



電話番号 (有料) 03-5539-5161
(無料) 0034-800-400652
参加者コード 9672646

Revit 2010 API

*Katsuaki Takamizawa Japan Manager,
Developer Technical Services*

*Akira Kudo
Developer Technical Services*


Autodesk

はじめに


はじめる前に

LiveMeeting と音声会議 の使用方法


フルスクリーン モード




フルスクリーンに拡張
ESC で解除



フィードバック



リアルタイム Q&A



電話をミュートする *6

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 2

Autodesk

はじめに

はじめる前に

音声が聞こえない方は下記の番号に電話して音声会議にご参加ください

電話番号（有料）03-5539-5186
（無料）0044-22-132431

参加者コード 4673390

注意: インターネット音声は使用できません

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 3

Autodesk

はじめに

アンケートにご協力下さい

1. Revit 製品の経験は？
2. Revit APIの経験は？
3. 下記のどれがあなたに該当しますか？
 - ① このトピックは直接自身の業務の関連している
 - ② このトピックは将来的に自分に有益であると期待している
 - ③ 現在このテクノロジーを評価中である
 - ④ 該当無し

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 4

Autodesk

はじめに

Agenda

はじめに

- 製品、Software Development Kit (SDK)

スタートアップ – Hello World

- 開発環境、外部コマンドとアプリケーションインタフェース、Revit.ini、RvtSamples と RvtMgdDbg

データベースと要素

- 要素の特定、全てのモデル要素の取得、フィルター、操作

ファミリーとタイプ

- 標準 vs システムファミリー、ロード、タイプ変更

パラメータ

- 組み込み vs 共有、外部アプリケーションとのやり取り、非表示ドキュメント毎のデータ

ジオメトリとグループ

- 2D と3D ジオメトリ、部屋、グループ

API アップデート

- リボン、イベント、VSTA

API 新機能

- 新しいサンプル
- ファミリーAPI とフォーム作成

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 5

Autodesk

はじめに

略語

▪ ADN	Autodesk Developer Network
▪ AEC	Architecture, Engineering, Construction
▪ API	Application Programming Interface
▪ BIM	Building Information Model
▪ MEP	Mechanical, Electrical, Plumbing
▪ RAC	Revit Architecture
▪ RME	Revit MEP
▪ RST	Revit Structure
▪ SDK	Software Development Kit

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 6

Autodesk

はじめに

Revit 製品ライン

3種類のAutodesk Revit 製品

- 建築 Revit Architecture
- 設備 Revit MEP (Mechanical, Electrical, Plumbing)
- 構造 Revit Structure

製品ビルドと配布方法

- DVDバージョン — ADNサイトにポスト
 - Software & Support > Revit > Downloads
- Webバージョン — Autodesk ホームページ
 - Products > Revit Architecture, Structure, MEP > Product Download
 - パブリックの製品サイトから最新バージョンがダウンロード可能
 - Webバージョンには修正が常時盛り込まれる (別途“パッチ”のサービスパックは現在英語バージョンのみ。今後パッチ形式になる予定)

Copyright © 2009 Autodesk Inc.

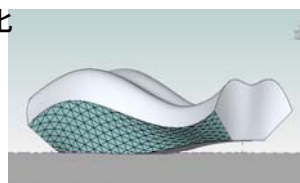
Revit Programming Introduction 7

Autodesk

はじめに

Revit Architecture 2010 製品機能

- コンセプト デザイン – 複雑なジオメトリ作成の簡略化
 - 直観的で柔軟なフォームの生成
 - スケッチの強化
 - 分割、パターンとパネルの適用
 - 統合されたワークフロー
- ユーザーインターフェース – 生産性の向上
 - リボン、アプリケーション メニュー、クイックアクセスツールバー
 - 強化されたツールチップ
- パフォーマンスと相互利用
 - ネイティブな 64ビット サポート
 - AutoCAD Civil 3D に書き出し
 - Autodesk Inventor からコンテンツの読み込み
 - gbXML の改善
 - API の強化



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 8

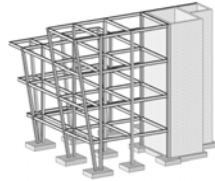
Autodesk

はじめに

Revit 2010 製品機能


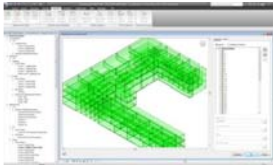

Autodesk Revit Structure 2010

- モデリングと解析の強化
 - 斜め柱
 - **S** ブリッジ モデリング
 - **S** スチール接続モデリング
 - **S** コンセプト形状解析



Autodesk Revit MEP 2010

- サステナブル・デザインと解析
 - ネイティブな冷暖房負荷エンジン



Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 9

はじめに

Revit SDK

製品と共にSDKを提供

- “インストールツールとユーティリティ”メニューからインストール

Revit 2010 テンポラリーフォルダにインストーラー

- C:\¥Autodesk¥RAC_2010_Japanese_Win_32bit¥support¥SDK¥RevitSDK.exe

*デベロッパセンターより最新版をご利用下さい

- <http://www.autodesk.com/developer> > Products & Technologies > Autodesk Revit

RevitAPI.dll は インストールされた全てのRevit製品に存在

Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 10

はじめに

SDK ドキュメント

- Read Me First.doc
- Getting Started with the Revit API.doc
- Revit Platform API Changes and Additions.doc
- Revit 2010 API Developer Guide.pdf *更新済み
- Revit API Class Diagram.png
- RevitAPI.chm
 - What's New 有り
- Ribbon Guidelines
 - Autodesk Icon and Ribbon design guidelines.pdf
 - Ribbon design guidelines.pdf
- Add-In Manager
- VSTA Samples
- Samples
 - Converting 2009 applications to Revit 2010 API.doc
 - Revit 2010 New Samples.doc
 - SamplesReadMe.htm
 - SDKSamples2010.sln

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 11

Autodesk

Autodesk

スタートアップ

Hello World

スタートアップ

Revit を拡張する

Revit を拡張する方法

1. 外部コマンドを追加
 - IExternalCommand を実装
 - Ribbon UIにコマンドが追加される
 - アドイン タブ > 外部ツール
2. 外部アプリケーションを定義
 - IExternalApplication を実装
 - アプリケーションはRibbon パネル、アイテム、ボタンが追加可能
 - 外部アプリケーションは外部コマンドを使用可能、1. は2. のサブセット

両方とも Revit.ini 内に指定

3. VSTA (Visual Studio Tools for Application) マクロの実装
 - 2つタイプのマクロ: アプリケーションとドキュメントレベル
 - 2、3の例外を除き、外部コマンドと文法そして機能的にはほとんど同じ
 - イベントはサポートされない

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 13

Autodesk

スタートアップ

Revit API DLL

- .NET API
 - Microsoft Visual Studio 2008
 - Microsoft .NET Framework 3.5
 - “[Revit Xyz YYYY]¥Program¥RevitAPI.dll” を参照
 - ‘Copy Local’ と Falseに設定
 - C# /VB.NET/managed C++ - .NET コンプライアント言語が使用可能
- Revit Architecture/Structure /MEP 製品
 - 同じ API DLL
 - 特定機能はArchitecture、MEP、またはStructureのみで使用可能

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 14

Autodesk

C# と VB.NET

スタートアップ

- C# とVB.NET は同等
- 生成されたILコードは同一
- 自動トランスレーターが利用可能
- Google で“c# vb.net translator”と検索
- Reflector はIL をC#, VB そしてmanaged C++ にデコンパイル
- 多くのSDK サンプルは C#
- VB.NET のサンプルもSDKに含まれる
- 今回の解説では C# とVB.NETの両方を使用

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 15

Autodesk

外部コマンド

スタートアップ

- Autodesk.Revit.IExternalCommand インタフェースを実装
- IExternalCommand.Execute メソッドを実装

```
Public Class ApplyParameter _  
    Implements Autodesk.Revit.IExternalCommand  
  
    Public Function Execute( _  
        ByVal commandData As Autodesk.Revit.ExternalCommandData, _  
        ByRef message As String, _  
        ByVal elements As Autodesk.Revit.ElementSet) _  
        As Autodesk.Revit.IExternalCommand.Result _  
    Implements Autodesk.Revit.IExternalCommand.Execute  
    '...  
End Class
```

VB.NET

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 16

Autodesk

スタートアップ

コマンドからの戻り値

```
public class Command : IExternalCommand
{
    public IExternalCommand.Result Execute(
        ExternalCommandData commandData,
        ref string message,
        ElementSet elements )
    {
        try
        {
            // . . .
        }
        catch( Exception ex )
        {
            message = ex.ToString();
            return IExternalCommand.Result.Failed;
        }
        return IExternalCommand.Result.Succeeded;
    }
}
```

C#

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 17

Autodesk

スタートアップ

外部コマンド - Revit.ini を変更

```
[ExternalCommands]
ECCount = 3

ECName1 = ApplyParameter
ECClassName1 = Revit.SDK.Samples.FireRating.VB.NET.ApplyParameter
ECAssembly1 = FireRating.dll
ECDescription1 = Add a new shared parameter to revit and apply it to all doors

ECName2 = ExportFireRating
ECClassName2 = Revit.SDK.Samples.FireRating.VB.NET.ExportFireRating
ECAssembly2 = FireRating.dll
ECDescription2 = Export fire rating values to an Excel file

ECName3 = ImportFireRating
ECClassName3 = Revit.SDK.Samples.FireRating.VB.NET.ImportFireRating
ECAssembly3 = FireRating.dll
ECDescription3 = Import fire rating values from the Excel file and apply them to doors
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 18

Autodesk

外部アプリケーション

- Autodesk.Revit.IExternalApplication インターフェースを実装
- OnStartup() と OnShutdown() メソッドを実装

```
public class App : IExternalApplication
{
    public IExternalApplication.Result
    OnStartup( ControlledApplication application )
    {
        CreateRibbonSamplePanel(application);
        CreateRibbonInfosPanel(application);
        return IExternalApplication.Result.Succeeded;
    }

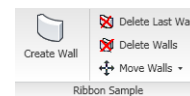
    public IExternalApplication.Result
    OnShutdown( ControlledApplication application )
    {
        return IExternalApplication.Result.Succeeded;
    }
}
```

- OnStartup() で UIを追加し、外部コマンドやイベントハンドラーと関連付ける
- セパレーター、プッシュ・プルダウンボタンを保持したパネルを追加可能
- Ribbon SDK のサンプルを参照

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 19

スタートアップ



Autodesk

外部アプリケーション - Revit.ini を変更

```
[ExternalApplications]
EACount=3

EAClassName1=RvtMgdDbg.App
EAAssembly1=C:\Program Files\Autodesk Revit Architecture 2010\Program\RvtMgdDbg.dll

EAClassName2=RvtSamples.Application
EAAssembly2=C:\%a%\lib\revit\2010\SDK\Samples\RvtSamples\CS\RvtSamples.dll

EAClassName3=Revit.SDK.Samples.Ribbon.CS.RibbonSample
EAAssembly3=C:\%a%\lib\revit\2010\SDK\Samples\Ribbon\Ribbon.dll
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 20

スタートアップ

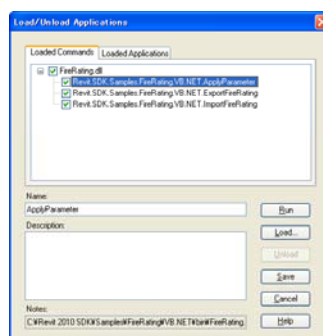
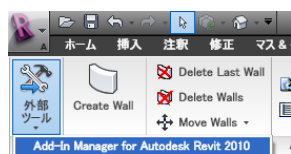
Autodesk

Add-In Manager

スタートアップ

外部コマンドや外部アプリケーションをRevit.iniファイルを直接変更せずにロード・実行する外部コマンド

- <SDK>¥Add-In Manager¥setup.exeでインストール
- “Run”で選択したコマンドを実行
- “Save”で設定を保存することでRevit.iniに変更を反映
 - Revitを再起動してUIに変更を表示



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 21

Autodesk

スタートアップ

スタートアップ

- 外部コマンド: HelloRevit と HelloWorld
- 外部アプリケーション: APIAppStartup, Toolbar
- “Hello World” 初歩的なサンプル

Lab 1-1

- どんな引数がコマンドに渡されるのか？
- コマンドからは何が返されるのか？

Lab 1-2

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 22

Autodesk

最小なコマンド

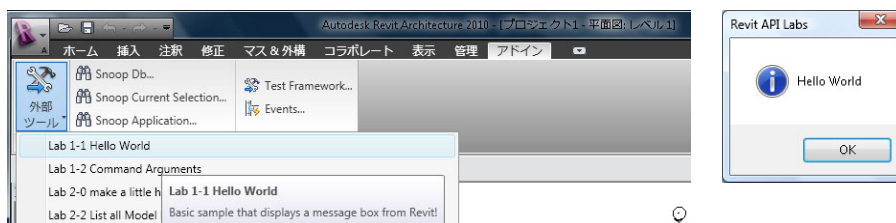
スタートアップ

```
Public Function Execute( _
    ByVal commandData As Autodesk.Revit.ExternalCommandData, _
    ByRef message As String, _
    ByVal elements As Autodesk.Revit.ElementSet ) _
    As Autodesk.Revit.IExternalCommand.Result _
    Implements Autodesk.Revit.IExternalCommand.Execute

    MsgBox("Hello World")
    Return IExternalCommand.Result.Succeeded

End Function
```

In VB.NET



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 23

Autodesk

コマンド 入力引数

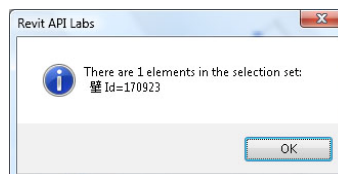
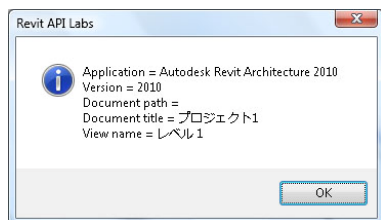
スタートアップ

```
' List the app, doc and view data
Dim revitApp As Revit.Application = commandData.Application
Dim doc As Revit.Document = revitApp.ActiveDocument
Dim view As Revit.Elements.View = commandData.View

Dim sMsg As String = "Application = " & revitApp.VersionName & " " _
    & revitApp.VersionNumber & vbCrLf
sMsg += "Document path = " & doc.PathName & vbCrLf ' Empty if not saved yet
sMsg += "Document title = " & doc.Title & vbCrLf
sMsg += "View name = " & view.Name
MsgBox(sMsg)

' List the current selection set
Dim sel As Selection = doc.Selection

sMsg = "There are " & sel.Elements.Size & " elements in the selection set:"
Dim elem As Revit.Element
For Each elem In sel.Elements
    sMsg += vbCrLf & " " & elem.Category.Name & " Id=" & elem.Id.Value.ToString
Next
MsgBox(sMsg)
```



In VB.NET

Copyright © 2009 Autodesk Inc.

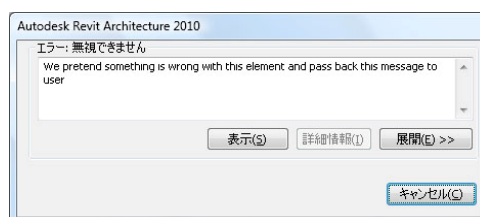
Revit Programming Introduction 24

Autodesk

コマンド出力引数

スタートアップ

```
' We pretend that something is wrong with the first element in the selection
' We pass the message back to Revit user and indicate the error result
If Not sel.Elements.IsEmpty Then
  Dim iter As ElementSetIterator = sel.Elements.ForwardIterator
  iter.MoveNext()
  Dim errElem As Revit.Element = iter.Current
  elements.Clear()
  elements.Insert(errElem)
  message = "We pretend something is wrong with this element and pass back this
  message to user"
  Return IExternalCommand.Result.Failed
Else
  Return IExternalCommand.Result.Succeeded
End If
```



Copyright © 2009 Autodesk Inc.

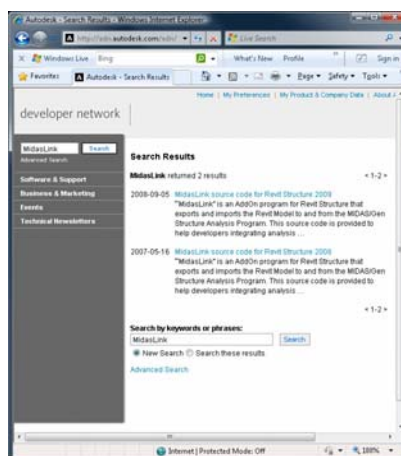
Revit Programming Introduction 25

Autodesk

アプリケーションのインストール

スタートアップ

- 一般的にインストーラーは Revit.ini を編集
- Revit のProgram フォルダーにアプリケーション配置を推奨
- MidasLink サンプルはセットアップのソースを含む



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 26

Autodesk

スタートアップ

RvtSamples

全ての外部コマンドを Revit.ini に追加!?

すべてのSDK サンプルアプリケーションを一度に登録
RvtSamples.txt
独自のエンタリーも追加可能

例題

- AdnSamples.txt
- labsで提供




Copyright © 2009 Autodesk Inc. Revit Programming Introduction 27

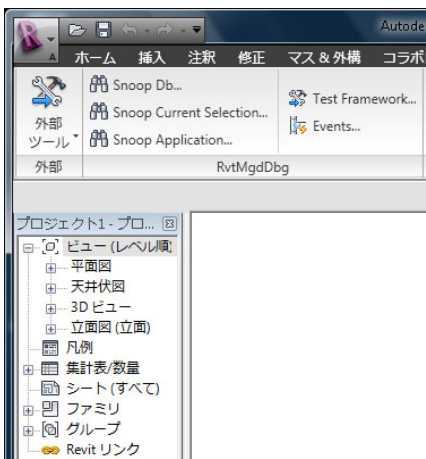
Autodesk

スタートアップ

RvtMgdDbg

ADN Web サイトから入手可能 - “RvtMgdDbg” で検索
Revit API の包括的なテスト
サンプルコードとユーティリティ クラス
簡単なテスト
Revit APIの学習を支援
調査

- アプリケーション
- ドキュメント
- 現在の選択
- リフレクション
- イベント
- テスト



Copyright © 2009 Autodesk Inc. Revit Programming Introduction 28

Autodesk

ステートアップ

RvtMgdDbg デモ

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 29

Autodesk

データベースと要素

Revit データベースのコンテンツの解析
 特定要素のためのイタレーションとフィルター
 RvtMgdDbg の使用

Autodesk

要素コレクション

要素コレクション

- オブジェクトモデル
- データベースの構造
- 要素タイプ
- アクセスと要素の特定
- タイプとカテゴリーでのフィルター
- 全てのモデル要素の取得
- モデル要素の編集

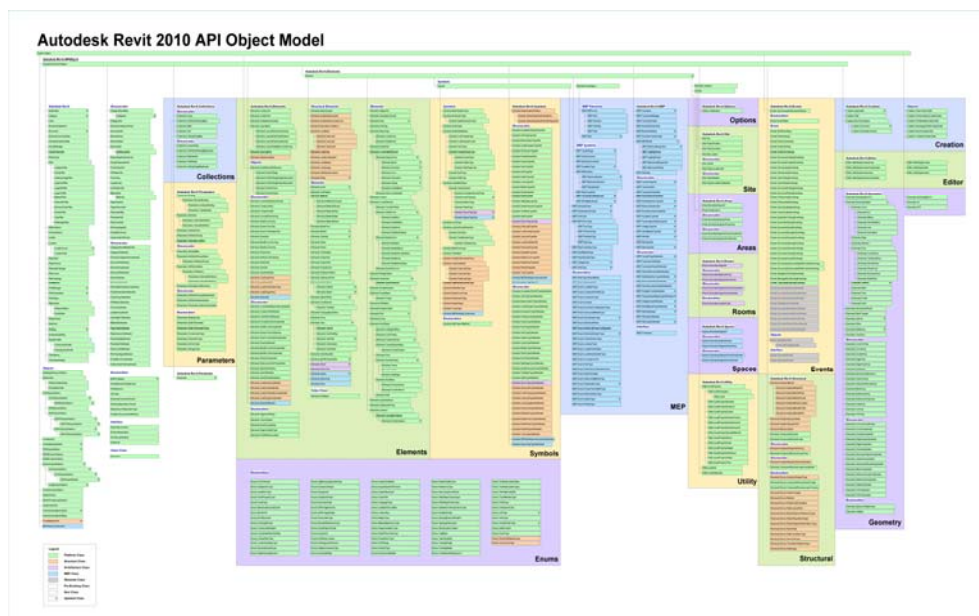
Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 31

Autodesk

Revit オブジェクトモデル

要素コレクション



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 32

Autodesk

要素コレクション

オブジェクトモデル (サブセット)

APIObject Application Document ExternalCommandData ElementSet ParameterSet Parameter Element Symbol GeometryObject # RST: AnalyticalModel AnalyticalModelFloor AnalyticalModelWall Creation APIObject Application Document	GeometryObject Curve Arc Ellipse Line NurbSpline Edge Face ConicalFace CylindricalFace HermiteFace PlanarFace ResolvedFace RuldeFace Instance Mesh Profile Solid Plane Reference Transform	Element HostObject CeilingAndFloor ContFooting Floor Wall Instance Opening TextElement BeamSystem Dimension SpotDimension FamilyBase Family Level Phase ProjectInfo # RST: BoundaryConditions LoadCase, Combination, Nature, Usage AreaReinforcement AreaReinforcementCurve LoadBase AreaLoad LineLoad PointLoad PathReinforcement Rebar # end of RST Symbol	Symbol AnnotationSymbolType BeamSystemType HostObjAttributes ContFootingType FloorType WallType InsertableObject FamilySymbol RoomTagType # RAC RebarTagType # RST
---	---	---	---

Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 33

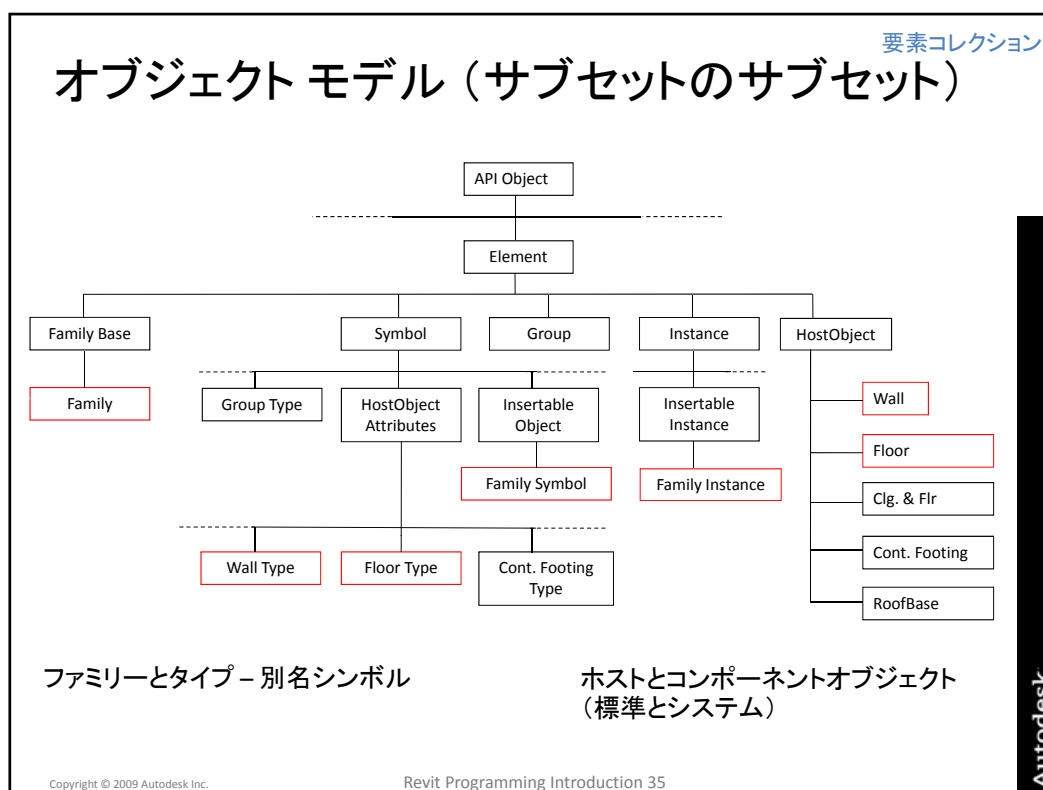
要素コレクション

重要な親クラス

- APIObject – オブジェクトのルーツ
- Element – BIM 要素
- Symbol – タイプ定義
- GeometryObject – ジオメトリ オブジェクト

Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 34



要素コレクション

現在の選択要素にアクセス

外部コマンドから Revit データーにアクセスする方法1
現在選択されている要素へはドキュメントクラスのメンバーからアクセス

- ExternalCommandData.Application.ActiveDocument.Selection.Elements

入力引数が ExternalCommandData を提供

```
Selection sel = doc.Selection;
sMsg = "There are " + sel.Elements.Size + " elements in the selection set:";
foreach( Element elem in sel.Elements )
{
    string s = (null == elem.Category) ? elem.GetType().Name : elem.Category.Name;
    sMsg += "YrYn " + s + " Id=" + elem.Id.Value.ToString();
}
// from Lab1_2_CommandArguments
```

Application

ActiveDocument

Elements

Selection

Elements

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 36

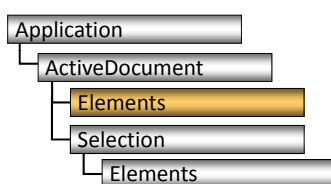
Autodesk

Revit データベース要素にアクセス

外部コマンドから Revit データーにアクセスする方法2

Document クラスは要素コレクションにアクセスを提供

- `Elements` – 全ての要素にアクセス
- `Elements(Type)` – 明示したタイプの要素にのみアクセス
- `Elements(Filter)` – フィルタ条件を満たした要素にのみアクセス
- `Elements(Type, ICollection<Element>)` – 明示タイプの要素を収集
- `Elements(Filter, ICollection<Element>)` – フィルタ条件を満たした要素を収集



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 37

Autodesk

Revit 要素のアクセス

- 引数が1つの `Document.Elements`
`ElementIterator` を返す
 - `iterator.MoveNext()`, `iterator.Current()` を介する
- 引数が2つの `Document.Elements`
 第2引数は包括的なコレクションの戻り値
 - `List<Element>` (C#)
 - `List(Of Element)` (VB.NET)
 - 標準的なコンテナアクセス 一例: `foreach`, `Count`, `Contains`
- `Document.Selection.Elements`
 - `ElementSet` を返す
 - `foreach`, `Size`, `Contains` を使用

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 38

Autodesk

要素を追加

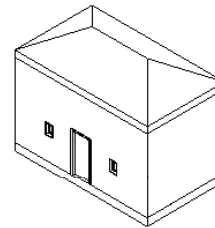
要素コレクション

- 追加

- Autodesk.Revit.Creation.Document で定義
- 120以上のメソッド
- 40以上のオブジェクトタイプをサポート
- 壁と床の例:

```
NewWall( CurveArray profile, bool structural ); // + 4 overloads  
NewSlab( CurveArray profile, Level, Line slopedArrow,  
         double angle, bool isImperial, bool isStructural );
```

- 未サポートのオブジェクト



Lab 2-0

- 削除

- Revit SDK DeleteDimensions と DeleteObject サンプルを参考

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 39

Autodesk

要素コレクション

要素コレクション

- Revit API では全て **APIObject** オブジェクト
- Revit BIM では全て **Element** オブジェクト
- すべて要素コレクションとして束ねられている
- SDKサンプルで ElementIterator と ElementFilterIterator を、
get_Elements()をLabで検索

Lab 2-1

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 40

Autodesk

全ての要素をリスト

要素コレクション

```
// *ALL* elements are bundled together and accessible via Document's ElementIterator
string line;
Element elem;
ElementIterator iter = commandData.Application.ActiveDocument.Elements;
while( iter.MoveNext() )
{
    elem = iter.Current as Element;
    line = "Id=" + elem.Id.Value.ToString(); // Element Id
    line += "; Class=" + elem.GetType().Name; // Element class (System.Type)
    // Element Category, not implemented for all classes, may return null:
    line += "; Category=" + ( null == elem.Category ? string.Empty : elem.Category.Name );
    // Element Name (different meaning for different classes, but mostly implemented logically)
    // Later, we'll see that more precise info on elements can be obtained in class-specific ways...
    line += "; Name=" + elem.Name;
    sw.WriteLine( line );
}
```

```
Id=16; Class=DimensionType; Category=; Name=Linear - 3mm Arial
Id=17; Class=Element; Category=; Name=
Id=19; Class=Element; Category=; Name=
Id=20; Class=FillPattern; Category=; Name=Solid fill
...
Id=126907; Class=Element; Category=; Name=M_W-Wide Flange
Id=126908; Class=Element; Category=; Name=W310X38.7
Id=126933; Class=Element; Category=; Name=Section Boxes
Id=126939; Class=Wall; Category=Walls; Name=Generic - 200mm
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 41

Autodesk

Revit 要素の特定

要素コレクション

- 通常のオブジェクト指向のアプローチはオブジェクトタイプ
あるいはクラスを使用
- Revit 要素の特定は、一元的には特定できない
- オブジェクトの特定は
 - カテゴリ
 - オブジェクトタイプまたはクラス (例: .NET System.Type)
 - 特定のパラメータ値
- 異なった判断基準の組み合わせを使用
- 場合により追加の判定が必要
- フィルター機能

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 42

Autodesk

要素コレクション

Revit 要素の特定

UI の要素名	Element/クラス から派生	カテゴリー
壁	HostObject/Wall	壁
ドア	Instance/InsertableInstance/FamilyInstance	ドア
ドアタグ	IndependentTag	ドアタグ
窓	Instance/InsertableInstance/FamilyInstance	窓
窓タグ	IndependentTag	窓タグ
開口部	Opening	長方形直線の壁開口部
床	HostObject/Floor	床
天井	Element	天井
屋根	HostObject/RoofBase/FootPrintRoof, ExtrusionRoof	屋根
柱	Instance/InsertableInstance/FamilyInstance	柱
コンポーネント(机)	Instance/InsertableInstance/FamilyInstance	家具
階段	Element	階段
手すり	Element	手すり
部屋	Room	部屋
部屋タグ	RoomTag	部屋タグ
グリッド	Grid	グリッド
モデル線分	ModelCurve/ModelLine	線分
参照面	ReferencePlane	参照面
寸法	Dimension	寸法
断面	Element	ビュー
文字	TextElement/TextNote	なし
レベル	Level	レベル
モデル グループ	Group	モデル グループ
インプレースを作成/壁	Instance/InsertableInstance/FamilyInstance	壁

Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 43

要素コレクション

要素 vs シンボル

UI の要素名	要素 Element/クラス から派生	カテゴリー	シンボル Symbol/クラスから派生	カテゴリー
壁	HostObject/Wall	壁	HostObjAttributes/WallType	壁
ドア	Instance/InsertableInstance/FamilyInstance	ドア	InsertableObject /FamilySymbol	ドア
ドアタグ	IndependentTag	ドアタグ	InsertableObject /FamilySymbol	ドアタグ
窓	Instance/InsertableInstance/FamilyInstance	窓	InsertableObject /FamilySymbol	窓
窓タグ	IndependentTag	窓タグ	InsertableObject /FamilySymbol	窓タグ
開口部	Opening	長方形直線の壁 開口部	< null >	---
床	HostObject/Floor	床	HostObjAttributes/FloorType	床
天井	< Element >	天井	HostObjAttributes	天井
屋根	HostObject/RoofBase/FootPrintRoof,ExtrusionRoof	屋根	HostObjAttributes/RoofType	屋根
柱	Instance/InsertableInstance/FamilyInstance	柱	InsertableObject /FamilySymbol	柱
コンポーネント(机)	Instance/InsertableInstance/FamilyInstance	家具	InsertableObject /FamilySymbol	家具
コンポーネント(本)	Instance/InsertableInstance/FamilyInstance	植物	InsertableObject /FamilySymbol	植物
階段	< Element >	階段	< Symbol >	階段
手すり	< Element >	手すり	< Symbol >	手すり
部屋	Room	部屋	< null >	---
部屋タグ	RoomTag	部屋タグ	< Symbol >	部屋タグ
グリッド	Grid	グリッド	LineAndTextAttrSymbol/GridType	< null >
モデル 線分	ModelCurve/ModelLine	線分	< null >	---
参照面	ReferencePlane	参照面	< null >	---
寸法	Dimension	寸法	DimensionType	< null >
断面	< Element >	ビュー	< Symbol >	< null >
文字	TextElement/TextNote	なし	LineAndTextAttrSymbol/TextElementType/TextNoteType	< null >
レベル	Level	レベル	LevelType	レベル
モデル グループ	Group	モデルグループ	GroupType	モデル グループ
インプレースを作成/壁	Instance/InsertableInstance/FamilyInstance	壁	InsertableObject /FamilySymbol	壁

Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 44

3D / モデル要素

要素コレクション

■ 3D 要素のみを取得したい場合

LabUtils.GetAllModelElements()

- 入力: Revit.Application
- 出力: ElementSet
- 選択条件
 - タイプは Symbol / FamilyBase でない
 - カテゴリーは null でない
 - ジオメトリは null でない

Lab 2-2

Copyright © 2009 Autodesk Inc.

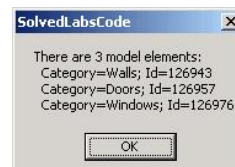
Revit Programming Introduction 45

Autodesk

3D / モデル要素

要素コレクション

```
public static ElementSet GetAllModelElements( Application app )
{
    ElementSet elems = app.Create.NewElementSet();
    Geo.Options opt = app.Create.NewGeometryOptions(); // this is ok
    ElementIterator iter = app.ActiveDocument.Elements;
    while( iter.MoveNext() )
    {
        Autodesk.Revit.Element elem = iter.Current as Autodesk.Revit.Element;
        if( !( elem is Symbol || elem is FamilyBase ) )
        {
            if( null != elem.Category )
            {
                if( null != elem.get_Geometry( opt ) )
                {
                    elems.Insert( elem );
                }
            }
        }
    }
    return elems;
}
```



Copyright © 2009 Autodesk Inc.

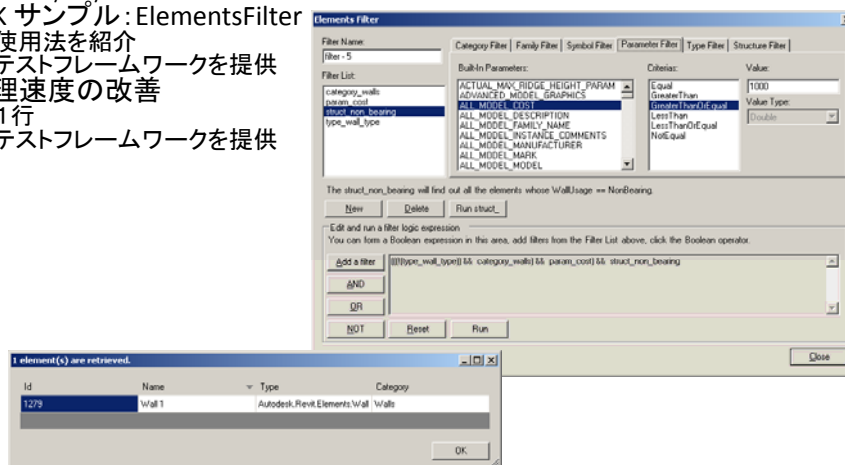
Revit Programming Introduction 46

Autodesk

要素フィルター

要素コレクション

- 個々のフィルターを使用して要素をデータベースから抽出
 - カテゴリー名 / 組み込みカテゴリー / ファミリー名 / シンボル名 / タイプ / 構造的分類 / パラメータ値 (例: eq, gt, ge, lt, le, ne)
- 個々のフィルターをブール演算で組み合わせる
 - and / or / not
- SDK サンプル: ElementsFilter
 - 使用法を紹介
 - テストフレームワークを提供
- 処理速度の改善
 - 1行
 - テストフレームワークを提供



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 47

Autodesk

要素のフィルターリングの比較(C#)

要素コレクション

フィルターリング無し

```
public static ElementSet GetAllWalls_noFilter( Application app )
{
    ElementSet elems = app.Create.NewElementSet();
    ElementIterator iter = app.ActiveDocument.Elements;
    while( iter.MoveNext() )
    {
        Element elem = iter.Current as Element;
        if( elem is Wall )
        {
            elems.Insert( elem );
        }
    }
    return elems;
}
```

フィルターリング有り

```
public static List<Element> GetAllWalls( Application app )
{
    List<Element> elements = new List<Element>();
    Filter filterType = app.Create.Filter.NewTypeFilter( typeof( Wall ) );
    app.ActiveDocument.get_Elements( filterType, elements );
    return elements;
}
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 48

Autodesk

要素コレクション

要素のフィルターリングの比較 (VB)

フィルターリング無し

```
Shared Function GetAllWalls_noFilter(ByVal app As Revit.Application) As ElementSet
    Dim elems As ElementSet = app.Create.NewElementSet
    Dim iter As IEnumerator = app.ActiveDocument.Elements
    Do While (iter.MoveNext())
        Dim elem As Revit.Element = iter.Current
        If TypeOf elem Is Wall Then
            elems.Insert(elem)
        End If
    Loop
    Return elems
End Function
```

フィルターリング有り

```
Shared Function GetAllWalls(ByVal app As Revit.Application) As List(Of Element)
    Dim elements As New List(Of Element)
    Dim filterType As Filter = app.Create.Filter.NewTypeFilter(GetType(Wall))
    Dim n As Integer = app.ActiveDocument.Elements(filterType, elements)
    Return elements
End Function
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 49

Autodesk

要素コレクション

ブール演算フィルターの使用例

C#

```
public static List<Element> GetAllModelGroupTypes( Application app )
{
    List<Element> elements = new List<Element>();
    Filter filterType = app.Create.Filter.NewTypeFilter( typeof( GroupType ) );
    Filter filterParam = app.Create.Filter.NewParameterFilter(
        BuiltInParameter.SYMBOL_FAMILY_NAME_PARAM, CriteriaFilterType.Equal,
        "Model Group" );
    Filter filterAnd = app.Create.Filter.NewLogicAndFilter( filterType, filterParam );
    app.ActiveDocument.get_Elements( filterAnd, elements );
    return elements;
}
```

VB

```
Shared Function GetAllModelGroupTypes(ByVal app As Revit.Application) As List(Of Element)
    Dim elements As New List(Of Element)
    Dim filterType As Filter = app.Create.Filter.NewTypeFilter(GetType(GroupType))
    Dim filterParam = app.Create.Filter.NewParameterFilter( _
        BuiltInParameter.SYMBOL_FAMILY_NAME_PARAM, _
        CriteriaFilterType.Equal, "Model Group" )
    Dim filter As Filter = app.Create.Filter.NewLogicAndFilter(filterType, filterParam)
    Dim n As Integer = app.ActiveDocument.Elements(filter, elements)
    Return elements
End Function
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 50

Autodesk

特定要素のクラス

要素コレクション

- 例えば、どうやって全てのドアや壁を取得できるか?
- 壁は個別のクラス
 - APIObject ← Element ← HostObject ← Wall
 - (in-place ファミリー壁は FamilyInstance)
 - Wall System.Type でフィルタ
- ドアはカテゴリーで識別可能
 - System.Type は包括的な FamilyInstance
 - 言語依存を避けるため BuiltInCategory 列挙タイプを使用
 - FamilyInstance System.Type とドア カテゴリーでフィルター

Lab 2-3

Copyright © 2009 Autodesk Inc.

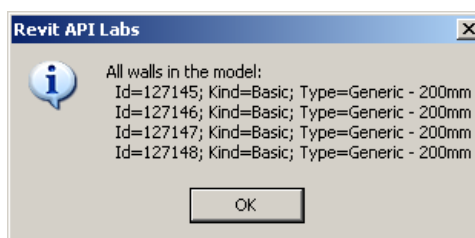
Revit Programming Introduction 51

Autodesk

壁を取得

要素コレクション

```
public static List<Element> GetAllWalls( Application app )
{
    List<Element> elements = new List<Element>();
    Filter filterType = app.Create.Filter.NewTypeFilter( typeof( Wall ) );
    app.ActiveDocument.get_Elements( filterType, elements );
    return elements;
}
```



Copyright © 2009 Autodesk Inc.

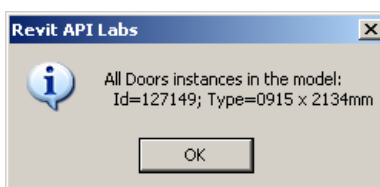
Revit Programming Introduction 52

Autodesk

全てのドアを取得

要素コレクション

```
public static List<Element> GetAllStandardFamilyInstancesForACategory(
    Application app,
    BuiltInCategory bic )
{
    List<Element> elements = new List<Element>();
    Filter filterType = app.Create.Filter.NewTypeFilter( typeof( FamilyInstance ) );
    Filter filterCategory = app.Create.Filter.NewCategoryFilter( bic );
    Filter filterAnd = app.Create.Filter.NewLogicAndFilter( filterType, filterCategory );
    app.ActiveDocument.get_Elements( filterAnd, elements );
    return elements;
}
```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 53

Autodesk

要素を編集

要素コレクション

- 壁を選択
- 現在の選択から壁を抽出
- 壁の上部と下部のレベル拘束を取得
- ジオメトリをレベルと壁の曲線から計算
- 柱タイプのファミリーシンボルを取得
- 柱を始点、中点、終点に挿入
- 壁を柱より移動して離す

■ Lab 2-4

- 必要条件
 - 壁の上部が拘束されている
 - “M_Wood Timber Column” タイプがロード済み

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 54

Autodesk

壁を選択する

要素コレクション

```

Dim doc As Revit.Document = commandData.Application.ActiveDocument
Dim ss As ElementSet = doc.Selection.Elements

' must have single element only
If Not ss.Size = 1 Then
    MsgBox( "You must pre-select a single element!" )
    Return IExternalCommand.Result.Cancelled
End If

' must be a wall
Dim iter As ElementSetIterator = ss.ForwardIterator
iter.MoveNext()
Dim elem As Revit.Element = iter.Current
If Not TypeOf elem Is Wall Then
    MsgBox( "Selected element is NOT a wall!" )
    Return IExternalCommand.Result.Cancelled
End If
Dim wall As Wall = elem

```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 55

Autodesk

壁の上部と下部

要素コレクション

```

' wall must be constrained to a Level at the top (more on Parameters
  later...)
Dim topLev As Level = Nothing
Try
    topLev =
        doc.Element(wall.Parameter(BuiltInParameter.WALL_HEIGHT_TYPE).AsElementId)
Catch
    topLev = Nothing
End Try
If topLev Is Nothing Then
    MsgBox( "Selected Wall is NOT constrained to a Level at the Top!" )
    Return IExternalCommand.Result.Cancelled
End If

' get the bottom Level as well (this should never fail)
Dim botLev As Level = Nothing
Try
    botLev =
        doc.Element(wall.Parameter(BuiltInParameter.WALL_BASE_CONSTRAINT).AsElementId)
Catch
    botLev = Nothing
End Try
If botLev Is Nothing Then
    MsgBox( "Selected Wall is NOT constrained to a Level at the Bottom?!" )
    Return IExternalCommand.Result.Cancelled
End If

```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 56

Autodesk

柱のジオメトリ

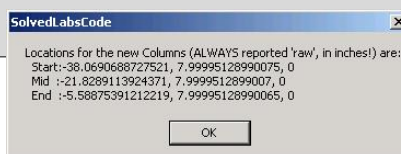
要素コレクション

```
' Calculate the location points for the 3 columns (assuming straight wall)
Dim locations As New XYZArray()

Dim locCurve As LocationCurve = wall.Location
Dim ptStart As XYZ = locCurve.Curve.EndPoint(0)
Dim ptEnd As XYZ = locCurve.Curve.EndPoint(1)
Dim ptMid As New XYZ((ptStart.X + ptEnd.X) / 2, (ptStart.Y + ptEnd.Y) / 2, (ptStart.Z +
ptEnd.Z) / 2)

locations.Append(ptStart)
locations.Append(ptMid)
locations.Append(ptEnd)

Dim sMsg As String = "Locations for the new Columns (ALWAYS reported 'raw', in inches!)
are:"
sMsg += vbCrLf & " Start:" & ptStart.X & ", " & ptStart.Y & ", " & ptStart.Z
sMsg += vbCrLf & " Mid : " & ptMid.X & ", " & ptMid.Y & ", " & ptMid.Z
sMsg += vbCrLf & " End : " & ptEnd.X & ", " & ptEnd.Y & ", " & ptEnd.Z
MsgBox(sMsg)
```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 57

Autodesk

柱タイプ

要素コレクション

```
/// Helper to get specified Type for specified Family as FamilySymbol object
public static FamilySymbol GetFamilySymbol(
    Application app,
    string familyName,
    string typeName )
{
    Filter filterType = app.Create.Filter.NewTypeFilter( typeof( FamilySymbol ) );
    Filter filterFamilyName = app.Create.Filter.NewFamilyFilter( familyName );
    Filter filter = app.Create.Filter.NewLogicAndFilter( filterType, filterFamilyName );
    List<Element> elementList = new List<Element>();
    int n = app.ActiveDocument.get_Elements( filter, elementList );
    // we have a list of symbols for a given family.
    // loop through the list and find a match
    foreach( Element e in elementList )
    {
        if( e.Name.Equals( typeName ) )
        {
            return e as FamilySymbol;
        }
    }
    return null;
}

' Get Family Type for the new instances.
' Names must match a column type available in the model
Dim sFamilyName As String = "M_Wood Timber Column"
Dim sTypeName As String = "191 x 292mm"
Dim symbol As FamilySymbol = LabUtils.GetFamilySymbol( _
doc, sFamilyName, sTypeName)
If symbol Is Nothing Then
    MsgBox("Cannot find Family=" & sFamilyName & " Type=" & sTypeName
-
& " in the model - load it first!")
    Return IExternalCommand.Result.Cancelled
End If
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 58

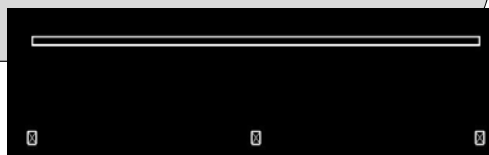
Autodesk

柱挿入と壁の移動

要素コレクション

```
' Insert Columns as FamilyInstances
Dim pt As XYZ
For Each pt In locations
    Try
        Dim column As FamilyInstance = doc.Create.NewFamilyInstance(pt, symbol, botLev,
            Structural.Enums.StructuralType.NonStructural)
        Dim paramTopLevel As Parameter =
            column.Parameter(BuiltInParameter.FAMILY_TOP_LEVEL_PARAM)
        paramTopLevel.Set(topLev.Id)
    Catch e As Exception
        MsgBox("Failed to create or adjust this column!?",)
    End Try
Next

' Finally, move the wall so the columns are surely visible. For simplicity,
' move the wall "perpendicularly" to the Location curve by its length
Dim wallPerpAxis As New XYZ(-(ptEnd.Y - ptStart.Y), ptEnd.X - ptStart.X, 0)
If Not CType(wall.Location.Move(wallPerpAxis), Boolean) Then
    MsgBox("Failed to Move the wall!?",)
End If
```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 59

Autodesk

ファミリーとタイプ

Autodesk

ファミリーとタイプ

ファミリーとタイプ

- シンボルとも呼ぶ
- 標準 vs システムファミリー
- ロード済みのファミリーとシンボルを使用
- 新規にファミリーとシンボルをロード
- 要素タイプを特定し、変更
 - 標準とシステムファミリー

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 61

Autodesk

ファミリーとタイプをリスト

ファミリーとタイプ

- 頻繁に判定する必要があること
 - モデルにどの標準 ファミリーとシンボルがロード済みか
 - それらはどのカテゴリに属するか

Lab 3-1

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 62

Autodesk

ファミリーイテレーション

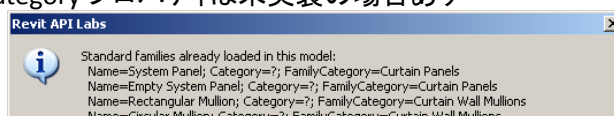
ファミリーとタイプ

```

List<Element> families = new List<Element>();
Filter filterFamily = app.Create.Filter.NewTypeFilter( typeof( Family ) );
doc.get_Elements( filterFamily, families );
string sMsg = "Standard families already loaded in this model:";
foreach( Family f in families )
{
    // Get its category name; notice that the Category property is not
    // implemented for the Family class; use FamilyCategory instead;
    // notice that that is also not always implemented; in that case,
    // use the workaround demonstrated below, looking at the contained
    // family symbols' category:
    sMsg += "YrYn Name=" + f.Name
        + "; Category=" + ( ( null == f.Category ) ? "?" : f.Category.Name )
        + "; FamilyCategory=" + ( ( null == f.FamilyCategory ) ? "?" :
        f.FamilyCategory.Name );
}
LabUtils.InfoMsg( sMsg );

```

- Categoryプロパティは常に未実装
- FamilyCategory プロパティは未実装の場合あり



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 63

Autodesk

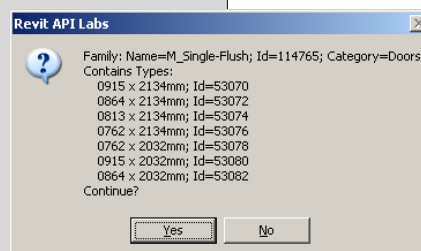
ファミリーシンボル イテレーション

ファミリーとタイプ

```

// Loop through the collection of families, and now look at
// the child symbols (types) as well. These symbols can be
// used to determine the family category.
foreach( Family f in families )
{
    string catName;
    bool first = true;
    // Loop all contained symbols (types)
    foreach( FamilySymbol symb in f.Symbols )
    {
        // you can determine the family category
        // from its first symbol.
        if( first )
        {
            first = false;
            catName = symb.Category.Name;
            sMsg = "Family: Name=" + f.Name
                + "; Id=" + f.Id.Value.ToString()
                + "; Category=" + catName
                + "YrYnContains Types:";
        }
        sMsg += "YrYn " + symb.Name + "; Id=" + symb.Id.Value.ToString();
    }
    // Show the symbols for this family and allow user to proceed
    // to the next family (OK) or cancel (Cancel)
    sMsg += "YrYnContinue?";
    if( !LabUtils.QuestionMsg( sMsg ) )
    {
        break;
    }
}

```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 64

Autodesk

ファミリーとタイプのロード

ファミリーとタイプ

- RFA ファイルから追加の標準ファミリーとシンボルをロード

```
doc.LoadFamily()  
doc.LoadFamilySymbol()
```

Lab 3-2

```
Dim doc As Revit.Document = commandData.Application.ActiveDocument  
  
'Load a whole Family  
If Not CType(doc.LoadFamily(familyFilename), Boolean) Then  
    MsgBox( "ERROR in loading Family " & familyFilename & "?" )  
Else  
    MsgBox( "Successfully loaded Family " & familyFilename & "!" )  
End If  
  
'Load only a specific Symbol (Type)  
' The symbol MUST exist in the corresponding catalog (TXT) file - same as in the UI  
If Not CType(doc.LoadFamilySymbol(familyFilename, symbolName), Boolean) Then  
    MsgBox( "ERROR in loading FamilySymbol " & familyFilename & " : " & symbolName & "?" )  
Else  
    MsgBox( "Successfully loaded FamilySymbol " & familyFilename & " : " & symbolName & "!" )  
End If
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 65

Autodesk

標準ファミリーとタイプを判定

ファミリーとタイプ

- 要素のファミリーとタイプを判定
- 窓を選択
- 全ての窓のシンボルをリスト
- 選択した窓のファミリーシンボルとファミリーをリスト

Lab 3-3

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 66

Autodesk

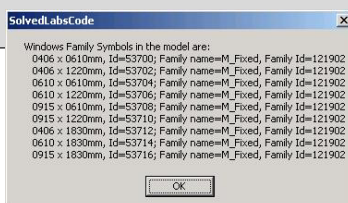
窓シンボルをリスト

ファミリーとタイプ

```

BuiltInCategory bic = BuiltInCategory.OST_Windows;
Filter filterCategory = app.Create.Filter.NewCategoryFilter( bic );
Filter filterType = app.Create.Filter.NewTypeFilter( typeof( FamilySymbol ) );
Filter filterAnd = app.Create.Filter.NewLogicAndFilter( filterCategory, filterType );
List<Element> familySymbols = new List<Element>();
app.ActiveDocument.get_Elements( filterAnd, familySymbols );
string sMsg = "The loaded windows family symbols in the model are:";
foreach( Element e in familySymbols )
{
    FamilySymbol symb = e as FamilySymbol;
    sMsg += "YrYn    " + symb.Name + ", Id=" + symb.Id.Value.ToString();
    Family fam = symb.Family;
    sMsg += "; Family name=" + fam.Name + ", Family Id=" + fam.Id.Value.ToString();
}

```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 67

Autodesk

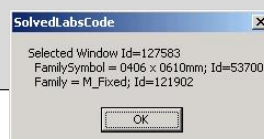
ファミリーとシンボルを取得

ファミリーとタイプ

```

' Now loop the selection set and check for standard
' Family Instances of "Windows" category
For Each elem In doc.Selection.Elements
    If TypeOf elem Is FamilyInstance Then
        Dim inst As FamilyInstance = elem
        Dim catInst As Category = Nothing
        Try
            ' just in case
            catInst = inst.Category
        Catch
            End Try
        If (Not catInst Is Nothing) AndAlso catInst.Equals(catWindows) Then
            sMsg = "Selected Window Id=" & elem.Id.Value.ToString & vbCrLf
            Dim fs1 As FamilySymbol = inst.Symbol
            sMsg += " FamilySymbol = " & fs1.Name & "; Id=" & _
                & fs1.Id.Value.ToString & vbCrLf
            Dim fl As Family = fs1.Family
            sMsg += " Family = " & fl.Name & "; Id=" & fl.Id.Value.ToString
        End If
    End If
    ' Report each Window data
    MsgBox(sMsg)
Next

```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 68

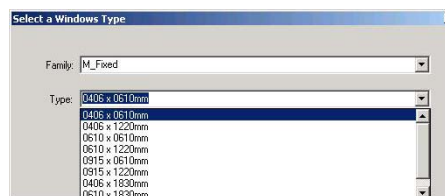
Autodesk

ファミリーインスタンスのタイプを変更

ファミリーとタイプ

- シンプルなダイアログでファミリーインスタンスのタイプを変更
- ファミリーインスタンスのカテゴリを判定 (例: 窓)
- 適用できる全てのシンボルのリストを構築
 - 全ての要素を検索し、個々のファミリーのカテゴリをチェックする
 - もし一致すれば、全てのシンボルを含める
 - ダイアログのリストボックスで表示する
- 選択されたシンボルを割り当てる

Lab 3-4



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 69

Autodesk

選択されたインスタンスのカテゴリを判定

ファミリーとタイプ

```

Dim inst As FamilyInstance
Dim instCat As Category
Dim ss As ElementSet = doc.Selection.Elements
' Make sure we have a single FamilyInstance selected
If Not ss.Size = 1 Then
    MsgBox("You must pre-select a single element!")
    Return IExternalCommand.Result.Cancelled
Else
    Dim itTmp As ElementSetIterator = ss.ForwardIterator
    itTmp.MoveNext()
    Dim elTmp As Revit.Element = itTmp.Current
    If Not TypeOf elTmp Is FamilyInstance Then
        MsgBox("Selected element is NOT a standard family instance!")
        Return IExternalCommand.Result.Cancelled
    Else
        inst = elTmp
        instCat = inst.Category
    End If
End If

```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 70

Autodesk

カテゴリーの全てのシンボルをリスト

ファミリーとタイプ

```
Dim dictFamilyToSymbols As New Dictionary(Of String, ArrayList)

' using Revit 2009 element filtering
Dim families As List(Of Revit.Element) = New List(Of Revit.Element)
Dim filter As Revit.Filter = app.Create.Filter.NewTypeFilter(GetType(Family))

Dim nRetVal = doc.Elements(filter, families)
Dim f As Family
Dim categoryMatches = False
For Each f In families
    categoryMatches = False
    If (f.FamilyCategory Is Nothing) Then ' not always implemented
        For Each sym As FamilySymbol In f.Symbols
            categoryMatches = sym.Category.Id.Equals(instCat.Id)
        Exit For
    Next
Else
    categoryMatches = f.FamilyCategory.Id.Equals(instCat.Id)
End If

If (categoryMatches) Then
    Dim familySymbols As New ArrayList
    For Each sym As FamilySymbol In f.Symbols
        familySymbols.Add(sym)
    Next
    dictFamilyToSymbols.Add(f.Name, familySymbols)
End If
Next
```

Copyright © 2009 Autodesk Inc.

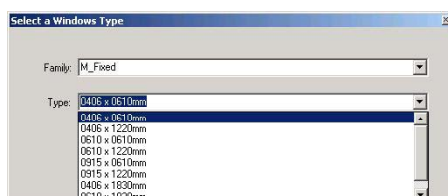
Revit Programming Introduction 71

Autodesk

新たにタイプを割り当てる

ファミリーとタイプ

```
' Display the form and change the type
Dim frm As New Lab3_4_Form(dictFamilyToSymbols)
If frm.ShowDialog = Windows.Forms.DialogResult.OK Then
    Try
        inst.Symbol = frm.cmbType.SelectedItem
        MsgBox("Successfully changed Family:Type to " & _
            & frm.cmbFamily.Text & " : " & frm.cmbType.Text)
    Catch
    End Try
End If
```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 72

Autodesk

システムファミリーとタイプ

ファミリーとタイプ

- 一般的な標準ファミリーより編集は容易
- HostObjectから派生した専用のクラス
- 壁と床クラスは十分に公開されている
- RoofBase / FootPrintRoof / ExtrusionRoof

Lab 3-5

- 全ての壁タイプをリストして最後のものを保存: Document プロパティーのWallTypesを使用
- 全ての床タイプをリスト、最後のものを保存
- 全ての選択した壁と床のタイプを変更

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 73

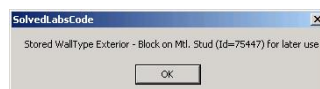
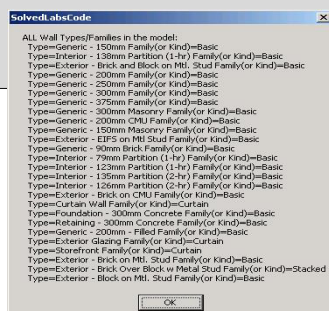
Autodesk

全ての壁タイプをリスト

ファミリーとタイプ

```
' Find ALL Wall Types and their System Families (or Kinds)
Dim newWallType As WallType = Nothing ' store the last one
Dim sMsg As String = "ALL Wall Types/Families in the model:"

' We could again iterate all elements and check for WallType class,
' but it's simpler directly from the doc:
Dim doc As Revit.Document = commandData.Application.ActiveDocument
For Each wt As WallType In doc.WallTypes
    sMsg += vbCrLf & " Type=" & wt.Name & " Family(or Kind)=" & wt.Kind.ToString
    newWallType = wt
Next
MsgBox(sMsg)
MsgBox( "Stored WallType " & newWallType.Name _
    & " (Id=" & newWallType.Id.Value.ToString & ") for later use" )
```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 74

Autodesk

全ての床タイプをリスト

ファミリーとタイプ

```
' Find ALL Floor Types ... see updated code and more comments for Revit 2009 API in Labs code
Dim newFloorType As FloorType = Nothing ' at the same time store the last one to use to change the floor type later
sMsg = "ALL FLOOR Types in the model:"
' There is no dedicated property to get floor types, so need to iterate all elements again
Dim iter As ElementIterator = doc.Elements
Do While (iter.MoveNext())
  Dim elem As Revit.Element = iter.Current
  If TypeOf elem Is Symbols.FloorType Then
    Dim ft As Symbols.FloorType = elem
    sMsg += vbCrLf & "  Type=" & ft.Name & ", Id=" & ft.Id.Value.ToString
    ' In 9.0, it looks like "Foundation Slab" system family from "Structural Foundations" category
    ' ALSO contains FloorType class instances. Exclude those as choices for standard floor types
    Dim famName As String
    Try
      famName = ft.Parameter(Parameters.BuiltInParameter.SYMBOL_FAMILY_NAME_PARAM).AsString
    Catch
      famName = "?"
    End Try
    Dim cat As Category = ft.Category
    sMsg += ", Family=" & famName & ", Category=" & cat.Name
    ' store only if proper Floors category
    If doc.Settings.Categories.Item(BuiltInCategory.OST_Floors).Equals(cat) Then
      newFloorType = ft
    End If
  End If
Loop
MsgBox(sMsg)
MsgBox( "Stored FloorType " & newFloorType.Name & " (Id=" & newFloorType.Id.Value.ToString & ") for later use")
```

Solved! absCode

Stored FloorType Generic Floor - 400mm - Filled (Id=40350) for later use

OK

Solved! absCode

ALL FLOOR Types in the model:
Type=Generic Floor - 400mm, Id=226, Family=Floor, Category=Floors
Type=Wood Joist 220mm - Wood Finish, Id=25390, Family=Floor, Category=Floors
Type=Steel Bar Joist - VCT on LW Concrete, Id=25398, Family=Floor, Category=Floors
Type=Generic Floor - 400mm - Filled, Id=40350, Family=Floor, Category=Floors
Type=150mm Foundation Slab, Id=109351, Family=Foundation Slab, Category=Structural Foundations

OK

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 75

Autodesk

壁と床タイプを変更

ファミリーとタイプ

```
' Change the Type for selected Walls and Floors
Dim sel As ElementSet = doc.Selection.Elements
Dim iWall As Integer
Dim el As Revit.Element
For Each el In sel
  If TypeOf el Is Wall Then ' Check for walls
    Dim wall As Wall = el
    iWall += 1
    Dim oldWallType As WallType = wall.WallType
    wall.WallType = newWallType
    MsgBox("Wall " & iWall.ToString & ": Id=" & wall.Id.Value.ToString & vbCrLf & _
      " changed from OldType=" & oldWallType.Name & "; Id=" & oldWallType.Id.Value.ToString & _
      " to NewType=" & wall.WallType.Name & "; Id=" & wall.WallType.Id.Value.ToString)
  ElseIf TypeOf el Is Floor Then
    Dim f As Floor = el
    iFloor += 1
    Dim oldFloorType As FloorType = f.FloorType
    f.FloorType = newFloorType
    MsgBox("Floor " & iFloor.ToString & ": Id=" & f.Id.Value.ToString & vbCrLf & _
      " changed from OldType=" & oldFloorType.Name & "; Id=" & oldFloorType.Id.Value.ToString & _
      " to NewType=" & f.FloorType.Name & "; Id=" & f.FloorType.Id.Value.ToString)
  End If
Next
```

Solved! absCode

Wall 1: Id=127653
changed from OldType=Generic - 200mm; Id=249 to NewType=Exterior - Block on Mt. Stud; Id=75447

OK

Solved! absCode

Floor 1: Id=127821
changed from OldType=Generic Floor - 400mm; Id=226 to NewType=Generic Floor - 400mm - Filled; Id=40350

OK

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 76

Autodesk

パラメーター

Autodesk

パラメーター

パラメーター

- 要素パラメーターへのアクセス
- 組み込み VS 共有パラメーター
- 外部アプリケーションとのデータのやり取り
- 要素ごとのカスタムデータを保存するための方法
- 共有パラメーターファイルとグループの編集
- パラメータを非表示
- 隠されたパラメーター
- ドキュメントごとのデータ(モデルごと)を保存する方法
- SDK サンプル : FireRating

Autodesk

パラメーターにアクセス

パラメーター

■ 要素パラメーターにアクセスする方法？

- Element.Parameters をループする
- 名前、組み込みパラメーター列挙タイプ、定義、またはGUID が分っていれば、それを直接取得

Parameter(Guid) - 共有パラメーターのGUIDよりパラメーターを取得
 Parameter(String) - パラメーター名よりパラメーターを取得
 Parameter(Definition) - パラメーターの定義オブジェクトより取得
 Parameter(BuiltInParameter) - 組み込みパラメーターのIDより取得

Lab 4-1

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 79

Autodesk

全ての要素パラメーターをループ

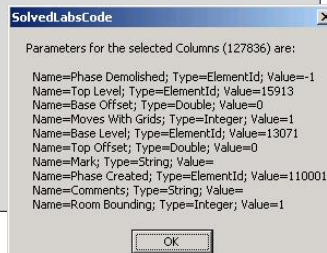
パラメーター

```

' List all UI-visible Params
Dim sMsg As String = "Parameters for the selected " & elem.Category.Name & " (" &
  elem.Id.Value.ToString & ") are:" & vbCrLf
Dim param As Parameter
For Each param In elem.Parameters
  Dim paramName As String = param.Definition.Name
  Dim paramType As String = param.StorageType.ToString
  Dim paramValue As String = LabUtils.GetParamAsString(param)
  sMsg += "  Name=" & paramName & "; Type=" & paramType & "; Value=" & paramValue
Next
MsgBox(sMsg)

Public Shared Function GetParamAsString(ByVal param As Parameter) As String
  Dim str As String
  Select Case param.StorageType
    Case StorageType.Double
      str = param.AsDouble.ToString
    Case StorageType.Integer
      str = param.AsInteger.ToString
    Case StorageType.String
      str = param.AsString
    Case StorageType.ElementId
      str = param.AsElementId.Value.ToString
  End Select
  Return str
End Function

```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 80

Autodesk

組み込みパラメーターへのアクセス

パラメーター

```
' If we know WHICH param we are looking for, then:
' A) a standard parameter, can be got via BuiltInParameter signature of Parameter method:
Dim parInBuilt As Parameter
Try
    parInBuilt = elem.Parameter(BuiltInParameter.FAMILY_BASE_LEVEL_OFFSET_PARAM)
    If Not parInBuilt Is Nothing Then
        Dim parInBuiltName As String = parInBuilt.Definition.Name
        Dim parInBuiltType As String = LabUtils.GetParamStorageType(parInBuilt).ToString
        Dim parInBuiltValue As String = LabUtils.GetParamAsString(parInBuilt)
        MsgBox("FAMILY_BASE_LEVEL_OFFSET_PARAM: Name=" & parInBuiltName & "
            & "; Type=" & parInBuiltType & "; Value=" & parInBuiltValue) =
    Else
        MsgBox("FAMILY_BASE_LEVEL_OFFSET_PARAM is NOT available for this element")
    End If
Catch
    MsgBox("FAMILY_BASE_LEVEL_OFFSET_PARAM is NOT available for this element")
End Try
```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 81

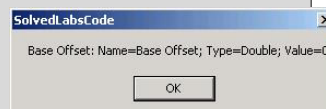
Autodesk

名前の付いたパラメーターにアクセス

パラメーター

```
'Helper to get *specific* parameter by name
Shared Function GetElemParam(ByVal elem As Revit.Element, ByVal name As String) As
    Parameter
    Dim parameters As Autodesk.Revit.ParameterSet = elem.Parameters
    Dim parameter As Autodesk.Revit.Parameter
    For Each parameter In parameters
        If (parameter.Definition.Name = name) Then
            Return parameter
        End If
    Next
    Return Nothing
End Function
```

```
' C) use GetElemParam utility to get it by hard coded-name
' (this works for either standard or shared!):
Const csParamToFind As String = "Base Offset"
Dim parByName As Parameter = LabUtils.GetElemParam(elem, csParamToFind)
If parByName Is Nothing Then
    MsgBox(csParamToFind & " is NOT available for this element")
Else
    Dim parByNameName As String = parByName.Definition.Name
    Dim parByNameType As String = LabUtils.GetParamStorageType(parByName).ToString
    Dim parByNameValue As String = LabUtils.GetParamAsString(parByName)
    MsgBox(csParamToFind & ": Name=" & parByNameName & "; Type=" & parByNameType & ";
        Value=" & parByNameValue)
End If
```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 82

Autodesk

パラメーター

パラメーター コレクションは不完全

- ほとんどの要素はパラメーターコレクションにリストされているより多くのパラメーターを持つ
- 目標としているパラメーターを検索する方法:
パラメーターを個々の組み込みパラメータ列挙タイプの値で取得
- BuiltInParamsChecker
- RvtMgdDbg の 'Built-in Enums Snoop...'
- 例) ドア要素がどの部屋に属するか:
ドア要素のパラメーターコレクションでリストされない
ELEM_ROOM_ID 組み込みパラメーターより、Room プロパティーを取得可能。(全てのファミリーインスタンスにあるパラメーター)

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 83

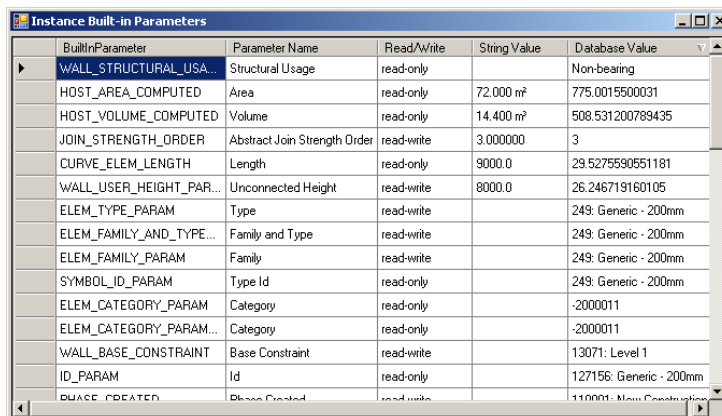
Autodesk

パラメーター

BuiltInParamsChecker

選択した要素の全ての有効な組み込みパラメーターを判定する

- 全ての組み込みパラメーターの列挙タイプをループ
- それぞれに有効なパラメーターを取得
- DataGridView を使用してテーブルでソートした結果



BuiltInParameter	Parameter Name	Read/Write	String Value	Database Value
WALL_STRUCTURAL_USA...	Structural Usage	read-only		Non-bearing
HOST_AREA_COMPUTED	Area	read-only	72.000 m²	775.0015500031
HOST_VOLUME_COMPUTED	Volume	read-only	14.400 m³	508.531200789435
JOIN_STRENGTH_ORDER	Abstract Join Strength Order	read-write	3.000000	3
CURVE_ELEM_LENGTH	Length	read-only	9000.0	29.5275590551181
WALL_USER_HEIGHT_PAR...	Unconnected Height	read-write	8000.0	26.246719160105
ELEM_TYPE_PARAM	Type	read-write		249: Generic - 200mm
ELEM_FAMILY_AND_TYPE...	Family and Type	read-write		249: Generic - 200mm
ELEM_FAMILY_PARAM	Family	read-write		249: Generic - 200mm
SYMBOL_ID_PARAM	Type Id	read-only		249: Generic - 200mm
ELEM_CATEGORY_PARAM	Category	read-only		-2000011
ELEM_CATEGORY_PARAM...	Category	read-only		-2000011
WALL_BASE_CONSTRAINT	Base Constraint	read-write		13071: Level 1
ID_PARAM	Id	read-only		127156: Generic - 200mm
PHASE_CREATED	Phase Created	read-write		110001: New Construction

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 84

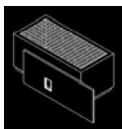
Autodesk

パラメーターの書き出し

パラメーター

- Revit BIM データーの外部のアプリケーションへ書き出し
- 例) COM Interop を使用して Revit から MS Excel ファイルへの書き出し

Lab 4-2



ID	Unconnected Height	Comments	Base Constraint	Phase Description	Top Is Attached	Structural Usage	Base Extension Distance	Volume	Area
1	127025	13 1230968	13071	-1	0	0	0	271.857829	0
2	127063	13 1230968	13071	-1	0	0	0	629.689322	0
3	127076	13 1230968	13071	-1	0	0	0	272.911744	0
4	127127	13 1230968	13071	-1	0	0	0	189.8086763	0
5	127171	13 1230968	13071	-1	0	0	0	243.0214105	0

- 外部プロセスクライアント
- 有効なカテゴリを持つシンボルではない全ての要素を取得しマップを作成
- Excel を起動して新規でワークブックを作成
- カテゴリごとにシートを追加
- 各カテゴリのセットの全ての要素を検索
- 使用している全てのパラメータ名を決定
- パラメータ名をリストするタイトル行を追加
- 各要素の全てのパラメータを行ごとに追加

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 85

Autodesk

全要素のマップを作成

パラメーター

```

' First extract and group the data from Revit in a convenient Map class:
' (Key=category name, Val=Set of Elements)
' In Revit 2009, we use an element filter instead ... see updated Labs code
Dim sortedElements As Autodesk.Revit.Collections.Map = revitApp.Create.NewMap()
Dim iter As IEnumerator = doc.Elements
Do While (iter.MoveNext())
    ' We look for all non-Symbol Elements which have a Category
    Dim element As Revit.Element = iter.Current
    If Not (TypeOf element Is Symbol) Then
        Dim category As Category = element.Category
        If Not (category Is Nothing) Then
            Dim elementSet As ElementSet
            ' If we already have this Key, get its Value (Set)
            ' Otherwise, create the new Key and Value
            If sortedElements.Contains(category.Name) Then
                elementSet = sortedElements.Item(category.Name)
            Else
                elementSet = revitApp.Create.NewElementSet()
                sortedElements.Insert(category.Name, elementSet)
            End If
            ' Add the element to the Set
            elementSet.Insert(element)
        End If
    End If
Loop

```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 86

Autodesk

Excel を起動してワークブックを追加

パラメーター

```
Imports MsExcel = Microsoft.Office.Interop.Excel

' Launch/Get Excel (via COM Interop)
Dim excel As MsExcel.Application = New MsExcel.ApplicationClass()
If (excel Is Nothing) Then
    MsgBox("Failed to get or start Excel!?",)
    Return IExternalCommand.Result.Failed
End If
excel.Visible = True ' Make it visible "live" to the user

' Add a new work-book and delete the default work-sheets
Dim workbook As MsExcel.Workbook = excel.Workbooks.Add()
Dim worksheet As MsExcel.Worksheet
Do While workbook.Sheets.Count > 1
    worksheet = workbook.Sheets.Item(1)
    worksheet.Delete()
Loop

' Loop all collected Categories and create one worksheet for each
Dim mapIter As Autodesk.Revit.Collections.MapIterator = sortedElements.ForwardIterator
Do While (mapIter.MoveNext())
    ' retrieve stored category and ElementSet
    Dim categoryName As String = mapIter.Key
    Dim elementSet As Autodesk.Revit.ElementSet = mapIter.Current
    ' create and name the worksheet
    worksheet = excel.Worksheets.Add()
    worksheet.Name = categoryName
End While
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 87

Autodesk

全てのテーブルヘッダーを決定

パラメーター

```
Dim allParamNamesEncountered As Autodesk.Revit.Collections.Set _
    = revitApp.Create.NewSet()

' loop through all the elements passed to the method
Dim setIter As IEnumerator = elementSet.ForwardIterator
Do While (setIter.MoveNext())
    Dim el As Autodesk.Revit.Element = setIter.Current
    Dim parameters As Autodesk.Revit.ParameterSet = el.Parameters
    If Not (parameters.IsEmpty) Then
        ' Another way to loop the parameters is via ParameterSetIterator:
        Dim definitionNames As Autodesk.Revit.Collections.Set _
            = revitApp.Create.NewSet()
        Dim paramIter As Autodesk.Revit.ParameterSetIterator _
            = parameters.ForwardIterator
        Do While paramIter.MoveNext()
            Dim parameter As Autodesk.Revit.Parameter = paramIter.Current
            Dim name As String = parameter.Definition.Name
            If Not allParamNamesEncountered.Contains(name) Then
                allParamNamesEncountered.Insert(name)
            End If
        Loop
    End If
End While
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 88

Autodesk

テーブルヘッダーを作成

パラメーター

```
' add the HEADER row in Bold
worksheet.Cells(1, 1).Value = "ID"
Dim paramName As String
Dim column As Integer = 2
For Each paramName In allParamNamesEncountered
    worksheet.Cells(1, column).Value = paramName
    excel.Columns(column).EntireColumn.AutoFit()
    column = column + 1
Next
excel.Rows("1").Font.Bold = True
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 89

Autodesk

共有パラメーター

パラメーター

- 共有パラメーターを使用してRevitの要素に要素ごとのデータを追加
- これらのパラメーターをサードパーティー アプリケーションへ書き出し、また読み込む

Lab 4-3 (3 コマンド)

- SDK サンプルのFireRating と類似
- 全てのドアに共有パラメータを作成
- 共有パラメーターデータを Excel に書き出す
- 共有パラメーターデータを Excel から読み込む

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 90

Autodesk

共有パラメーターの作成

パラメーター

- 共有パラメーター定義ファイルを取得


```
Class Parameters.DefinitionFile
Application.Options.SharedParametersFilename
Application.OpenSharedParameterFile
```
- 共有パラメーターグループを取得


```
Class Autodesk.Revit.Parameters.DefinitionGroup
DefinitionFile.Groups
DefinitionFile.Groups.Create
```
- 共有パラメーター定義を取得


```
Class Parameters.Definition
DefinitionGroup.Definitions
DefinitionGroup.Definitions.Create
```
- ドアにバインドするためにカテゴリーセットを作成


```
CategorySet = revitApp.Create.NewCategorySet()
catSet.Insert( doc.Settings.Categories.Item( BuiltInCategory.OST_Doors ) )
```
- パラメーターをバインド


```
Dim binding As Parameters.Binding = revitApp.Create.NewInstanceBinding( catSet )
doc.ParameterBindings.Insert( fireRatingParamDef, binding )
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 91

Autodesk

共有パラメーターの書き出し

パラメーター

- Excel を起動
- 'Doors' カテゴリーの全ての標準ファミリーインスタンスを取得
- 値にアクセスするため共有パラメーターの GUID を判定


```
Dim guid As Guid = guid.Empty
Dim file As Parameters.DefinitionFile = revitApp.OpenSharedParameterFile
Dim group As Parameters.DefinitionGroup = file.Groups.Item(defGroup)
Dim definition As Parameters.Definition = group.Definitions.Item(defName)
Dim externalDefinition As Parameters.ExternalDefinition = definition
guid = externalDefinition.GUID
```
- ループして全てのドアに対してExcel行に書き出す
 - ID / レベル / タグ / 耐火等級


```
worksheet.Cells(row, 1).Value = door.Id.Value
worksheet.Cells(row, 2).Value = door.Level.Name
Dim tagParameter As Parameter = door.Parameter( BuiltInParameter.ALL_MODEL_MARK )
worksheet.Cells(row, 3).Value = tagParameter.AsString
Dim parameter As Parameter = door.Parameter( paramGuid )
worksheet.Cells(row, 4).Value = parameter.AsDouble
row = row + 1
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 92

Autodesk

共有パラメーターの読み込み

パラメーター

- ファイルを選択、Excel を起動、ファイルを開く
- データーの読み込み

```
Dim id As Integer
Dim fireRatingValue As Double
Dim row As Integer = 2
Do
    ' Extract relevant XLS values
    id = worksheet.Cells(row, 1).Value
    If id <= 0 Then Exit Do
    fireRatingValue = worksheet.Cells(row, 4).Value
    ' Get document's door element via Id
    Dim elementId As Autodesk.Revit.ElementId
    elementId.Value = id
    Dim door As Autodesk.Revit.Element = revitApp.ActiveDocument.Element( elementId )
    ' Set the param
    If Not (door Is Nothing) Then
        Dim parameter As Parameter = LabUtils.GetElemParam( door, sharedParamName )
        parameter.Set( fireRatingValue )
    End If
    row = row + 1
Loop
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 93

Autodesk

モデル共有パラメーター

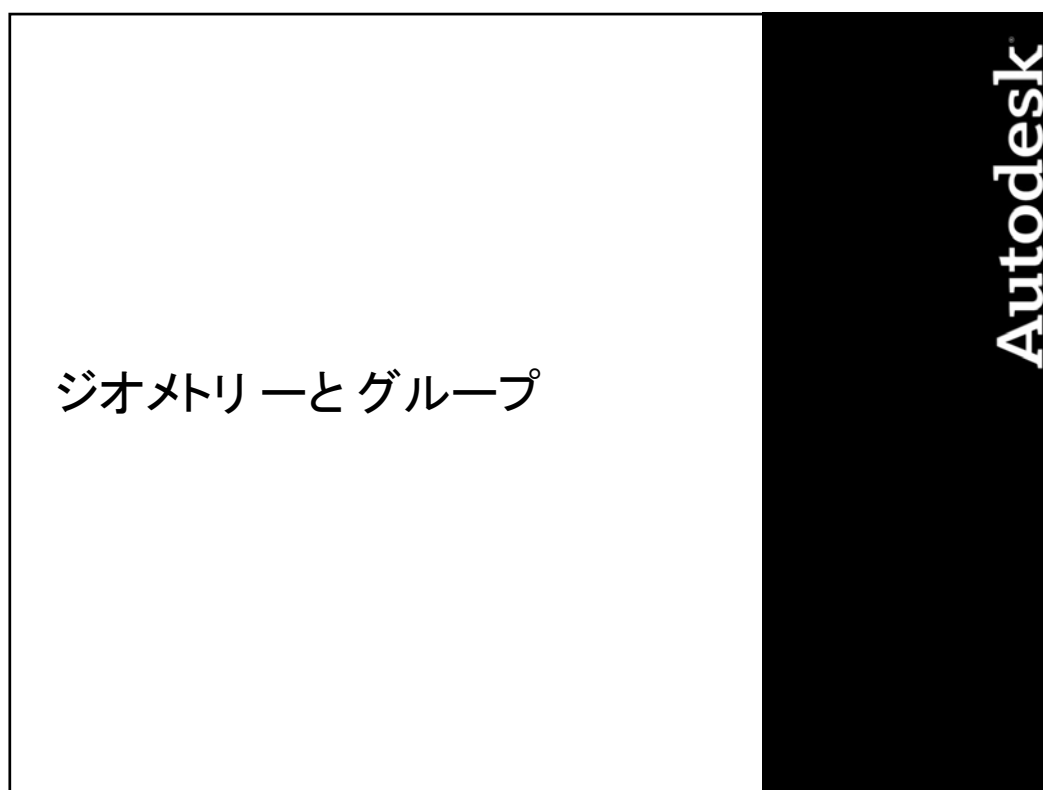
パラメーター

- 共有パラメーターを使用してどのようにドキュメント毎(モデルごと)のデーターを保存するか？
- 単一(singleton)要素である“Project Information” カテゴリに非表示共有パラメーターをバインド
- 共有パラメーター定義は表示属性を持つ ExternalDefinitionを使用する

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 94

Autodesk



グループ

ジオメトリとグループ

- モデルで利用できるタイプを決定
- メンバーである要素を決定
- グループインスタンスタイプを編集

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 96

Autodesk

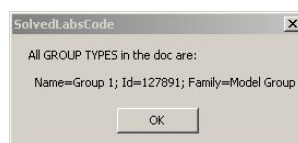
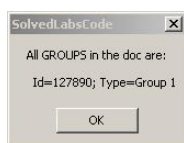
グループとグループタイプをリスト

ジオメトリとグループ

■ グループやグループタイプで標準要素フィルター

```
List<Element> elements = new List<Element>();
Filter filterType = app.Create.Filter.NewTypeFilter( typeof( Group ) );
app.ActiveDocument.get_Elements( filterType, elements );
```

Lab 5-1



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 97

Autodesk

モデル グループをリスト

ジオメトリとグループ

- 標準要素フィルター
- SYMBOL_FAMILY_NAME_PARAM をチェックを追加

```
List<Element> elements = new List<Element>();
Filter filterType = app.Create.Filter.NewTypeFilter( typeof( GroupType ) );
Filter filterParam = app.Create.Filter.NewParameterFilter(
    BuiltInParameter.SYMBOL_FAMILY_NAME_PARAM, CriteriaFilterType.Equal,
    "Model Group" ); // BEWARE: In the browser, it says only "Model"
Filter filterAnd = app.Create.Filter.NewLogicAndFilter( filterType, filterParam );
app.ActiveDocument.get_Elements( filterAnd, elements );
```

Lab 5-1



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 98

Autodesk

グループタイプの入れ替え

ジオメトリとグループ

- グループ オブジェクトの GroupTypeプロパティーを設定

```
Dim gp As Group = elem
Dim gt As GroupType
gp.GroupType = gt
```

Lab 5-2



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 99

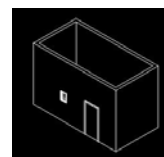
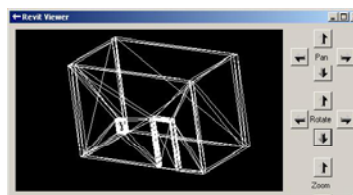
Autodesk

3D ジオメトリにアクセス

ジオメトリとグループ

Revit SDK のビューアー サンプル

- RevitViewer
 - シンプルなジオメトリビューアー ヘルプ クラス
 - 他のビューアー サンプルが使用
- ElementViewer
 - 単一、または複数選択された要素のワイヤーフレームモデルを表示
- RoomViewer
 - 選択された部屋のワイヤーフレームビューアー
- AnalyticalViewer
 - 単一、または複数選択された要素の解析モデルを表示
- ObjectViewerFamilyExplorer
 - 解析または物理モデルを表示して選択された要素のパラメーターを設定
- FamilyExplorer
 - 要素のプロファイルを表示



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 100

Autodesk

3D 要素ジオメトリを取得

ジオメトリとグループ

```
Private mOptions As Autodesk.Revit.Geometry.Options
mOptions = app.Create.NewGeometryOptions
mOptions.DetailLevel = Geometry.Options.DetailLevels.Fine

For Each elem In selSet
    Dim geom As Autodesk.Revit.Geometry.Element = elem.Geometry(mOptions)
    DrawElement(geom)
Next

If selSet.Size > 0 Then
    mViewer.ShowModal()
End If
```

- 要素ジオメトリの詳細レベルを指定
 - 簡略 / 標準 / 詳細
- あとはジオメトリの表示を行うのみ

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 101

Autodesk

要素ジオメトリを表示

ジオメトリとグループ

```
Private Sub DrawElement(ByVal elementGeom As Geometry.Element)
    Dim geomObject As Autodesk.Revit.Geometry.GeometryObject
    For Each geomObject In elementGeom.Objects
        If (TypeOf geomObject Is Autodesk.Revit.Geometry.Curve) Then
            DrawCurve(geomObject)
        ElseIf (TypeOf geomObject Is Autodesk.Revit.Geometry.Instance) Then
            DrawInstance(geomObject)
        ElseIf (TypeOf geomObject Is Autodesk.Revit.Geometry.Mesh) Then
            DrawMesh(geomObject)
        ElseIf (TypeOf geomObject Is Autodesk.Revit.Geometry.Solid) Then
            DrawSolid(geomObject)
        End If
    Next
End Sub

Private Sub DrawCurve(ByVal geomCurve As Geometry.Curve)
    DrawPoints(geomCurve.Tessellate)
End Sub
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 102

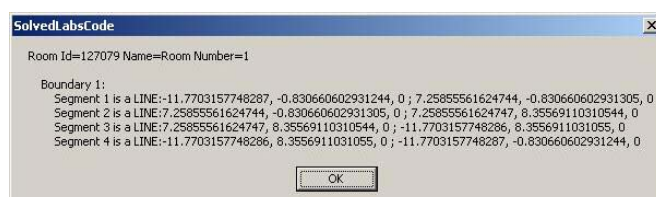
Autodesk

2D ジオメトリにアクセス

ジオメトリとグループ

- 部屋のId、名前、番号を表示
- 部屋の境界線をイタレートする
- SDK サンプル RoomViewer を参考

Lab 5-3



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 103

Autodesk

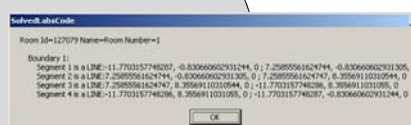
部屋境界をイタレート

ジオメトリとグループ

```

Dim boundaries As BoundarySegmentArray = room.Boundary
Dim boundary As BoundarySegmentArray
Dim iB As Integer = 0
For Each boundary In boundaries
    iB += 1
    sMsg += vbCrLf & " Boundary " & iB & ":"
    Dim iSeg As Integer = 0
    Dim segment As BoundarySegment
    For Each segment In boundary
        iSeg += 1
        Dim crv As Geometry.Curve = segment.Curve
        If TypeOf crv Is Geometry.Line Then ' LINE
            Dim line As Line = crv
            Dim ptS As XYZ = line.EndPoint(0)
            Dim ptE As XYZ = line.EndPoint(1)
            sMsg += vbCrLf & " Segment " & iSeg & " is a LINE:" & _
                ptS.X & ", " & ptS.Y & ", " & ptS.Z & "; " & _
                ptE.X & ", " & ptE.Y & ", " & ptE.Z
        ElseIf TypeOf crv Is Geometry.Arc Then ' ARC
            Dim arc As Arc = crv
            Dim ptS As XYZ = arc.EndPoint(0)
            Dim ptE As XYZ = arc.EndPoint(1)
            Dim r As Double = arc.Radius
            sMsg += vbCrLf & " Segment " & iSeg & " is an ARC:" & _
                ptS.X & ", " & ptS.Y & ", " & ptS.Z & "; " & _
                ptE.X & ", " & ptE.Y & ", " & ptE.Z & "; R=" & r
        End If
    Next
Next

```



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 104

Autodesk

5分間休憩いたします

休憩後は

API アップデート

- リボン、イベント、VSTA

API 新機能

- ファミリーAPI とフォーム作成

の説明を行います



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 105

Autodesk

API アップデート

更新された API

Autodesk

Revit 2010 API

API アップデート

更新されたAPI

- リボン UI
- イベント
- VSTA
- その他

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 107

Autodesk

リボン API

ユーザーインターフェースのカスタマイズ

Autodesk

リボン API

リボンAPI

- UI のカスタマイズはリボン API でのみ可能
 - メニューとツールバーはリボンに移行する必要あり
- WPF の知識は必要なし
- UI カスタマイズのガイドラインを用意
 - Ribbon design guidelines.pdf
 - Autodesk Icon Guidelines.pdf
- 将来的なリリースで拡張予定

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 109

Autodesk

リボン パネルの追加

リボンAPI

- 外部アプリケーションのリボンパネルをアドイン タブに追加
- リボンパネルの要素を追加
 - ボタン
 - ドロップダウン リスト ボタン
 - セパレーター
- リボンタブの追加は不可

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 110

Autodesk

リボンクラス

リボンAPI

RibbonPanel

- アドイン タブのリボンパネル

PushButton

- ボタン

PushButtonData

- ボタンの情報

PulldownButton

- ドロップダウン リスト ボタン

PulldownButtonData

- ドロップダウン リスト ボタンの情報

Copyright © 2009 Autodesk Inc.

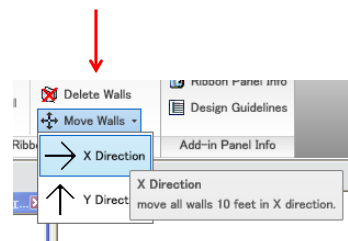
Revit Programming Introduction 111

Autodesk

SDK リボン サンプル

リボンAPI

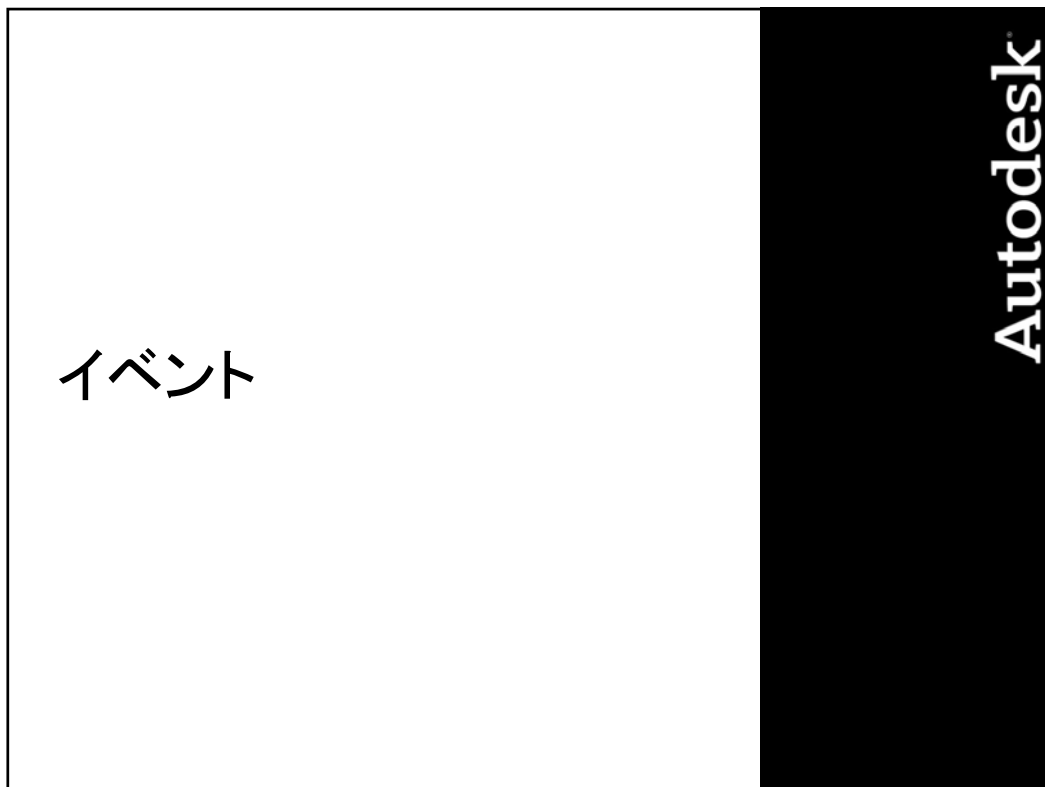
- [Revit SDK]¥Samples¥Ribbon
- 外部アプリケーションの2つのリボンパネル
- “Create Wall” ボタン
- スタックされた3つボタン
- “Move Walls” ドロップダウン リスト ボタン
- ツールチップ
- セパレーター



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 112

Autodesk



イベント

Revit 2010 に新規追加されたイベント

- .NET イベント標準に準拠
- “pre” イベントで処理のキャンセルが可能
- 既存のイベントは非推奨 (*deprecated*)
- 現在VSTA では未サポート

Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 114

Revit 2010 に新規追加されたイベント

イベント

- ファイル書き出し
 - Application.FileExported, Application.FileExporting
- ファイル読み込み
 - Application.FileImported, Application.FileImporting
- ビュー印刷
 - Application.ViewPrinting
- ドキュメント印刷
 - Application.DocumentPrinting, Application.DocumentPrinted
- ビューのアクティブ化
 - Application.ViewActivating, Application.ViewActivated
- 中央ファイルへの保存
 - DocumentSynchronizingWithCentral, DocumentSynchronizedWithCentral

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 115

Autodesk

SDK のイベント サンプル

イベント

ApplicationEvents サンプルを削除して Samples¥Event フォルダに5つのサンプルを追加

- AutoStamp
 - ビューが印刷された時にドキュメントを自動的にスタンプ
- AutoUpdate
 - DocumentOpened イベントを監視してイベントハンドラーでモデルを編集
- CancelSave
 - DocumentSaving と DocumentSavingAs 引数でドキュメント保存処理をキャンセル
- Event Monitor
 - 全てのアプリケーションレベルのイベントを監視してウィンドに表示
- PrintLog
 - 印刷に関連するイベントを監視して印刷に関する情報をログファイルに出力

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 116

Autodesk

イベントサンプル : ドキュメント保存を回避 イベント

- DocumentSaving イベントを監視
- ドキュメントを保存するかを判断
- イベントの監視解除

Lab 6

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 117

Autodesk

VSTA

Visual Studio Tools for Applications

Autodesk

VSTA

Visual Studio Tools for Applications

- マイクロソフト社のテクノロジー
- アプリケーションの拡張を提供
- NET フレームワークを基に構築
- 製品に統合
- Visual Studio UI
- VBAの後継の位置付け

Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 119

VSTA

Revit VSTA

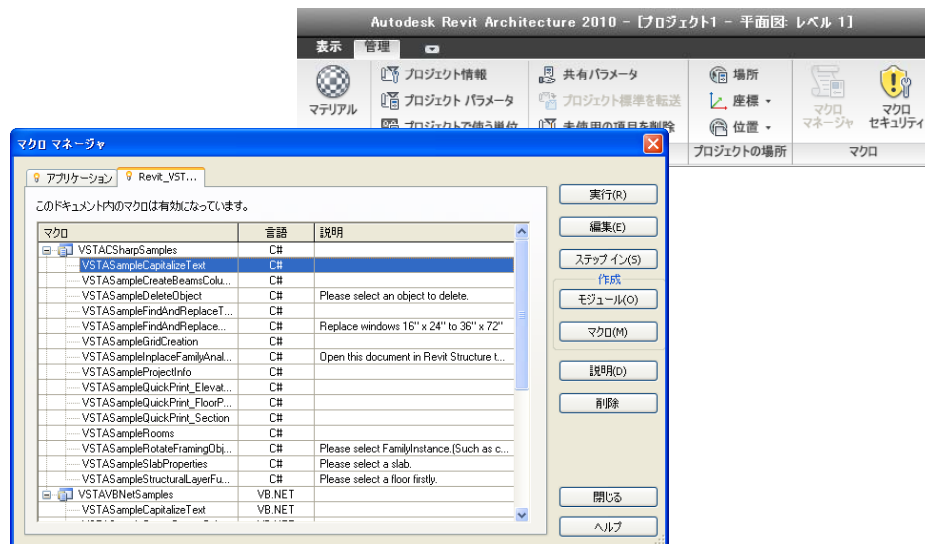
- 2つのレベルのマクロ
 - ドキュメントレベル – .rvt プロジェクトファイルに保存
 - アプリケーション レベル – <Revit install>\Program\%VstaMacros%\AppHookup に保存
- 外部コマンドとほぼ同様なシンタックス
- インストール
 - 既定値として自動でインストール
 - 全てのRevit 製品に実装
- ドキュメント
 - Revit Architecture ユーザーガイド > Revit VSTA でマクロを作成
- サンプル
 - Revit SDK > VSTA Samples フォルダ
 - 21 ドキュメント マクロ
 - 2 アプリケーション マクロ
 - C# とVB.NET をサポート

Autodesk

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 120

Revit VSTA マクロ マネージャー

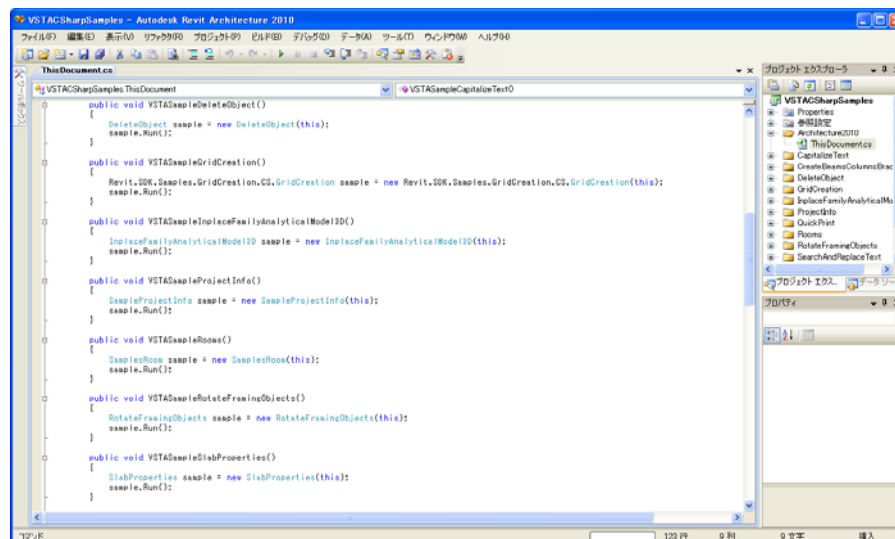
管理タブ > マクロ マネージャー



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 121

Revit VSTA IDE

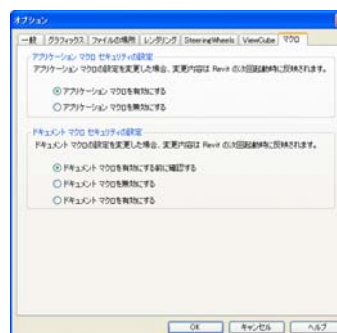


Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 122

2010 VSTA機能

- VSTA 2.0 の使用
- マクロ セキュリティー設定
 - アプリケーションマクロの有効/無効化
 - ドキュメントマクロの有効/無効化
- 複数のVSTA IDEを起動可能
- マクロ環境ではイベントは未サポート



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 123

Autodesk

Revit VSTA ドキュメント



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 124

Autodesk



API 新機能

Revit 2010 API 新機能


Revit 2010 API

API アップデート

新規追加された API

- ファミリー API
- フォーム作成 API
- MEP API
- モデル検査ユーティリティ

Copyright © 2009 Autodesk Inc. Revit Programming Introduction 126



新しいサンプル

Revit SDK の新しいサンプル

ファミリー API

- AutoJoin
- AutoParameter
- CreateAirHandler (rme)
- CreateTruss (rst)
- DWGFamilyCreation
- GenericModelCreation
- TypeRegeneration
- ValidateParameters
- WindowWizard

フォーム作成 API

- DistanceToPanels
- MeasurePanelArea
- NewForm (5 コマンド)
- PanelEdgeLengthAngle

イベント

- AutoStamp
- AutoUpdate
- CancelSave
- EventsMonitor
- PrintLog

Revit MEP

- AutoRoute
- AvoidObstruction
- TraverseSystem

その他

- RaytraceBounce
- Ribbon

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 127

Autodesk

ファミリーAPI

ファミリー作成

Autodesk

ファミリー API

ファミリーAPI

- 最も公開の要望が多かったAPI
- ファミリードキュメントの作成や編集
- プロジェクトにロードされたファミリーの編集
- 注意: インプレースファミリーのアクセスは未公開



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 129

Autodesk

ドキュメント クラスの更新

ファミリーAPI

ドキュメント クラスに追加されたメソッドとプロパティー

- Autodesk.Revit.Document クラス
 - FamilyCreate – 要素作成
 - FamilyManager – ファミリーマネージャーオブジェクトを取得
 - IsFamilyDocument – ファミリードキュメントを識別
 - OwnerFamily – ファミリードキュメントのファミリー名
 - EditFamily – プロジェクトにロードされたファミリーを編集
 - LoadFamily – ファミリーのロード

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 130

Autodesk

ファミリー マネージャー

ファミリーAPI

ファミリーマネージャーでタイプやパラメーターを編集

- Autodesk.Revit.FamilyManager クラスで定義
- Document.FamilyManager プロパティで取得
- ファミリのタイプの追加、削除、名前変更
- ファミリーパラメーター、共有パラメーターの追加と削除
- パラメータ値や式の設定

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 131

Autodesk

要素の作成

ファミリーAPI

要素ファクトリーで要素をファミリーに作成

- Autodesk.Revit.Creation.FamilyItemFactory クラスで定義
- Document.FamilyCreate プロパティで取得
- 各要素作成用のメソッドを公開
- 要素の種類:
 - 寸法
 - 注釈
 - 曲線
 - 参照面
 - 文字
 - ソリッド
 - ファミリーインスタンス
 - 位置合わせ
 - 接続 (MEP のみ)

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 132

Autodesk

要素の表示設定

ファミリーAPI

個々の要素で独自の表示設定

- Autodesk.Revit.FamilyElementVisibility クラス
- 要素のSetVisibility メソッドを使用
- 表示設定
 - 詳細レベル
 - ビュー

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 133

Autodesk

ファミリー API サンプル

ファミリーAPI

SDK サンプルのFamilyCreation フォルダー

AutoJoin

- ファミリーモデリングとマスの使用に複数の一般フォームのジオメトリを自動的に結合

AutoParameter

- ファミリードキュメントにファミリー、共有パラメーターを追加

CreateAirHandler – RME

- パイプとダクト接続で空気処理機を作成

CreateTruss – RST

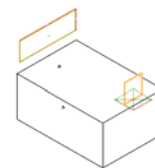
- トラスファミリードキュメントにモノトラスを作成

DWGFAMILYCreation

- DWG ファイルをファミリードキュメントに読み込み、インスタンスにタイプパラメーターを追加

GenericModelCreation

- 押し出し、ブレンド、回転、スイープを作成



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 134

Autodesk

ファミリー API サンプル

ファミリー API

SDK サンプルのFamilyCreation フォルダー（続き）

TypeRegeneration

- タイプ作成を試みて結果をレポート

ValidateParameters

- 全タイプでパラメーターの値をチェックして結果をレポート

WindowWizard

- ウィザードUIを実装してウィンドファミリーを作成



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 135

Autodesk

フォーム作成

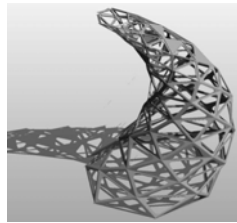
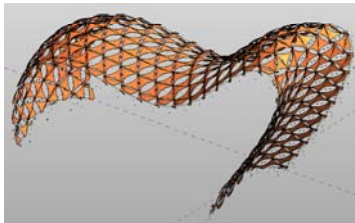
フォーム作成 API

Autodesk

フォーム作成 API

フォーム作成

- 新しい点と曲線オブジェクト
- サーフェスの分割、パターン化、パネル化



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 137

Autodesk

ReferencePoint と CurveByPoints

フォーム作成

- 参照点の作成方法
 - XYZ
 - トランスフォーム
 - エッジ
 - エッジとエッジの交点
 - エッジと面の交点
 - 面
 - 平面
- ReferencePointArray – 曲線作成の参照点の配列
- NewCurveByPoints(ReferencePointArray points)

Copyright © 2009 Autodesk Inc.


Revit Programming Introduction 138

Autodesk

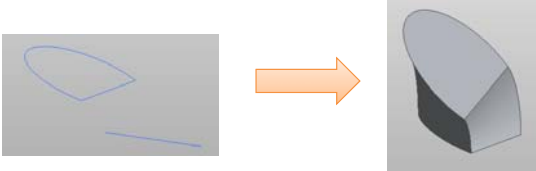
フォーム

フォーム作成

- 押し出し
 - NewExtrusionForm(bool isSolid, ReferenceArray profile, XYZ direction)



- 回転
 - NewRevolveForm(bool isSolid, ReferenceArray profile, Reference axis, double startAngle, double endAngle);



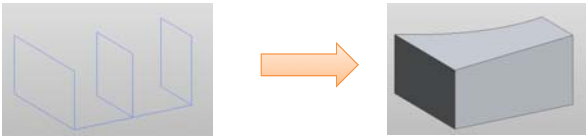
Copyright © 2009 Autodesk Inc. Revit Programming Introduction 139

Autodesk

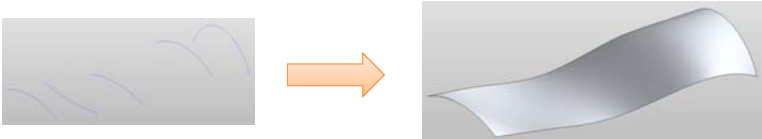
フォーム

フォーム作成

- スイープ
 - NewSweptBlendForm(bool isSolid, ReferenceArray path, ReferenceArray profiles)



- ロフト
 - NewLoftForm(bool isSolid, ReferenceArray profiles)



Copyright © 2009 Autodesk Inc. Revit Programming Introduction 140

Autodesk

フォーム

フォーム作成

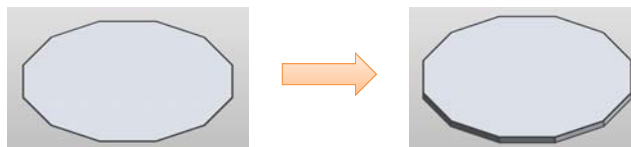
■ キャップ サーフェス

- NewFormByCap(bool isSolid, ReferenceArray profile)



■ 厚みのあるキャップサーフェス

- NewFormByThickenSingleSurface(bool isSolid, Form singleSurfaceForm, XYZ thickenDir)



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 141

Autodesk

フォームの修正

フォーム作成

■ サブ要素の削除、移動、回転、縮尺

```

while (i < formLoft.ProfileCount)
{
    int move = 3;
    ReferenceArray ref_Array
        = formLoft.get_CurveLoopReferencesOnProfile( i, iCurveLoopIndex );
    Reference loft_ref = ref_Array.get_Item(0);
    while (move < 14)
    {
        XYZ XYZmove = new XYZ(move, 0, 0);
        formLoft.MoveSubElement(loft_ref, XYZmove);
        doc.Save();
        XYZmove = new XYZ(0, move, 0);
        formLoft.MoveSubElement(loft_ref, XYZmove);
        doc.Save();
        XYZmove = new XYZ(0, 0, move);
        formLoft.MoveSubElement(loft_ref, XYZmove);
        doc.Save();
        move = move + 2;
    }
    move = 0;
    i = i + 1;
}

```

Copyright © 2009 Autodesk Inc.

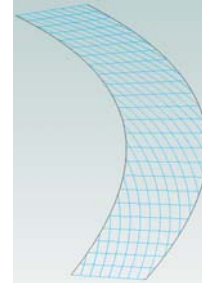
Revit Programming Introduction 142

Autodesk

サーフェスを分割

フォーム作成

- AllGridRotation、USpacingRule、VSpacingRule で定義された平行四辺形のグリッド
- NewDividedSurface(Reference hostReference)



```
form = e as Autodesk.Revit.Elements.Form;
DividedSurfaceData dsData = form.GetDividedSurfaceData();
if (dsData != null)
{
    foreach (Reference r in dsData.GetReferencesWithDividedSurfaces())
    {
        dsRefs.Append(r);
    }
}
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 143

Autodesk

パターンを適用

フォーム作成

- ObjectType で定義したグリッドにTilePatternを適用
- 分割したサーフェスに TilePattern を適用
TilePatternIteration サンプル



```
int count = 0;
TilePatterns tilepatterns = doc.Settings.TilePatterns;
foreach (Autodesk.Revit.Enums.TilePatternsBuiltIn TilePatternEnum in
    Enum.GetValues(typeof(Autodesk.Revit.Enums.TilePatternsBuiltIn)))
{
    if (count.Equals(3))
    {
        pDividedSurface.ObjectType
            = tilepatterns.GetTilePattern( TilePatternEnum );
        break;
    }
    ++count;
}
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 144

Autodesk

カーテン パネル

フォーム作成

- 壁が窓やドアのホストのようにグリッドがホスト
- 分割したサーフェスに適用するとサーフェスのグリッドにパネルのファミリーインスタンスを作成
- 新しいプロパティ
 - Family.IsCurtainPanelFamily
 - Family.CurtainPanelHorizontalSpacing
 - Family.CurtainPanelVerticalSpacing
 - Family.CurtainPanelTilePattern



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 145

Autodesk

カーテン パネルの例

フォーム作成

```
GridNode gn = new GridNode();
int u = 0;
while (u < pDividedSurface.NumberOfUGridlines)
{
    gn.UIndex = u;
    int v = 0;
    while (v < )
    {
        gn.VIndex = v;
        if (pDividedSurface.IsSeedNode(gn))
        {
            FamilyInstance familyinstance
                = pDividedSurface.GetTileFamilyInstance(gn, 0);
            if (familyinstance != null)
            {
                // implement logic here with each tile for example
                // change the type to glazing
                familyinstance.Symbol = pFamilySumbol.glazed;
            }
        }
        v = v + 1;
    }
    u = u + 1;
}
```

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 146

Autodesk

フォーム作成 SDK サンプル

フォーム作成

- [Revit SDK]¥Samples¥Massing
- DistanceToPanels
 - 選択オブジェクトから全ての分割したサーフェスパネルの距離を計測
- MeasurePanelArea
 - 分割サーフェスの計測と修正
- NewForm (5 つのコマンド)
 - 押し出し、回転、スイープ、ロフトの作成
- PanelEdgeLengthAngle
 - 分割サーフェスの計測

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 147

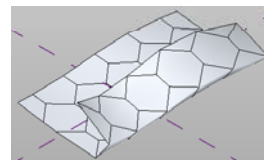
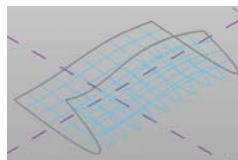
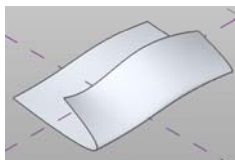
Autodesk

フォーム作成 サンプル

フォーム作成

- 参照点と曲線を使用してロフトを作成
- サーフェスの分割
- 分割されたサーフェスにパターンを適用

Lab 7



Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 148

Autodesk

参考資料

DevTV Introduction to Revit Programming

- <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2484975>

Discussion Groups

- <http://discussion.autodesk.com> > Revit Architecture > Revit API

The Building Coder, Jeremy Tammik's Revit API Blog

- <http://thebuildingcoder.typepad.com>

Autodesk Developer Network

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

ADN DevHelp Online

- <http://adn.autodesk.com>

ADN JDevHelp (日本語)

- <http://www.autodesk.co.jp> (→ パートナー → ADN → OTW → ADN)

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 149

Autodesk

アンケートにご協力下さい

1. プレゼンテーションの音質は？
2. プレゼンテーションの画質は？
3. プレゼンテーションの材料は？
4. プレゼンテーションの内容と進行は？

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 150

Autodesk

ご質問はありますか？

* 6でミュートを解除してお話下さい

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 151

Autodesk

ご参加ありがとうございました!

Copyright © 2009 Autodesk Inc.

Revit Programming Introduction 152

Autodesk

The Autodesk logo is centered within a solid black rectangular frame. The word "Autodesk" is written in a white, bold, serif typeface. A small registered trademark symbol (®) is positioned to the upper right of the final letter 'k'.

Autodesk®