

# What's New in the Revit API

Jeremy Tammik  
Forge Platform Development



# Agenda

- Must Do
- New Functionality
- Revit Roadmap

# Disclaimer

*We may make statements regarding planned or future development efforts for our existing or new products and services. These statements are not intended to be a promise or guarantee of future availability of products, services or features but merely reflect our current plans and based on factors currently known to us. These planned and future development efforts may change without notice. Purchasing decisions should not be made based upon reliance on these statements.*

*These statements are being made as of February 2017 and we assume no obligation to update these forward-looking statements to reflect events that occur or circumstances that exist or change after the date on which they were made. If this presentation is reviewed after this date, these statements may no longer contain current or accurate information.*



# Must Do



## Must Do | Edit in Perspective Views



Autodesk Confidential Information

 AUTODESK.

## Must Do | APIs Enabled in Perspective Views

- **Modification** of many object types is now allowed in perspective views
- Most **commands** allowed in 3D views are now allowed in perspective views  
Exception: Annotation & MEP tab.
- **External API Commands** and applications are now enabled by default in perspective views.
- **Macros, Macro Manager tools, Dynamo scripts**, and the **Dynamo editor** are also enabled

## Must Do | Subelements

- Subelements are a way for parts of an element to behave like real elements without the overhead of a full element
- Subelements have typical element behaviors:
  - Create/Delete, Select, Reference, Category, Type, Bounding Box, Geometry, Unique Id, Parameters
- **Subelement** class can refer to either an Element or a specific subelement
- Example parent elements:
  - Rebar, RebarContainer, FabricSheet
  - Stairs in MultistoryStairs, Railings, Continuous Rails

## Must Do | Application Version

	<b>Deprecated Property</b>	<b>New Property</b>
<b>Application</b>	Application.IsSubscriptionUpdate	Application. <a href="#">SubVersionNumber</a>
<b>RevitAddinUtility</b>	RevitProduct.isSubscriptionUpdate	RevitProduct. <a href="#">ReleaseSubVersion</a>

Return a string representing the major-minor version number of the application  
Format MajorVersion.MinorVersion.Update, for example, "2018.0.0"

## Must Do | Visual Assets

- **Asset\*** classes (e.g. AssetProperty) moved from Autodesk.Revit.Utility to **Autodesk.Revit.DB.Visual**
- **AssetPropertyType** enum values renamed (integer values are unchanged)

Old Values	Replacement	Integer Value
APT_Undefined	Unknown	0
APT_Properties	Properties	1
APT_Boolean	Boolean	2
APT_EngineeringUnits	Enumeration	3
APT_Integer	Integer	4
APT_Float	Float	5
APT_Double	Double1	6
...	...	...

## Must Do | Direct Shapes

- New **DirectShape** behaviors
  - **Tagging**
    - DirectShapes can be tagged with Revit tag tools
  - **Dimensions to Edges**
    - Referenceable DirectShapes now support dimensioning to edge references
  - **Host Connectors**
    - Referenceable DirectShapes can host connector elements in families
  - **Host Rebar**
    - Direct shapes of some categories can act as a rebar host

## Must Do | Miscellaneous Changes

- Dynamic Updaters on Reload Latest
  - Dynamic updaters are now triggered on Reload Latest for the elements added or changed in the central file.
- Export to DWG/DXF API
  - ACADVersion::R2018 added and is the default option
- UIDocument.PromptForFamilyInstancePlacement()
  - Now works like PickObject() methods - the placement operation will be cancelled when the "x" button of Revit is clicked

## **Must Do | Obsolete API Removal**

- All APIs marked as deprecated in Revit 2017 have been removed.



# New Functionality



## New | Reviewable Warnings

- Document.GetWarnings()
  - List of failure messages generated from persistent (reviewable) warnings in the document

## New | Family Instance References

- `FamilyInstance.GetReference*()`
  - Enable access to `FamilyInstance` references that correspond to reference planes and lines in the family
  - Find references by name or by type
    - `FamilyInstanceReferenceType` – enum describing reference types available in “Is Reference” and “Reference” parameters (e.g. Strong, Weak, Left, etc)

## New | Display External 3D Graphics

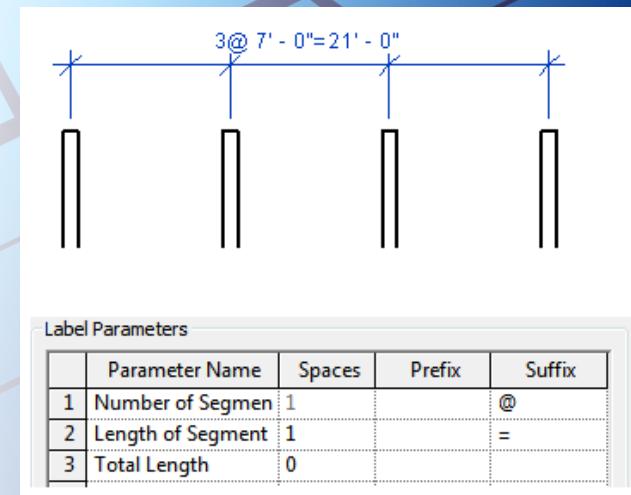
- Use [DirectContext3D](#) API to display geometry in Revit by pushing directly to Revit graphics
- Revit's rendering pipeline asks registered servers to provide the geometry for rendering
  - [DirectContext3D.IDirectContext3DServer](#) - The interface to be implemented
  - [DirectContext3D.DrawContext](#) - provides drawing functionality for use by DirectContext3D servers
  - [DirectContext3D.VertexBuffer](#) - stores vertex data for rendering

## New | Display External 3D Graphics

- **DirectContext3DDocumentUtils**
  - Support persistence and user manipulation of streamed graphics
- **DirectContext3DHandleSettings**
  - Access to Visibility/Graphics override settings applied to DirectContext3D handles
- **DirectContext3DHandleOverrides**
  - Access to DirectContext3DHandleSettings stored by a view

## New | Dimensions

- **DimensionEqualityLabelFormatting**
  - Set properties of dimension equality formulas for continuous linear or angular dimensions
  - Access via DimensionType.[Get/SetEqualityFormula\(\)](#)
- **DimensionType.Get/SetUnitsFormatOptions()**
  - Access the Unit Format for a dimension style
- **OrdinateDimensionSetting**
  - Customize ordinate dimension's text position, orientation, line style, and visibility
  - Access via DimensionType.[Get/SetOrdinateDimensionSetting\(\)](#)



## New | Tags

[SpatialElementTag](#) - base class for Room, Area, and Space tags

- [HasElbow](#) - Identifies if the tag's leader has an elbow point or not.
- [TagText](#) - text displayed by the tag

[IndependentTag](#)

- [Create\(\)](#)
  - replaces Revit.Creation.Document.NewTag()
  - supports elements and subelements
- [GetTaggedReference\(\)](#)
  - reference to the item which has been tagged
- [IsTaggedOnSubelement\(\)](#)
  - Identifies if tag is referencing a subelement
- [HasElbow](#)
  - Indicates if the leader on the tag has an elbow point

## New | Geometry API

- **Face.GetSurface()**
  - returns a copy of face's surface
- **RevolvedSurface.GetProfileCurveInWorldCoordinates()**
  - returns copy of profile curve expressed in the world coordinate system
- **RuledSurface.HasFirstProfilePoint()**
- **RuledSurface.HasSecondProfilePoint()**
  - check if a point was used to define one of the surface profiles

## New | Parts

Access and manipulate the offset applied to a given face of a Part element

- Part.[ResetFaceOffset\(\)](#)
- Part.[GetFaceOffset\(\)](#)

## New | Shared Coordinates

- SiteLocation.[GeoCoordinateSystemId](#)
  - Read-only geographic coordinate system can be acquired from DWG file
  - Returns geographic coordinate system ID, e.g. "Beijing1954/a.GK3d-40"
- Document.[Acquire/PublishCoordinates\(\)](#)
  - Acquire project coordinates from RVT or DWG link or publish to specified ProjectLocation
- ProjectLocation.[Create\(\)](#)
  - Creates new project location from specified SiteLocation

## New | Links

- IExternalResourceServer now supports **CAD**, **DWF**, and **IFC** links
- CADLinkType now supports **Reload()** and **LoadFrom()** operation
- RevitLinkType.**UpdateFromIFC()**
  - Allows specification of IFC file by name
- RevitLinkInstance.**Create(ImportPlacement)**
  - Create new instance of a Revit link according to placement type
- ImportInstance.**Create(\*)**
  - support creation of DWG or DXF instances

## New | DWG Export

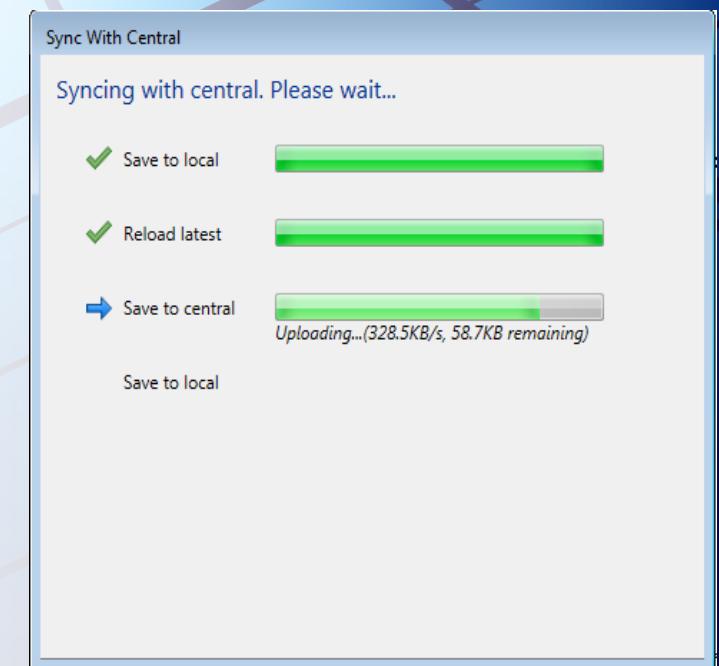
- ExportColorMode.[TrueColorPerView](#)
  - Colors from the Revit project will be exported as 24-bit RGB values as specified in view
- ACADExportOptions.[UseHatchBackgroundColor](#)
- ACADExportOptions.[HatchBackgroundColor](#)
  - Define color that will be set as hatch background color on the exported hatch
- ExportDWGSettings.[FindByName\(\)](#)
  - Returns the pre-defined DWG export settings
- ExportDWGSettings.[GetActivePredefinedSettings\(\)](#)
  - Returns the active DWG export settings

## New | Miscellaneous

- Level.[FindAssociatedPlanViewId\(\)](#)
  - returns first-found plan view associated with specified level
- ShapeImporter.[Get/SetDefaultLengthUnit\(\)](#)
  - length unit to be used during import if not specified in SAT file

## New | C4R Worksharing Events

- Events. `WorksharedOperationProgressChanged` notifies of Collaboration for Revit's synch progress
  - Event consists of several phases with event args:
    - `DocumentSaveToLocalProgessChangedEventArgs`
    - `DocumentReloadLatestProgessChangedEventArgs`
    - `DocumentSaveToCentralProgessChangedEventArgs`



## New | Dockable Frames

- Custom Dockable Panes now support display of dynamic UI elements (e.g. web browser controls)
- [IFrameworkElementCreator](#)
  - New interface to support dynamic content
  - [CreateFrameworkElement\(\)](#)
    - constructs and returns the WPF Framework element embedded in dockable pane
- [DockablePaneDataProvider.GetFrameworkElement\(\)](#)
- [DockablePaneDataProvider.FrameworkElementCreator](#)
  - provide ability for application to delivery a framework element ot a dockable pane

## New | Multi-Story Stairs

- **MultistoryStairs**

- create stairs that span multiple levels
- [Add/RemoveStairsByLevelIds\(\)](#) - Adds or remove stairs to the given levels.
- [Unpin\(\)](#) – Enables custom modification of one story of stairs

- **Stairs.MultistoryStairsId**

- indicates the id of the associated MultistoryStairs element

- **StairsPath.CreateOnMultistoryStairs()**

- support creation of new stairs paths for stairs in a multistory stairs element

## New | Railings

- Railing
  - Now support hosting railings on multistory stairs
  - [Get/SetMultistoryStairsPlacementLevels\(\)](#)  
[GetSubelementOnLevel\(\)](#)
    - provide access to Railings hosted on multistory stairs
  - [Create\(..., multistoryStairsId, levelId, ...\)](#)
    - Places railing on a given level of given multistory stair
  - [Create\(..., stairsId,...\)](#)
    - Now supports multistory stair as input
  - HostId
    - Now supports stairs or stairs components from multistory stairs

## New | HVAC

- **HVACLoadType**
  - new base class for building type and space type
  - New **properties** to control air change, area per person, latent heat, lighting load, power, and heat gain settings
- Subclasses:
  - **HVACLoadSpaceType**
    - the type element of space
    - Access via MassZone.**SpaceTypeId** and Space.**SpaceTypeId**
  - **HVACLoadBuildingType**
    - the type element to access building type properties

## New | MEP Fabrication Parts

- `FabricationPart.SplitStraight()`
  - Splits the fabrication part into two at specified point
- Detailed Fabrication additions
  - Several methods, properties classes and enumerations have been added to allow the user to access detailed fabrication information
- Part status
  - Properties were added to fabrication part to allow the user to query and set the part fabrication status field
- Hanger rod additions
  - Better control over hanger rod thicknesses

## New | Electrical

- `ElectricalSystem.Create()` methods
  - replace obsoleted APIs
- `PanelScheduleView.AddSpace()/AddSpare()`
  - add a space or spare at specific cell
- enum `ElectricalCircuitPathMode`
  - An enumerated type indicates the circuit path mode
- `ElectricalSystem` new `properties`
  - Control the mode, offset, and path of electrical circuit path

## New | MEP Analytical Connections

- **MEPAnalyticalConnection**
  - an analytical element that connects mechanical equipment to a piping network
  - `GetFlow()` - returns the up-to-date flow value
- **MEPAnalyticalConnectionType**
  - Type element of an `MEPAnalyticalConnection`
  - The type's `PressureLoss` value is included in the network critical path calculation

## New | Rebar

- Rebar supports shape-driven and **free-form** layouts
  - `Rebar.IsRebarFreeForm/IsRebarShapeDriven`
- Layout specific functionality in accessors:
  - `Rebar.GetShapeDrivenAccessor()`
    - Replaces shape-driven only Rebar class members
  - `Rebar.GetShapeDrivenAccessor()/GetFreeFormAccessor()`
    - Return shape-driven & free-form accessors respectively
- **IRebarUpdateServer**
  - Interface used to drive the generation and update of free-form geometry

## New | Structural Steel Sections

- StructuralSection & derived classes
  - have new [properties](#) and [input parameters](#)
- StructuralSectionUtils.[GetStructuralElementDefinitionData](#)
  - defines the section and the position of the structural element.
- StructuralSection.[GetStructuralSectionGeneralShape](#)
  - general shape provides information about the geometry
- StructuralSectionAnalysisParams
  - defines common set of parameters for structural analysis.
- StructuralSectionGeneral\* (e.g. StructuralSectionGeneralC)
  - define parameter sets for specific shapes

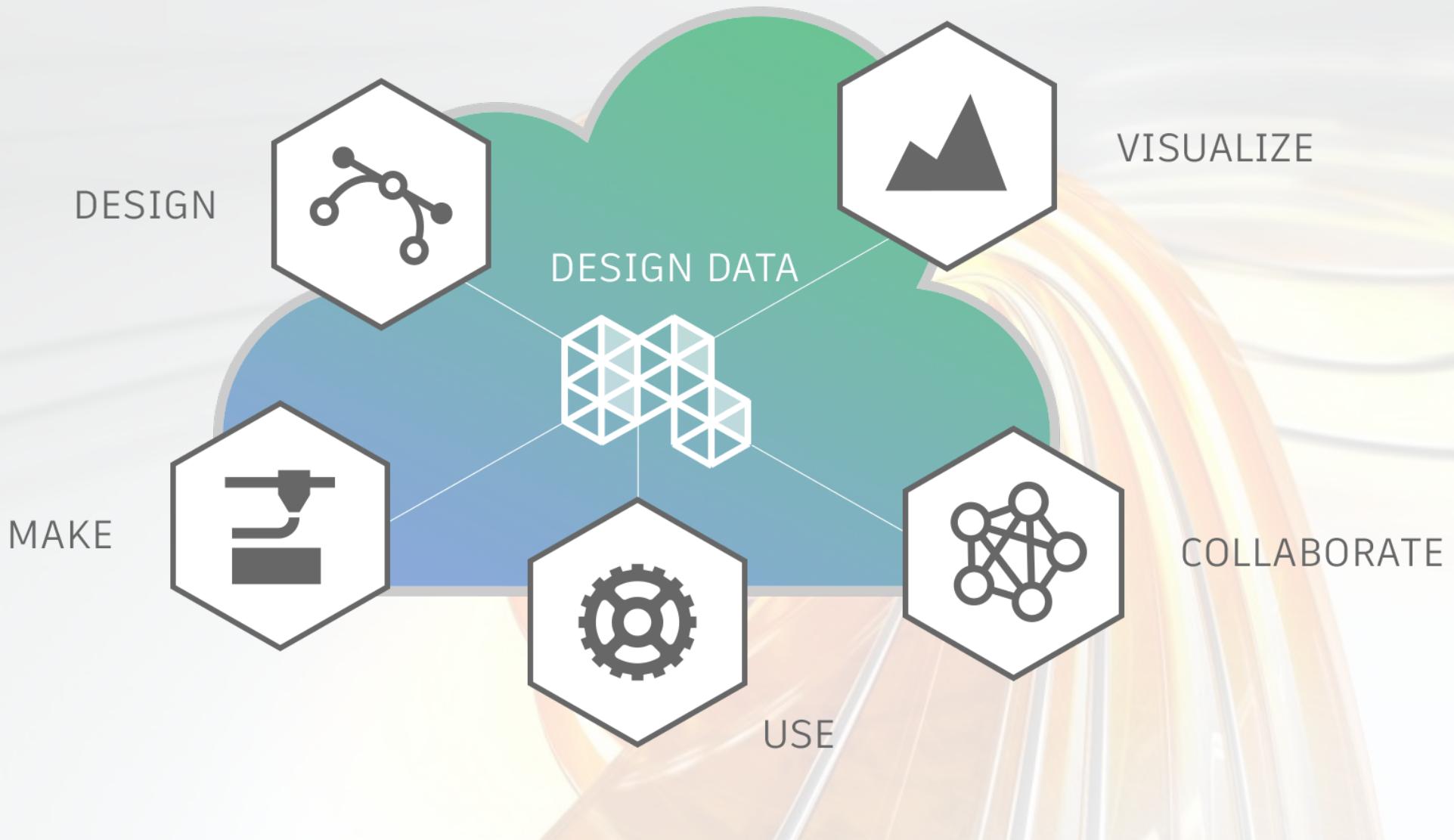
# Revit Roadmap





# PUBLIC REVIT ROADMAP

[www.autodesk.com/revitroadmap](http://www.autodesk.com/revitroadmap)





# REVIT IDEAS

[www.autodesk.com/revitideas](http://www.autodesk.com/revitideas)



Autodesk Confidential Information

 AUTODESK



## REVIT PREVIEW

- Monthly online builds and updates
- Quarterly install builds
- Access
  - ADN Beta Program at [beta.autodesk.com](http://beta.autodesk.com)
  - Invitation link on product download page at ADN extranet ([adn.autodesk.com](http://adn.autodesk.com))

# Q & A





Autodesk is a registered trademark of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that August appear in this document.

© 2015 Autodesk, Inc. All rights reserved.