

```

#region Assembly RevitAPI.dll, v2.0.50727
// C:\Program Files\Autodesk\Revit Architecture 2012\Program\RevitAPI.dll
#endregion

using Autodesk.Revit.DB;
using Autodesk.Revit.DB.Architecture;
using Autodesk.Revit.DB.Electrical;
using Autodesk.Revit.DB.Mechanical;
using Autodesk.Revit.DB.Plumbing;
using Autodesk.Revit.DB.Structure;
using System;
using System.Collections.Generic;

namespace Autodesk.Revit.Creation
{
    // Summary:
    //     The Document Creation object is used to create new instances of elements
    //     within the Autodesk Revit project.
    //
    public class Document : ItemFactoryBase
    {
        Wall NewWall(CurveArray profile, bool structural);
        Wall NewWall(Curve curve, Level level, bool structural);
        Wall NewWall(CurveArray profile, WallType wallType, Level level, bool structural);
        Wall NewWall(CurveArray profile, WallType wallType, Level level, bool structural, XYZ
normal);
        Wall NewWall(Curve curve, WallType wallType, Level level, double height, double offset,
bool flip, bool structural);
        ElementSet NewWalls(List<ProfiledWallCreationData> dataList);
        ElementSet NewWalls(List<RectangularWallCreationData> dataList);

        Opening NewOpening(Element hostElement, CurveArray profile, bool bPerpendicularFace);
        Opening NewOpening(Element famInstElement, CurveArray profile, eRefFace iFace);
        Opening NewOpening(Level bottomLevel, Level topLevel, CurveArray profile);
        Opening NewOpening(Wall wall, XYZ pntStart, XYZ pntEnd);

        CurtainSystem NewCurtainSystem(FaceArray faces, CurtainSystemType curtainSystemType);
        ElementSet NewCurtainSystem(ReferenceArray faces, CurtainSystemType curtainSystemType);

        Floor NewFloor(CurveArray profile, bool structural);
        Floor NewFloor(CurveArray profile, FloorType floorType, Level level, bool structural);
        Floor NewFloor(CurveArray profile, FloorType floorType, Level level, bool structural, XYZ
normal);
        Floor NewFoundationSlab(CurveArray profile, FloorType floorType, Level level, bool
structural, XYZ normal);
        ContFooting NewFoundationWall(ContFootingType contFootingType, Wall wall);
        Floor NewSlab(CurveArray profile, Level level, Line slopedArrow, double slope, bool
isStructural);
        SlabEdge NewSlabEdge(SlabEdgeType SlabEdgeType, Reference reference);
        SlabEdge NewSlabEdge(SlabEdgeType SlabEdgeType, ReferenceArray references);
        RebarBarType NewRebarBarType();

        BeamSystem NewBeamSystem(CurveArray profile, Level level);
        BeamSystem NewBeamSystem(CurveArray profile, SketchPlane sketchPlane);
    }
}

```

```

BeamSystem NewBeamSystem(CurveArray profile, Level level, XYZ direction, bool is3d);
BeamSystem NewBeamSystem(CurveArray profile, SketchPlane sketchPlane, XYZ direction, bool
is3d);

ExtrusionRoof NewExtrusionRoof(CurveArray profile, ReferencePlane refPlane, Level level,
RoofType roofType, double extrusionStart, double extrusionEnd);
Fascia NewFascia(FasciaType FasciaType, Reference reference);
Fascia NewFascia(FasciaType FasciaType, ReferenceArray references);
FootPrintRoof NewFootPrintRoof(CurveArray footPrint, Level level, RoofType roofType, out
ModelCurveArray footPrintToModelCurvesMapping);
Gutter NewGutter(GutterType GutterType, Reference reference);
Gutter NewGutter(GutterType GutterType, ReferenceArray references);
Truss NewTruss(TrussType trussType, SketchPlane sketchPlane, Curve baseCurve);

Grid NewGrid(Arc arc);
Grid NewGrid(Line line);
ElementSet NewGrids(CurveArray curves);

TopographySurface NewTopographySurface(ICollection<XYZ> points);
SpotDimension NewSpotCoordinate(View view, Reference reference, XYZ origin, XYZ bend, XYZ
end, XYZ refPt, bool hasLeader);
SpotDimension NewSpotElevation(View view, Reference reference, XYZ origin, XYZ bend, XYZ
end, XYZ refPt, bool hasLeader);

FamilyInstance NewFamilyInstance(Curve curve, FamilySymbol symbol, Level level,
StructuralType structuralType);
FamilyInstance NewFamilyInstance(XYZ location, FamilySymbol symbol, Level level,
StructuralType structuralType);
FamilyInstance NewFamilyInstance(XYZ location, FamilySymbol symbol, Element host, Level
level, StructuralType structuralType);

Area NewArea(ViewPlan areaView, UV point);
AreaLoad NewAreaLoad(Element host, XYZ force, bool isReaction, AreaLoadType symbol);
AreaLoad NewAreaLoad(ICollection<XYZ> points, XYZ force, bool isReaction, AreaLoadType symbol);
AreaLoad NewAreaLoad(CurveArray curvesArr, int[] refPntIdxs, int[] ends, ICollection<XYZ> forces,
bool isReaction, AreaLoadType symbol);
AreaLoad NewAreaLoad(CurveArray curvesArr, int[] refPntIdxs, int[] ends, XYZ force0, XYZ
force1, XYZ force2, bool isReaction, AreaLoadType symbol);
AreaReinforcement NewAreaReinforcement(Element host, CurveArray curves, XYZ direction);
ElementSet NewAreas(List<AreaCreationData> dataList);
AreaTag NewAreaTag(ViewPlan areaView, Area room, UV point);
ModelCurve NewAreaBoundaryLine(SketchPlane sketchPlane, Curve geometryCurve, ViewPlan
areaView);
BoundaryConditions NewAreaBoundaryConditions(Element hostElement, TranslationRotationValue
X_Translation, double X_TranslationSpringModulus, TranslationRotationValue Y_Translation,
double Y_TranslationSpringModulus,
TranslationRotationValue Z_Translation, double Z_TranslationSpringModulus);
BoundaryConditions NewAreaBoundaryConditions(Reference reference, TranslationRotationValue
X_Translation, double X_TranslationSpringModulus, TranslationRotationValue Y_Translation,
double Y_TranslationSpringModulus,
TranslationRotationValue Z_Translation, double Z_TranslationSpringModulus);
BoundaryConditions NewLineBoundaryConditions(Element hostElement, TranslationRotationValue
X_Translation, double X_TranslationSpringModulus, TranslationRotationValue Y_Translation,
double Y_TranslationSpringModulus,

```

```

TranslationRotationValue Z_Translation, double Z_TranslationSpringModulus,
TranslationRotationValue X_Rotation, double X_RotationSpringModulus);
    BoundaryConditions NewLineBoundaryConditions(Reference reference, TranslationRotationValue
    X_Translation, double X_TranslationSpringModulus, TranslationRotationValue Y_Translation,
    double Y_TranslationSpringModulus,
TranslationRotationValue Z_Translation, double Z_TranslationSpringModulus,
TranslationRotationValue X_Rotation, double X_RotationSpringModulus);
    BoundaryConditions NewPointBoundaryConditions(Reference reference, TranslationRotationValue
    X_Translation, double X_TranslationSpringModulus, TranslationRotationValue Y_Translation,
    double Y_TranslationSpringModulus,
TranslationRotationValue Z_Translation, double Z_TranslationSpringModulus,
TranslationRotationValue X_Rotation, double X_RotationSpringModulus, TranslationRotationValue
Y_Rotation, double Y_RotationSpringModulus,
TranslationRotationValue Z_Rotation, double Z_RotationSpringModulus);
    PointLoad NewPointLoad(Reference reference, XYZ force, XYZ moment, bool isReaction,
    PointLoadType symbol, SketchPlane plane);
    PointLoad NewPointLoad(XYZ point, XYZ force, XYZ moment, bool isReaction, PointLoadType
    symbol, SketchPlane plane);
    LineLoad NewLineLoad(Element host, IList<XYZ> forces, IList<XYZ> moments, bool uniform,
    bool projected, bool isReaction, LineLoadType symbol, SketchPlane plane);
    LineLoad NewLineLoad(IList<XYZ> points, IList<XYZ> forces, IList<XYZ> moments, bool
    uniform, bool projected, bool isReaction, LineLoadType symbol, SketchPlane plane);
    LineLoad NewLineLoad(Reference reference, IList<XYZ> forces, IList<XYZ> moments, bool
    uniform, bool projected, bool isReaction, LineLoadType symbol, SketchPlane plane);
    LineLoad NewLineLoad(XYZ point, XYZ force, XYZ moment, XYZ point1, XYZ force1, XYZ moment1,
    bool uniform, bool projected, bool isReaction, LineLoadType symbol, SketchPlane plane);
    LoadCase NewLoadCase(string name, LoadNature nature, Category category);
    LoadCombination NewLoadCombination(string name, int typeInd, int stateInd, double[]
    factors, LoadCaseArray cases, LoadCombinationArray combinations, LoadUsageArray usages);
    LoadNature NewLoadNature(string name);
    LoadUsage NewLoadUsage(string name);

MechanicalSystem NewMechanicalSystem(Connector baseEquipmentConnector, ConnectorSet
connectors, DuctSystemType ductSystemType);
FamilyInstance NewCrossFitting(Connector connector1, Connector connector2, Connector
connector3, Connector connector4);
Duct NewDuct(Connector connector1, Connector connector2, DuctType ductType);
Duct NewDuct(XYZ point, Connector connector, DuctType ductType);
Duct NewDuct(XYZ point1, XYZ point2, DuctType ductType);
FamilyInstance NewElbowFitting(Connector connector1, Connector connector2);
FlexDuct NewFlexDuct(IList<XYZ> points, FlexDuctType ductType);
FlexDuct NewFlexDuct(Connector connector1, Connector connector2, FlexDuctType ductType);
FlexDuct NewFlexDuct(Connector connector, IList<XYZ> points, FlexDuctType ductType);
FlexPipe NewFlexPipe(IList<XYZ> points, FlexPipeType pipeType);
FlexPipe NewFlexPipe(Connector connector1, Connector connector2, FlexPipeType pipeType);
FlexPipe NewFlexPipe(Connector connector, IList<XYZ> points, FlexPipeType pipeType);
FamilyInstance NewTakeoffFitting(Connector connector, MEPCurve curve);
FamilyInstance NewTeeFitting(Connector connector1, Connector connector2, Connector
connector3);
FamilyInstance NewTransitionFitting(Connector connector1, Connector connector2);
FamilyInstance NewUnionFitting(Connector connector1, Connector connector2);

ElectricalSystem NewElectricalSystem(Connector connector, ElectricalSystemType elecSysType);
ElectricalSystem NewElectricalSystem(ElementSet electComponents, ElectricalSystemType

```

```

elecSysType);
Wire NewWire(Curve curve, View view, Connector startConnector, Connector endConnector,
WireType wireType, WiringType wiringType);

PathReinforcement NewPathReinforcement(Element host, CurveArray curves, bool flip);

Room NewRoom(Phase phase);
Room NewRoom(Level level, UV point);
Room NewRoom(Room room, PlanCircuit circuit);
ModelCurveArray NewRoomBoundaryLines(SketchPlane sketchPlane, CurveArray curves, View view);
ElementSet NewRooms(Level level);
ElementSet NewRooms(List<RoomCreationData> dataList);
ElementSet NewRooms(Level level, Phase phase);
ElementSet NewRooms(Phase phase, int count);
RoomTag NewRoomTag(Room room, UV point, View view);

Space NewSpace(Phase phase);
Space NewSpace(Level level, UV point);
Space NewSpace(Level level, Phase phase, UV point);
ModelCurveArray NewSpaceBoundaryLines(SketchPlane sketchPlane, CurveArray curves, View view);
ElementSet NewSpaces(Phase phase, int count);
ElementSet NewSpaces(Level level, Phase phase, View view);
SpaceTag NewSpaceTag(Space space, UV point, View view);
Zone NewZone(Level level, Phase phase);

ViewPlan NewAreaViewPlan(string pViewName, Level pLevel, AreaElemType areaElemType);
ViewDrafting NewViewDrafting();
ViewSheet NewViewSheet(FamilySymbol titleBlock);
IndependentTag NewTag(View dbview, Element elemToTag, bool addLeader, TagMode tagMode,
TagOrientation tagOrientation, XYZ pnt);
}
}

```