

Major changes and renovations to the Revit API

API CHANGES	3
CEFSHARP REMOVED	3
OPTION FOR ADD-IN DEPENDENCY ISOLATION	3
PARAMETER API CHANGES	4
REBAR HOOKS AND END STATE PARAMETERS	4
LENGTH PARAMETER FOR STRUCTURAL BEAMS AND COLUMNS	5
REFERENCE LABEL PARAMETERS	6
GEOMETRY API CHANGES	6
CURVE INTERSECTION API	6
VIEWPORT API CHANGES	6
BROWSER ORGANIZATION API CHANGES	7
COMPOUNDSTRUCTURE API CHANGES	7
CORE LAYERS NO LONGER REQUIRED	7
ENERGY ANALYSIS API CHANGES	7
FAILURES API CHANGES	8
EVENTS API CHANGE	8
STAIRS API CHANGE	8
STRUCTURE API CHANGES	8
REBAR CRANKING API ADDITIONS AND CHANGES	8
SHAPE-DRIVEN REBAR	14
FREE FORM REBAR	14
STRUCTURAL CONNECTIONS	15
ELECTRICAL API CHANGES	15
NEW CLASSES REPRESENT TYPES AND MATERIALS	15
ELECTRICALSYSTEM API ADDITIONS	15
WIRETYPE MEMBER API CHANGES	16
ADDITIONAL ELECTRICAL API CHANGES	16
ZONE API CHANGES	17
IFC API CHANGES	18
IMPORTER OPTIONS	18
CATEGORY MAPPING	19
OBSOLETE API REMOVAL	19
CONSTRUCTORS	19
PROPERTIES	19
METHODS	20
ENUMS	20
API ADDITIONS	20
ADD-IN MANAGEMENT API ADDITIONS	20
DOCUMENT API ADDITION	21
GEOMETRY API ADDITIONS	21
SOLID UTILITIES	21
IMPORT/EXPORT DATA EXCHANGE API ADDITIONS	21
NURBSSURFACEDATA	21
WALL & COMPOUND STRUCTURE API ADDITIONS	22

WALL ATTACHMENTS	22
COMPOUNDSTRUCTURE PRIORITY	22
TOPOSOLID API ADDITIONS	22
SUBDIVISION CREATION	22
STABILITY FOR CUTTING AND VOID OPERATIONS	22
CEILING API ADDITIONS	23
ROOM, AREA AND SPACE API ADDITIONS	23
PARAMETER API ADDITIONS	23
VIEW API ADDITIONS	23
GRAPHICS API ADDITIONS	23
TEMPORARY GRAPHICS	23
DIRECTLY DRAWN GRAPHICS (DIRECTCONTEXT3D)	24
INDEPENDENT TAG API ADDITION	24
FAILURE API ADDITION	24
IFC API ADDITIONS	24
EXPORTERIFC UTILITIES	24
IFC PARAMETER MAPPING	24
STRUCTURAL ANALYTICAL API ADDITIONS	25
CONNECTORS IN STRUCTURAL ANALYTICAL ELEMENTS	25
POINTS IN STRUCTURAL ANALYTICAL NETWORKS	25
STRUCTURAL API ADDITIONS	25
STRUCTURAL CONNECTIONS	25
LOAD COMBINATIONS	25
REINFORCEMENT API ADDITIONS	26
FREE FORM REBAR CREATION	26
REBAR NUMBERING	26
REBAR CONSTRAINT RELATIONSHIP	26
FABRIC SHEET AND AREA	26
MEP ANALYSIS API ADDITIONS	27
MEP DUCT/PIPE PRESSURE LOSS CALCULATIONS	27
ELECTRICAL API ADDITIONS	27
BROWSER ORGANIZATION FOR PANEL SCHEDULES	27
MEP FABRICATION API ADDITIONS	27
ENERGY ANALYSIS API ADDITIONS	28

API changes

CefSharp removed

All Revit dependencies on CefSharp have been modified to no longer have that dependency. Therefore, all the binaries of CefSharp have been removed from Revit's install. Note that there still may be dependency conflicts for different add-ins that depend on and load CefSharp for their own purposes, even without a Revit specified version.

Option for Add-in Dependency Isolation

A new option has been introduced for add-in developers to be able to run add-ins in a separate Assembly load context from Revit and from other add-ins.

- This will isolate the add-in from dependencies and dependent assemblies loaded by Revit or by other add-ins in the same session.
- This should reduce or eliminate assembly version conflicts seen in Revit sessions due to conflicting add-ins.

Adoption of this feature is optional for this release. Developers can set the option "UseRevitContext" to false per add-in manifest (it applies to all registered add-ins in the given manifest). The default for add-ins where this is not specified is "true".

If the option is used, all add-ins loaded from the same folder will be placed into the same external context. Add-ins from different folder installation locations will go to different contexts.

Developers can also optionally set a custom ContextName for the add-in. If this is used, add-ins from different folder locations that use the same context name will be loaded into the same context.

Add-in developers are advised to check the dependencies that they load are correct and complete for their own add-in as they may no longer be able to rely on Revit or another add-in loading the dependency prior to the add-in being loaded.

Sample use of the new manifest flags:

```
<?xml version="1.0" encoding="utf-8"?>
<RevitAddIns>
    <AddIn Type="DBApplication">
        <Name>SampleApplication</Name>
        <FullClassName>SampleApplication.Application</FullClassName>
        <Assembly>SampleApplication.dll</Assembly>
        <ClientId>C96B32A3-98C6-4B47-99DA-562E64689C6F</ClientId>
        <VendorId>Autodesk</VendorId>
    </AddIn>
    <ManifestSettings>
        <UseRevitContext>False</UseRevitContext>
        <ContextName>SampleContextName</ContextName>
    </ManifestSettings>
</RevitAddIns>
```

In the RevitAddInUtility API, the new properties:

- RevitAddInManifestSettings.UseRevitContext
- RevitAddInManifestSettings.ContextName

provide read and write access to these settings, while the new property:

- RevitAddInManifest.ManifestSettings

provides access to the overall settings for a manifest.

Parameter API changes

Classification and Assembly Codes

The BuiltInParameter values related to classification and assembly codes have been renamed in the API to match changes made to the user interface.

Original BuiltInParameter Entry	Original Parameter Name	Original ParameterTypeId	New BuiltInParameter Entry	New Parameter Name	New ParameterTypeId
OMNICLASS_CODE	OmniClass Number	ParameterTypeId.OMNICLASS_CODE	CLASSIFICATION_CODE	Classification Number	ParameterTypeId.ClassificationCode
OMNICLASS_DESCRIPTION	OmniClass Title	ParameterTypeId.OMNICLASS_DESCRIPTION	CLASSIFICATION_DESCRIPTION	Classification Title	ParameterTypeId.ClassificationDescription
UNIFORMAT_CODE	Assembly Code	ParameterTypeId.UNIFORMAT_CODE	ASSEMBLY_CODE	Assembly Code	ParameterTypeId.AssemblyCode
UNIFORMAT_DESCRIPTION	Assembly Description	ParameterTypeId.UNIFORMAT_DESCRIPTION	ASSEMBLY_DESCRIPTION	Assembly Description	ParameterTypeId.AssemblyDescription

Rebar hooks and end state parameters

The BuiltInParameter values related to rebar hook end states were renamed with the introduction of rebar cranking options.

Original BuiltInParameter Entry	Original Parameter Name	Original ParameterTypeId	New BuiltInParameter Entry	New Parameter Name	New ParameterTypeId
REBAR_SHAPE_HOOK_ROTATION_AT_START	Hook Rotation At Start	ParameterTypeId.RebarShapeHookRotationAtStart	REBAR_SHAPE_TERMINATION_ROTATION_AT_START	Rotation At Start	ParameterTypeId.RebarShapeTerminationRotationAtStart
REBAR_SHAPE_HOOK_ROTATION_AT_END	Hook Rotation At End	ParameterTypeId.RebarShapeHookRotationAtEnd	REBAR_SHAPE_TERMINATION_ROTATION_AT_END	Rotation At End	ParameterTypeId.RebarShapeTerminationRotationAtEnd
REBAR_HOOK_ROTATION_AT_START	Hook Rotation At Start	ParameterTypeId.RebarHookRot	REBAR_TERMINATION_ROTATION_AT_START	Rotation At Start	ParameterTypeId.RebarTerminationRotationAtStart

		ationAtStar t			
REBAR_HOOK_ROTATION_AT_END	Hook Rotation At End	Parameter Typeld.RebarHookRotationAtEnd	REBAR_TERMINATION_ROTATION_AT_END	Rotation At End	ParameterTypeld.RebarTerminationRotationAtEnd
REBAR_HOOK_ROTATION_AT_START_SCHEDULES_TAGS_FILTERS	Hook Rotation At Start	Parameter Typeld.RebarHookRotationAtStartSchedulesTagsFilters	REBAR_TERMINATION_ROTATION_AT_START_SCHEDULES_TAGS_FILTERS	Rotation At Start	ParameterTypeld.RebarTerminationRotationAtStartSchedulesTagsFilters
REBAR_HOOK_ROTATION_AT_END_SCHEDULES_TAGS_FILTERS	Hook Rotation At End	Parameter Typeld.RebarHookRotationAtEndSchedulesTagsFilters	REBAR_TERMINATION_ROTATION_AT_END_SCHEDULES_TAGS_FILTERS	Rotation At End	ParameterTypeld.RebarTerminationRotationAtEndSchedulesTagsFilters
REBAR_FREE_FORM_HOOK_START_PLANE_ANGLE	Hook Orientation At Start	Parameter Typeld.RebarFreeFormHookStartPlaneAngle	No specific replacement; functionally replaced by REBAR_TERMINATION_ROTATION_AT_START (see above)	N/A	N/A
REBAR_FREE_FORM_HOOK_END_PLANE_ANGLE	Hook Orientation At End	Parameter Typeld.RebarFreeFormHookEndPlaneAngle	No specific replacement; functionally replaced by REBAR_TERMINATION_ROTATION_AT_END (see above)	N/A	N/A
REBAR_ELEM_HOOK_START_ORIENTATION	Hook Orientation At Start	Parameter Typeld.RebarElemHookStartOrientation	REBAR_ELEM_TERMINATION_START_ORIENTATION	Orientation At Start	ParameterTypeld.RebarElemTerminationStartOrientation
REBAR_ELEM_HOOK_END_ORIENTATION	Hook Orientation At End	Parameter Typeld.RebarElemHookEndOrientation	REBAR_ELEM_TERMINATION_END_ORIENTATION	Orientation At End	ParameterTypeld.RebarElemTerminationEndOrientation

Length parameter for structural beams and columns

The BuiltInParameter INSTANCE_LENGTH_PARAM's user visible name was changed for structural beams and columns. Scripts that looks for the parameter by name may be impacted by this change. This may also have an impact on the parameter name in other Revit languages. Code that references this parameter by its BuiltInParameter identifier or ForgeTypeld should be unaffected by this change.

Original Parameter Name	Updated Parameter Name
Length	System Length

Reference label parameters

The type parameter VIEWER_REFERENCE_LABEL_TEXT (ParameterTypeId.ViewerReferenceLabelText) now represents a default value for the reference labels of sections, elevations, and callouts. This parameter has been renamed to "Default Reference Label". The displayed value for a particular view instance is now stored in the instance parameter VIEWER_REFERENCE_LABEL (ParameterTypeId.ViewerReferenceLabel) - user visible name: "Reference Label".

Geometry API changes

Curve Intersection API

A new overload and corresponding classes have been added to in order to more accurately represent various edge cases in determining the intersection relationship between two curves. As a result of this unification, older Curve.Intersect() overloads have been deprecated:

Deprecated member(s)	Replacement member(s)
SetComparisonResult Curve.Intersect (Curve otherCurve)	CurveIntersectResult Curve.Intersect(Curve otherCurve, CurveIntersectionResultOption.Simple)
SetComparisonResult Curve.Intersect (Curve otherCurve, IntersectionResultArray intersections)	CurveIntersectResult Curve.Intersect(Curve otherCurve, CurveIntersectionResultOption.Detailed)

The method returns the new class:

- CurveIntersectResult

encapsulates the results of an intersection check between two curves. It offers the following information:

- CurveIntersectResult.Result - the classification of intersection or overlap between the two curves.
- CurveIntersectResult.GetOverlaps() - the list of points specifying individual points or ranges where the two curves overlap. The points include details about the CurveOverlapPointType, the 3D point of the overlap, plus the parameters for the point on each curve.

Note that if Curve.Intersect() was invoked with CurveIntersectResultOption.Simple, the result will not contain overlap information. This may save processing time when the only required information is the type of intersection between the curves.

Viewport API changes

The new class:

- Autodesk.Revit.DB.ViewPosition

represents a named assigned position for a viewport on a sheet. It exposes the following members:

- ViewPosition.ViewAnchor
- ViewPosition.Position
- ViewPosition.Create()
- ViewPosition.GetPlacedViewportIdsForViewPosition()

The ViewPosition can be associated to a Viewport using:

- Viewport.SetViewAnchorAndPosition()
- Viewport.GetPositionAtViewAnchor()

This replaces the previously existing ViewPositioning capabilities:

Deprecated member	Replacement member
Viewport.ViewportPositioning	Viewport.ViewAnchor

Browser Organization API changes

The following properties are deprecated and may be removed in later version of Revit.

Deprecated member(s)	Replacement member(s)	Notes
FolderItemInfo.ElementId	FolderItemInfo.GetGroupingParameterIdPath()	The parameter Id path used to determine the folder items in the browser.
BrowserOrganization.SortingParameterId	BrowserOrganization.GetSortingParameterIdPath()	The parameter Id path used to determine the sorting order of items in the browser.

CompoundStructure API changes

Core Layers no longer required

elements were required to have a core layer. Now, with this enhancement, users can create compound elements without a core layer, and APIs related to creating and modifying and deleting layers in CompoundStructures now support this new option.

Energy Analysis API changes

The following members were removed from this version of Revit. Use the already existing replacement methods if needed:

Removed member(s)	Replacement member(s)
EnergyAnalysisOpening.GetPolyloop()	EnergyAnalysisOpening.GetPolyloops()
EnergyAnalysisSurface.GetPolyloop()	EnergyAnalysisSurface.GetPolyloops()

The following members are deprecated and may be removed in later version of Revit.

Deprecated member(s)
Autodesk.Revit.DB.GBXMLExportOptions.Encoding
Autodesk.Revit.DB.GBXMLExportOptions.ExportEnergyModelType
Autodesk.Revit.DB.GBXMLExportOptions.ExportAnalyticalSystems

The following obsolete API members have been removed in this release. The options are no longer available in the UI. Classes and methods in detailed energy analysis replace these. Consult the API documentation from prior releases for information on the replacements to use.

Removed member(s)
Autodesk.Revit.DB.GBXMLExportOptions(massZonelds, massIds)
Autodesk.Revit.DB.GBXMLExportOptions.IsConceptual
Autodesk.Revit.DB.GBXMLExportOptions.GetMassIds()
Autodesk.Revit.DB.GBXMLExportOptions.GetMassZonelds()

Failures API changes

The following failure definitions are deprecated and may be removed in a later version of Revit.

Deprecated member(s)	Replacement member(s)
BuiltInFailures.ImportFailures.ATFNonUniformScalingTransform	BuiltInFailures.ImportFailures.ImportContainedUnsupportedTransform
BuiltInFailures.ImportFailures.ATFContainedUnsupportedGeometricData	BuiltInFailures.ImportFailures.ImportContainedUnsupportedGeometricData
BuiltInFailures.RebarFailures.Inverted3dShape	N/A

Events API change

The events:

- Application.DocumentClosing
- Application.DocumentClosed

previously would be invoked when the event Application.DocumentOpening had been canceled (due to file corruption or other reasons). Now, these events will not be invoked in those cancellation situations.

Stairs API change

The property:

- StairsPathType.ArrowheadTypeId

can now be assigned a value of ElementId.InvalidElementId, which corresponds to "None" for this property.

Structure API changes

Rebar Cranking API Additions and Changes

There are multiple new classes and members and some deprecations and replacements to support the new capabilities around Rebar cranking and corresponding settings.

First, API references to Hooks and end conditions of rebar and rebar shapes have been renamed and/or replaced with new classes. This includes the renaming of parameters (see Parameter API changes), as well as the new enumerated type:

- Autodesk.Revit.DB.Structure.RebarTerminationOrientation

where this type replaces RebarHookOrientation:

Deprecated member	Replacement member	Details
RebarHookOrientation	RebarTerminationOrientation	The Enum was renamed to reflect that it is used by hooks and cranks too. Both have the same options.

Other new classes related to rebar cranking include:

- Autodesk.Revit.DB.Structure.RebarCrankTypeUtils - utilities related to Rebar Crank Types and related operations
- Autodesk.Revit.DB.Structure.BarTerminationsData - stores data about reinforcement's terminations (e.g. hooks, cranks, end treatments)
- Autodesk.Revit.DB.Structure.RebarShapeTerminationsData - stores data about rebar shape's terminations (e.g. hooks, cranks, end treatments)
- Autodesk.Revit.DB.Structure.RebarCrankOverridableParameters - stores the formula parameter ids defined in the RebarShape family which are associated with crank length, crank offset length, crank straight length and crank angled length parameters
- Autodesk.Revit.DB.Structure.RebarEndType - enumerated type containing options to be added at the end of a Rebar.

As a result of these changes and new data classes, new and replacement members have been added to several previously available classes.

Autodesk.Revit.DB.Structure.Rebar

New members:

- Rebar.GetCrankTypeId()
- Rebar.SetCrankTypeId()
- Rebar.EnableCrankLengthOverride()
- Rebar.IsCrankLengthOverrideEnabled()
- Rebar.GetCrankRatio()
- Rebar.SetCrankRatioOverride()
- Rebar.GetCrankOffsetLength()
- Rebar.SetCrankOffsetLengthOverride()
- Rebar.GetCrankStraightLength()
- Rebar.SetCrankStraightLengthOverride()
- Rebar.GetCrankAngledLength()
- Rebar.SetCrankAngledLengthOverride()
- Rebar.GetCrankLength()
- Rebar.SetCrankLengthOverride()
- Rebar.GetOverridableCrankParameters()

Replacements:

Deprecated member(s)	Replacement member(s)
Rebar.CreateFromCurves(Document doc, RebarStyle style, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, Element host, XYZ norm, IList<Curve> curves, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient, double terminationRotationAngleAtStart, double terminationRotationAngleAtEnd, ElementId endTreatmentTypeIdAtStart, ElementId	Rebar.CreateFromCurves(Document doc, RebarStyle style, RebarBarType barType, Element host, XYZ norm, IList<Curve> curves, BarTerminationsData barTerminationsData, bool useExistingShapeIfPossible, bool createNewShape)

Deprecated member(s)	Replacement member(s)
endTreatmentTypeldAtEnd, bool useExistingShapeIfPossible, bool createNewShape	
Rebar.CreateFromCurves(Document doc, RebarStyle style, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, Element host, XYZ norm, IList<Curve> curves, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient, bool useExistingShapeIfPossible, bool createNewShape)	Rebar.CreateFromCurves(Document doc, RebarStyle style, RebarBarType barType, Element host, XYZ norm, IList<Curve> curves, BarTerminationsData barTerminationsData, bool useExistingShapeIfPossible, bool createNewShape)
Rebar.CreateFromCurvesAndShape(Document doc, RebarShape rebarShape, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, Element host, XYZ norm, IList<Curve> curves, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient, double terminationRotationAngleAtStart, double terminationRotationAngleAtEnd, ElementId endTreatmentTypeldAtStart, ElementId endTreatmentTypeldAtEnd)	Rebar.CreateFromCurvesAndShape(Document doc, RebarShape rebarShape, RebarBarType barType, Element host, XYZ norm, IList<Curve> curves, BarTerminationsData barTerminationsData)
Rebar.CreateFromCurvesAndShape(Document doc, RebarShape rebarShape, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, Element host, XYZ norm, IList<Curve> curves, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient)	Rebar.CreateFromCurvesAndShape(Document doc, RebarShape rebarShape, RebarBarType barType, Element host, XYZ norm, IList<Curve> curves, BarTerminationsData barTerminationsData)
Rebar.RebarShapeMatchesCurvesAndHooks(RebarShape rebarShape, RebarBarType barType, XYZ norm, IList<Curve> curves, RebarHookType startHook, RebarHookType endHook, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient)	RebarShape.RebarShapeMatchesCurvesAndTerminationsData(RebarShape rebarShape, RebarBarType barType, XYZ norm, IList<Curve> curves, RebarShapeTerminationsData rebarShapeTerminationsData)
Rebar.RebarShapeMatchesCurvesHooksAndEndTreatment(RebarShape rebarShape, RebarBarType barType, XYZ norm, IList<Curve> curves, RebarHookType startHook, RebarHookType endHook, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient, double terminationRotationAngleAtStart, double terminationRotationAngleAtEnd, ElementId endTreatmentTypeldAtStart, ElementId endTreatmentTypeldAtEnd)	RebarShape.RebarShapeMatchesCurvesAndTerminationsData(RebarShape rebarShape, RebarBarType barType, XYZ norm, IList<Curve> curves, RebarShapeTerminationsData rebarShapeTerminationsData)
Rebar.GetHookOrientation(int end)	Rebar.GetTerminationOrientation(int end)
Rebar.SetHookOrientation(int end, RebarHookOrientation terminationOrientation)	Rebar.SetTerminationOrientation(int end, RebarTerminationOrientation terminationOrientation)
Rebar.GetHookRotationAngle(int end)	Rebar.GetTerminationRotationAngle(int end);
Rebar.SetHookRotationAngle(double rotationAngle, int end)	Rebar.SetTerminationRotationAngle(double rotationAngle, int end)

Autodesk.Revit.DB.Structure.RebarBarType

New members:

- RebarBarType.GetAutoCalculatedCrank()

- RebarBarType.SetAutoCalculatedCrank()
- RebarBarType.GetCrankRatio()
- RebarBarType.SetCrankRatio()
- RebarBarType.CanCrankParametersBeComputedUsingCrankRatio()
- RebarBarType.GetCrankOffsetLength()
- RebarBarType.SetCrankOffsetLength()
- RebarBarType.CanCrankParametersBeComputedUsingCrankOffsetLength()
- RebarBarType.GetCrankStraightLength()
- RebarBarType.SetCrankStraightLength()
- RebarBarType.CanCrankParametersBeComputedUsingCrankStraightLength()
- RebarBarType.GetCrankAngledLength()
- RebarBarType.SetCrankAngledLength()
- RebarBarType.CanCrankParametersBeComputedUsingCrankAngledLength()
- RebarBarType.GetCrankLength()
- RebarBarType.SetCrankLength()

Autodesk.Revit.DB.Structure.RebarContainer

Deprecated member(s)	Replacement member(s)
RebarContainer.AppendItemFromCurves(RebarStyle style, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, XYZ normal, IList<Curve> curves, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient, bool useExistingShapeIfPossible, bool createNewShape)	RebarContainer.AppendItemFromCurves(RebarStyle style, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, XYZ normal, IList<Curve> curves, RebarTerminationOrientation startHookOrient, RebarTerminationOrientation endHookOrient, bool useExistingShapeIfPossible, bool createNewShape)
RebarContainer.AppendItemFromCurvesAndShape(RebarShape rebarShape, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, XYZ normal, IList<Curve> curves, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient)	RebarContainer.AppendItemFromCurvesAndShape(RebarShape rebarShape, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, XYZ normal, IList<Curve> curves, RebarTerminationOrientation startHookOrient, RebarTerminationOrientation endHookOrient)

Autodesk.Revit.DB.Structure.RebarContainerItem

Deprecated member(s)	Replacement member(s)
RebarContainerItem.SetFromCurves(RebarStyle style, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, XYZ norm, IList<Curve> curves, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient, bool useExistingShapeIfPossible, bool createNewShape);	RebarContainerItem.SetFromCurves(RebarStyle style, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, XYZ norm, IList<Curve> curves, RebarTerminationOrientation startHookOrient, RebarTerminationOrientation endHookOrient, bool useExistingShapeIfPossible, bool createNewShape)
RebarContainerItem.SetFromCurvesAndShape(RebarShape rebarShape, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, XYZ norm, IList<Curve> curves, RebarHookOrientation startHookOrient, RebarHookOrientation endHookOrient)	RebarContainerItem.SetFromCurvesAndShape(RebarShape rebarShape, RebarBarType barType, RebarHookType startHook, RebarHookType endHook, XYZ norm, IList<Curve> curves, RebarTerminationOrientation startHookOrient, RebarTerminationOrientation endHookOrient)
RebarContainerItem.SetHookOrientation(int end, RebarHookOrientation hookOrientation)	RebarContainerItem.SetTerminationOrientation(int end, RebarTerminationOrientation hookOrientation)
RebarContainerItem.GetHookOrientation(int end)	RebarContainerItem.GetTerminationOrientation(int end)

Autodesk.Revit.DB.Structure.RebarFreeFormAccessor

New method:

- RebarFreeFormAccessor.GetCrankTypeldAtIndex()

Deprecated member(s)	Replacement member(s)
RebarFreeFormAccessor.GetHookOrientationAngle(int end);	Rebar.GetTerminationRotationAngle(int end)
RebarFreeFormAccessor.SetHookOrientationAngle(int end, double angle);	Rebar.SetTerminationRotationAngle(double rotationAngle, int end)
RebarFreeFormAccessor.GetHookOrientationAngleAtIndex(int barPositionIndex, int end);	RebarFreeFormAccessor.GetTerminationRotationAngleAtIndex(int barPositionIndex, int end)
RebarFreeFormAccessor.GetHookPlaneNormalForBarIdx(int end, int barPositionIndex)	RebarFreeFormAccessor.GetTerminationPlaneNormalForBarIndex(int end, int barPositionIndex)
RebarFreeFormAccessor.SetHookPlaneNormalForBarIdx(int end, int barPositionIndex, XYZ terminationPlaneNormal);	RebarFreeFormAccessor.SetTerminationPlaneNormalForBarIndex(int end, int barPositionIndex, XYZ terminationPlaneNormal)
RebarFreeFormAccessor.CanBeHookNormal(int barIndex, int end, XYZ normal);	RebarFreeFormAccessor.CanBeTerminationPlaneNormal(int barIndex, int end, XYZ normal)
RebarFreeFormAccessor.GetHookOrientationAtIndex(int barPositionIndex, int end);	RebarFreeFormAccessor.getTerminationOrientationAtIndex(int barPositionIndex, int end)

Autodesk.Revit.DB.Structure.RebarShape

Deprecated member(s)	Replacement member(s)
RebarShape.Create(Document doc, RebarShapeDefinition definition, RebarShapeMultiplanarDefinition multiplanarDefinition, RebarStyle style, StirrupTieAttachmentType attachmentType, int startHookAngle, RebarHookOrientation startHookOrientation, int endHookAngle, RebarHookOrientation endHookOrientation, int higherEnd)	RebarShape.Create(Document doc, RebarShapeDefinition definition, RebarShapeMultiplanarDefinition multiplanarDefinition, RebarStyle style, StirrupTieAttachmentType attachmentType, int higherEnd, RebarShapeTerminationsData rebarShapeTerminationsData)
RebarShape.Create(Document doc, RebarShapeDefinition definition, RebarShapeMultiplanarDefinition multiplanarDefinition, RebarStyle style, StirrupTieAttachmentType attachmentType, int startHookAngle, RebarHookOrientation startHookOrientation, int endHookAngle, RebarHookOrientation endHookOrientation, int higherEnd, double terminationRotationAngleAtStart, double terminationRotationAngleAtEnd, ElementId endTreatmentTypeldAtStart, ElementId endTreatmentTypeldAtEnd)	RebarShape.Create(Document doc, RebarShapeDefinition definition, RebarShapeMultiplanarDefinition multiplanarDefinition, RebarStyle style, StirrupTieAttachmentType attachmentType, int higherEnd, RebarShapeTerminationsData rebarShapeTerminationsData)
GetDefaultHookAngle(int end);	RebarShape.GetTerminationsData()
RebarShape.GetDefaultHookOrientation(int end)	RebarShape.GetTerminationsData()
RebarShape.GetEndTreatmentTypeld(int end)	RebarShape.GetTerminationsData()
RebarShape.GetHookRotationAngle(int end)	RebarShape.GetTerminationsData()

Deprecated member(s)	Replacement member(s)
RebarShape.SetEndTreatmentTypeId(ElementId endTreatmentId, int end);	RebarShape.Create(Document doc, RebarShapeDefinition definition, RebarShapeMultiplanarDefinition multiplanarDefinition, RebarStyle style, StirrupTieAttachmentType attachmentType, int higherEnd, RebarShapeTerminationsData rebarShapeTerminationsData)
RebarShape.SetHookRotationAngle(double rotationAngle, int end)	RebarShape.Create(Document doc, RebarShapeDefinition definition, RebarShapeMultiplanarDefinition multiplanarDefinition, RebarStyle style, StirrupTieAttachmentType attachmentType, int higherEnd, RebarShapeTerminationsData rebarShapeTerminationsData)
RebarShape.IsMatchIgnoringHooks(RebarShape otherShape)	RebarShape.IsMatchIgnoringTerminations(RebarShape otherShape)

Autodesk.Revit.DB.Structure. RebarUpdateCurvesData

Deprecated member(s)	Replacement member(s)
RebarUpdateCurvesData.GetHookOrientationAngle(int end)	RebarUpdateCurvesData.GetTerminationRotationAngle(int end)
RebarUpdateCurvesData.SetHookOrientationAngle(int end, double angle)	RebarUpdateCurvesData.SetTerminationRotationAngle(int end, double angle)
RebarUpdateCurvesData.GetHookPlaneNormalForBarIdx(int end, int barPositionIndex)	RebarUpdateCurvesData.GetTerminationPlaneNormalForBarIdx(int end, int barPositionIndex)
RebarUpdateCurvesData.SetHookPlaneNormalForBarIdx(int end, int barPositionIndex, XYZ terminationPlaneNormal)	RebarUpdateCurvesData.SetTerminationPlaneNormalForBarIdx(int end, int barPositionIndex, XYZ terminationPlaneNormal)

Autodesk.Revit.DB.Structure. RebarBendData

New members:

- RebarBendData.CrankOffsetLength0
- RebarBendData.CrankOffsetLength1
- RebarBendData.CrankStraightLength0
- RebarBendData.CrankStraightLength1
- RebarBendData.CrankAngledLength0
- RebarBendData.CrankAngledLength1
- RebarBendData.CrankRatio0
- RebarBendData.CrankRatio1
- RebarBendData.CrankLength0
- RebarBendData.CrankLength1
- RebarBendData.TerminationRotationAngle0
- RebarBendData.TerminationRotationAngle1

Deprecated member(s)	Replacement member(s)
RebarBendData.RebarBendData(RebarBarType barType, RebarHookType hookType0, RebarHookType hookType1, RebarStyle style, RebarHookOrientation	RebarBendData.RebarBendData(RebarBarType barType, RebarStyle style, BarTerminationsData barTerminationsData)

Deprecated member(s)	Replacement member(s)
terminationOrientation0, RebarHookOrientation terminationOrientation1)	
RebarBendData.HookOrient0	RebarBendData.TerminationOrientation0
RebarBendData.HookOrient1	RebarBendData.TerminationOrientation1

Dimension visibility

The new properties

- BendingDetailCustomFieldProperties.AngularDimensionsForCrankEnabled
- RebarBendingDetailType .AngularDimensionsForCrankEnabled

identifies if angular dimensions that have a reference set on a crank will be shown.

Extensions to enumerations related to Cranking

Several enumerated types, including:

- Autodesk.Revit.DB.Structure.RebarHandleBehavior
- Autodesk.Revit.DB.Structure.RebarCouplerError
- Autodesk.Revit.DB.Structure.RebarSpliceError
- Autodesk.Revit.DB.Structure.RebarSpliceByRulesError

have been extended to new statuses and values related to Rebar Cranking.

Shape-driven Rebar

The following property is deprecated and may be removed in a later version of Revit.

Deprecated member(s)	Replacement member(s)
ReinforcementSettings.NumberVaryingLengthRebarsIndividually	ReinforcementSettings.NumberingMethod

Free Form Rebar

The following methods are deprecated and may be removed in a later version of Revit.

Deprecated member(s)	Replacement member(s)
Rebar.CreateFreeForm(Document doc, RebarBarType barType, Element host, IList<CurveLoop> curves, out RebarFreeFormValidationResult error);	Rebar.CreateFreeForm(Document doc, RebarBarType barType, Element host, IList<CurveLoop> curves, RebarStyle style)
Rebar.CreateFreeForm(Document doc, RebarBarType barType, Element host, IList<IList<Curve>> curves, out RebarFreeFormValidationResult error);	Rebar.CreateFreeForm(Document doc, RebarBarType barType, Element host, IList<IList<Curve>> curves, RebarStyle style);

These new members return objects of the new class

- RebarFreeFormCreationResult

which encapsulates the result of the rebar free form creation, including the created Rebar element and information about errors.

Structural Connections

The following methods from Autodesk.Revit.DB.Structure.StructuralConnectionHandlerType class are deprecated and may be removed in a later version of Revit:

Deprecated member(s)	Replacement member(s)
AddElementsToCustomConnection(StructuralConnectionHandler structuralConnectionHandler, IList<Reference> references)	UpdateCustomConnectionType(StructuralConnectionHandler structuralConnectionHandler, IList<Reference> addReferences, IList<Reference> removeReferences)
RemoveMainSubelementsFromCustomConnection(StructuralConnectionHandler structuralConnectionHandler,IList<Subelement> subelements)	UpdateCustomConnectionType(StructuralConnectionHandler structuralConnectionHandler, IList<Reference> addReferences, IList<Reference> removeReferences)

Electrical API changes

This version of Revit introduces new elements, classes and relationships used to define the properties of cables, conductors and wires.

New classes represent types and materials

The new classes:

- Autodesk.Revit.DB.Electrical.ConductorMaterial
- Autodesk.Revit.DB.Electrical.TemperatureRating
- Autodesk.Revit.DB.Electrical.InsulationMaterial
- Autodesk.Revit.DB.Electrical.ConductorSize
- Autodesk.Revit.DB.Electrical.CableSize
- Autodesk.Revit.DB.Electrical.CableType

represent the properties and relationships that can be assigned to an electrical system or to a particular cable or wire.

ElectricalSystem API additions

The new properties:

- ElectricalSystem.CableType
- ElectricalSystem.CableSize
- ElectricalSystem.HotConductorSize
- ElectricalSystem.GroundConductorSize
- ElectricalSystem.NeutralConductorSize

- ElectricalSystem.OtherConductorSize
- ElectricalSystem.OtherConductorsNumber

provide read and write access to the assigned options and types for an Electrical System.

WireType member API changes

To support the changes made to the electrical data model some members of WireType were changed to a new data type:

Changed properties	Old property type	New property type	Notes
WireMaterial	Autodesk.Revit.DB.Electrical.WireMaterialType	Autodesk.Revit.DB.ElementId	The conductor material of the wire type.
TemperatureRating	Autodesk.Revit.DB.Electrical.TemperatureRatingType	Autodesk.Revit.DB.ElementId	The conductor temperature rating of the wire type.
Insulation	Autodesk.Revit.DB.Electrical.InsulationType	Autodesk.Revit.DB.ElementId	The conductor insulation material of the wire type. InsulationType object can no longer set to WireType, use Autodesk.Revit.DB.Electrical.InsulationMaterial instead.
MaxSize	Autodesk.Revit.DB.Electrical.WireSize	System.String	The max conductor size name of the wire type. WireSize object can no longer set to WireType, use Autodesk.Revit.DB.Electrical.ConductorSize instead.

Additional Electrical API changes

The following classes are deprecated and may be removed in later version of Revit.

Deprecated classes	Replacement classes	Notes
Autodesk.Revit.DB.Electrical.WireSize	Autodesk.Revit.DB.Electrical.ConductorSize	ConductorSizes will be created from the existing WireSizes during document upgrade.
Autodesk.Revit.DB.Electrical.CorrectionFactor	None	There is no replacement because Revit no longer supports this.
Autodesk.Revit.DB.Electrical.GroundConductorSize	Autodesk.Revit.DB.Electrical.ConductorSize	ConductorSizes will be created from the existing GroundConductorSizes during document upgrade.
Autodesk.Revit.DB.Electrical.InsulationType	Autodesk.Revit.DB.Electrical.InsulationMaterial	InsulationMaterials will be created from the existing InsulationTypes during document upgrade.
Autodesk.Revit.DB.Electrical.TemperatureRatingType	Autodesk.Revit.DB.Electrical.TemperatureRating	TemperatureRatings will be created from the existing TemperatureRatingTypes during document upgrade.

Autodesk.Revit.DB.Electrical.WireMaterialType	Autodesk.Revit.DB.Electrical.ConductorMaterial	ConductorMaterials will be created from the existing WireMaterialTypes during document upgrade.
---	--	---

The following members have been deprecated and may be removed in later version of Revit:

Deprecated member(s)	Replacement member(s)	Notes
ElectricalSystem.WireType	ElectricalSystem.CableType	WireMaterialType object can no longer set to WireType, use Autodesk.Revit.DB.Electrical.ConductorMaterial instead.
ElectricalSystem.WireSizeString	ElectricalSystem.CableSize	TemperatureRatingType object can no longer set to WireType, use Autodesk.Revit.DB.Electrical.TemperatureRating instead.
ElectricalSystem.VoltageDrop	None	There is no replacement for this property because Revit no longer supports this.
ElectricalSetting.WireMaterialTypes	ConductorMaterial.GetConductorMaterialIds()	
ElectricalSetting.AddWireMaterialType()	ConductorMaterial.Create()	
ElectricalSetting.RemoveWireMaterialType()	Document.Delete()	
ElectricalSetting.AddWireType()	WireType.Duplicate()	WireMaterialType object can no longer set to the WireType, use property WireType.WireMaterial instead. TemperatureRatingType object can no longer set to the WireType, use property WireType.TemperatureRating instead. InsulationType object can no longer set to the WireType, use property WireType.Insulation instead. WireSize object can no longer set to the WireType, use property WireType.MaxSize instead.

Zone API changes

System-Zone elements have been expanded to support sketch-based or space-based elements, and now have an assignable type.

The following method has been deprecated and replaced:

Deprecated member(s)	Replacement member(s)
GenericZone.Create()	GenericZone.CreateSketchBased() - use this to create sketch-based Zones like Create() used to GenericZone.CreateSpaceBased() - use this to create space-based Zones (a new option in this release)
Document.NewZone()	Create the GenericZone object instead.
Zone.IsDefaultZone	Use the properties of GenericZone

Zone.GrossArea	
Zone.Volume	
Zone.GrossVolume	
Zone.Perimeter	
Zone.ServiceType	
Zone.CalculatedSupplyAirflow	
Zone.CalculatedHeatingLoad	
Zone.CalculatedCoolingLoad	
Zone.HeatingSetPoint	
Zone.CoolingSetPoint	Use the properties of GenericZone and SystemZoneElementType
Zone.HeatingAirTemperature	
Zone.CoolingAirTemperature	
Zone.HumidificationSetPoint	
Zone.DehumidificationSetPoint	
Zone.AddSpaces	GenericZone.AddSpaces
Zone.RemoveSpaces	GenericZone.RemoveSpaces

The new classes:

- Autodesk.Revit.DB.Analysis.GenericZoneType
- Autodesk.Revit.DB.Analysis.GenericZoneTypeDomainData
- Autodesk.Revit.DB.Mechanical.SystemZoneElementType
- Autodesk.Revit.DB.Mechanical.SystemZoneTypeData

are used to access the element types of Zones and the domain-specific data they may contain.

The new method:

- Autodesk.Revit.DB.Analysis.GenericZone.GetSpaceIds()

returns the spaces contained in a space-based zone.

IFC API changes

Importer Options

The new class:

- Autodesk.Revit.DB.IFC.IFCHybridImportOptions

establishes extra IFC Import Options for AnyCAD processing mode, including options for how to process grids, levels and property sets.

With this new options class, methods of IFCHybridImport have been deprecated and replaced:

Deprecated member(s)	Replacement member(s)
IFCHybridImport.ImportElements(Document, String)	IFCHybridImport.ImportElements(Document, String, IFCHybridImportOptions)
IFCHybridImport.UpdateElements(Document, String)	IFCHybridImport.UpdateElements(Document, String, IFCHybridImportOptions)

The new properties:

- IFCImportOptions.LinkOrientation
- IFCImportOptions.LinkPosition

indicate the placement option and orientation options for an IFC import.

Category Mapping

The following replacement methods have been introduced:

Deprecated member(s)	Replacement member(s)
IFCCategoryTemplate.SetActiveTemplate(Document)	IFCCategoryTemplate.SetActiveTemplate()

Also, the newly added property:

- IFCCategoryTemplate.IsValidCategoryMappingFile

indicates whether the Category Mapping File is valid or not.

Obsolete API removal

The following API members and classes which had previously been marked Deprecated have been removed in this release. Consult the API documentation from prior releases for information on the replacements to use:

Constructors

- Autodesk.Revit.DB.ElementId(int id) (constructor using a 32-bit integer value)

Properties

- Autodesk.Revit.DB.ElementId.IntegerValue
- Rebar.DistributionType
- Floor.SlabShapeEditor
- RoofBase.SlabShapeEditor
- DuctPressureDropData.Viscosity
- DuctSettings.AirViscosity
- AnalyticalBusData.Voltage
- AnalyticalPowerSourceData.Voltage
- AnalyticalTransferSwitchData.Voltage

- AnalyticalDistributionNodePropertyData.NumberOfPhases

Methods

- OptionalFunctionalityUtils.IsPDFImportAvailable()
- RebarConstraintsManager.GetConstraintCandidatesForHandle(RebarConstrainedHandle)
- RebarConstraintsManager.SetPreferredConstraintForHandle(RebarConstrainedHandle, RebarConstraint)
- FilterNumericRuleEvaluator.Evaluate(Int32, Int32)
- ParameterFilterRuleFactory.CreateBeginsWithRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateNotBeginsWithRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateContainsRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateNotContainsRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateEndsWithRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateNotEndsWithRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateEqualsRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateNotEqualsRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateGreaterOrEqualsRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateGreaterRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateLessOrEqualsRule(ElementId, String, Boolean)
- ParameterFilterRuleFactory.CreateLessRule(ElementId, String, Boolean)
- SlabShapeEditor.DrawPoint(XYZ)
- SlabShapeEditor.DrawSplitLine(SlabShapeVertex, SlabShapeVertex)
- LinearArray.IsValidArraySize()
- RadialArray.IsValidArraySize()
- AnalyticalSurfaceBase.IsCurveLoopValid()

Enums

- Autodesk.Revit.DB.Structure.DistributionType

API additions

Add-in Management API additions

With the introduction of user control for loaded add-ins, new APIs have been added to facilitate the management of add-in settings in and out of Revit. These APIs are a part of RevitAddinUtility.dll and documented in RevitAddinUtility.chm.

The new class:

- Autodesk.RevitAddIns.Manager.AddInsManagerSettings

contains the settings to enable/disable loading of add-ins registered with Revit. It offers the following members:

- static AddInsManagerSettings.Get() - gets the settings for the current Windows user.
- AddInsManagerSettings.GetAllAddInItemSettings() - returns all Add-in settings.
- AddInsManagerSettings.GetAddInItemSettings() - returns a particular Add-in settings object for the given id.
- AddInsManagerSettings.CreateAddInItemSettings() - creates a new Add-in settings object.
- AddInsManagerSettings.Save() - saves any changes made.
- AddInsManagerSettings.DisableAllAddIns - used to disable (or enable) loading of all Add-ins (for those that are permitted to be disabled).

The new class:

- Autodesk.RevitAddIns.Manager.AddInItemSettings

represents the settings of an Add-in registered with Revit. It offers the following properties:

- AddInItemSettings.Disabled - used to disable (or enable) loading of a particular add-in in the next Revit session.
- AddInItemSettings.GUID - the add-in id.
- AddInItemSettings.Name - the add-in name.
- AddInItemSettings.Vendor - the add-in vendor.
- AddInItemSettings.LoadTime - the time taken to load the add-in the last time it loaded.

Document API addition

The new method:

- Document.IsInEditMode()

identifies if Revit is currently in an active edit mode or not.

The new method with a corresponding new enum:

- Autodesk.Revit.DB.EditModeType Document.GetActiveEditMode()

returns the active edit mode type, where the value `None` indicates that Revit is not in edit mode.

Geometry API additions

Solid utilities

The new methods:

- SolidUtils.ComputesTopologicallyClosed()
- SolidUtils.ComputesGeometricallyClosed()

run a computation to check the closure of the input solid or shell and the one or more components it contains.

Import/Export Data Exchange API additions

NurbsSurfaceData

Added new element-wise accessors to NurbsSurfaceData in order to avoid deep-copying large arrays of values when only one element from the array is needed.

The new methods:

- NurbsSurfaceData.GetNumberOfKnotsU()
- NurbsSurfaceData.GetKnotU()
- NurbsSurfaceData.GetNumberOfKnotsV()
- NurbsSurfaceData.GetKnotV()
- NurbsSurfaceData.GetNumberControlPoints()

- NurbsSurfaceData.GetControlPoint()
- NurbsSurfaceData.GetNumberOfWeights()
- NurbsSurfaceData.GetWeight()

Wall & Compound Structure API additions

Wall attachments

The new methods:

- Wall.GetAttachmentIds()
- Wall.AddAttachment()
- Wall.RemoveAttachment(ElementId targetId)
- Wall.RemoveAttachment(ElementId targetId, AttachmentLocation attachmentLocation)
- Wall.IsValidTargetAttachment()

provide access to the attachment settings for walls on their top or bottom (to specific model elements like roofs, floors, ceilings, toposolids, or other walls).

CompoundStructure Priority

This enhancement allows for the customization of CompoundStructure layer priority without needing to change its function, providing more flexibility in the design process. The new APIs:

- CompoundStructureLayer.LayerPriority
- CompoundStructure.GetLayerPriority()
- CompoundStructure.SetLayerPriority()
- CompoundStructure.IsValidLayerPriority()
- CompoundStructure.ResetLayerPriority()
- CompoundStructure.ResetAllLayersPriorities()

provide read and write access to the priority values, plus the option to reset them to default.

Toposolid API additions

Subdivision creation

The new method:

- Toposolid.CreateSubDivision(Document, ElementId, IList<CurveLoop>)

creates a Toposolid subdivision element with the input toposolid as its host, and a specified type to be used for the new subdivision.

Stability for cutting and void operations

The new static routines:

- Toposolid.IsCutVoidStabilityEnabled(Document)
- Toposolid.SetCutVoidStability(Document, bool)

access the option to use random shifting to find a successful Boolean operation for cutting and joining Toposolid objects.

Ceiling API additions

The new method:

- Revit.DB.Ceiling.GetCeilingGridLine()

returns the geometric representation of the ceiling grid, optionally including the ceiling boundary.

Room, Area and Space API additions

The new methods in SpatialElement:

- SpatialElement.GetDefaultLocation() - Allows user to get the default location of the spatial element. The default location is a point where a spatial element is placed when it is created automatically without specifying a point.
- SpatialElement.Recenter() - Allows user to move the spatial element to its default location.

The new method in SpatialElementTag:

- SpatialElementTag.MoveToReferenceLocation()

allows user to move the SpatialElementTag to the location of SpatialElement that the tag is associated with.

Parameter API additions

The new method:

- ParameterUtils.GetBuiltInParameterGroupTypeId()

gets the parameter group identifier corresponding to the given built-in parameter identifier.

View API additions

The new property to TemporaryViewModes:

- TemporaryViewModes.AcceleratedGraphicsMode - Represents the current state of the Accelerated Graphics mode in the associated view.

The new value to Enum TemporaryViewModes:

- AcceleratedGraphicsMode - Represents the view is in the mode that shows Accelerated Graphics.

Graphics API additions

Temporary Graphics

The new method:

- `TemporaryGraphicsManager.SetToolTip()`

sets the tooltip for the given temporary graphics object.

[Directly drawn graphics \(DirectContext3D\)](#)

The new methods:

- `DirectContext3DHandleSettings.GetColor()`
- `DirectContext3DHandleSettings.SetColor()`

get and set the color override applied to the handle and the associated DirectContext3D graphics.

Also, the index buffer parameter input in:

- `DirectContext3D.FlushBuffer()`

has been updated to optional, where null is permitted, but only if indexCount is also 0.

Independent Tag API addition

The new method:

- `Autodesk.Revit.DB.IndependentTag.HasTagText()`

allows users to determine if the IndependentTag object has a valid tag text.

Failure API addition

The new method:

- `LabelUtils.GetFailureSeverityName()`

returns the user-visible name for the severity of a warning.

IFC API additions

[ExporterIFC utilities](#)

The new method:

- `ExporterIFC. GetFamilyInstanceAssemblyOffset (FamilyInstance)`

returns the translation for the family instance's adjustment within an Assembly.

[IFC Parameter Mapping](#)

The new classes and enums:

- Autodesk.Revit.DB.IFCParameterTemplate
- Autodesk.Revit.DB.IFCPropertyMappingInfo
- Autodesk.Revit.DB.PropertySelectionType
- Autodesk.Revit.DB.PropertySetupType

allows control over the way that Revit parameters are mapped to IFC properties, including the ability to create, retrieve and modify IFC parameter mapping templates.

Structural Analytical API additions

Connectors in Structural Analytical elements

The new property:

- AnalyticalElement.ConnectorManager

retrieves the connector manager of a given structural analytical element. It will return connectors of Domain.StructuralAnalytical.

Points in Structural Analytical networks

The new method:

- Hub.GetPointId()

returns the id of the associated Point element.

Structural API additions

Structural connections

The new method:

- StructuralSettings.BoundarySetbackDisabledForSteelElements()

allows users to disable the auto-join setback distances for all structural steel framing, slanted columns, trusses, beam systems and braces.

Load Combinations

Several new values have been added to the Autodesk.Revit.DB.Structure.LoadCombinationState enum:

- ServiceabilityQuasiPermanent
- ServiceabilityFrequent
- ServiceabilityRare
- Accidental
- Special

Reinforcement API additions

Free Form Rebar creation

Added the ability to create Rebar Free Form using server GUIDS.

The new static properties:

- `Rebar.FreeFormAlignedServerGuid`
- `Rebar.FreeFormSurfaceServerGuid`

returns the GUID of the to-be-used external IRebarUpdateServer for computing Aligned or Surface Free Form Rebar.

The new enums:

- `SurfaceDistributionRebarHandles`
- `AlignedDistributionRebarHandles`

correspond to the values that might be returned by `RebarConstrainedHandle.GetCustomHandleTag()` for Aligned or Surface Free Form Rebar.

Rebar Numbering

The new property:

- `ReinforcementSettings.NumberingMethod`

provides access to the setting for how varying length bars are numbered.

Rebar Constraint Relationship

The new method:

- `RebarConstraint.GetTargetRebarHandleBehavior()`

gets the `RebarHandleBehavior` of the handle of the other Rebar Element to which this `RebarConstraint` is attached. (The `RebarConstraintType` of the `RebarConstraint` must be 'ToOtherRebar').

Fabric Sheet and Area

The new members:

- `FabricArea.DistributionOfWiresAtCover`
- `FabricSheet.DistributionOfWiresAtCover`

provide information about which distribution is set closest to the cover.

MEP Analysis API additions

MEP Duct/Pipe pressure loss calculations

The new members:

- `MEPAnalyticalSegment.OverrideType`
- `MEPAnalyticalSegment.SetOverrideType()`
- `MEPAnalyticalSegment.GetLossCoefficientOverride()`
- `MEPAnalyticalSegment.SetLossCoefficientOverride()`
- `MEPAnalyticalSegmentGetPressureDropOverride()`
- `MEPAnalyticalSegment.SetPressureDropOverride()`

support override of the analytical segment coefficient and pressure drop. These allow application of specified loss coefficients or pressure drop values in the network-based flow calculation.

The new methods:

- `MEPAnalyticalSegment.GetFlowConvergenceCharacteristic()`
- `MEPAnalyticalSegment.GetFlowTransitionCharacteristic()`

provide information about the flow characteristics for a given segment.

Electrical API additions

Browser Organization for panel schedules

Browser organization options are now enabled for panel schedules. Users can organize panel schedules by some properties of panels.

The new methods:

- `BrowserOrganization.GetCurrentBrowserOrganizationForPanelSchedules()`
- `BrowserOrganization.GetSortingParameterIdPath()`

allows access to the `BrowserOrganization` that applies to the Panel Schedules section of the project browser and provides access to the parameter Id path used to determine the sorting order of items in the browser.

MEP Fabrication API additions

The new methods:

- `FabricationPart.GetAllFabricationBodyConnectorDefinitions()`
- `FabricationPart.GetAllFabricationDoubleWallConnectorDefinitions()`

return all identifiers for fabrication connector definitions that are compatible with and valid to be assigned to the input connector, for the body and double wall connections respectively.

Energy Analysis API additions

The following read-only properties have been added for access to zone analysis properties:

- EnergyAnalysisZone.HeatingSetPoint
- EnergyAnalysisZone.CoolingSetPoint
- EnergyAnalysisZone.UseHumidificationSetPoint
- EnergyAnalysisZone.UseDehumidificationSetPoint
- EnergyAnalysisZone.HumidificationSetPoint
- EnergyAnalysisZone.DehumidificationSetPoint
- EnergyAnalysisZone.OutsideAirPerPerson
- EnergyAnalysisZone.UseOutsideAirPerPerson
- EnergyAnalysisZone.OutsideAirPerArea
- EnergyAnalysisZone.UseOutsideAirPerArea
- EnergyAnalysisZone.AirChangesPerHour
- EnergyAnalysisZone.UseAirChangesPerHour
- EnergyAnalysisZone.CADObjectUniqueId