

IFC2x3_pset Dataset Extension

Quick Reference Guide

revision: May 21, 2010 - 15:25

Table of Contents

Terms of Reference.....	3
Installation of IFC2x3_pset dataset extension	4
Installation options	5
IFC2x3_pset dataset extension for an individual project	6
IFC2x3_pset dataset extension for selected projects	7
IFC2x3_pset dataset extension for all projects	8
The IFC2x3_pset DataGroup system.....	10
IFC2x3_pset DataGroup types.....	10
IFC2x3_pset DataGroup definition files	10
IFC2x3_pset DataGroup catalog items	12
IFC2x3_pset instance data	13
IFC Export	14
'Output' tab.....	14
IFC File.....	14
IFC File Info.....	14
IFC Schema	14
Mapping of Family/Parts to IFC entities	14
'Comment' tab.....	16
Project, Site, Building.....	16
IFC Storey Containment.....	16
'Settings' tab	18
Generate 3D Spaces from 2D Spaces.....	18
Override DataGroup Values from Settings File.....	18
IfcExport	20
Export message log	20
IFC Import.....	22
'Input' tab	22
IFC File.....	22
'Settings' tab	22
Use IFCMaterialName As Part/Family Name	22
Use Symbology from Part/Family instead of IFC File	24
Override DataGroup Values from Setting File	26
Ignore Storey Containment	27
IfcImport	27
Appendix.....	28
Configuration Variables	28

Terms of Reference

[buildingSMART](#), the International Alliance for Interoperability, is an organization of building industry stakeholders for the creation of an **industry standard for data interoperability across Building Information Modeling (BIM) applications** used over a building's life cycle. This standard is regarded as a prerequisite for improving building design and engineering workflows using BIM methods, hence eliminating the high cost and waste created by inadequate interoperability. A **vendor-independent neutral file format, IFC (Industry Foundation Classes) capture both geometry and properties of 'intelligent' building objects and their relationships within building information models**, thus facilitating the sharing, exchange, and coordination of information across otherwise incompatible applications. Bentley is fully committed to the objectives of the IAI and their buildingSMART initiatives and is involved in several regional chapters of the IAI as well as in several IFC-related projects and initiatives.

For further information, see Bentley's [IFC White Paper](#)

Bentley's IFC2x interface for the Coordination View was certified in 2003, and although not officially certified, IFC2x2 is also supported. In March 2007, in accordance with the official IAI facilitated approval procedure, Bentley Architecture 08.09.03 **passed the certification for import and export of IFC2x Edition 3 (IFC2x3)**, which is based on the **Extended Coordination View** definition. As the **IFC interface is built on the common Bentley Building technology platform**, Bentley Structural, Bentley Building Mechanical Systems, and Bentley Building Electrical Systems effectively also support the IFC2x3 Extended Coordination View.



The main objective of IFC is the **transfer of BIM information**, i.e. geometry and associated properties, from one BIM application to another for the purpose of **coordination** (hence 'Coordination View') and/or the extraction/handover of certain property values for **downstream tasks**, such as spatial analysis.

IFC2x2 and IFC2x3 support **common property sets**, i.e. object property names and formats agreed upon by the Model Support Group of the IAI and the IFC implementers.

To import and export geometry with **basic properties**, e.g. element type, length, height, width, etc., the IFC2x3 property sets are not required. However, to import or export geometry with **extended properties**, such as fire rating, thermal transmittance, exterior/interior, load bearing/non-load bearing, combustibility, shading coefficient, etc., the IFC2x3 property sets must be appended to the Bentley Building DataGroup system.

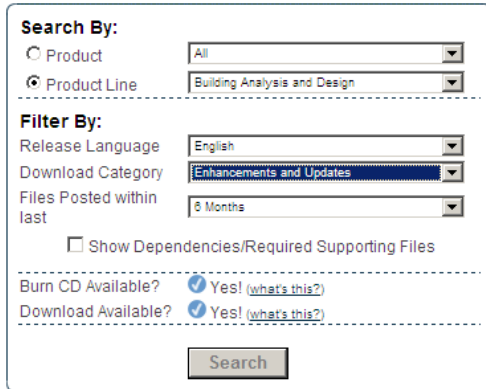
Bentley Architecture handles these properties through **IFC2x3-specific DataGroup schema files**, which are included in *IFC2x3_pset_DatasetExtension.zip*. When extracted, the **Bentley Architecture DataGroup schema is augmented** with the necessary folders and files. Configuration variables must be added in the PCF-files to add the necessary search paths. Additional non-common properties can be accommodated by modifying the DataGroup schema with corresponding DataGroup types and definition files. Please contact your Account Manager or [Professional Services](#) if you require consultancy.

IFC2x3_pset_DatasetExtension_081107 contains 70 IFC2x3 property sets. If required, further property sets can be added using the DataGroup Definition editor.

For further information, see the relevant chapters in the Quick Reference Guide.

Installation of IFC2x3_pset dataset extension

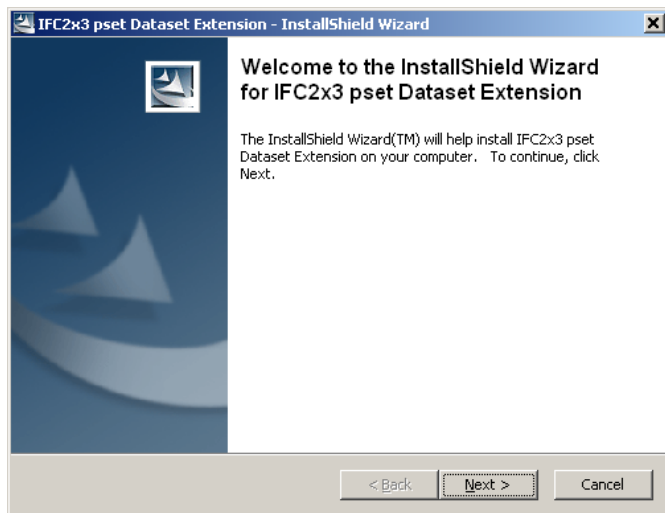
- ❑ download the *IFC2x3_pset_Dataset Extension* from Bentley SELECTservices searching by Product Line 'Building Analysis and Design', Download Category 'Enhancements and Updates'



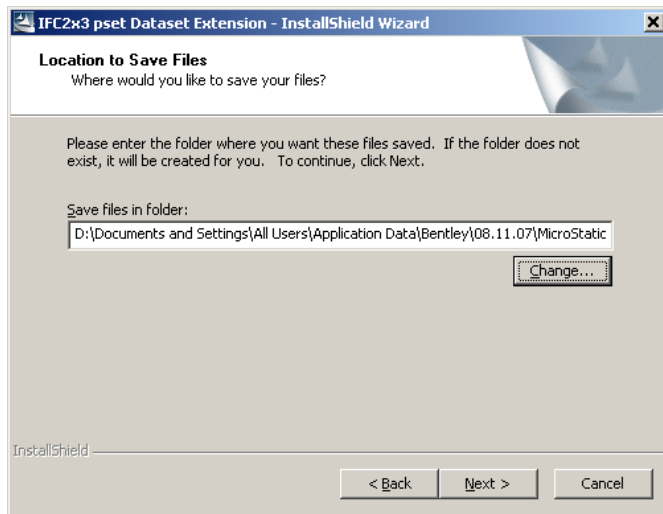
The screenshot shows a search interface with the following fields and options:

- Search By:**
 - ☐ Product: All
 - ☒ Product Line: Building Analysis and Design
- Filter By:**
 - Release Language: English
 - Download Category: Enhancements and Updates
 - Files Posted within last: 8 Months
- ☐ Show Dependencies/Required Supporting Files
- Burn CD Available? ☒ Yes! (what's this?)
- Download Available? ☒ Yes! (what's this?)
- Search** button

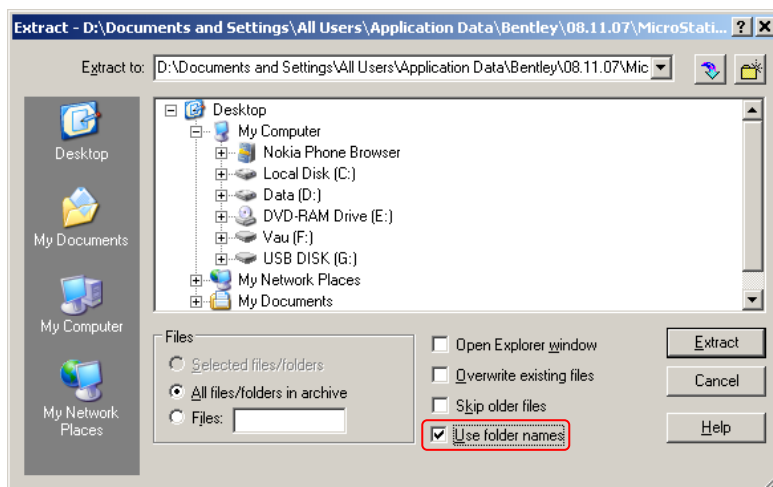
- ❑ download the *IFC2x3_pset_DatasetExtension* to your system, e.g. C:\BentleyDownloads
- ❑ open or double-click *ifc2x3psetdataset08110701en.exe* to install the *IFC2x3_pset_DatasetExtension* to the appropriate workspace



- ❑ navigate to the folder where you want to install the *IFC2x3_pset_DatasetExtension* (see Installation options below)



- ❑ unzip *IFC2x3_pset_DatasetExtension_081107.zip* with the 'Use folder names' ticked



Installation options

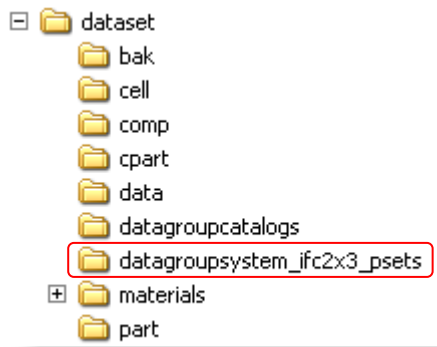
The IFC2x3_pset dataset extension can be added to the DataGroup system in several ways:

- ❑ to the project dataset, typically located in the project folder, for use on an individual project only;
or
- ❑ to a dataset-independent folder, for instance located in
.../Workspace/TriForma/DatasetExtensions/IFC2x3_psets, for use on selected projects;
or
- ❑ to a dataset-independent folder, for instance located in
.../Workspace/TriForma/DatasetExtensions/IFC2x3_psets, for use on all projects

IFC2x3_pset dataset extension for an individual project

To use the IFC2x3 property sets for an individual project:

- ❑ save *IFC2x3_pset_DatasetExtension_081107.zip* to the dataset folder of the project, e.g. `.../projects/examples/<Project_name>/dataset`
- ❑ unzip *IFC2x3_pset_DatasetExtension_081107.zip* with the 'Use folder name' option enabled; this adds the subfolder *datagroupsystem_ifc2x3_psets* to the *dataset* folder with the necessary files



- ❑ add a subfolder *datagrouppcatalogs* to the project's *dataset* folder for project-specific DataGroup catalog items
- ❑ add the following configuration variables (in black) to those already defined (in grey) in the project's PCF-file, i.e. `<Project_name>.pcf`:

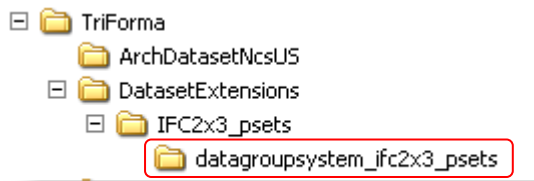
```
#-----  
# Add search paths for project-level portions of the TriForma Dataset  
#-----  
PROJ_DATASET          = $(_USTN_PROJECTDATA)dataset/  
  
TFDIR_PART             < $(PROJ_DATASET)part/  
TFDIR_CELL             < $(PROJ_DATASET)cell/  
TFDIR_COMP             < $(PROJ_DATASET)comp/  
TFDIR_CPART           < $(PROJ_DATASET)cpart/  
MS_CELLLIST           < $(PROJ_DATASET)cell/*. *  
MS_MATERIAL            < $(PROJ_DATASET)materials/  
MS_BACKUP              < $(PROJ_DATASET)bak/  
DG_CATALOGS_PATH       > $(PROJ_DATASET)datagrouppcatalogs/  
DG_PATH                > $(PROJ_DATASET)datagroupsystem_ifc2x3_psets/  
DG_CATALOGS_PATH       > $(PROJ_DATASET)datagrouppcatalogs/
```

Note: Changes to the IFC2x3_psets extension will only affect this project.

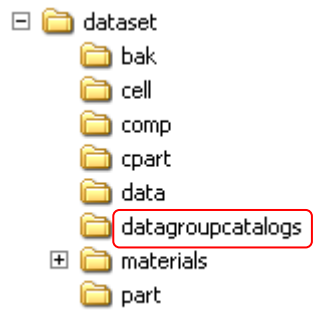
IFC2x3_pset dataset extension for selected projects

To use the IFC2x3 property sets in selected projects:

- ☐ in the .../Workspace/TriForma folder, create a subfolder 'DatasetExtensions'
- ☐ in the folder 'DatasetExtensions', create a subfolder 'IFC2x3_psets'
- ☐ save *IFC2x3_pset_DatasetExtension_081107.zip* to the folder .../Workspace/TriForma/DatasetExtensions/IFC2x3_psets
- ☐ unzip *IFC2x3_pset_DatasetExtension_081107.zip* with the 'Use folder name' option enabled; this adds the subfolder *datagroupsystem_ifc2x3_psets* to the *IFC2x3_psets* folder with the IFC2x3 pset files



- ☐ in the *dataset* folder of each project that should use the IFC2x3 psets, add a subfolder *datagroupcatalogs* for project-specific DataGroup catalog items



- ☐ add the following configuration variables (in black) to those already defined (in grey) in the PCF-files of each project that the IFC2x3 psets are to be used for, i.e.<Project_name>.pcf:

```
#-----  
# Add search paths for project-level portions of the TriForma Dataset  
#-----  
PROJ_DATASET          = $_(USTN_PROJECTDATA)dataset/  
  
TFDIR_PART             < $(PROJ_DATASET)part/  
TFDIR_CELL             < $(PROJ_DATASET)cell/  
TFDIR_COMP             < $(PROJ_DATASET)comp/  
TFDIR_CPART            < $(PROJ_DATASET)cpart/  
MS_CELLLIST            < $(PROJ_DATASET)cell/*.*  
MS_MATERIAL            < $(PROJ_DATASET)materials/  
MS_BACKUP              < $(PROJ_DATASET)bak/  
DG_CATALOGS_PATH       > $(PROJ_DATASET)datagroupcatalogs/  
DG_CATALOGS_PATH       > $(PROJ_DATASET)datagroupcatalogs/
```

```
#-----
# Add search paths to support the IFC2x3 property sets
#-----
IFC_PSETS          = $(TF_DATASETS)DatasetExtensions/IFC2x3_psets/

DG_PATH            > $(IFC_PSETS)datagroupsystem_ifc2x3_psets/
```

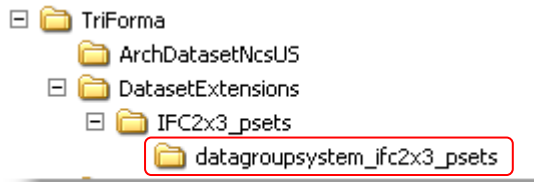
Notes:

1. The IFC_PSETS configuration variable defines the path to the *IFC2x3_psets* folder, which is subsequently used to define the location of the *datagroupsystem_ifc2x3_psets* subfolder. If the folder location is changed or different, only the search path defined by IFC_PSETS needs to be edited. Any other location is possible as long as the paths to the *datagroupsystem_ifc2x3_psets* folder is defined by the configuration variable DG_PATH in the PCF-file(s).
2. Changes to the IFC2x3_psets will affect all projects with these configuration variables in their corresponding PCF-file.
3. Other dataset extensions, e.g. for company-specific content, can be added to the *DatasetExtensions* folder by adding appropriate folders and subfolders and defining corresponding configuration variables, e.g. for company-level standards:

IFC2x3_pset dataset extension for all projects

To use the IFC2x3 property sets in all projects:

- ☐ in the .../Workspace/TriForma folder, create a subfolder 'DatasetExtensions'
- ☐ in the folder 'DatasetExtensions', create a subfolder 'IFC2x3_psets'
- ☐ save *IFC2x3_pset_DatasetExtension_081107.zip* to the folder .../Workspace/TriForma/DatasetExtensions/IFC2x3_psets
- ☐ unzip *IFC2x3_pset_DatasetExtension_081107.zip* with the 'Use folder name' option enabled; this adds the subfolder *datagroupsystem_ifc2x3_psets* to the *IFC2x3_psets* folder with the IFC2x3 pset files



- ☐ add the following configuration variables in the UCF-file, e.g. .../Workspace/Users/triforma.ucf:

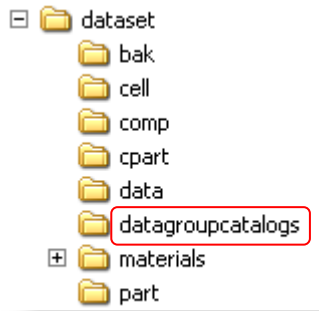
```
#-----
# Add search paths to support the IFC2x3 property sets
#-----
IFC_PSETS          = $(TF_DATASETS)DatasetExtensions/IFC2x3_psets/

DG_PATH            > $(IFC_PSETS)datagroupsystem_ifc2x3_psets/
```

Note:

The IFC_PSETS configuration variable defines the path to the *IFC2x3_psets* folder, which is subsequently used to define the location of the *datagroupsystem_ifc2x3_psets* subfolder. If the folder location is changed or different, only the search path defined by IFC_PSETS needs to be edited. Any other location is possible as long as the paths to the *datagroupsystem_ifc2x3_psets* folder is defined by the configuration variable DG_PATH in the PCF-file(s).

- ❑ in the *dataset* folder of each project, add a subfolder *datagroupcatalogs* for project-specific DataGroup catalog items



- ❑ add the following configuration variable (in black) to those already defined (in grey) in the PCF-files of each project that the IFC2x3 psets are to be used for, i.e.<Project_name>.pcf:

```
#-----  
# Add search paths for project-level portions of the TriForma Dataset  
#-----  
PROJ_DATASET          = $(_USTN_PROJECTDATA)dataset/  
  
TFDIR_PART             < $(PROJ_DATASET)part/  
TFDIR_CELL             < $(PROJ_DATASET)cell/  
TFDIR_COMP             < $(PROJ_DATASET)comp/  
TFDIR_CPART            < $(PROJ_DATASET)cpart/  
MS_CELLLIST            < $(PROJ_DATASET)cell/*. *  
MS_MATERIAL            < $(PROJ_DATASET)materials/  
MS_BACKUP              < $(PROJ_DATASET)bak/  
DG_CATALOGS_PATH       > $(PROJ_DATASET)datagroupcatalogs/  
DG_CATALOGS_PATH       > $(PROJ_DATASET)datagroupcatalogs/
```

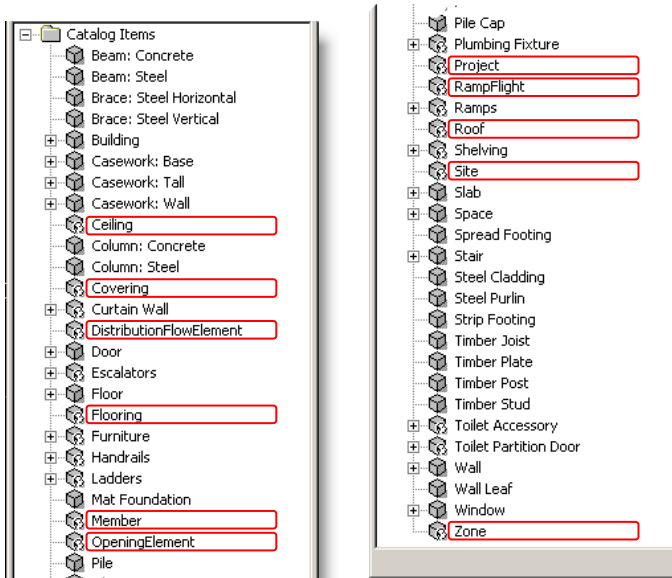
Note: Changes to the IFC2x3_property sets will affect all projects

The IFC2x3_pset DataGroup system

With the IFC2x3_pset_DatasetExtension files properly installed and configured, the following modifications are made to the Bentley Architecture DataGroup system:

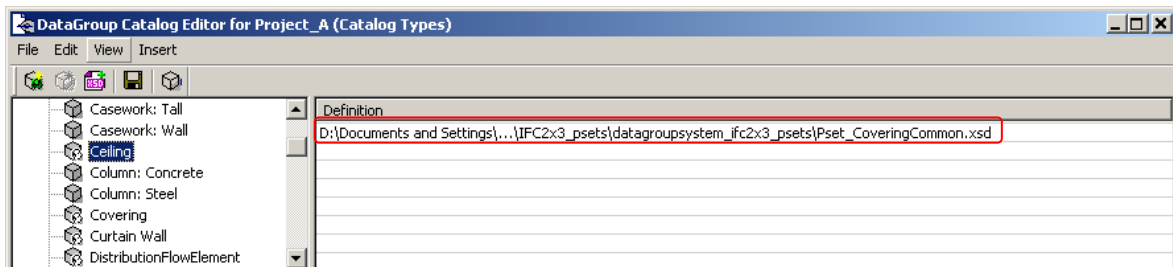
IFC2x3_pset DataGroup types

DataGroup types that are not already part of the Bentley Architecture delivered schema, i.e. Ceiling, Covering, DistributionFlowElement, Flooring, Member, OpeningElement, Project, RampFlight, Roof, Site and Zone, are added to the DataGroup system

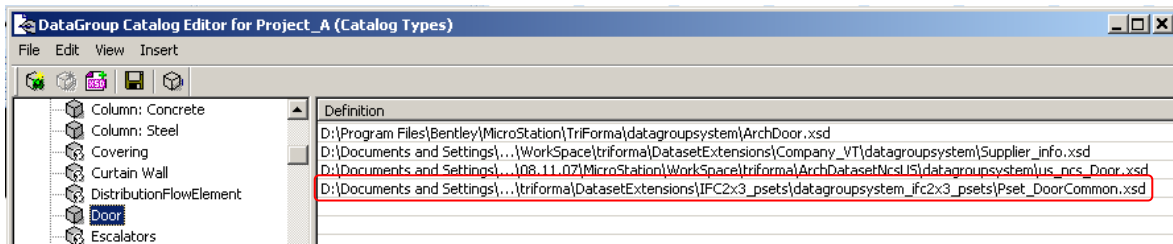


IFC2x3_pset DataGroup definition files

IFC2x3_pset DataGroup definition files are added to the appropriate DataGroup types:

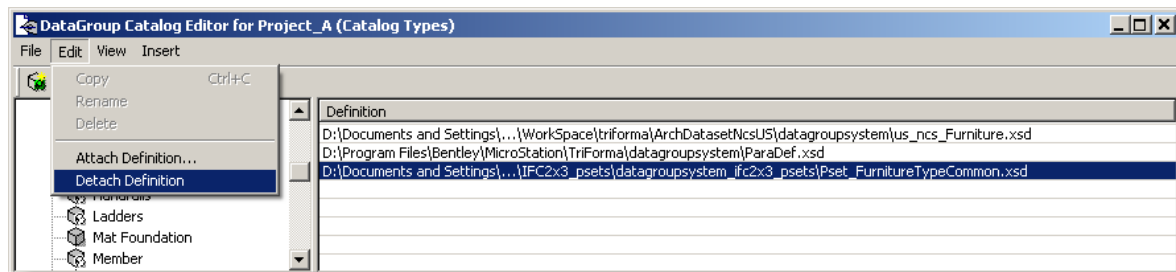


(DataGroup entity with IFC2x3_pset definition file only)



(DataGroup entity with Bentley Architecture and IFC2x3_pset definition files)

IFC2x3_pset DataGroup definitions can be attached or detached using the DataGroup Catalog Editor (Catalog Types).



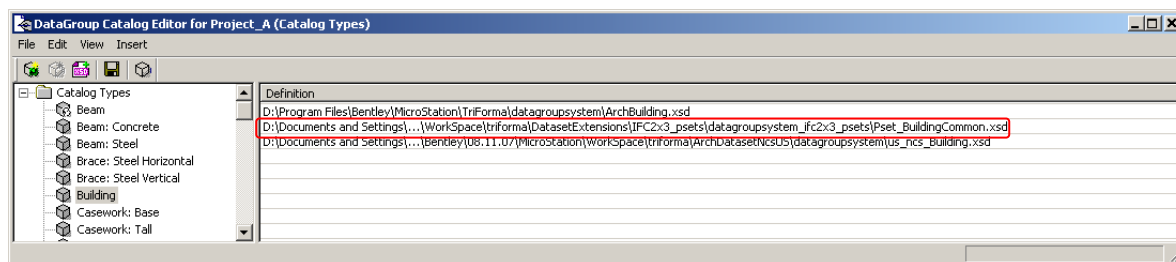
Note:

Attachments of IFC2x3 property sets (definition files) are defined in file 'ifc_pset_catalogtypeexts.xml'. In this file, all property sets (InstanceDataDefinitions) except the 'common' ones, are commented out, i.e. will not be used.

To attach further InstanceDataDefinitions

- ☐ open 'ifc_pset_catalogtypeexts.xml' with a text editor
Caution: it is strongly recommended to use a XML editor!

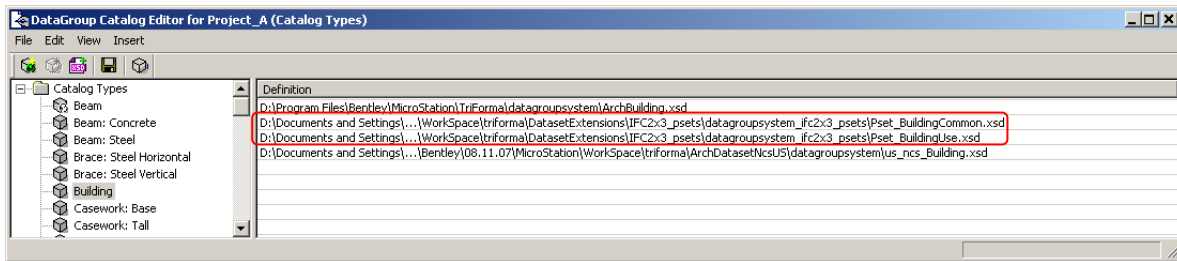
```
<CatalogTypeExtension type="Building">
  <InstanceDataDefinition defType="USER" definition="Pset_BuildingCommon"/>
  <!--<InstanceDataDefinition defType="USER" definition="Pset_BuildingUse"/> -->
  <!--<InstanceDataDefinition defType="USER" definition="Pset_BuildingEnergyTarget"/> -->
  <!--<InstanceDataDefinition defType="USER" definition="Pset_OutsideDesignCriteria"/> -->
</CatalogTypeExtension>
```



(only Pset_BuildingCommon.xsd is attached)

- ☐ delete the '<!--' characters before and the '-->' characters after the appropriate line(s); this will effectively attach them, e.g.

```
</CatalogTypeExtension>
<CatalogTypeExtension type="Building">
  <InstanceDataDefinition defType="USER" definition="Pset_BuildingCommon"/>
  <InstanceDataDefinition defType="USER" definition="Pset_BuildingUse"/>
  <!--<InstanceDataDefinition defType="USER" definition="Pset_BuildingEnergyTarget"/> -->
  <!--<InstanceDataDefinition defType="USER" definition="Pset_OutsideDesignCriteria"/> -->
```



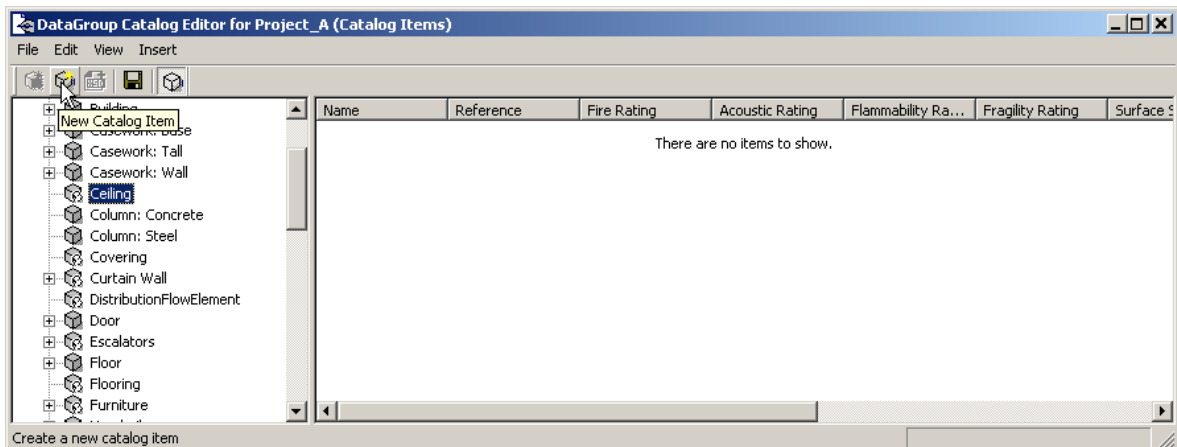
(Pset_BuildingCommon.xsd and Pset_BuildingUse.xsd are attached)

Caution: Syntax errors could corrupt the XML-file and with that the entire DataGroup system!
Backup the file before editing!

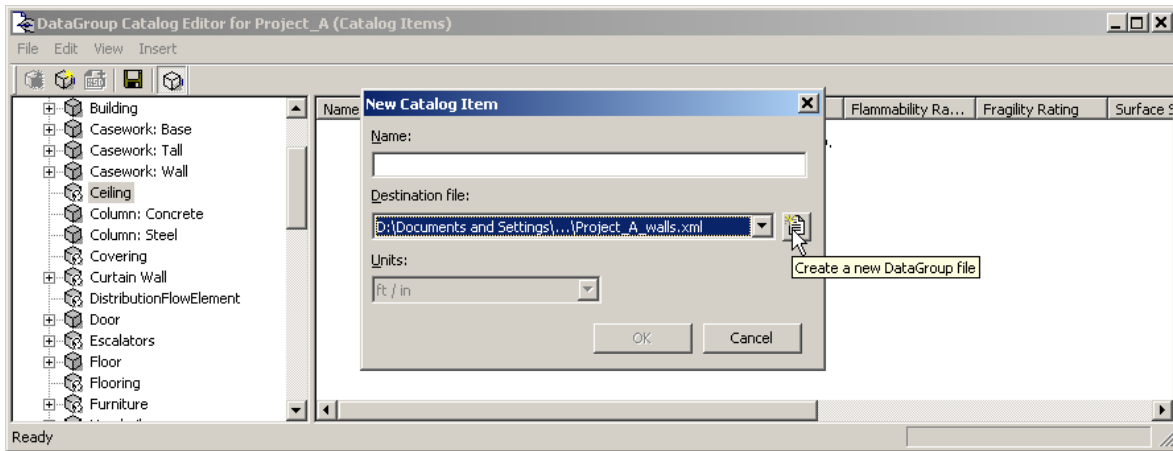
IFC2x3_pset DataGroup catalog items

To add DataGroup catalog items to DataGroup types not delivered with Bentley Architecture, e.g. Ceiling, Flooring, etc.

- ☐ open the DataGroup Catalog Editor (Catalog Items)
- ☐ click the 'New Catalog Item' button (or right-click on the catalog type and select 'New Catalog Item' from the pop-up menu)



- ☐ if necessary, create a new DataGroup file in the appropriate project-level or company-level *datagroupcatalogs* folder for the new catalog item(s)

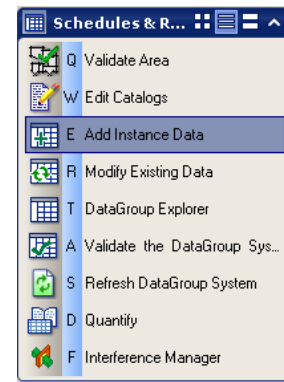
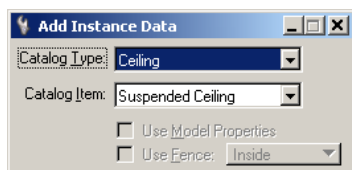


- ❑ add the new DataGroup catalog item to the appropriate project-level or company-level destination file

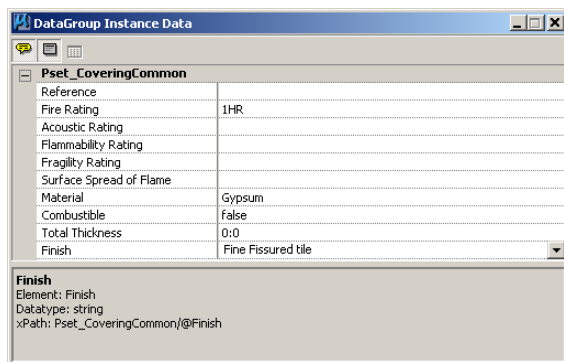
IFC2x3_pset instance data

To attach DataGroup data to DataGroup types, which Bentley Architecture currently does not automatically attach instance data to, e.g. Ceiling, Flooring, etc.

- ❑ select the 'Add Instance Data' tool
- ❑ In the 'Add Instance Data' form, select the appropriate Catalog Type and Catalog Item:



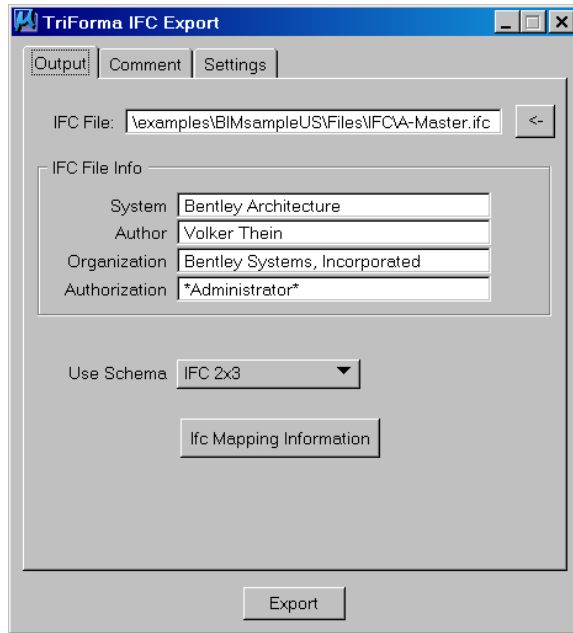
- ❑ In the 'DataGroup Instance Data' form, fill in the property values to be added:



- ❑ Identify the element(s) with a data point to attach the instance data or accept the attachment to the selection set with a data point anywhere in the design file.

IFC Export

To open the TriForma IFC Export tool from Bentley Architecture, Bentley Structural, and Bentley Building Mechanical Systems, go to File > Export > IFC.



'Output' tab

IFC File

By default, the IFC export file name will be equal to the active DGN file name with extension .ifc and saved into the 'setting' folder of the Bentley Architecture dataset. It is possible to navigate to a different output folder and to provide a different name. The configuration variable TFDIR_IFC in the project's PCF-file allows defining another default folder for exported files, e.g. TFDIR_IFC = \$(_USTN_PROJECTDATA)out/

IFC File Info

The information given for System, Author, Organization, and Authorization will be written to the file header of the IFC export file.

IFC Schema

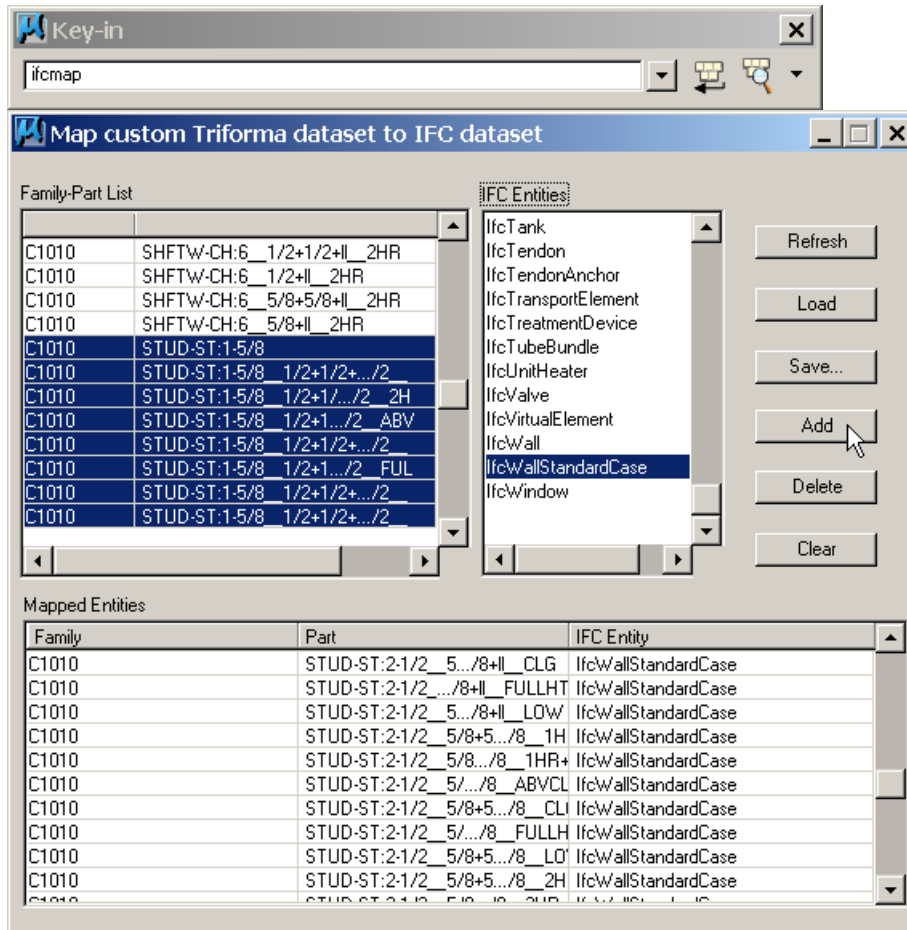
By pressing the button, the IFC2x3, IFC2x2, or IFC2x schema can be selected for export.

Mapping of Family/Parts to IFC entities

For export to IFC, Bentley Architecture family/parts are mapped to IFC entities and saved in an IFC-mapping file (ifcmapping.set). To add additional family/parts to the ifcmapping.set file:

- ☐ click the 'IFC Mapping Information' button to open the 'Map custom Triforma dataset to IFC dataset' form

Note: this form can also be opened with key-in 'ifcmap'

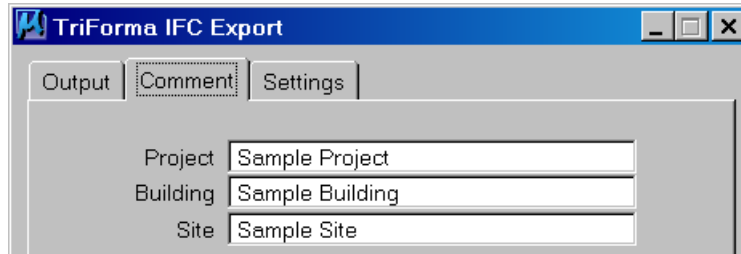


- ☐ select one or multiple family/parts from the Family-Part list
- ☐ select the corresponding IFC entity from the IFC Entities list
- ☐ click 'Add'
- ☐ click the 'Save' button to save the settings to the file ifcmapping.set. By default, this file is located in the 'setting' folder of the Bentley Architecture dataset. The folder location and file name can be changed with the configuration variable IFC_Part_Mapping in the PCF-file, e.g.
IFC_Part_Mapping = \$(_USTN_PROJECTDATA)out/ifcmapping.set

‘Comment’ tab

Project, Site, Building

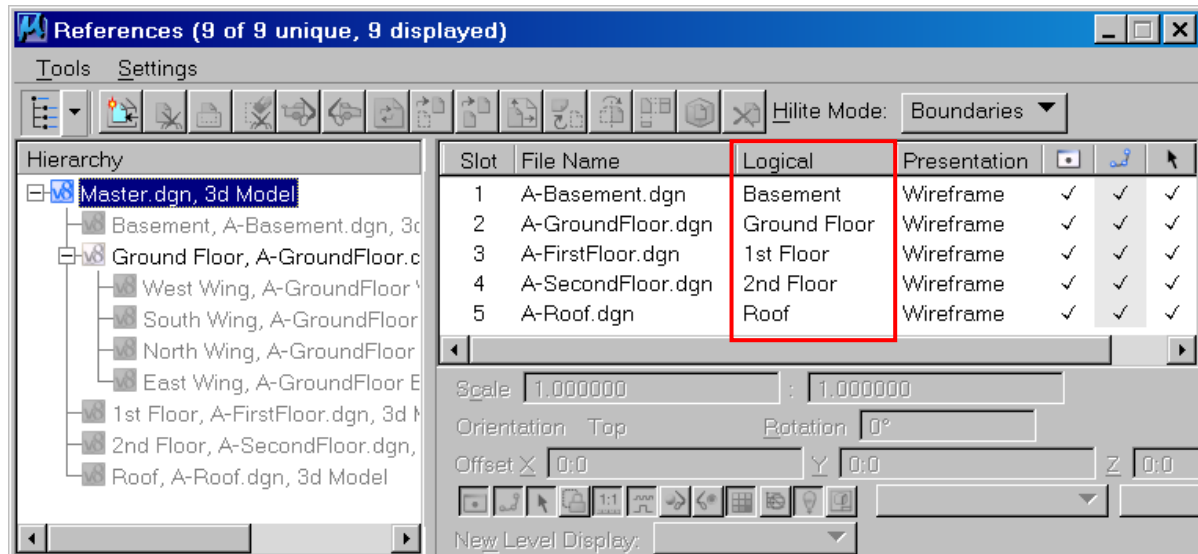
Information provided for ‘Project’, ‘Building’, and ‘Site’ is written to IfcProject, IfcSite, and IfcBuilding respectively, for instance:

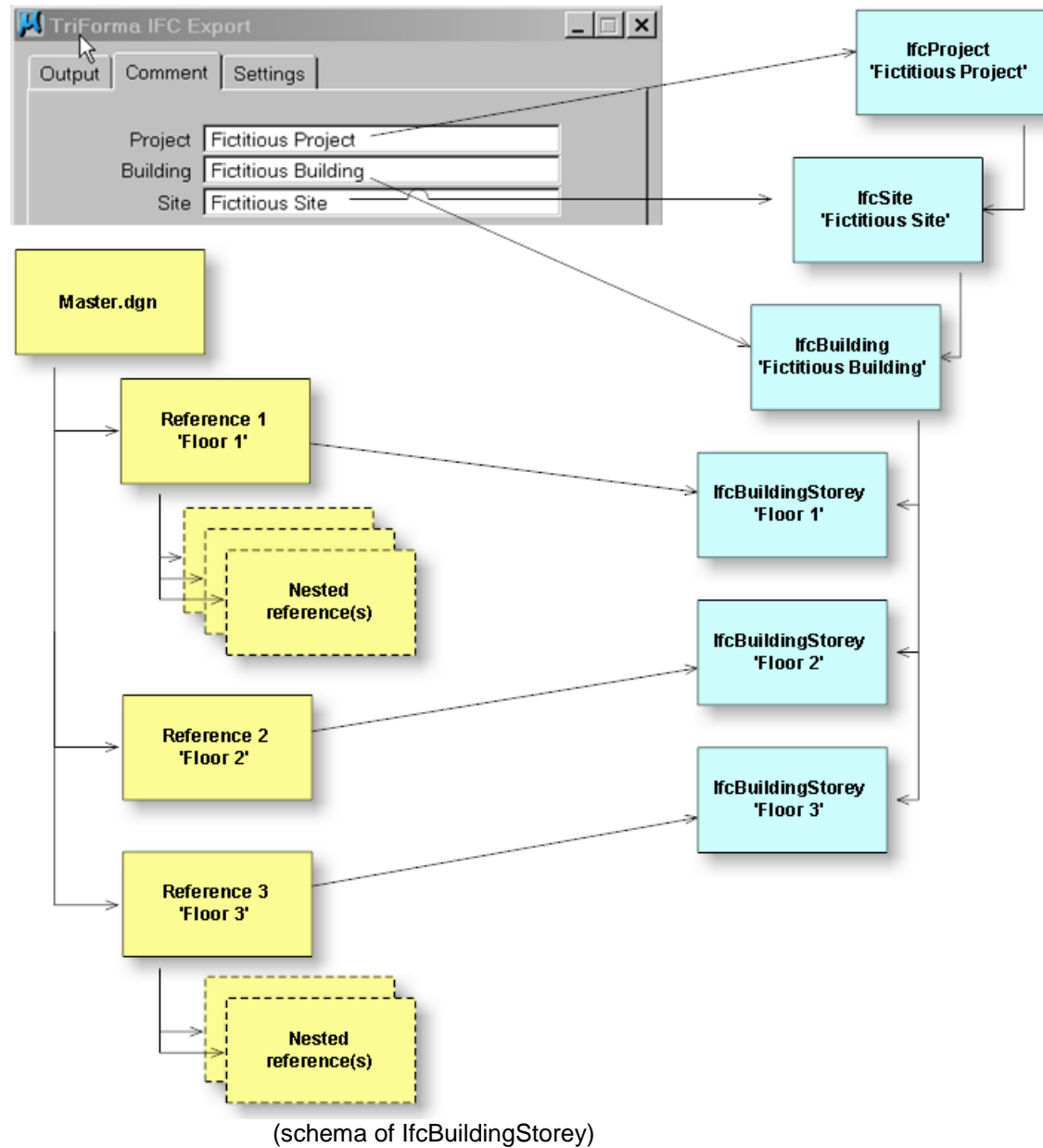
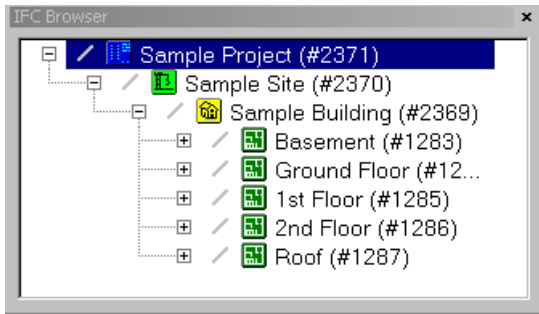


IFC Storey Containment

The logical name(s) of 1st level references are used for storey or floor names of IfcBuildingStorey. Therefore, for multi-storey buildings, files or models need to be attached as references to a (typically empty) master DGN-file, where the logical name is used as the IfcBuildingStorey name.

If a storey or floor is made up by multiple files or models, they must be attached as 2nd level references to a (typically empty) DGN-file, which in turn must be attached as 1st level reference to the master DGN-file. The logical name of this 1st level reference is used as IfcBuildingStorey name for all the attached 2nd level references.





‘Settings’ tab

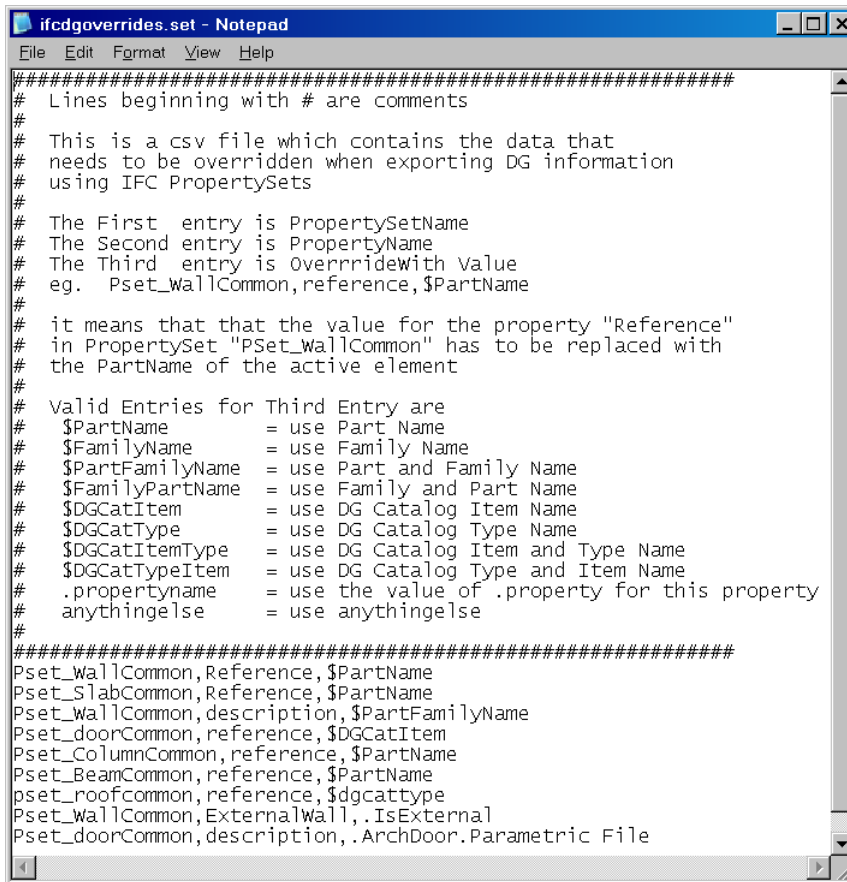
The screenshot shows the 'TriForma IFC Export' dialog box with the 'Settings' tab selected. The dialog has three tabs: 'Output', 'Comment', and 'Settings'. The 'Settings' tab contains two sections: 'Tolerance' and 'Additional Settings'. In the 'Tolerance' section, there are two input fields: 'Max Facet Length' with a value of '0.00' and 'Tolerance' with a value of '1000.0'. Below these is a checkbox labeled 'Compress (Slow!)' which is currently unchecked. The 'Additional Settings' section contains three checkboxes: 'Generate 3D Spaces from 2D Spaces' (checked), 'Override DataGroup Values from Setting File' (checked), and 'Zip Output File' (unchecked). At the bottom of the dialog is an 'Export' button.

Generate 3D Spaces from 2D Spaces

If checked, the 2D Space boundaries of Bentley Architecture will be converted to 3D spaces in the IFC export file using the values of the DataGroup property 'Ceiling Height'.

Override DataGroup Values from Settings File

If checked, certain Family and Part names or DataGroup property values are written to IfcProperty values of IFC2x3 property sets via mapping in the file ifcdgoverrides.set, which is typically located in the 'setting' folder of the Bentley Architecture dataset. An explanation and valid values are giving at the top of the file.



```
#####
# Lines beginning with # are comments
#
# This is a csv file which contains the data that
# needs to be overridden when exporting DG information
# using IFC PropertySets
#
# The First entry is PropertySetName
# The Second entry is PropertyName
# The Third entry is OverrideWith value
# eg. Pset_WallCommon,reference,$PartName
#
# it means that that the value for the property "Reference"
# in PropertySet "Pset_WallCommon" has to be replaced with
# the PartName of the active element
#
# Valid Entries for Third Entry are
# $PartName      = use Part Name
# $FamilyName    = use Family Name
# $PartFamilyName = use Part and Family Name
# $FamilyPartName = use Family and Part Name
# $DGCatItem     = use DG Catalog Item Name
# $DGCatType     = use DG Catalog Type Name
# $DGCatItemType = use DG Catalog Item and Type Name
# $DGCatTypeItem = use DG Catalog Type and Item Name
# .propertyname  = use the value of .property for this property
# anythingelse   = use anythingelse
#####
Pset_WallCommon,Reference,$PartName
Pset_SlabCommon,Reference,$PartName
Pset_WallCommon,description,$PartFamilyName
Pset_doorCommon,reference,$DGCatItem
Pset_ColumnCommon,reference,$PartName
Pset_BeamCommon,reference,$PartName
pset_roofcommon,reference,$dgcattype
Pset_WallCommon,ExternalWall,.IsExternal
Pset_doorCommon,description,.ArchDoor.Parametric File
```

For instance, if the following entries in ifcdgoverrides.set are ...

Pset_WallCommon,Reference,\$PartName
Pset_WallCommon,description,\$PartFamilyName

... the following property values for Reference and Description are written to IfcWalls respectively:

Properties	
Element Properties IFC Properties IFC Relations	
Name	Value
pset_wallcommon	
Reference	BIMSample Exterior Metal Stud
AcousticRating	
FireRating	1 HR
Combustible	FALSE
SurfaceSpreadOfFlame	
ThermalTransmittance	0.
IsExternal	TRUE
ExternalWall	TRUE
ExtendToStructure	FALSE
LoadBearing	FALSE
Compartmentation	FALSE
Description	BIMSample Exterior Metal Stud : C1010_BIMSample
Reference	BIMSample Brick
AcousticRating	
FireRating	2 HR
Combustible	FALSE
SurfaceSpreadOfFlame	
ThermalTransmittance	2.28
IsExternal	TRUE
ExternalWall	TRUE
ExtendToStructure	FALSE
LoadBearing	TRUE
Compartmentation	FALSE

IfcExport

Click the 'Export' button to start the IFC export process. All elements in the active file and attached references, whether displayed or not, will be exported. To export selected elements or levels only, create a selection set of all elements to be exported.

Export message log

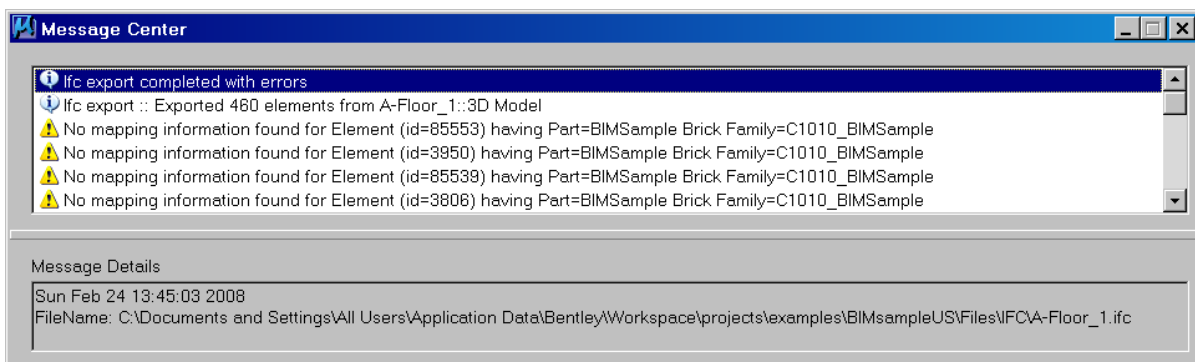
If no errors are encountered, the message 'Ifc export completed successfully' is displayed in the message field.



If errors are encountered, a message 'IFC export completed with errors' is displayed in the message window.



The message log file can be opened by clicking the 'I' button.



This information is particularly useful to identify Families and Parts that have not been mapped to IfcEntities, thus have been exported as IfcBuildingElementProxy entities. Open the 'Map custom TriForma dataset to IFC dataset' tool to map the missing families/parts to IfcEntities, then export again.

To save the message file right-click the message window and select the 'Save Messages...' option.



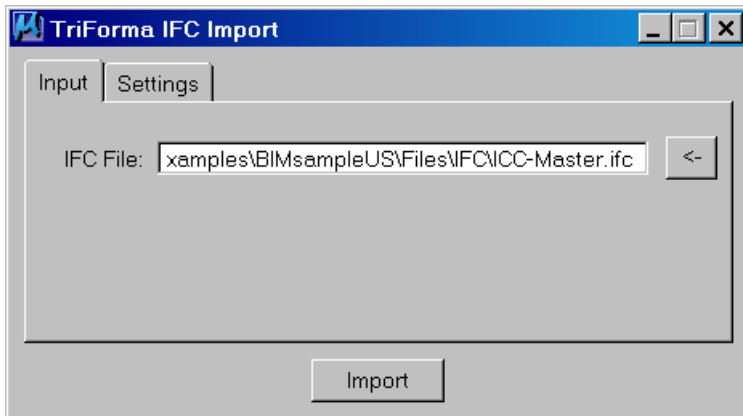
To change the properties of the message file, e.g. the maximum number of messages per file, right-click the message window and select the 'Properties' option.



IFC Import

To open the TriForma IFC Import tool from Bentley Architecture, Bentley Structural, and Bentley Building Mechanical Systems, go to File > Import > IFC.

'Input' tab

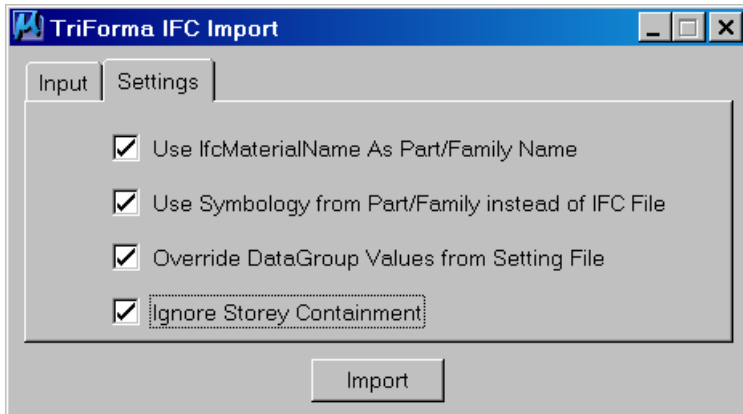


IFC File

If the desired IFC-file and folder is not displayed in the 'IFC File' window, navigate to the output folder, click on the file name, and click 'Open'. With the configuration variable TFDIR_IFC in the project's PCF-file a default folder can be defined, e.g.

TFDIR_IFC = \$(_USTN_PROJECTDATA)out/

'Settings' tab



Use IfcMaterialName As Part/Family Name

The option 'Use IfcMaterialName As Part/Family Name' determines, which information is applied as Part to imported elements:

- ☐ if checked, the values of the 'Material' property in the ifc-import file are applied as 'Part' name to imported elements,

- ☐ if unchecked, the Parts corresponding to the IfcEntity in the 'IFC' family of the Bentley Architecture dataset are applied to imported elements.

For instance, for IfcWalls with Material value 'P30N32' in the ifc-import file ...

Material	
Material (LayerSet)	Wall: Insitu P30N32 200.000
Material Layers	
1. Layer	
Width	200.000000
Material	P30N32
IsVentilated	False

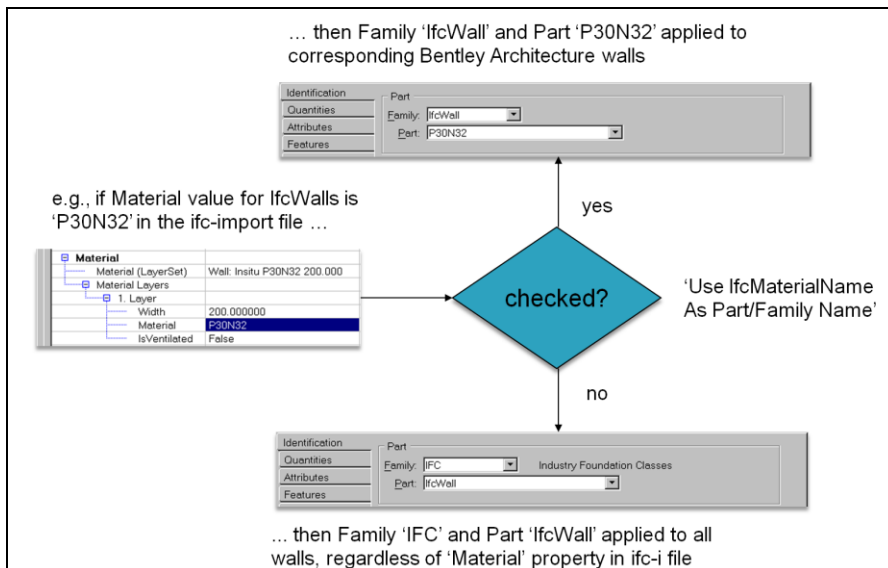
... Family 'IfcWall' and Part 'P30N32' will be applied to the corresponding Bentley Architecture walls, if the option is checked.

Identification	Part
Quantities	Family: IfcWall
Attributes	Part: P30N32
Features	

However, if the option is unchecked, Family 'IFC' and Part 'IfcWall' will be applied to all walls, regardless of the 'Material' property in the ifc-import file.

Identification	Part
Quantities	Family: IFC Industry Foundation Classes
Attributes	Part: IfcWall
Features	

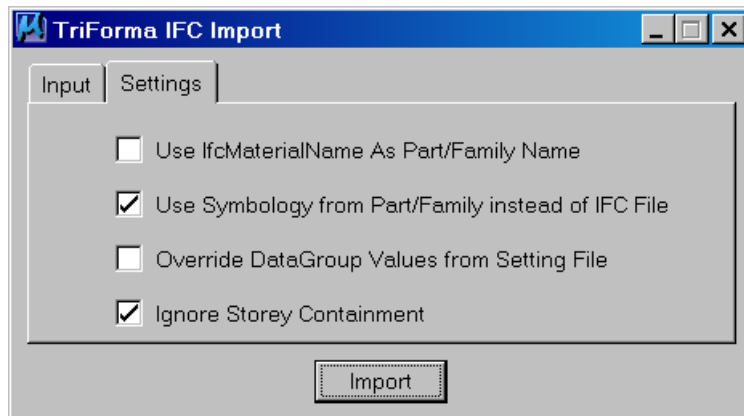
Note: The 'IFC' family is supplied with every Bentley Architecture dataset, either in imperial (ifc_imp_parts.xml) or metric (ifc_si_parts.xml).



Use Symbology from Part/Family instead of IFC File

The option 'Use Symbology from Part/Family instead of IFC File' determines, which symbology is applied to imported elements:

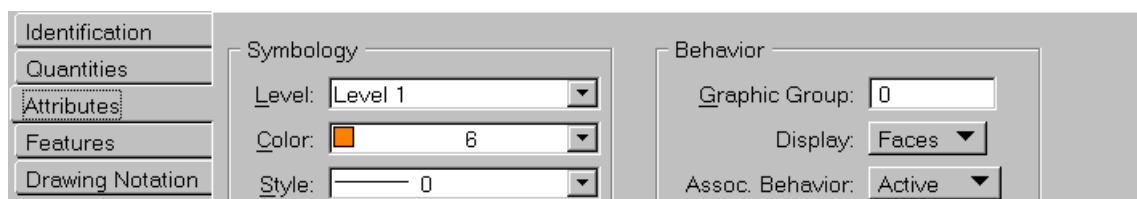
- ☐ if checked, the symbology specified for the Part corresponding to the IfcEntity in the 'IFC' Family of the Bentley Architecture dataset is used,
- ☐ if unchecked, the symbology in the ifc-import file is used for imported elements.



For instance, if the imported element is a wall, the symbology specified for Part 'IfcWall' in Family 'IFC' is used, if the option is checked.

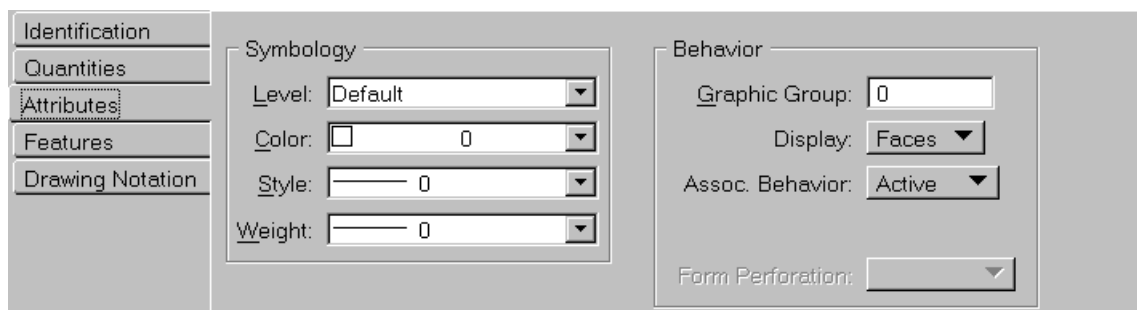
Part ▲	Description	Level	Color	Style	Weight
IfcWall		Level 1	6	0	0
IfcWallStandardCase		Level 1	6	0	0
IfcWindow		Level 1	5	0	0

(Part setting in the Dataset Explorer)

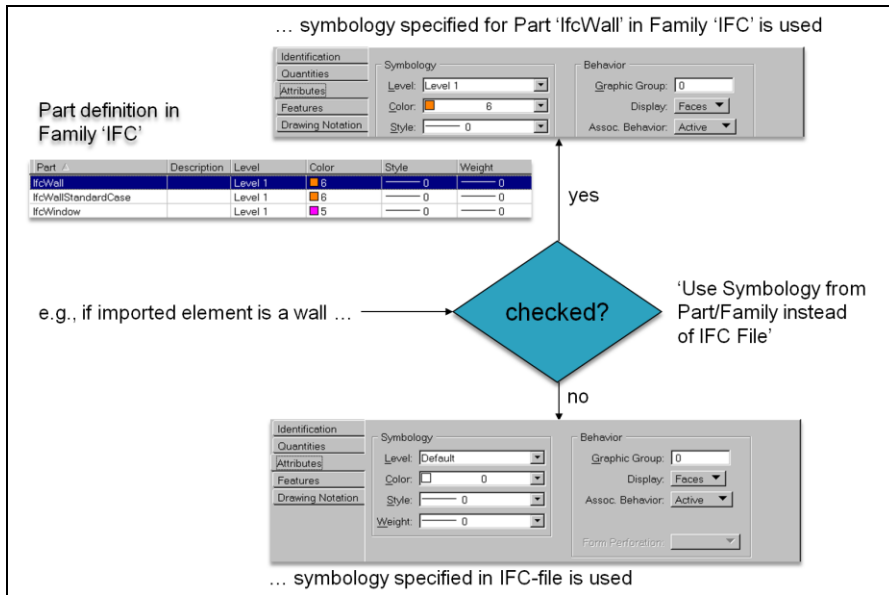


(Symbology information of imported element)

If unchecked, the symbology specified in the IFC-import file is used.

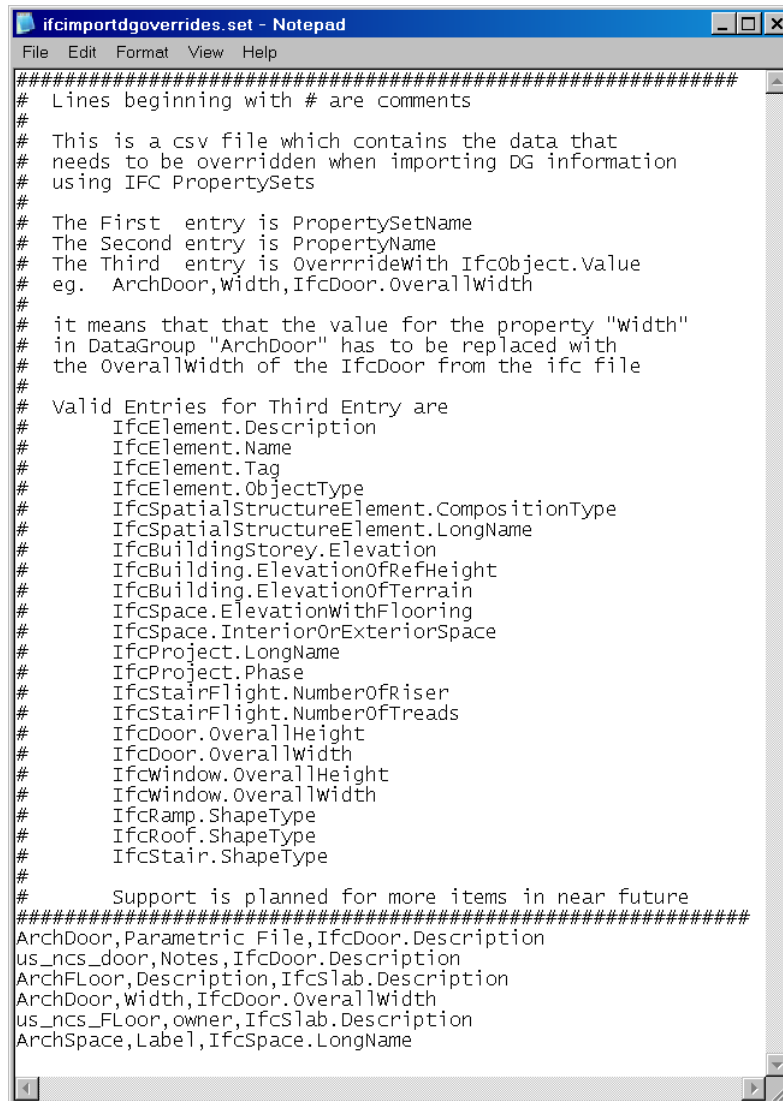


Note: The 'IFC' family is supplied with every Bentley Architecture dataset, either in imperial (ifc_imp_parts.xml) or metric (ifc_si_parts.xml).



Override DataGroup Values from Setting File

If checked, certain IFC property values are written to Bentley Architecture DataGroup properties via mapping in the file ifcimportdgooverrides.set, which is typically located in the 'setting' folder of the Bentley Architecture dataset. An explanation and valid values is given at the top of the file.

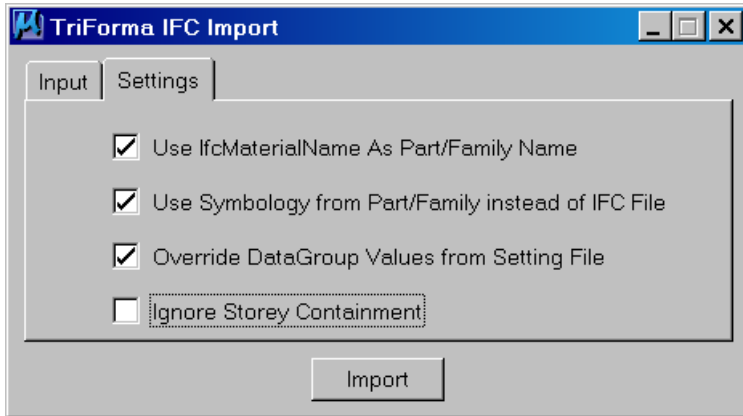


```
#####
# Lines beginning with # are comments
#
# This is a csv file which contains the data that
# needs to be overridden when importing DG information
# using IFC PropertySets
#
# The First entry is PropertySetName
# The Second entry is PropertyName
# The Third entry is OverrideWith IfcObject.Value
# eg. ArchDoor,Width,IfcDoor.OverallWidth
#
# it means that that the value for the property "width"
# in DataGroup "ArchDoor" has to be replaced with
# the OverallWidth of the IfcDoor from the ifc file
#
# Valid Entries for Third Entry are
#   IfcElement.Description
#   IfcElement.Name
#   IfcElement.Tag
#   IfcElement.ObjectType
#   IfcSpatialStructureElement.CompositionType
#   IfcSpatialStructureElement.LongName
#   IfcBuildingStorey.Elevation
#   IfcBuilding.ElevationOfRefHeight
#   IfcBuilding.ElevationOfTerrain
#   IfcSpace.ElevationWithFlooring
#   IfcSpace.InteriorOrExteriorSpace
#   IfcProject.LongName
#   IfcProject.Phase
#   IfcStairFlight.NumberOfRiser
#   IfcStairFlight.NumberOfTreads
#   IfcDoor.OverallHeight
#   IfcDoor.OverallWidth
#   IfcWindow.OverallHeight
#   IfcWindow.OverallWidth
#   IfcRamp.ShapeType
#   IfcRoof.ShapeType
#   IfcStair.ShapeType
#
# Support is planned for more items in near future
#####
ArchDoor,Parametric File,IfcDoor.Description
us_ncs_door,Notes,IfcDoor.Description
ArchFloor,Description,IfcSlab.Description
ArchDoor,Width,IfcDoor.OverallWidth
us_ncs_Floor,owner,IfcSlab.Description
ArchSpace,Label,IfcSpace.LongName
```

Ignore Storey Containment

If checked, all data will be imported into the active DGN model, i.e. Storey Containment information in the IFC import file will be ignored.

If unchecked, separate DGN models will be created for each IfcBuildingStorey found in the IFC import file, the data written to them, and the DGN models will be referenced to the active DGN model



IfcImport

Click the 'Import' button to start the IFC import process.

Appendix

Configuration Variables

Name of config. variable	Associated value	Use
TFDIR_IFC	defines a folder	For import: IFC-files are searched in the folder defined by this variable
		For export: IFC-files are generated in the folder defined by this variable
		Note: IFC Module will not load if this config. variable is not defined
TFIFC_NOVERTICAL	0 or 1	By default, a vertical application, such as Bentley Architecture, Bentley Structural, and Bentley Building Mechanical Systems is required to access the IFC Functionality
		By setting this config. variable to 1, IFC functionality can be used even if there is no vertical application
IFC_VERSION	'2x', '2x2', or '2x3'	By default, the export schema in the IFC-export dialog box is set to '2x'
		By setting this variable to any of the given values, the default export schema can be changed
IFC_PART_MAPPING	defines a file	Path(s) and file name(s) of the ifcmapping.set file to be used
		If this variable is not defined, then ifcmapping.set defined by config. variable TFDIR_SETTING is used
IFC_PROPERTY_OVERRIDES	defines a folder	Folder in which the ifcimportdgooverrides.set and ifcdgooverrides.set files are used
		If this variable is not defined, then the folder defined by TFDIR_SETTING is used
IFC_DONT_FIT_VIEWS	0 or 1	When an IFC-file is imported, all the views are fit, to show the entire model
		By setting this variable to 1, the original window view is retained
TFIFC_NO_DATAGROUP	0 or 1	When an IFC-file is imported or exported, DataGroup data is exported or imported as ifcpropertysets
		By setting this variable to 1, DataGroup data is not exported or imported
TFIFC_PREFS	defines a folder	Folder where the IFC preferences are stored: if this variable is not defined, then the folder defined by TFDIR_PREFS is used
		If TFDIR_PREFS is not defined, then the folder

		defined by config. variable _USTN_HOMEPREFS is used
IFC_ZIP_EXTRACT_DIR	defines a folder	When importing a zipped IFC-file, this config. variable defines the folder where the contents of the ZIP-file is extracted
		If this config. variable is not defined, then the ZIP-file is extracted in the folder defined by config. variable MS_TMP
IFC_DONT_USE_DGGUID	0 or 1	for export: by default, the DataGroup GUID (Global Unique Identifier) is used as the IFCGUID for IFC Property Sets
		By setting this config. variable to 1, GUIDs are created on the run rather than using the DataGroup GUIDs