IFC to SVF2 Translation Method v4

Model Derivative API Whitepaper

AUTODESK
Platform Services

Updated: Oct. 2025

Table of Contents

Document Revision Information

Ver	Date	Amendment	Auother
1.0	2025/10/23	First Version	E.K, Autodesk Developer Advocacy and Support

1. Introduction

This whitepaper explains the IFC to SVF/SVF2 conversion method version 4 (IFC v4), an option used for POST Create Translation Job in Model Derivative API, with special emphasis on changes and enhancements since the previous version 3.

Background

Autodesk Platform Services (APS) is an API platform that Autodesk modulates its own core technologies and releases them as a set of Web API. To enable our customers to utilize web technologies such as HTML5, JavaScript, RESTful APIs, and GraphQL, making it easy to create new, creative applications like LEGO blocks.

One of its flagship API services, the Model Derivative API, enables our customers to convert over 70 file formats for viewing, extract and filter data, and, with any browser, view 2D and 3D drawings and models that include data and geometry. The Industry Foundation Classes (IFC) are also one of its supported formats, enabling our customers working with Open BIM to integrate their workflow with Autodesk technologies. To process IFC source files, the Model Derivative API provides four existing IFC conversion methods: legacy, modern, v3, and v4. In this white paper, we use the “IFC v4” as an abbreviation for the IFC conversion method v4, and a similar one, “IFC v3”, for the IFC conversion method v3, which are NOT the IFC schema versions, such as IFC2x3, IFC4, or IFC4x3.

IFC v4 is the latest IFC to SVF/SVF2 conversion method. To continue embracing Open BIM and help our customers reduce the complexities of working with IFC files using Autodesk technologies, we continually evolve, enhance, and improve our IFC conversion method. Additionally, to reduce the complexities of maintaining the IFC conversion methods across Autodesk products and services, we streamlined the architecture into a unified approach in IFC v4.

Unlike previous versions of conversion methods, IFC v4 doesn't utilize or rely on technologies from Navisworks and Revit. The processed results will be closer to the source IFC file.

Currently, IFC v4 is available on the APS Model Derivative API and is the recommended IFC conversion method. In this white paper, we will **highlight the key changes, updates, and enhancements** in IFC v4 and compare them to IFC v3. It will help APS developers working with IFC understand, be aware of them, and learn how to **migrate from v3 to v4**. Note. These changes also apply to IFC models stored on Autodesk Docs and BIM 360 Document Management.

In the following sections, we will cover these topics:

- High-level comparison of technologies between IFC v3 and v4.

- Key changes, updates, and enhancements in IFC v4, e.g.,
    - **Model Hierarchy:** The node name, node location, virtual nodes, etc.
    - **Object properties:** The name of the default property set, IFC Guid, etc.
    - **Geometry Handling:** More IFC4x3 support, large coordinate handling, etc.
    - **Model Derivative Manifest:** The response of GET manifest changes for IFC v4.
- Migration Guides and tips for APS developers using the Viewer SDK and Model Derivative API.

2. High-level Comparison between IFC to SVF/SVF2 Translation Method v3 and v4

The table below shows a high-level comparison between IFC v3 and v4. The table shows the primary technologies used in each IFC conversion method, where Autodesk uses the conversion method, supported IFC scheme versions, and some key information that our customers frequently ask for.

Comparison/Co-nversion Method	**v3**	**v4**
Context	Revit-based IFC conversion method that utilizes Revit techniques integrated with Navisworks.	Streamlined unified architecture without using the techniques of Navisworks and Revit.
Which service uses it as the default?	**Desktop:** Navisworks 2023-2024 **Cloud:** Model Derivative **Saas:** New projects in BIM 360 Document Management and Autodesk Docs created before July 22nd.	**Desktop:** Navisworks 2025 and later. **Cloud:** Model Derivative **Saas:** New projects in BIM 360 Document Management and Autodesk Docs created after July 22nd.
File process speed	Quicker than Modern for large IFC models based on internal tests.	Its performance is as good as v3, but it would be faster than v3 in some cases, based on internal tests.
IFC Schema support	IF2x3, limited IFC4, and partial IFC 4.3	IF2x3, IFC4 and official IFC4.3 support
IFC GUID Field Name	IfcGUID	GlobalId
Default length & property units	Native IFC file units	Native IFC file units
Large Coordinates	No, require using the Large Coordinates Removal tool to patch the source IFC file	Improved handling, which brings better results than v3. There is no need to use the removal tool.
Status	Under Maintenance Mode. No further updates or improvements for it.	Recommended Choice

3. Key Changes and Enhancements in IFC v4

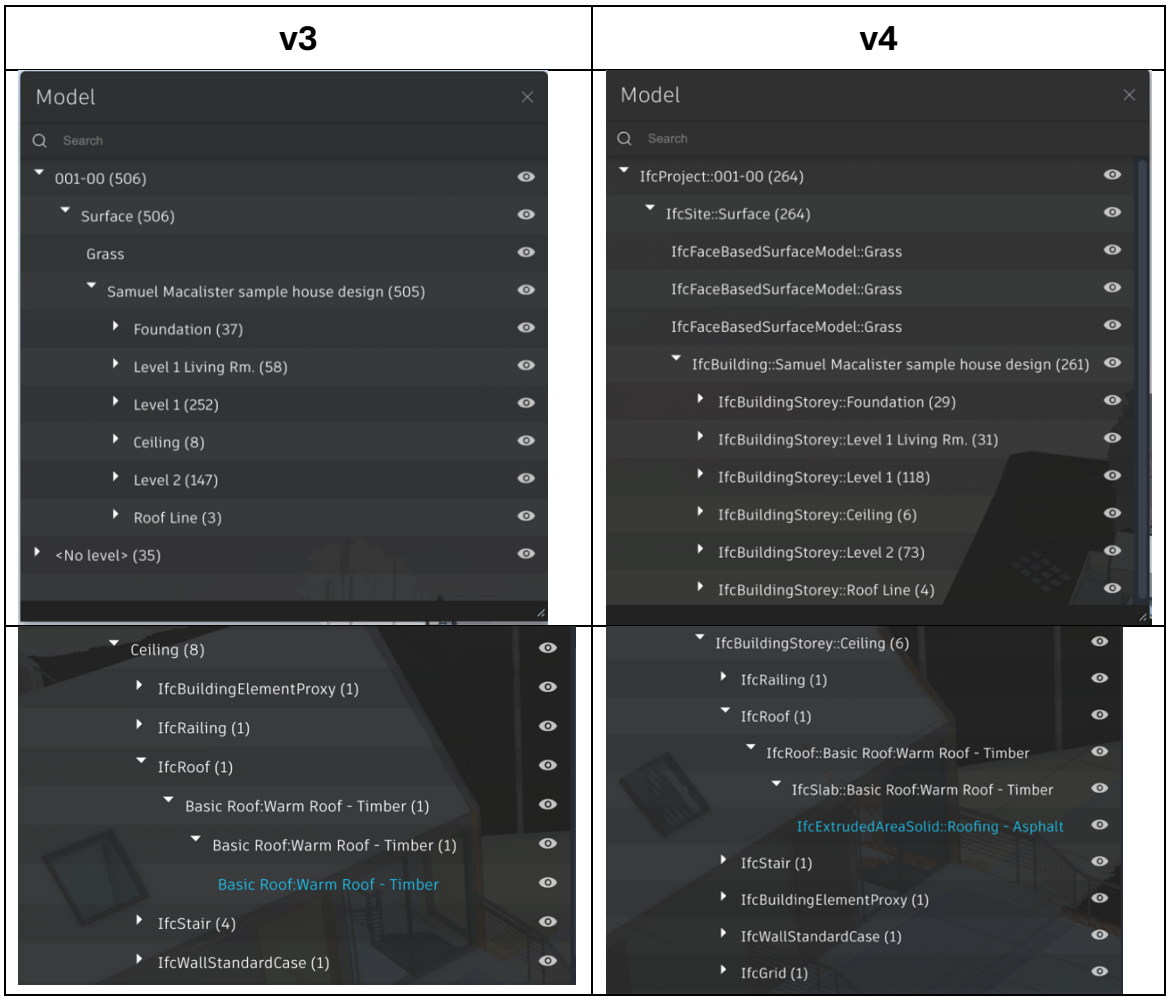
In this section, we will show the key changes and enhancements in hierarchy, properties, and geometries in IFC v4, compared to IFC v3. We will show you the differences using examples.

3.1 Hierarchy Changes and Enhancements

This section will cover changes, updates, or enhancements to the model hierarchy, specifically the model instance tree in the APS Viewer and the response of GET /{urn}/metadata/{modelGuid} of the Model Derivative API.

3.1.1 Node Naming Changes

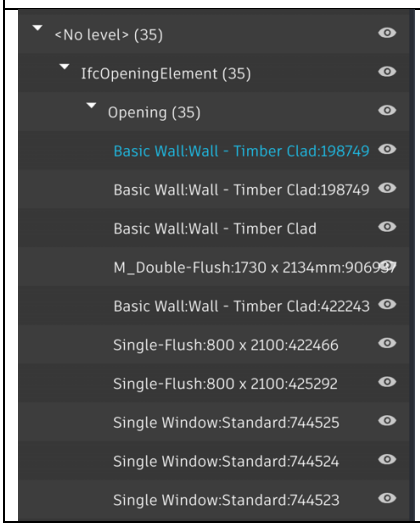
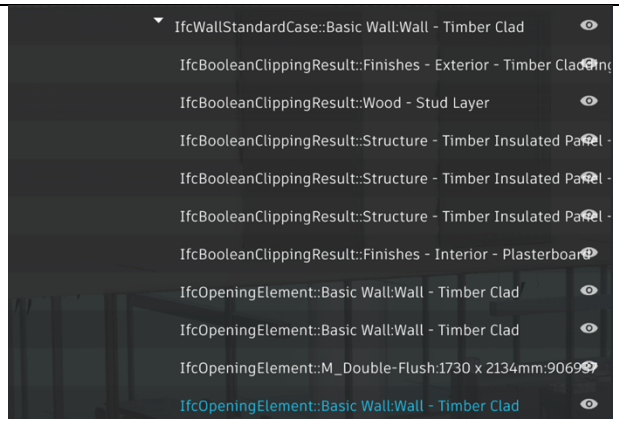
IFC v4 includes the IFC types mapped to node names, facilitating the easy identification of objects.



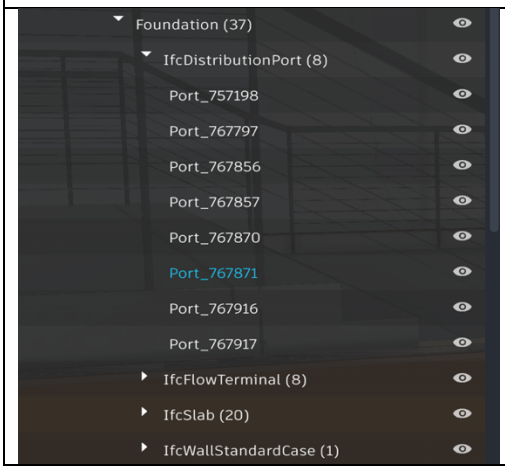
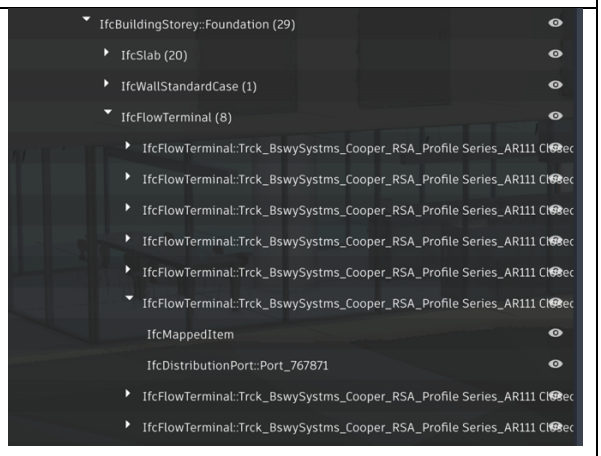
3.1.2 Node Location Reorganization

IFC v4 reorganizes the location of some nodes or node types, compared to IFC v3.

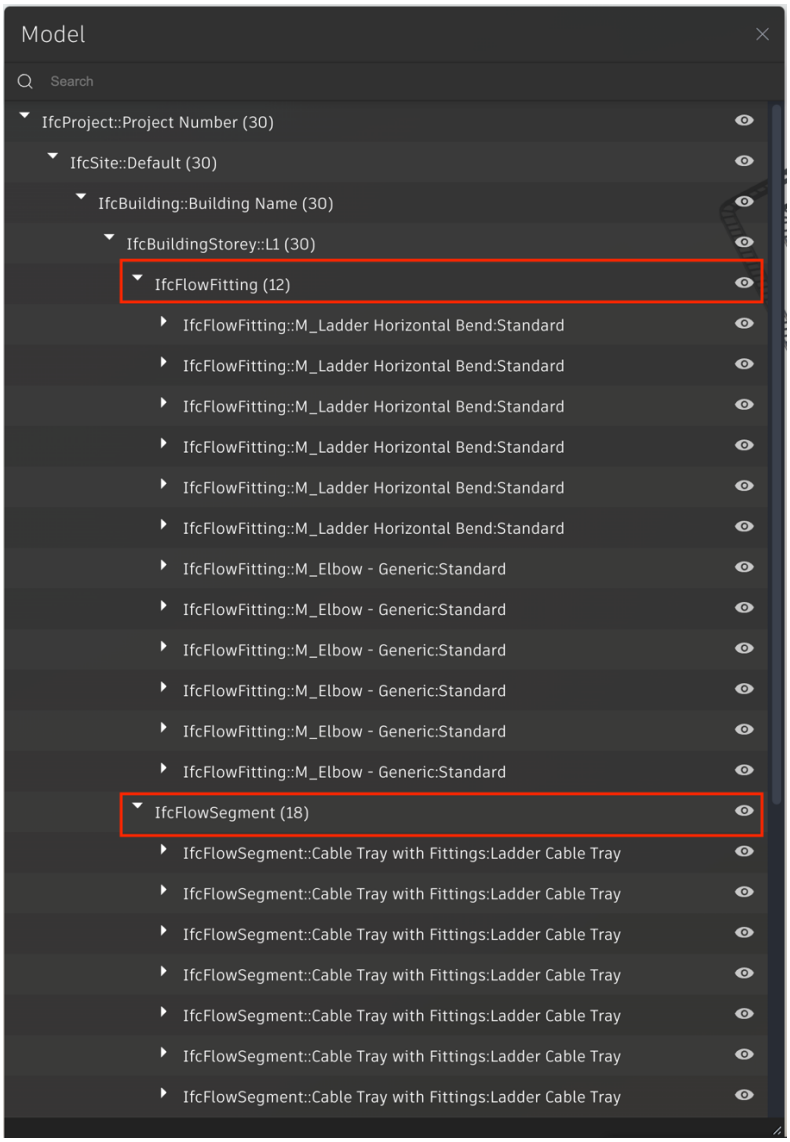
3.1.2.1 The OpeningElement was previously listed under a separate group, "IfcOpeningElement", within the "<No Level>". Now, it's under the host object where it belongs. No more "<No Level>".

v3	v4
	

3.1.2.2 Similar to IfcOpeningElement, IfcDistributionPort is now listed under the object to which it's attached, rather than being an independent group in v3.

v3	v4
	

3.1.2.3 IFC v4 will attempt to fold/merge/group objects in the instance tree as much as possible, so it may create virtual nodes that do not exist in the IFC source file to facilitate grouping of objects. For example, the IfcFlowFitting and IfcFlowSegment do not exist in the source IFC file. They are virtual nodes for grouping objects of the same type together.



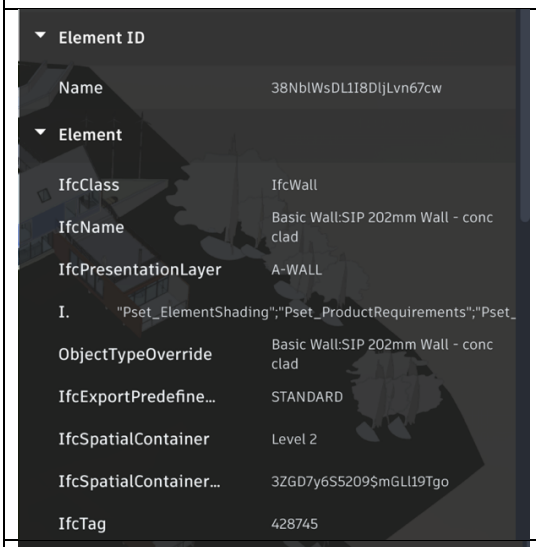
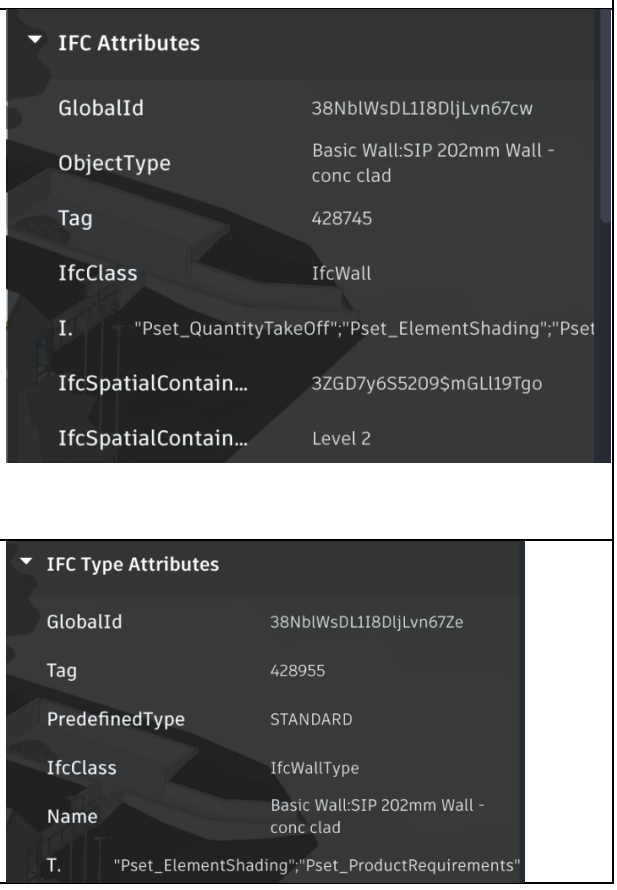
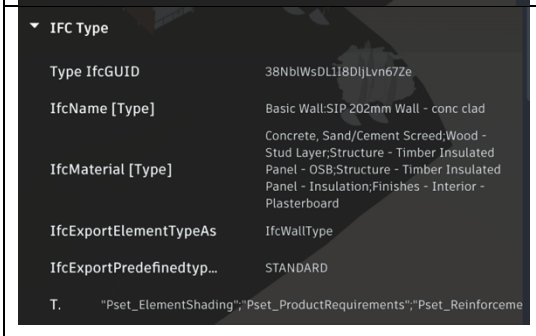
3.2 Property Changes and Enhancements

This section will cover the changes, updates, and enhancements in model objects' properties, i.e., object properties in the APS Viewer or the response of the following Model Derivative Properties API:

- GET /{urn}/metadata/{modelGuid}/properties
- POST /{urn}/metadata/{modelGuid}/properties:query

3.2.1 IFC Object Primary Property Set Renamed

IFC v4 uses "IFC Attributes" and "IFC Type Attributes" as the primary property set names for object and type properties, as opposed to "Elements" in v3, which is from Revit's concept, and may cause confusion when item types aren't derived from IfcElement, e.g., IfcSystem, IfcDistributionPort, IfcProject, etc.

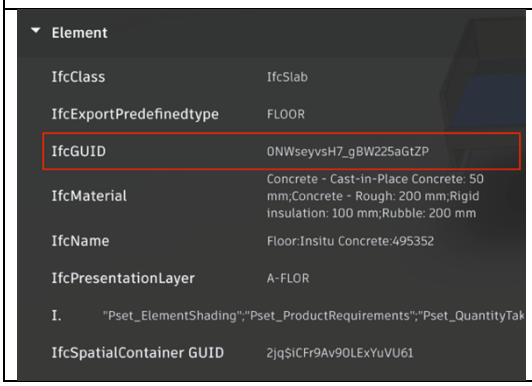
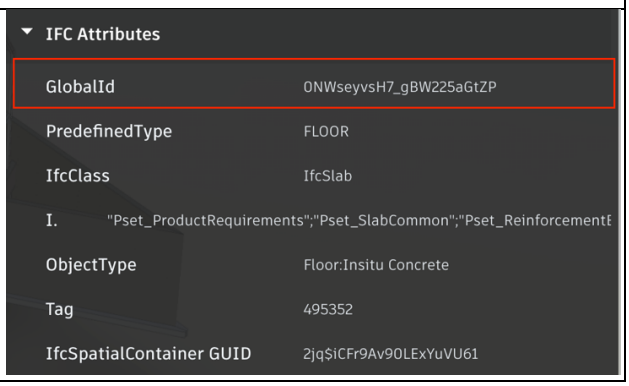
<table>
<tr><th>v3</th><th>v4</th></tr>
<tr><td> 
</td><td rowspan="2"> 
</td></tr>
<tr><td> 
</td></tr>
<tr><td></td><td>
</td></tr>
</table>

3.2.2 More descriptive property naming directly from the source IFC file

To make property naming more descriptive and avoid confusion, IFC v4 uses descriptive property naming directly from the IFC source file. There will be no more Revit and Naviworsk-related properties in the object properties, like Hidden, Required, Icon, IfcExportElementTypeAs, etc.

3.2.3 Ifc GUID renamed to follow the IFC standards

Following 3.2.2, the property name of an IFC entity's globally unique identifier is now GlobalId, compared to "IfcGuid" in IFC v3. All IFC objects are derived from IfcRoot, and the IFC standard uses "GlobalId" as the globally unique identifier.

<table>
<tr><th>v3</th><th>v4</th></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

3.2.4 SVF/SVF2 Object External ID format changed

In IFC v3, the occurrence path of the tree node is used as the external ID in SVF, while the IFC GUID (IfcGuid) is used instead in SVF2 for objects derived from IfcRoot (a.k.a. rooted objects, e.g., IfcProject, IfcWall). For non-rooted objects, it uses the IFC GUID of its parent plus the index of itself as the external id.

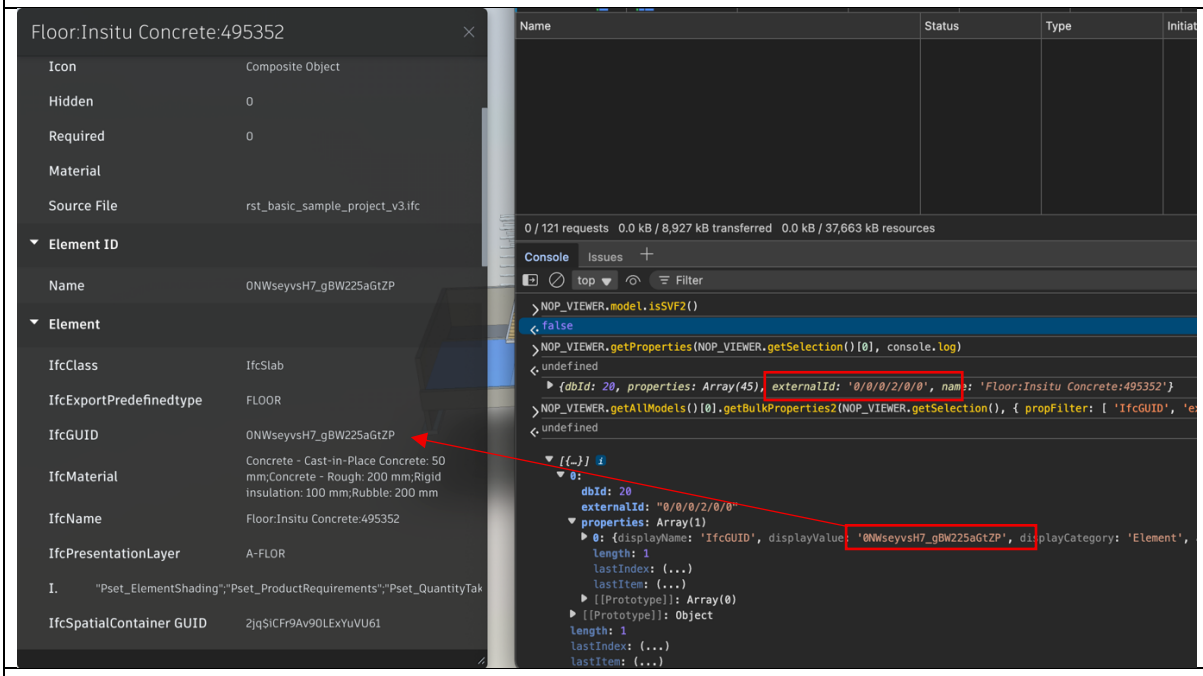
To maintain consistency, IFC v4 follows a similar pattern, which still utilizes the IFC GUID (IFC GlobalId) for rooted objects. However, for non-rooted objects, it uses a combination of the parent's IFC GUID, along with other attributes such as names and local strings, to make them as unique and persistent as possible. Additionally, the external ID values in IFC v4 will be encoded in base64 format in both SVF and SVF2.

Additionally, IFC v4 will attempt to fold objects in the instance tree as much as possible, so it may create virtual nodes to group objects that do not exist in the IFC source file. In this case, the external Id of a virtual node will be the combination of its first child's IFC GUID plus its IFC type, in base64 format.

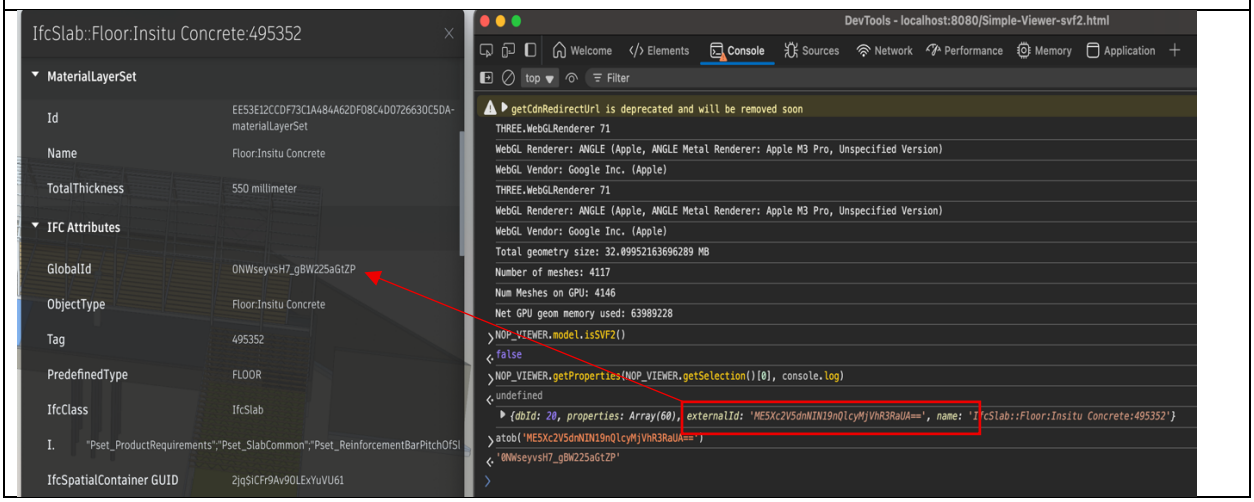
We take the external id of a rooted object, IfcSlab, as an example below. This table compares IFC v3 and v4 when loading the same source IFC model translated with different IFC conversion methods in SVF. Each image in the table shows the locations of the external ID and the IFC GUID. See the IFC GUID value in the viewer's property panel on the left, and see external id values in the Web Browser's developer console on the right.

In the case of IFC v3, the upper image, the external id shows the occurrence path of the IfcSlab object, not the actual IFC GUID value in the external id, which is meaningless. We must make another function call or iterate the object's properties to find the actual IFC GUID value. On the contrary, IFC v4, the lower image, fixes the problem to make them meaningful. Just the external ID value for the IFC GUID is a base64 encoded string. We need to decode it to get the value.

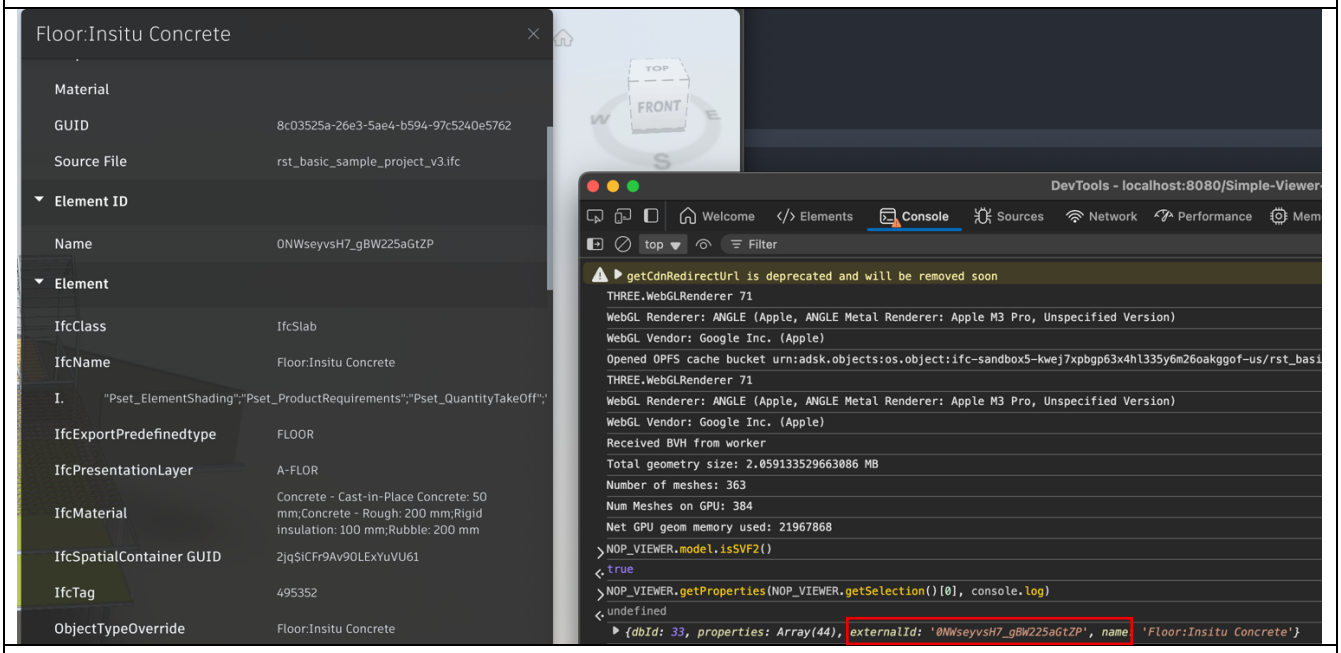
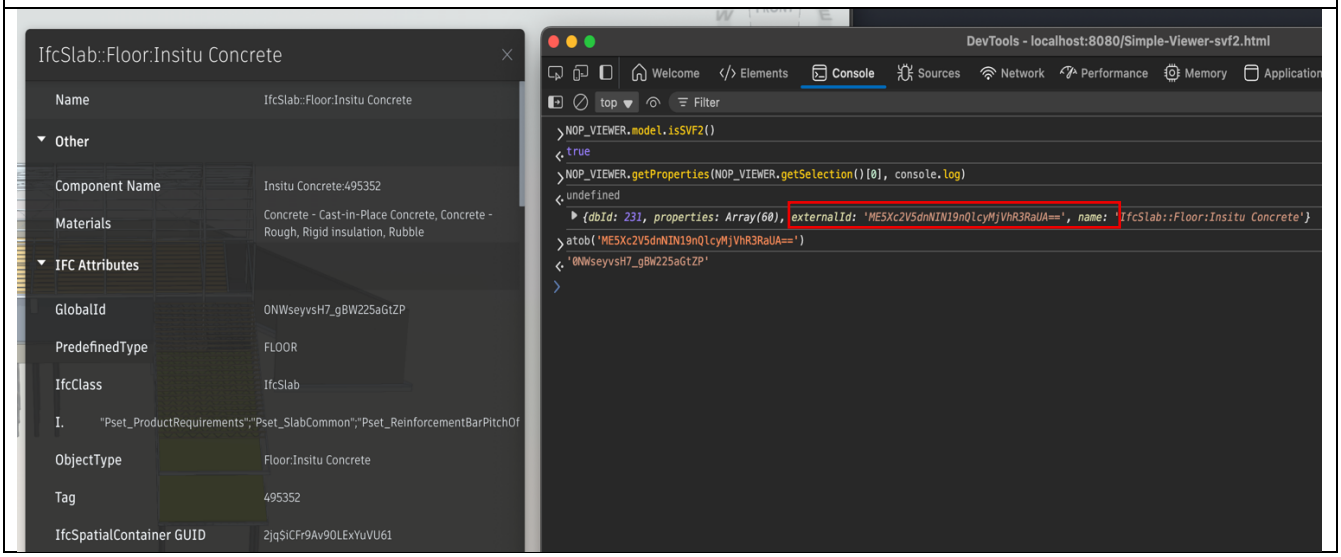
v3 in SVF



v4 in SVF



The table below shows a similar comparison between IFC v3 and v4 for external id, but the model is translated and loaded in SVF2. In the case of IFC v3, the upper image, the external id now shows the IFC GUID value of the object, but the IFC GUID is gone from the viewer property panel. The lower image is for IFC v4 in SVF2, which still shows the base64 encoded IFC GUID value in the external id. We can see the IFC v3 model in SVF and SVF2 uses different stready in the external Id, but IFC v4 uses a consistent one. We don't need extra code logic to get the IFC GUID from the external id in IFC v4 when loading the same model in SVF and SVF2.

v3 in SVF2

v4 in SVF2


3.2.5 Enhanced IFC Relationships support

IFC v4 has now improved the IFC relationships processing compared to IFC v3. Here are the supported IFC relationships in IFC v4:

IfcRelConnectsPorts	IfcRelConnectsPortToElement	IfcRelNests (including alignments)
IfcRelAggregates	IfcRelDecomposes	IfcRelAssignsToProduct
IfcRelAssociatesClassification	IfcRelAssociatesMaterial	IfcRelContainedInSpatialStructure
IfcRelDefinesByType	IfcRelFillsElement	IfcRelVoidsElement

3.3 Geometries Enhancements and Changes

This section will cover the changes, updates, and enhancements in geometry handling in IFC v4; For example, the more coverage in IFC4x3, the better the model processing success rate, as well as improved large coordinate handling.

3.3.1 IFC v4 officially supports IFC4x3

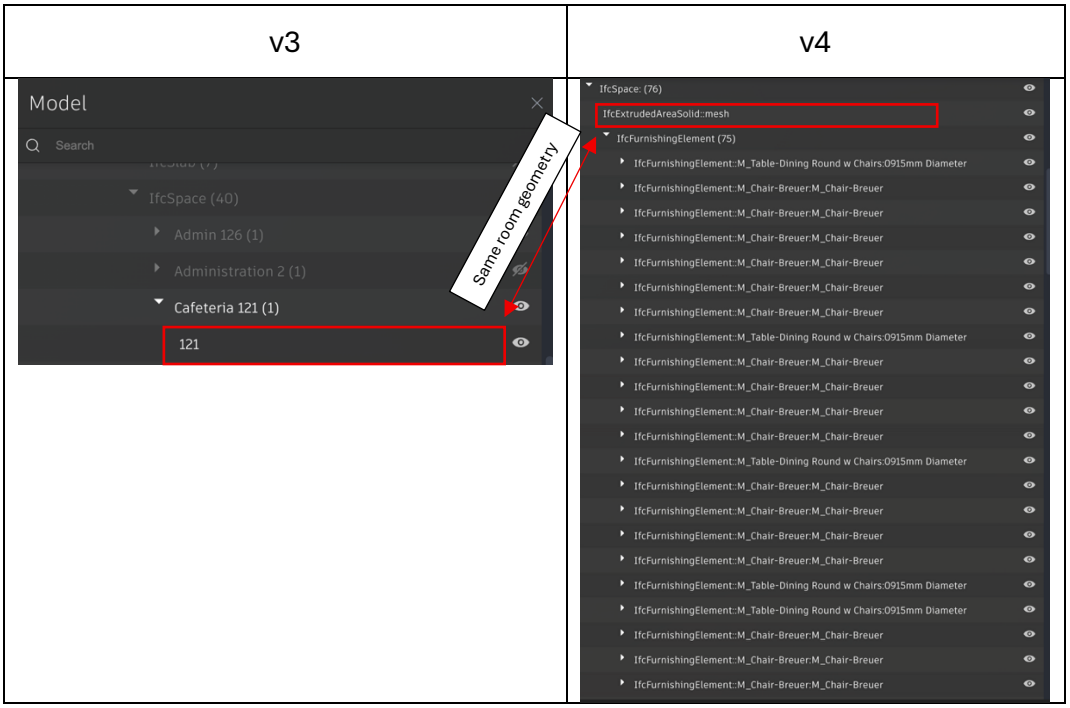
IFC v4 has enhanced IFC4x3 (IFC 4.3) support, providing more comprehensive coverage of the latest IFC specifications, particularly IFC Alignments and IfcBridge in civil engineering, compared to IFC v3. Additionally, IFC v4 supports handling IFC4x3-specific geometries, such as IfcExtrudedAreaSolidTapered, IfcClothoid curves, and Gradient curves.

3.3.2 Better Large Coordinate Handling in IFC v4

IFC v4 has introduced an improved algorithm for better and more stable handling of models with large coordinates. We no longer need to rely on the Large Coordinates Removal Tool to patch IFC source files.

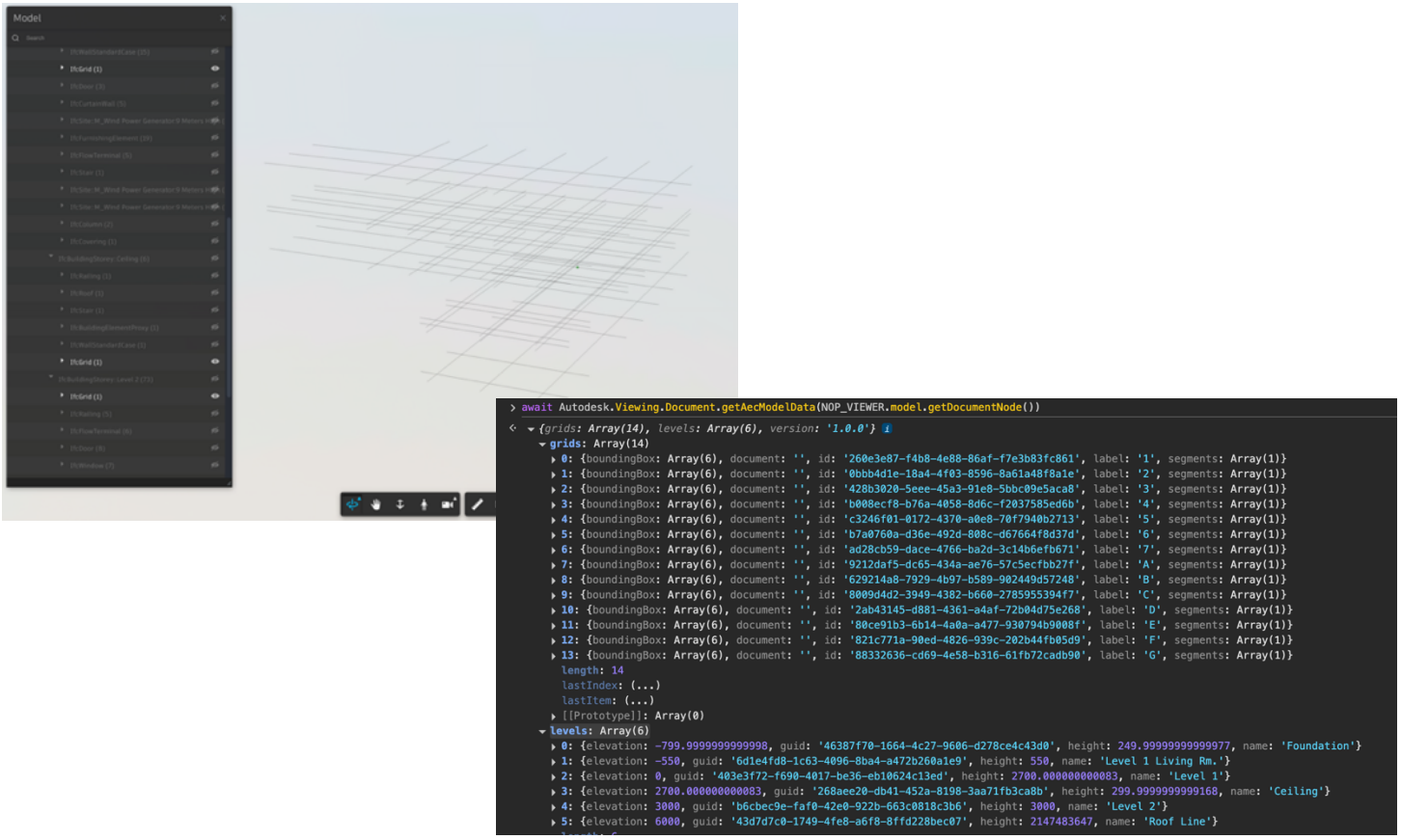
3.3.3 IfcSpace Geometry Handling in IFC v4

In IFC v4, the geometry of IfcSpace is exposed but hidden by default, with child objects visible. The table below shows the translation results for the same IfcSpace object using IFC v3 and v4. In IFC v3, we can only see the room geometry; In IFC v4, we can also see related objects in the same room.



3.3.4 IfcGrid support

IFC v4 adds support for processing IfcGrid geometries natively, so we don't need to rely on extra code to visualize IfcGrids in APS Viewer. Additionally, it also extracts IfcGrid data from "AecModelData.json", similar to IFC v3, but only when building/level/grids exist in the IFC model.

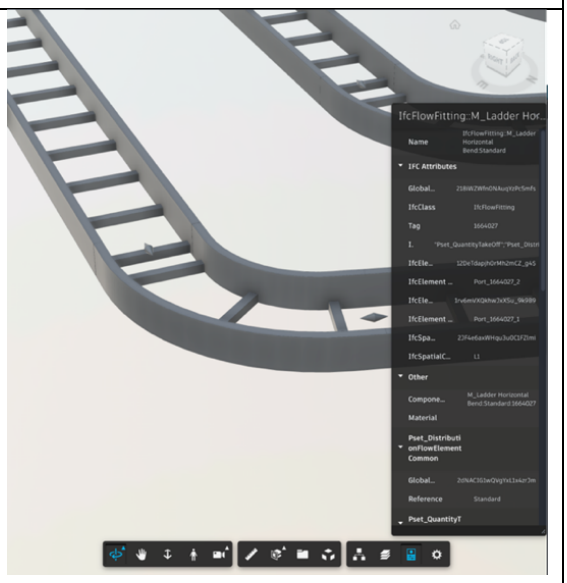


3.3.5 Some failure cases from IFC v3 can be processed well in IFC v4

Based on the updates and enhancements we discussed in sections 3.3.1–3.3.4, IFC v4 can now effectively process some reported failure cases in IFC v3, thereby increasing the success rate of working IFC files using Autodesk technologies.

3.3.6 Geometry default color changed

IFC v3's default rendering color is from Navisworks. IFC v3 will apply the default rendering color to the object geometry that has no assigned materials. IFC v4 uses the default rendering color from APS Viewer. The default colors in IFC v3 and v4 differ, which may cause the object to appear differently in color in IFC v3 and v4.

v3	v4
	

4. Manifest Changes

In this section, we will discuss the changes to the Model Derivative API manifest, i.e., the response of GET manifest.

4.1. IFC conversion method identifier changed

In the Model Derivative manifest, we can determine which IFC conversion method we used to translate an IFC model by checking the "Document Information". For IFC v4, the “IFC loader” value will be “4”, as seen in the following image.


```
"Document Information": {
    "IFC Application Name": "Autodesk Revit 2020 (ENU)",
    "IFC Application Version": "2020",
    "IFC Organization": "Autodesk Revit 2020 (ENU)",
    "ViewDefinition": "CoordinationView_V2.0",
    "implementation_level": "2;1",
    "schema_identifiers": "IFC2X3",
    "IFC Loader": "4"
}
```


4.2. Viewable node name changed

IFC v3 uses the source IFC filename as the name of the viewable bubble (type: "geometry", role: "3d"). Instead, the bubble name in IFC v4 is always "Scene".

v3


```
"outputType": "svf2",
"children": [
    {
        "guid": "d16f9899-ea0f-b40e-73fa-876f51b1352a",
        "type": "geometry",
        "role": "3d",
        "name": "rac_basic_sample_project_v3.ifc",
        "status": "success",
        "viewableID": "rac_basic_sample_project_v3.ifc",
        "hasThumbnail": "true",
        "progress": "complete",
        "useAsDefault": true,
```


v4


```
"outputType": "svf2",
"children": [
    {
        "guid": "c969710c-57d0-432b-a3f5-364114b619aa",
        "type": "geometry",
        "role": "3d",
        "name": "Scene",
        "viewableID": "rac_basic_sample_project_v4.ifc",
        "status": "success",
        "progress": "complete",
        "hasThumbnail": "true",
```


4.3. Conditional generation for AecModelData.json

IFC v4 will produce “AecModelData.json” based on the presence of IfcBuilding, IfcGrid, and IfcBuildingStorey, which may impact workflows that expect this file.

4.4. No more nwModelToWorldTransfrom

As the screenshot below shows, there used to be a transformation matrix in the Model Derivative manifest called "nwModelToWorldTransform" for mapping IFC models translated with IFC v3 from world coordinates to model coordinates. However, unfortunately, IFC v4 doesn't support this, as it's a concept that only exists in Navisworks technologies. The SVF/SVF2 models translated with IFC v4 are always in model space, so there is no need to use "nwModelToWorldTransform" to remap coordinates. If you need to transform IFC models translated with IFC v4 into world coordinates, please contact us and share your use cases with us. We will help you pass your use cases to our engineering team for evaluation.

5. IFC v4 Migration Guides and Tips

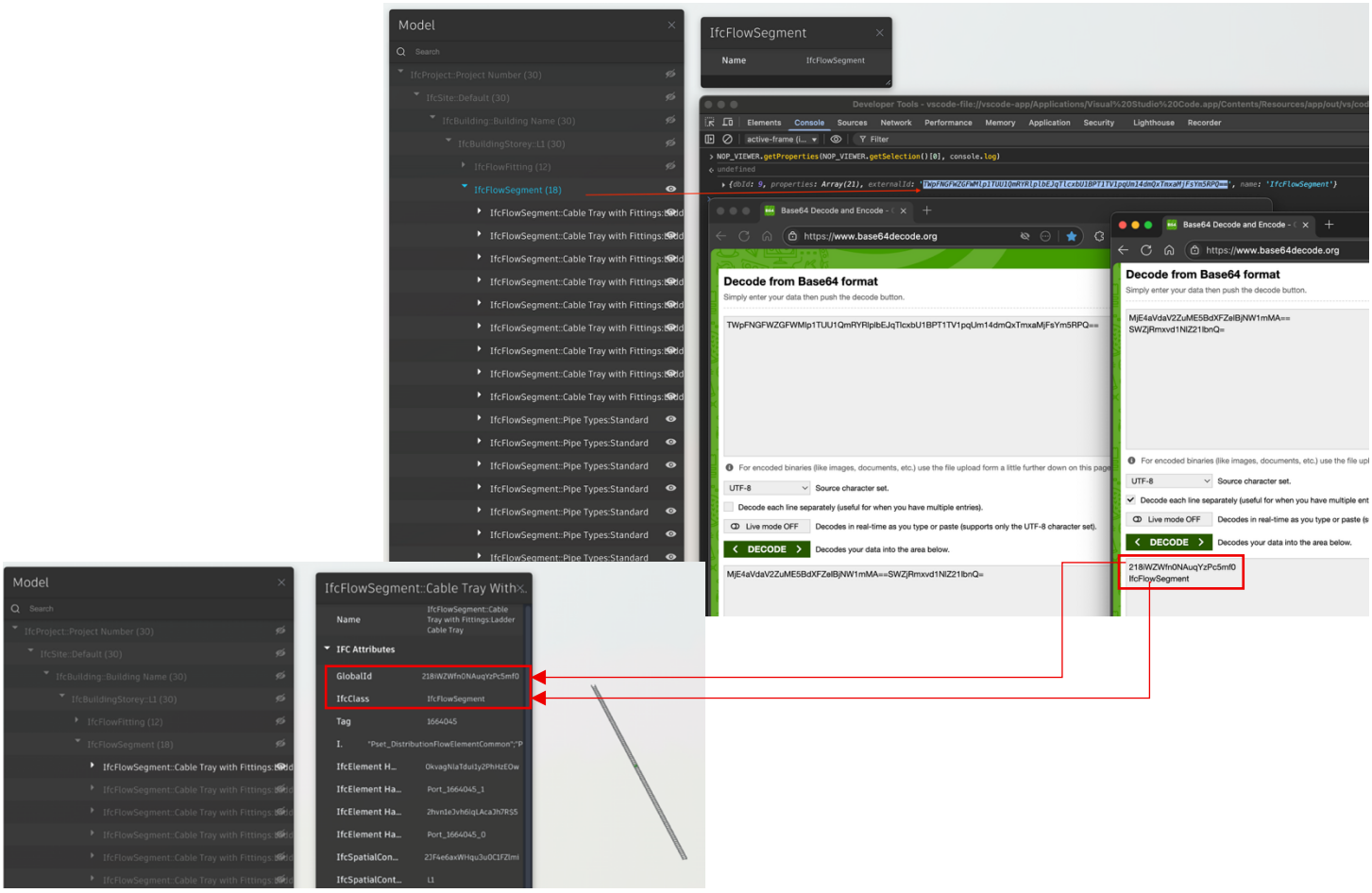
In this section, we will provide the migration guides and tips for users using the APS Viewer SDK and Model Derivative Properties API. We cover both IFC models stored on developer-managed OSS buckets and in BIM 360 Document Management and Files in Autodesk Docs.

5.1 Migration Guides and Tips for the typical workflow

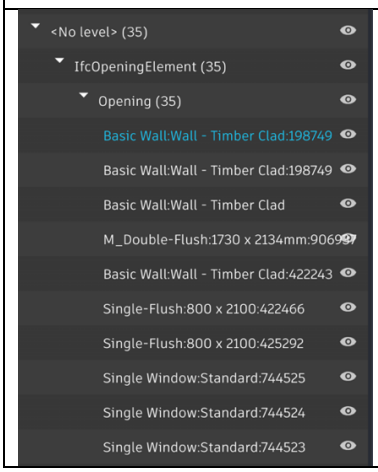
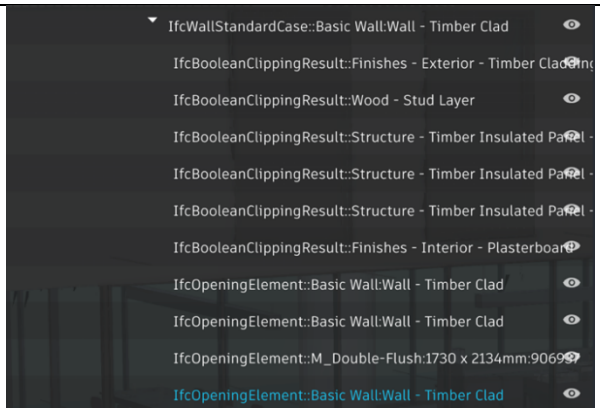
This section covers the typical workflow of using the APS Viewer SDK and Model Derivative Properties API for both IFC models stored on developer-managed OSS buckets and BIM 360 Document Management/Files in Autodesk Docs.

5.1.1. Model Hierarchy Changes

- IFC v4 will attempt to fold/merge objects in the instance tree as much as possible, so it may create “virtual” nodes to group objects that do not exist in the IFC source file. To identify a “virtual” node group, we can check its external id value; it is similar to non-rooted objects, but it will be the combination of its first child's IFC GUID plus its IFC type. Taking the IfcFlowSegment as an example, we can see this pattern after base64 decoding twice in the following image:



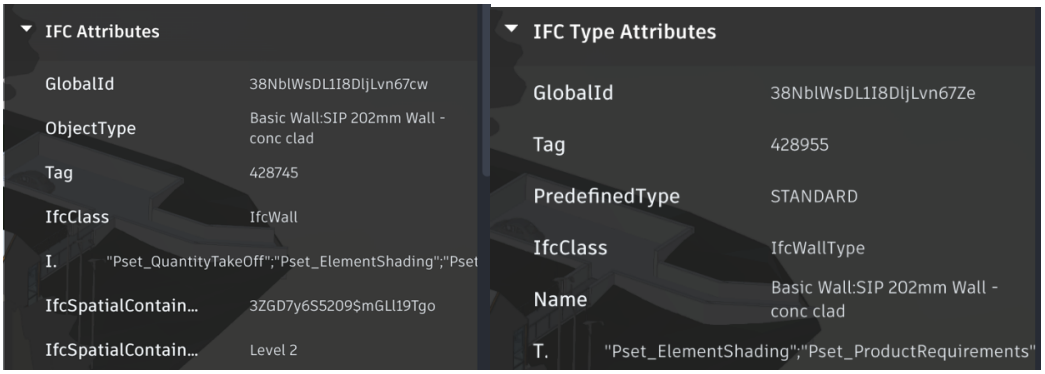
- The node location may be reorganized in IFC v4, so there is no longer a "<NO LEVEL>" node. Therefore, when iterating the model instance tree for objects like "IfcOpeningElement" or "IfcDistributionPort", please try to iterate by each IfcBuildingStorey or the object where it belongs instead.

v3	v4
	

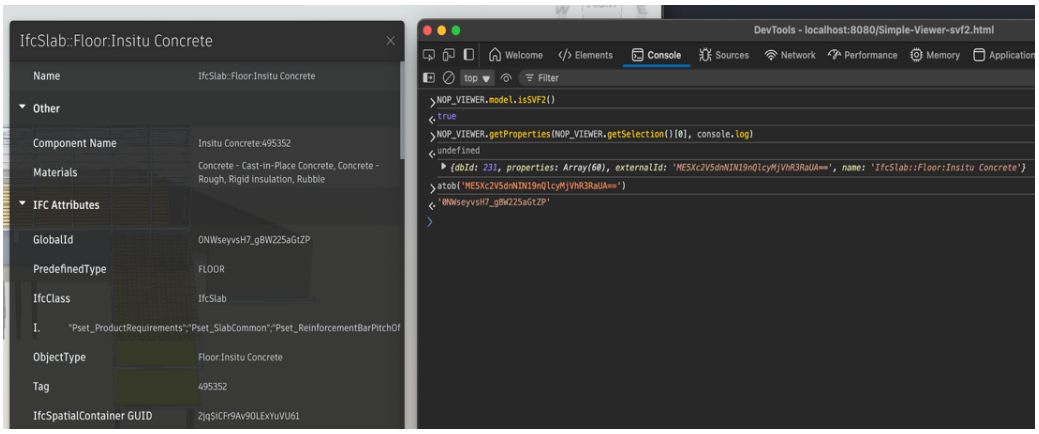
- The node name for an IFC object includes the IFC types prefix in IFC v4. Make sure to include the type when doing some name-related operation.

5.1.2. Object Property Changes

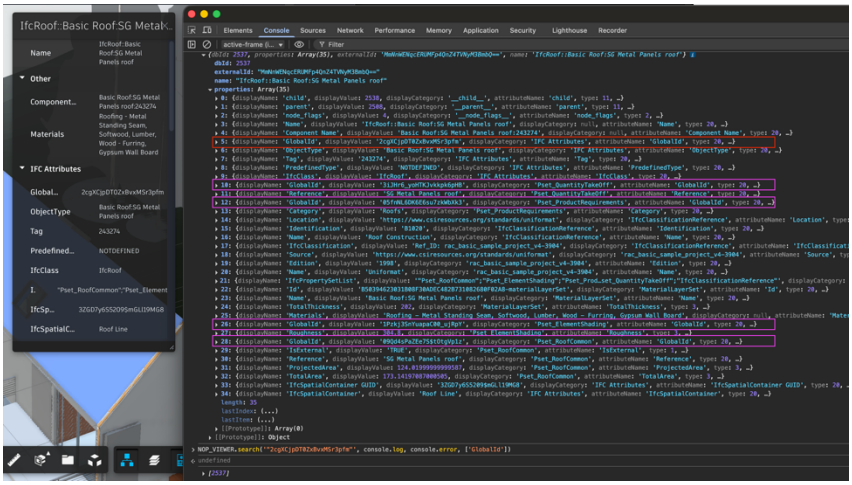
- No more Revit and Navisworks-related property names and properties, as IFC v4 uses descriptive property naming directly from the IFC source file. Try using the original property name from the IFC source files as much as possible when searching IFC properties.
- The "Element" property is no longer used in IFC v4, as it's a concept unique to Revit. Use "IFC Attributes" and "IFC Type Attributes" when searching for IFC properties by "displayCategory" using the Viewer API.



- The external ID of an IFC object in IFC v4 will be encoded in base64 format, which also applies to the Model Derivative Properties API. Make sure to decode external ids using a JavaScript function, such as atob('base64EncodedGlobalId'), to retrieve the original IFC GUID value.



- To find or search objects by IFC Guid, we can use viewer.search(), as the IFC Guid in IFC v4 is visible on the viewer's property panel, unlike IFC v3. However, it may return not just one matched result, as the attribute name "GlobalId" will appear in other similar properties but different display categories. Or we can use the advanced property searching approach to only search "GlobalId" under IFC Attributes".




```
function userFunction( pdb, idVals ) {
    const attrNames = ['IfcGUID', 'GlobalId'];

    // Now iterate over all parts to find out which one is the largest.
    const result = [];

    pdb.enumObjects(( dbId ) => {
        // For each part, iterate over their properties.
        pdb.enumObjectProperties( dbId, ( attrId, valId ) => {
            const def = pdb.getAttributeDef( attrId );

            const propName = def.name;
            const displayName = def.displayName;
            const displayCategory = def.category;

            if( attrNames.includes(propName) || attrNames.includes(displayName) ) {
                const value = pdb.getAttrValue( attrId, valId );
                if( idVals.includes( value ) && displayCategory == 'IFC Attributes' ) {
                    result.push( dbId );
                    return true;
                }
            }
        });
    });

    // Return results
    return result;
}
undefined
NOP_VIEWER.model.getPropertyDb().executeUserFunction( userFunction, [ '2cgXCjpDT0ZxBvxMSr3pfm' ] ).then( ( result ) => console.log( result ) )
Promise {<pending>}
[2537]
```

5.1.3. Manifest Changes

- As mentioned in section 4.4 above, IFC v4 doesn't generate "nwModelToWorldTransform" for mapping IFC models translated from IFC v3, as it's a concept specific to Naviworks technologies. The SVF/SVF2 models translated with IFC v4 are always in model space, so there is no need to use "nwModelToWorldTransform" to remap coordinates back to model space.
- IFC v4 won't always generate "AecModelData.json". It only generates this file based on the presence of IfcBuilding, IfcGrid, and IfcBuildingStorey. So, if you don't see AecModelData.json generated in the Model Derivative manifest, please check if the source IFC model has IfcBuilding, IfcGrid, and IfcBuildingStorey included.

5.2 Migration Guides and Tips specific to for BIM 360 and Autodesk Docs

This section covers BIM 360 Document Management and Files in Autodesk Docs specific workflow for using the APS Viewer SDK and Model Derivative API.

5.2.1 IFC v4 availability on Docs

The IFC v4 conversion method will be available by default for ACC projects created after July 22nd. For existing ACC projects, they will continue using IFC v3 by default. The IFC v4 migration guides and tips discussed in Section 5.1 also apply to BIM360 and Autodesk Docs specific workflows.

5.2.2 Don’t use the workflow that mixes IFC v3 and IFC v4

To prevent compatibility issues and workflow conflicts, don’t mix existing projects using IFC v3 with new projects using IFC v4.

For example, the ACC Bridge tool is the enabling technology that lets you easily share project information with other Autodesk Construction Cloud (ACC) projects or accounts. When working with IFC files across projects using the Bridge tool, please **DON’T** bridge the IFC files across projects that use different IFC conversion methods (e.g., v3 vs v4). Mixing IFC v3 and v4 with the bridge would cause some ACC features not to work, and it would require a manual process to resolve the issue (e.g., re-uploading files).