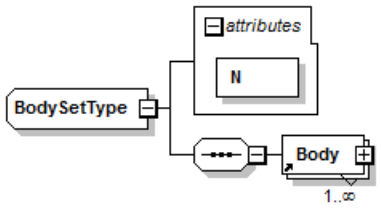
Data dictionary for QIF Library Topology.xsd (normative)

schema location: **..\QIFLibrary\Topology.xsd**
attributeFormDefault: **unqualified**
elementFormDefault: **qualified**
targetNamespace: **http://qifstandards.org/xsd/qif2**

Complex types

Simple types

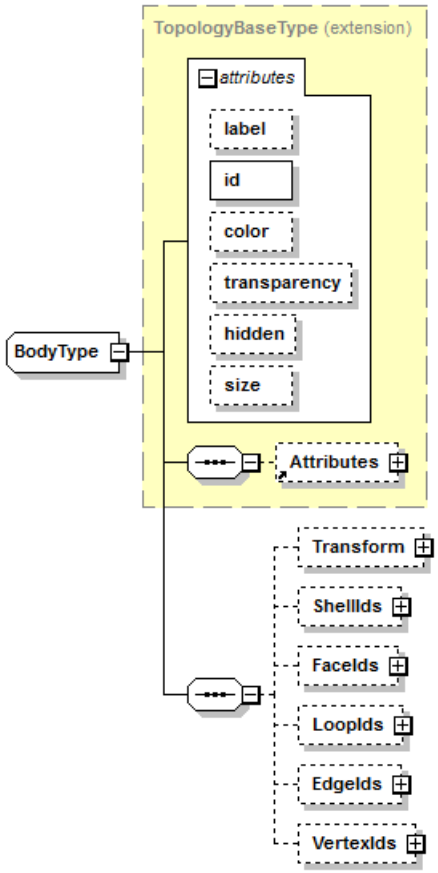
complexType **BodySetType**

diagram	
children	**Body**
used by	element **BodySet**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute is the number of bodies in this set.
annotation	documentation: The BodySetType represents a container for storing all model bodies.

attribute **BodySetType/@N**

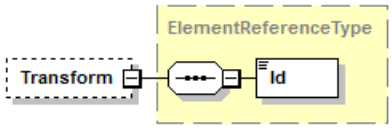
type	**NaturalType**
properties	use required
facets	Kind Value Annotation minInclusive 1
annotation	documentation The required N attribute is the number of bodies in this set.

complexType **BodyType**

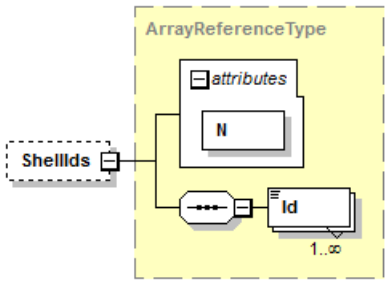
diagram	
type	extension of **TopologyBaseType**
properties	base TopologyBaseType
children	**Attributes** **Transform** **ShellIds** **FaceIds** **LoopIds** **EdgeIds** **VertexIds**
used by	element **Body**
attributes	Name Type Use Default Fixed Annotation label **xs:string** documentation The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree. id **QIFIdType** required documentation The required id attribute is the unique model entity identifier.

	color	**ColorType**		documentation The optional color attribute defines the RGB color property of a model entity.
	transparency	**xs:double**	0.0	documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**	0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**		documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
annotation	documentation The BodyType is the b-rep body type - a solid represented as a set of one outer and zero to many inner shells.			

element **BodyType/Transform**

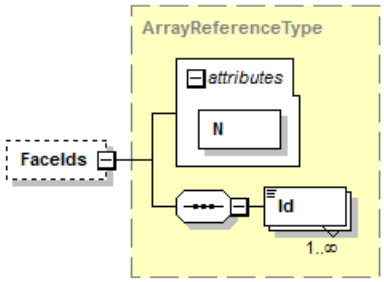
diagram	
type	**ElementReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
annotation	documentation The optional Transform element is an identifier of the transformation matrix.

element **BodyType/ShellIds**

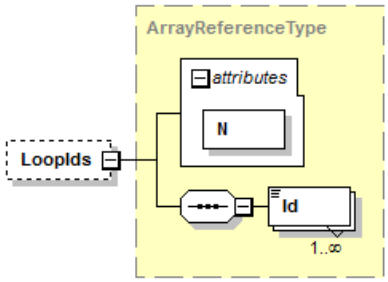
diagram	
type	**ArrayReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation The required N attribute shows how many Id elements are

	present in this array.
annotation	documentation The optional ShellIds element is an array of shell identifiers participating in forming this body.

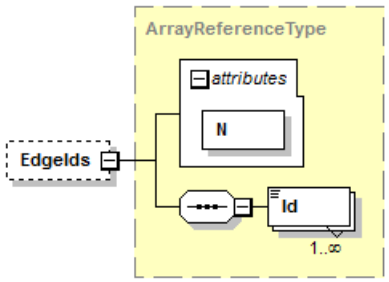
element **BodyType/FaceIds**

diagram	
type	**ArrayReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute shows how many Id elements are present in this array.
annotation	documentation The optional FaceIds element is an array of face identifiers participating in forming this body.

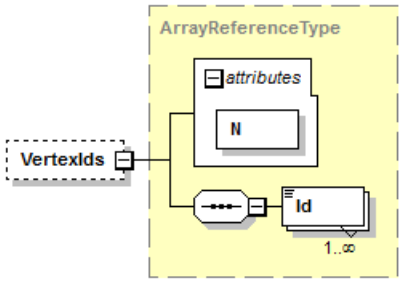
element **BodyType/LoopIds**

diagram	
type	**ArrayReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute shows how many Id elements are present in this array.
annotation	documentation The optional LoopIds element is an array of loop identifiers participating in forming this body.

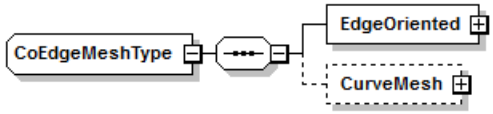
element **BodyType/EdgeIds**

diagram	
type	**ArrayReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute shows how many Id elements are present in this array.
annotation	documentation The optional EdgeIds element is an array of edge identifiers participating in forming this body.

element **BodyType/VertexIds**

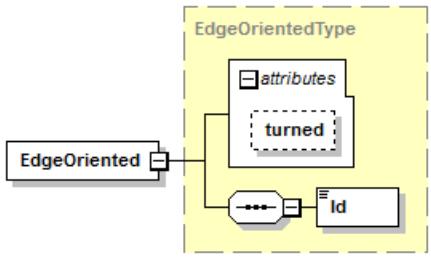
diagram	
type	**ArrayReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute shows how many Id elements are present in this array.
annotation	documentation The optional VertexIds element is an array of vertex identifiers participating in forming this body.

complexType **CoEdgeMeshType**

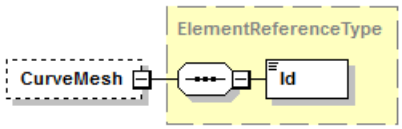
diagram	

children	**EdgeOriented** **CurveMesh**
used by	element **CoEdgesMeshType/CoEdgeMesh**
annotation	documentation The CoEdgeMeshType represents the part of mesh face trimming loop. A mesh co-edge defines a mesh curve (i.e. the projection of an underlying 3D Curve of an oriented Edge onto an underlying mesh surface of a mesh face).

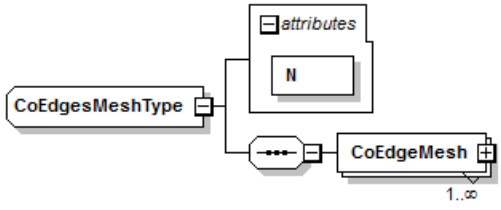
element **CoEdgeMeshType/EdgeOriented**

diagram	
type	**EdgeOrientedType**
properties	content complex
children	**Id**
attributes	Name: turned; Type: **xs:boolean**; Use: ; Default: 0; Fixed: ; Annotation: documentation: The optional turned attribute shows if the referenced edge must be reversed from the origin edge orientation.
annotation	documentation The EdgeOriented element is a reference to edge with a given orientation.

element **CoEdgeMeshType/CurveMesh**

diagram	
type	**ElementReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
annotation	documentation The optional CurveMesh element is a reference to mesh curve. This is projection of underlying 3D Curve of an oriented Edge onto an underlying mesh surface of a mesh face.

complexType **CoEdgesMeshType**

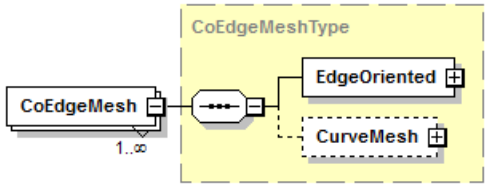
diagram	

children	**CoEdgeMesh**
used by	element **LoopMeshType/CoEdgesMesh**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute shows how many objects are presented in this array.
annotation	documentation: The CoEdgesMeshType is an array of mesh co-edges.

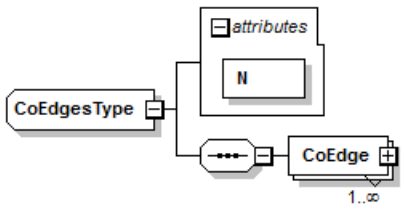
attribute **CoEdgesMeshType/@N**

type	**NaturalType**
properties	use required
facets	Kind: minInclusive; Value: 1; Annotation:
annotation	documentation: The required N attribute shows how many objects are presented in this array.

element **CoEdgesMeshType/CoEdgeMesh**

diagram	
type	**CoEdgeMeshType**
properties	minOcc 1 maxOcc unbounded content complex
children	**EdgeOriented** **CurveMesh**
annotation	documentation: Each CoEdgeMesh element is a mesh co-edge.

complexType **CoEdgesType**

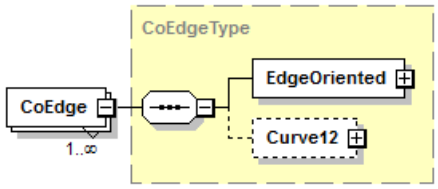
diagram	
children	**CoEdge**
used by	element **LoopType/CoEdges**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute

	shows how many objects are presented in this array.
annotation	documentation The CoEdgesType is an array of co-edges.

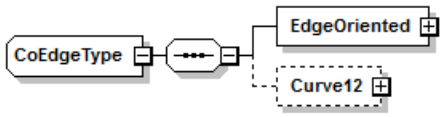
attribute **CoEdgesType/@N**

type	**NaturalType**
properties	use required
facets	Kind Value Annotation minInclusive 1
annotation	documentation The required N attribute shows how many objects are presented in this array.

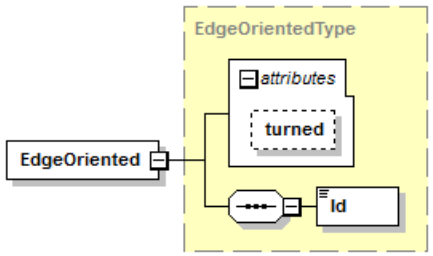
element **CoEdgesType/CoEdge**

diagram	
type	**CoEdgeType**
properties	minOcc 1 maxOcc unbounded content complex
children	**EdgeOriented** **Curve12**
annotation	documentation Each CoEdge element is a co-edge.

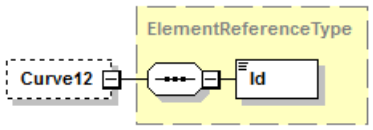
complexType **CoEdgeType**

diagram	
children	**EdgeOriented** **Curve12**
used by	element **CoEdgesType/CoEdge**
annotation	documentation The CoEdgeType represents the part of face trimming loop. A co-edge defines a parameter space curve (i.e. the projection of an underlying 3D Curve of an oriented Edge onto an underlying surface of a face).

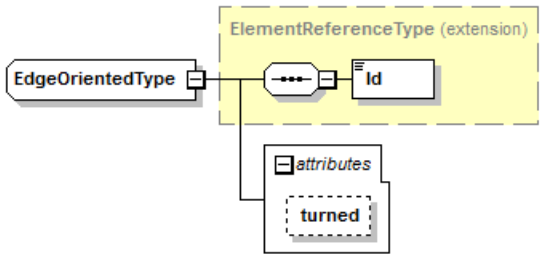
element **CoEdgeType/EdgeOriented**

diagram	
type	**EdgeOrientedType**
properties	content complex
children	**Id**
attributes	Name: turned; Type: **xs:boolean**; Use: ; Default: 0; Fixed: ; Annotation: documentation: The optional turned attribute shows if the referenced edge must be reversed from the origin edge orientation.
annotation	documentation The EdgeOriented element is a reference to edge with a given orientation.

element **CoEdgeType/Curve12**

diagram	
type	**ElementReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
annotation	documentation The optional Curve12 element is a reference to 2D curve. This is projection of underlying 3D Curve of an oriented Edge onto an underlying surface of a face.

complexType **EdgeOrientedType**

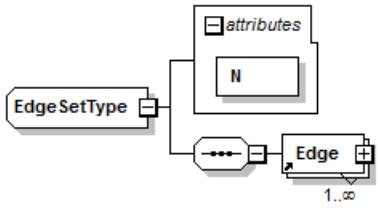
diagram	

type	extension of **ElementReferenceType**
properties	base ElementReferenceType
children	**Id**
used by	elements **CoEdgeType/EdgeOriented** **CoEdgeMeshType/EdgeOriented**
attributes	Name: turned; Type: **xs:boolean**; Use: ; Default: 0; Fixed: ; Annotation documentation: The optional turned attribute shows if the referenced edge must be reversed from the origin edge orientation.
annotation	documentation: The EdgeOrientedType is a reference to edge with a given orientation.

attribute **EdgeOrientedType/@turned**

type	**xs:boolean**
properties	default 0
annotation	documentation: The optional turned attribute shows if the referenced edge must be reversed from the origin edge orientation.

complexType **EdgeSetType**

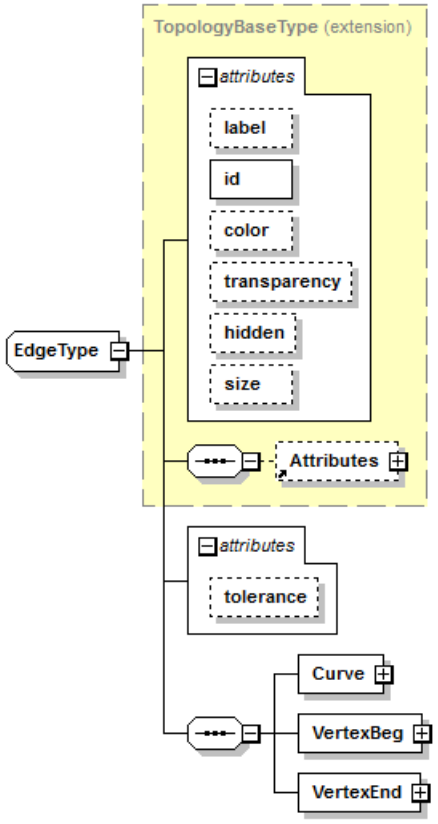
diagram	
children	**Edge**
used by	element **EdgeSet**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute is the number of edges in this set.
annotation	documentation: The EdgeSetType represents a container for storing all model edges.

attribute **EdgeSetType/@N**

type	**NaturalType**
properties	use required
facets	Kind: minInclusive; Value: 1; Annotation:

annotation	documentation The required N attribute is the number of edges in this set.

complexType **EdgeType**

diagram	
type	extension of **TopologyBaseType**
properties	base TopologyBaseType
children	**Attributes** **Curve** **VertexBeg** **VertexEnd**
used by	element **Edge**

attributes

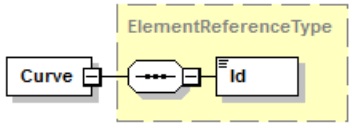
Name	Type	Use	Default	Fixed	Annotation
label	**xs:string**				documentation The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree.
id	**QIFIdType**	required			documentation The required id attribute is the unique model entity identifier.
color	**ColorType**				documentation The optional color attribute

				defines the RGB color property of a model entity.
	transparency	**xs:double**	0.0	documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**	0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**		documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
	tolerance	**xs:double**		documentation The optional tolerance attribute specifies the edge tolerance for the case of tolerant body.
annotation	documentation The EdgeType is the b-rep edge type - a topological entity. It represents a bounded piece of a 3D curve.			

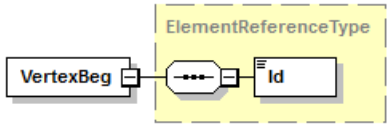
attribute **EdgeType/@tolerance**

type	**xs:double**
annotation	documentation The optional tolerance attribute specifies the edge tolerance for the case of tolerant body.

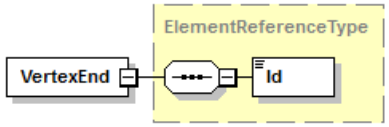
element **EdgeType/Curve**

diagram	
type	**ElementReferenceType**
properties	content complex
children	**Id**
annotation	documentation The Curve element is the identifier of a 3D curve that is the underlying geometry of this edge.

element **EdgeType/VertexBeg**

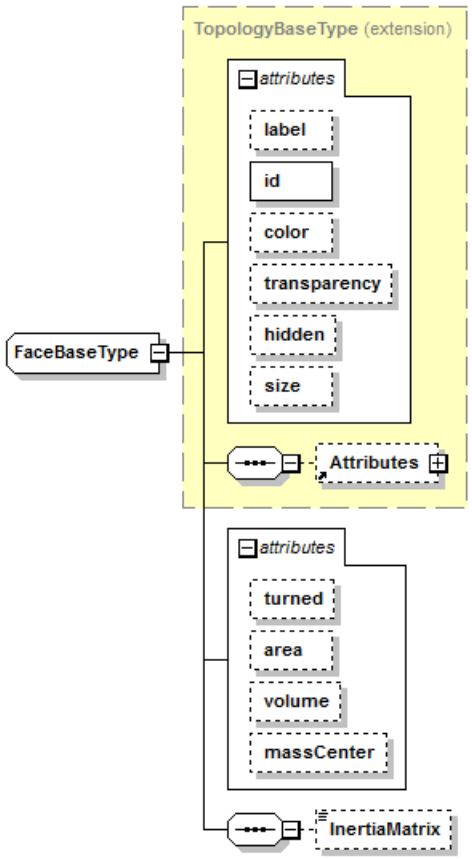
diagram	
type	**ElementReferenceType**
properties	content complex
children	**Id**
annotation	documentation The VertexBeg element is the identifier of the vertex which bounds this edge at the beginning of the edge. The 'underlying' parameter of VertexBeg must be less then the 'underlying' parameter of VertexEnd. Or, in other words, the edge always follows the natural parameterization of the underlying 3D curve. If there is a need to pass an edge in the opposite (to the natural parameterization of the underlying curve) direction then the corresponding flag must be defined on the loop level.

element **EdgeType/VertexEnd**

diagram	
type	**ElementReferenceType**
properties	content complex
children	**Id**
annotation	documentation The VertexEnd element is the identifier of the vertex which bounds this edge at the end of the edge. The 'underlying' parameter of VertexEnd must be bigger then the 'underlying' parameter of VertexBeg. Or, in other words, the edge always follows the natural parameterization of the underlying 3D curve. If there is a need to pass an edge in the opposite (to the natural parameterization of the underlying curve) direction then the corresponding flag must be defined on the loop level.

complexType **FaceBaseType**

diagram



type	extension of **TopologyBaseType**
properties	base TopologyBaseType abstract true
children	**Attributes** **InertiaMatrix**
used by	element **FaceBase** complexTypes **FaceMeshType** **FaceType**

attributes

Name	Type	Use	Default	Fixed	Annotation
label	**xs:string**				documentation The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree.
id	**QIFIdType**	required			documentation The required id attribute is the unique model entity identifier.
color	**ColorType**				documentation The optional color attribute defines the RGB color property of a model entity.

	transparency	**xs:double**	0.0	documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**	0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**		documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
	turned	**xs:boolean**	0	documentation The optional turned attribute shows if the face orientation must be reversed from the orientation of the underlying surface. If the value is true, the face orientation must be opposite the surface orientation. If the value is false, the two orientations must be the same.
	area	**xs:double**		documentation The optional area attribute is the face area - the face 'contribution' to its shell volume. This

	volume	**xs:double**	value is defined in the model units. documentation The volume attribute is the face partial volume - the face 'contribution' to its shell volume. It is calculated relative to the parent block origin. This value is defined in the model units.
	massCenter	**PointSimpleType**	documentation The optional massCenter attribute is the center of mass of the face.
annotation	documentation The FaceBaseType is the abstract base type for faces. A face represents a bounded portion of a geometric surface which can be defined as a parametric surface or a mesh surface.		

attribute **FaceBaseType/@turned**

type	**xs:boolean**
properties	default 0
annotation	documentation The optional turned attribute shows if the face orientation must be reversed from the orientation of the underlying surface. If the value is true, the face orientation must be opposite the surface orientation. If the value is false, the two orientations must be the same.

attribute **FaceBaseType/@area**

type	**xs:double**
annotation	documentation The optional area attribute is the face area - the face 'contribution' to its shell volume. This value is defined in the model units.

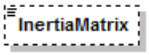
attribute **FaceBaseType/@volume**

type	**xs:double**
annotation	documentation The volume attribute is the face partial volume - the face 'contribution' to its shell volume. It is calculated relative to the parent block origin. This value is defined in the model units.

attribute **FaceBaseType/@massCenter**

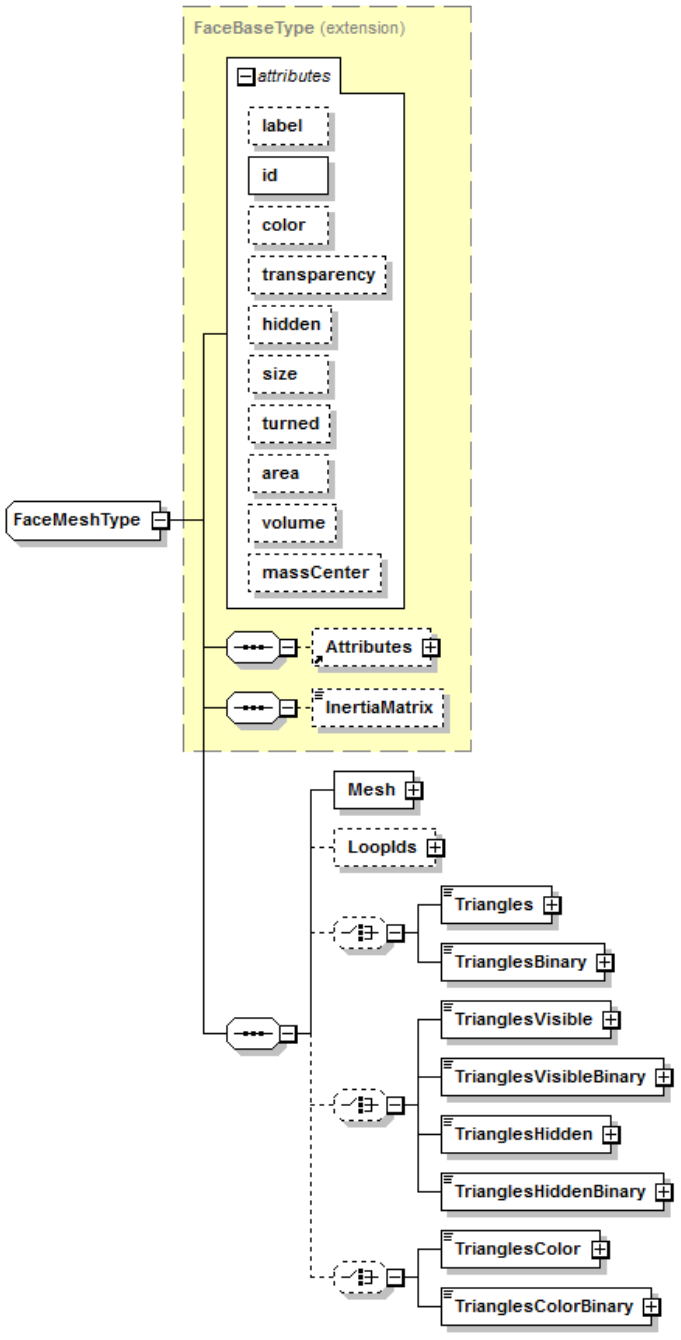
type	**PointSimpleType**
facets	Kind Value Annotation length 3
annotation	documentation The optional massCenter attribute is the center of mass of the face.

element **FaceBaseType/InertiaMatrix**

diagram	
type	**Matrix33Type**
properties	minOcc 0 maxOcc 1 content complex
facets	Kind Value Annotation length 9
annotation	documentation The optional InertiaMatrix element is the inertia tensor, which consists of the face moments of inertia about the three coordinate axes at the parent block origin.

complexType **FaceMeshType**

diagram



type	extension of **FaceBaseType**
properties	base FaceBaseType
children	**Attributes** **InertiaMatrix** **Mesh** **LoopIds** **Triangles** **TrianglesBinary** **TrianglesVisible** **TrianglesVisibleBinary** **TrianglesHidden** **TrianglesHiddenBinary** **TrianglesColor** **TrianglesColorBinary**
used by	element **FaceMesh**

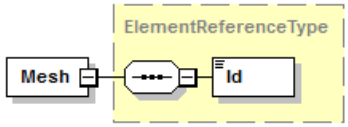
attributes

Name	Type	Use	Default	Fixed	Annotation
label	**xs:string**				documentation The optional label attribute is the model

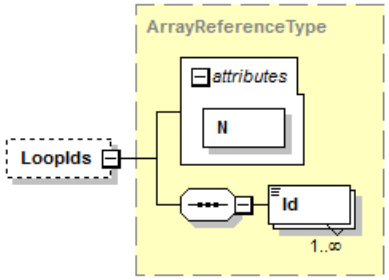
					entity "nameplate". Normally it can be seen at the entity item in the project tree.
	id	**QIFIdType**	required		documentation The required id attribute is the unique model entity identifier.
	color	**ColorType**			documentation The optional color attribute defines the RGB color property of a model entity.
	transparency	**xs:double**		0.0	documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**		0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**			documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
	turned	**xs:boolean**		0	documentation The optional turned attribute shows if the face orientation must be reversed from the orientation of the underlying surface. If the

			value is true, the face orientation must be opposite the surface orientation. If the value is false, the two orientations must be the same.
	area	**xs:double**	documentation The optional area attribute is the face area - the face 'contribution' to its shell volume. This value is defined in the model units.
	volume	**xs:double**	documentation The volume attribute is the face partial volume - the face 'contribution' to its shell volume. It is calculated relative to the parent block origin. This value is defined in the model units.
	massCenter	**PointSimpleType**	documentation The optional massCenter attribute is the center of mass of the face.
annotation	documentation The FaceMeshType is the b-rep mesh face type. It is built on a mesh surface bounded by a set of closed triangulation paths (polylines formed from the triangle edges).		

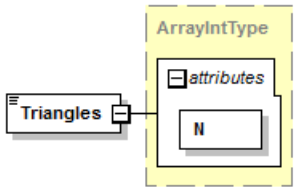
element **FaceMeshType/Mesh**

diagram	
type	**ElementReferenceType**
properties	content complex
children	**Id**
annotation	documentation The Mesh element is the identifier of the underlying mesh surface.

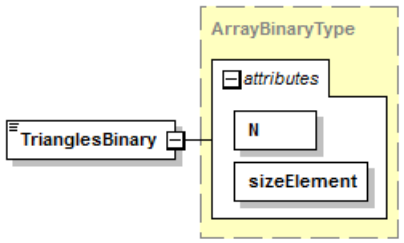
element **FaceMeshType/LoopIds**

diagram	
type	**ArrayReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute shows how many Id elements are present in this array.
annotation	documentation The optional LoopIds element is an array of identifiers of the face trimming contours. If the hasOuter attribute is true then the first id will point to the outer loop otherwise all ids will point to inner loops only.

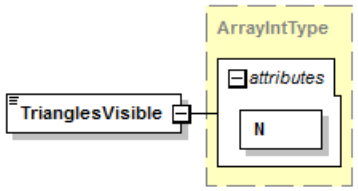
element **FaceMeshType/Triangles**

diagram	
type	**ArrayIntType**
properties	content complex
attributes	Name: N; Type: **xs:unsignedInt**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute shows how many objects are present in this array.
annotation	documentation The Triangles element is an array of triangle indexes of the underlying mesh surface. All elements of this array must be unique and must lie in the range [0, number of triangles in the underlying mesh surface - 1].

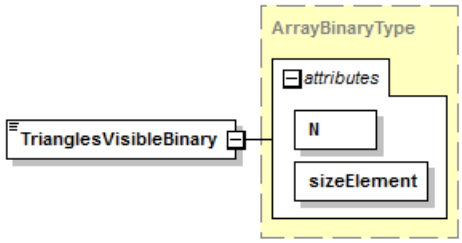
element **FaceMeshType/TrianglesBinary**

diagram	
type	**ArrayBinaryType**
properties	content complex
attributes	Name / Type / Use / Default / Fixed / Annotation N — **xs:unsignedInt** — required — documentation: The required N attribute shows how many elements are present in this array. sizeElement — **xs:unsignedInt** — required — documentation: The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.
annotation	documentation The TrianglesBinary element is a binary array of triangle indexes of the underlying mesh surface. All elements of this 32-bit integer array must be unique and must lie in the range [0, number of triangles in the underlying mesh surface - 1].

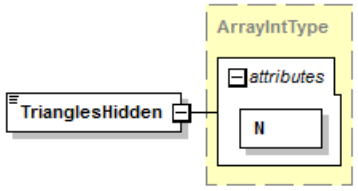
element **FaceMeshType/TrianglesVisible**

diagram	
type	**ArrayIntType**
properties	content complex
attributes	Name / Type / Use / Default / Fixed / Annotation N — **xs:unsignedInt** — required — documentation: The required N attribute shows how many objects are present in this array.
annotation	documentation The TrianglesVisible element is an array of visible triangle indices. All elements of this integer array must be unique and must lie in the range [0, number of triangles of the face interior triangles].

element **FaceMeshType/TrianglesVisibleBinary**

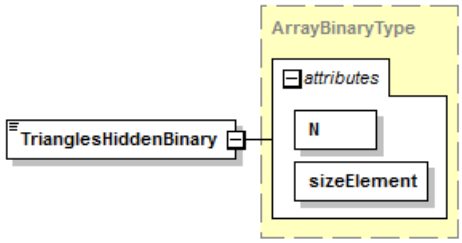
diagram	
type	**ArrayBinaryType**
properties	content complex
attributes	Name / Type / Use / Default / Fixed / Annotation N — **xs:unsignedInt** — required — documentation: The required N attribute shows how many elements are present in this array. sizeElement — **xs:unsignedInt** — required — documentation: The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.
annotation	documentation The TrianglesVisibleBinary element is a binary array of visible triangle indices. All elements of this integer array must be unique and must lie in the range [0, number of triangles of the face interior triangles].

element **FaceMeshType/TrianglesHidden**

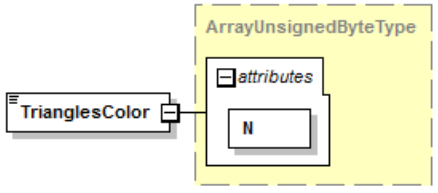
diagram	
type	**ArrayIntType**
properties	content complex
attributes	Name / Type / Use / Default / Fixed / Annotation N — **xs:unsignedInt** — required — documentation: The required N attribute shows how many objects are present in this array.

annotation	documentation The TrianglesHidden element is an array of hidden triangle indices. All elements of this integer array must be unique and must lie in the range [0, number of triangles of the face interior triangles].

element **FaceMeshType/TrianglesHiddenBinary**

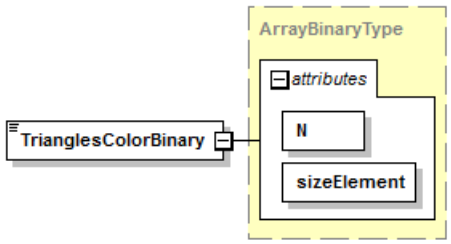
diagram	
type	**ArrayBinaryType**
properties	content complex
attributes	Name / Type / Use / Default / Fixed / Annotation N — **xs:unsignedInt** — required — documentation: The required N attribute shows how many elements are present in this array. sizeElement — **xs:unsignedInt** — required — documentation: The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.
annotation	documentation The TrianglesHiddenBinary element is a binary array of hidden triangle indices. All elements of this integer array must be unique and must lie in the range [0, number of triangles of the face interior triangles].

element **FaceMeshType/TrianglesColor**

diagram	
type	**ArrayUnsignedByteType**
properties	content complex
attributes	Name / Type / Use / Default / Fixed / Annotation N — **xs:unsignedInt** — required — documentation: The required N attribute shows how

	many objects are present in this array.
annotation	documentation The TrianglesColor element is an array of unsigned byte values which defines colors of the face interior triangles. Each element of this array is a triplet of unsigned byte numbers - the RGB color: the red-component, the green-component and the blue-component. The number of array elements corresponds to the number of triangles in the face interior.

element **FaceMeshType/TrianglesColorBinary**

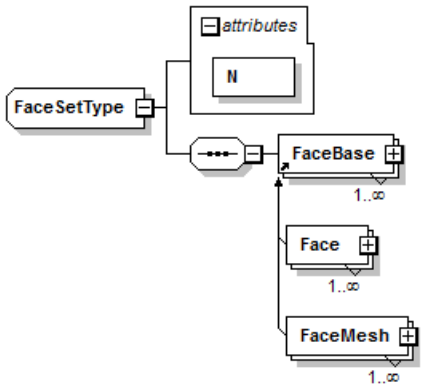
diagram	
type	**ArrayBinaryType**
properties	content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
N	**xs:unsignedInt**	required			documentation The required N attribute shows how many elements are present in this array.
sizeElement	**xs:unsignedInt**	required			documentation The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.

annotation	documentation The TrianglesColorBinary element is a binary array of unsigned byte values which defines colors of the face interior triangles. Each element of this array is a triplet of unsigned byte numbers - the RGB color: the red-component, the green-component and the blue-component. The number of array elements corresponds to the number of triangles in the face interior.

complexType **FaceSetType**

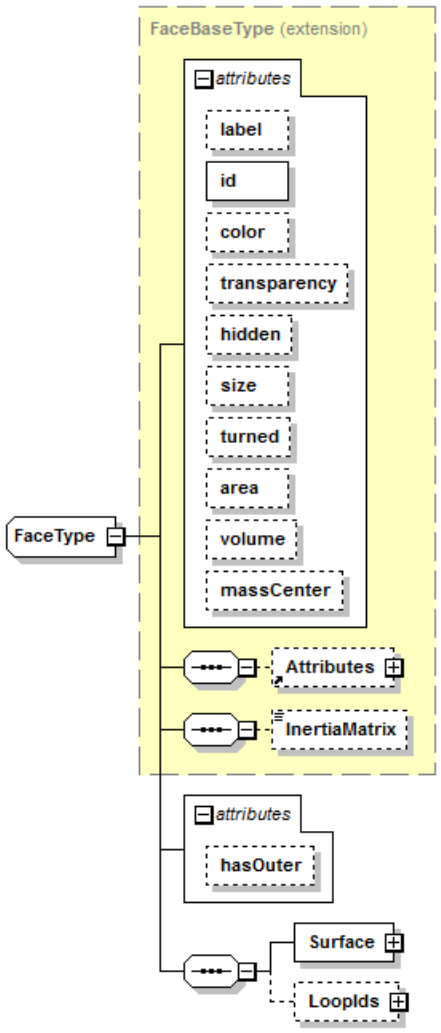
diagram	
children	**FaceBase**
used by	element **FaceSet**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute is the number of faces in this set.
annotation	documentation The FaceSetType represents a container for storing all model faces.

attribute **FaceSetType/@N**

type	**NaturalType**
properties	use required
facets	Kind: minInclusive; Value: 1; Annotation:
annotation	documentation The required N attribute is the number of faces in this set.

complexType **FaceType**

diagram



type	extension of **FaceBaseType**
properties	base FaceBaseType
children	**Attributes** **InertiaMatrix** **Surface** **LoopIds**
used by	element **Face**

attributes

Name	Type	Use	Default	Fixed	Annotation
label	**xs:string**				documentation The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree.
id	**QIFIdType**	required			documentation The required id attribute is the unique model entity identifier.

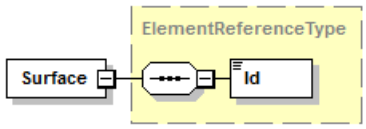
	color	**ColorType**		documentation The optional color attribute defines the RGB color property of a model entity.
	transparency	**xs:double**	0.0	documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**	0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**		documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
	turned	**xs:boolean**	0	documentation The optional turned attribute shows if the face orientation must be reversed from the orientation of the underlying surface. If the value is true, the face orientation must be opposite the surface orientation. If the value is false, the two orientations must be the same.

	area	**xs:double**		documentation The optional area attribute is the face area - the face 'contribution' to its shell volume. This value is defined in the model units.
	volume	**xs:double**		documentation The volume attribute is the face partial volume - the face 'contribution' to its shell volume. It is calculated relative to the parent block origin. This value is defined in the model units.
	massCenter	**PointSimpleType**		documentation The optional massCenter attribute is the center of mass of the face.
	hasOuter	**xs:boolean**	1	documentation The optional hasOuter attribute shows if the face has its outer loop explicitly represented in the loop array. If this attribute is false, the natural border of the underlying surface must be taken as the face outer loop.
annotation	documentation The FaceType is the b-rep face type - a bounded portion of a surface.			

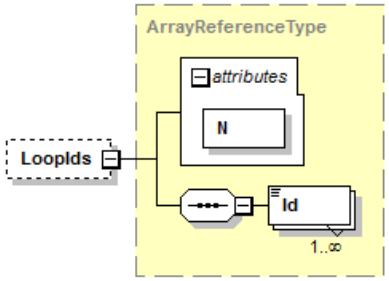
attribute **FaceType/@hasOuter**

type	**xs:boolean**
properties	default 1
annotation	documentation The optional hasOuter attribute shows if the face has its outer loop explicitly represented in the loop array. If this attribute is false, the natural border of the underlying surface must be taken as the face outer loop.

element **FaceType/Surface**

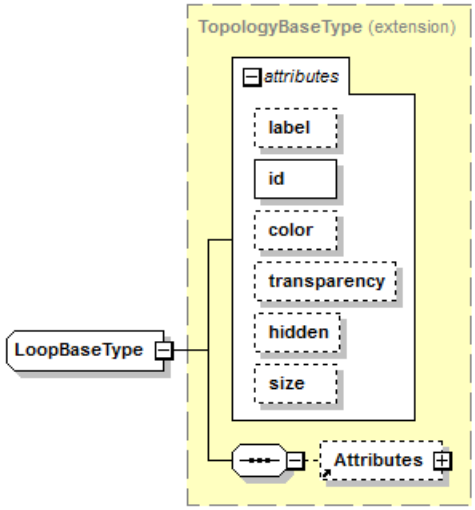
diagram	
type	**ElementReferenceType**
properties	content complex
children	**Id**
annotation	documentation The Surface element is the identifier of the underlying surface.

element **FaceType/LoopIds**

diagram	
type	**ArrayReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute shows how many Id elements are present in this array.
annotation	documentation The optional LoopIds element is an array of identifiers of the face trimming contours. If the hasOuter attribute is true then the first id will point to the outer loop otherwise all ids will point to inner loops only.

complexType **LoopBaseType**

diagram



type: extension of **TopologyBaseType**

properties:
- base: TopologyBaseType
- abstract: true

children: **Attributes**

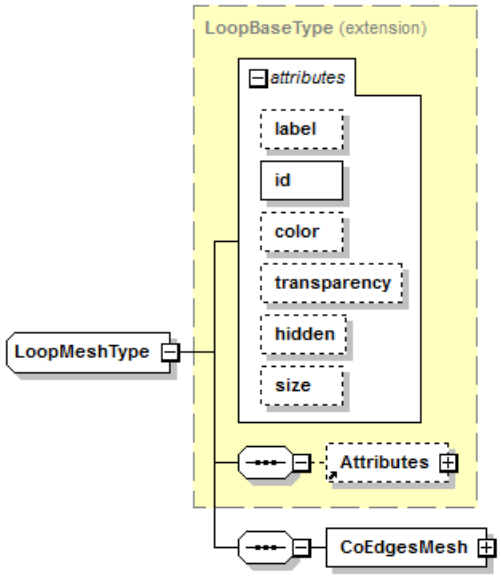
used by:
- element: **LoopBase**
- complexTypes: **LoopMeshType** **LoopType**

attributes

Name	Type	Use	Default	Fixed	Annotation
label	**xs:string**				documentation: The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree.
id	**QIFIdType**	required			documentation: The required id attribute is the unique model entity identifier.
color	**ColorType**				documentation: The optional color attribute defines the RGB color property of a model entity.
transparency	**xs:double**		0.0		documentation: The optional transparency attribute defines the transparency property of a model entity.
hidden	**xs:boolean**		0		documentation: The optional hidden attribute defines the visibility

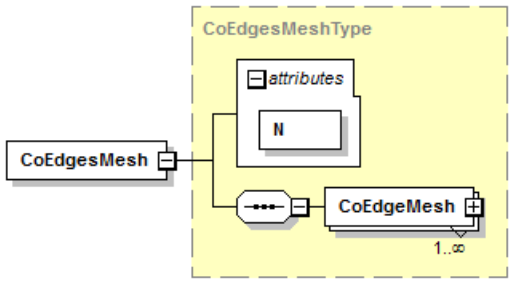
	size **DoublePositiveType**	property of a model entity in the graphical window. documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
annotation	documentation The LoopBaseType is the abstract base type for loops. A loop represents a circuit of edges bounding a face.	

complexType **LoopMeshType**

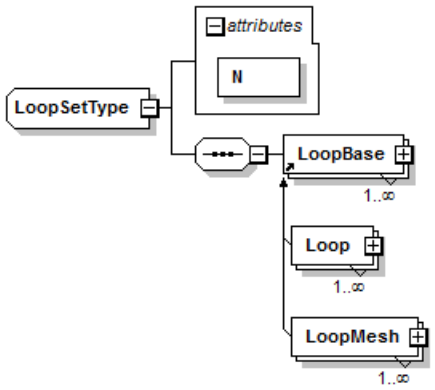
diagram	
type	extension of **LoopBaseType**
properties	base LoopBaseType
children	**Attributes** **CoEdgesMesh**
used by	element **LoopMesh**
attributes	Name: label; Type: **xs:string**; Use; Default; Fixed; Annotation: documentation The optional label attribute is the model entity "nameplate". Normally it

					can be seen at the entity item in the project tree.
	id	**QIFIdType**	required		documentation The required id attribute is the unique model entity identifier.
	color	**ColorType**			documentation The optional color attribute defines the RGB color property of a model entity.
	transparency	**xs:double**		0.0	documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**		0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**			documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
annotation	documentation The LoopMeshType is the b-rep mesh loop type - a topological entity. It is a circuit of edges bounding a face.				

element **LoopMeshType/CoEdgesMesh**

diagram	
type	**CoEdgesMeshType**
properties	content complex
children	**CoEdgeMesh**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute shows how many objects are presented in this array.
annotation	documentation The CoEdgesMesh is an array of mesh co-edges that forms mesh loop.

complexType **LoopSetType**

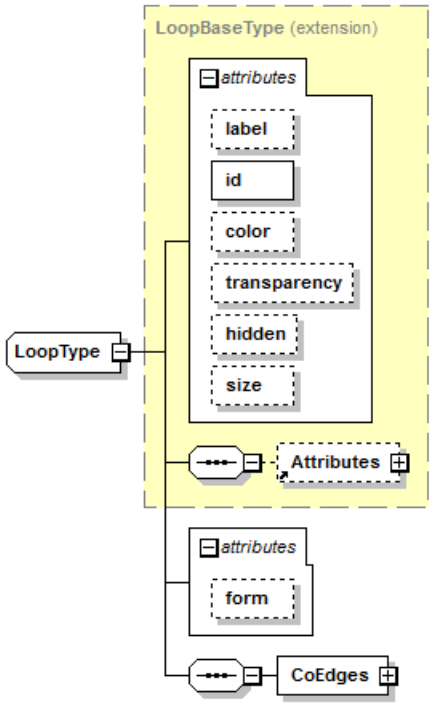
diagram	
children	**LoopBase**
used by	element **LoopSet**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute is the number of loops in this set.
annotation	documentation The LoopSetType represents a container for storing all model loops.

attribute **LoopSetType/@N**

type	**NaturalType**

properties	use required
facets	Kind Value Annotation minInclusive 1
annotation	documentation The required N attribute is the number of loops in this set.

complexType **LoopType**

diagram	
type	extension of **LoopBaseType**
properties	base LoopBaseType
children	**Attributes** **CoEdges**
used by	element **Loop**

attributes	Name	Type	Use	Default	Fixed	Annotation
	label	**xs:string**				documentation The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree.
	id	**QIFIdType**	required			documentation The required id attribute is the unique model entity identifier.
	color	**ColorType**				documentation The optional color attribute defines the

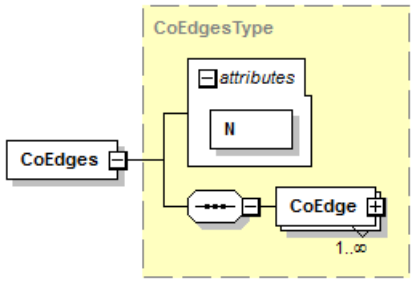
	transparency	**xs:double**	0.0	RGB color property of a model entity. documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**	0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**		documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
	form	**LoopFormEnumType**	UNKNOWN	documentation The optional form attribute specifies the loop type which can take one of the following values: 'UNKNOWN', 'OUTER', 'INNER' or 'SLIT'.
annotation	documentation The LoopType is the b-rep parametric loop type - a topological entity. It is a circuit of edges bounding a face.			

attribute **LoopType/@form**

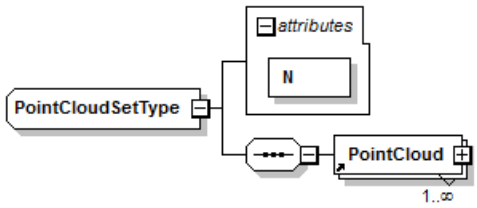
type	**LoopFormEnumType**		
properties	default UNKNOWN		
facets	Kind	Value	Annotation
	enumeration	UNKNOWN	
	enumeration	OUTER	
	enumeration	INNER	
	enumeration	SLIT	

annotation	documentation The optional form attribute specifies the loop type which can take one of the following values: 'UNKNOWN', 'OUTER', 'INNER' or 'SLIT'.

element **LoopType/CoEdges**

diagram	
type	**CoEdgesType**
properties	content complex
children	**CoEdge**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute shows how many objects are presented in this array.
annotation	documentation The CoEdges is an array of co-edges that forms loop.

complexType **PointCloudSetType**

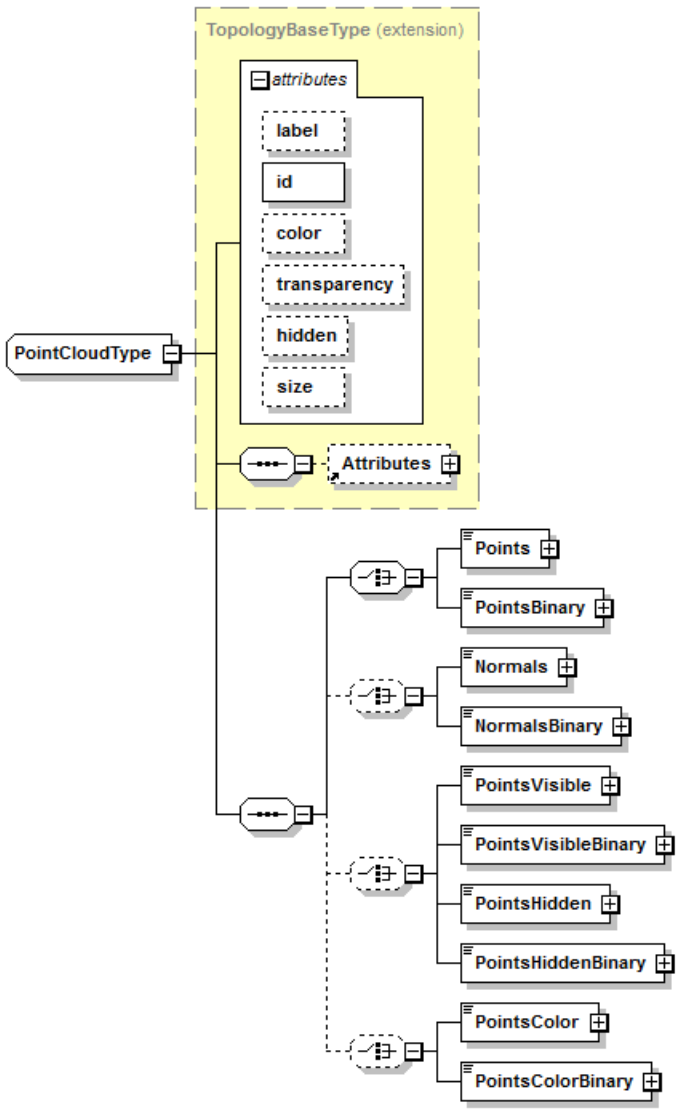
diagram	
children	**PointCloud**
used by	element **PointCloudSet**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute is the number of points in this set.
annotation	documentation The PointCloudSetType represents a container for storing all model point clouds.

attribute **PointCloudSetType/@N**

type	**NaturalType**

properties	use required
facets	Kind Value Annotation minInclusive 1
annotation	documentation The required N attribute is the number of points in this set.

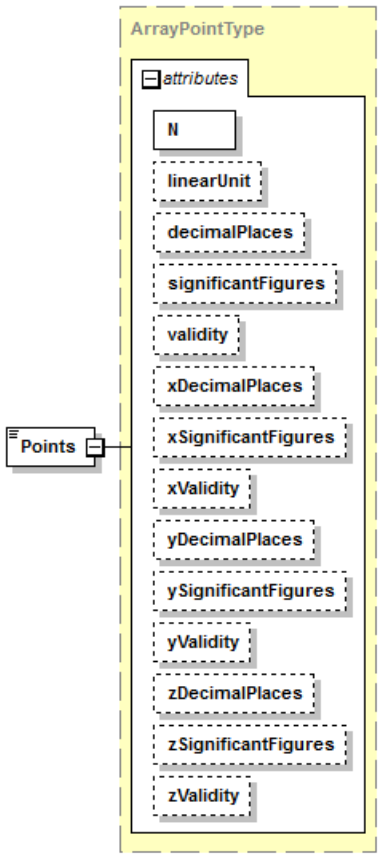
complexType **PointCloudType**

diagram	
type	extension of **TopologyBaseType**
properties	base TopologyBaseType
children	**Attributes** **Points** **PointsBinary** **Normals** **NormalsBinary** **PointsVisible** **PointsVisibleBinary** **PointsHidden** **PointsHiddenBinary** **PointsColor** **PointsColorBinary**
used by	element **PointCloud**
attributes	Name Type Use Default Fixed Annotation label **xs:string** documentation The optional label attribute

					is the model entity "nameplate". Normally it can be seen at the entity item in the project tree.
	id	**QIFIdType**	required		documentation The required id attribute is the unique model entity identifier.
	color	**ColorType**			documentation The optional color attribute defines the RGB color property of a model entity.
	transparency	**xs:double**		0.0	documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**		0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**			documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
annotation	documentation The PointCloudType describes a set of 3D points with optional normals.				

element **PointCloudType/Points**

diagram



type **ArrayPointType**

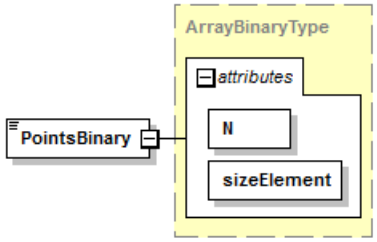
properties content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
N	**xs:positiveInteger**	required			documentation: The required N attribute gives the number of points represented by the array. The number of entries in the array must be 3N.
linearUnit	**xs:token**				
decimalPlaces	**xs:nonNegativeInteger**				
significantFigures	**xs:nonNegativeInteger**				
validity	**ValidityEnumType**				
xDecimalPlaces	**xs:nonNegativeInteger**				
xSignificantFigures	**xs:nonNegativeInteger**				
xValidity	**ValidityEnumType**				
yDecimalPlaces	**xs:nonNegativeInteger**				
ySignificantFigures	**xs:nonNegativeInteger**				
yValidity	**ValidityEnumType**				

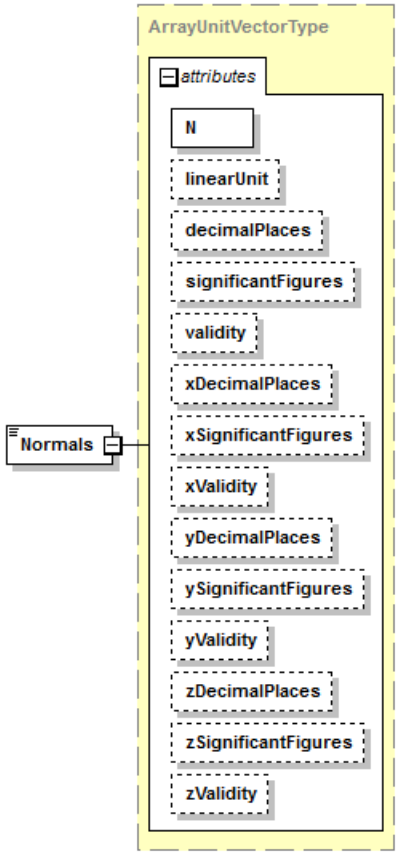
	zDecimalPlaces **xs:nonNegativeInteger** zSignificantFigures **xs:nonNegativeInteger** zValidity **ValidityEnumType**
annotation	documentation The Points element is an array of 3D points.

element **PointCloudType/PointsBinary**

diagram	
type	**ArrayBinaryType**
properties	content complex
attributes	Name / Type / Use / Default / Fixed / Annotation N **xs:unsignedInt** required — documentation: The required N attribute shows how many elements are present in this array. sizeElement **xs:unsignedInt** required — documentation: The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.
annotation	documentation The PointsBinary element is a binary array of 3D points.

element **PointCloudType/Normals**

diagram



type **ArrayUnitVectorType**

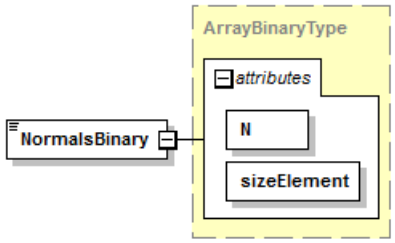
properties content complex

attributes

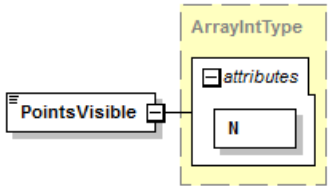
Name	Type	Use	Default	Fixed	Annotation
N	**xs:positiveInteger**	required			documentation: The required N attribute gives the number of unit vectors represented by the array. The number of entries in the array must be 3N.
linearUnit	**xs:token**				
decimalPlaces	**xs:nonNegativeInteger**				
significantFigures	**xs:nonNegativeInteger**				
validity	**ValidityEnumType**				
xDecimalPlaces	**xs:nonNegativeInteger**				
xSignificantFigures	**xs:nonNegativeInteger**				
xValidity	**ValidityEnumType**				
yDecimalPlaces	**xs:nonNegativeInteger**				
ySignificantFigures	**xs:nonNegativeInteger**				
yValidity	**ValidityEnumType**				

	zDecimalPlaces **xs:nonNegativeInteger** zSignificantFigures **xs:nonNegativeInteger** zValidity **ValidityEnumType**
annotation	documentation The Normals element is an array of normals. The number of elements in this array must equal the number of 3D points in this point cloud.

element **PointCloudType/NormalsBinary**

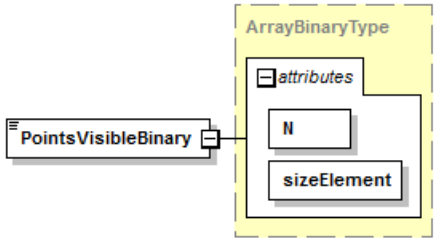
diagram	
type	**ArrayBinaryType**
properties	content complex
attributes	Name / Type / Use / Default / Fixed / Annotation N **xs:unsignedInt** required — documentation: The required N attribute shows how many elements are present in this array. sizeElement **xs:unsignedInt** required — documentation: The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.
annotation	documentation The NormalsBinary element is a binary array of normals. The number of elements in this array must equal the number of 3D points in this point cloud.

element **PointCloudType/PointsVisible**

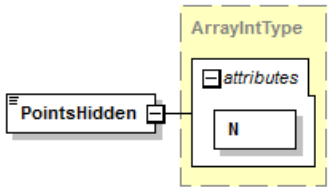
diagram	
type	**ArrayIntType**
properties	content complex

attributes	Name	Type	Use	Default	Fixed	Annotation
	N	**xs:unsignedInt**	required			documentation The required N attribute shows how many objects are present in this array.
annotation	documentation The PointsVisible element is an array of visible point indices. All elements of this integer array must be unique and must lie in the range [0, number of points].					

element **PointCloudType/PointsVisibleBinary**

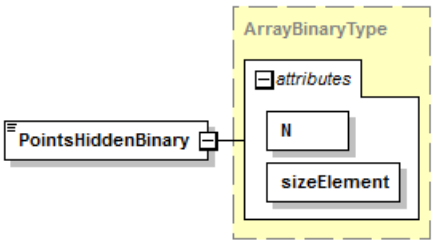
diagram						
type	**ArrayBinaryType**					
properties	content complex					
attributes	Name	Type	Use	Default	Fixed	Annotation
	N	**xs:unsignedInt**	required			documentation The required N attribute shows how many elements are present in this array.
	sizeElement	**xs:unsignedInt**	required			documentation The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.
annotation	documentation The PointsVisibleBinary element is a binary array of visible point indices. All elements of this integer array must be unique and must lie in the range [0, number of points].					

element **PointCloudType/PointsHidden**

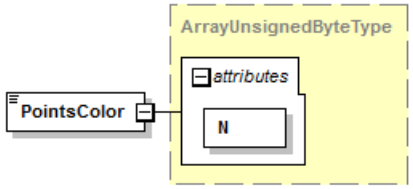
diagram	

type	**ArrayIntType**
properties	content complex
attributes	Name: N; Type: **xs:unsignedInt**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute shows how many objects are present in this array.
annotation	documentation: The PointsHidden element is an array of hidden point indices. All elements of this integer array must be unique and must lie in the range [0, number of points].

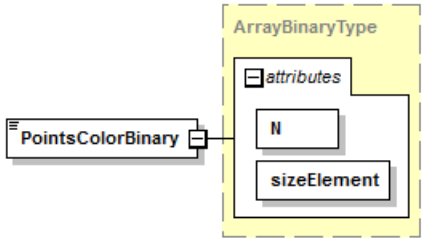
element **PointCloudType/PointsHiddenBinary**

diagram	
type	**ArrayBinaryType**
properties	content complex
attributes	Name: N; Type: **xs:unsignedInt**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute shows how many elements are present in this array. Name: sizeElement; Type: **xs:unsignedInt**; Use: required; Default: ; Fixed: ; Annotation documentation: The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.
annotation	documentation: The PointsHiddenBinary element is a binary array of hidden point indices. All elements of this integer array must be unique and must lie in the range [0, number of points].

element **PointCloudType/PointsColor**

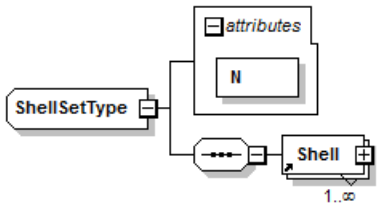
diagram	
type	**ArrayUnsignedByteType**
properties	content complex
attributes	Name: N; Type: **xs:unsignedInt**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute shows how many objects are present in this array.
annotation	documentation The PointsColor element is an array of unsigned byte values which defines colors of the points. Each element of this array is a triplet of unsigned byte numbers - the RGB color: the red-component, the green-component and the blue-component. The number of array elements corresponds to the number of points.

element **PointCloudType/PointsColorBinary**

diagram	
type	**ArrayBinaryType**
properties	content complex
attributes	Name: N; Type: **xs:unsignedInt**; Use: required; Default: ; Fixed: ; Annotation documentation: The required N attribute shows how many elements are present in this array. Name: sizeElement; Type: **xs:unsignedInt**; Use: required; Default: ; Fixed: ; Annotation documentation: The required sizeElement attribute shows the size (in bytes) of one element stored in the array. The total size of the binary array can be calculated as: N*sizeElement.
annotation	documentation The PointsColorBinary element is a binary array of unsigned byte values which defines colors of the points. Each element

	of this array is a triplet of unsigned byte numbers - the RGB color: the red-component, the green-component and the blue-component. The number of array elements corresponds to the number of points.

complexType **ShellSetType**

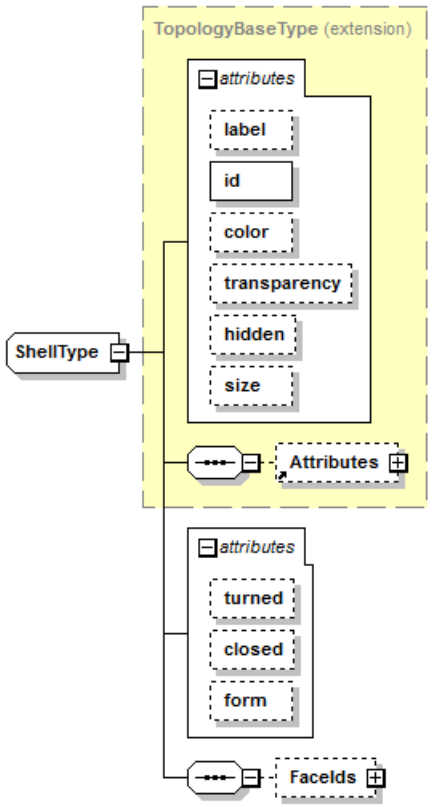
diagram	
children	**Shell**
used by	element **ShellSet**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute is the number of shells in this set.
annotation	documentation The ShellSetType represents a container for storing all model shells.

attribute **ShellSetType/@N**

type	**NaturalType**
properties	use required
facets	Kind: minInclusive; Value: 1; Annotation:
annotation	documentation The required N attribute is the number of shells in this set.

complexType **ShellType**

diagram



type: extension of **TopologyBaseType**

properties: base TopologyBaseType

children: **Attributes** **FaceIds**

used by: element **Shell**

attributes

Name	Type	Use	Default	Fixed	Annotation
label	**xs:string**				documentation: The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree.
id	**QIFIdType**	required			documentation: The required id attribute is the unique model entity identifier.
color	**ColorType**				documentation: The optional color attribute defines the RGB color property of a model entity.
transparency	**xs:double**		0.0		documentation: The optional

				transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**	0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**		documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
	turned	**xs:boolean**	0	documentation The optional turned attribute shows if the shell orientation must be reversed from the orientation of the component faces. If the value is true, the shell orientation must be opposite the faces orientation. If the value is false, the two orientations must be the same.
	closed	**xs:boolean**	0	documentation The optional closed attribute shows if the shell is a closed one (there are no gaps or open contours).

	form	**ShellFormEnumType**	UNKNOWN	documentation The optional form attribute specifies the shell type which can take one of the following values: 'UNKNOWN', 'OUTER' or 'INNER'.
annotation	documentation The ShellType is the b-rep shell type - a set of connected faces.			

attribute **ShellType/@turned**

type	**xs:boolean**
properties	default 0
annotation	documentation The optional turned attribute shows if the shell orientation must be reversed from the orientation of the component faces. If the value is true, the shell orientation must be opposite the faces orientation. If the value is false, the two orientations must be the same.

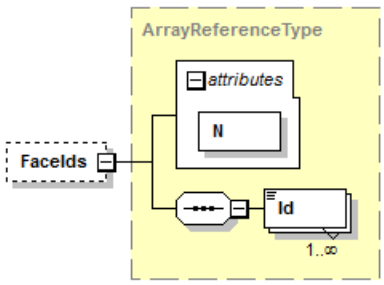
attribute **ShellType/@closed**

type	**xs:boolean**
properties	default 0
annotation	documentation The optional closed attribute shows if the shell is a closed one (there are no gaps or open contours).

attribute **ShellType/@form**

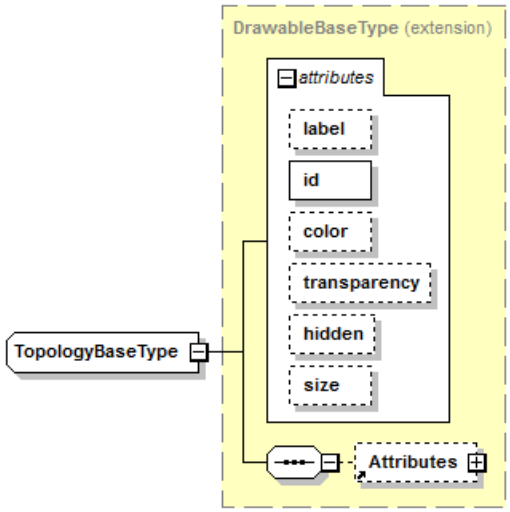
type	**ShellFormEnumType**
properties	default UNKNOWN
facets	Kind Value Annotation enumeration UNKNOWN enumeration OUTER enumeration INNER
annotation	documentation The optional form attribute specifies the shell type which can take one of the following values: 'UNKNOWN', 'OUTER' or 'INNER'.

element **ShellType/FaceIds**

diagram	

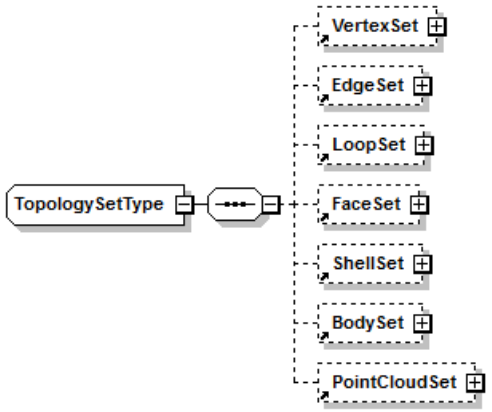
type	**ArrayReferenceType**
properties	minOcc 0 maxOcc 1 content complex
children	**Id**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute shows how many Id elements are present in this array.
annotation	documentation The optional FaceIds element is an array of connected faces which form this shell.

complexType **TopologyBaseType**

diagram	
type	extension of **DrawableBaseType**
properties	base DrawableBaseType abstract true
children	**Attributes**
used by	complexTypes **BodyType EdgeType FaceBaseType LoopBaseType PointCloudType ShellType VertexType**
attributes	Name: label; Type: **xs:string**; Use: ; Default: ; Fixed: ; Annotation: documentation: The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree. Name: id; Type: **QIFIdType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required id attribute is the unique model entity identifier.

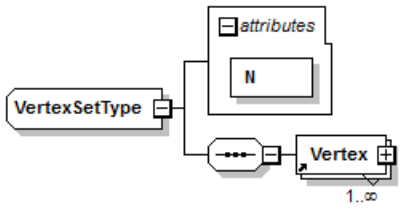
	color	**ColorType**		documentation The optional color attribute defines the RGB color property of a model entity.
	transparency	**xs:double**	0.0	documentation The optional transparency attribute defines the transparency property of a model entity.
	hidden	**xs:boolean**	0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**		documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
annotation	documentation The TopologyBaseType is the abstract base type for all topology elements.			

complexType **TopologySetType**

diagram	
children	**VertexSet EdgeSet LoopSet FaceSet ShellSet BodySet PointCloudSet**

used by	element **TopologySet**
annotation	documentation The TopologySetType describes the main container for storing all topological entities presented in the CAD scene.

complexType **VertexSetType**

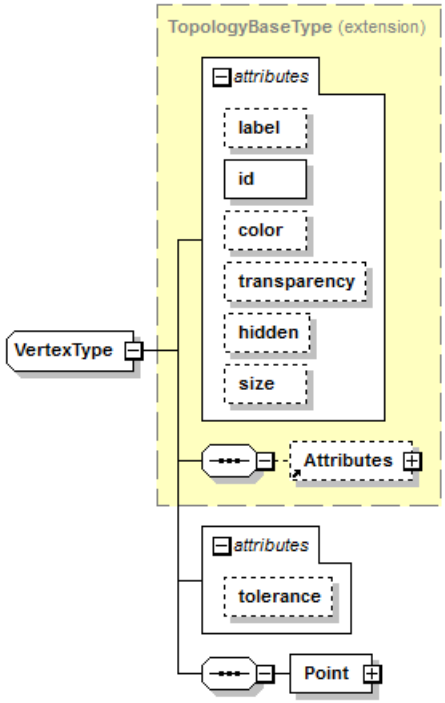
diagram	
children	**Vertex**
used by	element **VertexSet**
attributes	Name: N; Type: **NaturalType**; Use: required; Default: ; Fixed: ; Annotation: documentation: The required N attribute is the number of vertices in this set.
annotation	documentation The VertexSetType represents a container for storing all model vertices.

attribute **VertexSetType/@N**

type	**NaturalType**
properties	use required
facets	Kind: minInclusive; Value: 1; Annotation:
annotation	documentation The required N attribute is the number of vertices in this set.

complexType **VertexType**

diagram



type: extension of **TopologyBaseType**

properties: base TopologyBaseType

children: **Attributes** **Point**

used by: element **Vertex**

attributes

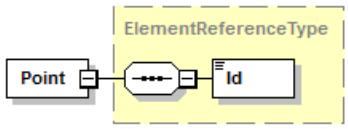
Name	Type	Use	Default	Fixed	Annotation
label	**xs:string**				documentation The optional label attribute is the model entity "nameplate". Normally it can be seen at the entity item in the project tree.
id	**QIFIdType**	required			documentation The required id attribute is the unique model entity identifier.
color	**ColorType**				documentation The optional color attribute defines the RGB color property of a model entity.
transparency	**xs:double**		0.0		documentation The optional transparency attribute defines the transparency

				property of a model entity.
	hidden	**xs:boolean**	0	documentation The optional hidden attribute defines the visibility property of a model entity in the graphical window.
	size	**DoublePositiveType**		documentation The optional size attribute defines a recommended size for visualization of an infinite drawable element such as plane, cylinder, axis, etc., or objects without explicit geometric parameters (e.g. coordinate system).
	tolerance	**xs:double**		documentation The optional tolerance attribute specifies the maximum distance from the vertex underlying geometry (3D point) to the ends of all the adjacent edges that are terminated in the neighborhood of this vertex. This value can be defined only for the case of the tolerant body.
annotation	documentation The VertexType is the b-rep vertex type - a topological entity. Each instance of VertexType lies at a point and is normally used to bound an edge.			

attribute **VertexType/@tolerance**

type	**xs:double**
annotation	documentation The optional tolerance attribute specifies the maximum distance from the vertex underlying geometry (3D point) to the ends of all the adjacent edges that are terminated in the neighborhood of this vertex. This value can be defined only for the case of the tolerant body.

element **VertexType/Point**

diagram	
type	**ElementReferenceType**
properties	content complex
children	**Id**
annotation	documentation The Point element is the identifier of a 3D point - the underlying geometry of the vertex.

simpleType **LoopFormEnumType**

type	restriction of **xs:string**
properties	base xs:string
used by	attribute **LoopType/@form**
facets	Kind Value Annotation enumeration UNKNOWN enumeration OUTER enumeration INNER enumeration SLIT
annotation	documentation The LoopBaseTypeType enumerates values that describe the loop type.

simpleType **ShellFormEnumType**

type	restriction of **xs:string**
properties	base xs:string
used by	attribute **ShellType/@form**
facets	Kind Value Annotation enumeration UNKNOWN enumeration OUTER enumeration INNER
annotation	documentation The ShellFormEnumType enumerates values that describe the shell type.

