschema-0/
[3] Exploring ArcObjects, 2001.

# Appendix A: Binary versus normalized serialization

This sample shows a polygon being serialized in binary form.

```
<Value xsi:type="esri:Polygon">
  <Bytes>
DwAAAADjGTT0121AQH5v05+5kkAgufyHtCSCQNCWcymuYZdAAQAAAAcAAAAAAAAAAYP
iImJJUeUDQ
lnMprmGXQCC5/Ie0JIJAMI/8wWCylkDAndLB+n+AQAiPNo54pJRAIEHxY0z5ekBQhA
1Pr9ySQKDT
SEtlZnRAQH5v05+5kkAA4xk09NdtQAiPNo54pJRAYPiImJJUeUDQlnMprmGXQAAAAA
AAAPA/AAAA
AAAAGEAAAAAAAADwPwAAAAAAAABAAAAAAAAACEAAAAAAAAAQQAAAAAAAAABRAAAAAAA
AAGEAAAAAA
AADwPwAAAAAAACRAAAAAAAAATkAAAAAAAAAkQAAAAAAAAADRAAAAAAAAAPkAAAAAAAA
BEQAAAAAAA
AElAAAAAAAAATkAAAAAAAAAkQA==
  </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>esriDTTable</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>AQAAAI1AAGAAtEFB4zoAoMdKJkE=</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>
    <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>
          <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>
```