

AR20462 - IFC Technical Overview and Survey of Autodesk Products, Including Revit 2017

Angel Velez

Sr. Principal Engineer, Product Owner, Autodesk Revit
@avelezsosa



Class summary

This class will start with a technical overview of Industry Foundation Classes (IFC) itself, with an emphasis on the new IFC4 schema and its Model View Definitions (MVDs). We will then take a high-level look at how IFC is supported across the 2017 Autodesk product line. Next we will take a look at Revit's use of IFC, and the UI options available. Finally, the class will briefly look at Revit 2017 software IFC open-source .NET code and the associated Revit software API.

Key learning objectives

At the end of this class, you will be able to:

- Explore the IFC file format, emphasizing IFC 4 and certification
- What's new in 2017 for Autodesk IFC support (including Revit)
- Understand Revit IFC options
- Learn how to make simple changes to the Revit IFC open source

Introduction to IFC

What is IFC? Standard terms

IFC = Industry Foundation Classes

- The format used to exchange data between applications

Schema = The definition of a particular version of IFC

- IFC2x2, IFC2x3, IFC4

MVD = Model View Definition

- The subset of IFC used for a particular workflow



ABCs of IFC (IFC)

IFC = Industry Foundation Classes

- Started in 1994 by Autodesk and industry consortium
- “Classes” refers to initial intention to create common C++ objects to define architectural concepts; now file format based on STEP and XML
- IAI (International Alliance for Interoperability) created to support and promote IFC, now buildingSMART



ABCs of IFC (Schema)

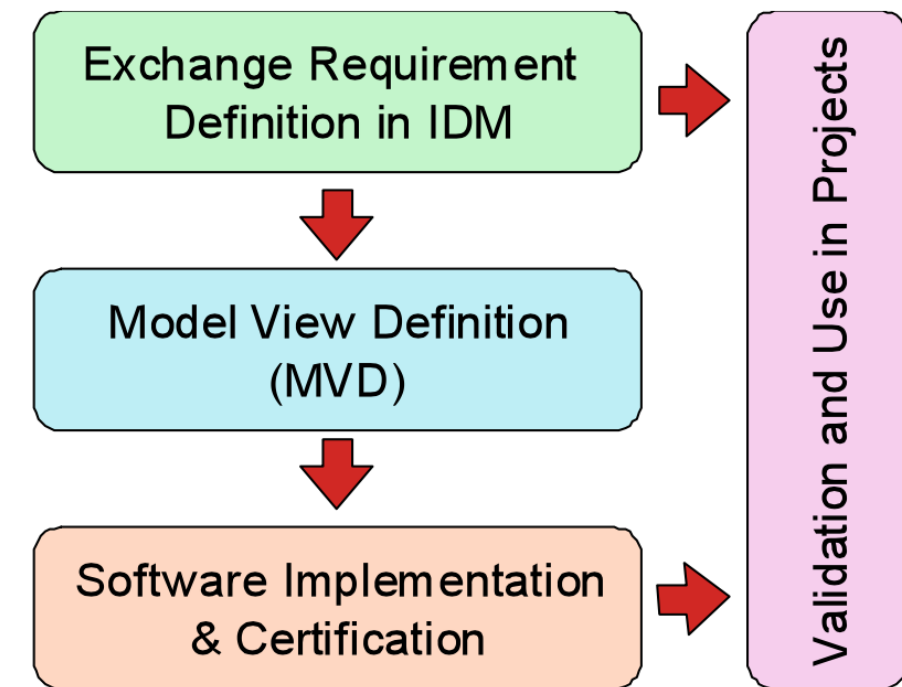
Schema = The definition of a particular version of IFC

- Active versions:
 - IFC2x3 – supported since 2007, most common version
 - IFC4 – partial support since 2012, in progress
- Older versions exist but are generally obsolete

ABCs of IFC (MVD)

MVD = Model View Definition

- An MVD is the subset of IFC suited for a particular workflow. Examples are:
 - Coordination View 2.0
 - FM (Facilities Handover) View (a.k.a. COBie)
 - Design Transfer View
- An IFC file must be generated based on some MVD, perhaps with compatible “Add-ons” such as:
 - QTO (Quantity Take-off)
 - 1st or 2nd level space boundaries
- MVDs are generally associated with a schema
 - IFC2x3 Coordination View 2.0
 - IFC4 Design Transfer View



Revit IFC 2017 Supported MVDs (Top 6)

Certified

- IFC2x3 Coordination View 2.0
- IFC2x3 Coordination View
- IFC2x2 Singapore BCA e-Plan Check

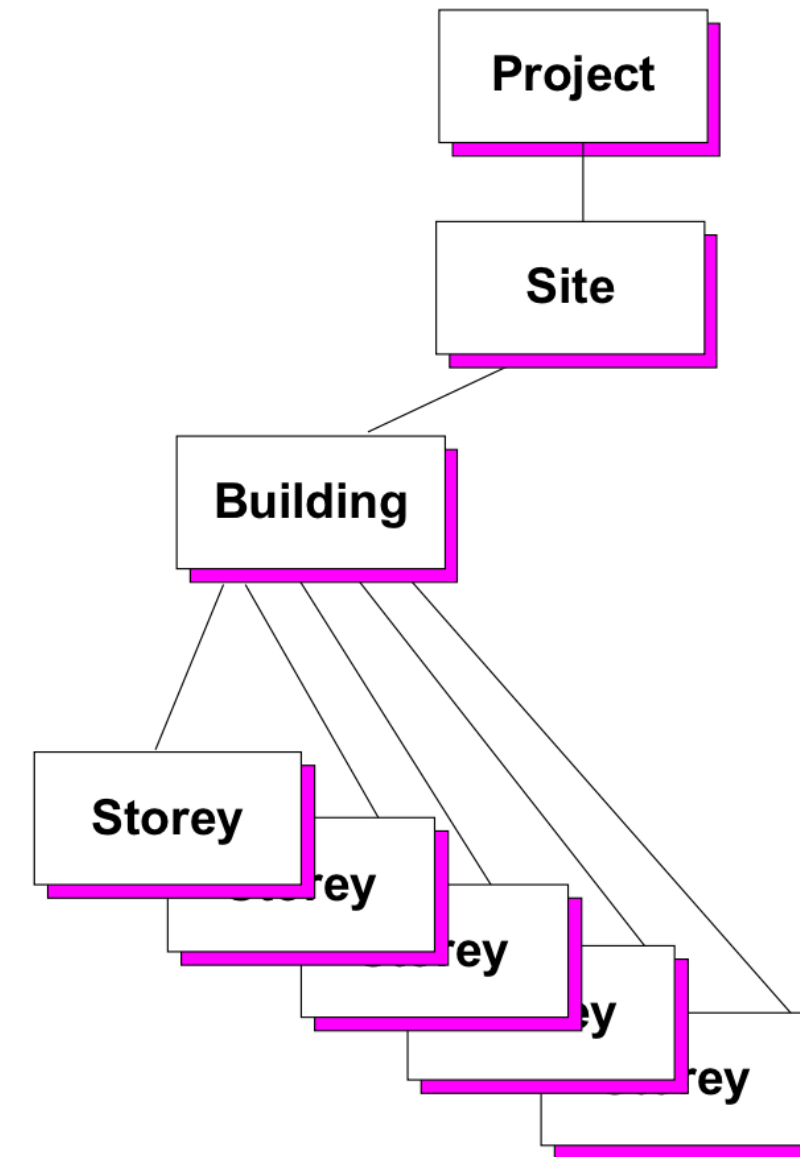
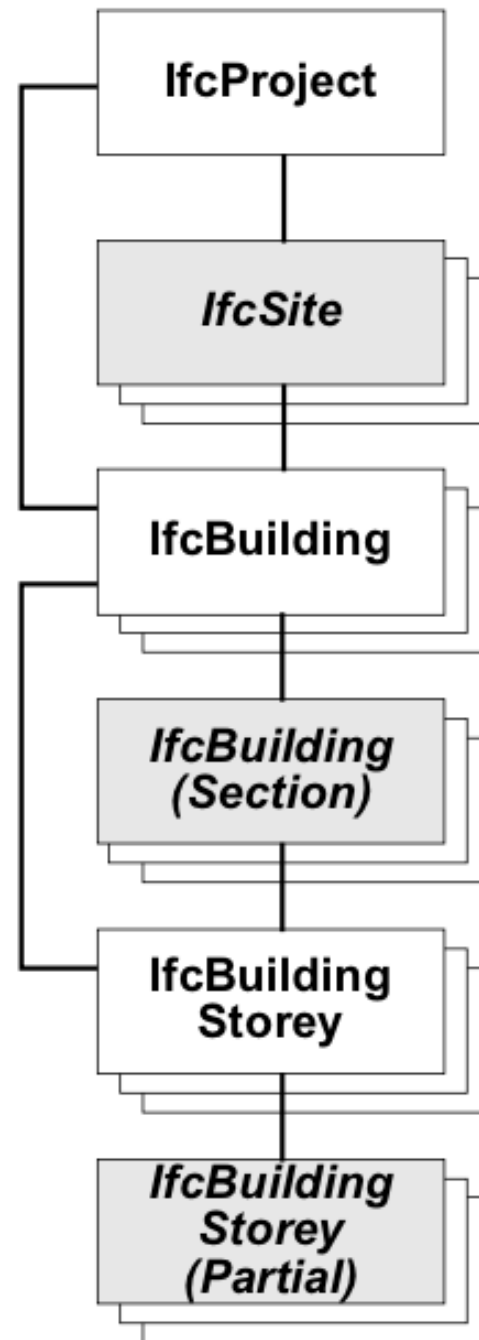


Not Certified

- IFC2x3 (Extended) FM Handover View (COBie)
- IFC4 Reference View
- IFC4 Design Transfer View

Introduction to IFC4

IFC top level structure



IFC top level format to Revit comparison

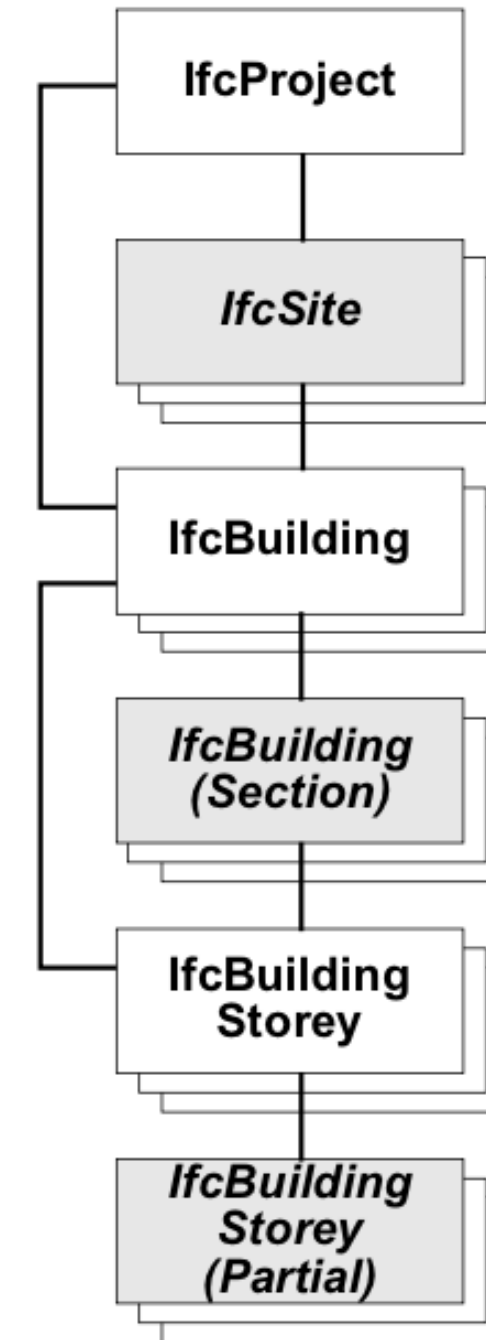
IfcProject = Revit Document

IfcSite = Revit Site (if it has any geometry)

IfcBuilding = Revit parameters in Project Information

IfcBuildingStorey = Revit Levels

IfcBuildingElements = Revit Elements



IfcElement hierarchy



ENTITY IfcWall;

ENTITY IfcRoot;

GlobalId : IfcGloballyUniqueId;

OwnerHistory : IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition;

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

IsDecomposedBy : **SET OF** IfcRelDecomposes **FOR** RelatingObject;

Decomposes : **SET** [0:1] **OF** IfcRelDecomposes **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject;

ObjectType : **OPTIONAL** IfcLabel;

INVERSE

IsDefinedBy : **SET OF** IfcRelDefines **FOR** RelatedObjects;

ENTITY IfcProduct;

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;

Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement;

Tag : **OPTIONAL** IfcIdentifier;

What's new in IFC4?

- General cleanup of IFC entity names and fields
 - IfcDoorStyle -> IfcDoorType, like all the other types
 - All elements have a PredefinedType, called “PredefinedType”
 - All elements have a entity : entity type association (especially MEP elements)
- Better support for topography and geolocation
- More “StandardCase” entities to encourage better import
- Much better support for generic geometry
 - IfcAdvancedBRep

IFC4 Reference View MVD

From <http://www.buildingsmart-tech.org/specifications/ifc-view-definition/ifc4-coordination-views/ifc4-cvs-summary>:

The overall goal of the Reference View is to provide building information that may be consumed by the widest array of software applications that do not require modifying geometry. Such applications enable viewing, estimating, building, operating, and other downstream analysis.

Reference View MVD <-> Revit Link IFC

Why IFC4 Reference View?

- Use it when you need to send information to a browser that doesn't support IFC4 Design Transfer View.
 - Otherwise: Don't bother.
- But what about:

Reference View MVD <-> Revit Link IFC?

- Use Design Transfer View for smaller files and better images.

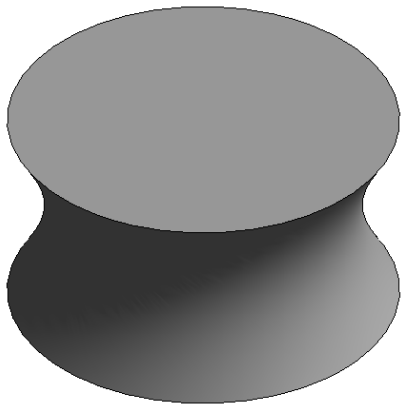
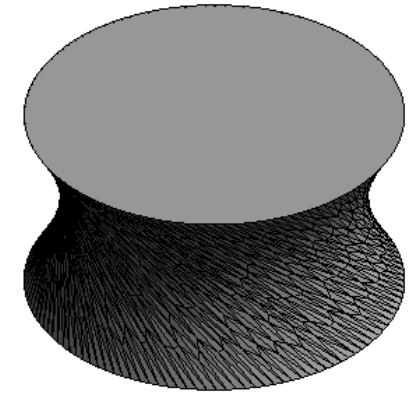
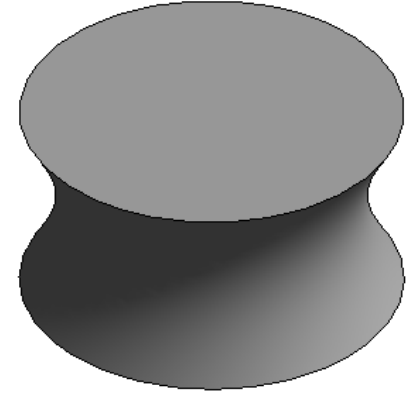
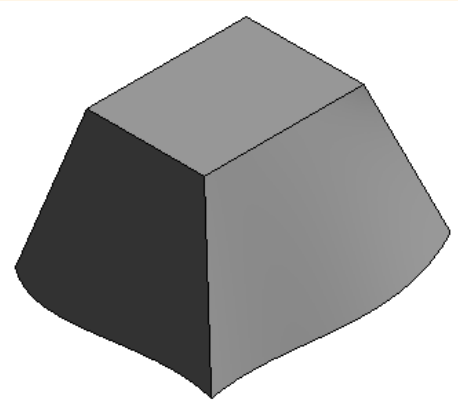
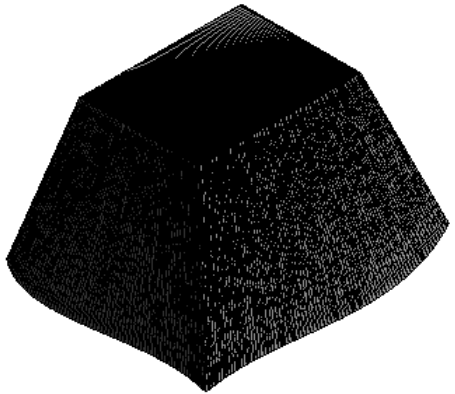
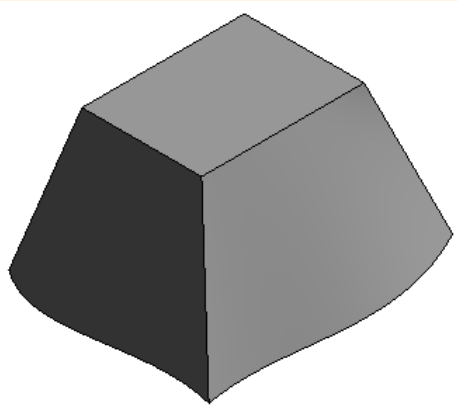

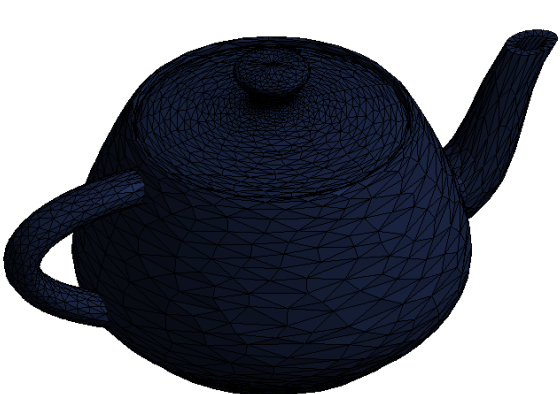

IFC4 Design Transfer View MVD

From <http://www.buildingsmart-tech.org/specifications/ifc-view-definition/ifc4-coordination-views/ifc4-cvs-summary>:

The overall goal of the Design Transfer View is to provide building information with support for editing of interconnected elements. Such applications enable inserting, deleting, moving, and modifying physical building elements and spaces. The target scenario is an architect providing building design information to an engineer for a particular discipline, where geometric modifications may need to be made.

Design Transfer View MVD <-> Revit Open IFC

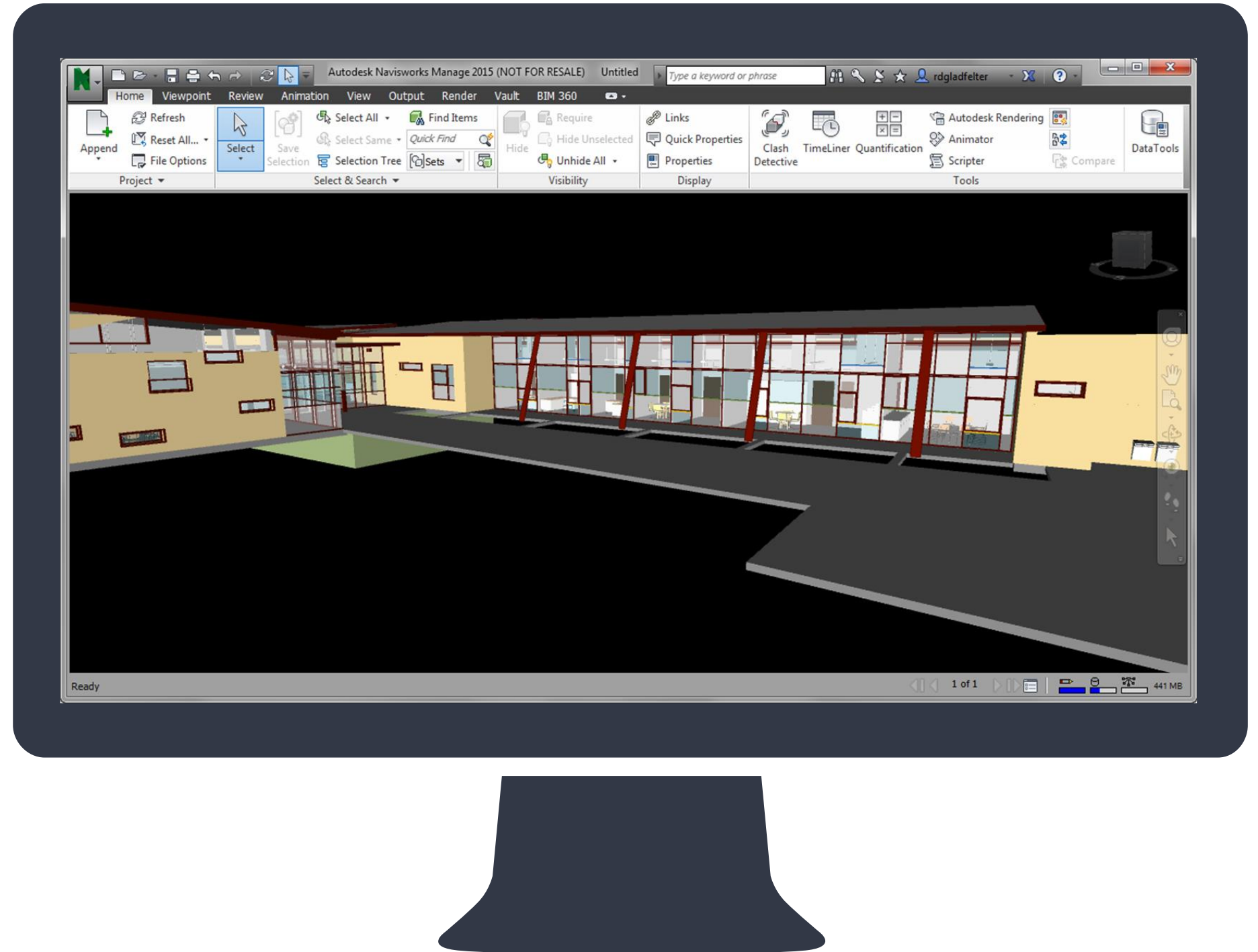
Why IFC4 Design Transfer View?






Object	Revit	IFC2x3 CV 2.0	IFC4 DTV
Hyperboloid			
Blended Solid			
Teapot			

IFC in Autodesk

Support of IFC Standard

	Autodesk Revit
	Autodesk Inventor
	Autodesk Navisworks



	AutoCAD Architecture/MEP
	Autodesk InfraWorks & Civil 3D
	Autodesk A360 & BIM 360 Glue
	Autodesk Robot Structural Analysis
	Autodesk Advance Steel
	Fabrication CADmep

IFC 2x3 Coordination View 2.0 Certifications

Autodesk Revit



ACA



In progress: Advance Steel, Navisworks Manage

<http://buildingsmart.org/compliance/certified-software/>

<http://www.buildingsmart-tech.org/certification/ifc-certification-2.0/ifc2x3-cv-v2.0-certification/participants>

IFC 4 Certifications

IFC 4 Certifications

- No certification yet
 - Beta in progress
 - Reference View certification to begin – any day
 - Export first, then import
 - Design Transfer View certification delayed



Autodesk IFC4 Support

- Revit:
 - IFC4 Reference View: awaiting certification
 - Full round-trip support in Revit 2015+ via Link IFC
 - IFC4 Design Transfer View: in progress
 - Full round-trip support in Revit 2017 via Link IFC
 - Export IfcAdvancedBRep
 - New MEP entities supported (e.g., IfcAirTerminal)
- Navisworks
 - 2017.1: same IFC4 support as Revit (import only)

Should I use IFC2x3 or IFC4?

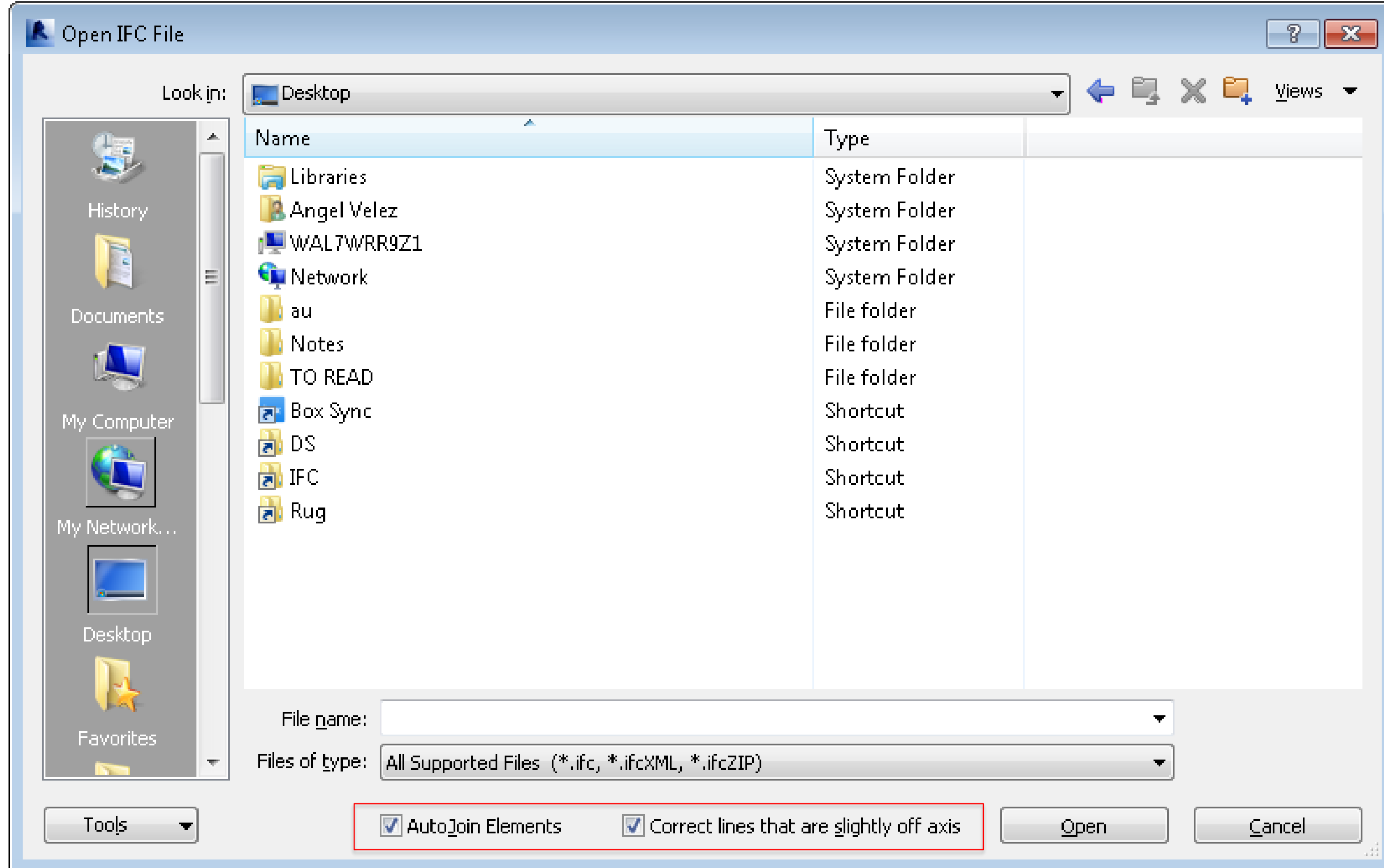
- Answer: Yes.
- Use IFC2x3:
 - If you are using an older version of software that doesn't support IFC4
 - If you need to use only certified MVDs
 - For archival purposes
- Use IFC4:
 - To generate smaller, better output
 - To give us feedback to fix issues before you are required to use it for new projects

Revit IFC: A Quick Guide

Revit IFC Overview

- Import IFC
 - Link IFC
 - Preferred for active collaboration
 - Emphasize fidelity and performance
 - IFC file read-only for reference
 - Open IFC
 - Preferred for one time transfer of data
 - Emphasize ability to edit over fidelity and performance
 - UI needs updating; will discuss current state (including hacks!)
- Export IFC
 - “Alternate” UI now standard as of Revit 2017

Revit Open IFC



Revit Open IFC UI Options

- AutoJoin Elements
 - Checked: Create joins for all elements that would automatically join to each other when created or modified (e.g. walls, lines)
 - Get same functionality as when creating Revit objects via UI
 - Unchecked: Skip this part of the import
 - Generally faster import
 - May avoid wall join problems

Revit Open IFC UI Options

- Correct lines that are slightly off axis
 - Checked: Modify imported lines to avoid Revit warnings
 - Avoid annoying “slightly off axis” warnings
 - Unchecked: Leave imported lines in original position
 - Minimize change in intent

Import IFC Options

Import IFC Options

Default Template for IFC Import:

Import IFC Class Mapping:

IFC Class Name	IFC Type	Revit Category	Revit Sub-Category
IfcAirTerminal		Air Terminals	
IfcAirTerminalType		Air Terminals	
IfcAnnotation		Generic Annotations	
IfcBeam		Structural Framing	
IfcBeamType		Structural Framing	
IfcBoiler		Mechanical Equipment	
IfcBoilerType		Mechanical Equipment	
IfcBuildingElementPart		Parts	
IfcBuildingElementPartType		Parts	
IfcBuildingElementProxy		Generic Models	
IfcBuildingElementProxyType		Generic Models	
IfcCableCarrierFitting		Cable Tray Fittings	
IfcCableCarrierFittingType		Cable Tray Fittings	
IfcCableCarrierSegment		Cable Trays	
IfcCableCarrierSegmentType		Cable Trays	
IfcColumn		Columns	
IfcColumn	[LoadBearing]	Structural Columns	
IfcColumnType		Columns	
IfcColumnType	[LoadBearing]	Structural Columns	
IfcController		Specialty Equipment	
IfcControllerType		Specialty Equipment	
IfcCovering		Generic Models	
IfcCovering	CEILING	Ceilings	

Import IFC Options

- Default template for IFC import
 - Open/Link use your default template if this isn't set
 - Populate with materials, shared parameters, etc.
 - Works equally well with Open and Link IFC.

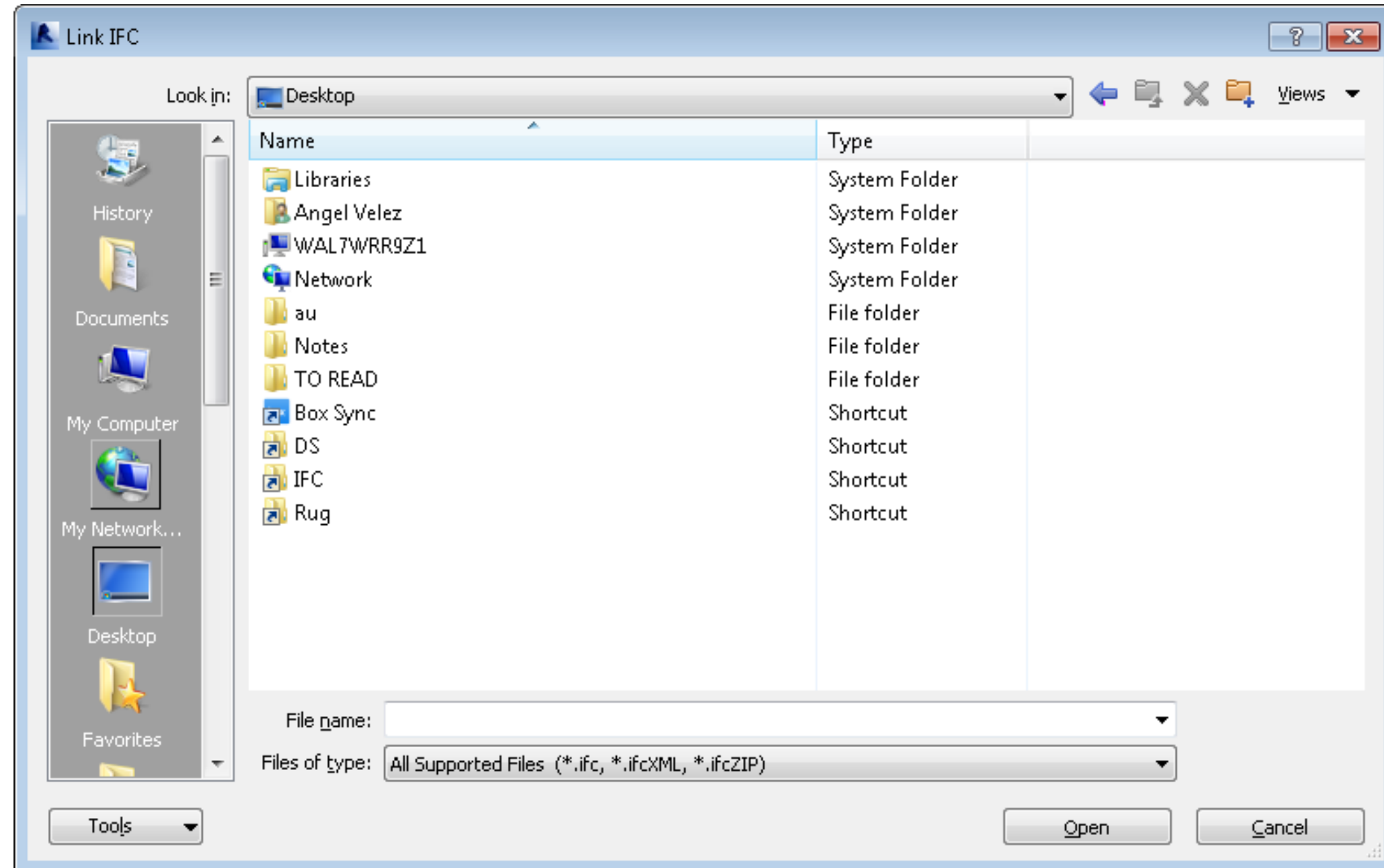
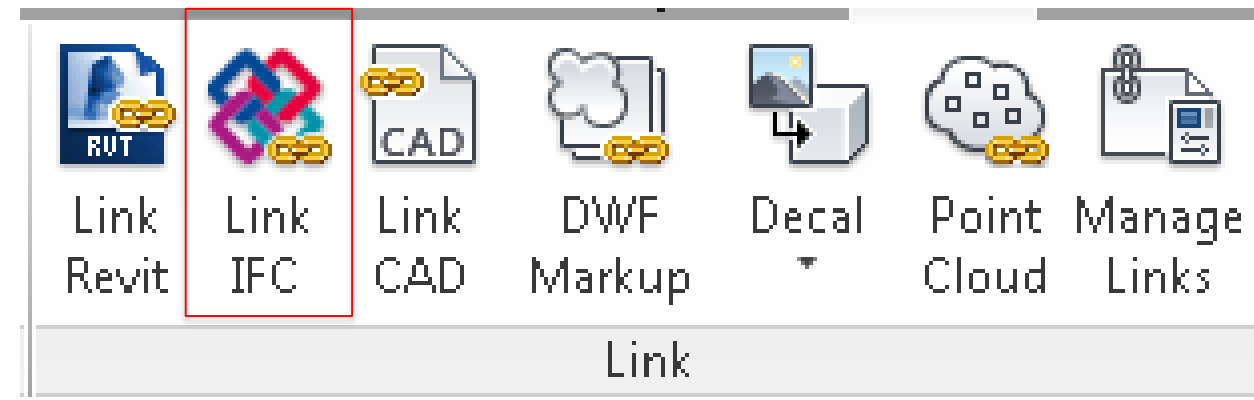
Import IFC Options

- Import IFC class mapping file
 - The file used to populate the default IFC entity to Revit category mappings
 - May be ignored if Revit thinks it knows better (e.g. IfcWall always creates a Revit wall)
 - Has some special entries (e.g. IfcColumn + LoadBearing)
 - “Don’t Import” – ignore this IFC entity
 - Only works for Open IFC

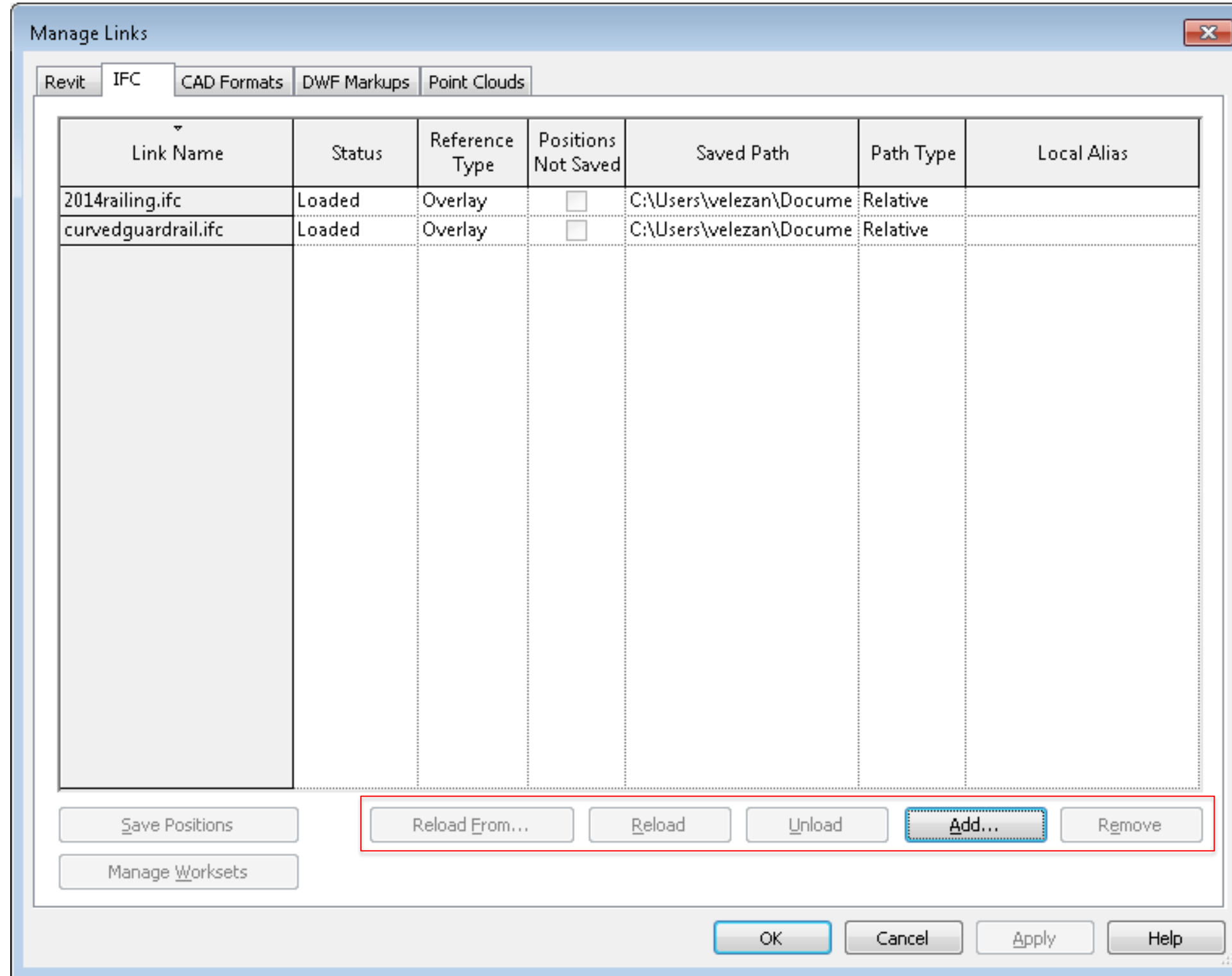
Import IFC Options

- Import IFC class mapping file (part 2)
 - Use the “Standard” button to reset the mapping file...
 - But first, delete the existing file on disk
 - (We’ll see this trick again on the export side)

Revit Link IFC



Revit Link IFC – Option 2



Revit Link IFC

- Link IFC keeps associativity with original IFC file
 - Checks date on open IFC, updates cache if necessary
 - Will try to maintain references if possible.
 - “Reload” in Manage Links allows for forced reload
 - ... but only if the IFC cache is out of date
 - Want to force a reload? Delete the .ifc.RVT file.
 - “Reload from” changes the associated IFC file
 - Didn't work right? Delete the .ifc.RVT file.

Export IFC - 2017

Export IFC

File name:

C:\Users\velezan\Desktop\Project1.ifc

Browse ...

Current selected setup:

<In-Session Setup>

Modify setup ...

IFC Version:

IFC 2x3 Coordination View 2.0

Projects to export:

☒ Project1

[How do I specify an export setup?](#)

Export

Cancel

Export IFC - 2017

Modify Setup

<In-Session Setup>
<IFC2x3 Coordination View 2.0 Setup>
<IFC2x3 Coordination View Setup>
<IFC2x3 GSA Concept Design BIM 2010 Setup>
<IFC2x3 Basic FM Handover View Setup>
<IFC2x2 Coordination View Setup>
<IFC2x2 Singapore BCA e-Plan Check Setup>
<IFC2x3 Extended FM Handover View Setup>
<IFC4 Reference View Setup>
<IFC4 Design Transfer View Setup>

General Additional Content Property Sets Level of Detail Advanced

IFC version IFC 2x3 Coordination View 2.0

File type IFC

Phase to export Default phase to export

Space boundaries None

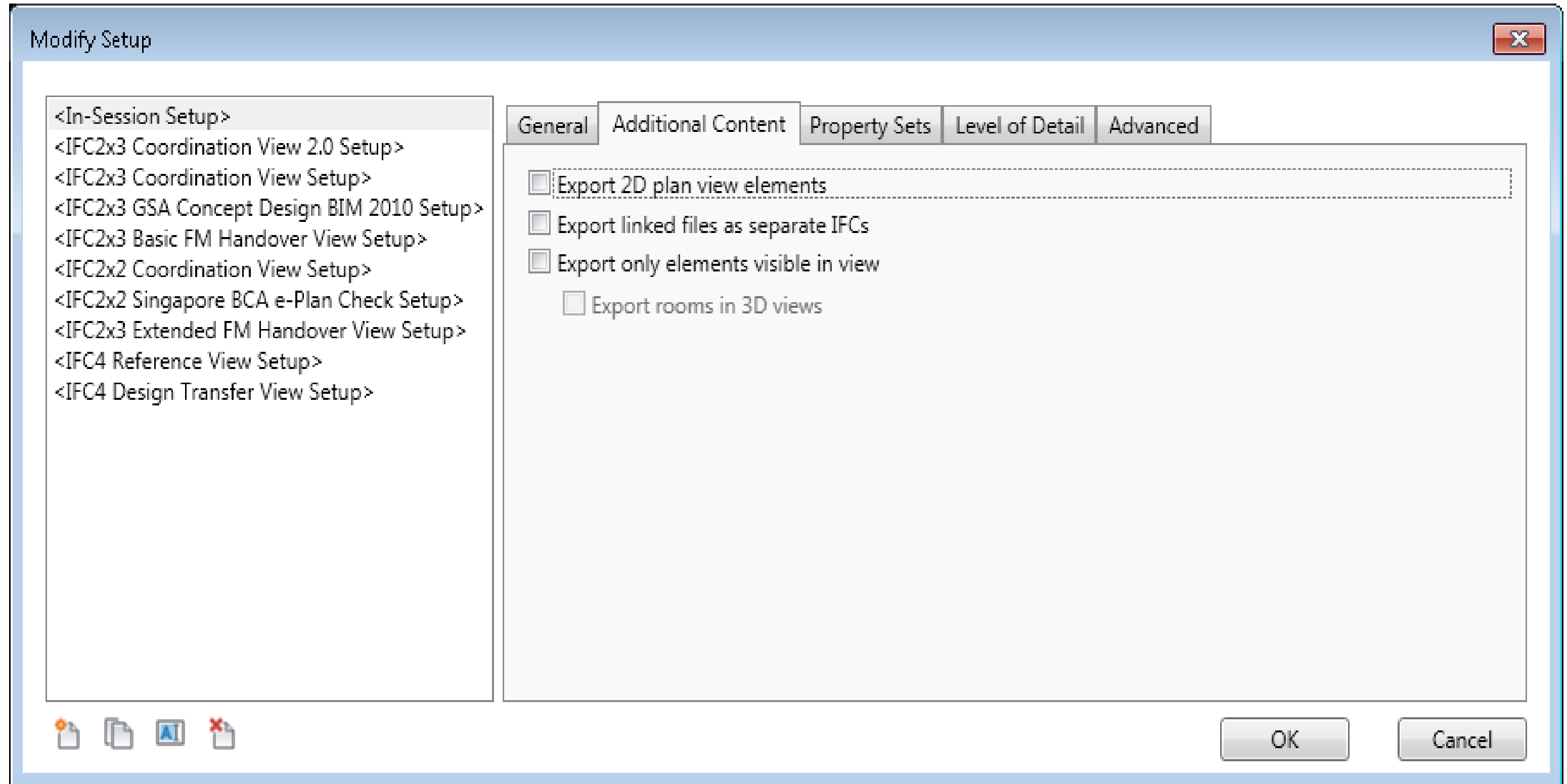
☐ Split Walls, Columns, Ducts by Level

File Header Information...

Project Address...

OK Cancel

Export IFC - 2017



Export IFC - 2017

Modify Setup

<In-Session Setup>
<IFC2x3 Coordination View 2.0 Setup>
<IFC2x3 Coordination View Setup>
<IFC2x3 GSA Concept Design BIM 2010 Setup>
<IFC2x3 Basic FM Handover View Setup>
<IFC2x2 Coordination View Setup>
<IFC2x2 Singapore BCA e-Plan Check Setup>
<IFC2x3 Extended FM Handover View Setup>
<IFC4 Reference View Setup>
<IFC4 Design Transfer View Setup>

General Additional Content Property Sets Level of Detail Advanced

☐ Export Revit property sets

☒ Export IFC common property sets

☐ Export base quantities

☐ Export schedules as property sets
☐ Export only schedules containing IFC, Pset, or Common in the title

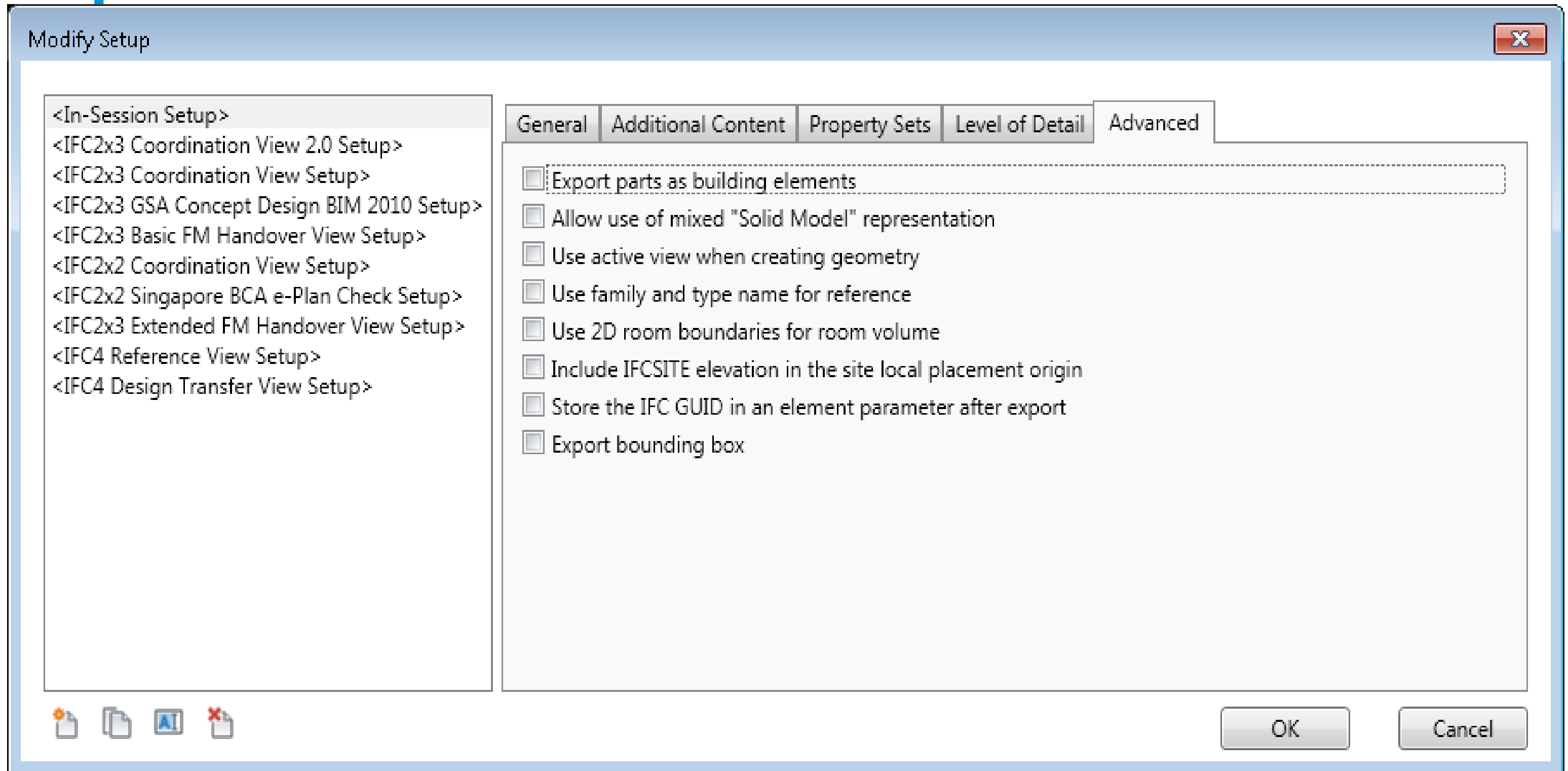
☐ Export user defined property sets
E:\Revit\dev\Debugx64\AddIns\IFCExporterUI\DefaultUserDefinedParameterSets.t Browse ...

☐ Export parameter mapping table
Browse ...

Classification Settings...

OK Cancel

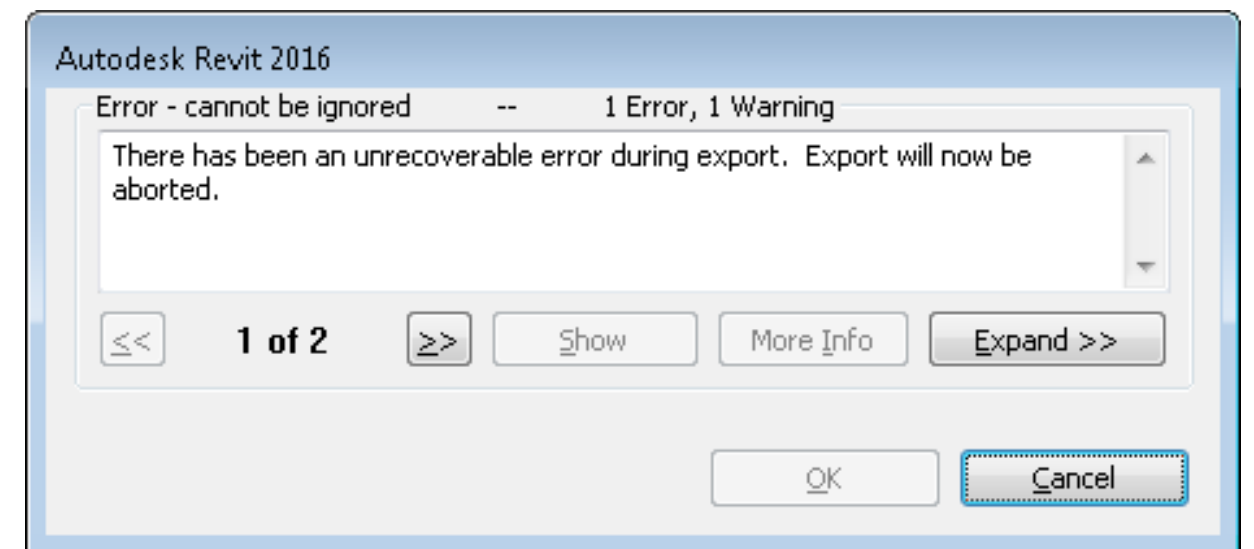
Export IFC - 2017



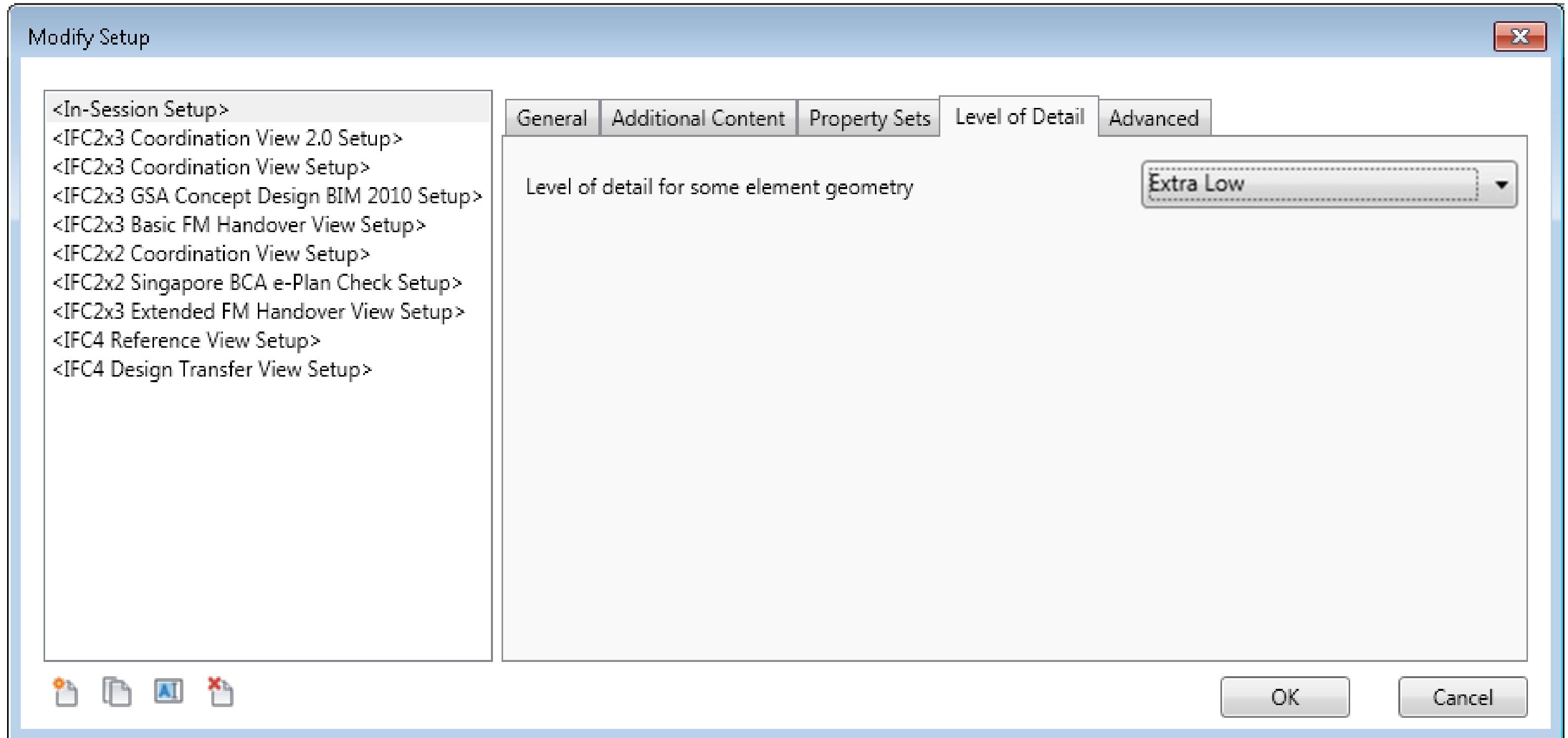
Help, my file won't export! (It's too big.) - Old

- Export only elements visible in view
 - Even better: use coarse or medium view
- Don't export space boundaries if you don't need them
- Don't export property sets if you don't need them
- Use Advanced "Solid Model" representation
- Split file as last resort
 - Look for long, thin elements or elements with many openings/reveals

- Long term solutions
 - New IFC toolkit with no size limitations
 - IFC4 Design Transfer View

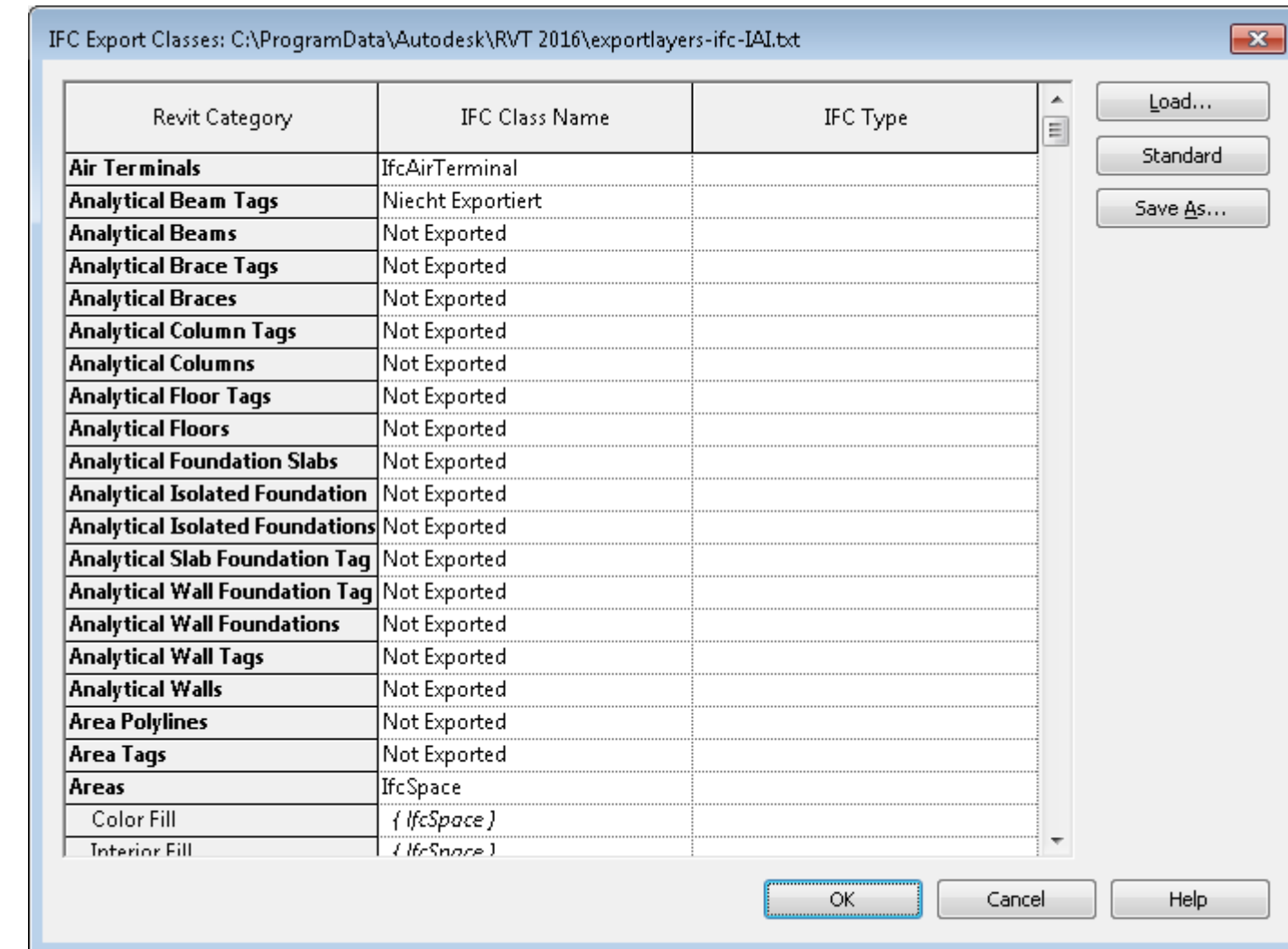


Help, my file won't export! (It's too big.) - New



Help, my file won't export! (It isn't too big...)

- Latest version of IFC?
 - Check your mapping file.
 - Is everything “Not Exported”?
 - Different language OS?
 - If so:
 - Delete the file on disk.
 - Press the “Standard” button.
 - Make sure sub-categories are exported
-
- All else fails: SourceForge and E-mail
 - Many problems can be solved by an open source update.



Revit IFC DIY

Why open source?

- Customer flexibility and customization
- Countrification
- Updates independent of Revit's release cycle
- Allow for outside contributions

Modification



IFC for Revit

This is the .NET code for the Revit 2012-2016 IFC open source.

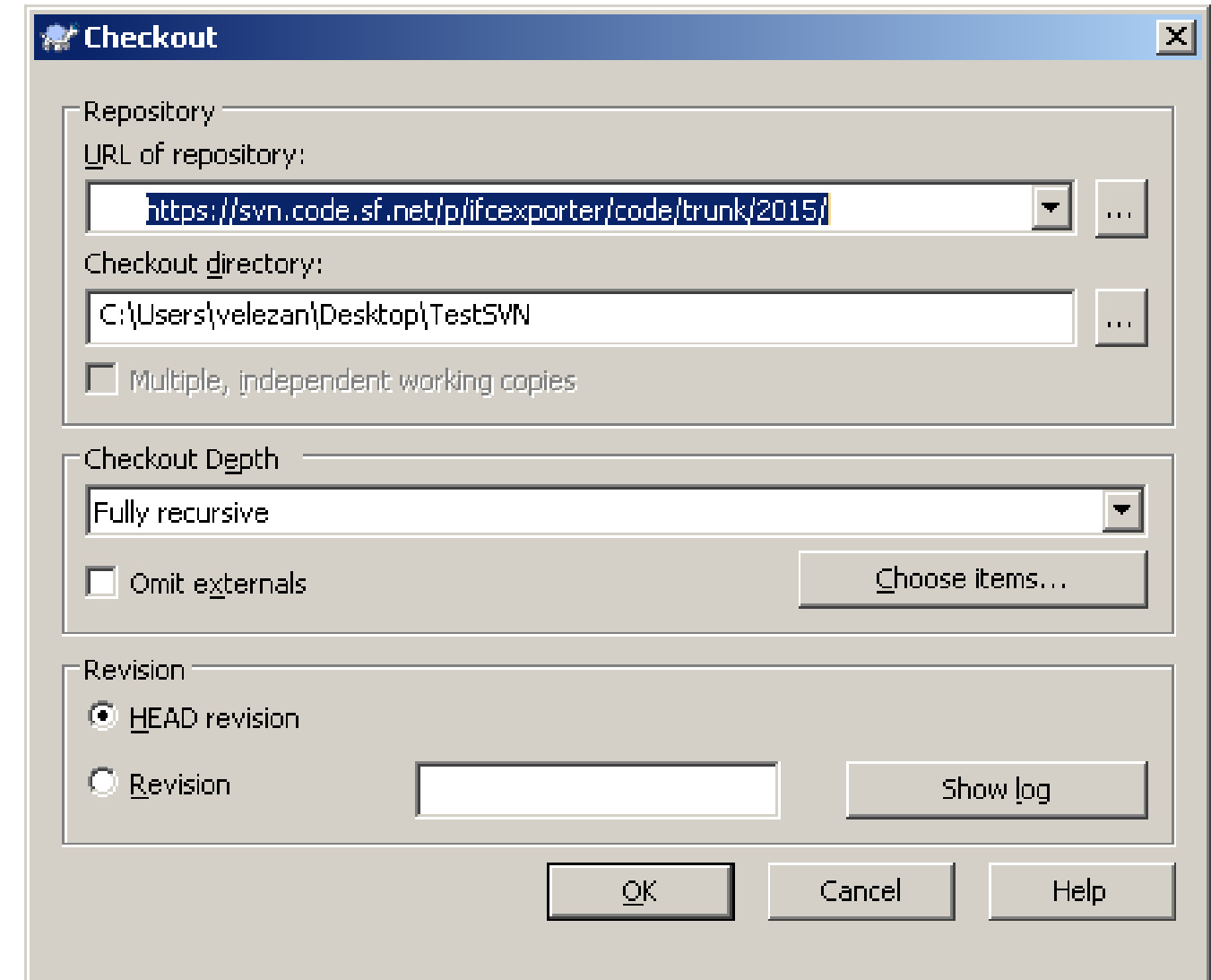
Brought to you by: [angelvelezsosa](#), [aparella](#), [ekfour](#), [jmli1011](#), and 2 others

- Need Subversion (SVN) to download and modify source code
- ZIP file included for those without some version of SVN
- All local modifications allowed, uploading is restricted
- Contributions controlled by Open Source steering committee
- All contributions are welcome!



Download the code

- Step-by-step details in class hand-out
 - New: Wizard available
- Create a directory for the source code
- Checkout the code
- Update the project files
- Optional: change the version number



What is (and isn't) the open source

- The open source includes:
 - Export IFC
 - Link IFC
 - Top-level call to Open IFC native code
- The open source doesn't include (yet):
 - Some export code that hasn't been ported to open source
 - Open IFC
- All of Link IFC, minus the hooks into Revit, are in the open source



Modify the exporter: add a property set – old

6.4.4.9 Pset_ManufacturerTypeInfoInformation

PSET_TYPEDRIVENOVERRIDE / IfcElement

EN Defines characteristics of types (ranges) of manufactured products that may be given by the manufacturer. Note that the term 'manufactured' may also be used to refer to products that are supplied and identified by the supplier or that are assembled off site by a third party provider. HISTORY: This property set replaces the entity IfcManufacturerInformation from previous IFC releases. IFC 2x4: AssemblyPlace property added.

[buildingSMART Data Dictionary](#)



▪ GlobalTradeItemNumber

P_SINGLEVALUE / IfcIdentifier

EN GlobalTradeItemNumber: The Global Trade Item Number (GTIN) is an identifier for trade items developed by GS1 (www.gs1.org).

▪ ArticleNumber

P_SINGLEVALUE / IfcIdentifier

EN ArticleNumber: Article number or reference that is be applied to a configured product according to a standard scheme for article number definition as defined by the manufacturer. It is often used as the purchasing number.

▪ ModelReference

P_SINGLEVALUE / IfcLabel

EN ModelReference: The model number or designator of the product model (or product line) as assigned by the manufacturer of the manufactured item.

▪ ModelLabel

P_SINGLEVALUE / IfcLabel

EN ModelLabel: The descriptive model name of the product model (or product line) as assigned by the manufacturer of the manufactured item.

▪ Manufacturer

P_SINGLEVALUE / IfcLabel

EN Manufacturer: The organization that manufactured and/or assembled the item.

▪ ProductionYear

P_SINGLEVALUE / IfcLabel

EN ProductionYear: The year of production of the manufactured item.

▪ AssemblyPlace

P_ENUMERATEDVALUE / IfcLabel / PEnum_AssemblyPlace: FACTORY, OFFSITE, SITE, OTHER, NOTKNOWN, UNSET

EN AssemblyPlace: Enumeration defining where the assembly is intended to take place, either in a factory or on the building site.

```
private static void InitPSMTypeInfoInformation(ICollection<PropertySetDescription>
commonPropertySets)
{
    PropertySetDescription psm = new PropertySetDescription();
    psm.Name = "Pset_ManufacturerTypeInfoInformation";
    psm.EntityTypes.Add(IFCEntityType.IfElement);
    psm.AddEntry(PropertySetEntry.CreateIdentifier("ArticleNumber"));
    psm.AddEntry(PropertySetEntry.CreateLabel("ModelReference"));
    psm.AddEntry(PropertySetEntry.CreateLabel("ModelLabel"));
    PropertySetEntry ifcPSE =
PropertySetEntry.CreateLabel("Manufacturer");
    ifcPSE.RevitBuiltInParameter =
        BuiltInParameter.ALL_MODEL_MANUFACTURER;
    psm.AddEntry(ifcPSE);
    psm.AddEntry(PropertySetEntry.CreateLabel("ProductionYear"));

    if (ExportSchema == IFCVersion.IFC4)
    {
        psm.AddEntry(
            PropertySetEntry.CreateIdentifier("GlobalTradeItemNumber"));
        psm.AddEntry(
            PropertySetEntry.CreateEnumeratedValue("AssemblyPlace",
            PropertyType.Label,
            typeof(Toolkit.IFC4.
                PsetManufacturerTypeInfoInformation_AssemblyPlace)));
    }
    commonPropertySets.Add(psm);
}
```

Modify the exporter: add a property set – new

6.4.4.9 Pset_ManufacturerTypeInfo

PSET_TYPEDRIVENOVERRIDE / IfcElement

EN Defines characteristics of types (ranges) of manufactured products that may be given by the manufacturer. Note that the term 'manufactured' may also be used to refer to products that are supplied and identified by the supplier or that are assembled off site by a third party provider. HISTORY: This property set replaces the entity IfcManufacturerInformation from previous IFC releases. IFC 2x4: AssemblyPlace property added.

[buildingSMART Data Dictionary](#)



- **GlobalTradeItemNumber**
P_SINGLEVALUE / IfcIdentifier
EN GlobalTradeItemNumber: The Global Trade Item Number (GTIN) is an identifier for trade items developed by GS1 (www.gs1.org).
- **ArticleNumber**
P_SINGLEVALUE / IfcIdentifier
EN ArticleNumber: Article number or reference that is be applied to a configured product according to a standard scheme for article number definition as defined by the manufacturer. It is often used as the purchasing number.
- **ModelReference**
P_SINGLEVALUE / IfcLabel
EN ModelReference: The model number or designator of the product model (or product line) as assigned by the manufacturer of the manufactured item.
- **ModelLabel**
P_SINGLEVALUE / IfcLabel
EN ModelLabel: The descriptive model name of the product model (or product line) as assigned by the manufacturer of the manufactured item.
- **Manufacturer**
P_SINGLEVALUE / IfcLabel
EN Manufacturer: The organization that manufactured and/or assembled the item.
- **ProductionYear**
P_SINGLEVALUE / IfcLabel
EN ProductionYear: The year of production of the manufactured item.
- **AssemblyPlace**
P_ENUMERATEDVALUE / IfcLabel / PEnum_AssemblyPlace: FACTORY, OFFSITE, SITE, OTHER, NOTKNOWN, UNSET
EN AssemblyPlace: Enumeration defining where the assembly is intended to take place, either in a factory or on the building site.

PropertySet: Pset_ManufacturerTypeInfo			T	IfcElementType
GlobalTradeItemNumber	Identifier			
ArticleNumber	Identifier			Number
ModelReference	Label			
ModelLabel	Label			
Manufacturer	Label			
ProductionYear	Label			
AssemblyPlace	Label			

- No programming: just modify a text file
 - To do: add UI to make even simpler
- Some limitations:
 - Can't specify calculated values
 - No checking of enumerated values
 - Very picky about formatting



DIY Export

Instead of modifying the exporter directly, a user can create an exporter based on the existing IFC exporter. To do this, you will need to do the following:

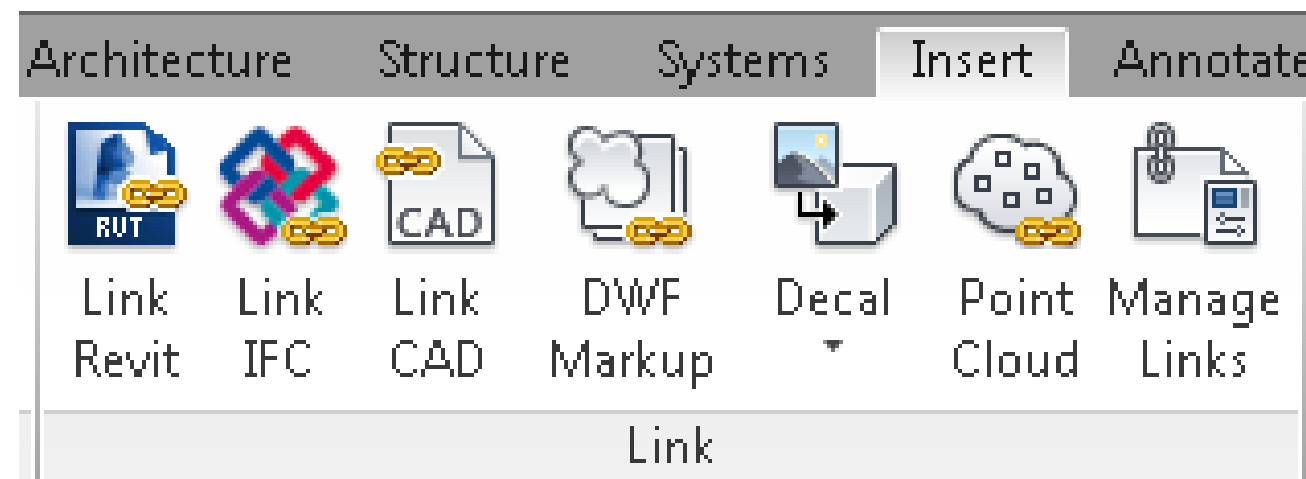
- Create a new ExporterApplication class in your custom workspace
- Create a new Exporter class, to override the base exporter
- Override virtual functions as necessary
- Details are in the hand-out

Export customization ideas

- Support for a new IFC entity from data accessible via Revit API
- Support for elements that are non-standard IFC entities
- Support for extended properties for materials
- Support for non-geometric data gathered by custom UI or extensible storage (e.g. file header, user information, zones)
- Support for additional UI options to choose between different export needs

Overview of the import code

- Import Registered as a service like the export code
 - Revit.IFC.Import/Importer.cs
- Uses same Import IFC class settings as Open IFC
- Link and Open IFC both go through the Open Source



Overview of the link code, pt. 2

- Code path dependent on the import intent and action
 - IFCImportIntent: Parametric vs. Reference
 - IFCImportAction: Open vs. Link
 - Link IFC = IFCImportAction.Link + IFCImportIntent.Reference
 - Open IFC = IFCImportAction.Open + IFCImportIntent.Parametric
 - Open IFC branch calls ProcessIFCProject to do majority of work
- Two steps: Process and Create
 - Managed by the static IFCImportFile.Create function

Open IFC

- Open IFC still implemented in native code
- But (from before):
 - The open source includes top-level call to Open IFC native code
 - Meaning: internal Open IFC can be overridden
- Limitations:
 - Some element creation API in better shape than others
 - Example: not possible to create in-place family via API
 - Some element interactions difficult to control via API
 - Example: wall joins
- Intention is to move Open IFC into open source

Create

- Create() function works on data created in Process() step
- Create() function may create different data dependent on the import intent and action
 - Currently only covers link case
 - Could create non-Revit data also

Link IFC Requirements for Created Elements

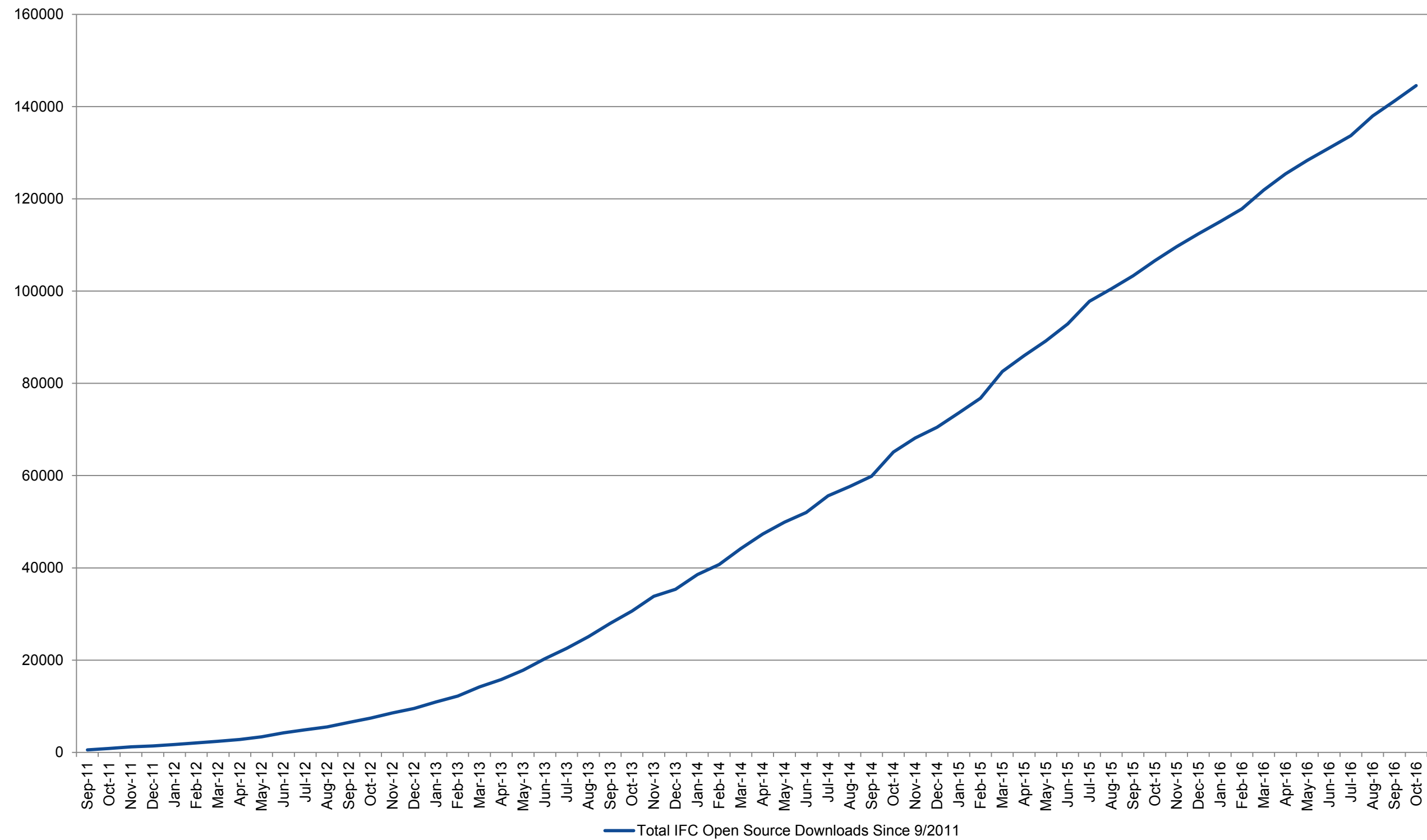
- Fast
- High fidelity
- Maintain parameter data
- Proper category
- Proper materials
- Read-only
- Maintain associativity with IFC file

DirectShapes

- DirectShape element satisfies those requirements
 - API only
 - Maintains parameter data
 - Settable category
 - Settable materials
 - Read-only shape
- Less functionality than parametric Revit elements
 - Can be referenced
- Also used for Import SAT/Rhino since 2017.1 release

Conclusion

Open Source: Total Downloads



Autodesk commitment to IFC

- Revit ships with high-quality IFC support
 - Continue to incrementally improve functionality and UI
- Autodesk supports IFC and open standards
 - Active engagement with buildingSMART to develop future MVDs to expand IFC's reach
 - Will continue rapid response to reported issues
 - Will support new workflows

