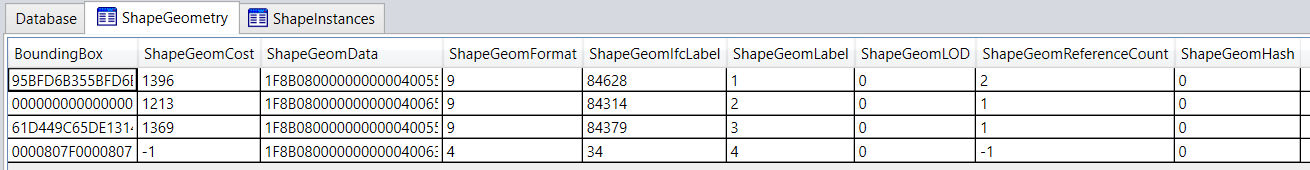
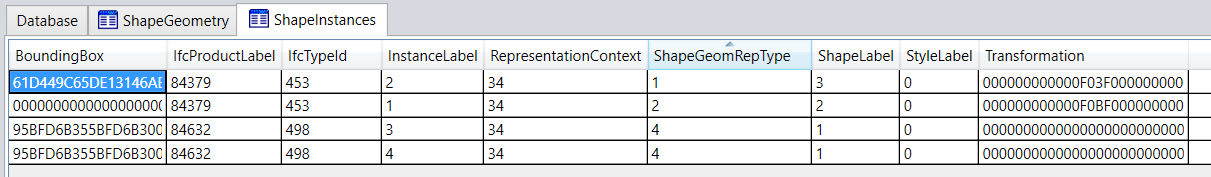
# ShapeGeometry



|  |  |
| --- | --- |
| BoundingBox |  |
| ShapeGeomCost |  |
| ShapeGeomData |  |
| ShapeGeomFormat | XbimGeometryType  Undefined = 0x0,  BoundingBox = 0x01,  MultipleBoundingBox = 0x02,  TriangulatedMesh = 0x03,  /// Regions (clusters of elements in a model) are stored for the project in one database row.  /// Use XbimRegionCollection.FromArray(ShapeData) for deserialising.  Region = 0x4,  /// For products with no geometry use TransformOnly to store the transform matrix associated with the placement.  TransformOnly = 0x5,  /// 128 bit hash of a geometry  TriangulatedMeshHash = 0x6,  /// The xBIM variant of the PLY format, a set of nominally planar polygons, stored in ascii format  Polyhedron = 0x7,  /// A triangulated Polyhedron mesh  TriangulatedPolyhedron = 0x8,  /// The xBIM variant of the PLY format, a set of nominally planar polygons but stored in binary format  PolyhedronBinary = 0x9, |
| ShapeGeomIfcLabel | EntityLabel of the relative IfcGeometricRepresentationItem in the model |
| ShapeGeomLabel | This is the ID that is pointed from the ShapeInstances.ShapeLabel **(Unique Key)** |
| ShapeGeomLOD |  |
| ShapeGeomReferenceCount |  |
| ShapeGeomHash |  |

# ShapeInstances



|  |  |
| --- | --- |
| BoundingBox |  |
| IfcProductLabel |  |
| IfcTypeId |  |
| InstanceLabel |  |
| RepresentationContext | EntityLabel of the IFC context |
| ShapeGeomRepType | Enum XbimGeometryRepresentationType  /// boolean operations with voids and extensions are included in the resulting representation  OpeningsAndAdditionsIncluded = 1,  /// boolean operations with voids and extensions are excluded in the resulting representation  OpeningsAndAdditionsExcluded = 2,  /// representation of voids and extensions only  OpeningsAndAdditionsOnly = 4 |
| ShapeLabel | Pointer to ShapeGeometry.ShapeGeomLabel |
| StyleLabel |  |
| Transformation |  |

# Geometry format

Byte Version [= 1]

Int32 numVertices

Int32 numTriangles

For NumVertices {

Single x coord of point

Single y coord of point

Single z coord of point

}

Int32 numFaces

For numFaces {

Int32 numTrianglesInFace

Bool IsPlanar = (numTrianglesInFace > 0)

numTrianglesInFace = abs(numTrianglesInFace)

if (isPlanar) {

SingleNormal

// to be documented

}

Else {

// to be documented

}

}

# New Boolean ops:

for each (XbimSolid^ s in solidSet)

{

BOPAlgo\_BOP aBOP;

aBOP.AddArgument(r);

aBOP.AddTool(s);

aBOP.SetOperation(BOPAlgo\_CUT);

aBOP.Perform();

r = aBOP.Shape();

}