

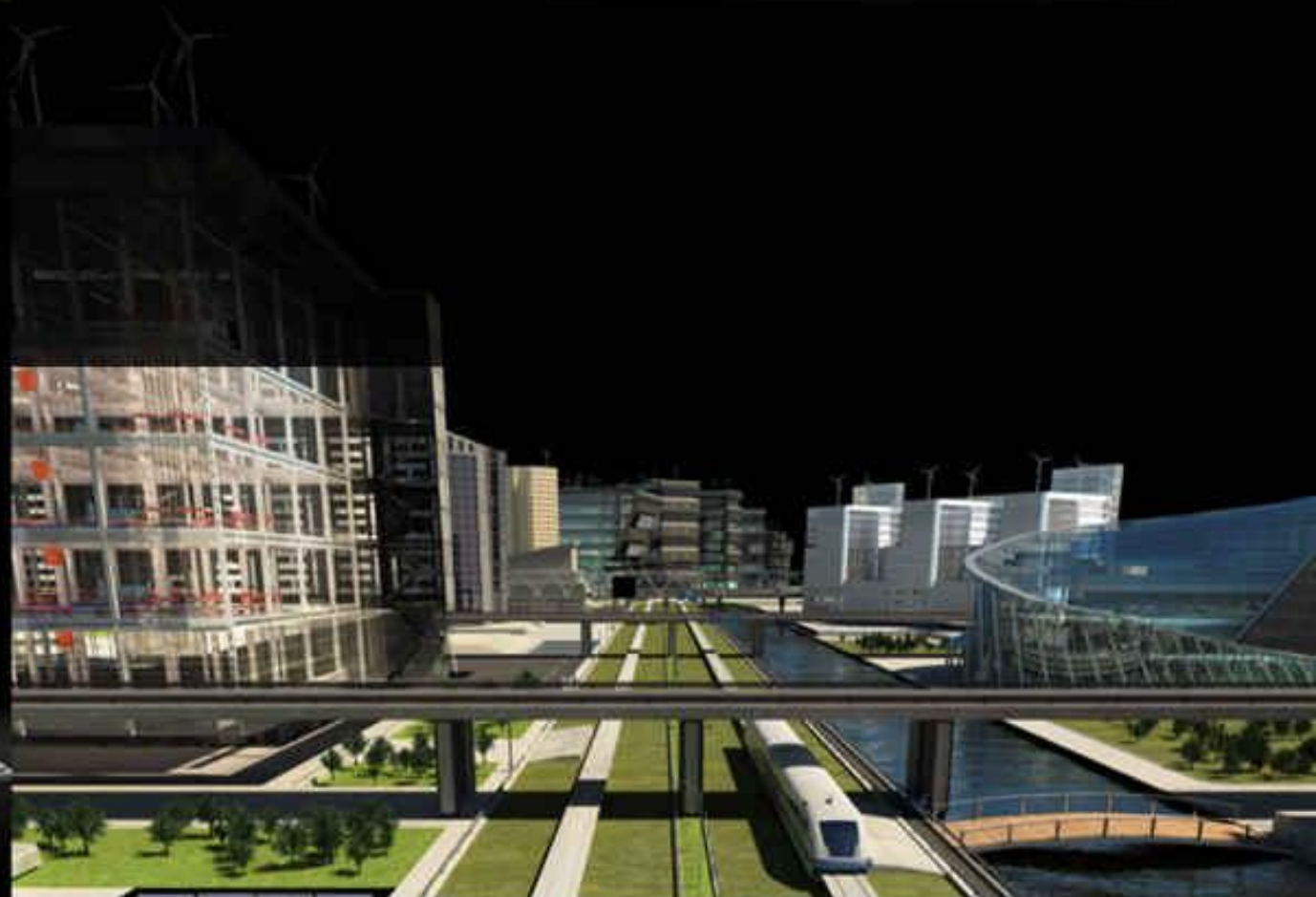


Revit 2012 API 新功能

叶雄进 Joe Ye

Developer Technical Services

June 2, 2011



关于讲师

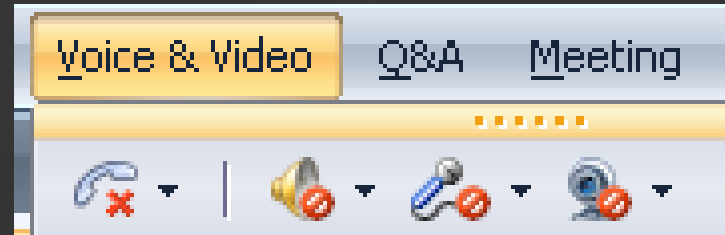
- 叶雄进 Joe Ye
- Joe.Ye@autodesk.com
 - ADN DevTech部门
 - 4年多软件开发咨询
 - 5年土木行业软件研发
- 支持APIs
 - Revit
 - AutoCAD Architecture
 - AutoCAD



准备工作

工具设置

- 使用计算机的声音

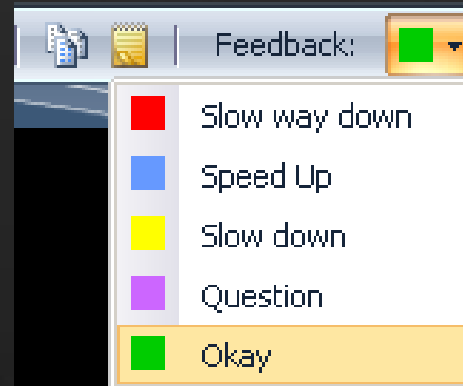


- 全屏模式

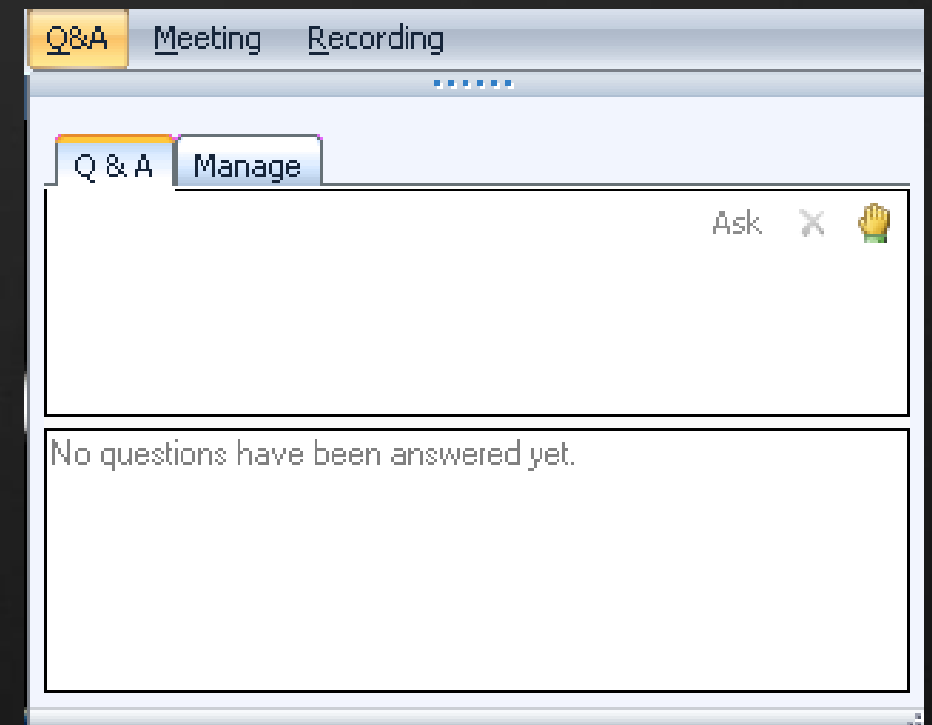


Expand to full screen mode
Press ESC to return

- 速度反馈



- 实时提问



- 开始不要打开您的计算机麦克，问答时可打开

今天的内容可下载

讲解内容录下来, 将发布在:

ADN Extranet

Revit > Knowledgebase > [Whitepapers and Training Videos](#)

Developer Center: www.autodesk.com/developers

API Training & Consulting > [Schedule](#)

调查

- 您的工作描述？
 - 建筑师, 工程师, 施工工程师, 程序员, 经理, 其它
- 您对Revit产品了解程度如何??
 - 经验非常丰富, 经验很丰富, 没有经验
- 您对Revit API的熟悉程度如何?
 - 经验丰富, 经验很丰富, 没有经验
- 为何参加这个讲座
 - 这个专题直接影响我今天的工作
 - 我期待在未来使用这个技术
 - 我在评测这项技术
 - 其它

内容

Revit 2012 API和SDK概览

移植2011的插件到 Revit 2012

Revit 2012 新 APIs

- 几何属性访问 API
- 建造模型
- 点云
- Revit Structure API 新功能

Break

- 扩展存储
- 板状构件的多层构造访问
- Revit MEP API新功能

新的 SDK samples

概览: API 和 SDK

SDK and getting started

Revit SDK

获取Revit SDK途径

- 产品安装包/安装光盘中有SDK
 - <extraction folder>\Utilities\SDK\RevitSDK.exe 或在安装光盘中搜索“SDK”可找到
 - 从2012版的产品安装程序中选择 “Install Tools and Utilities”
- ADN网站
 - Software & Support > Revit > Downloads
- Autodesk网站最新下载
 - <http://www.autodesk.com/developer> -->Products & Technologies--> Revit
 - <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2484975>

SDK 中的文档

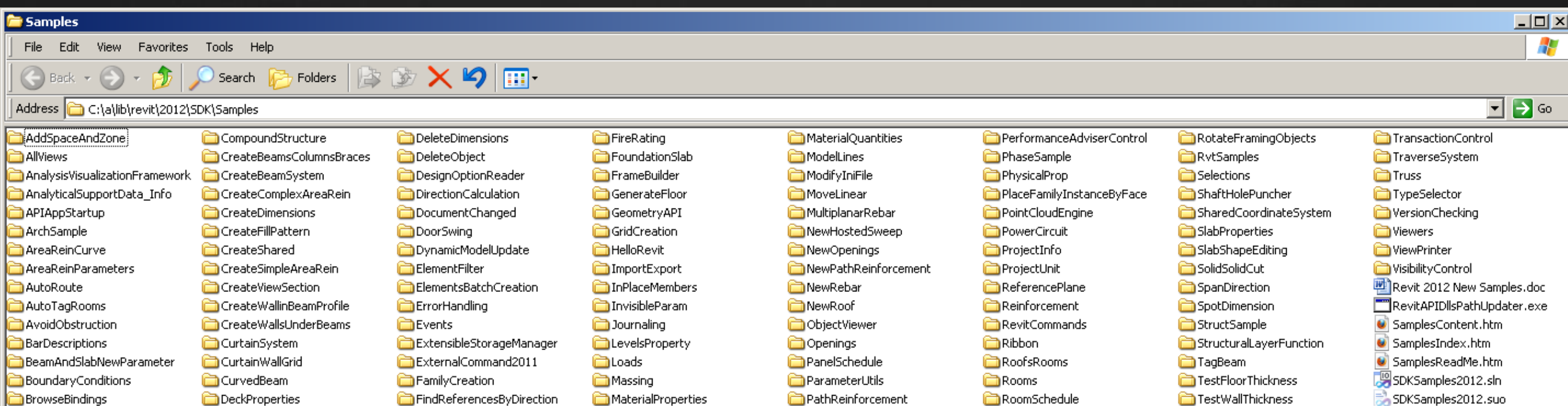
- 一次性阅读
 - Getting Started with the Revit API.doc
 - Revit Platform API Changes and Additions.doc
- 常用参考资料
 - Revit API Developer Guide.pdf
 - RevitAPI.chm
 - What's New 部分的内容与 Revit Platform API Changes and Additions 文档相同
- 需要时查阅
 - RevitAddInUtility.chm – 制作安装程序
 - Revit Server SDK – 访问服务器信息
 - REX SDK – Revit 扩展框架
 - Point Clouds Indexer API
- 重要的工具
 - Add-In Manager: 快速加载插件, 运行其中的命令
 - RevitLookup: 对象属性查看
- VSTA Samples
- Samples

Revit Wiki 帮助: <http://autodesk.com/revitapi-wikihelp> in 2012



Revit SDK 中的例程

- 文档
 - Revit 2012 New Samples.doc
 - SamplesReadMe.htm
- 批处理修改引用路径工具
 - RevitAPIDllsPathUpdater.exe
- 主要 samples solution
 - SDKSamples2012.sln
- 一个文件夹可能包含一个或多个源代码工程



Revit 二次开发如何起步

- Revit API 二次开发起步视频讲座
 - DevTV: [介绍 Revit 2011 编程基础\(中文\)](#), 可下载观看
 - http://download.autodesk.com/media/adn/Revit_2011_API_DevTV_Chinese.zip
 - DevTV: 介绍 Revit 2011 Programming (Part 1 and 2) (英语)
 - [Revit Developer Center](#)
- 其它讲座视频以及 AU 讲座视频
 - Developer Center > API Training and Consulting > Schedules > [Webcast Archives](#) filter by “Revit API”
 - [AU](#) online classes
 - 英文: <http://au.autodesk.com/>
 - 中文: <http://au.autodesk.com.cn>
- [Revit二次开发从入门到精通之路](#)
 - 博客: 我爱Revit—二次开发专栏博客
 -

Revit 2012 APIs 概览

今天的演示深入讲解几个2012API重大新功能

在线DevDay 视频介绍(英语)

- 完整概要性的介绍2012的所有新API
- http://download.autodesk.com/media/adn/DevDay_Online-Revit_2012_API.zip



Revit 2012 API 的变动和改进

- .NET Framework 4.0
- 不再使用 Revit.ini 文件来加载程序
- VendorId 开发者标签必需
- 人工模型更新模式
- 片状构件的材料层以及墙分隔缝/条
- 线形和填充样式
- 独立标签
- 导入导出APIs
- 保存和关闭文档API
- 引用类的属性改动
- 事件的变动
- 对象的移动/镜像/旋转/阵列 的变动
- 结构分析模型以及钢筋



Revit 2012 新API

- 打开和设置为当前文档, 设置当前视图
- 创建客户命令页 (ribbon tab)
- 工作集访问
- 几何功能的增强
- 房间和空间3维几何信息分析
- 具体能耗分析模型
- 概念能耗分析模型
- 分析可视化模型 (Analysis visualization framework)
- MEP API 增强
- 钢筋增强
- 施工模型
- 点云Point clouds
- 扩展存储



移植Revit 2011的插件到 Revit 2012

Migrating existing Revit 2011 add-ins to Revit 2012

移植

- Revit 使用 .NET Framework 4.0 (VSTA .NET 3.5)
- Visual Studio
 - VS 2008 (不能调试)
 - VS 2010 (.NET 3.5 or 4.0可都可)
- 插件加载
 - 不再用 .ini 文件
 - `<VendorId>ADSK</VendorId>` (必需的)
 - `<VendorDescription>Autodesk, www.autodesk.com</VendorDescription>` (可选)
 - RDS: 注册的开发者代号 www.autodesk.com/symbreg
- 重新引用接口DII
 - RevitAPI.dll
 - RevitAPIUI.dll

移植 (续.)

- 模型更新模式
 - 只能用人工更新模式
 - 删除模型更新模式 >> 保留也不会有效果
 - 调用Document.Regeneration()。当你要访问构件的几何数据前，如果对模型做了影响几何数据的改动，调用该语句更新模型
- 一些函数参数变化，如，
 - Document.Import().Export
 - Compound Structure and wall sweeps
- 完整信息请看
 - “*Revit Platform API Changes and Additions.docx*”
 - *RevitAPI.chm* >> *What's New in Revit 2012 API help*

几何相关的 API

几何实体的生成，布尔运算以及工具

几何 API

- 提取族实例的原始几何数据
 - 被切割或连接操作之前的几何形状

```
Options m_options = new Options();  
GeometryElement originalGeo = instance.GetOriginalGeometry(m_options);
```

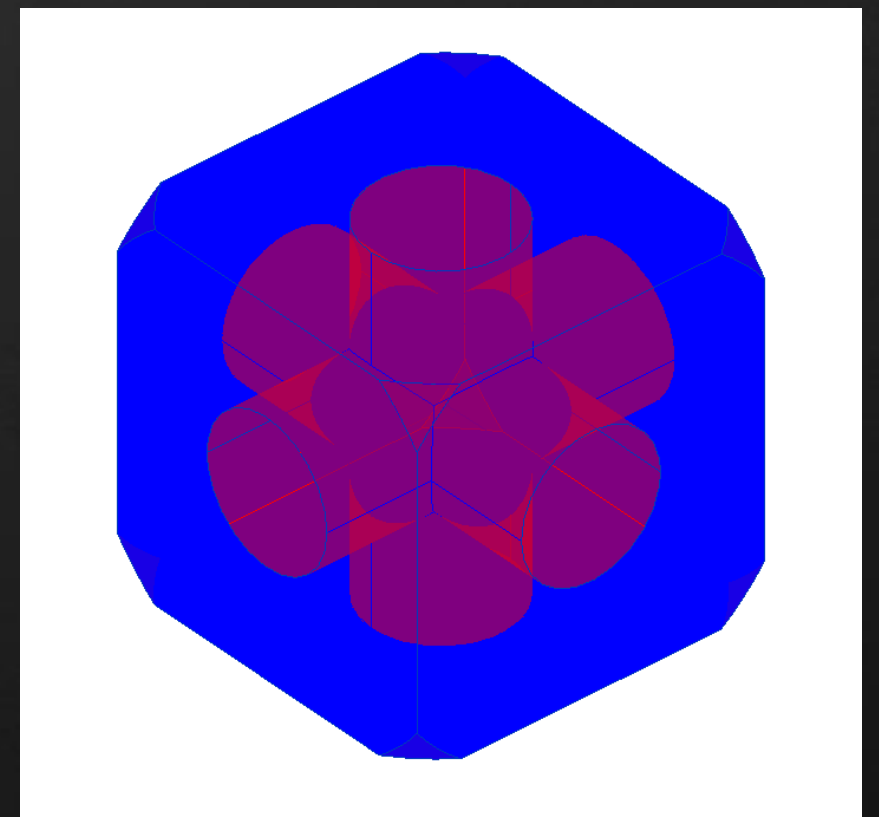
- GeometryCreationUtilities – 创建临时几何体,但是不能添加到模型中(可用于几何体相交过滤, 分析可视化, 布尔运算, 投影等)
 - CreateBlendGeometry()
 - CreateExtrusionGeometry()
 - CreateRevolvedGeometry()
 - CreateSweptGeometry()
 - CreateSweptBlendGeometry()
- BooleanOperationsUtils – 几何体布尔运算
 - ExecuteBooleanOperation()
 - ExecuteBooleanOperationModifyingOriginalSolid()
 - GeometryObject.IsElementGeometry

几何 API

- 演示

GeometryCreation_BooleanOperation

- 对象创建和几何实体布尔运算



几何 API

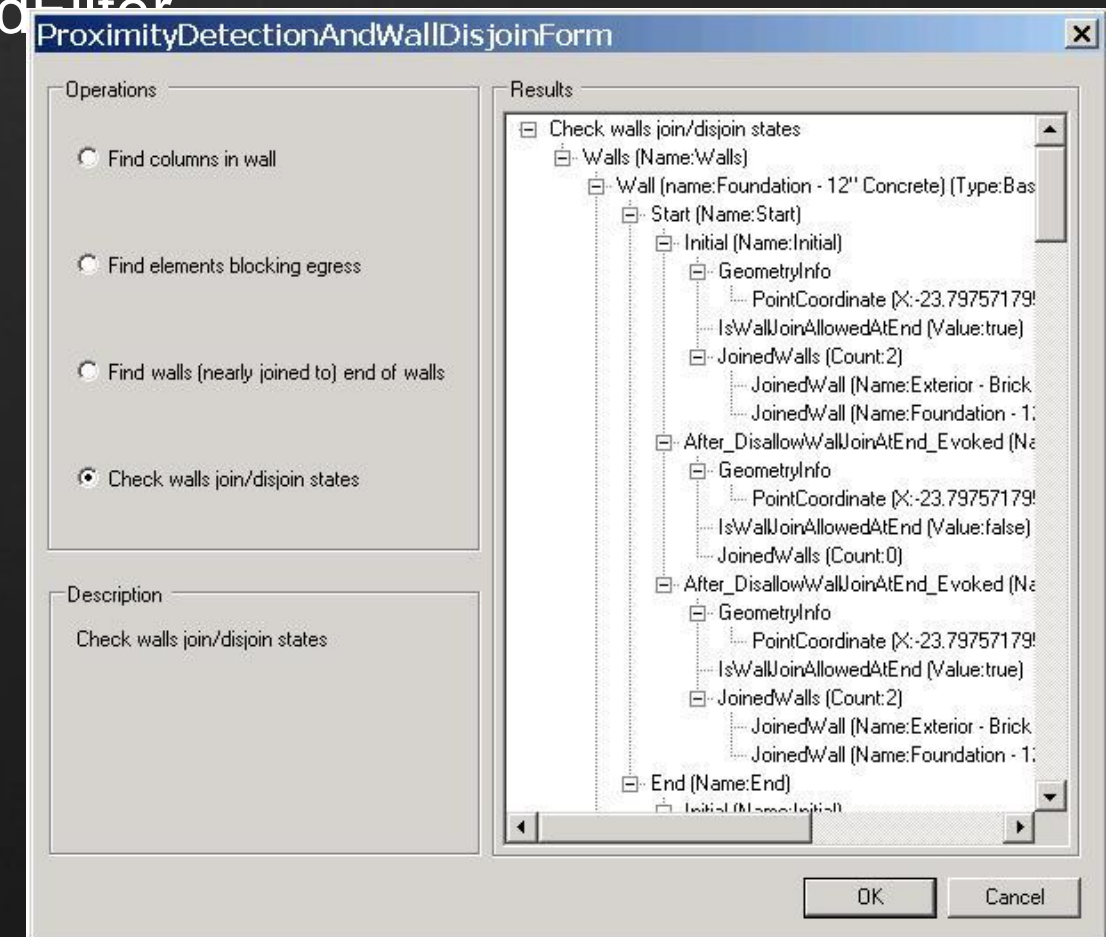
- HostObjectUtils – 返回楼板/屋顶/墙的侧面和上下地面
- ExtrusionAnalyzer – 把几何体投影为拉伸体
 - 获取这个拉伸体的形状
- 几何过滤器 – 找到与几何体相交的Revit模型元素
 - ElementIntersectsElementFilter
 - ElementIntersectsSolidFilter

几何 API

- 演示:

ProximityDetection_WallJoinControl

- 演示了两个过滤器的用法:
 - 与对象相交过滤器: `ElementIntersectsElementFilter`
 - 与几何体相交过滤器: `ElementIntersectsSolidFilter`



几何 API

- FamilyInstance.HostFace – 返回family instance所在面的引用

```
Reference corbelHostFace = corbel.HostFace;
```

- 模型更新和几何数据的状态 - 如果模型几何信息更新了,但是没有调用Regenerate(), 调用Element.Geometry 属性抛出异常
- GeometryObject.GraphicsStyleId – 返回几何原始对象(Line, Face, Edge等)的样式ElementId
- Edge.AsCurve() – 返回Edge所对应的Curve
- Solid.ComputeCentroid – 计算几何实体的质心坐标

几何 API

▪ 矩阵变换Transformation

- GeometryElement.GetTransformed() – 返回一个几何元素经变换后的几何元素
- Instance.GetTransform() – 返回一个Instance的变换矩阵(从这个Instance到WCS
- Instance.GetTotalTransform() – 返回总矩阵(如.导入实例的正北转换矩阵).

▪ 引用的序列化和逆序列化Serialization/deserialization

- Reference.ConvertToStableRepresentation() – 把Reference转换成对应的稳定字符串。
- Reference.ParseFromStableRepresentation() – 恢复字符串成为Reference, 可以在不同的命令中稳定获得这个字符串代表的Reference

```
public string SerializeReference(Document doc, Reference reference)
{
    String ids = String.Empty;
    ids = reference.ConvertToStableRepresentation(doc);
    TaskDialog.Show("SerializeReference", "Representation:\n" + ids);
    return ids;
}
```

几何 API

- 面域 (Face regions) (用Split Face 命令创建的)
 - Face.HasRegions – 这个面上是否有面域
 - Face.GetRegions() – 返回这个面上的所有面域
- 新的重载函数创建Hermite Spline
 - NewHermiteSpline(controlPoints, bPeriodic, startTangent, endTangent) – 可以指定起点和终点的切线方向
- 多义线 (PolyLine)
 - 新增的几何对象
 - 代表一系列点所形成的相连线段
 - 可以从Element.Geometry 属性获得多义线

建造模型

零件，部件以及部件图

建造模型

■ 零件(Parts)

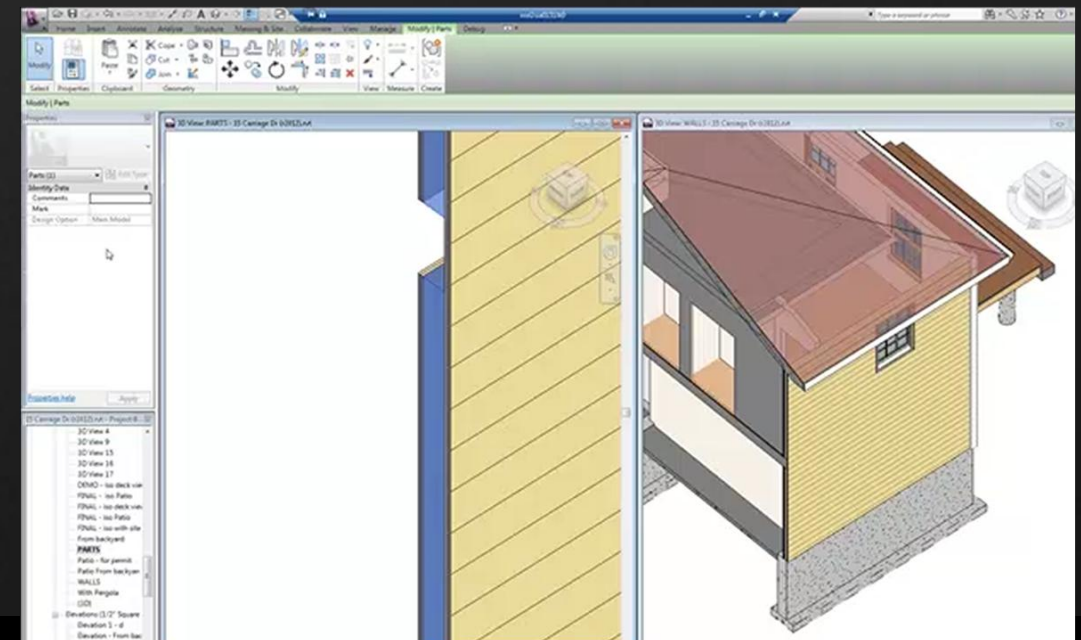
- 一个建筑构件拆开得到的零件
- 对于独立的材料表, 标注, 以及过滤和导出均有帮助
- 可以进一步细分成更小的零件
- 零件会自动更新以反应出基本元素的改动

■ 部件(Assemblies)

- 集合多个实例成一个部件
- 一个部件中的多个对象可以作为一个单位来操纵
- 同一个部件类型可以被插入到模型中的多个地方
- 可以用于创建图纸以及部件视图 (列在工程浏览器的assembly节点下)

■ 配套的新 API

- 允许对象
 - 被分成更小的子零件
 - 收集起来组成一个部件
 - 显示在部件视图中



建造模型

- 部件类型和实例

- Autodesk.Revit.DB.Assembly.**AssemblyInstance**
 - 组合多个对象来标记, 过滤, 创建明细表以及创建部件视图
- Autodesk.Revit.DB.Assembly.**AssemblyType**
 - 表示部件的类型

- 成员访问

- AssemblyInstance.GetMemberIds() – 获得部件中的零件
- AssemblyInstance.SetMemberIds() – 设置部件实例包含哪些对象Ids

建造模型

- 部件视图

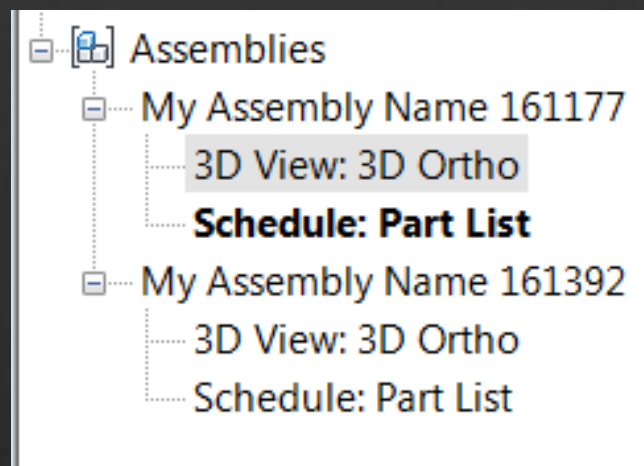
- AssemblyViewUtils – 部件视图工具类, 下面函数可以创建不同视图
 - Create3DOrthographic()
 - CreateDetailSection()
 - CreateMaterialTakeOff()
 - CreatePartList()
 - CreateSheet()
- 部件视图只显示部件中的对象

- 零件工具

- 提供一套方法来创建和操纵零件
 - CreateParts()
 - DivideParts()
 - GetAssociatedParts() – 返回所有指定对象有关的零件
 - GetDividedParents() – 返回一个零件中的所有对象
 - Others

建造模型

- 演示
- 生成一个部件
- 生成部件3D视图
- 生成部件的明细表视图



Part List		
Category	Family and Ty	Count
Columns	Rectangular C	1
Walls	Basic Wall: G	1
Windows	Fixed: 36" x 4	1
Windows	Fixed: 36" x 4	1

生成部件

```
ICollection<ElementId> elemIds = commandData.Application.ActiveUIDocument.Selection.GetElementIds();
ElementId categoryId = null;
ElementId keyElemId = null;
foreach (ElementId elemId in elemIds)
{
    // use category of one of the assembly elements
    keyElemId = elemId;
    categoryId = doc.get_Element(elemId).Category.Id;
    break;
}
Transaction transaction = new Transaction(doc);
if (AssemblyInstance.IsValidNamingCategory(doc, categoryId, elemIds))
{
    transaction.Start("Create Assembly Instance");
    AssemblyInstance assemblyInstance = AssemblyInstance.Create(doc, elemIds, categoryId);
    transaction.Commit(); // commit the transaction that creates the assembly instance before modifyi

    transaction.Start("Set Assembly Name");
    assemblyInstance.AssemblyTypeName = "My Assembly Name " + keyElemId.ToString();
    transaction.Commit();

    if (assemblyInstance.AllowsAssemblyViewCreation()) // create assembly views for this assembly ins
    {
        transaction.Start("View Creation");
        View3D view3d = AssemblyViewUtils.Create3DOrthographic(doc, assemblyInstance.Id);
        View partList = AssemblyViewUtils.CreatePartList(doc, assemblyInstance.Id);
        transaction.Commit();
    }
}
```

点云- Point Cloud

点云引擎和API

点云APIs

三种类型的APIs:

- 操作点云API
 - 在Revit内创建并操作点云实例
- 自定义点云引擎 API
 - 读入点云数据到Revit的API
- 点云数据转换 API
 - 转换客户数据格式为.pcg 格式(能被Revit和Autodesk其它产品识别)
 - C/C++ API
 - 文档在Revit SDK 'Point Cloud Indexing API' 子文件夹中

点云操作API

- Autodesk.Revit.DB.PointClouds 命令空间
- 主要类
 - PointCloudInstance: 代表点云实体对象
 - PointCloudType – 点云类型
 - CloudPoint – 点云中的一个点. (x, y, z, 点云坐标系), 以及颜色
 - PointCollection – 点的集合
 - PointCloudFilter – 体积过滤, 多面组成
 - PointCloudFilterFactory.CreateMultiPlanerFilter() 生成体积过滤器
- 两种方式的访问
 - IEnumerable – 传统的 “for each” 访问 IPointCloudEngine 接口
 - GetPointBufferPointer()
 - 直接访问非托管数据
 - 只能用C# or C++/CLI
 - 不安全
 - 高效

点云操作 API

- 与用户交互的工具
 - PointCloudInstance.SetSelectionFilter()
 - PointCloudInstance.FilterAction – 高亮/隔离
 - Selection.PickBox()

如：提醒用户选择部分点云，高亮显示

点云引擎API

- 实现引擎

- IPointCloudEngine –通过PointCloudEngineRegistry [.RegisterPointCloudEngine](#)来注册
- IPointCloudAccess – 根据Revit的需要查询一个点。方法: GetName(), ReadPoints(), etc.
- IPointSetIterator – 返回点的集合到Revit. 方法: ReadPoints(), Free()

- 基于文件和不基于文件

- 基于文件的 – 从外部文件中读入点云所需要的数据 (如 .pcg,, .xml)
- 不基于文件 – 可以从其它地方读入 (如, database, server)

Revit Structure

结构分析模型线，钢筋功能增强

Revit Structure API新功能

- AnalyticalModel 从 Element派生, 不是IDisposable
 - 现在AnalyticalModel 对象保存在数据库中
 - 分析模型线只存在于 AnalyticalModel 实例中
- 新的RebarHostData 方法
 - GetExposedFaces()
 - GetCommonCoverType()
 - GetRebarsInHost(), GetAreaReinforcementsInHost(), GetPathReinforcementsInHost()
- NewBeamSystem() 新增重载函数, 需要 Level 或 SketchPlane 作为参数确定位置平面
- NewTruss() 需要 SketchPlane参数

Rebar API新功能

- 创建钢筋形状的APIs发生很大的改变
 - 步骤改变
 - 创建、编辑钢筋形状的方法
 - ElementId 用来代表参数Parameter
 - 这块是升级的要点
- 其它
 - 对象创建的方法移到各自的类, 成为成员函数
 - Document.Create.NewRebarHookType → RebarHookType.Create
 - Document.Create.NewRebar → Rebar.CreateFromCurves & Rebar.CreateFromRebarShape

休息

5分钟后继续...

扩展存储 API

Extensible Storage

扩展存储

- 创建自定义的结构化存储架构
- 把自定义数据附着到Revit模型中的对象上
- 这个自定义数据存储技术可以取代不可见的共享参数
- 基于自定义架构的数据保存在Rvt文件中
- 可以被所有人写入，读出，或只是创建者，或只是某一个具体的应用程序

存储数据到一个对象上的步骤

1. 创建并命名新的数据架构(Schema)
2. 设置数据架构(Schema)读写访问权限
3. 给数据架构定义一个或多个字段
4. 基于数据架构创建一个数据对象(Entity)
5. 给数据对象中的字段赋值
6. 关联数据对象Entity和Revit模型中的对象

```

void StoreDataInWall(Wall wall, XYZ dataToStore)
{
    Transaction createSchemaAndStoreData = new Transaction(wall.Document, "tCreateAndStore");
    createSchemaAndStoreData.Start();
    SchemaBuilder schemaBuilder =
        new SchemaBuilder(new Guid("720080CB-DA99-40DC-9415-E53F280AA1F0"));
    schemaBuilder.SetReadAccessLevel(AccessLevel.Public); // allow anyone to read the object
    schemaBuilder.SetWriteAccessLevel(AccessLevel.Vendor); // restrict writing to this vendor
only
    schemaBuilder.SetVendorId("ADSK"); // required because of restricted write-access
    schemaBuilder.SetSchemaName("WireSpliceLocation");
    // create a field to store an XYZ
    FieldBuilder fieldBuilder =
        schemaBuilder.AddSimpleField("WireSpliceLocation", typeof(XYZ));
    fieldBuilder.SetUnitType(UnitType.UT_Length);
    fieldBuilder.SetDocumentation("A stored location value representing a wiring splice in a
wall.");

    Schema schema = schemaBuilder.Finish(); // register the Schema object
    Entity entity = new Entity(schema); // create an entity (object) for this schema (class)
    // get the field from the schema
    Field fieldSpliceLocation = schema.GetField("WireSpliceLocation");
    // set the value for this entity
    entity.Set<XYZ>(fieldSpliceLocation, dataToStore, DisplayUnitType.DUT_METERS);

    wall.SetEntity(entity); // store the entity in the element

    // get the data back from the wall
    Entity retrievedEntity = wall.GetEntity(schema);
    XYZ retrievedData =
        retrievedEntity.Get<XYZ>(schema.GetField("WireSpliceLocation"),
            DisplayUnitType.DUT_METERS);
    createSchemaAndStoreData.Commit();
}

```

扩展存储

■ 涉及到的类

- Schema 类 代表一个数据架构
- Schema实例由 SchemaBuilder 类来定义并创建
- 字段由FieldBuilder 类来创建
- 数据实体Entity 类保存着数据(以Schema的架构定义)
- 数据实体Entity实例可以关联到任何的Revit对象

■ 例子和更多信息

- 本培训随带材料中包含一个简单的例子 ExtensibleStorage
- DynamicModelUpdate 简单演示扩展存储
- ExtensibleStorageManager SDK sample稍复杂些, 与XML读写联系起来
- Developer Guide Chapter 22.5 Extensible Storage

板状构件的多层构造和墙分隔条/缝

Compound Structure and Wall Sweeps

复合板结构 API功能

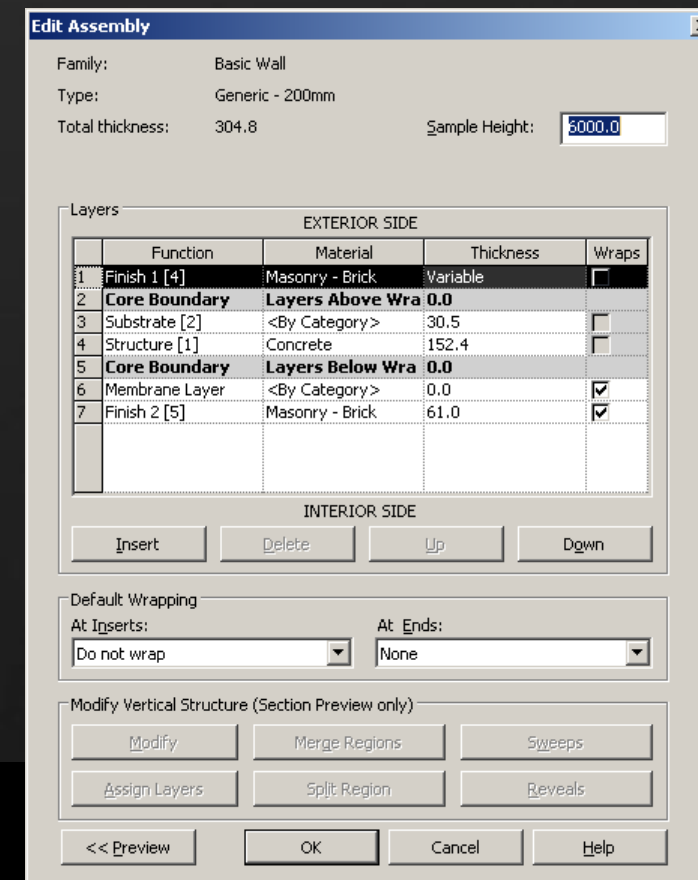
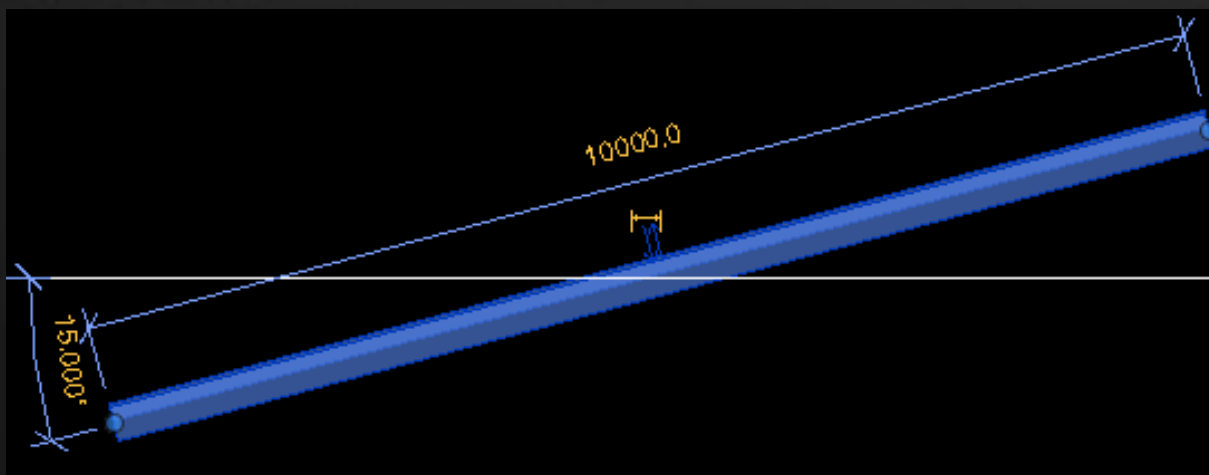
- 读写并创建所有的复合材料层
- 包含垂直分区以及分隔条和分隔缝
- Layers 属性被 GetLayers 和 SetLayers methods取代
- 增加函数使CompoundStructureLayer可以被编辑

墙分隔条

- 新的 WallSweep 类表示墙分隔条或分隔缝
- 从 compound structure.AddWallSweep 中添加到墙上
- WallSweep.Create 创建独立的分隔条/缝

板状多层结构以及分隔条/缝演示

- CompoundStructure SDK 例子
- 为墙创建一个垂直复合结构
- 在墙类型中创建并赋值新的层
- 展示了下面函数的用法:
 - HostObjAttributes 方法: GetCompoundStructure and SetCompoundStructure
 - CompoundStructure 方法: SetLayers, SetNumberOfShellLayers, SetParticipatesInWrapping, GetSegmentIds, GetAdjacentRegions, GetSegmentEndPoints, SplitRegion, FindEnclosingRegionAndSegments, AddWallSweep



Revit MEP 2012 API

占位符，保温层，里衬，管道设置

Revit MEP 2012 API 新功能

- 主要新功能
 - 管道的设置和大小
 - 管道, 风管占位符 (MechanicalUtils.ConvertDuctPlaceholders 以及 PlumbingUtils.ConvertPipePlaceholders)
 - 管道, 风管的保温层和里衬
- 其它新功能
 - Spare 和 space 回路
 - 托架和线管 domain
 - Connector, MEPSystem 和 Space 的属性
 - 断开连接警告
- 与 MEP 有关的 APIs
 - 详细能耗分析模型 API
 - 概念 (初略) 能耗分析 API

管道的设置和大小

- MEP 管道的设置现在可以通过编程实现
- 读写访问权
- 静态方法 `PipeSettings.GetPipeSettings` 和 `PipeSizeSettings.GetPipeSizeSettings` 获得管道设置以及管尺寸修改

风管, 管道占位符

- 占位符标识计划的布置
- 系统布置能包含最少的信息, 最大的灵活性
- 后面可以转换成真正的风管或管道
- 可以赋予尺寸, 倾斜度以及其它属性
- MepPlaceHolder 例子包含在本讲座的附带资料中
 - 两个命令: CreatePlaceholders 和 ConvertPlaceholders

保温层和里衬

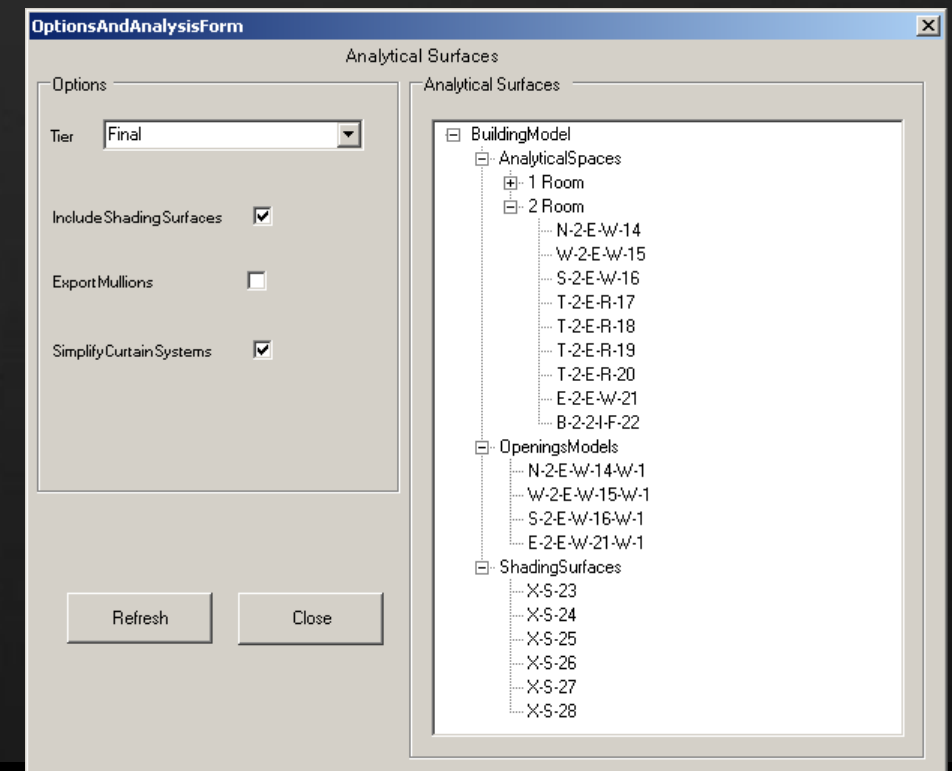
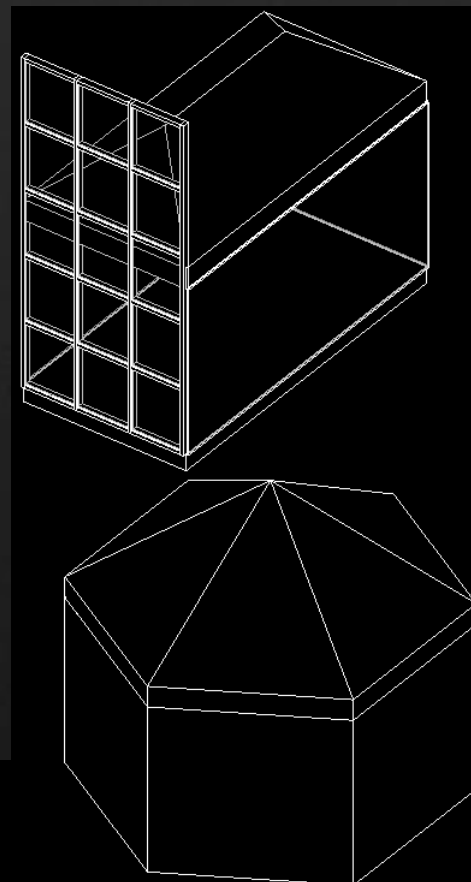
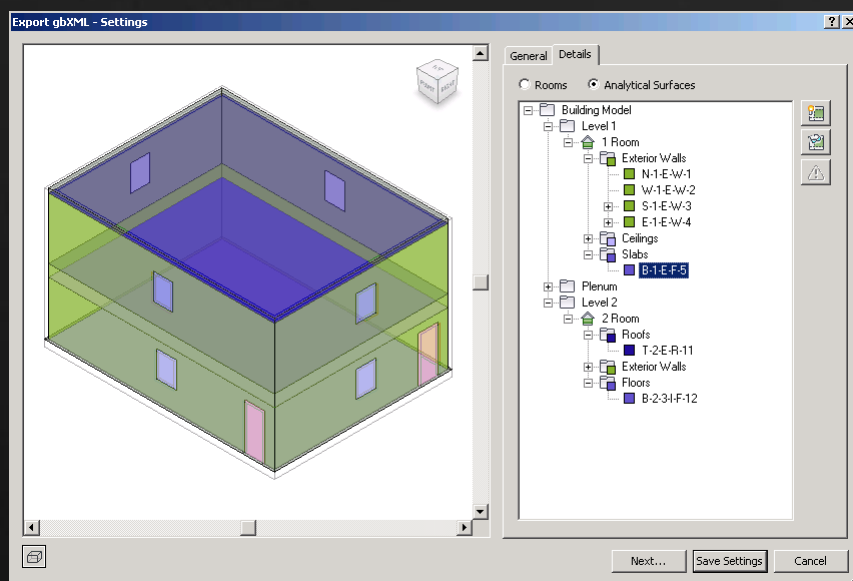
- 之前，族里定义了保温层和里衬，一式三份的几何体
- 现在保温层和里衬可以编程添加
- DuctInsulation, PipeInsulation, DuctLining 三个新类
- 适用于风管，管道以及接头
- 支持读写和创建操作
- 与父对象相关联，可以作为独立对象来访问

其它增强和变化

- Spare 和 space circuits
 - 新的ElectricalSystem.CircuitType 属性表示了电路的属性 (circuit, spare or space); 当系统是spare 或space circuit, ElectricalSystem.AddToCircuit 方法抛出异常,
- 托架和线管的domain
 - Autodesk.Revit.DB.Domain 枚举新增加了DomainCableTrayConduit来代表托架和线管
- 连接件Connector
 - 添加新的只读属性 JointType, GenderType 和 EngagementLength
- MEPSystem
 - 增加新属性MEPSystem.IsEmpty, 判断是否系统包含对象
- 图形方式警告没有连接好的地方
 - 通过Application的属性, 可以打开/关闭警告机制
- 空间Space属性
 - 新属性 Space.BaseHeatLoadOn 表示 Space.LatentHeatGainperPerson 和 Space.SensibleHeatGainperPerson 这两个属性是采用默认值还是用户定义值

详细能耗分析模型 API

- 从物理模型生成热分析模型
- 获取能耗分析详细模型，以树状形式呈现
- 与导出到 gbXML 和 Heating and Cooling Loads相似
- 分析模型由spaces, zones 和 planar surfaces
 - EnergyAnalysisDetailModel.Create()
 - Methods GetAnalyticalSpaces, Surfaces, Openings, ShadingSurfaces
- 演示: EnergyAnalysisModel SDK 例子



概念能耗分析 API

- 概念能耗分析基于建筑概念模型
- 新的重载函数 `Document.Export` 来获取
- `MassGBXMLExportOptions` 参数
- 生成一个 `gbXML` 文件
- 需要模型中有一个概念体量族实例

SDK中的新例子

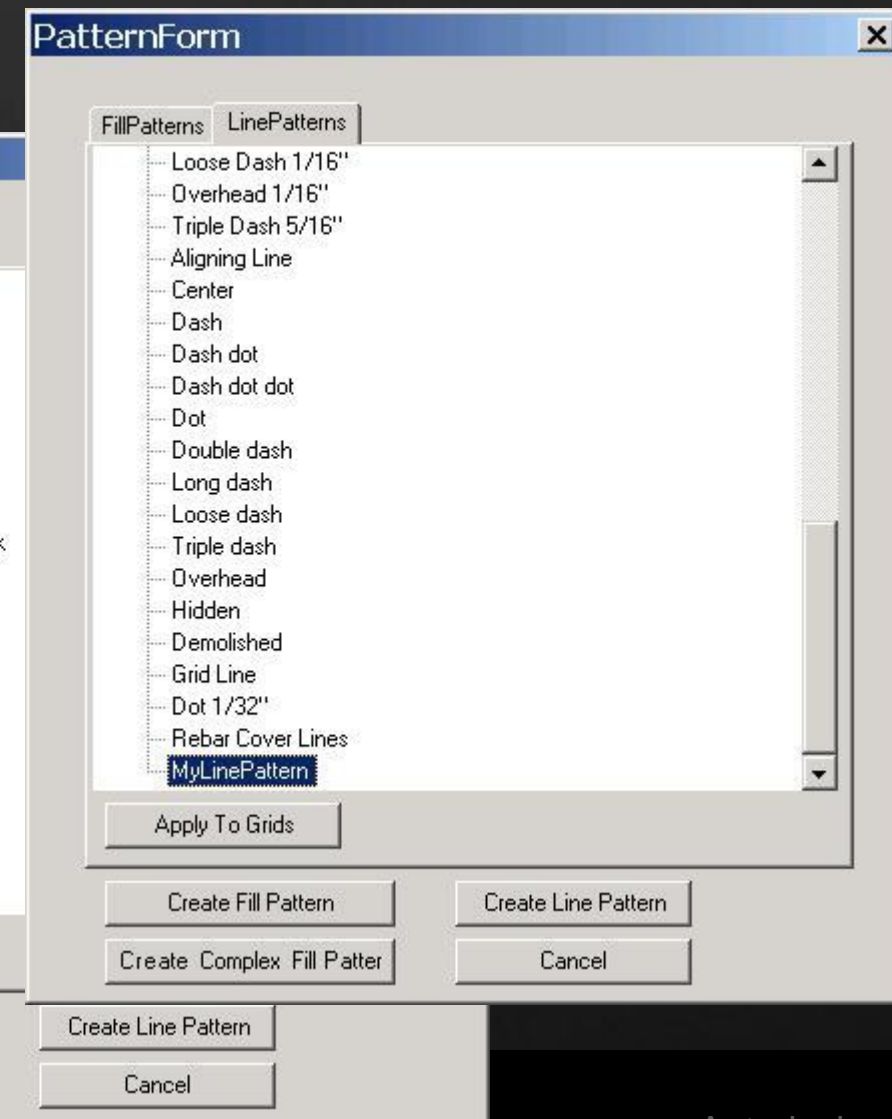
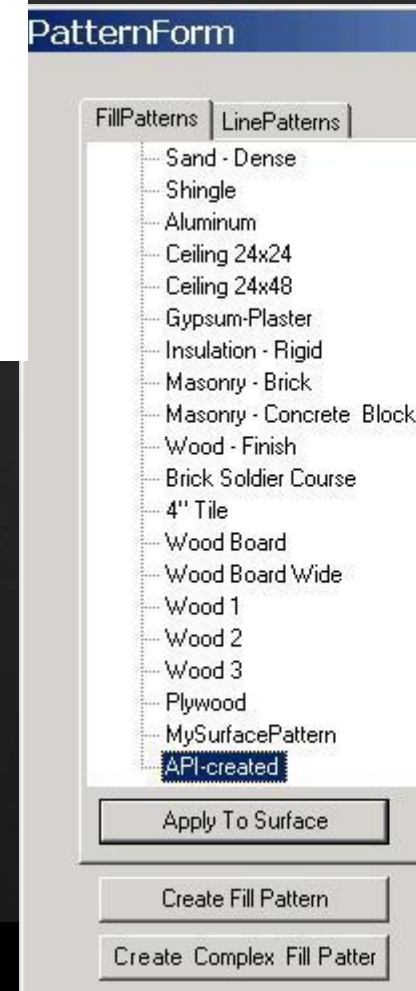
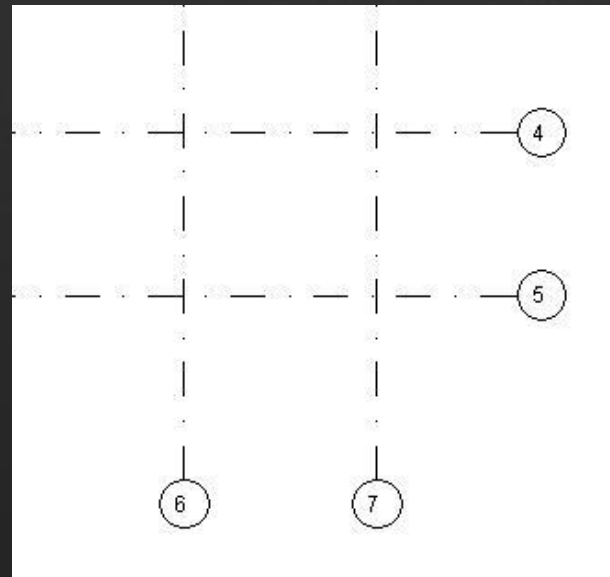
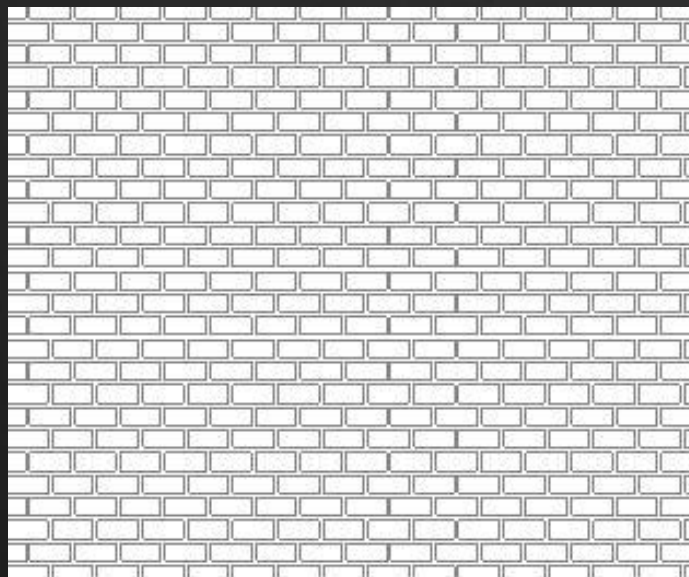
快速了解新例子功能

New Samples in Revit 2012 SDK

- CreateFillPattern
- CompoundStructure
- EnergyAnalysisModel
- ExtensibleStorageManager
- PerformanceAdviserControl
- PointCloudEngine
- MultiplanarRebar
- MultithreadedCalculation
- SlaveSymbolGeometry
- GeometryCreation_BooleanOperation
- ProximityDetection_WallJoinControl

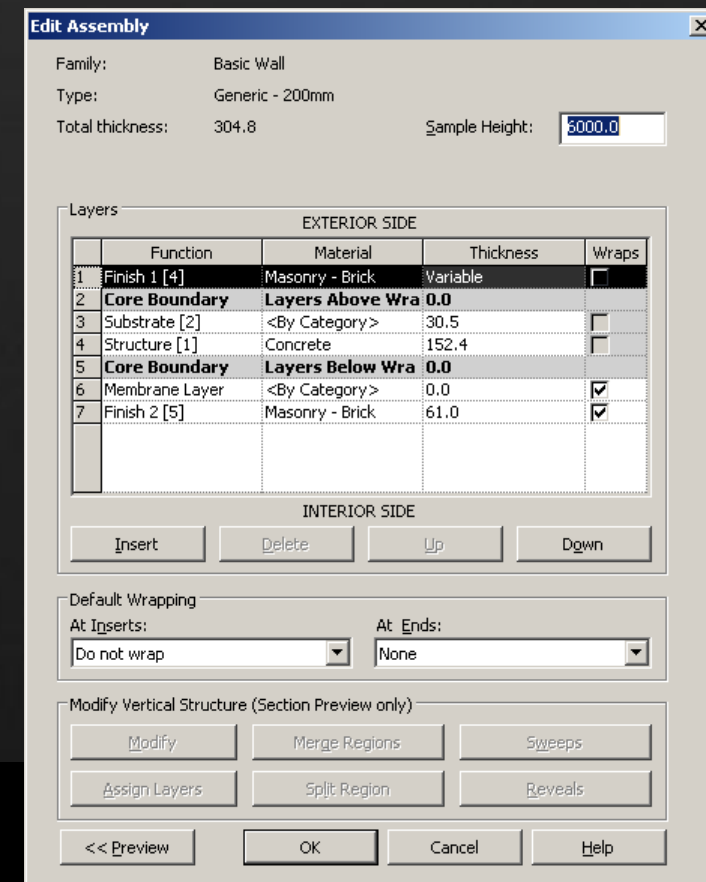
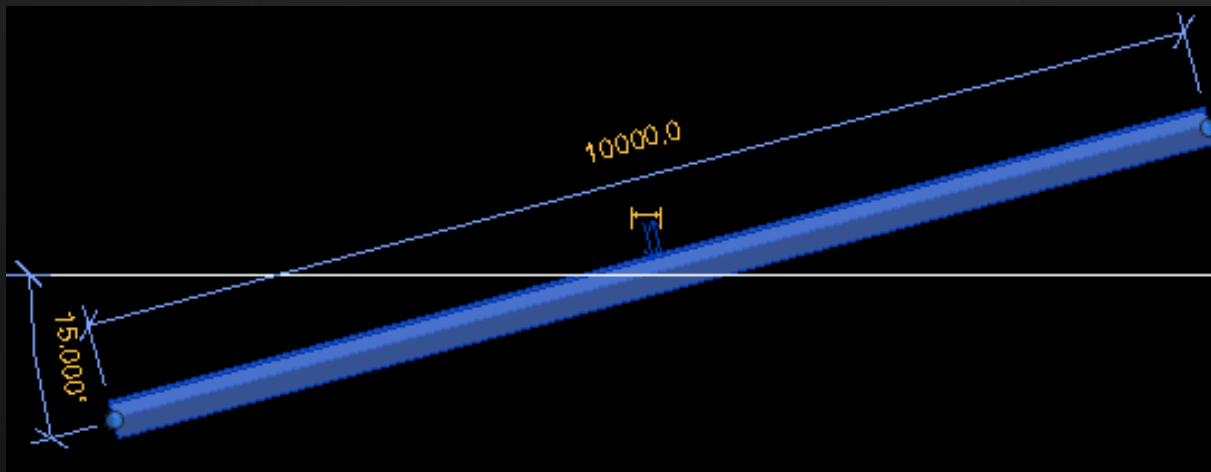
Create Fill Pattern

- 创建和读取填充样式和线型样式
- 然后应用到选中的面或轴线



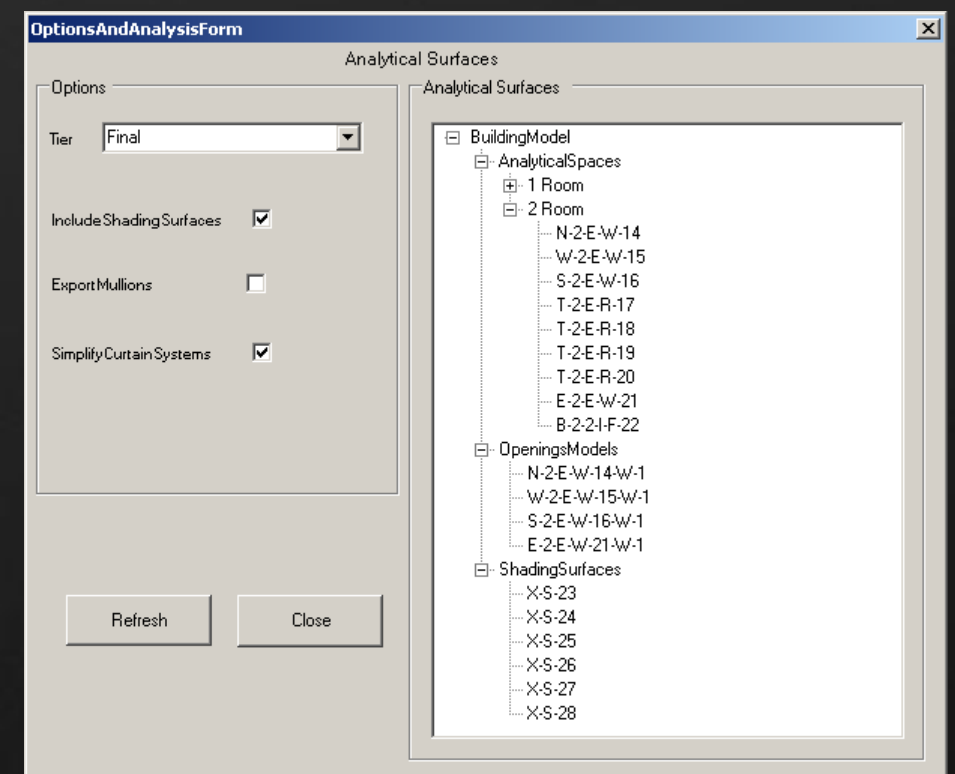
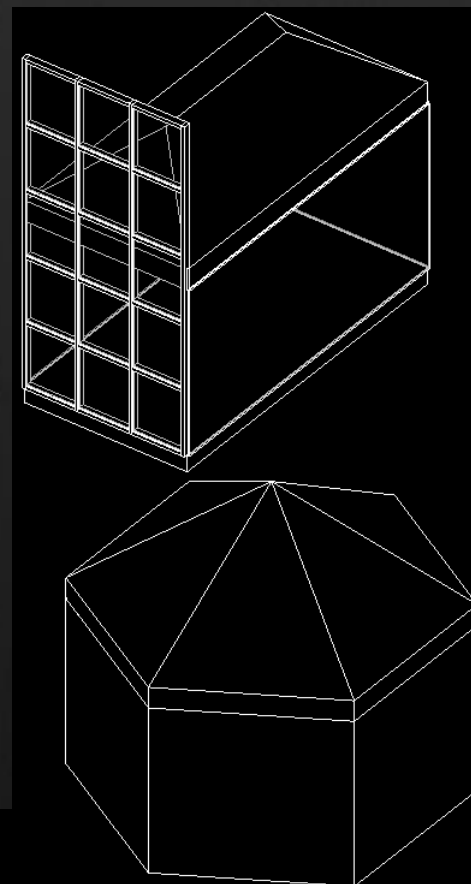
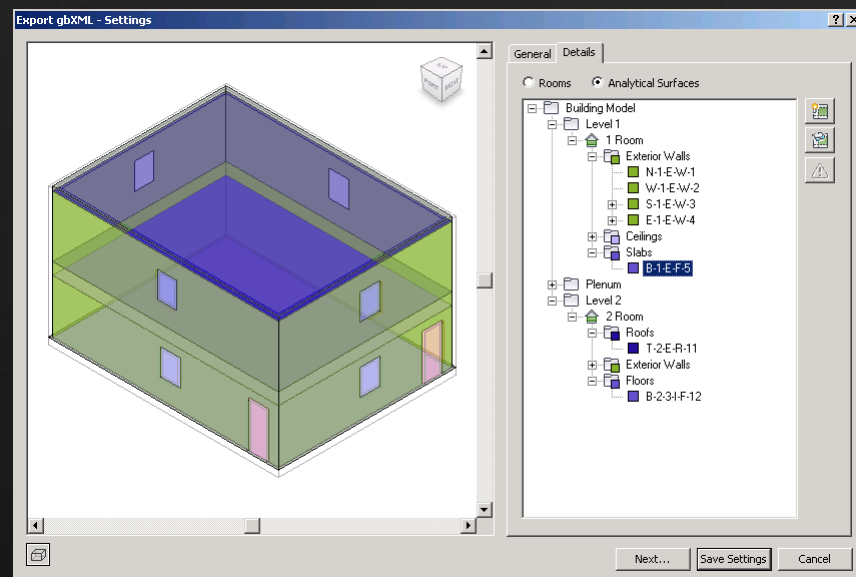
板状模型的多层材料

- 为墙创建一个垂直复合结构
- 在选中的墙类型上创建并复制新的复合结构




EnergyAnalysisModel

- 从建筑物理模型生成能耗分析模型
- 获取能耗分析详细模型，并且以树状显示结果
- 结果导出如同导出 gbXML 一样
- 模型由spaces, zones 和 planar surfaces组成



ExtensibleStorageManager

 Extensible Storage Manager

This application's Id

dec00020-2010-4338-8de4-addcadcadcad

Schema Data

Path

SchemaId

b67588cb-0001-28c5-24a4-a0f1b762a207

New Id

Name

A schema name

Documentation

Schema documentation

VendorId

adsk

ApplicationId

dec00020-2010-4338-8de4-addcadcadcad

Read Access

☐ Application ☒ Public ☐ Vendor

Write Access

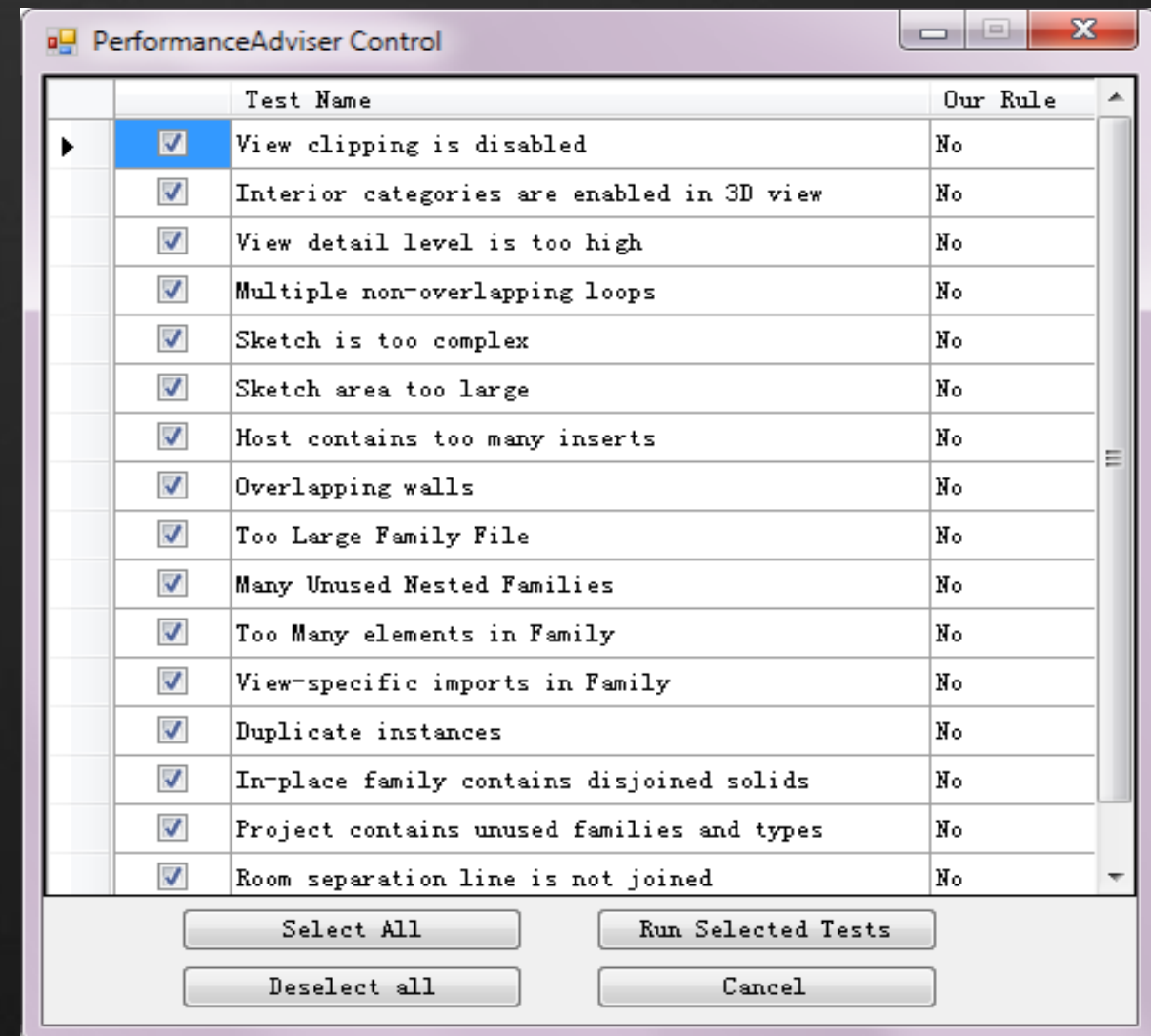
☐ Application ☒ Public ☐ Vendor

Commands

Create new schema, store data, save schema Xml - Simple	Look up schema and extract entity data
Edit Existing Data - Simple	Create SchemaWrapper from Schema
Create new schema, store data, save schema Xml - Complex	Create SchemaWrapper from Xml

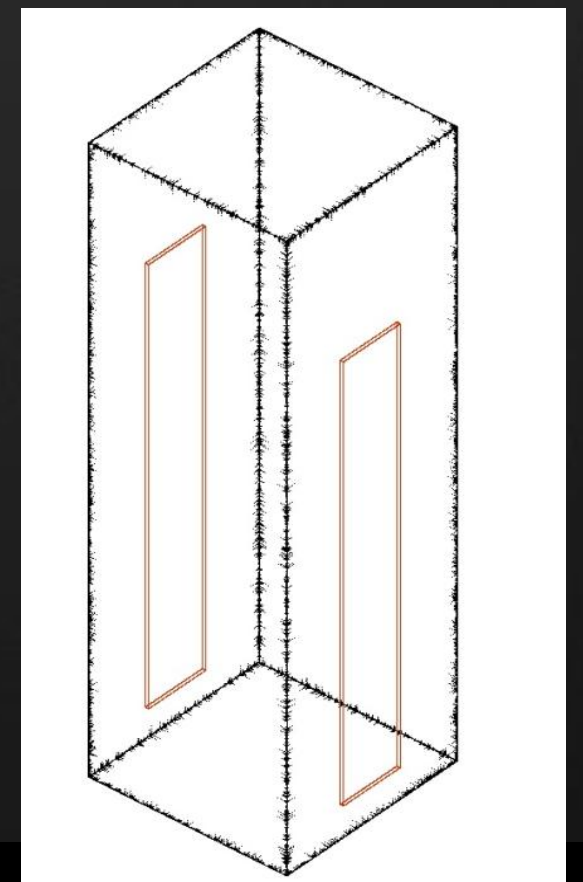
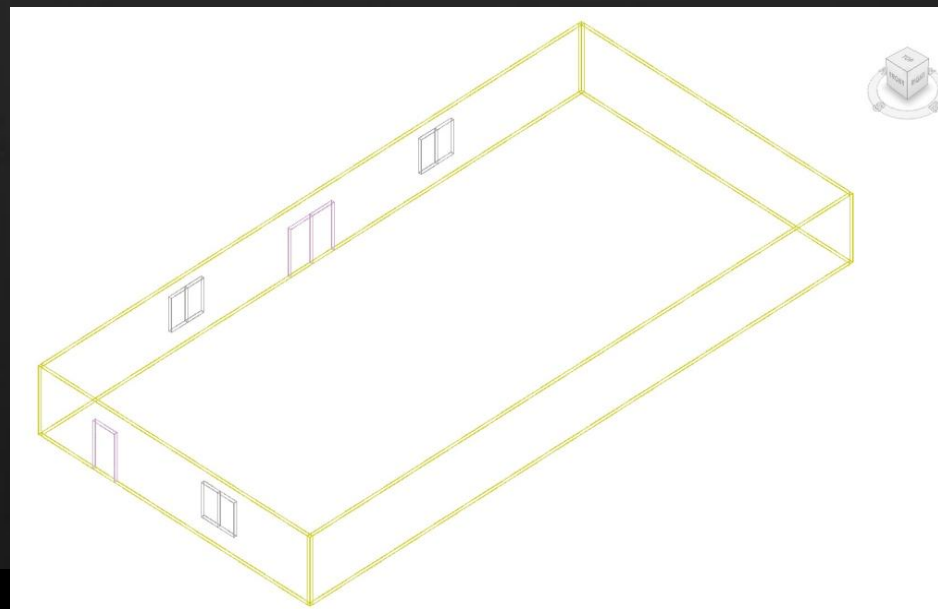
PerformanceAdviserControl

- 展示 PerformanceAdviser API 的用法
- 选择规则运行到对象，性能以及标准测试
- 当有对象满足选中的规则标准时，显示警告
- 创建注册自定义的规则



PointCloudEngine

- 例子演示了如何创建点云引擎
- 实现了两种点云引擎类型(基于文件和不基于文件的点云)
- 点云引擎需要实现的接口
 - IPointCloudEngine
 - IPointCloudAccess
 - IPointSetIterator
- 注册点云引擎
- 导出点云



MultiplanarRebar

- 演示了如何给倾斜牛腿配钢筋
- 创建了3D钢筋形状, 垂直, 箍筋

Corbel Reinforcement Options

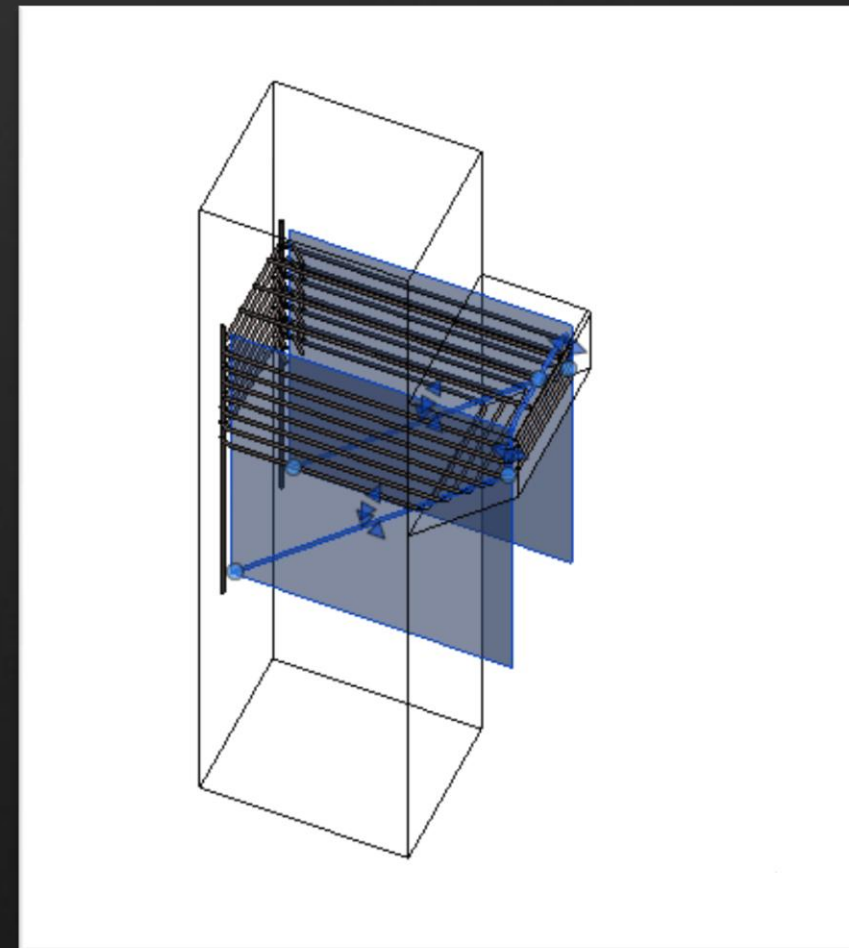
Host Straight Bars
Bar Type: #4

Top Bars
Bar Type: #4 Bar Count: 3

Stirrup Bars
Bar Type: #4 Bar Count: 3

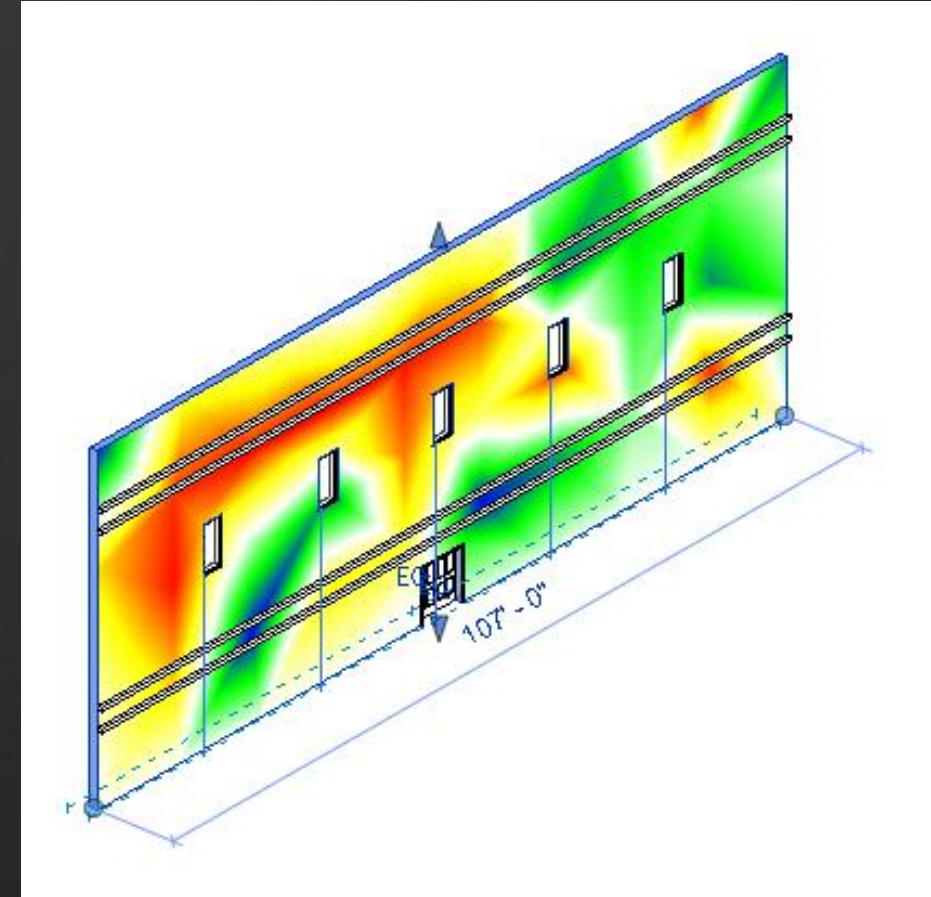
Multiplanar Bars
Bar Type: #4

OK Cancel



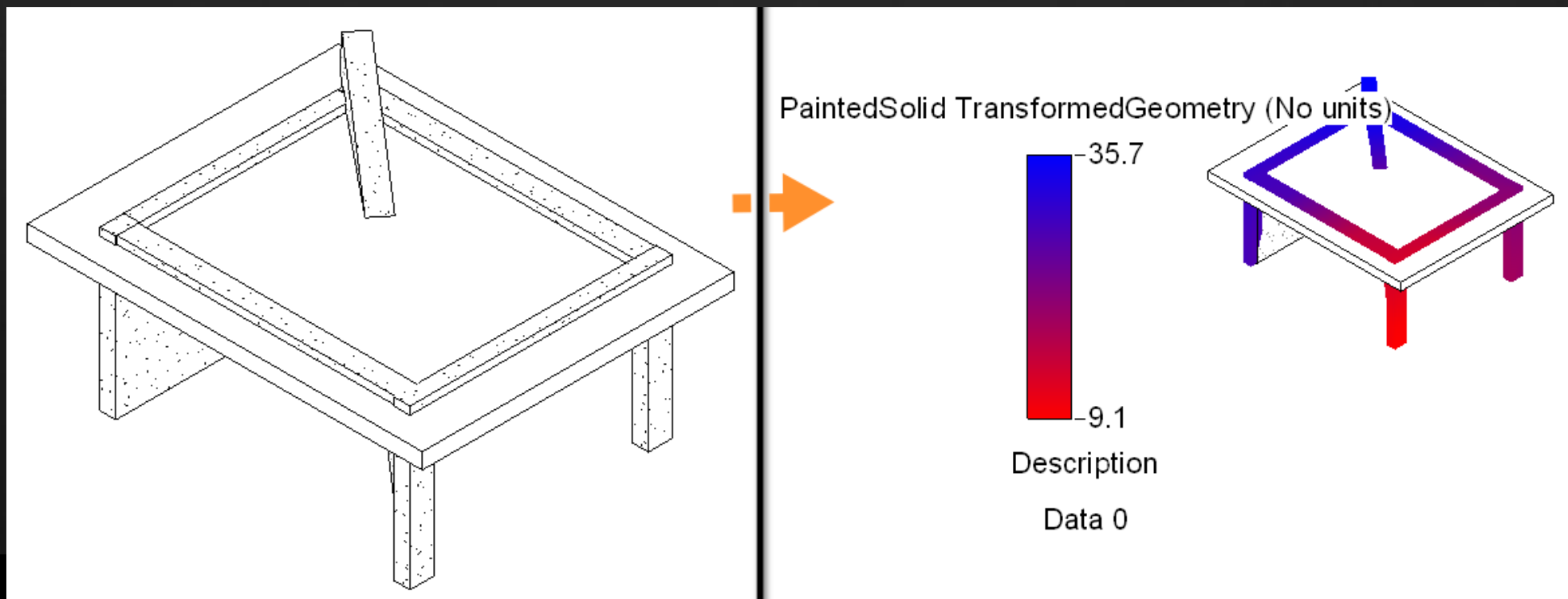
MultithreadedCalculation

- 在独立的线程中创建分析数据
- 显示数据到预先选择的对象上
- 使用当前视图的 SpatialFieldManager来显示结果
- 实现Iupdater接口来更新分析数据



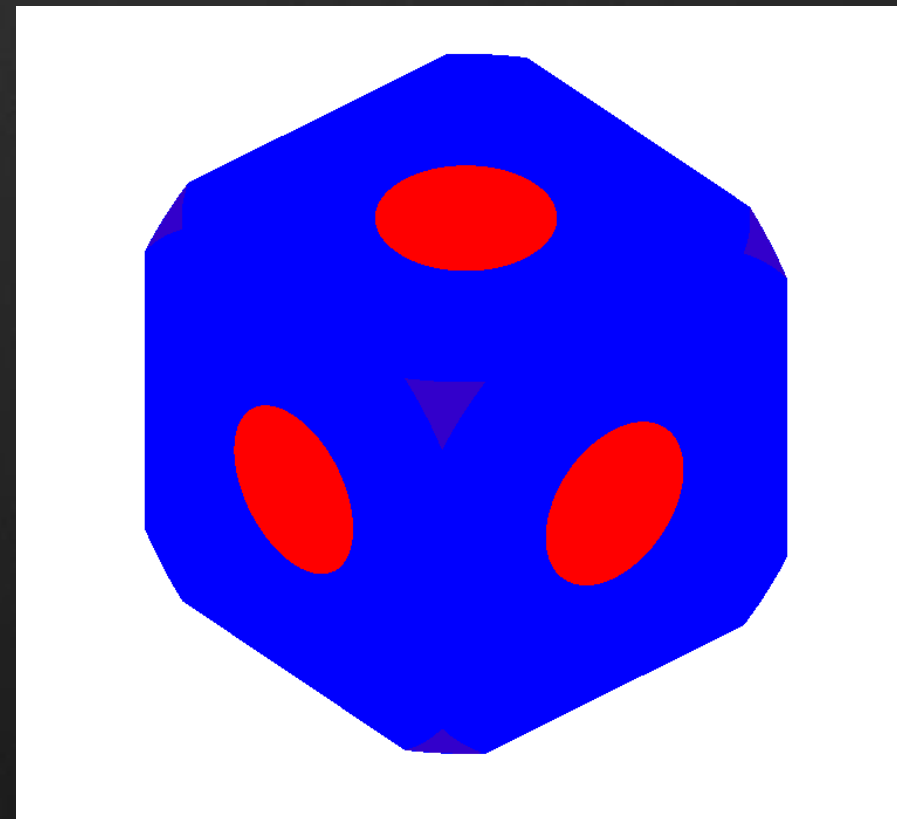
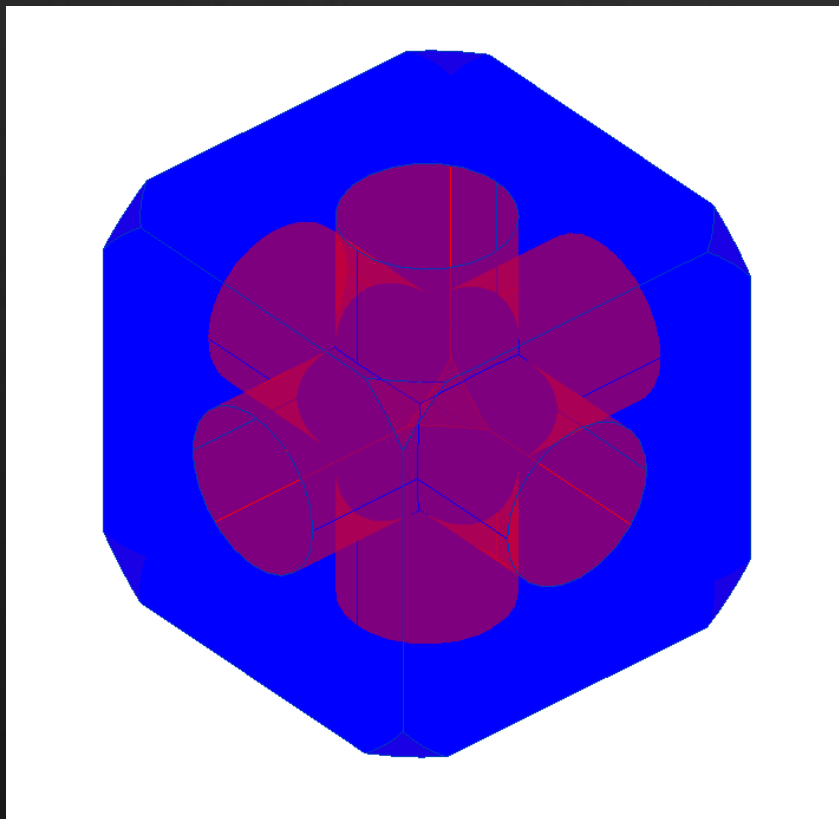
SlaveSymbolGeometry

- 读取族实例所隶属的多种几何数据(原始, 切割或连接后)
- 演示了如何访问一个族实例的隶属几何信息
- 使用 SpatialFieldManager 来喷绘几何体
- 从 FamilyInstance 的方法获得几何数据
- `get_Geometry()`, `GetOriginalGeometry()`, `GetTransformed()`
- 不同种类的几何数据放在不同的视图中



GeometryCreation_BooleanOperation

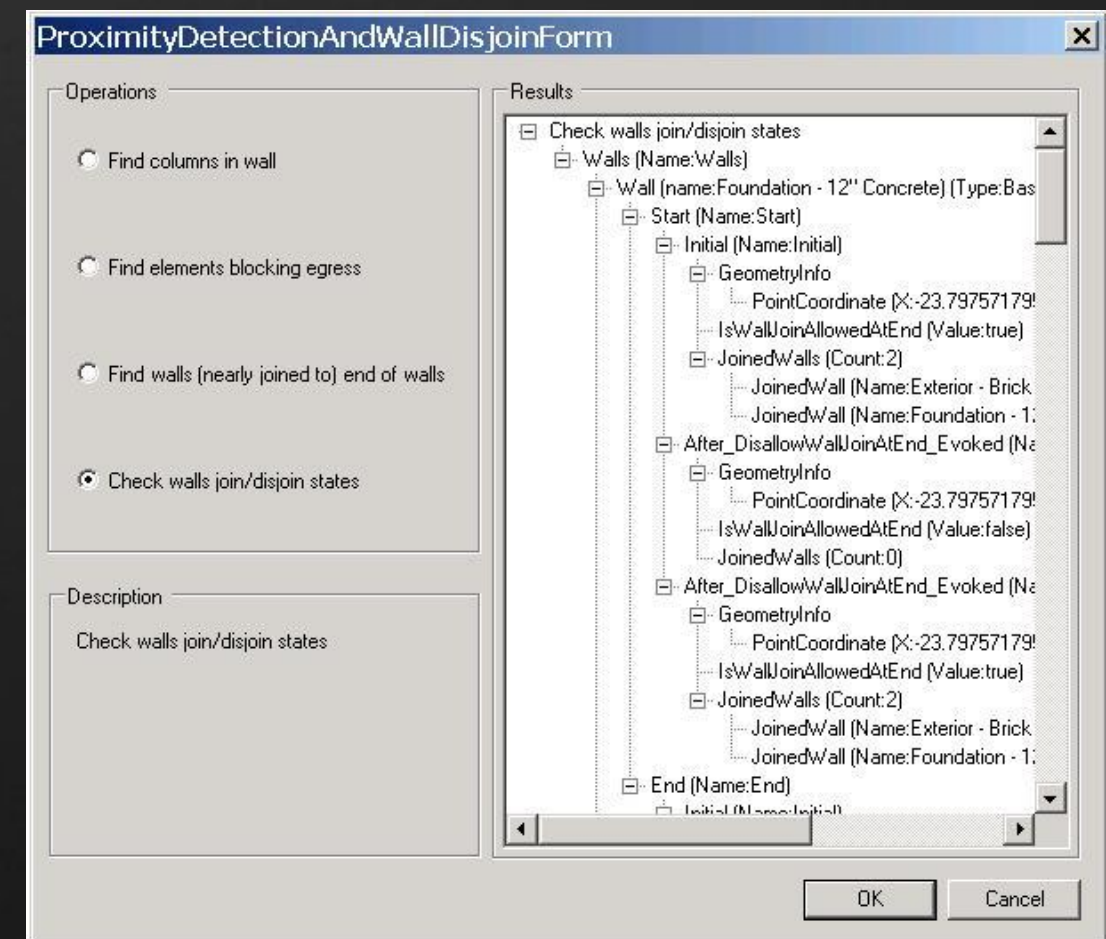
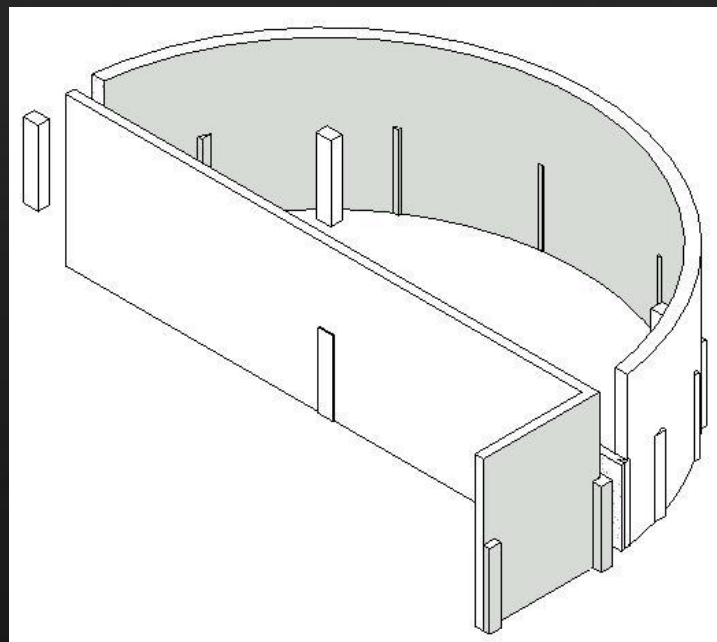
- 创建几何实体(在内存中)
- 使用几何布尔运算
- 创建Constructive solid geometry
- 使用分析可视化框架显示几何体



ProximityDetection_WallJoinControl

- 演示了如下四种功能
 - 1) 从墙中找到柱子
 - 2) 找到挡住门的对象
 - 3) 找到一个墙的相邻墙
 - 4) 检查墙的连接状态(连接or没有连接)

■



总结

总结

Revit 2012 API和SDK概览

移植2011的插件到 Revit 2012

Revit 2012 新 APIs

- 几何属性访问 API
- 建造模型
- 点云
- Revit Structure API 新功能

Break

- 扩展存储
- 板状构件的多层构造访问
- Revit MEP API新功能

新的 SDK samples

深入学习

SDK中的资源

Revit 2011 API DevTV中文

- http://download.autodesk.com/media/adn/Revit_2011_API_DevTV_Chinese.zip
- 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>

Revit API 培训和免费现场服务

Revit 二次开发DevLabs (二天)

- Aug 1 北京, Autodesk的会议室(国贸大厦1座7层)

Revit API 教室培训

- July 28 & 29 , 2011 北京

如果想参加, 请在网站登记注册:

Developer Center >> API Training and Consulting >> Schedule

http://www.adskconsulting.com/adn/cs/api_course_sched.php

Revit API 期望调查, 2011

- <http://www.zoomerang.com/Survey/WEB22C9QACTE6V/>

最后提交日期: **June 3, 2011**

今日培训实况可下载...

今日培训过程被录制, 并将发布到网上:

ADN 网站

<http://adn.autodesk.com>

Revit > Knowledgebase > Whitepapers and Training Videos

官网开发者中心 (Developer Center)

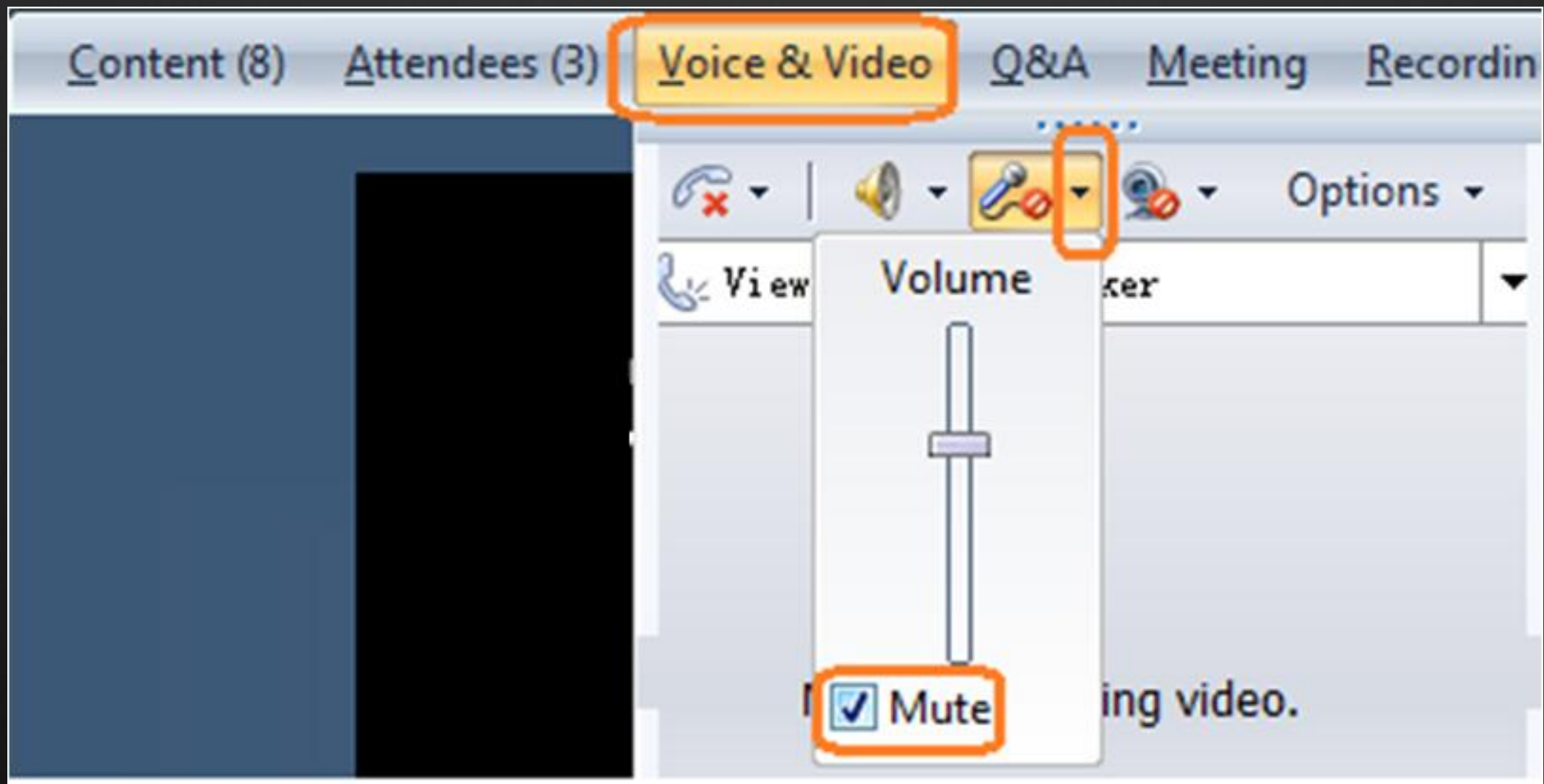
API Training & Consulting > Schedule

<http://www.autodesk.com/developer>

调查

问答

- 打开计算机麦克，参与者都能听到提问





Thank you!



Autodesk®