Revit 2012几何数据访问以及对象关系判断

叶雄*进* *Joe Ye*

Developer Technical Services

June 2, 2011

Autodesk

关于讲师

- 叶雄进 Joe Ye
- Joe.Ye@autodesk.com
  - ADN DevTech部门
- 9年AEC行业软件经验
  - 4年多建筑领域软件开发咨询
  - 5年土木行业软件研发
- 支持APIs
  - Revit
  - AutoCAD Architecture
  - AutoCAD

要求与目标:

- 要求:
  - 学习本教程之前需要了解如何编写简单的命令
    - 若不了解，可参看Revit编程基础视频讲座
    - http://download.autodesk.com/media/adn/Revit_2011_API_DevTV_Chinese.zip
  - 了解.NET编程的最基本知识
    - 若不了解，请看Autodesk提供的一个供初学者教程(英语)：My First Revit Plug-in.
    - http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=16777469
  - 熟悉Revit产品
- 本课程目标:
  - 学习这些内容后，对几何数据访问有一个全面的了解
  - 了解对象关系判断的API可以有哪些方法
  - 如果对Revit编程熟悉，能拓宽编程思路

课程提纲

- 第一部分：获取对象的几何数据
  - 几何数据访问的方法
  - 例子演示
    - 读取系统族实例（如墙）的底面
    - 读取标准族实例（如柱）的底面
  - 几何相关类：
    - Reference
    - Transform
    - XYZ
    - Line
  - 编程小贴士

- 第二部分：对象几何关系的判断
  - 射线法
  - 相交法
  - 已有API判断对象关系

第一部分:读取对象几何数据

Principle of getting Geometry Data

几何数据访问原则

- Element.Geometry属性用于获得实体对象的几何数据
  public GeometryElement Geometry[Options options] { get; }

```
 public static FaceArray GetFaces(Element elem)
        {
            List<Face> faces = new List<Face>();

            Autodesk.Revit.DB.Options geoOptions =
Command.CommandData.Application.Application.Create.NewGeometryOptions();
            geoOptions.ComputeReferences = true;

            GeoElement geoElem = elem.get_Geometry(geoOptions);
            GeometryObjectArray geoElems = geoElem.Objects;

            foreach (object o in geoElems)
            {
                GeoSolid geoSolid = o as GeoSolid;
                if (null == geoSolid)
                {
                    continue;
                }

                return geoSolid.Faces;
            }

            return null;
        }
```

访问Geometry属性的参数

- `GeoElement geoElem = elem.get_Geometry(geoOptions);`

- Options类，用于指定返回几何数据的特征
  - 返回的几何对象可否带参考信息：设置 ComputeReferences属性 为true或false
  - 设置返回的几何信息的详细程度：设置DetailLevel（粗略，中等，详细等）
  - 返回哪个视图的几何信息：　　设置View属性

- 两种创建Options对象的方法：
  - Application.Creation.NewGeometryOptions()
  - Options的构造函数 New Options（）

代码

```
public static FaceArray GetFaces(Element elem)
        {
            List<Face> faces = new List<Face>();

            Autodesk.Revit.DB.Options geoOptions = Command.CommandData.Application.Application.Create.NewGeometryOptions();
            geoOptions.ComputeReferences = true;

            GeoElement geoElem = elem.get_Geometry(geoOptions);
            GeometryObjectArray geoElems = geoElem.Objects;

            foreach (object o in geoElems)
            {
                GeoSolid geoSolid = o as GeoSolid;
                if (null == geoSolid)
                {
                    continue;
                }

                return geoSolid.Faces;
            }

            return null;
        }
```

模型几何数据的组成

Geometry属性返回GeometryElement对象

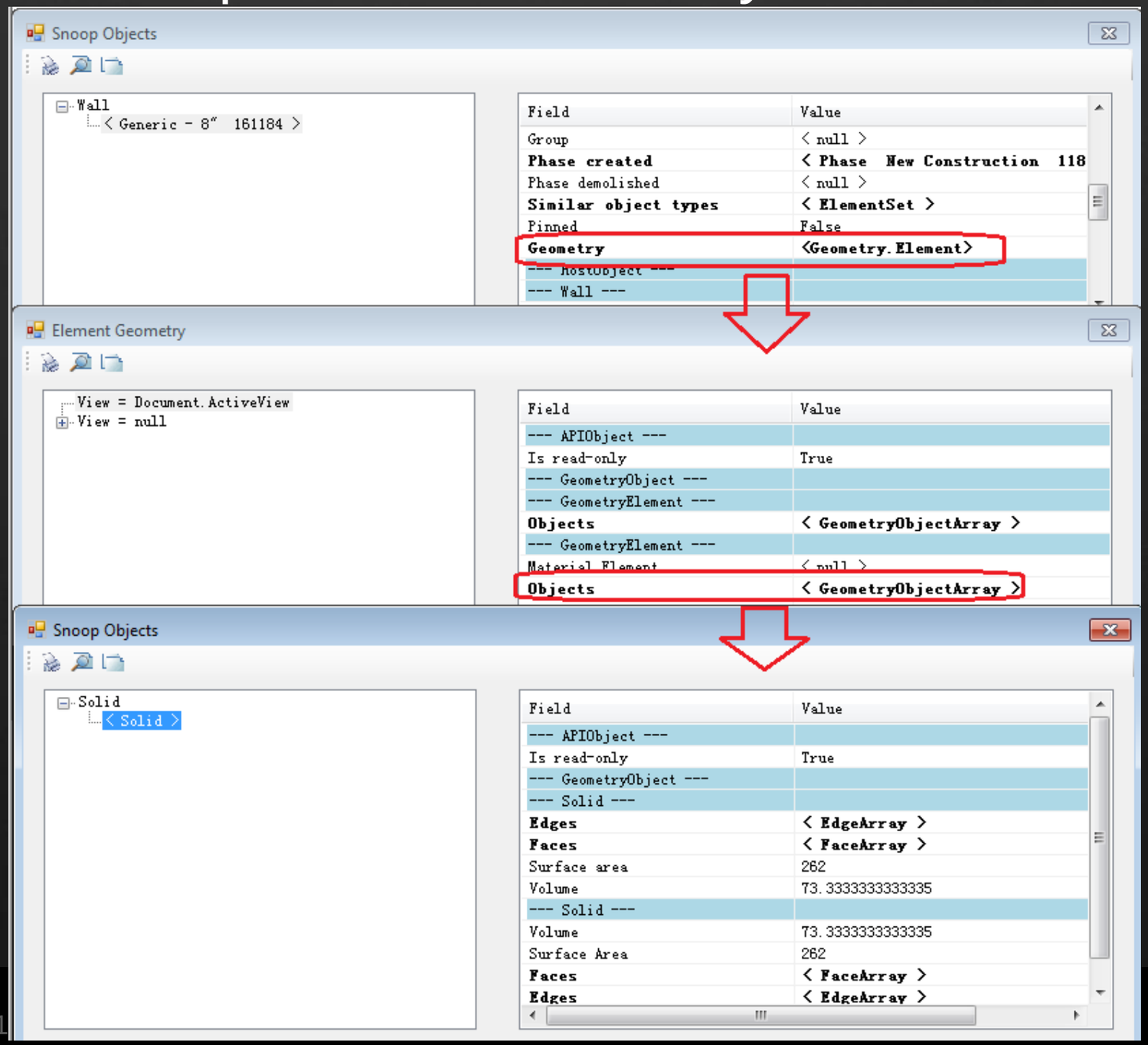
GeometryElement.Objects返回GeometryObjectArray,
其中包含这些对象:

- Autodesk.Revit.DB..::.GeometryElement
- Autodesk.Revit.DB..::.GeometryInstance（包含标准族实例几何信息）
- Autodesk.Revit.DB..::.Mesh （网格）
- Autodesk.Revit.DB..::.Solid （三维实体）
- Autodesk.Revit.DB..::.Edge （棱边）
- Autodesk.Revit.DB..::.Face （表面）
- Autodesk.Revit.DB..::.Point （点）

（这些对象都从GeometryObject派生）

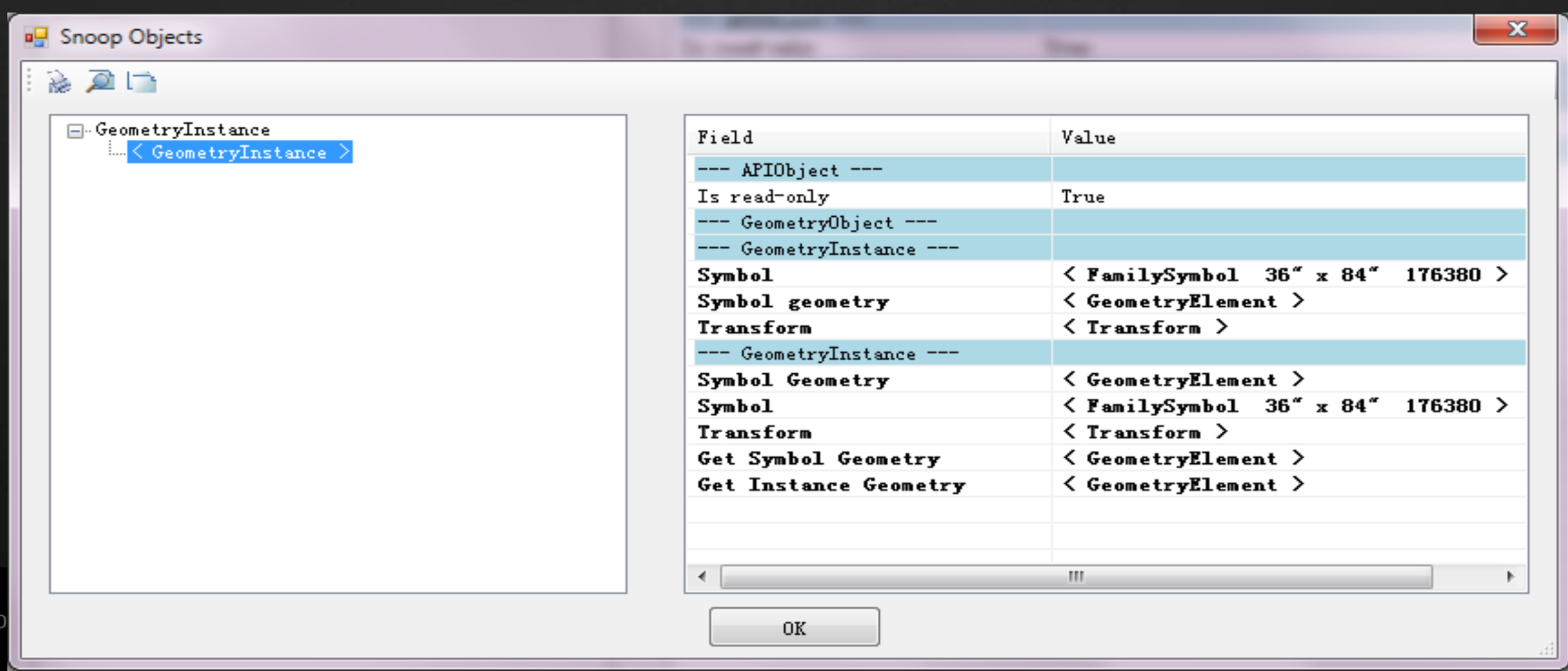
系统族实例(如墙)中包含的几何信息

- 系统族的Geometry直接包含该实例的几何实体
- 下图使用RevitLookup查看墙的Geometry属性返回的信息



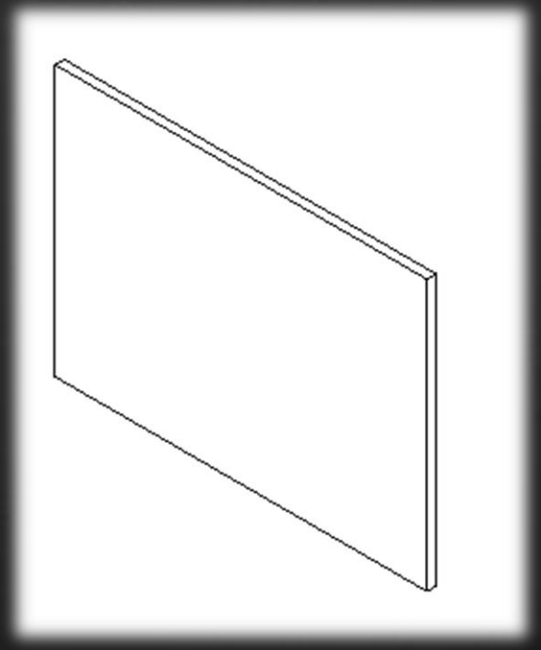
标准族实例(如:门)中包含的几何信息(未被切割)

- 未被几何切割:
- 标准族是通过族文件来定义的, 所以标准族实例的几何数据是GeometryInstance
- GeometryInstance中包含了族定义的几何数据
- GeometryInstance.GetSymbolGeometry: 族类型定义几何数据
- GeometryInstance.GetInstanceGeometry:族实例的几何数据



获得垂直墙的底面

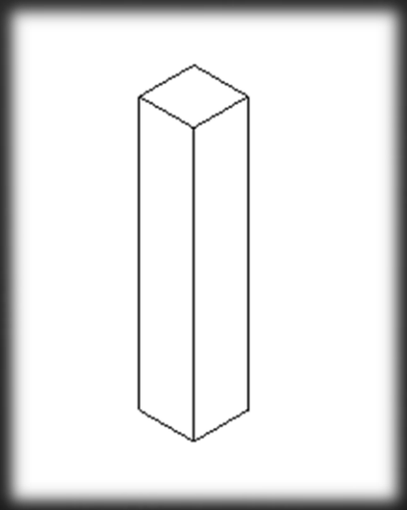
- 1. 调用Geometry属性获取墙几何信息
- 2. 从GeometryElement中得到代表墙的Solid
- 3. 从Solid中遍历其中的每一个面
- 4. 若面的法向量是垂直向下，单位化后变成（0，0，-1）



Demo1

获得垂直柱子的底面(未切割)

- 1. 调用Geometry属性获取柱子的几何信息
- 2. 从GeometryElement中得到代表柱子族的GeometryInstance
- 3. 从GeometryInstance中获得该柱子Solid
- 4. 从Solid中遍历其中的每一个面
- 5. 若面的法向量是垂直向下，单位化后变成(0，0，-1)



Demo2

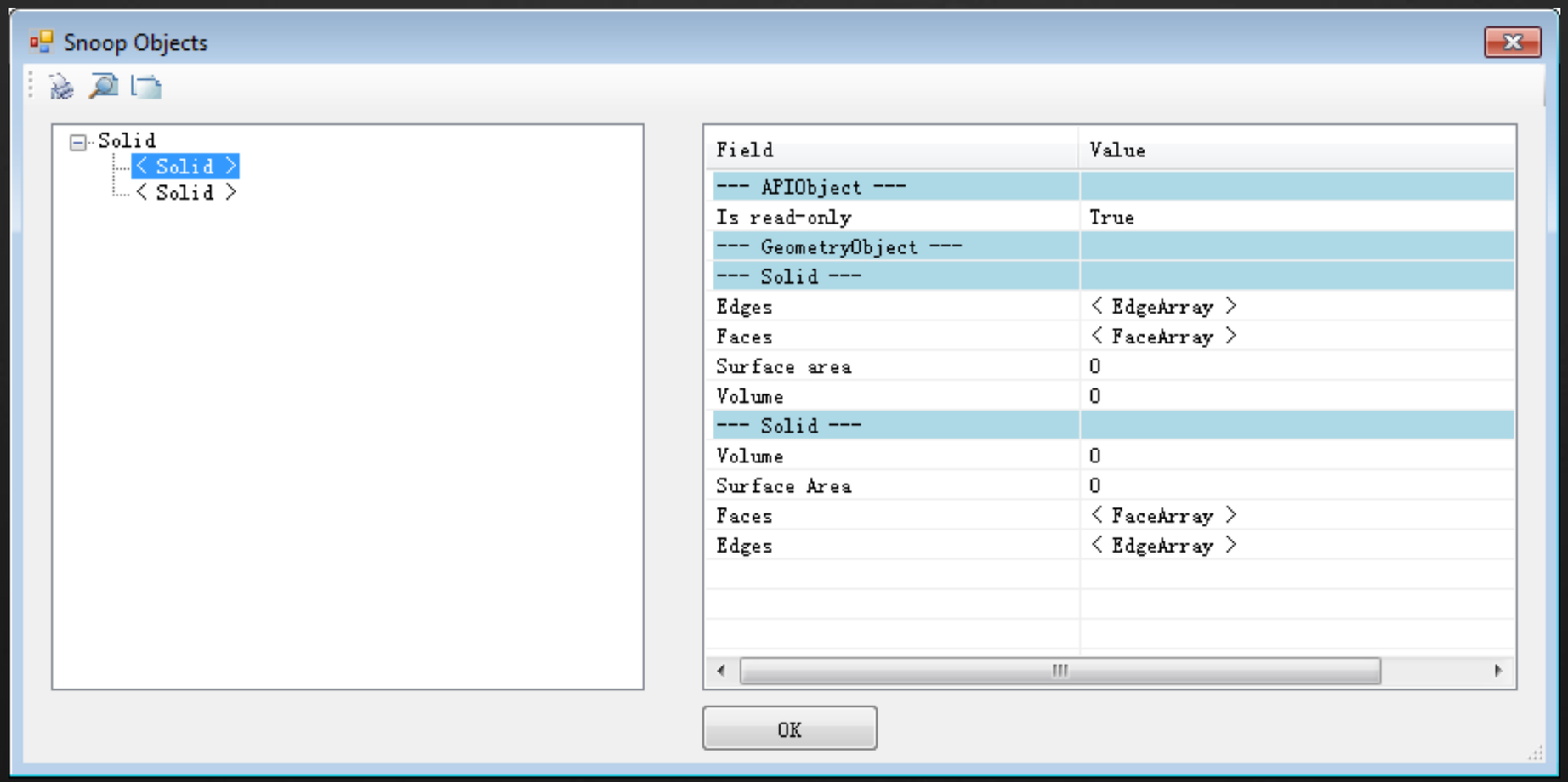
切割标准族实例的几何数据访问

- 当对象发生切割后，通过Element.Geometry返回的是切割后的数据。此时返回的几何数据直接包含Solid，不再保存在GeometryInstance对象中，因为标准族数据不能表达被切割族实例的几何数据。

- FamilyInstance.GetOriginalGeometry 返回族实例切割之前的数据

Revit对象几何数据的基本常识

- 1. 几何数据可以读取，但不能直接用修改几何数据
- 2. 几何形状可以通过修改对象的参数，或对对象进行切割来实现
- 3. 对象几何数据中可能包含没有属性的Solid，需要排除它。



坐标转换类:Transform

- 几何数据与坐标转换关系
  - 矩阵形成
  - （BasisX = Transform.BasisX; BasisY = Transform.BasisY, BasisZ = Transform.BasisZ)

$$\begin{pmatrix} BasisX.X & BasisY.X & BasisZ.X & Origin.X \\ BasisX.Y & BasisY.Y & BasisZ.Y & Origin.Y \\ BasisX.Z & BasisY.Z & BasisZ.Z & Origin.Z \end{pmatrix}$$

- 坐标转换

$$\begin{pmatrix} BasisX.X & BasisY.X & BasisZ.X & Origin.X \\ BasisX.Y & BasisY.Y & BasisZ.Y & Origin.Y \\ BasisX.Z & BasisY.Z & BasisZ.Z & Origin.Z \end{pmatrix} \times \begin{pmatrix} XYZ.X \\ XYZ.Y \\ XYZ.Z \\ 1 \end{pmatrix}$$

- 比如从族的一个类型插入到模型中的多个位置，就是通过给族实例赋予不同的坐标转换实现，基本的几何体保持不变

Reference:参考引用

- 指向Revit中所有可见的几何元素，可指向一个点，边，曲线，面

- 包含的信息
  - 从ElementId属性获取所指向对象的Id
  - 从Element. GetGeometryObjectFromReference(Reference)获取所指向的几何元素

- 作用:代表所指向的对象
- 用于
  - 标注，代表Reference
  - 几何代表
  - 标识几何对象
  - Reference可以序列化成字符串保存起来（ ConvertToStableRepresentation ）
  - 可以从序列化的字符串反序列化成Reference对象（ Reference..::.ParseFromStableRepresentation ）

基本几何类

- 点，以及向量：XYZ
- 边：Edge，Line
- 面：Face

第二部分:判断对象空间位置关系

Relationship

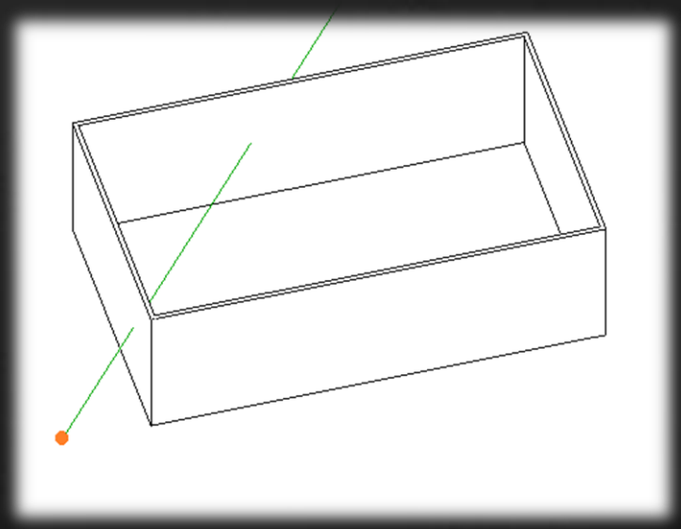
射线法判读对象空间关系

- 从一点沿给定向量发出一条射线，与模型中所有几何元素相交，返回相交几何元素的Reference

- Document 类的一个方法

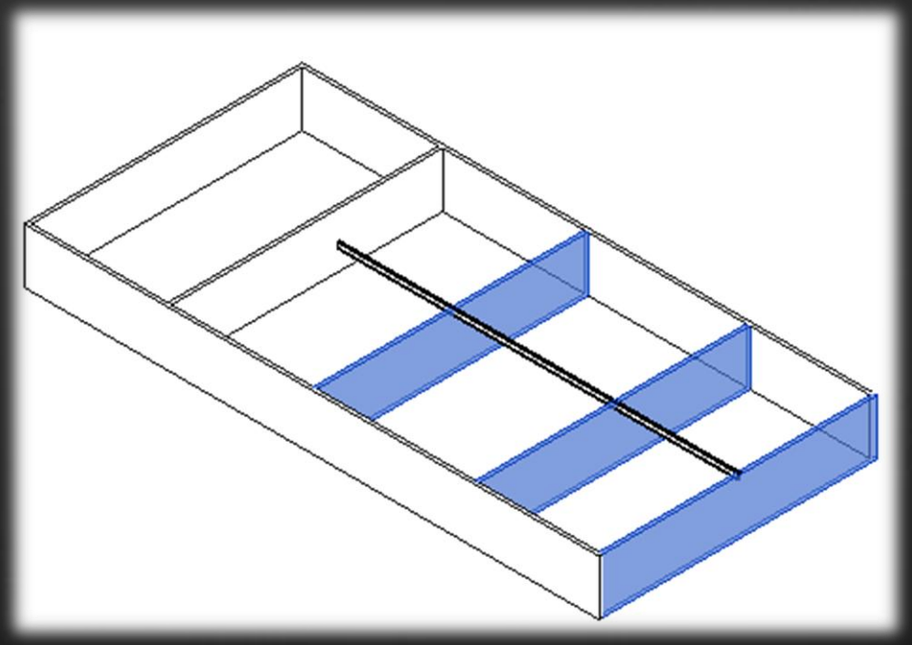
```
public IList<ReferenceWithContext> FindReferencesWithContextByDirection(
    XYZ pOrigin,
    XYZ pDirection,
    View3D pView
)
```

- ReferenceWithContext 包含信息：
  - 几何元素的引用，可以获得相交对象以及交点坐标
  - 交点到射线起点的近似距离



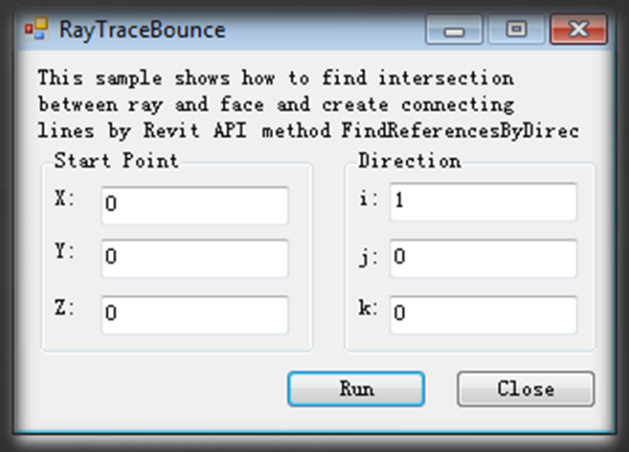
用法示例

- 获取支持梁的墙

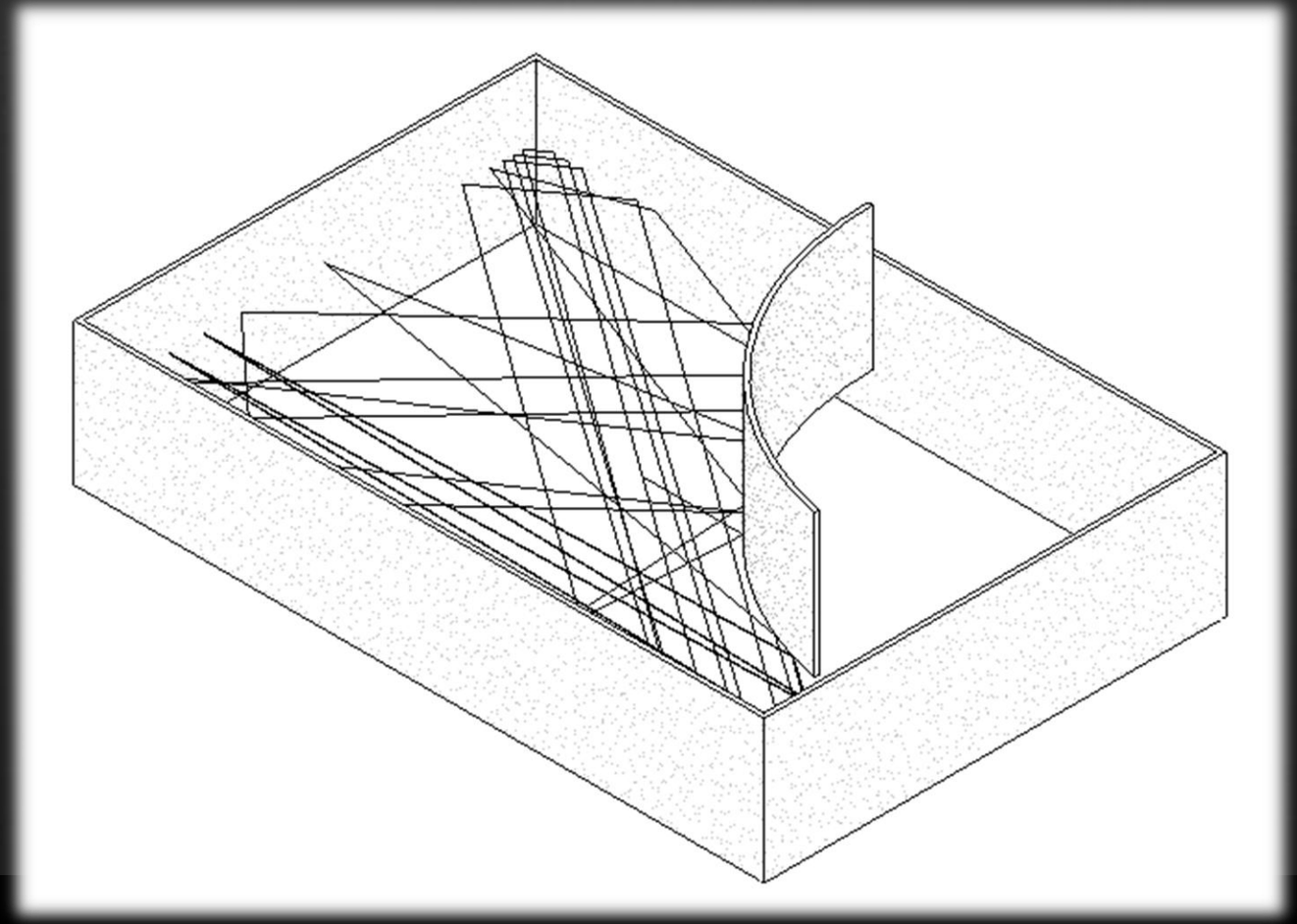


SDK中的例子

- 应用例子:
- SDK
- RaytraceBounce
- 目录:
  - SDK\Samples\FindReferencesByDirection



- 其它例子:
  - AvoidObstruction
  - FindColumns
  - MeasureHeight



可能的应用

- 判断位置关系：
- 比如想找一个构件的上方有什么构件，梁上的板？
  - 从梁上部发出垂直射线，穿越的第一个板既是梁上的板
- 比如一个连续梁下面有几个支撑主梁？
  - 从目标梁下面的一端发出射线，沿着梁的方向，射线所穿越的所有梁就是需要获得的主梁

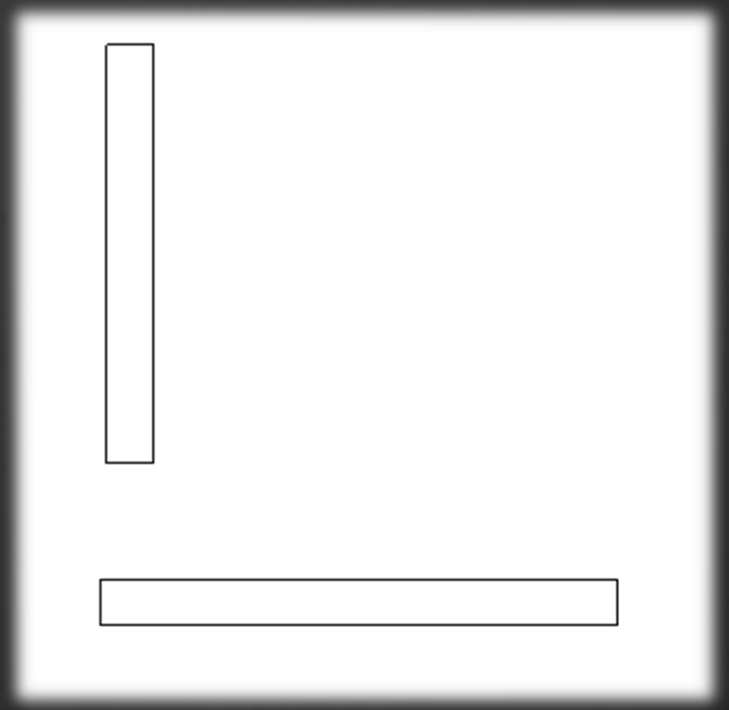
过滤器法判断对象空间关系

- 可用的API
  - ElementIntersectsElementFilter：与一个指定对象相交的过滤器
  - ElementIntersectsSolidFilter　：与一个给定Solid相交的过滤器，Solid可以是自己创建的

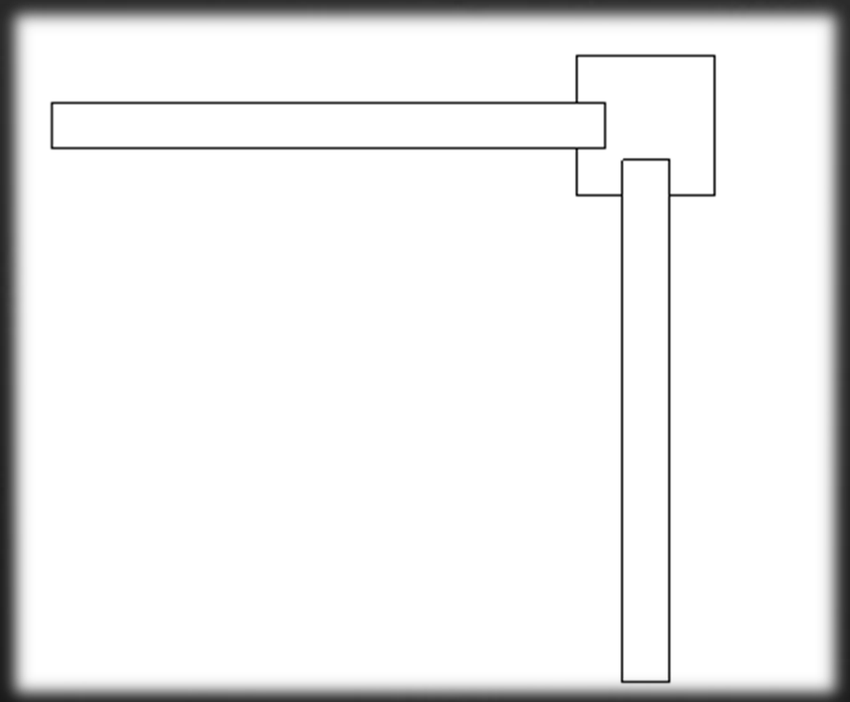
- 使用这两个Filter，获取模型中的对象

用法示例

- 用自创几何体相交过滤法



- 用对象相交过滤法



已有API判断对象关系

- Room 类
  - GetBoundarySegments属性：可找到组成该房间的所有边界墙，屋顶，楼板。

- AnalyticalModelSupport 类
  - 在RST中，如果模型中包含了对象支撑信息，可以从该类获取一个对象被哪些对象支撑。如果可以从楼板找到支撑该楼板的结构墙，柱以及梁。

深入学习

SDK中的资源

- 论坛讨论组
  - http://discussion.autodesk.com > Revit Architecture > Revit API

API 培训课程 & DevLab

- http://www.autodesk.com/apitraining

- The Building Coder, Jeremy Tammik 的 Revit API 博客
  - http://thebuildingcoder.typepad.com

- 我爱Revit-二次开发专栏博客(中文)
  - http://blog.csdn.net/joexiongjin

加入ADN

- http://www.autodesk.com/joinadn

- ADN网站有大量常见问题解决方案(只有ADN用户才能访问)
  - http://adn.autodesk.com

Thank you!

Autodesk

Autodesk®