

The Revit 2014 API

Jeremy Tammik
Developer Technical Services

Agenda

Introduction

Revit 2014 (aka Equinox) Development Themes

The Rice | Revit Equinox application compatibility

The Wine | Revit Equinox features + API

About Me

Jeremy Tammik

Developer Technical Services
EMEA, Autodesk



Jeremy is a member of the AEC workgroup of the DevTech team, providing developer support, training, and conferences to the Autodesk Developer Network ADN. He originally joined Autodesk in 1988 as the technology evangelist responsible for European developer support. In this capacity, he wrote articles, consulted, lectured on AutoCAD application programming techniques, and supported AutoCAD application developers in Europe, the U.S., Australia, and Africa. He was a cofounder of ADGE, the AutoCAD Developer Group Europe, and a prolific author on AutoCAD application development. He left Autodesk in 1994 to work as an HVAC application developer, and then rejoined the company in 2005.

Jeremy graduated with a MA in mathematics and physics in Marburg, Germany, in 1984, and worked first as a teacher and translator of both computer and human languages, then as a C++ programmer on early GUI and multitasking projects. He is fluent in five European languages, vegetarian, has four kids, plays the flute, likes reading, travelling, theatre improvisation and carpentry, loves mountains, oceans, sports and especially climbing.

Revit Products

- Four flavours of Revit
 - Revit Architecture
 - Revit MEP: Mechanical, Electrical, Plumbing
 - Revit Structure
 - Revit One-Box including all, part of Autodesk Building Design Suite Premium and Ultimate
- Product build and distribution
 - DVD version posted to ADN member web site
 - Software & Support > Autodesk Revit or Autodesk Building Design Suite > Downloads
 - Posted once only
 - Web and Web Update version on Autodesk home page
 - Autodesk home page > Support > Service Packs & Downloads
 - Autodesk home page > Revit Products > Product Trial
 - Latest download version from the public product site

The Revit API and SDK

- RevitAPI.dll and RevitAPIUI.dll are present in every Revit installation
- The SDK is provided with the product
 - From Installer under 'Install Tools and Utilities'
 - From the extraction of RTM download version
<extraction folder>\Utilities\SDK\Revit2014SDK.exe
- Download latest update from Developer Center
 - <http://www.autodesk.com/developrevit>

Revit SDK Documentation

Read Me First.doc

Getting Started with the Revit API.doc

Revit Platform API Changes and Additions.doc

RevitAPI.chm

Add-In Manager

Samples

- SamplesReadMe.htm
- SDKSamples2014.sln

Revit API Developer Guide in Autodesk Wiki Help

<http://wikihelp.autodesk.com/Revit/enu/2014>

My First Plugin

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

- A self-paced tutorial assuming no programming knowledge
- 7 lessons to get you started with Revit programming
- Video and lessons can be downloaded

My First Plug-in Training

 Share

 **My First Plug-In Training**

My First AutoCAD Plug-in Overview
My First Inventor Plug-in Overview
My First Revit Plug-in Overview
Lesson 1: The Basic Plug-in
Lesson 2: Programming Overview
Lesson 3: A First Look at Code
Lesson 4: Simple Selection of a Group
Lesson 5: Working with Room Geometry
Lesson 6: My Final Plug-in
Lesson 7: Learning More

My First Revit Plug-in Overview

Are you an Autodesk Revit power user with an interest in becoming even more productive? Would you like to automate or extend the capabilities of Revit, but are new to computer programming? If so, then this guide is designed for you.

"My First Plug-in" is a self-paced tutorial guide for a smooth introduction into the programming world. This is a "one-stop shop" learning path for users who know Autodesk products but are absolutely new to programming and are thinking about taking the plunge. In this guide, you will be working with Autodesk Revit .NET API and the C# programming language, however if you prefer the programming language VB.NET we have provided the code samples in this language as well. The supported product version is Autodesk Revit Architecture 2011 and succeeding versions.

Product: Autodesk Revit 2011 or higher
Programming Language: C# (and VB.NET - does not include written explanations for VB.NET code samples)
Application Programming Interface (API): Revit .NET API

More resources

Revit ADN Training Material

Revit training material for self-paced learning

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

DevTV and Webcast recordings

[Revit API Webcast archive](#)

Revit SDK Samples folder

Large number of sample projects on different topics

Look at SamplesReadme.htm

The Building Coder Revit API blog

<http://thebuildingcoder.typepad.com>

Revit Samples and Tools

SDK Visual Studio solution

Compile all samples in one fell swoop

RvtSamples

Load all samples for testing and debugging

RevitLookup

Interactive exploration of the Revit RVT database

Revit Equinox Development Themes

Platform Scalability

Support projects and teams of any size and complexity

Design | Construction | Fabrication

Expanding Revit capabilities for core and adjacent markets

Analysis and Simulation

Understanding and predicting building performance and behavior

Suites + Interoperability

Leveraging the Autodesk portfolio to support AEC projects

Countrification

Meeting regional AEC needs around the world

Revit Initiatives

Platform Scalability

- Performance

Design/Construction /Fabrication

- Piping for Construction
- Reinforcement
- Steel Detailing
- Concrete Modeling
- Displaced Views
- Point Clouds
- Customer

Suites & Interoperability

- Enhanced Visualization
- API Enhancements
- Autodesk Exchange

Analysis & Simulation

- Structural Analytical Model
- Analysis Visualization Framework

Requests

- Analytical Volumes
- MEP Calculations
- Physical Properties

Countrification

- Stairs & Railings
- Schedules

Platform Scalability

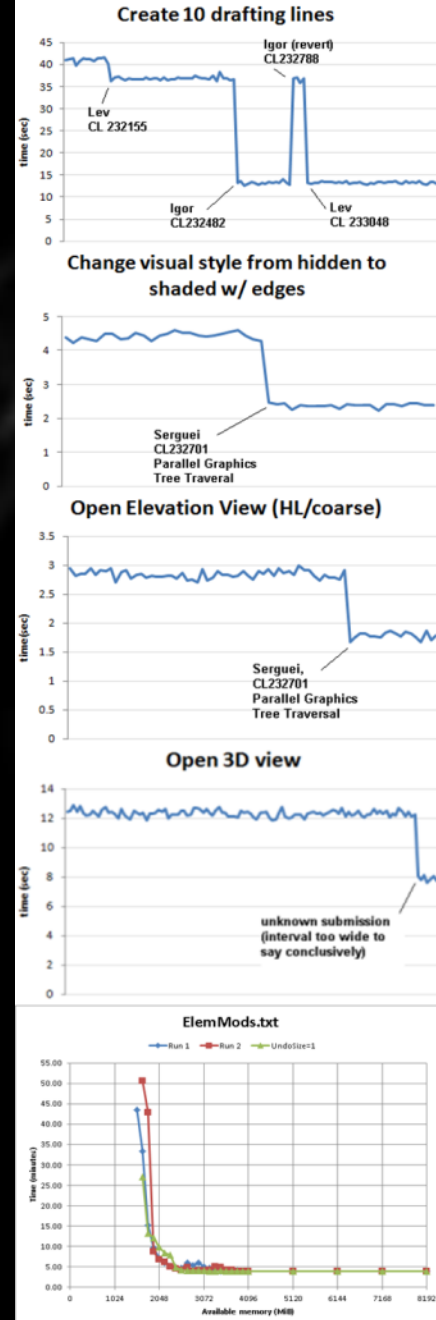
Performance

Customer Value

- More responsive Revit sessions
- Additional features become more practical

Deliverables

- Display optimization
- DWG & mesh handling improvements
- Family load performance



Design/Construction/Fabrication

Piping for Construction

Customer Value

- Higher level of detail in the piping model
- Improved workflow from design to fabrication
- Improved coordination/interference detection

Deliverables

- Standard parameters on pipe fittings
- Connection types and engagements lengths on fittings
- New pipe content based on standards



Design/Construction/Fabrication

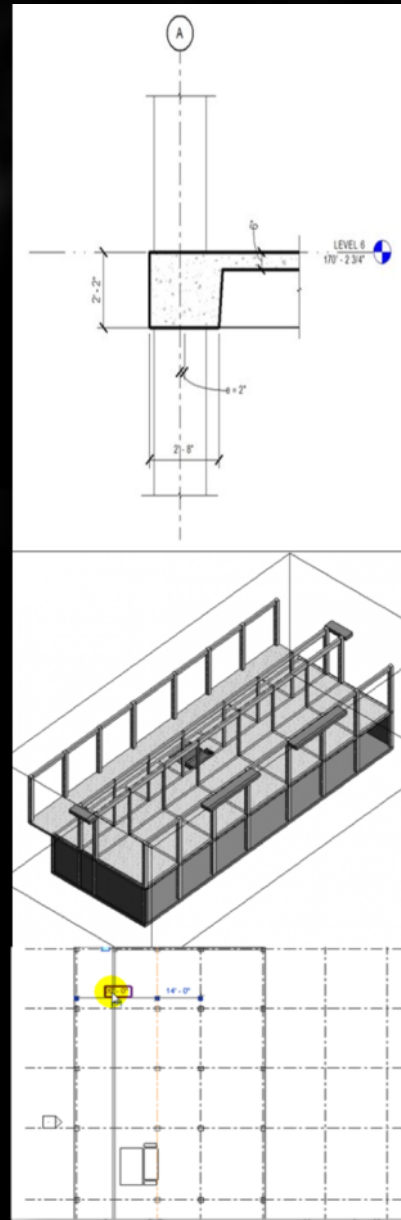
Concrete Modeling

Customer Value

- Accurately represent concrete design
- Indicate concrete construction sequencing

Deliverables

- Primacy of concrete joining



Design/Construction/Fabrication

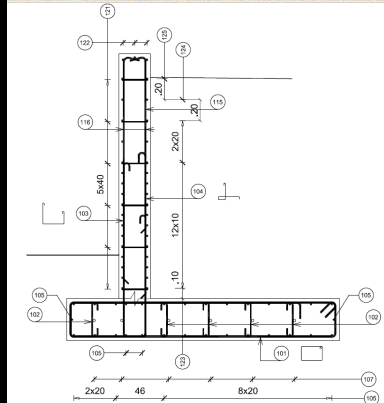
Reinforcement

Customer Value

- Accurate rebar schedule creation
- Shop drawing creation for typical elements
- More flexibility in Welded Wire Mesh (WWM) distribution

Deliverables

- Rebar shape definition according to European standards
- Reinforcement length tolerances for rebar and WWM
- New tagging type for liner rebar set
- Multi reference annotation
- Start point marking on the major direction symbol of WWM
- Override rebar placement rules



COUPE 4-4 Ech. 1/20

Design/Construction/Fabrication

Steel Detailing

Customer Value

- Integrity of data across structural workflow
- Accurate positioning for structural steel elements

Deliverables

- Precise positioning and geometry definition of steel



Design/Construction/Fabrication

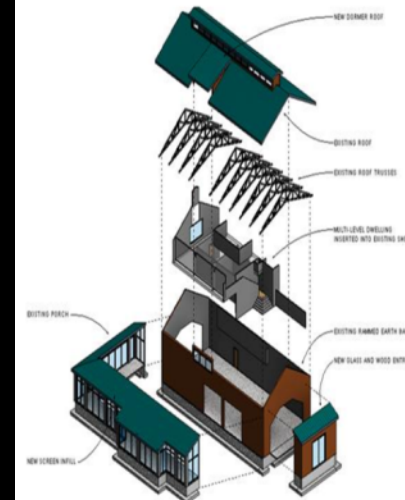
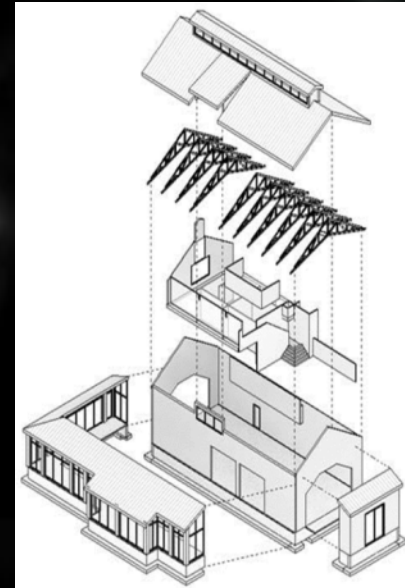
Displaced Views

Customer Value

- Improved visualization of building assemblies

Deliverables

- Configurable views of exploded models



Design/Construction/Fabrication

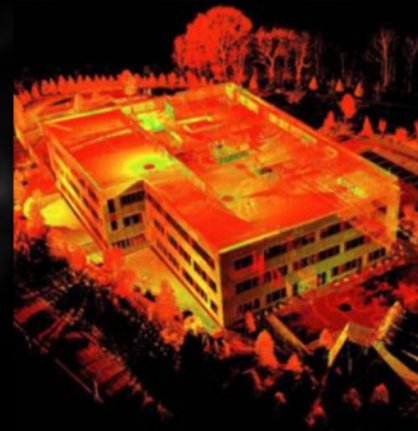
Point Clouds

Benefits

- Streamline as-built BIM authoring process for existing building renovation and retrofits
- Enhanced ability to connect construction planning & execution to BIM

Deliverables

- New point cloud engine
- Progressive display of point clouds



Design/Construction/Fabrication

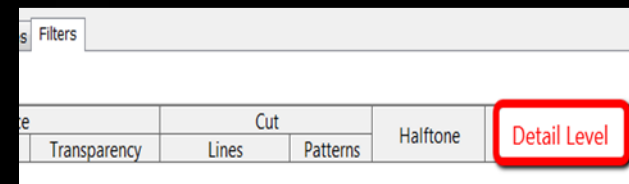
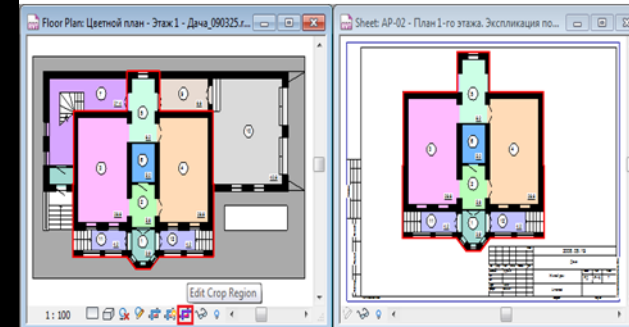
Customer Requests

Customer Value

- Transform the customer experience
- Increase industry focus to meet customer needs

Deliverables

- Non-rectangular crop regions
- Dimensions that can display dual units (Metric & Imperial)
- New Structural Annotation & Tag Types
- Multi-select of objects for “bring to front” & “send to back”
- Transparent objects cast shadows
- Better control & placement of 3D arcs



Suites & Interoperability

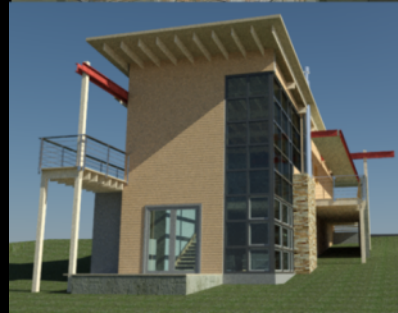
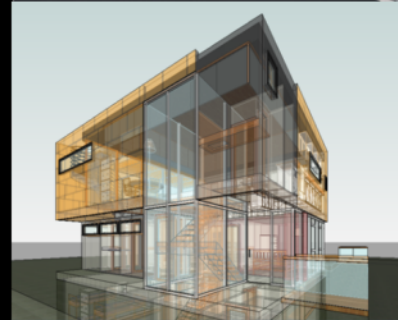
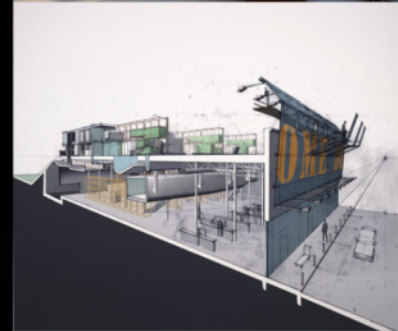
Enhanced Visualization

Customer Value

- Improved graphics performance
- Congruent visualization options
- Improved materials interaction

Deliverables

- Improved materials UI
- Improved ray tracing



Suites & Interoperability

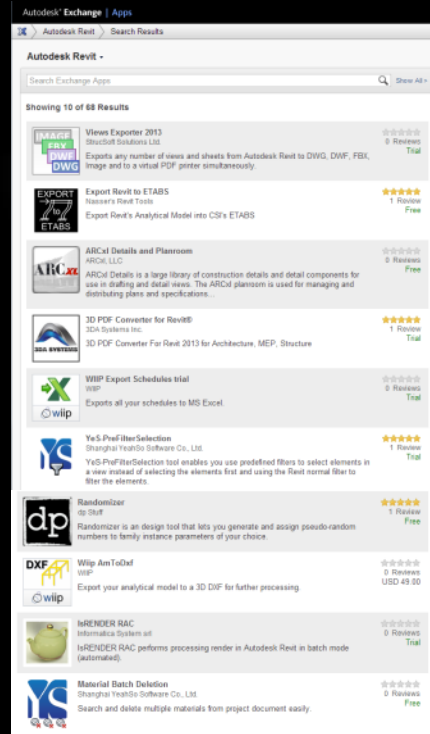
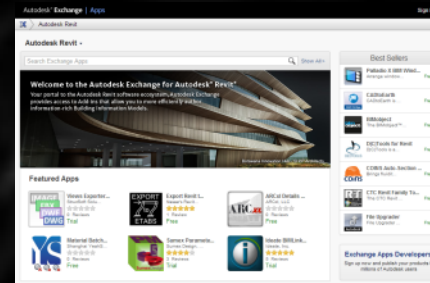
Autodesk Exchange

Customer Value

- Access to extended product functionality
- Expanded content library

Deliverables

- Improved download and install of applications
- Ability to load add-ins dynamically during a Revit session



Analysis & Simulation

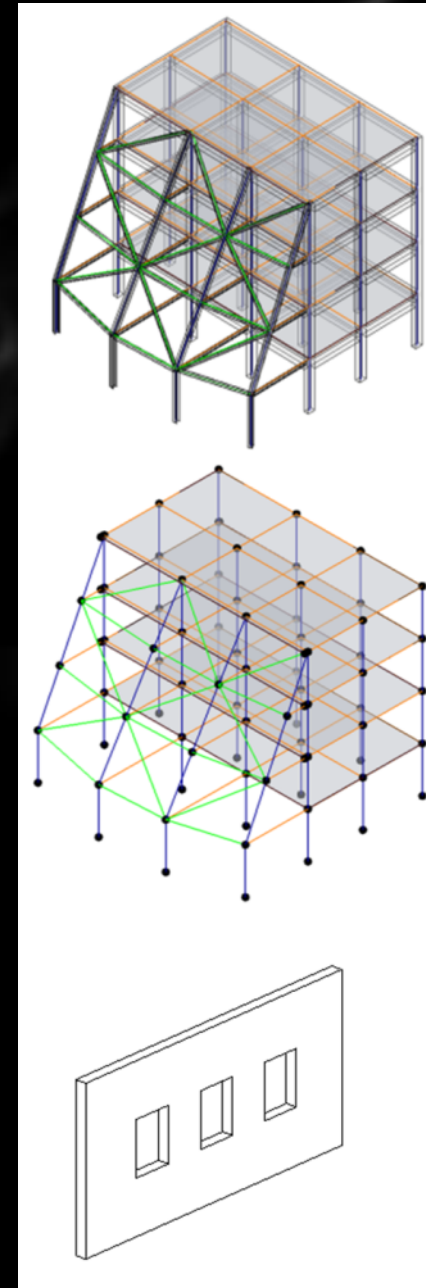
Structural Analytical Model

Customer Value

- Improved coordination of physical and analytical models
- Readiness for structural analysis directly from Revit models

Deliverables

- Improved analytical model behavior and usability
- Temporary view templates for better model visualization



Analysis & Simulation

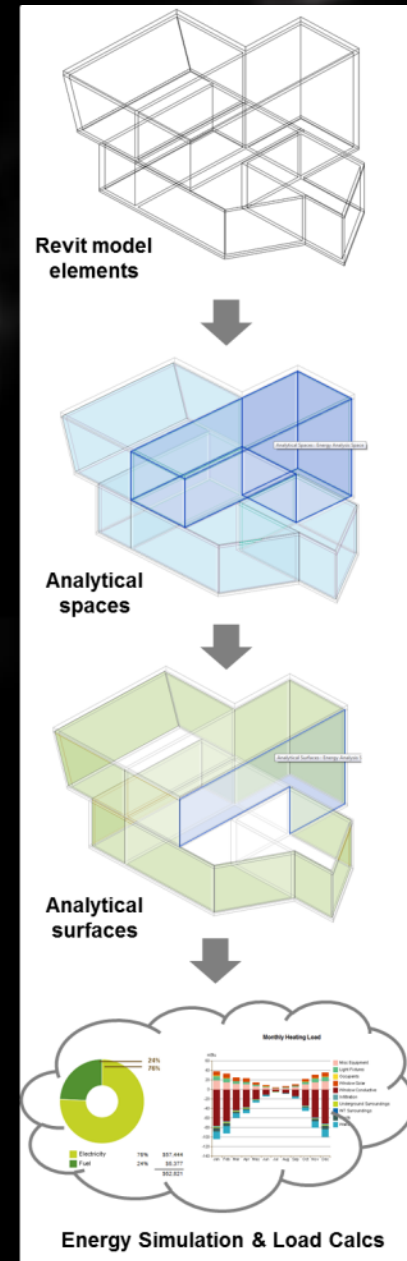
Analytical Volumes

Customer Value

- Ease of use with no time and skill wasted rebuilding models
- Trust in the reliability and repeatability of calculations & simulations

Deliverables

- Fast, reliable (thermal) analytical geometry components
- Improved the quality of gbXML export



Analysis and Simulation

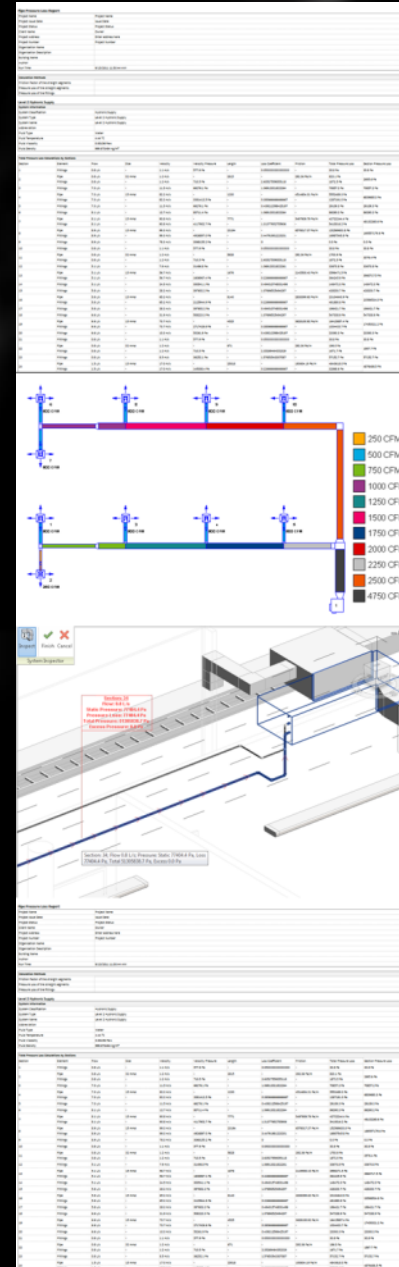
MEP Calculations

Customer Value

- Ability to use regional calculations within Revit MEP

Deliverables

- Moved duct and pipe sizing calculations to API
- Improve duct and pipe calculations
- Provide tools to validate calculations



Countrification

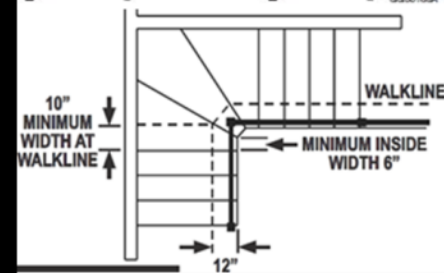
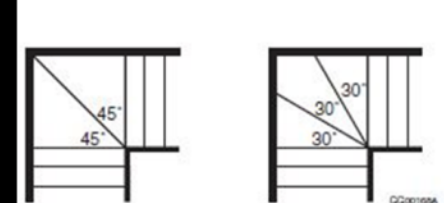
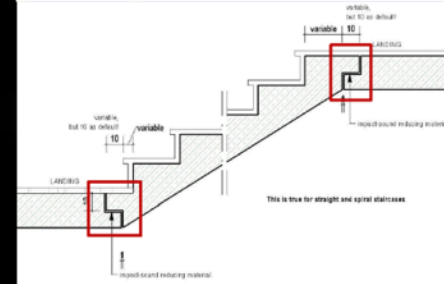
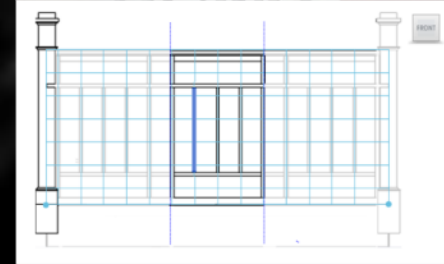
Stairs and Railings

Customer Value

- Easily model, edit and document stair & railing designs

Deliverables

- Railing pattern editing
- Stair fit & finish based on customer feedback



Countrification

Schedules

Customer Value

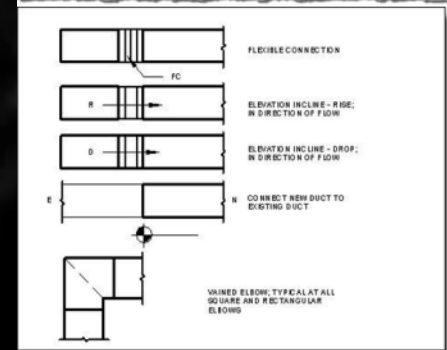
- Greater control of schedule formatting
- Increased access to data within the building information model

Deliverables

- Text properties and table formatting improvements
- Improved data access, display, and relationships
- Ability to display graphics in schedules
- Sorting, grouping, and filtering improvements

Long-term plan

- Reporting project

[illegible]

Door #	To Room	From Room	Size Width	Door Schedule		Frame Type
				Size Height	Hardware Group	
101A		101	1830	2134	3	
101B		101	1830	2134	3	
101C	101	102	1830	2134	3	
101D	101	102	1830	2134	3	
103	107	103	915	2134	(none)	
104A	104	107	915	2134	(none)	
104B	104	107	915	2134	(none)	
105A	105	107	915	2134	(none)	
105B	105	107	915	2134	(none)	
106A	106	107	915	2134	(none)	
106B	106	107	915	2134	(none)	
108A	108	107	915	2134	(none)	
108B	108	107	915	2134	(none)	
109	109	107	915	2134	(none)	
110	110	107	915	2134	(none)	
111	111	102	915	2134	(none)	
112	112	915	2134	2134	(none)	
114	114	102	915	2134	(none)	
115	115	102	1830	2134	(none)	
116	116	102	915	2134	(none)	
117	117	102	915	2134	(none)	
118	118	102	1730	2134	(none)	
119A	119	119	1730	2134	(none)	
119B	119	119	1730	2134	(none)	
120	120		1830	2134	(none)	
121	102	121	1830	2134	(none)	
122	121	122	915	2134	(none)	

	479	485	494	4936	5054	Document	
							Structural Column Schedule
60'-Floor							
FLOOR							
60'-Floor							
CORR							
60'-Bobby Level							
B							
Club Level							
CORR							
CLUB LEVEL CORRIDOR	47	48	49	50	51	52	53



The Rice

Revit Equinox application compatibility

The Rice

- Document Open + Save
- FilteredElementCollector
- Geometry API
- Family Symbol API
- Units API
- Materials API
- IFC API
- BeamSystem + Truss Creation
- Obsolete API cleanup

Document Open + Save

`Application.OpenDocumentFile(ModelPath, OpenOptions)`

`Application.OpenDocumentFile(String)`

`UIApplication.OpenAndActivateDocument(String)`

- Re-implemented with specific exceptions

`UIApplication.OpenAndActivateDocument(ModelPath, OpenOptions, Boolean bDetachAndPrompt)`

- New signature
- 12 exceptions

`Document.Save()`

`Document.SaveAs()`

- Return void instead of boolean
- Failures signaled with specific exceptions

FilteredElementCollector

Iteration and element deletion

- If a Revit Element is deleted while `FilteredElementCollector` iterates through document, continuing the iteration will throw an `InvalidOperationException`
- Prevents application crashes caused by changes in Revit's Element Table following Element deletion
- In general, it is best not to make any changes to the document while an iterator is running through the document.
- Affects
 - `FilteredElementIterator`
 - `FilteredElementIdIterator`
 - `FilteredElementCollector` foreach loops

Geometry API

	New Methods	Obsolete Methods
Curve Creation	Curve.CreateTransformed() Line.CreateBound() Line.CreateUnbound() Arc.Create() Ellipse.Create() NurbSpline.Create() HermiteSpline.Create()	Curve.Transformed Line.Bound Line.Unbound Autodesk.Revit.Creation.Application.NewLine() Autodesk.Revit.Creation.Application.NewLineBound() Autodesk.Revit.Creation.Application.NewLineUnbound() Autodesk.Revit.Creation.Application.NewArc() Autodesk.Revit.Creation.Application.NewEllipse() Autodesk.Revit.Creation.Application.NewNurbSpline() Autodesk.Revit.Creation.Application.NewHermiteSpline()
Curve Utilities	GetEndPoint() GetEndParameter() GetEndPointReference()	EndPoint EndParameter EndPointReference
Edge Utilities	Edge.GetFace() Edge.GetEndPointReference()	Edge.Face (property) Edge.EndPointReference (property)
Transform Initialization	Transform.CreateTranslation() Transform.CreateRotation() Transform.CreateRotationAtPoint() Transform.CreateReflection()	Transform.Translation (property) Transform.Rotation (property) Transform.Reflection (property)

Family Symbol API

- Family Symbols do not contain geometry until an instance exists in the document
- A Symbol's geometry will be empty and should not be accessed until it is active.
- Test if a symbol is active with `FamilySymbol.IsActive()`
- To activate a family symbol in order to retrieve the geometry use `FamilySymbol.Activate()`

Units API

Due to significant changes in Units, many Unit APIs have been removed rather than marked obsolete

Units class

- Access via `Document.GetUnits()` | `Document.SetUnits()`
- **Formatting options such as Rounding, Accuracy, and DisplayUnits**

Unit Formatting and Parsing

- `UnitFormatUtils.FormatValueToString()` – convert value to a string
- `UnitFormatUtils.TryParse()` – parse string into value

Conversion between Unit Types

- `UnitUtils.Convert()` – convert a value between unit types
- `UnitUtils.ConvertFromInternalUnits()` – convert a value from Revit's internal unit type.
- `UnitUtils.ConvertToInternalUnits()` – convert a value to Revit's internal unit type.

Materials API

`Material.SetRenderAppearance()` deprecated.

- Set using the related `AppearanceAssetElement`

`AssetProperty` changes

- No longer inherits from `APIObject`
- `GetTypeName()` return values changed
- New subclasses of `AssetProperty` exposed

IFC API

IFC Export is now an External Service

- Explicit interfaces in `ExporterIFCRegistry` are obsolete
- Register multiple external IFC exporters in the same session

IFC APIs moved to a new assembly

- Most IFC API classes now in `RevitAPIIFC.dll`
- Add-Ins using migrated APIs must reference the new DLL and rebuild to work with Revit 2014
- APIs to invoke an IFC export and import have not moved
- Applications without custom implementations need not change

Revit API Assemblies

C:\Program Files\Autodesk\Revit Equinox Beta 1

12/10/2012	01:11	PM	40,178,616	RevitAPI.dll
12/10/2012	01:11	PM	991,672	RevitAPIIFC.dll
12/10/2012	01:11	PM	185,272	RevitAPIMacros.dll
12/10/2012	01:11	PM	1,815,480	RevitAPIUI.dll
12/10/2012	01:11	PM	119,224	RevitAPIUIMacros.dll

BeamSystem + Truss Creation

BeamSystem Creation

- `BeamSystem.Create` methods replace `Document.NewBeamSystem` methods

Truss Creation

- `Truss.Create()` replaces `Document.NewTruss()`

Deprecated API Cleanup

APIs deprecated in Revit 2013 have been removed

- ~160 items
 - Classes
 - Members
 - Etc.

The Wine

Revit Equinox features + API



Integration

Support for seamless functionality

UI API

External commands supported in Project Browser as Active View

- API commands and macros enabled
- `UIDocument.Selection`
 - returns Elements selected in Project Browser
- Use of `ViewType.Internal` should be replaced by:
 - `ViewType.ProjectBrowser`
 - `ViewType.SystemBrowser`

Copy | Paste API

ElementTransformUtils

- `CopyElements(Document, ICollection<ElementId>, Document, Transform)`
 - Copy within Documents or between Documents
- `CopyElements(View, ICollection<ElementId>, View, Transform)`
 - Copy View-specific elements
- `ElementTransformUtils.GetTransformFromViewToView(View, View)`
 - Obtain transform between coordinates of two views

Copy | Paste API Demo

Dockable Dialog API

UIApplication

- `RegisterDockablePane` registers a dockable WPF pane at Revit startup with configurable docking settings
- `IDockablePaneProvider`
 - Called when the user brings the pane into focus
 - Adds pane to canvas
- `DockablePaneData`
 - Set initial `DockablePaneState`

RevitDockablePane

- Show | Hide dockable panes

Dockable Dialog API Demo

View API

View Control

- `UIView`
 - Zoom to fit
 - Zoom to sheet
 - Zoom by factor
 - Close View
- `PreviewControl`
 - `PreviewControl.UIView` supplies the `UIView` handle for the preview control
 - `PreviewControl.ScrollbarVisibility` controls scrollbar presence

Design | Construction | Fabrication

Non-Rectangular Crop Regions

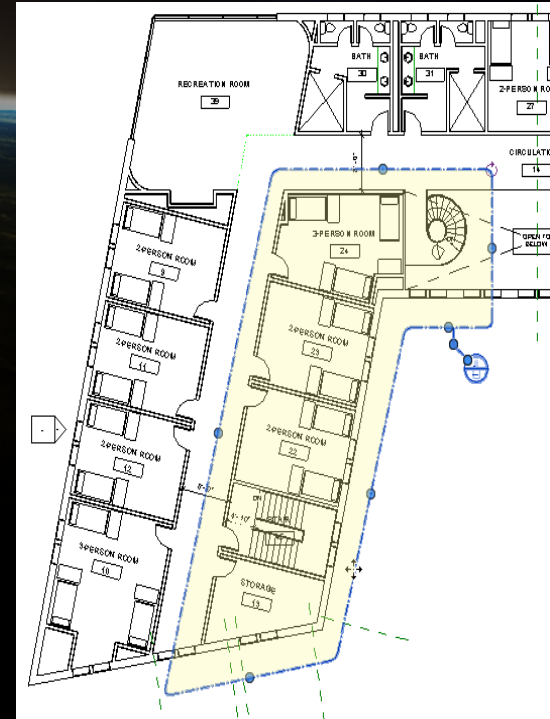
Customer Value

- Easier presentation of referenced model areas
- More efficient documentation layout
- Extension of existing UI patterns

Deliverables

- Ability to reference and crop regions of arbitrary shape*

*Arcs not supported



View API

Crop Regions

- Get | Set 3D graphic display options
 - Silhouettes | Transparency | Sunlight | Shadows | Background
- Get | Set Category overrides
- Get | Set Element overrides
- Get | Set | Add | Remove View Filters
 - Uses `ParameterFilterElement` to represent View Filters

View API

Crop Regions

- Get | Set boundaries on ViewCropRegionShapeManager
- Views and Reference Callouts supported

Viewports

- Get | Set viewport rotation
- Get | Set viewport center point

Countrification

Schedules

Customer Value

- Greater control of schedule formatting
- Increased access to data building model data

Deliverables

- Text properties and table formatting improvements
- Improved data access, display, and relationships
- Ability to display graphics in schedule headers
- Room calculation point expanded to all Elements

The screenshot displays a Revit interface with a door schedule and its details. The schedule table lists door elements with columns for Door #, To Room, From Room, Size, Height, Hardware Group, and Frame Type. Below the schedule, the 'Branch Panel: LP-3' details are shown, including a title block with project information and a table of door specifications. To the right of the details are three diagrams labeled R, D, and E, illustrating different door configurations and their connections to ductwork. The diagrams show a door frame with a handle and a lock, and a duct connection to the door frame.

Door #	To Room	From Room	Size	Height	Hardware Group	Frame Type
101A	101	101	1830	2134	3	
101B	101	101	1830	2134	3	
101C	101	101	1830	2134	3	
101D	101	101	1830	2134	3	
103	103	107	915	2134	(none)	
104A	104	107	915	2134	(none)	
104B	104	107	915	2134	(none)	
105A	105	107	915	2134	(none)	
105B	105	107	915	2134	(none)	
106A	106	107	915	2134	(none)	
106B	106	107	915	2134	(none)	
108A	108	107	915	2134	(none)	
108B	108	107	915	2134	(none)	
109	109	107	915	2134	(none)	
110	110	107	915	2134	(none)	
111	111	102	915	2134	(none)	
112	112	102	915	2134	(none)	
114	114	102	915	2134	(none)	
115	115	102	1830	2134	(none)	
116	116	102	915	2134	(none)	
117	117	102	915	2134	(none)	
118	118	102	1730	2134	(none)	
119A	119	119	1730	2134	(none)	
119B	119	118	1730	2134	(none)	
120	120	118	1830	2134	(none)	
121	121	121	1830	2134	(none)	
122	122	122	915	2134	(none)	
123	123	123	1830	2134	(none)	

Schedule API

- ViewSchedule is now a child of TableView with PanelSchedule
- Previously existing ViewSchedule APIs are still valid
- TableView API controls:
 - Header and Body table sections
 - Table Formatting
 - Cell contents
- Additional APIs control:
 - Header grouping
 - Cell formatting

Schedule API Demo

HTML Export
Formatting

Extensible Storage API

ExtensibleStorage changes

- `Element.GetEntitySchemaGuids()`
 - Access Schema GUIDs of any entities present on an Element
- `Schema.GetField()` | `Schema.ListFields()`
 - Access restricted by Schema read permission setting

ExtensibleStorageFilter

- `ElementQuickFilter` finds Elements with a given Schema GUID

Command API

Command event

- `AddInCommand.BeforeExecuted`
 - Read-Only event identifying a command before execution

Command posting

- `UIApplication.PostCommand()`
 - Execute a command when control returns from API
 - Post non-contextual commands listed in `Autodesk.Revit.UI.PostableCommand`
 - Post external commands created by any add-in
 - Only one command may be posted to Revit at a time

Add-In API

Automatic load of add-ins into active session

- Revit automatically loads add-ins from newly added .addin manifest files without restarting Revit
- ApplicationInitialized will not be called
- UIApplication.LoadAddIn()
 - Programmatically load add-ins listed in the add-in manifest

Decline active session load

- Use <AllowLoadIntoExistingSession> tag in .addin file
- Some operations can't be completed mid-session
 - Declaring failure handlers
 - Some External Service handling



Annotation + Visualization

Enhanced model documentation

Materials API

`Material.AppearanceAssetId`

- Assigns an `AppearanceAsset` to a `Material`

`AppearanceAssetElement`

- `Create()`
 - Creates an Asset Element for a rendering Asset and name
- `GetAppearanceAssetElementByName()`
 - Retrieves an Asset Element handle by the Asset name
- `SetRenderingAsset()`
 - Applies a rendering Asset to an Element

Multi-Reference Annotation API

- New tags referencing multiple identical Elements
- `MultiReferenceAnnotation` - instance
- `MultiReferenceAnnotationType` - type
- `IndependentTag.MultiReferenceAnnotationId`

Dimension Units API

Dimension Alternate Units

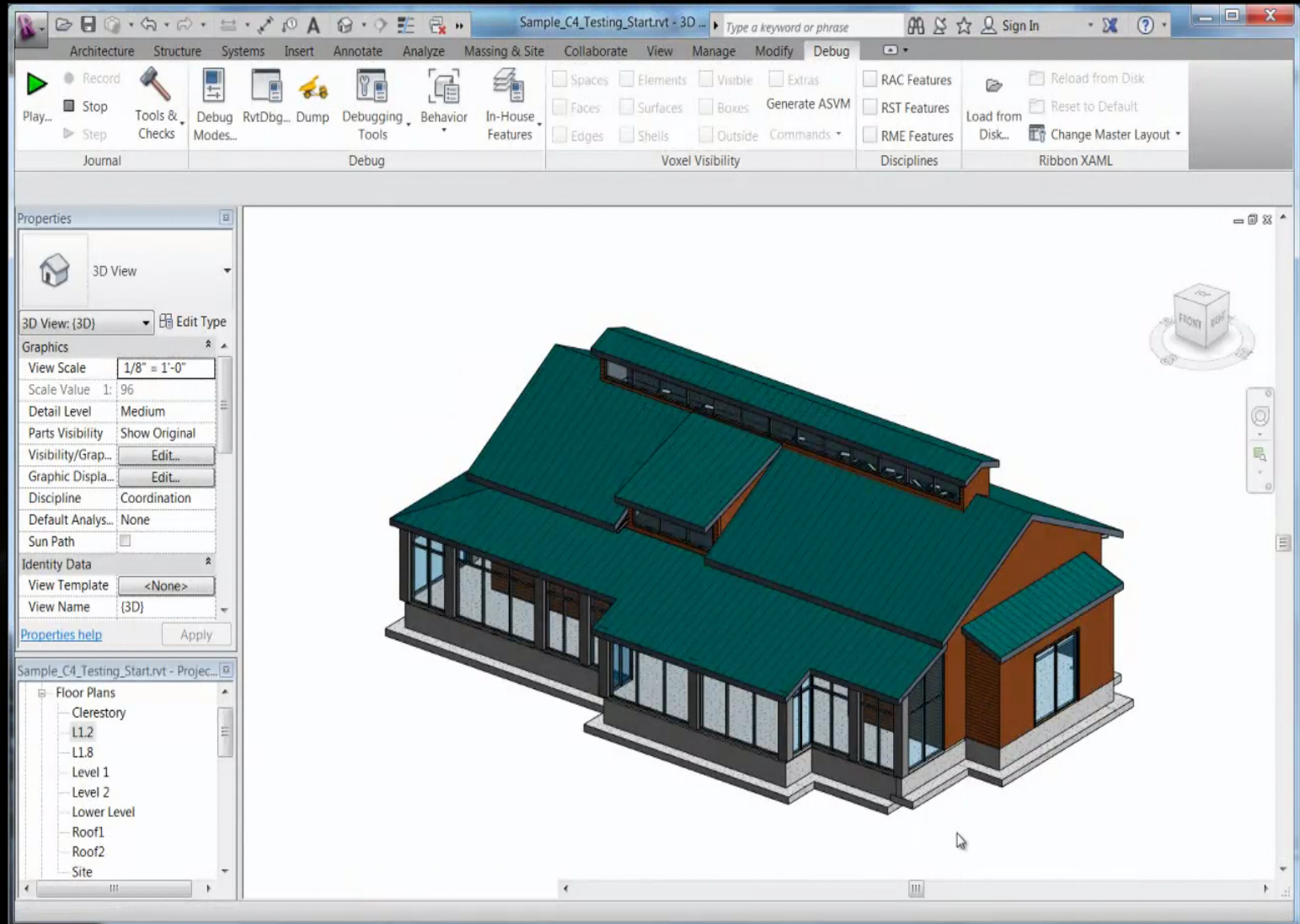
- `DimensionStyle.AlternateUnits`
- `DimensionStyle.Get | SetAlternateUnitsFormatOptions()`
- `DimensionStyle.AlternateUnitsPrefix | Suffix`

Dimension unit type

- `DimensionStyle.UnitType Get | Set`
for applicable unit types (Length | Angle | Slope)

Design | Construction | Fabrication

Displaced Views



Displaced Elements API

DisplacementElement

- Cause Elements to appear displaced from their actual locations

DisplacementPath

- Creates an annotation line connecting a displaced Element to its original location

Displaced Elements API Demo



Modeling Representation + Interoperability

SketchPlane API

`SketchPlane.Create(ElementId)`

- Creates a `SketchPlane` from a Grid, Reference Plane, or Level

`SketchPlane.GetPlane()`

- Returns the corresponding Plane.

`SketchPlane.GetPlaneReference()`

- Returns a reference to this element as a plane.

SketchPlane API

SketchPlane Creation

- `SketchPlane.Create(Document, Plane)` replaces `ItemFactoryBase.NewSketchPlane(Plane)`
- `SketchPlane.Create(Document, Reference)` replaces `ItemFactoryBase.NewSketchPlane(Reference)`
- `ItemFactoryBase.NewSketchPlane(PlanarFace)` is deprecated
- `SketchPlane.Create(Document, ElementID)` from a datum, Level, ref plane or Grid

Solid | Curve API

`Solid.IntersectWithCurve()`

- Calculates intersection between a closed Solid and a curve
- Includes option to return details of segments inside or outside the volume
- Curve segments and segment parameters available

Design | Construction | Fabrication

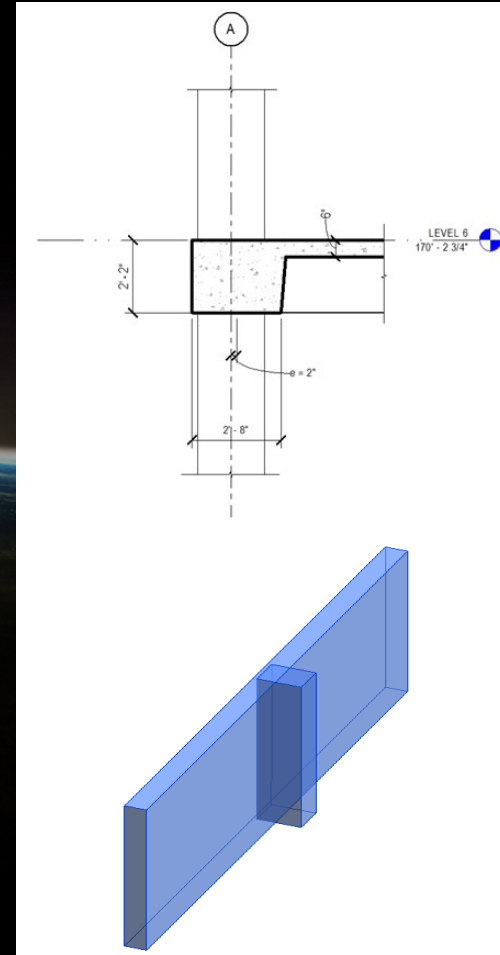
Concrete Modeling

Customer Value

- Accurately represent concrete design
- Indicate concrete construction sequencing

Deliverables

- Primacy of concrete joining
- Multi-select editing of joining primacy



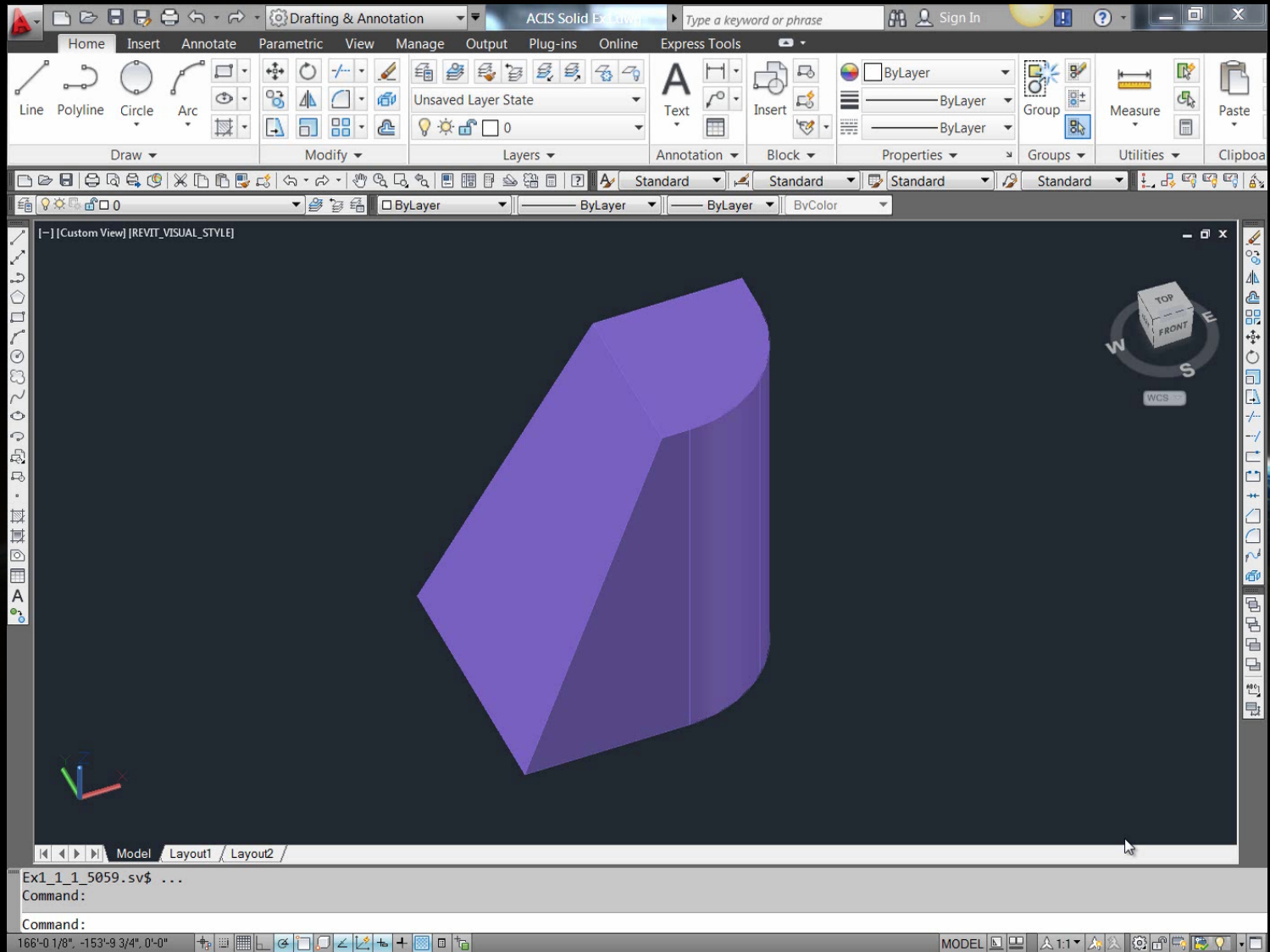
Join Geometry API

JoinGeometryUtils

- JoinGeometry
 - Creates a boolean union of two elements
- UnJoinGeometry
 - Removes a boolean join
- SwitchJoinOrder
 - Changes which Element is cut in a join
- DoesElementCutTarget
- AreElementsJoined
- GetJoinedElements

Suites + Interoperability

Imported Forms



FreeForm Element API

FreeFormElement

- A form sub-type containing non-parametric geometry
- Created from an input solid outline
- Can be added to Families
- Participates in joins and void cuts with other Elements
- Planer faces offset along the face normal vector

FreeFormElement.Create()

- Creates a new FreeFormElement

FreeFormElement.SetFaceOffset()

- Offsets planer faces a specified distance along the face normal

FreeForm Elements API Demo

Site API

SiteSubRegion

- Special class obtained from related TopographySurface
- Create | read | write subregion boundary

BuildingPad Element

- Create | read | write BuildingPad boundary

Site API

Editing TopographySurface

- Editing points in a TopographySurface now requires edit scope
- TopographyEditScope
 - Allows application to create and maintain editing session
- TopographySurface.AddPoints()
 - Now operates with the edit scope
 - No longer regenerates the document

Site API

Reading points TopographySurface

- TopographySurface methods:
ReadPoints, FindPoints, ContainsPoint,
GetBoundaryPoints, IsBoundaryPoint

Site API Demo

Beam + Brace Ends

Continued access through parameters API

- New built-in parameters replace Start | End Extension parameters
- Start | End Extension when not joined
- Start | End Join Cutback when joined

Justification Parameters (Beam Only)

- Justification can be set for whole member or ends individually
- Existing Justification parameters replaced
 - New parameters for Y + Z Start | End Justification and Offsets
- Justification enumerated by Y | Z Justification enums

Design | Construction | Fabrication

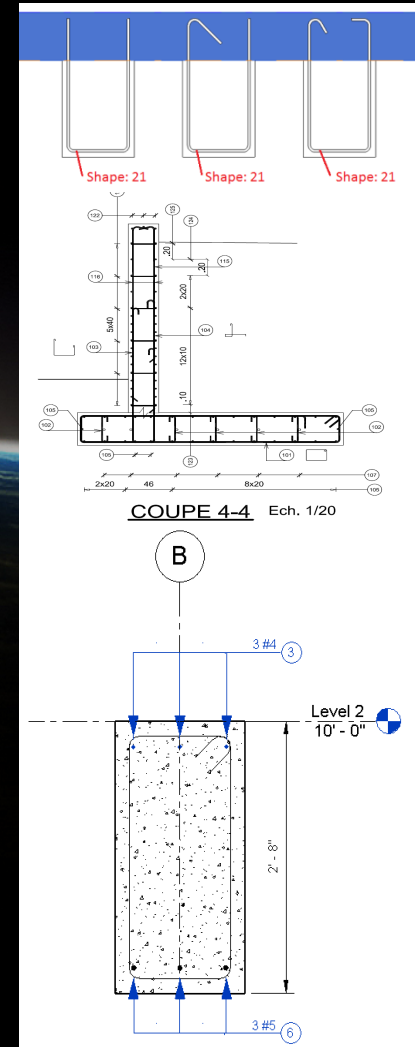
Reinforcement

Customer Value

- Improved rebar placement accuracy
- Bending tables and Schedules
- Shop drawings

Deliverables

- Rebar placement constraints customization
- Rebar shapes definition according to European standards
- Improved rebar tagging
- Reinforcement length tolerances
- Improved modeling and display of Welded Wire Mesh



Reinforcement

AreaReinforcement

- Create | Set un-obscured in View

PathReinforcement

- Create | Adjust length offset | Set un-obscured in View

FabricArea

- Create and manage reinforcement rounding settings

FabricSheet

- Create
- Get coordinate system
- Place in host
- Adjust cover offset

Reinforcement

Rebar

- Set hook orientation
- Compute driving curves
- Get constraints manager

Reinforcement | Fabric | RoundingManagers

- Access to reinforcement rounding overrides

MEP API

ElectricalLoadClassificationData

- Several new string properties corresponding to the electrical panel schedule load classification section

CSV Fitting Parameter Removal

- CSV files no longer driving MEP fitting parameters
- New APIs to manage fitting parameters

Fitting Angle Usage Settings API

- Duct | Pipe | Electrical

Duct Settings

- Size prefixes and suffixes + annotations
- Air density + viscosity

MEP API

Element Creation

- Available as static methods on respective classes
 - Replacement Duct
 - Pipe
 - FlexDuct
 - FlexPipe

MEP API

ConnectorElement

- Create connector Elements in families centered on faces
- Get Connector System type
- Get | Set Connector direction

MEP Calculations API

External Calculations

- New External Services supported for pipe and duct calculations
 - IPipePlumbingFixtureFlowServer
 - IPipePressureDropServer
 - IDuctPressureDropServer
 - IDuctFittingAndAccessoryPressureDropServer
 - IPipeFittingAndAccessoryPressureDropServer
 - IDuctFittingAndAccessoryPressureDropUIServer
 - IPipeFittingAndAccessoryPressureDropUIServer



Model Management

Ensuring coherent data interaction

Document Open API

OpenOptions

- **Audit**
 - Specifies whether to expand all Elements for corruption check
 - Defaults to false
- **AllowOpeningLocalByWrongUser**
 - Specifies whether a local file can be opened in ReadOnly mode by a user other than the owner
- **Get | Set OpenWorksetsConfiguration()**
 - Specify Worksets opened | closed when opening a document

Document Save API

`Document.Save | SaveAs (ModelPath, SaveAsOptions)`

- `SaveOptions.` | `SaveAs.Options.Compact`
 - Specifies if the OS should remove redundant data from the RVT
- `SaveAsOptions.MaximumBackups`
 - Specifies the maximum number of backups to keep on disk
- `SaveAsOptions.WorksharingOptions` for workshared RVTs:
 - `WorksharingSaveAsOptions.SaveAsCentral`
 - `WorksharingSaveAsOptions.OpenWorksetsDefault`
 - `WorksharingSaveAsOptions.ClearTransmitted`

Worksharing API

Reload Latest

- `Document.ReloadLatest()`
- `Document.HasAllChangesFromCentral()`

Synchronize with Central

- `Document.SynchronizeWithCentral()`

Element ownership

- `WorksharingUtils.CheckoutElements()`
- `WorksharingUtils.CheckoutWorksets()`
- `WorksharingUtils.RelinquishOwnership()`

Worksharing API

Create new local

- `WorksharingUtil.CreateNewLocal()`

Enable Worksharing

- `Document.EnableWorksharing()`

Linked Model API

Link Load | Unload

- `RevitLinkType.Load()`
- `RevitLinkType.LoadFrom()`
- `RevitLinkType.Unload()`

Link Path Get | Set

- Relative
- Absolute
- Server (read-only)

Link Create

- Specify Worksets to open when creating new link

Linked Model API

Identify Linked Documents

- `Document.IsLinked()`
- `RevitLinkType.Unload()`

Get Linked Document

- `RevitLinkInstance.GetLinkedDocument()`

Linked Model Interaction

- Room Tag creation for Linked Rooms
- Prompt for Linked Element Selection
- Convert geometric references between link and host
- Room boundary analysis
- Face-based family instance placement

Interoperability API

Import + Link

- Import | Link SAT
- Import | Link SketchUp
- Import DWF markup

Interoperability API

Export

- NavisWorks through Add-In
- Access tables mapping to formats such as DWG, IFC and DGN
 - ExportLayerTable
 - ExportLinetypeTable
 - ExportFontTable
 - ExportPatternTable
 - ExportLineweightTable

Interoperability API

Export

- CustomExporter
 - Access to rendering output pipeline
 - Allows exporting 3D views via a custom export context
- IExportContext
 - Output describes the model as it appears in Revit
 - Includes all geometry and material properties



Macro Development

Rapid prototyping

Macros

Ruby | Python development

- Macros can be coded in Ruby or Python as well as C# and VB.NET

API for macros and macro modules

- Located in `RevitAPIMacros.dll` and `RevitAPIUIMacros.dll`
- List | Add | Delete macro modules
- List | Edit | Delete | Create | Execute macros
- Access macro security settings

Learning More

- Revit Developer Center: Revit SDK, Samples, API Help File, DevTV Introduction to Revit Programming, My First Revit Plugin
 - <http://www.autodesk.com/developrevit>
- , Developer's Guide
 - <http://wikihelp.autodesk.com/Revit> > API Developer's Guide
- Revit API Webcasts and Trainings
 - http://www.adskconsulting.com/adn/cs/api_course_sched.php > Revit API
- Discussion Group
 - <http://discussion.autodesk.com> > Revit Architecture > Revit API
- API Training Classes
 - <http://www.autodesk.com/apitraining>
- The Building Coder, Jeremy Tammik's Revit API Blog
 - <http://thebuildingcoder.typepad.com>
- ADN, The Autodesk Developer Network
 - <http://www.autodesk.com/joinadn>
- DevHelp Online for ADN members
 - <http://adn.autodesk.com>

Autodesk®