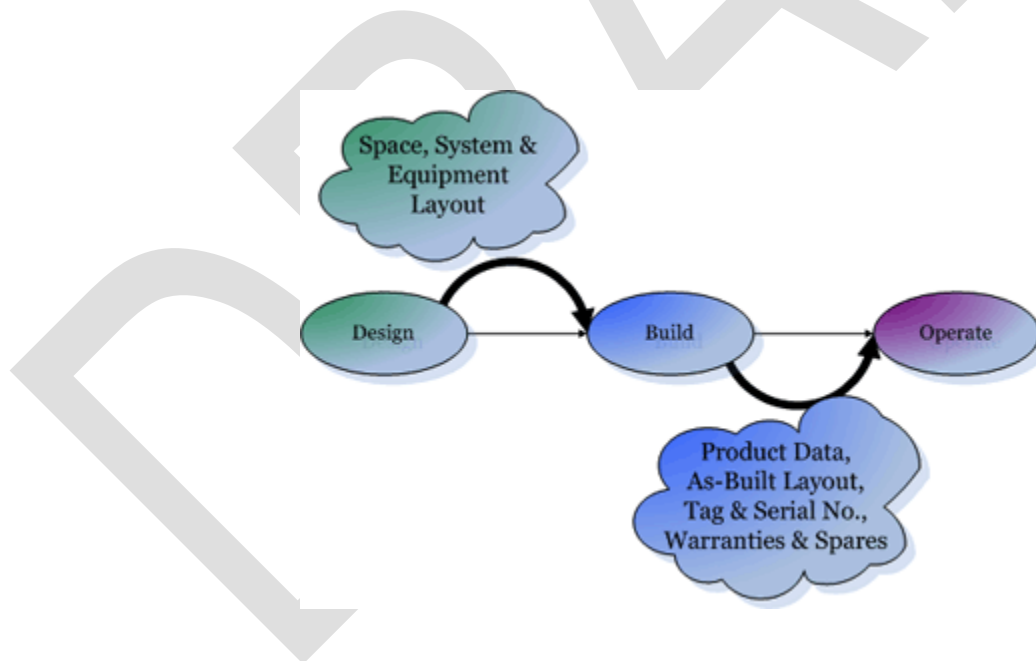


# COBie User Guide

revision: 15 Mar, 2013 - 05:10 AM



Note: This document is likely to change slightly for the released product.

# Table of Contents

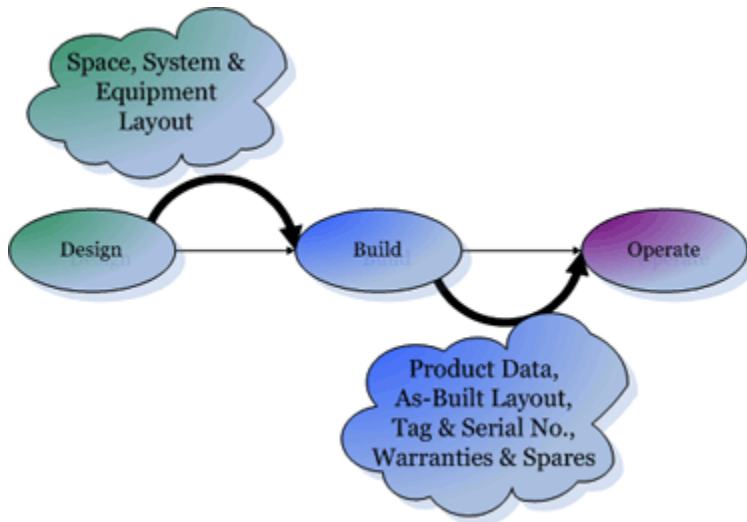
1	COBie.....	4
2	IFC concepts .....	5
2.1	Type vs. Instance .....	5
2.2	Shared vs. private properties .....	5
2.3	'Strong' vs. 'soft' typing .....	6
3	AECOSim Building Designer .....	8
3.1	Product overview .....	8
3.2	Datasets .....	8
3.2.1	Delivered datasets .....	8
3.2.1.1	Terminology IFC / Building Designer .....	8
3.2.1.2	Creation of 'strong' catalog items (types) .....	9
3.2.2	Dataset Extension .....	16
3.2.3	Company dataset .....	16
3.2.4	Project dataset .....	17
4	Workspace operations .....	18
4.1	Create project .....	18
4.2	Upgrade Projects to COBie .....	18
4.2.1	IFC_PROJECT=0 .....	19
4.2.2	IFC_PROJECT=1 .....	19
4.2.3	IFC_PROJECT=2 .....	20
4.3	Start AECOSim Building Designer .....	21
4.4	Create design file(s) .....	21
4.5	Open design file .....	22
4.6	Create custom DataGroup catalog type(s) .....	22
4.7	Create custom DataGroup property definitions .....	25
4.8	Attach DataGroup types with DataGroup definitions .....	30
4.9	Create a compound cell .....	33
4.10	Create DataGroup catalog items .....	35
4.11	Create DataGroup Catalog Items for IfcZones .....	40
4.12	Map DataGroup to IFC .....	42
4.12.1	Map DataGroup types to IFC Entities/Classes .....	42
4.12.2	Map Parts to IFC Entities/Classes .....	45
4.12.3	Map DataGroup items to IFC Entities/Classes .....	45
4.12.4	Map duplicate properties .....	46
5	Building Information Modeling .....	51
5.1	Define the Spatial Structure .....	51
5.2	Add Zone information to Spaces .....	52
5.3	Assemble the Building Information Model .....	53

6	IFC Export .....	55
6.1	Assign Buildings and Floors .....	57
7	IFC Transformation Utility .....	61

DRAFT

# 1 COBie

Based on the IFC 'Facility Management Handover' Model View Definition, COBie (Construction Operations Building information exchange) is an information exchange specification to capture data during design, construction, and commissioning for handover to operations and management. Data is created by architects and engineers, who provide floor, space, and equipment layouts, and contractors, who provide make, model, and serial numbers of installed equipment, possibly provided by product manufacturers.



COBie data can be delivered in 3 formats: IFC STEP Physical File Format (SPFF), ifcXML, and spreadsheets.

Further information at:

<http://www.wbdg.org/resources/cobie.php>

<http://buildingsmartalliance.org/index.php/projects/cobieguide/>

[http://www.nationalbimstandard.org/nbims-us-v2/pdf/NBIMS-US2\\_c4.2.pdf](http://www.nationalbimstandard.org/nbims-us-v2/pdf/NBIMS-US2_c4.2.pdf)

## 2 IFC concepts

Specified and developed by buildingSMART, IFC (Industry Foundation Classes) is an open, vendor-neutral BIM data repository for the semantic information of building objects, including geometry, associated properties, and relationships, to facilitate

- ❑ cross-discipline coordination of building information models, e.g. architecture, structural and building services,
- ❑ data sharing and exchange across IFC-compliant applications,
- ❑ handover and re-use of data for analysis and other downstream tasks.

Further information at <http://www.buildingsmart.org/standards/ifc/model-industry-foundation-classes-ifc>

### 2.1 Type vs. Instance


buildingSMART's IFC (Industry Foundation Classes), COBie, and most BIM applications distinguish between 'Types' and 'Instances':

Type	typically stored in a library; defines 'shared' properties for a specific component type, e.g. 'SingleTimberDoorF60_3x7' (type: single swing door; material: timber; fire rating: F60; size: 3' x 7'; color: blue);
Instance	typically stored in building information models; holds the 'shared' and possibly 'private' properties of a single occurrence of the component type, e.g. 'SingleTimberDoorF60_3'x7' in Room #001' (type: single; material: timber; fire rating: F60; size: 3' x 7'; door no: D001, sound proof: true; color: red);  inherits some or all 'shared' properties of its parent type, but can also have additional 'private' properties (door no: D001; sound proof: true) and/or redefined 'shared' properties that then become 'private' properties (color: red).

### 2.2 Shared vs. private properties

buildingSMART's IFC (Industry Foundation Classes), COBie, and most BIM applications distinguish between 'shared' and 'private' properties:

shared	defined by a Type, i.e. shared or common values across multiple instances
private	defined at the instance, i.e. unique property values for each instance of that type, which <ul style="list-style-type: none"> <li>• add to the shared property values</li> <li>• or redefine or override the shared property values</li> </ul>

 Property values redefined on the instance override the property values defined by the type.

Therefore, an instance can have (in order of 'strong' to 'soft' typing)

- a) only shared property values inherited from its type
- b) shared property values inherited from its type, plus additional private property values
- c) shared property values inherited from its type, some of which redefined on the instance
- d) shared property values inherited from its type, some of which redefined on the instance, plus additional private property values
- e) only private property values

	Type	Instance 1		Instance 1	
Properties	'SingleTimberDoorF60_3x7'	'SingleTimberDoorF60_3x7-R001'		'SingleTimberDoorF60_3x7-R002'	
Type	Single Swing Door	Single Swing Door	shared	Single Swing Door	shared
Material	Timber	Timber	shared	Timber	shared
Fire rating	F60	F61	shared	F61	shared
Size	3x6	3x7	shared	3x7	shared
Door no.		D001	added	D002	added
Sound proof		true	added	false	added
Color	blue	red	redefined	green	redefined

IFC Viewers, such as the 'FZK Viewer', may display 'shared' properties as 'PropertySets from entity type' and 'private' properties as 'PropertySets from entity'

PropertySets from entity type		PropertySets from entity	
<ul style="list-style-type: none"> <li>ePset_Door <ul style="list-style-type: none"> <li>Weatherstripping: FALSE</li> <li>Shielded: FALSE</li> <li>SoundProof: FALSE</li> <li>Shield: FALSE</li> <li>LightProof: FALSE</li> </ul> </li> <li>Pset_Warranty</li> <li>Pset_ManufacturerTypeInformation <ul style="list-style-type: none"> <li>ModelReference: Standard Fib-R-Door</li> </ul> </li> <li>Pset_ManufacturerOccurrence</li> <li>Pset_DoorCommon</li> <li>Pset_Asset <ul style="list-style-type: none"> <li>AssetAccountingType: Fixed</li> </ul> </li> <li>Door_PAZ</li> <li>Door <ul style="list-style-type: none"> <li>Frame   Material: STL</li> <li>Louver: FALSE</li> <li>Door   IsEgress: FALSE</li> <li>Door   Under Cut: 0.</li> <li>Door   Material: HC-WD</li> <li>Door   Type: Door Type A</li> <li>Glazing: FALSE</li> </ul> </li> </ul>		<ul style="list-style-type: none"> <li>Pset_ManufacturerTypeInformation</li> <li>Pset_DoorCommon <ul style="list-style-type: none"> <li>FireExit: FALSE</li> </ul> </li> <li>ePset_Door <ul style="list-style-type: none"> <li>PaintColor: To Match</li> <li>PaintNo: DP3</li> <li>RoomName: PH TECH</li> </ul> </li> <li>Pset_Warranty</li> <li>Pset_ServiceLife</li> <li>Pset_ManufacturerOccurrence</li> <li>Pset_Asset</li> <li>Fenestration <ul style="list-style-type: none"> <li>Frame Detail   Head: H1</li> <li>Frame Detail   Jamb: J1</li> </ul> </li> <li>Door Dimensions</li> <li>Door <ul style="list-style-type: none"> <li>Frame   Material: WD</li> <li>Frame   Type: S</li> <li>Door   Under Cut: 0.</li> <li>Door   Finish: DP3</li> <li>Door   Material: SC-WD</li> <li>Door   Type: F - Flush</li> </ul> </li> <li>Door_PAZ</li> <li>BaseQuantities</li> </ul>	

☞ 'PropertySets from entity type' displays all property values defined for the type (e.g. Frame | Material: STL), even if redefined in 'PropertySets from entity' (e.g. Frame | Material: WD)

## 2.3 'Strong' vs. 'soft' typing

BIM applications support either 'strong', 'soft', or a combination/degree of 'strong' and 'soft' typing:

- 'strong' requires the creation of a new type if any property is changed;  
more applicable during the construction documentation and product specification phase;  
supports a) and possibly b)
- 'soft' does not necessarily require the creation of a new type if a property is changed;  
it is up to the user to decide which property changes require a new type;  
more applicable during the conceptual design and design development phase;  
supports c), d), e).

☞ The IFC specification does not define whether a change of property requires a change of a type or which properties must be shared by all instances.

☞ Although not defined in the FM Handover Model View Definition specification, COBie has a preference for 'strong' typing.

DRAFT

## 3 AECOSim Building Designer

### 3.1 Product overview

AECOSim Building Designer V8i is a single, interdisciplinary Building Information Modeling application that combines architectural, structural, mechanical, and electrical design and construction documentation into a single consistent application. It empowers architects and engineers to design sustainable infrastructure and enables project teams to deliver well-designed and high performing buildings while taking advantage of the many benefits associated with BIM.


Supporting the design of large, complex, and distributed models, AECOSim Building Designer V8i includes clash detection capabilities for design review and constructability and provides integrated visualization with the Luxology rendering engine.

### 3.2 Datasets

Datasets add the 'I' to 'BIM', i.e. the *Information* to Building *Information* Modeling, powering the AECOSim Building Designer engine. Without datasets, the application is not able to create a building information model and to generate drawings, rendered images and animations, reports, schedules, analysis data and other output, which depend on different inputs from the datasets

#### 3.2.1 Delivered datasets

AECOSim Building Designer is delivered with an imperial dataset for use in the US. Metric datasets can be downloaded from Bentley's SELECT site that included local standards for a variety of countries, such as the UK, Australia, New Zealand, Singapore, Sweden, and Denmark.

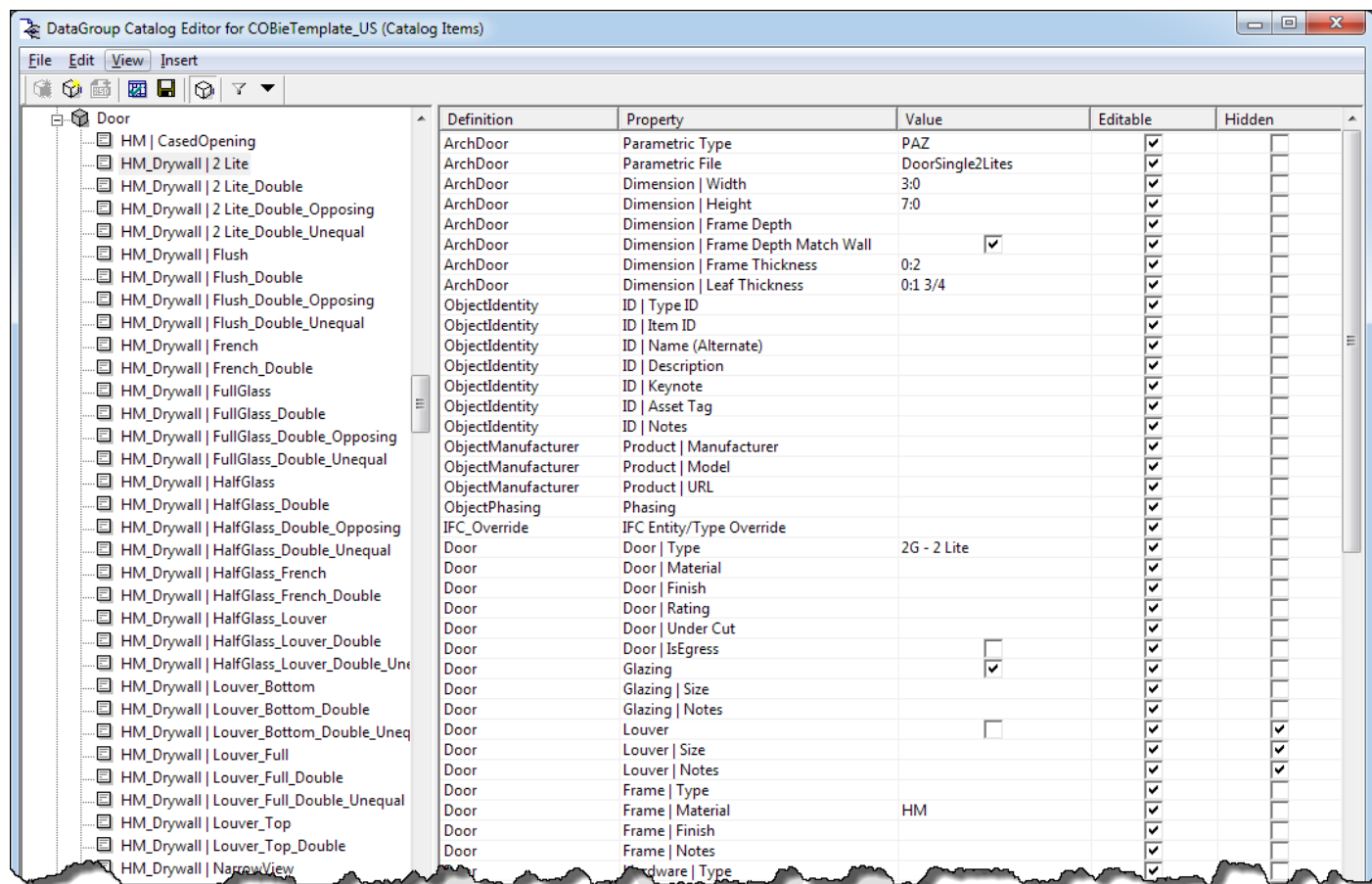
 The delivered datasets should not be modified by users as changes may be overwritten by subsequent installations or new content may not be included in modified files.

##### 3.2.1.1 Terminology IFC / Building Designer

IFC Class	DataGroup Catalog Type e.g. 'Door', 'Window', 'Column', 'Pump', 'Fan', 'LightFixture', ...
IFC Type	DataGroup Catalog Item e.g. 'SingleTimberDoorF60_3x7', 'SingleMetalDoorF60_3x7', 'DoubleTimberDoorF60_6x7', 'DoubleMetalDoorF60_6x7', ...

By default, delivered datasets support 'soft' typing, i.e. DataGroup Catalog Items only define certain property values, all of which could be redefined at the instance of that Catalog Item, because all properties are 'editable':





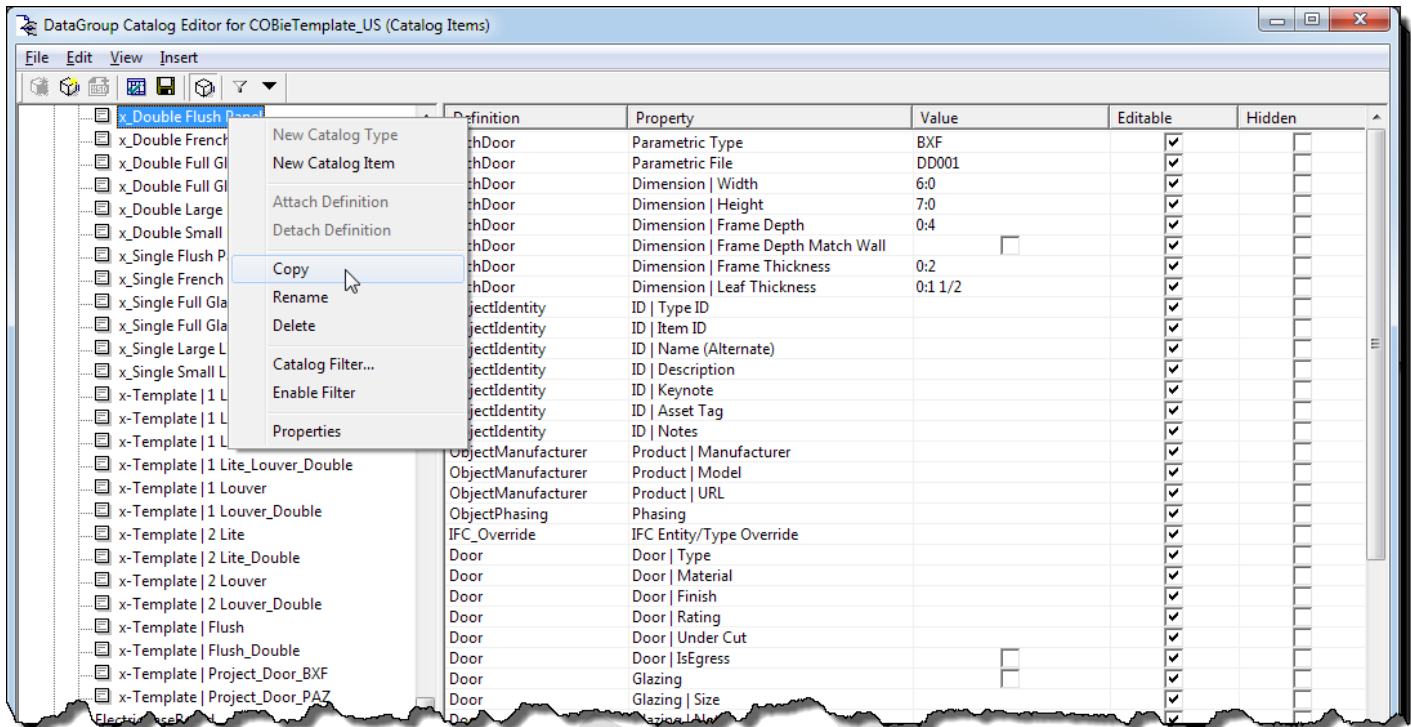
☞ 'Typing' can be made strong by making some or all properties 'non-editable'.

### 3.2.1.2 Creation of 'strong' catalog items (types)

#### 3.2.1.2.1 Architectural and Structural

- ☐ Open the DataGroup Catalog Editor
- ☐ Make a copy of a DataGroup Catalog Item using right-click option 'Copy' or 'Edit' > 'Copy' (Ctrl C).

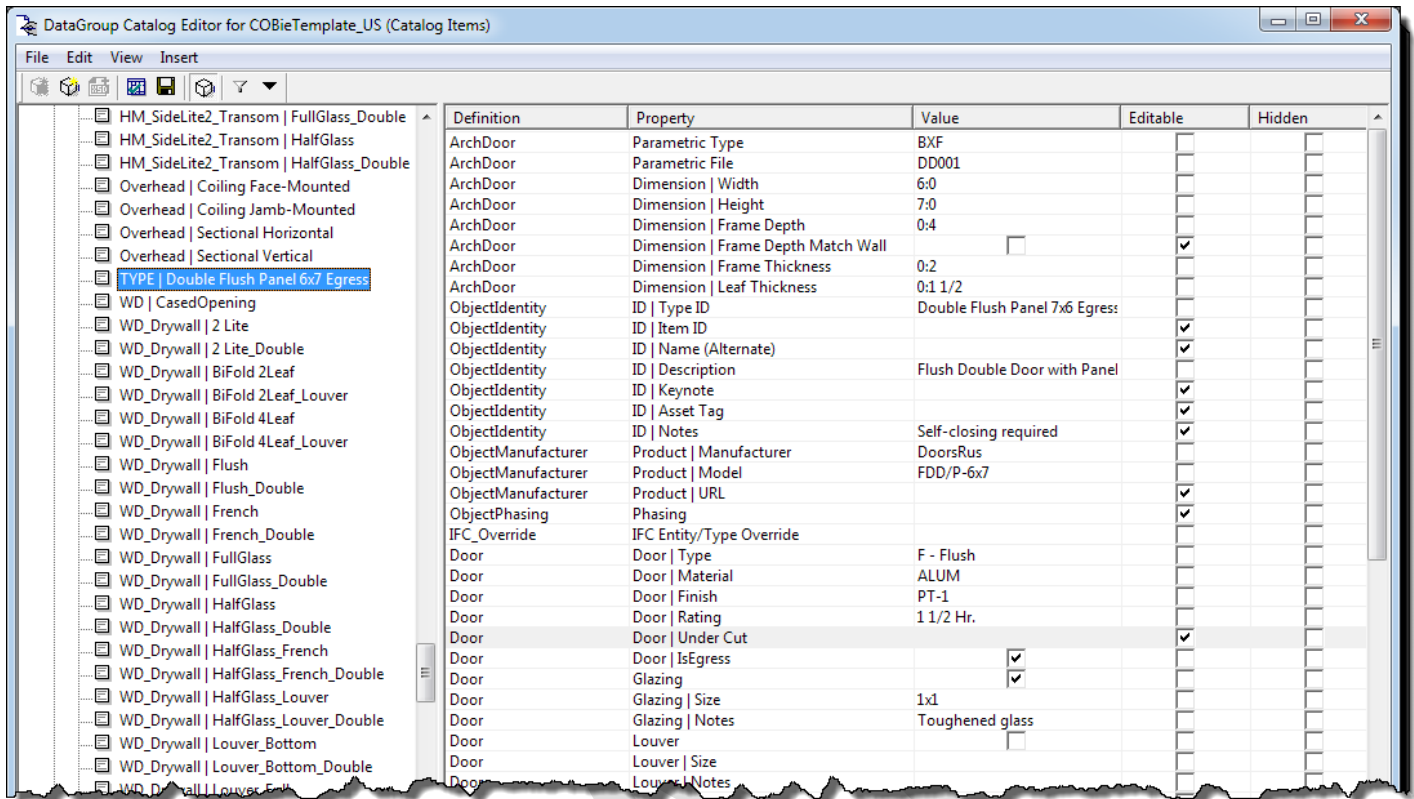
☞ Catalog items, typically prefixed with 'x\_' or 'x-template', are templates that when copied the copies are created in the project dataset.



- ☐ Rename the copy

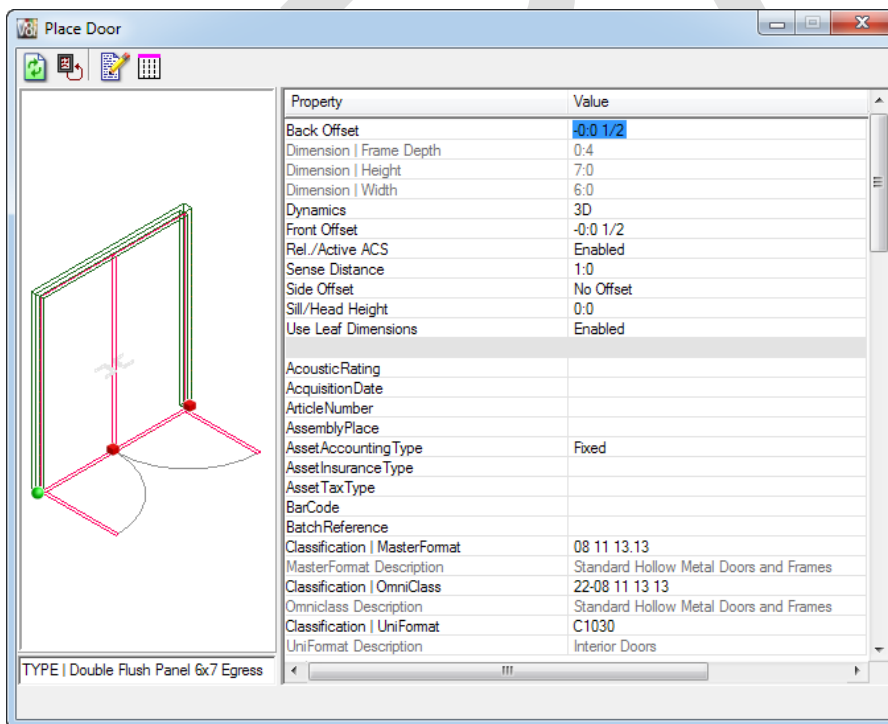
Consider a naming convention that indicates a strong type by adding TYPE as a pre- or suffix, e.g. 'TYPE | Double Flush Panel 6x7 Egress'.

- ☐ Enter values for all properties relevant for the type.
- ☐ Make all properties relevant for the type 'non-editable', but leave properties that can be added or redefined for the instance 'editable'.

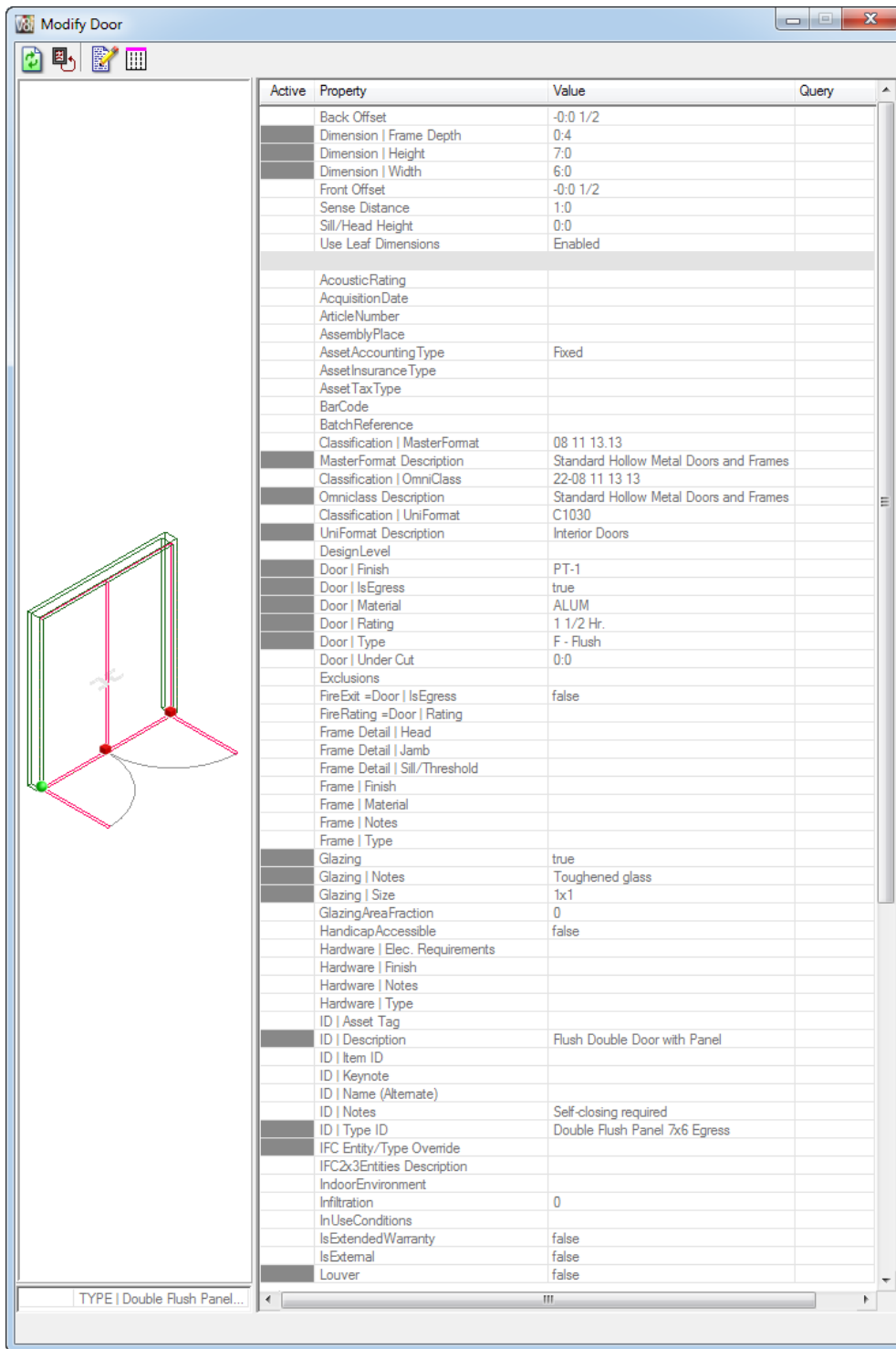


❑ Save the newly created type.

👉 When placing a 'strongly typed' component, 'non-editable' properties are greyed out, i.e. cannot be redefined for the instance, while 'editable' properties can be added or redefined.

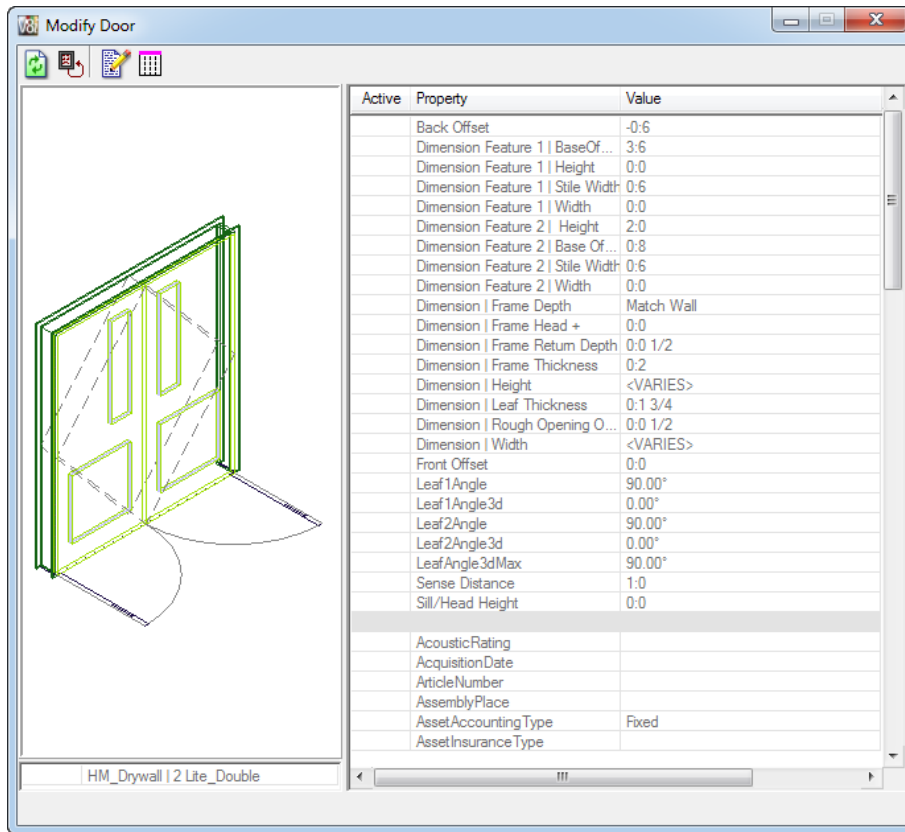


Instance data of 'non-editable' properties are locked, i.e. cannot be activated for editing or redefined, while editable properties can be added or redefined.



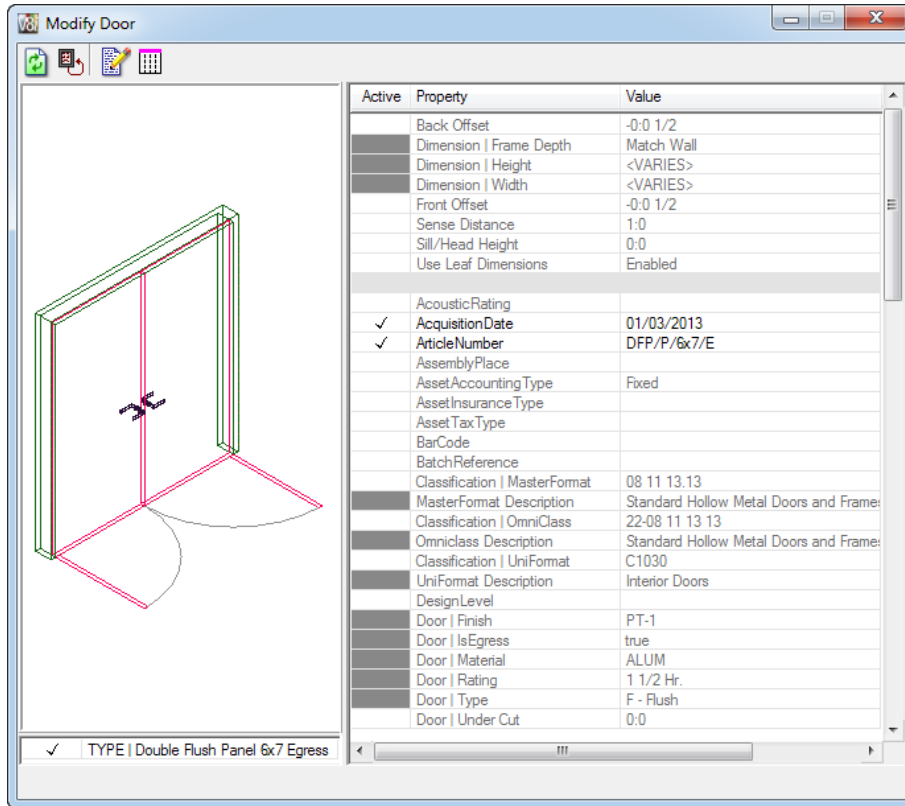
To change soft-typed instances, e.g. used during the early design stages, to strong typed catalog for use in the construction documentation phase:

- ❑ Select one instance or create a selection set of instances that should be changed to the same strong-typed catalog item and click 'Modify Properties of Selected Components' to open the 'Modify' dialog box.



- ❑ Activate the DataGroup Catalog Item field (bottom left) by adding a ✓-mark to the 'Active' field.
- ❑ From the pulldown menu, select the strong-typed catalog item, e.g. 'TYPE | Double Flush Panel 6x7 Egress'.

- ❑ To add or override (unless disabled) further properties, add ✓-marks in the corresponding row and add further values.



- ❑ Give a data point into each greyed out field to activate the corresponding property so that the strong-typed property values are applied to the selected instances.

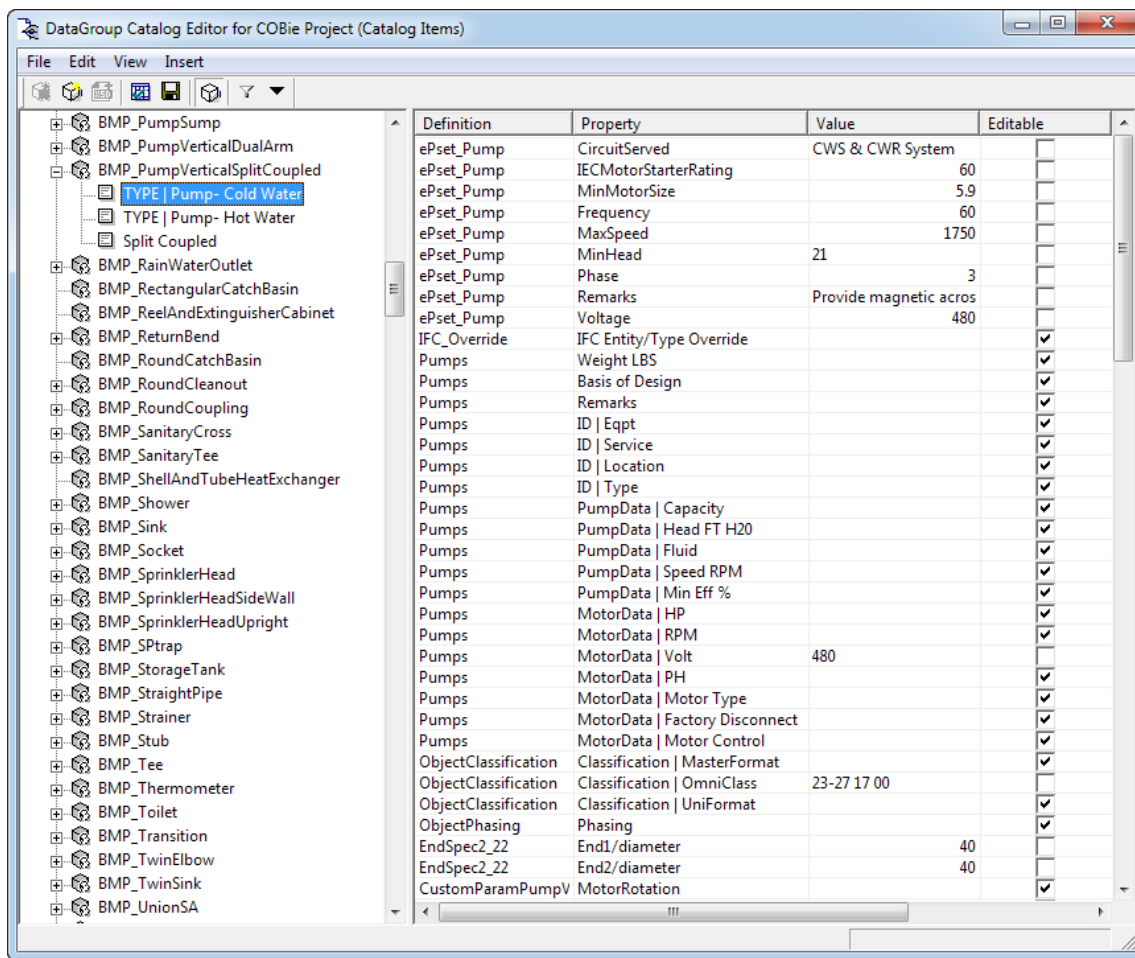
☞ Note the ✓-mark is not visible in the greyed out field.


- ❑ Give a data point into the model to change the instances.

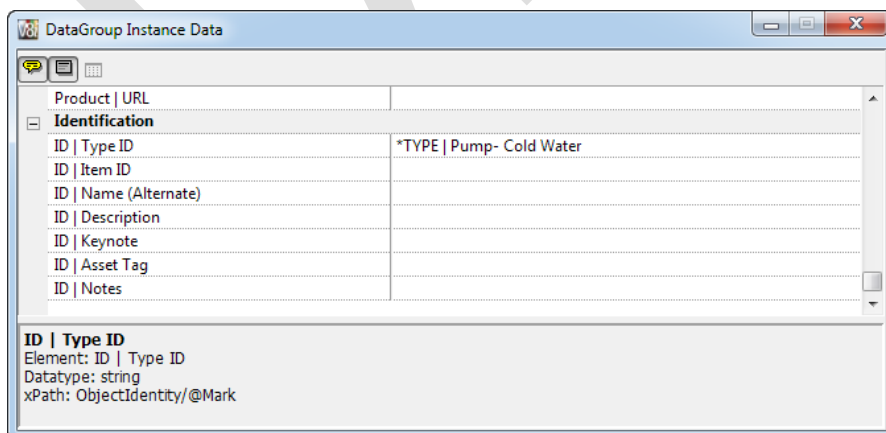
☞ Be aware that dimensional changes, e.g. door widths, may require repositioning of items!

### 3.2.1.2.2 MEP

- ❑ Open the DataGroup Catalog Editor.
- ❑ Make a copy of the default DataGroup Catalog Item using right-click option 'Copy' or 'Edit' > 'Copy' (Ctrl C), e.g. of 'Split Coupled'.
- ❑ Rename the copy, possibly indicating that it is a strong type by adding TYPE as a pre- or suffix, e.g. 'TYPE | Pump-Cold Water'.
- ❑ Enter values for all properties relevant for the type.
- ❑ Make all properties relevant for the type 'non-editable', but leave properties that can be added or redefined for the instance 'editable'.



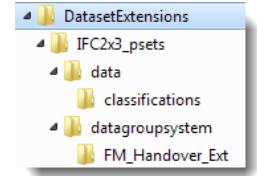
- ❑ Save the newly created type.
- ❑ Place the default DataGroup Catalog Item, e.g. 'Split Coupled'.
- ❑ Open the 'DataGroup Instance Data' dialog box with the 'Modify Properties of Selected Elements' tool  or right-click option 'Modify Selected Component'.
- ❑ For the 'ID | Type ID' property, enter the typed Catalog Item prefixed with a \*-character, e.g. '\*TYPE | Pump- Cold Water'; this means the element will be exported as type 'TYPE | Pump- Cold Water' instead of 'Split Coupled', including the properties defined for the type.





## 3.2.2 Dataset Extension

Also installed with AECOSim Building Designer is a dataset extension folder. It includes the IFC2x3 Pset extension, which includes IFC2x3 property sets, enumerations, and configurations, including those for the Facility Management Handover/COBie Model View Definition.



## 3.2.3 Company dataset

Users can add company specific dataset content to the Dataset Extension folder. A configuration file, e.g. CompanyDataset.cfg, must reside in the top folder of the company dataset that adds the required folders to the workspace, e.g.

```
#-----
# Add search paths for each subfolder in the ArchDatasetNcsUS dataset
# (Delete or comment out config. variable of subfolder that does not exist)
#-----
# MS_BACKUP                < $(MY_DATASET)bak/
TFDIR_CELL                 < $(MY_DATASET)cell/
ATFDIR_CELL                < $(MY_DATASET)cell/
MS_CELLIST                 < $(MY_DATASET)cell/*.cel
TFDIR_COMP                 < $(MY_DATASET)comp/
TFDIR_CPART                < $(MY_DATASET)cpart/
DG_CATALOGS_PATH          < $(MY_DATASET)datagrouppcatalogs/
DG_SCHEDULE_LAYOUT_PATH   < $(MY_DATASET)datagrouplayouts/
# DG_PATH                  > $(MY_DATASET)datagroupsystem/
TFDIR_FRAME                < $(MY_DATASET)frame/
ATFDIR_CASEWORK            < $(MY_DATASET)frame/casework/
ATFDIR_CASEBASE            < $(MY_DATASET)frame/casework/base/
ATFDIR_CASETALL            < $(MY_DATASET)frame/casework/tall/
ATFDIR_CASEWALL            < $(MY_DATASET)frame/casework/wall/
ATFDIR_DOOR                < $(MY_DATASET)frame/doors/
ATFDIR_WINDOW              < $(MY_DATASET)frame/windows/
#TFDIR_GUIDE               = $(MY_DATASET)guide/           # default folder for column grids
# MS_MATERIAL              < $(MY_DATASET)materials/
# MS_PATTERN               < $(MY_DATASET)materials/pattern/
# MS_BUMP                  < $(MY_DATASET)materials/bump/
TFDIR_PART                 < $(MY_DATASET)part/
```

A configuration variable in the Project Configuration Files (PCFs) must define the configuration variables for the location of the company content and a statement to read CompanyDataset.cfg must included, e.g.

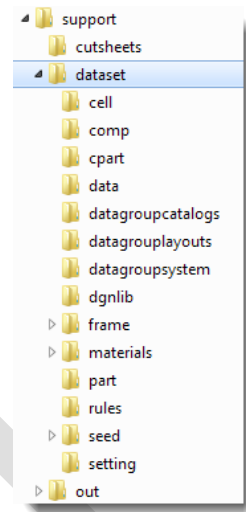
```
#=====
# Establish Company Directories
#=====
# Define path and folder name of company dataset
# (change path and folder name as necessary)
#-----
# COMP_DATASET = $(TF_DATASETS)DatasetExtensions/Company_Dataset/
#
#-----
# Read company cfg-file to add search paths to company dataset subfolders
#-----
# %if exists $(COMP_DATASET)CompanyDataset.cfg)
# %include $(COMP_DATASET)CompanyDataset.cfg
```



# %endif

### 3.2.4 Project dataset

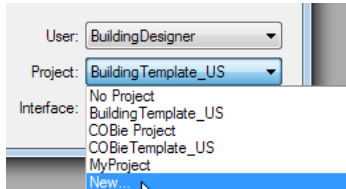
AECOSim Building Designer installs and configures a building template that includes an empty project dataset for project-specific content, which is added to the workspace.



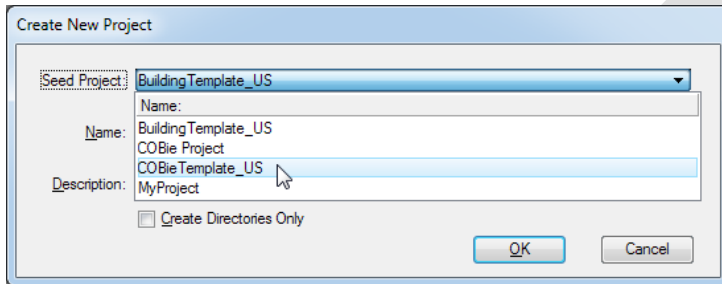
## 4 Workspace operations

### 4.1 Create project

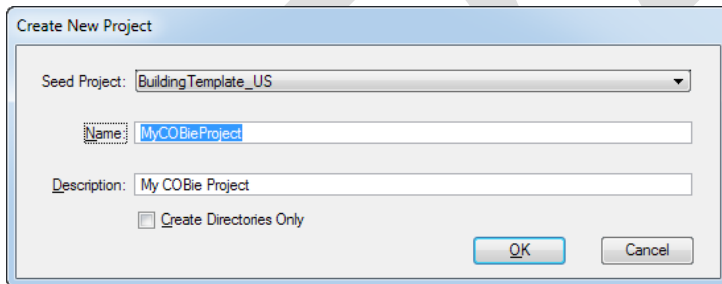
- ❑ Start AECOSim Building Designer
- ❑ In the 'File Open' dialog box, select 'New ...' from the Project dropdown menu to open the 'Create New Project' dialog box



- ❑ In the 'Create New Project' dialog box, select the COBie template project, e.g. 'COBieTemplate\_US', from the 'Seed Project' dropdown menu



- ❑ Enter a Name and Description for the new project



☞ Do not check 'Create Directories Only'

- ❑ Click 'OK' to create the new project

☞ A Project Configuration File (PCF) is created that includes the configuration variable IFC\_PROJECT=2, thus extending the workspace with the 'IFC2x3\_psets' and 'FM\_Handover\_Ext' (COBie) dataset extensions.

### 4.2 Upgrade Projects to COBie

If a project's workspace does not include the 'IFC2x3\_psets' and 'FM\_Handover\_Ext' (COBie) dataset extensions, e.g. legacy projects, they can be added by changing the value of the configuration variable IFC\_PROJECT from '0' to '1' or '2'.

- ❑ Open the project's Project Configuration File (PCF) with a text editor, such as Notepad
- ❑ In the PCF, edit configuration variable IFC\_PROJECT=<number>

```
#=====
# Extended Dataset Content:
#=====

# Enables the Dataset to be extended to include IFC2x3 property sets.
# IFC_PROJECT=0, dataset not extend,
# IFC_PROJECT=1, dataset extended with IFC psets
# IFC_PROJECT=2, dataset extended with IFC psets Plus FM Handover(COBie)

%level 4
    IFC_PROJECT : 0

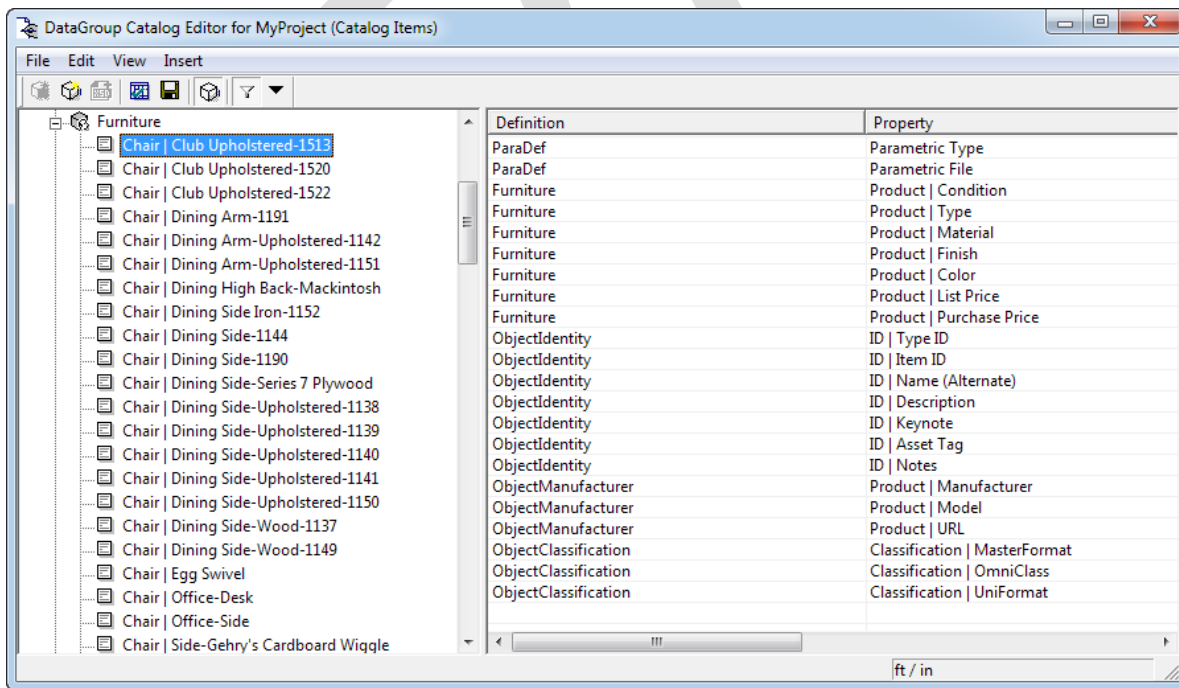
    %if $(IFC_PROJECT)==1 || $(IFC_PROJECT)==2
        %if exists $(IFCDIR)Dataset.cfg
            % include $(IFCDIR)Dataset.cfg
        %endif
    %endif

%level 3
```

☞ Once the value of IFC\_PROJECT has been edited to '1' or '2' it should not be reversed.

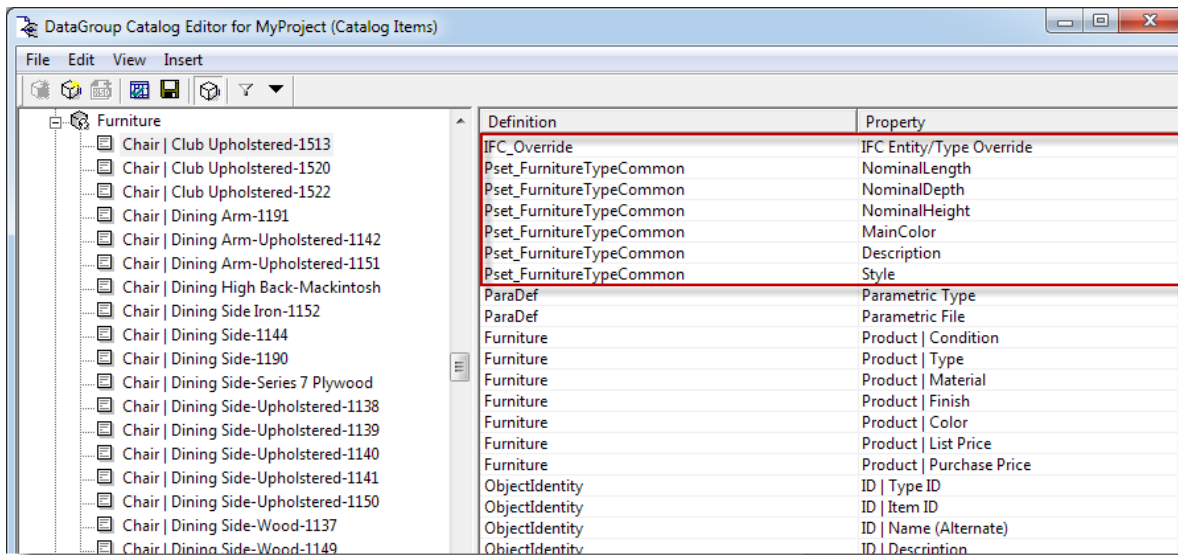
## 4.2.1 IFC\_PROJECT=0

Only the properties defined and configured by AECOSim Building Designer plus user-defined property sets and properties will be available, e.g.




## 4.2.2 IFC\_PROJECT=1

The common IFC2x3 property sets, such as Pset\_DoorCommon, Pset\_FurnitureTypeCommon, Pset\_WallCommon, etc., are added to the workspace. Their properties are available for adding instance data, which will subsequently be exported to IFC.



Additional IFC2x3 property sets can be added by editing IFC\_Pset\_catalogtypeexts.xml located in ... \DatasetExtensions\IFC2x3\_psets\datagroupsystem by removing the '<!--' in front of the string and '-->' at the end of the string, e.g.

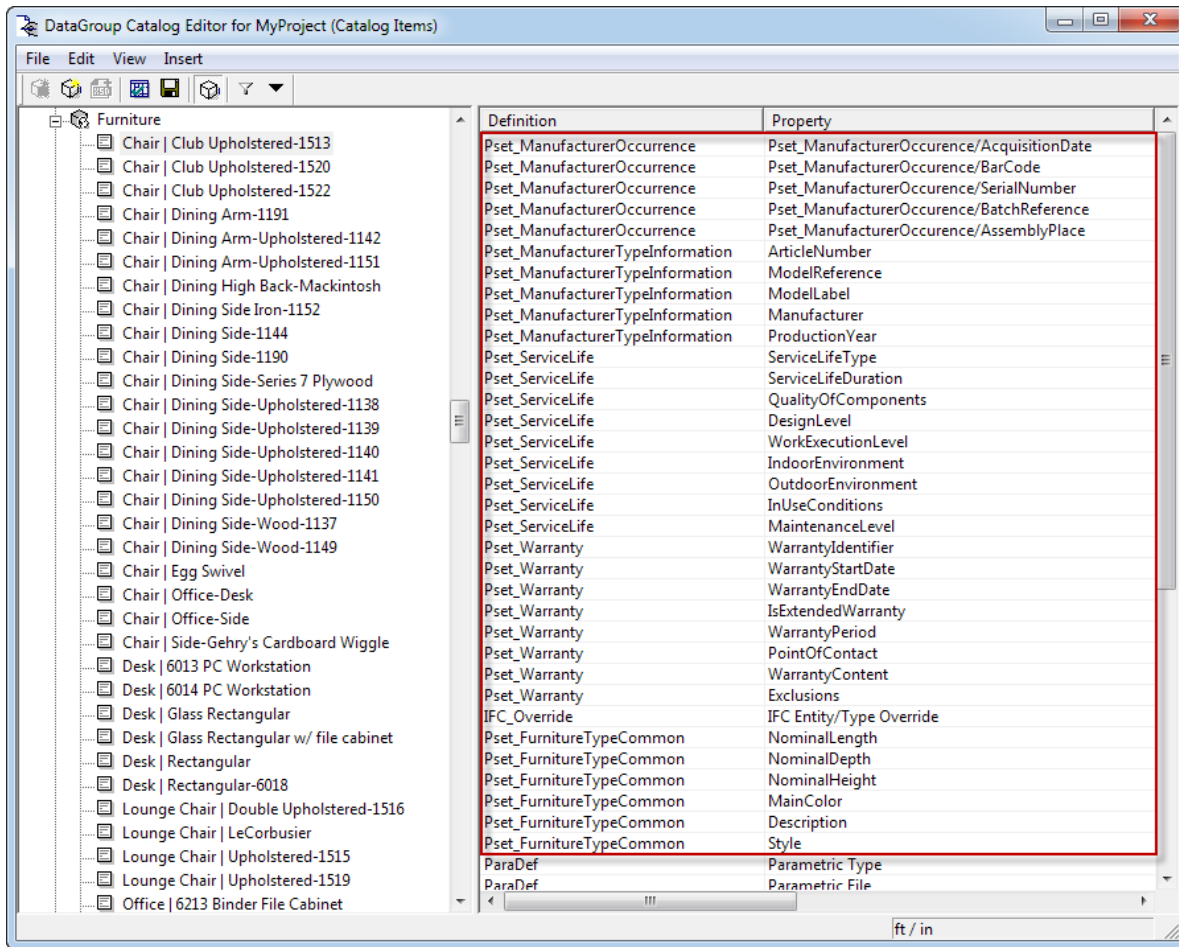
```
<CatalogTypeExtension type="Building">
  <InstanceDataDefinition defType="USER" definition="IFC_Override"/>
  <InstanceDataDefinition defType="USER" definition="Pset_BuildingCommon"/>
  <!-- <InstanceDataDefinition defType="USER" definition="Pset_BuildingUse"/> -->
  <!-- <InstanceDataDefinition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

 Editing xml-files could seriously damage the file and corrupt the entire workspace! Therefore, it should only be done with knowledge of the xml-syntax and ideally using an xml-editor with syntax checker. Always keep a backup of the original file.

### 4.2.3 IFC\_PROJECT=2

Property sets specified by the Facilities Management Handover/COBie ModelView Definition (MVD) are added to the workspace for relevant DataGroup types, i.e.

- Site, Building, Floor, and Space: Pset\_PropertyAgreement
- Serviceable items, i.e. FFE: Pset\_ManufacturerOccurrence
- Pset\_ManufacturerTypeInfo
- Pset\_ServiceLife
- Pset\_Warranty




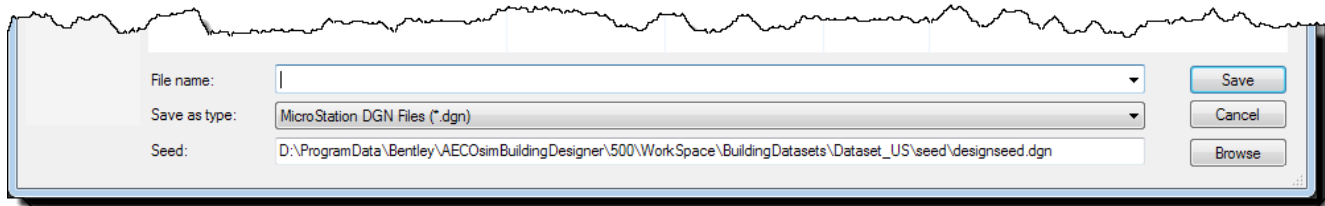
### 4.3 Start AECOSim Building Designer

Depending on the start-up mode, all or only discipline-specific application tools will be available in the user interface. Discipline-specific tools can be loaded and unloaded during the design session.

- ☐ Choose the start-up mode
  - ☐ AECOSim Building Designer
  - ☐ Architectural Building Designer
  - ☐ Structural Building Designer
  - ☐ Mechanical Building Designer
  - ☐ Electrical Building Designer

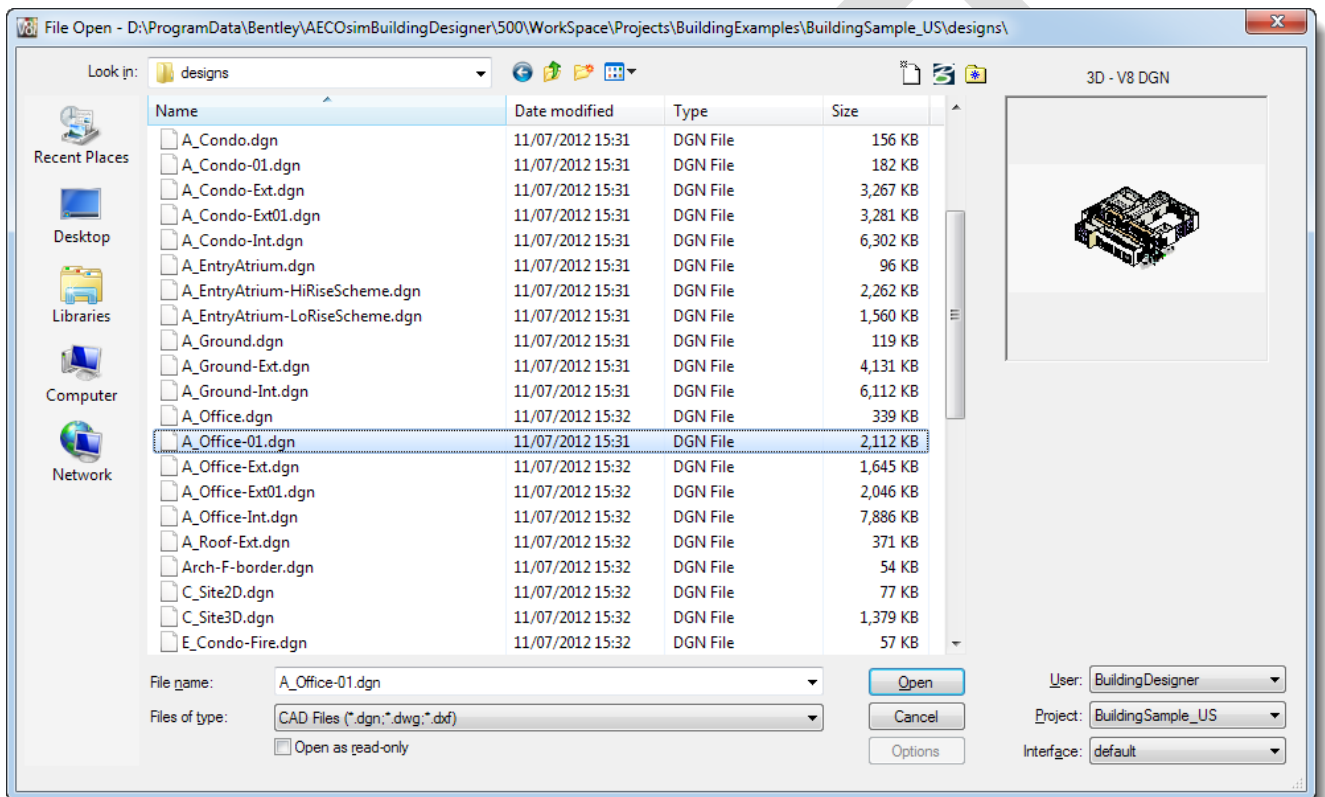
### 4.4 Create design file(s)

- ☐ Create a design file using the 'New File' button .
- ☐ Key-in the name for the new design file.
- ☐ To use another seed file than the one displayed by default, click 'Browse' and navigate to the seed file to be used for the new file.



## 4.5 Open design file

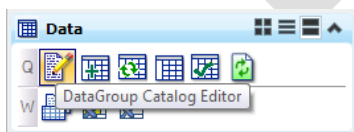
- ❑ Select the design file and click 'Open' (or double-click the design file).



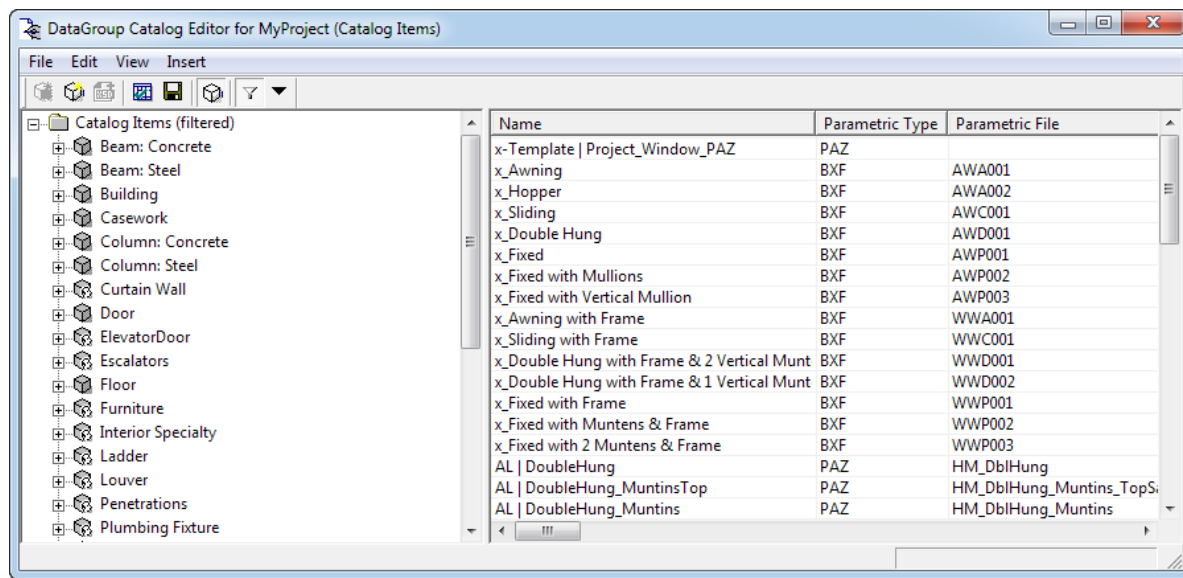
## 4.6 Create custom DataGroup catalog type(s)

AECOSim Building Designer includes 100's of DataGroup types of architecture, structural, mechanical, plumbing and electrical building components, such as beams, columns, diffusers, doors, grilles, plumbing fixtures, valves, walls, etc. Tools are provided to define company or project-specific DataGroup types.

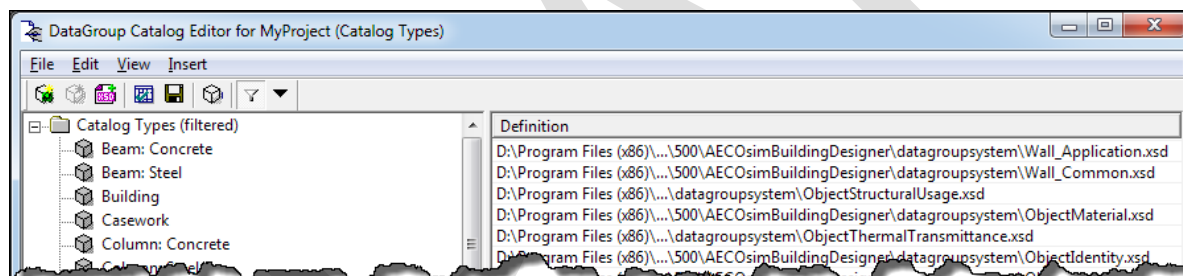
- ❑ Open the DataGroup Catalog Editor from the Data tool group.



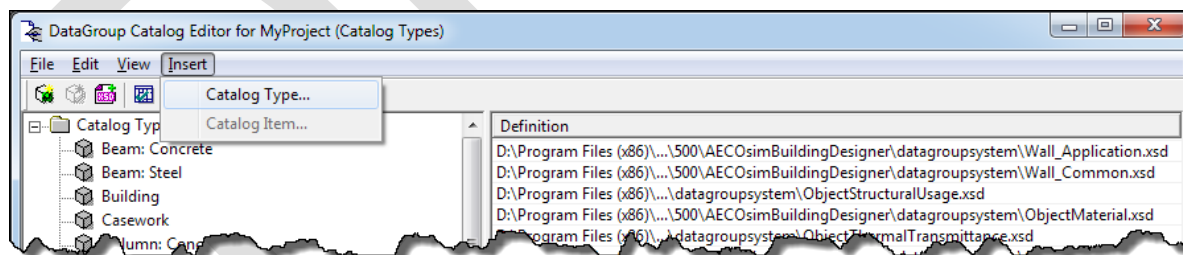
Note that the DataGroup Catalog Editor (Catalog Items) dialog box opens.



- Click 'Show Catalog Items' to open the DataGroup Catalog Editor (Catalog Types) dialog box



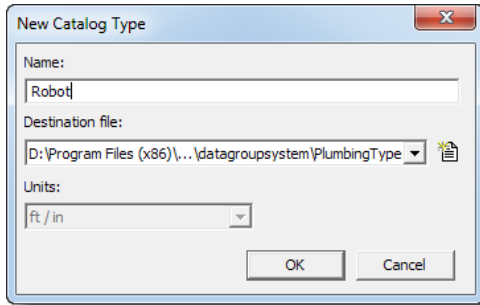
- From the 'Insert' tool bar, select 'Catalog Type...' (or right-click on Catalog Types and use right-click option 'New Catalog Type').



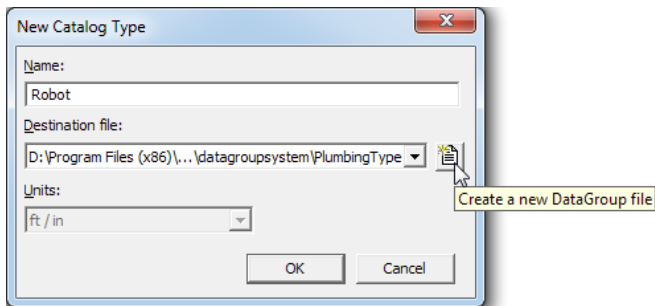
- In the 'New Catalog Type' dialog box



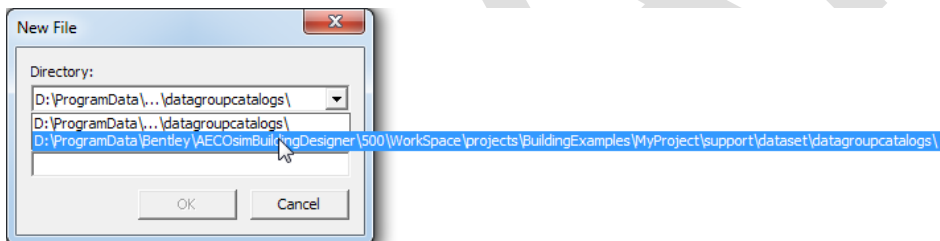
- enter a Name for the new catalog type, e.g. 'Robot' (Note: no space allowed)



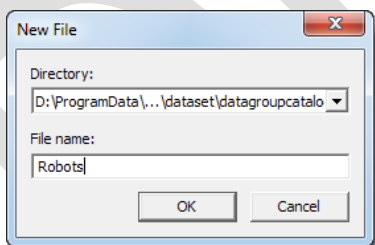
- click 'Create a new Datagroup file'



- in the 'New File' dialog box, select the Directory for the new file; typically the datagroupcatalogs folder in the project dataset, e.g. ...MyProject\support\dataset\datagroupcatalogs\



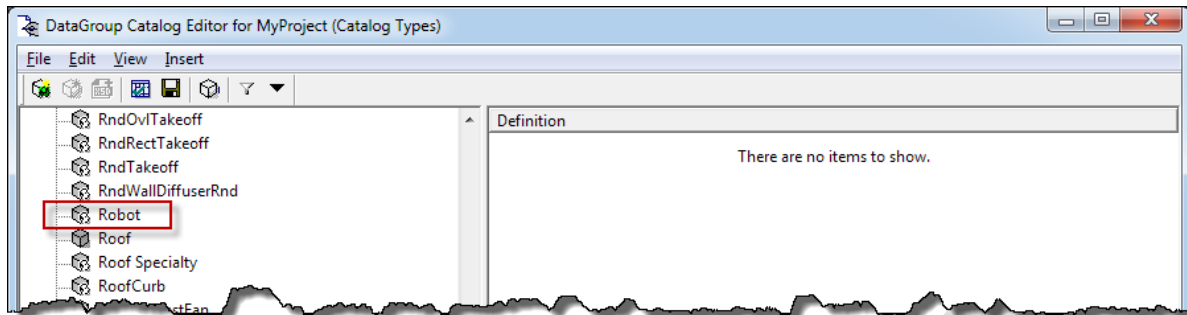
- enter a name for the new file, e.g. 'Robots' (Note: no space allowed; they can be added via DisplayNames)



- click 'OK' in the 'New File' dialog box
- click 'OK' in the 'New Catalog Type' dialog box

❑ Note that the new catalog type is added





Hint: you may need to switch off the 'Filter items by Catalog' option to see the new catalog type.

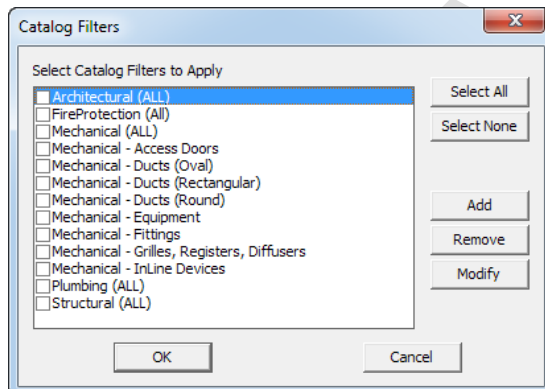


- ❑ Optionally: add the new DataGroup catalog type to one of the catalog filters

- open the Catalog Filter



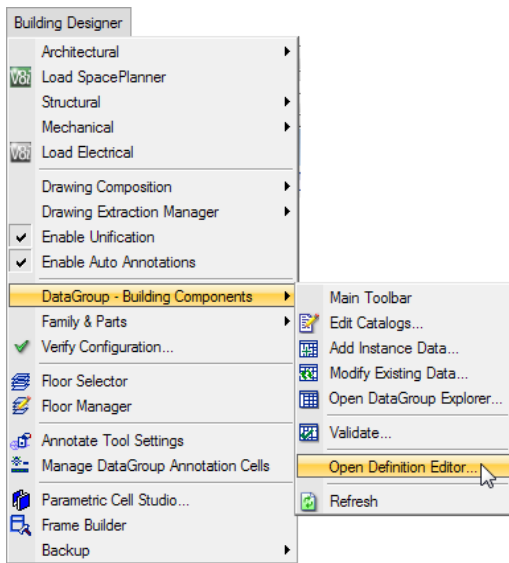
- select an existing Catalog Filter and click 'Add'




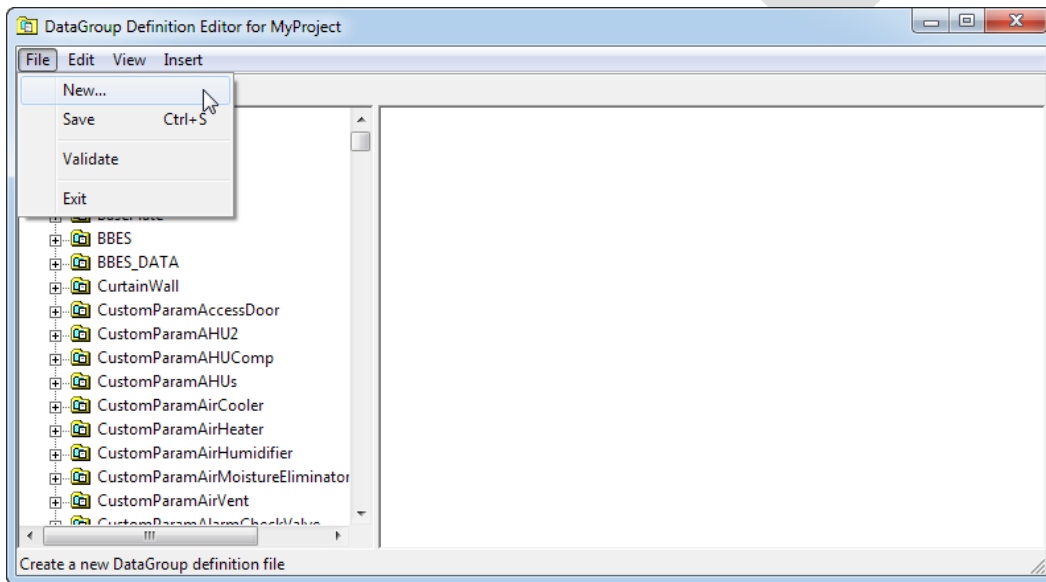
## 4.7 Create custom DataGroup property definitions

Every DataGroup catalog type needs to be associated with at least one property definition file. Created in the DataGroup Definition Editor, property definition files define the name of the property, optionally the display name, the format (string, integer, list, ...), if property is mandatory, the default value (if any), the enumerated list values (if any).

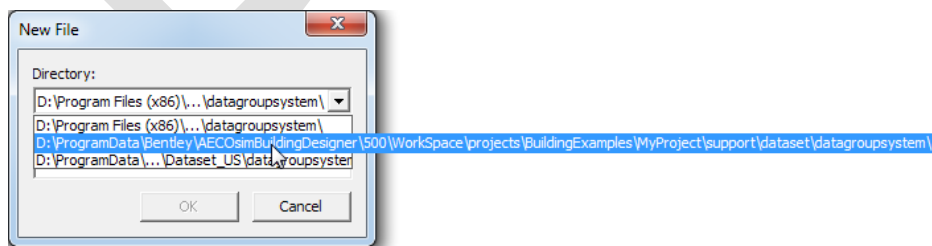
- ❑ Open the DataGroup Definition Editor from the Building Designer tool bar > DataGroup – Building Components > Open Definition Editor



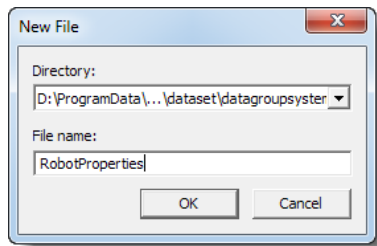
- ❑ From the 'File' tool bar, select 'New...' (or right-click on  and use option 'New Definition File')



- ❑ In the 'New File' dialog box
  - select the Directory for the new file; typically the datagroussystem folder in the project dataset, e.g. ...\\MyProject\\support\\dataset\\datagroussystem\\



- enter a name for the new file, e.g. 'RobotProperties' (Note: no space allowed; they can be added via DisplayNames)

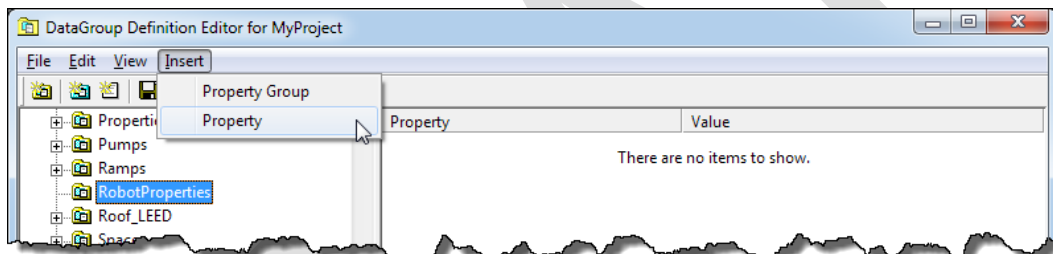


- click 'OK' when done

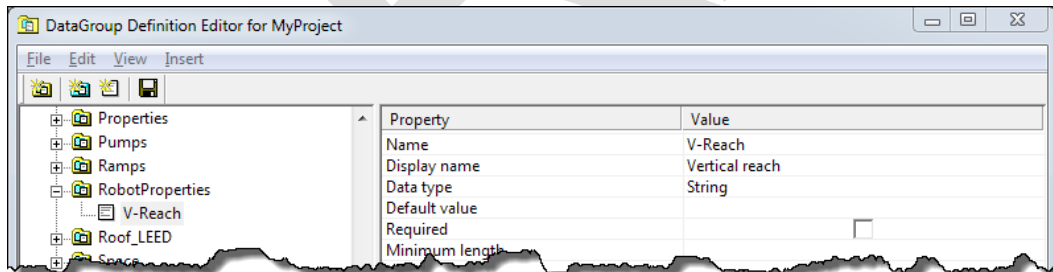
- Note the new property file has been added



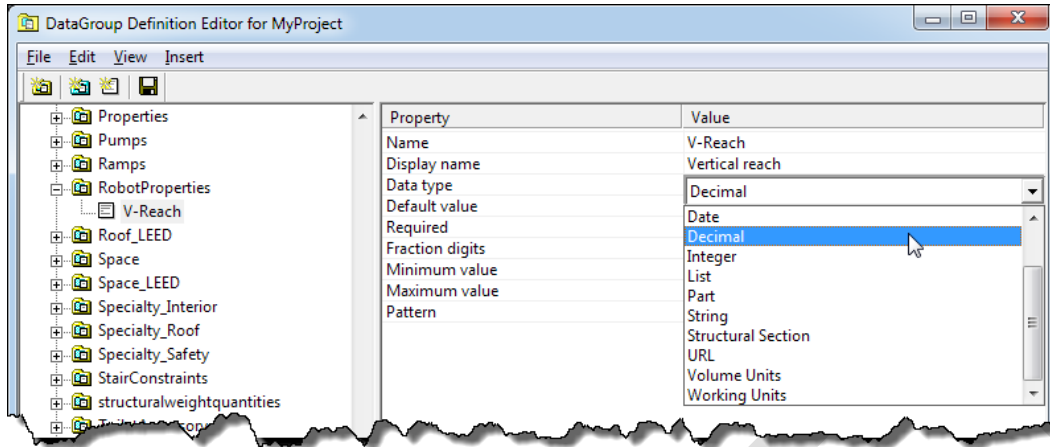
- From the 'Insert' tool bar, select 'Property' to add a property to the file



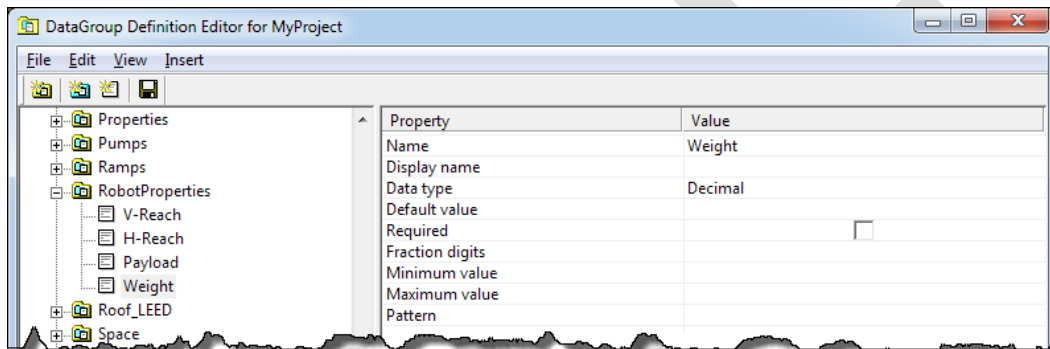
- Enter a Name for the new property, e.g. 'V-Reach' and optionally a Display name, e.g. 'Vertical reach'



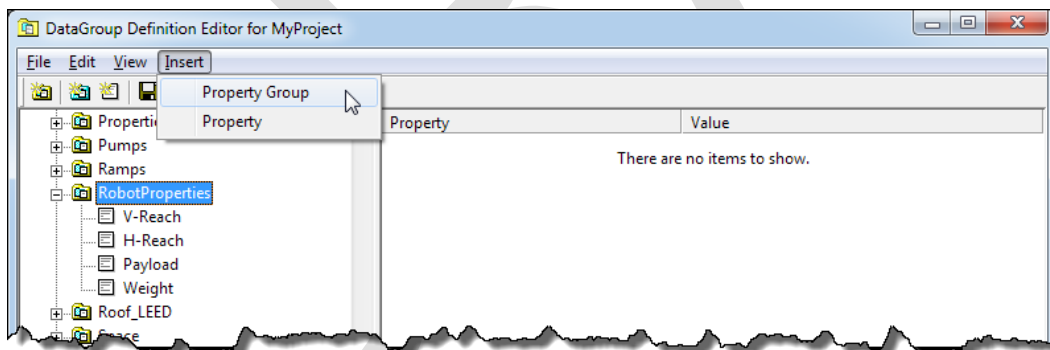
- ❑ From the Data type drop down menu, select an appropriate data type, e.g. 'Decimal'



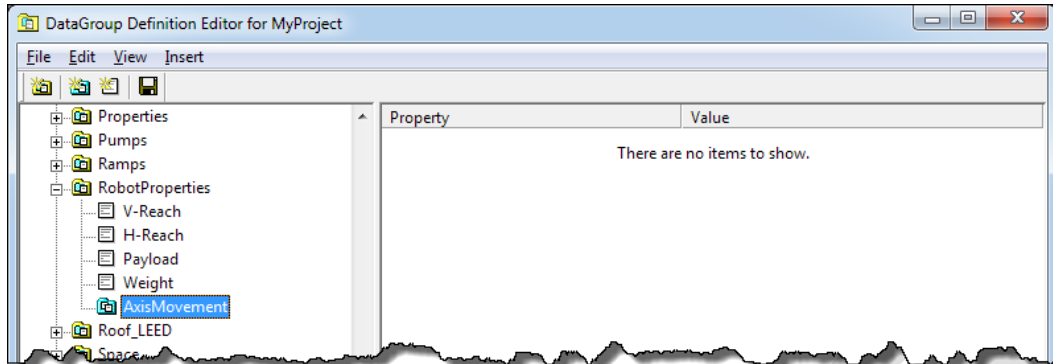
- ❑ Repeat for further properties, e.g.



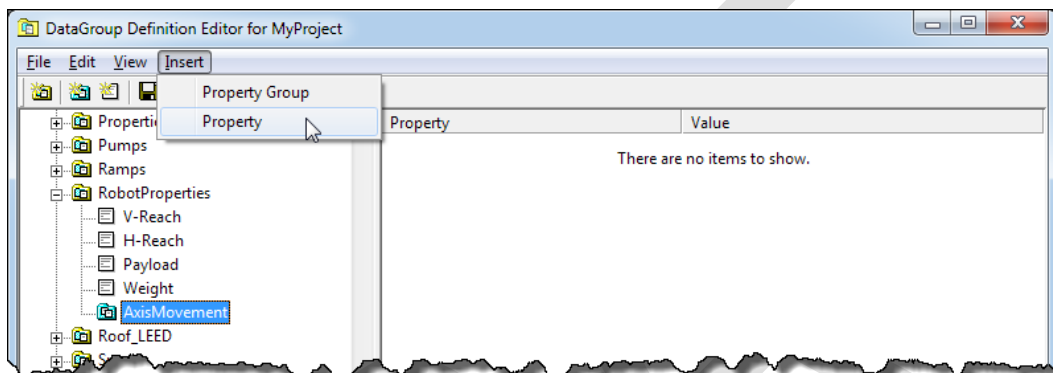
- ❑ To group properties, select 'Property Group' from the 'Insert' tool bar to create a property group



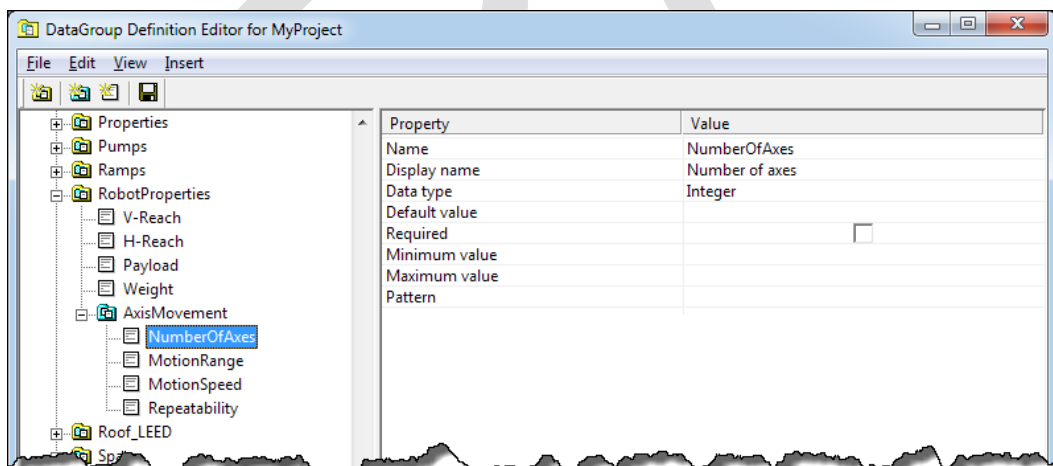
- ❑ Enter a name for the property group, e.g. 'AxisMovement'



- ❑ Select the property group, then select 'Property' from the 'Insert' tool bar

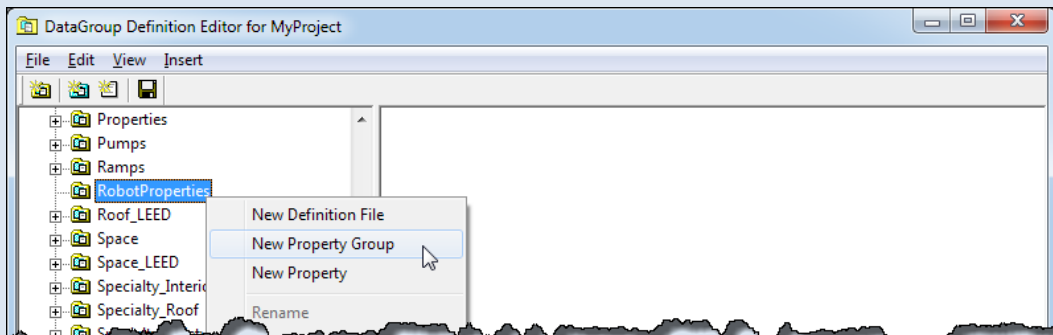


- ❑ Add properties to the property group as described above, e.g.



- ❑ 'Save Changes' before exiting!!!

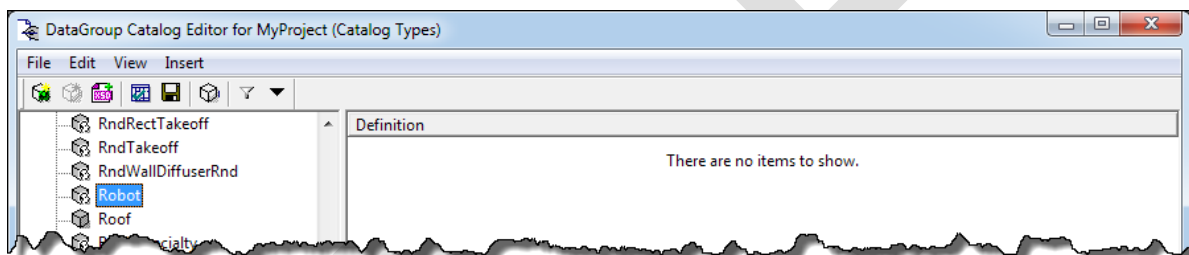
You can also right-click on the property file and select 'New Property Group' or 'New Property' from the menu



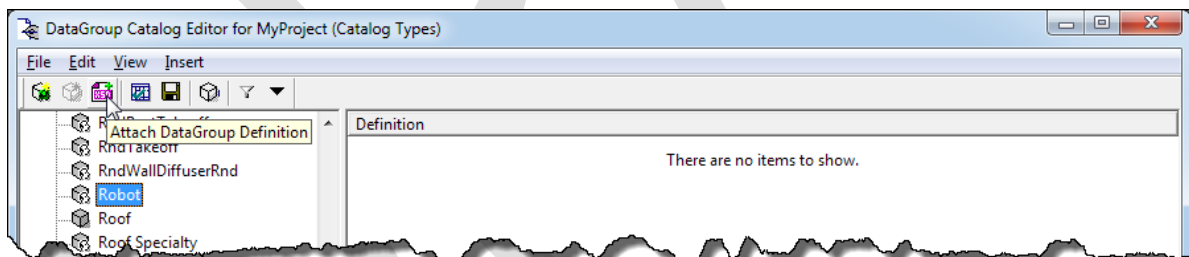
## 4.8 Attach DataGroup types with DataGroup definitions

DataGroup definitions need to be attached to DataGroup type(s), before the properties can be used.

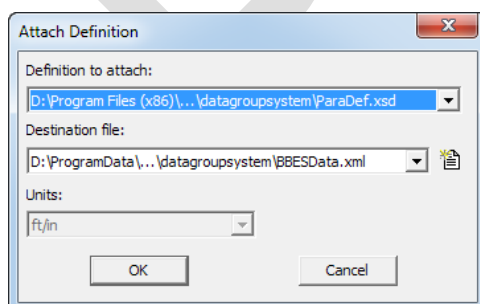
- ❑ Open the DataGroup Catalog Editor (Catalog Types)



- ❑ Click 'Attach DataGroup Definition'

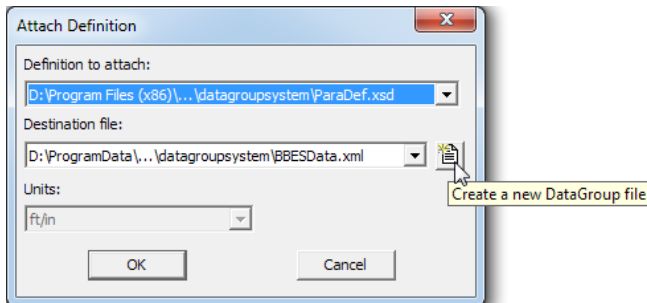


- ❑ In the 'Attach Definition' dialog box
  - select a Definition to attach from the dropdown list, e.g. ParaDef.xsd

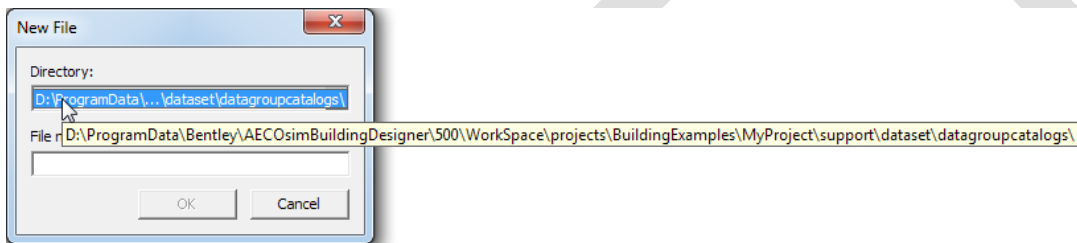


ParaDef.xsd is required for furniture, fixture, and equipment types to allow definition of type (CEL, BXC, PAZ) and the (parametric) cell name

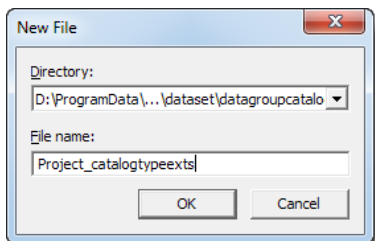
- if necessary, create a new Destination file by clicking the 'Create a new DataGroup file' button



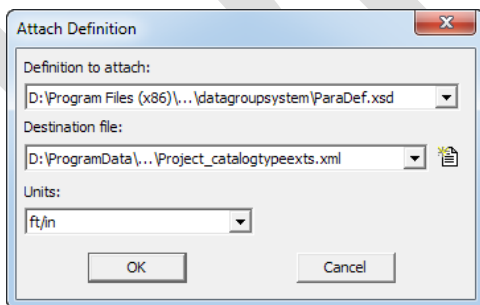
- in the 'New File' dialog box, select the Directory, typically the datagroupcatalogs folder in the project



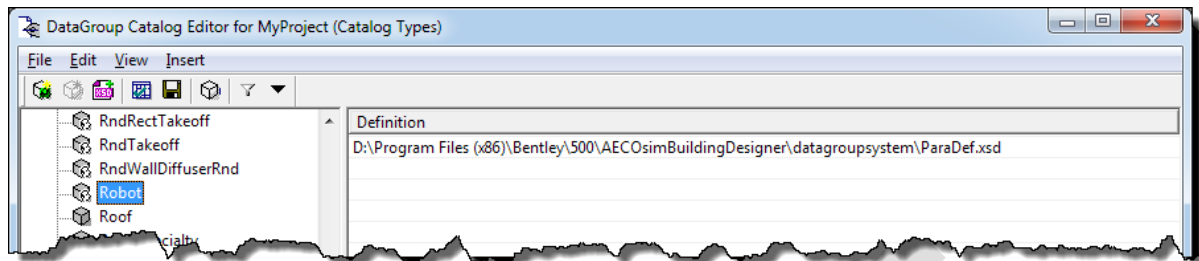
- enter a File name



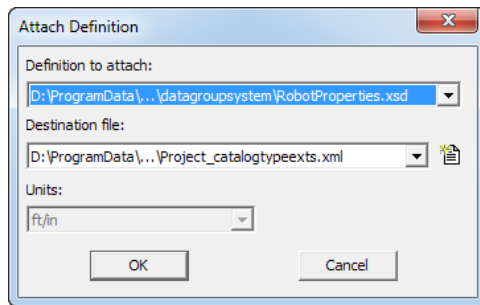
- click 'OK' in the 'New File' dialog box
- click 'OK' in the 'Attach Definition' dialog box



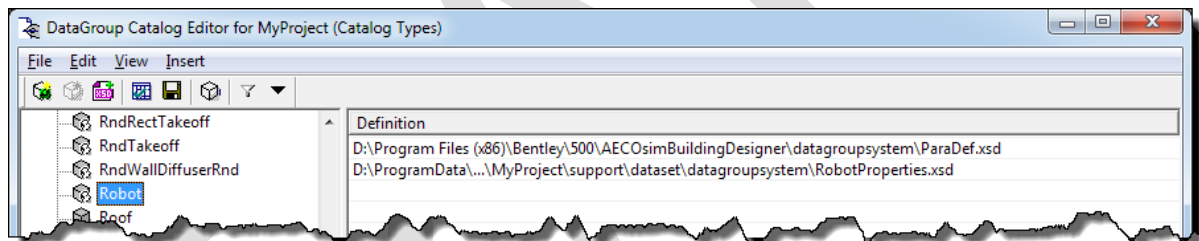
Note that the ParaDef.xsd is attached to the DataGroup type



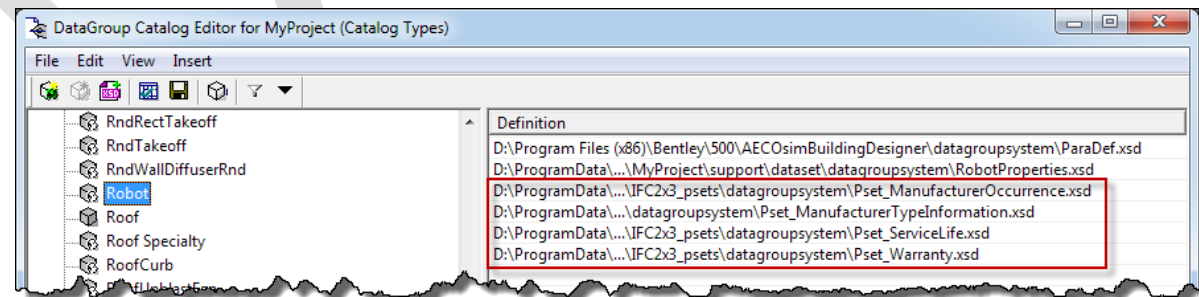
- Attach custom DataGroup definition files, e.g. 'RobotProperties.xsd'



Note that 'RobotProperties.xsd' is now attached.



- Attach the property sets required for the FM Handover/COBie MVD from ...\\DatasetExtensions\IFC2x3\_psets\datagroupsystem as described above, i.e.
  - Pset\_ManufacturerOccurrence
  - Pset\_ManufacturerTypeInfoInformation
  - Pset\_ServiceLife
  - Pset\_Warranty

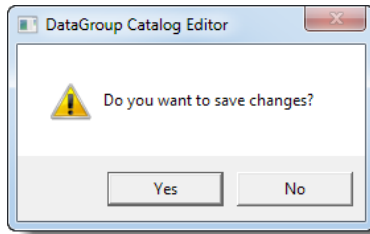


- Click 'Save' to save the changes made in the DataGroup Catalog Editor.





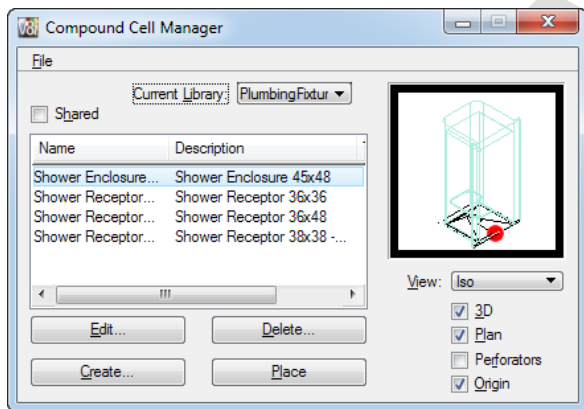
or when prompted



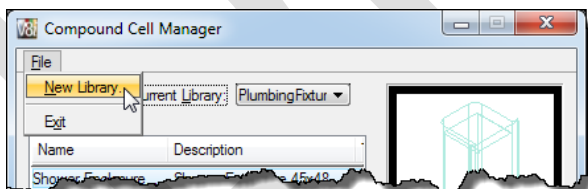
## 4.9 Create a compound cell

For most types of furniture, fixture, and equipment (FFE), compound cells should be used, as they incorporated both the 3D representation used for modeling and rendering as well as the 2D symbolic representation used in drawings. Stored in .bxc libraries, they are created via the Compound Cell Manager.

- ❑ Open the Compound Cell Manager



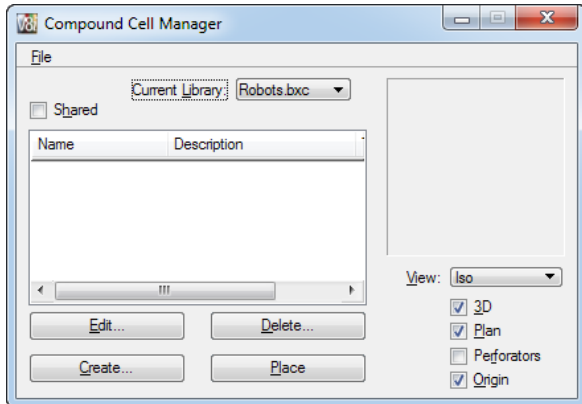
- ❑ If necessary, create a new Compound Cell library
  - click 'New Library...' from the File menu



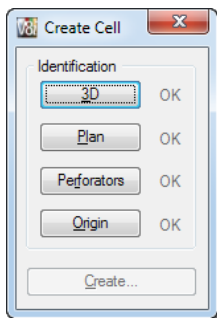
- in the 'Create Compound Cell Library' dialog box, navigate to the destination folder, e.g. ...\\support\\dataset\\cell, and enter a file name, e.g. Robots



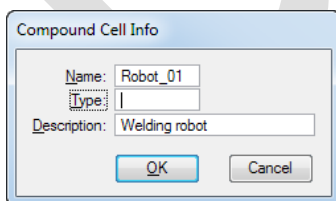
- click 'Save' to create the compound cell library



- ❑ Open the Current Library from the dropdown menu, e.g. Robots.bxc
- ❑ Click 'Create'

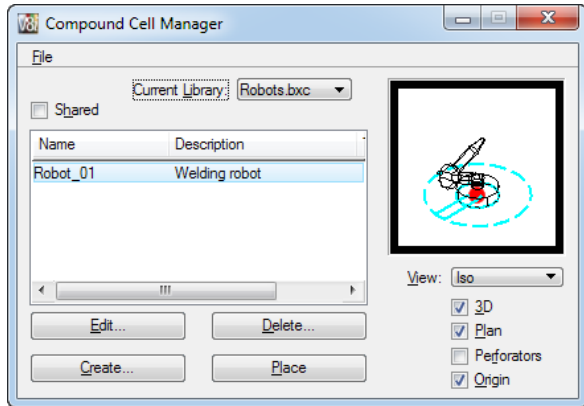


- ❑ In the 'Create Cell' dialog box
  - click '3D' and identify the 3D graphics, preferably a 3D cell  
Note that 'OK' next to the '3D' button turns black
  - click '2D' and identify the 2D graphics, preferably a 2D cell  
Note that 'OK' next to the '2D' button turns black
  - if the compound cell should include a perforator to punch holes through walls or slabs, click 'Perforator' and identify a shape to be used as perforator  
Note that 'OK' next to the 'Perforator' button turns black
  - identify the origin of the Compound Cell with the datapoint  
Note that 'OK' next to the 'Origin' button turns black
  - click 'Create' and enter a Name, Type (optional), and Description (optional).  
Note: the 'Create' button only becomes active when 'OK' is black for at least the '3D', '2D', and 'Origin'.



- click 'OK' to add the Compound Cell to the library

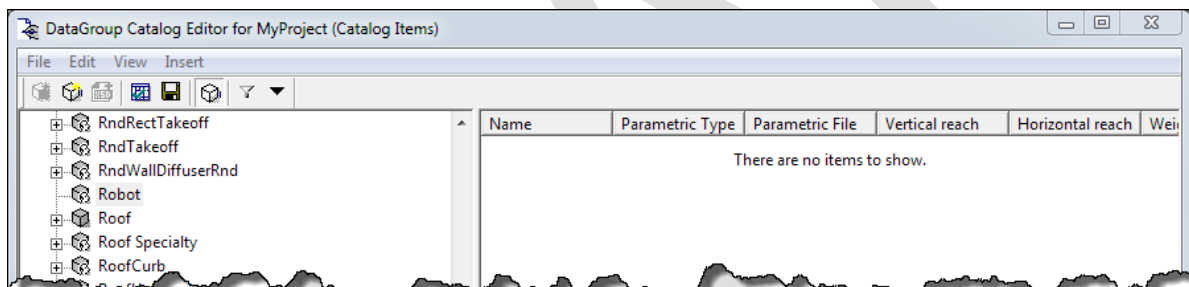
- Note that the Compound Cell is added to the library



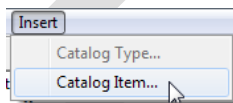
## 4.10 Create DataGroup catalog items

DataGroup catalog items define default property values for building components. When these components are placed or at any time afterwards, the default values can be modified or more property values added. On IFC export, some property values of catalog items are used as 'Type' information, e.g. 'AssetAccountingType'.

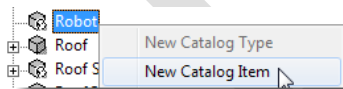
- Open the DataGroup Catalog Editor and select the DataGroup type to which to add a catalog item, e.g. 'Robots'



- From the 'Insert' menu, select 'Catalog Item...'

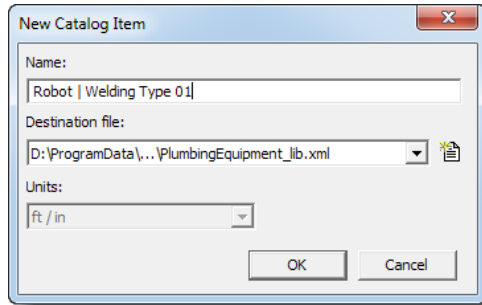


alternatively, right-click the type and select 'New Catalog Item'

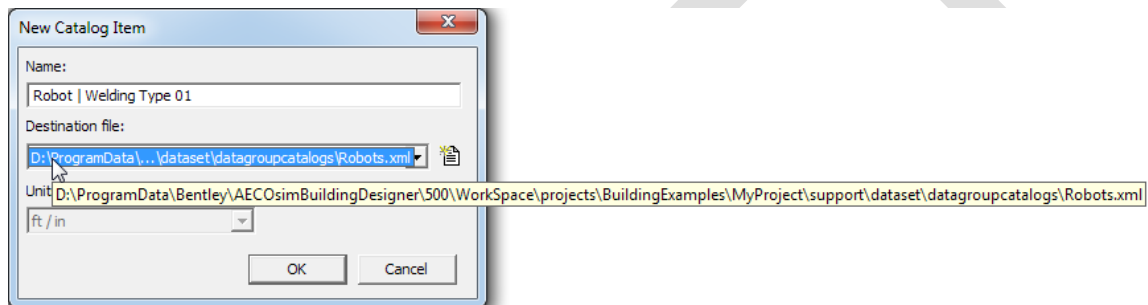


- In the 'New Catalog Item'

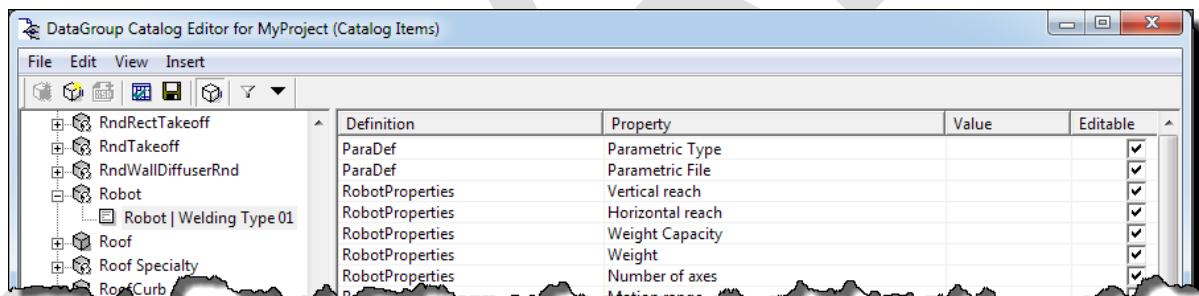
- enter a Name for the new catalog item, e.g. 'Robot | Welding Type 01'



- if necessary, create a new Destination file, typically the corresponding file in ...support\dataset\datagroupcatalogs, e.g. 'Robots.xml'

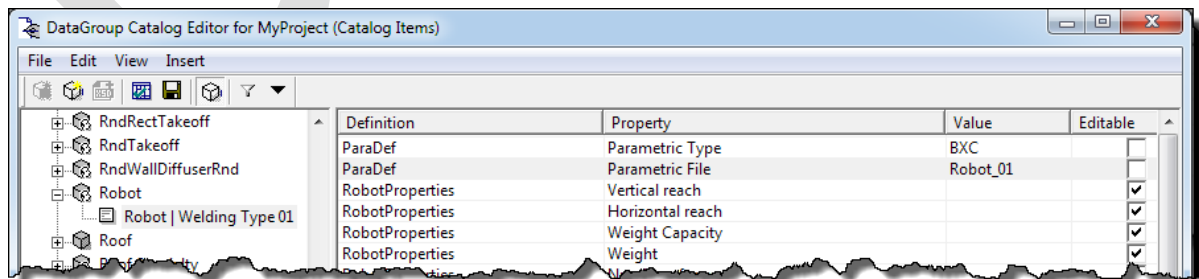


- click 'OK' to create the catalog item

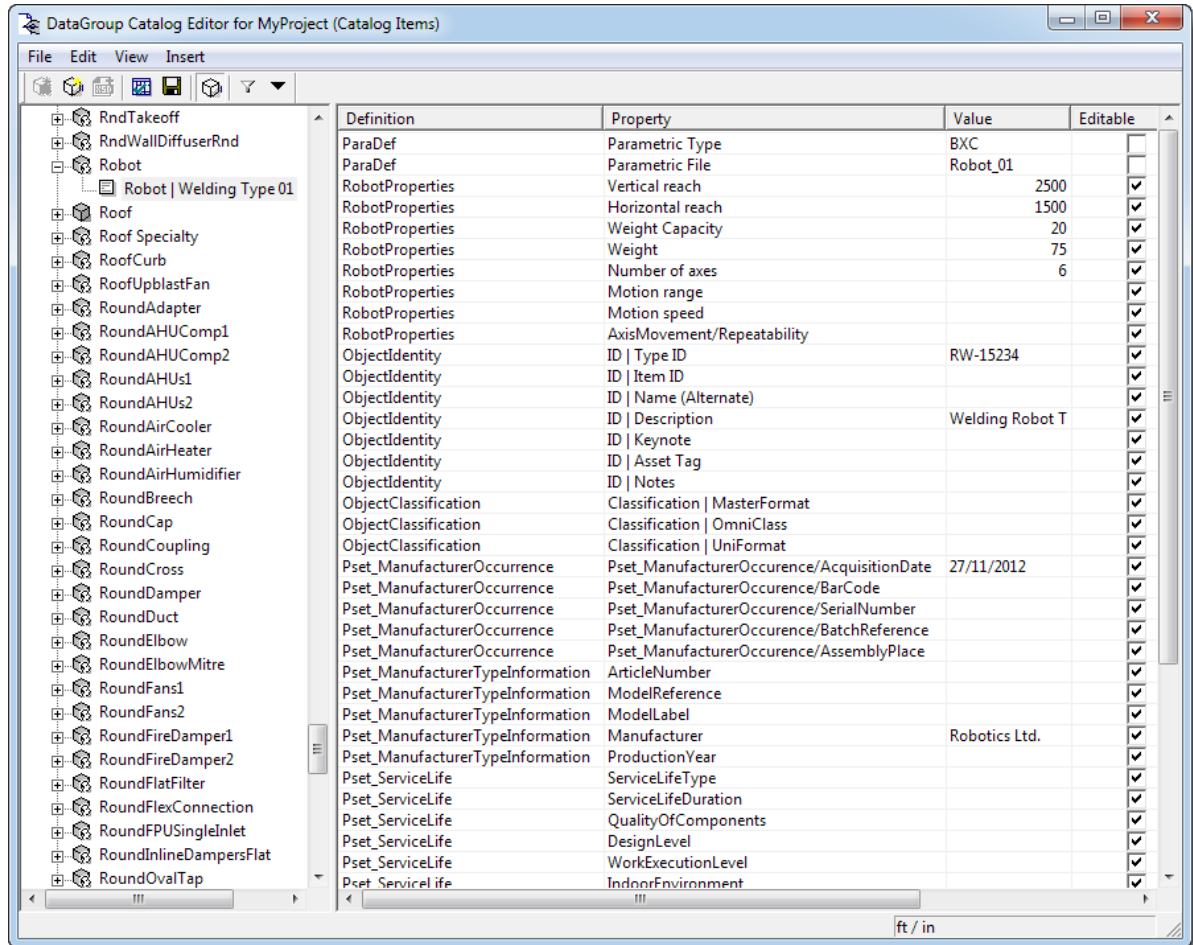


#### □ For the new catalog item

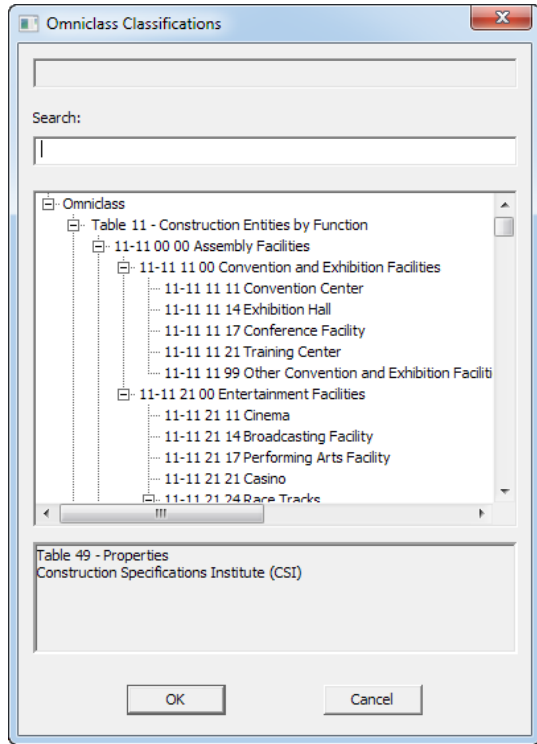
- select a Value for the 'Parametric Type' from the dropdown list, e.g. BXC
- enter the name of the Parametric File, e.g. Robot\_01
- remove the tick marks in the 'Editable' column to prevent these values to be change when placing the item



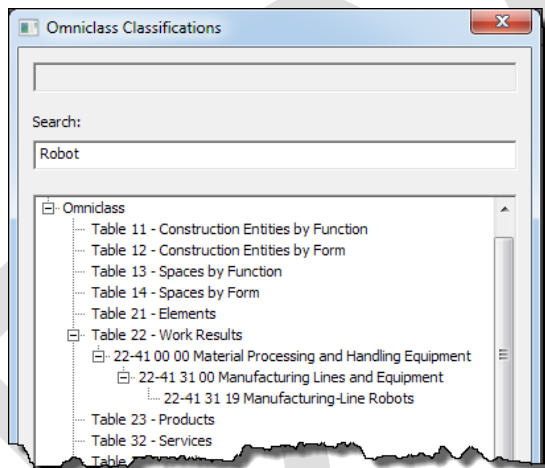
- enter further property default values or as required for the type



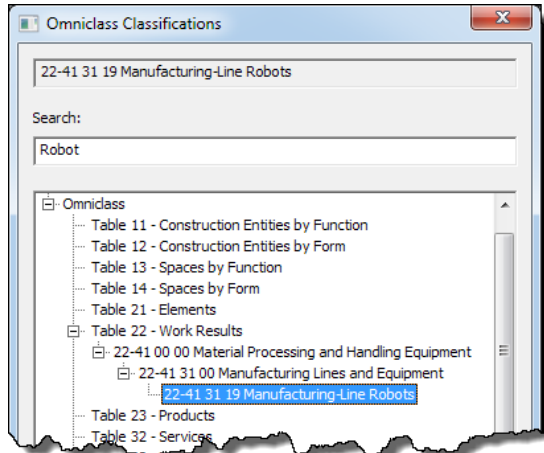
- open the Classification Browser by clicking the Classification cell, e.g. for 'Classification | OmniClass'



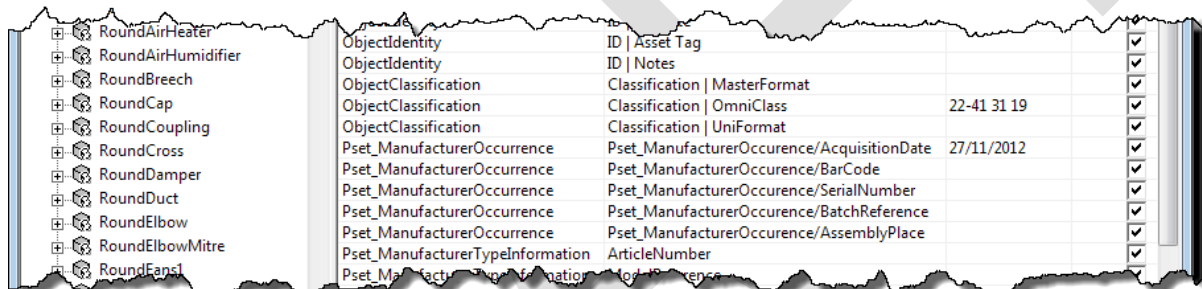
- enter a Search string, e.g. Robot



- click on the classification entry, e.g. '22-41 31 00 Manufacturing-Line Robots'



- click OK to add the classification code to the catalog item



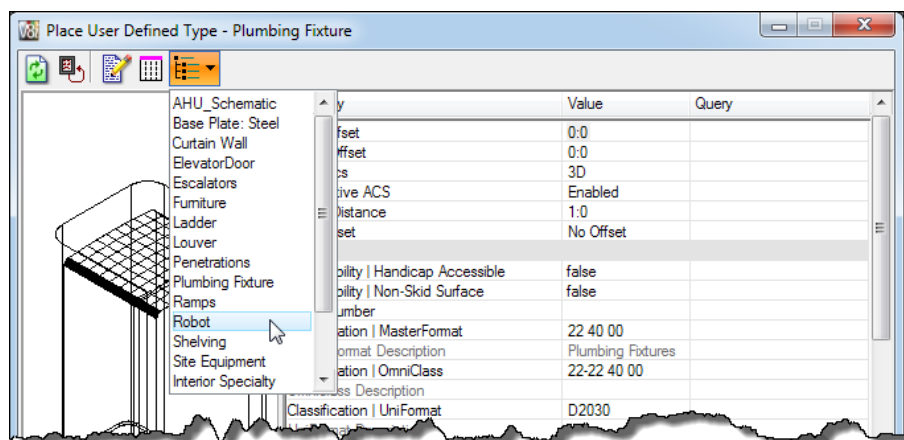
- ☐ Repeat for additional DataGroup catalog items.
- ☐ Click 'Save' to save the new catalog item(s).

☞ Consider informing the IFC-file recipients about the exact property names and formats so that they can add corresponding properties to their workspaces.

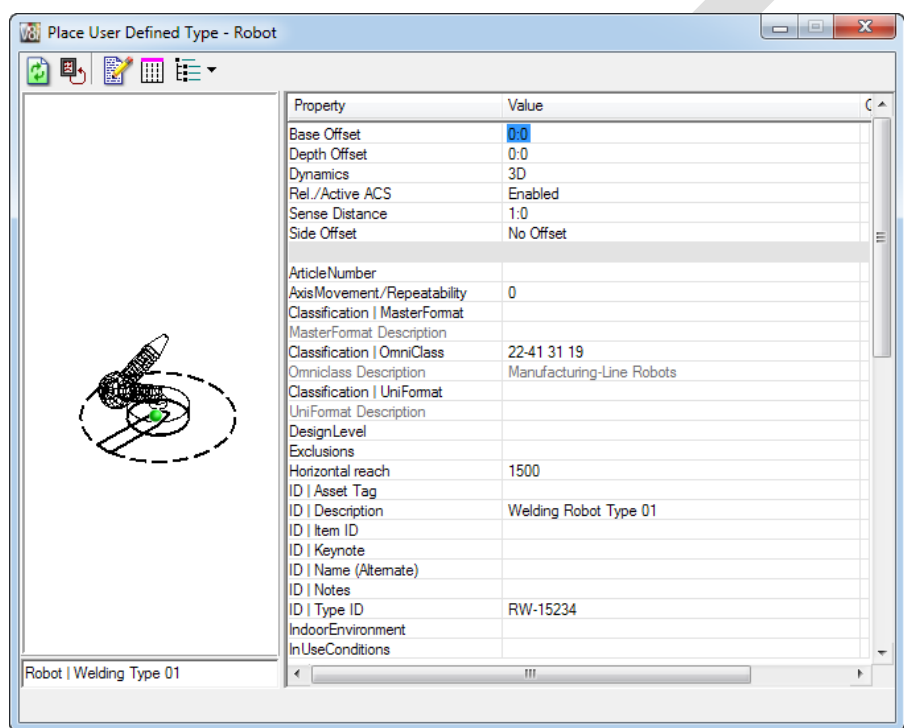
☞ User-defined items are placed with the 'Place Content' tool



- ☐ Open the 'Place User Defined Type' dialog box and select the Type from the 'Instance Data Type' dropdown menu, e.g. 'Robot'



- ❑ Select the item to be placed from the dropdown list, edit or add property values, and place the item.



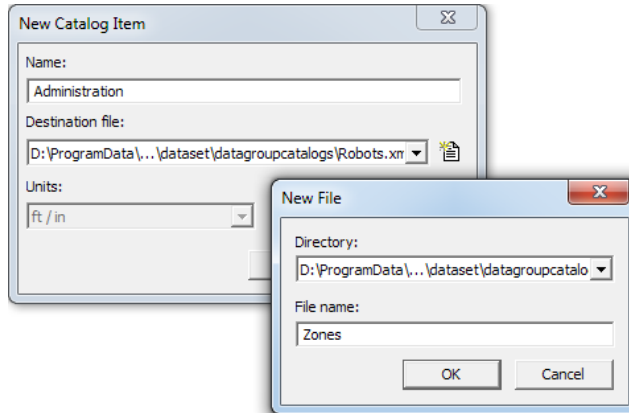
## 4.11 Create DataGroup Catalog Items for IfcZones

In order to assign Spaces to IfcZones, a DataGroup Catalog Item must exist for each Zone.

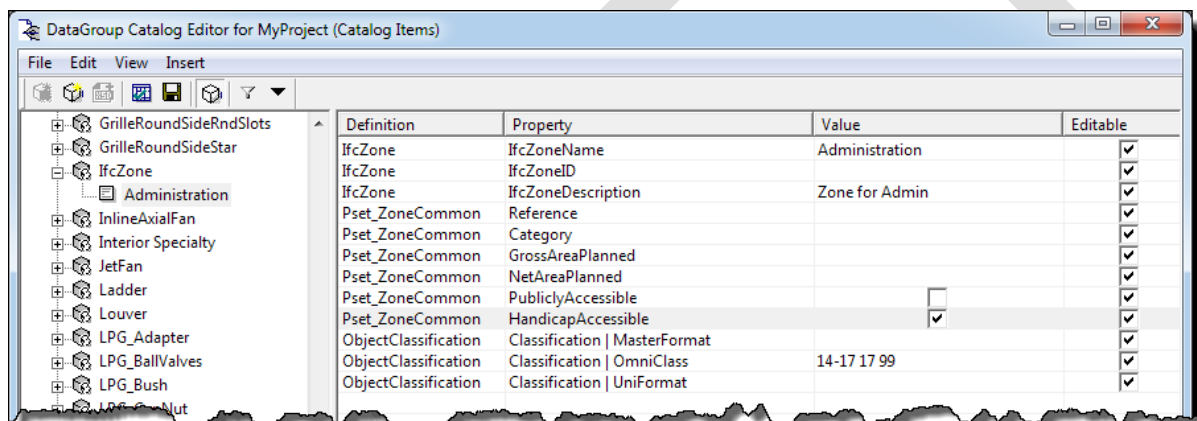
- ❑ Open the DataGroup Catalog Editor, select 'IfcZones' and create a catalog item for each Zone in the project as described above



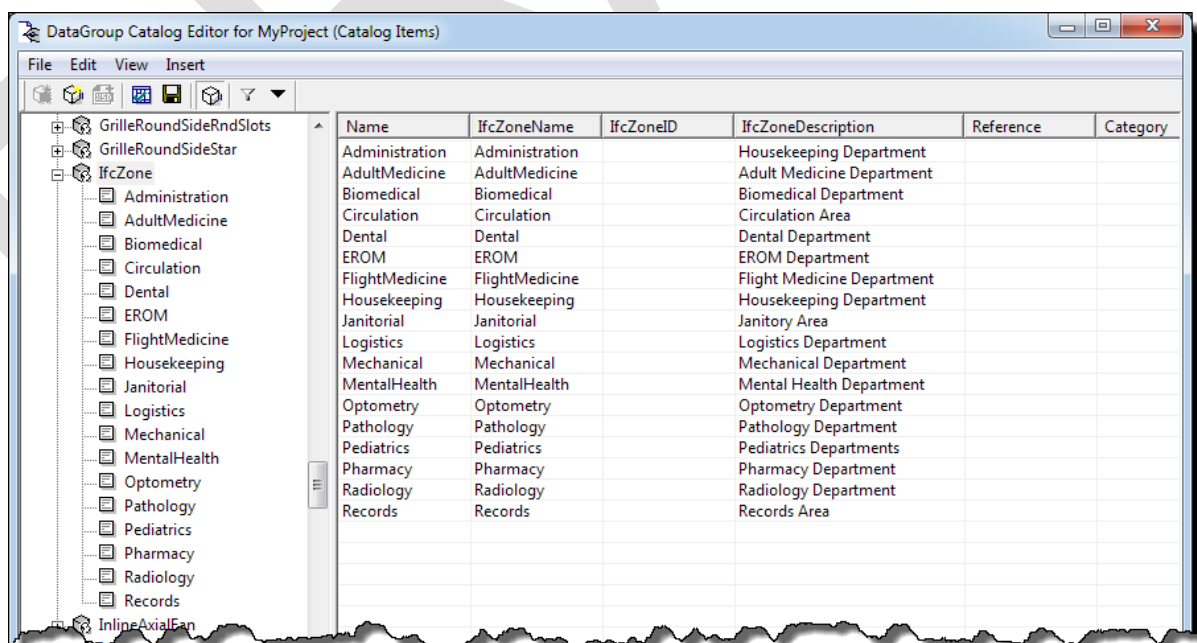
- if necessary, create a new catalog file 'Zones' in the datagrouppcatalogs folder of the project



- add default property values, e.g.



- create further IfcZones catalog items as required for the project



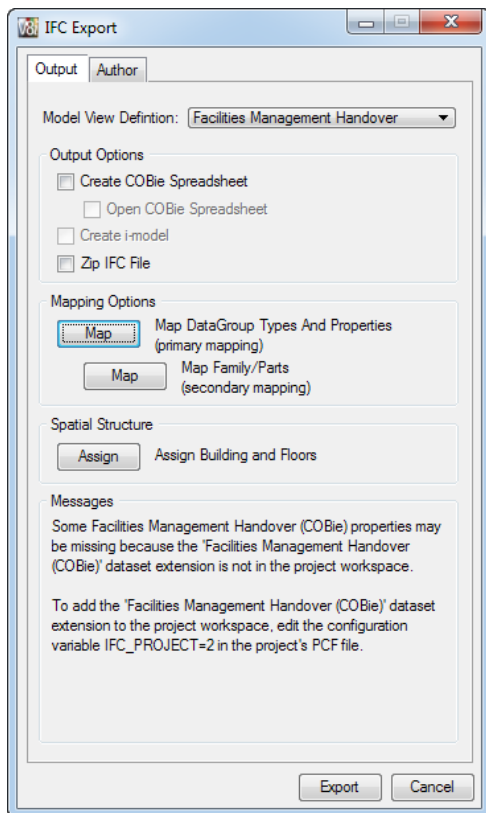
- ❑ click 'Save' to save the catalog items.

## 4.12 Map DataGroup to IFC

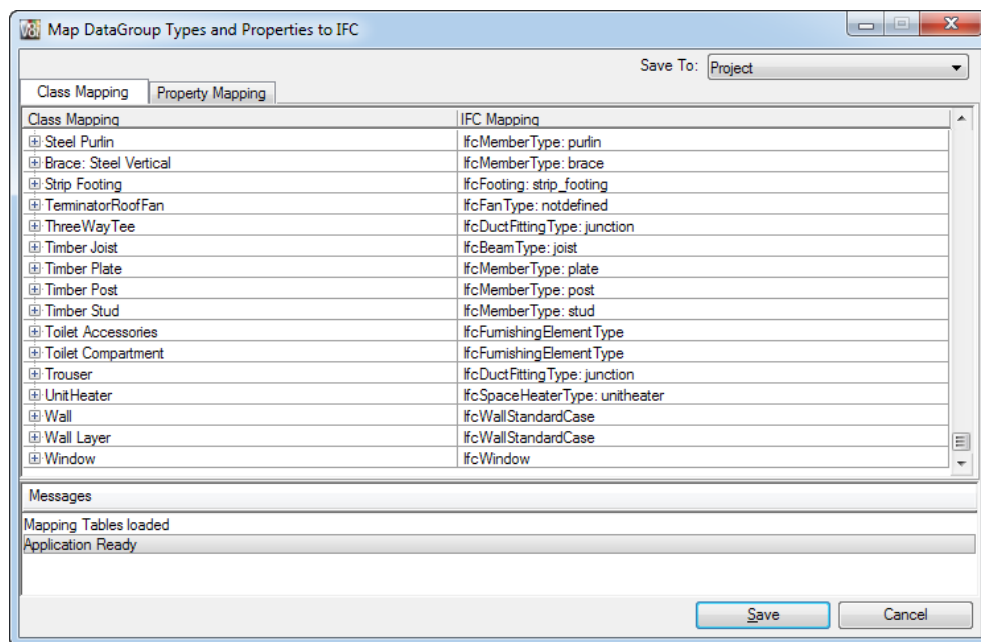
### 4.12.1 Map DataGroup types to IFC Entities/Classes

DataGroup types must be mapped to a corresponding IFC entities/class, otherwise they are recognized as IfcBuildingElementProxy and filtered out for COBie export. AECOSim Building Designer provides a tool to map DataGroup types to their corresponding IFC Class, Type and Predefined Type.

- ❑ Open the 'Map DataGroup Types and Properties to IFC' dialog box from the from the 'IFC Export' dialog File > Export > IFC...

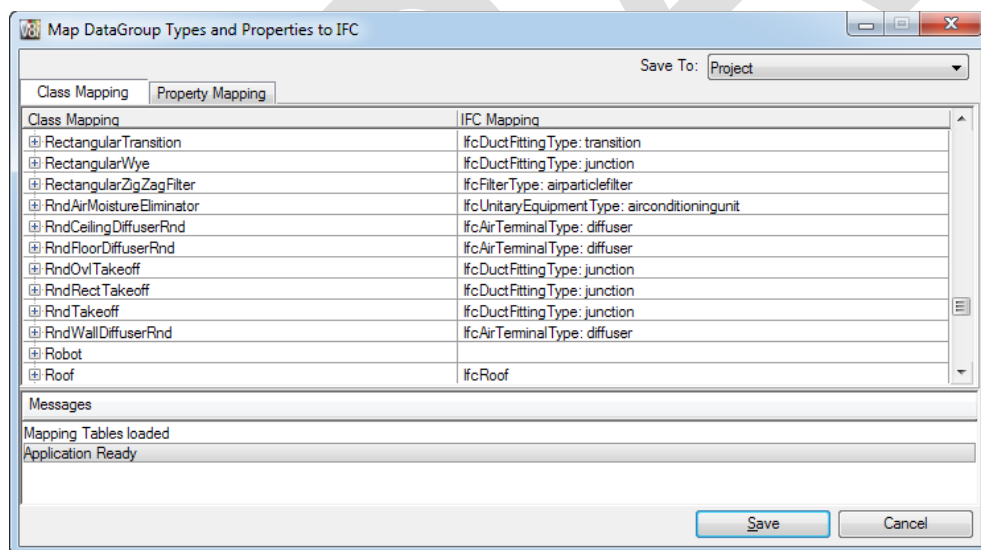


- ❑ Click 'Map' (Map DataGroup Types and Properties) to open the 'Map DataGroup to IFC' dialog box.
- ❑ If necessary, click the 'Type Mapping' tab on the 'Map DataGroup Types and Properties to IFC' dialog box.

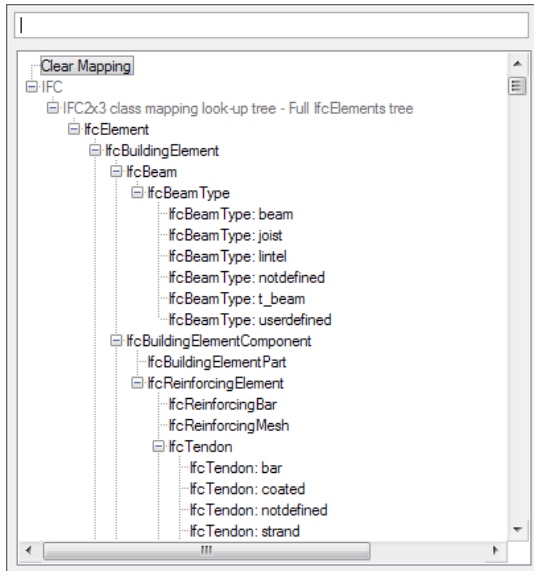


☞ All AECOSim Building Designer DataGroup Catalog Types are mapped to a corresponding IFC entities/class and should not be remapped unless necessary for a particular project

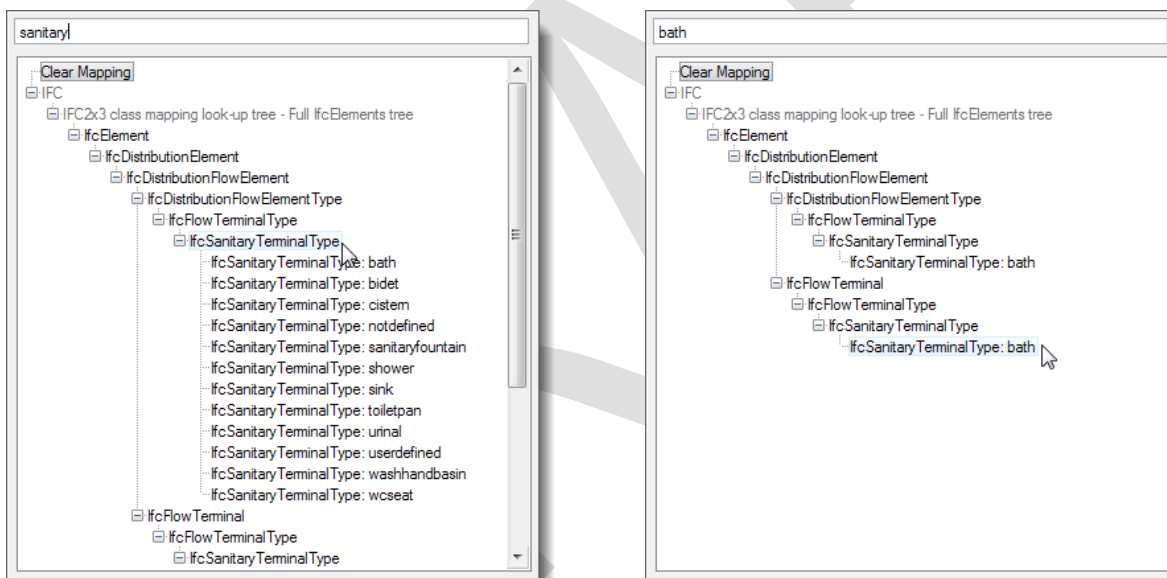
- ☐ To map unmapped DataGroup Catalog Types, click on the empty cell in the 'IFC Mapping' column to open the 'IFC 2x3 Entities Classification' dialog box.



- ❑ Scroll through the list to find the IFC Class, Type and/or Predefined Type.



- ❑ Alternatively, enter a search string in the 'Search' field to display only items that include the search string, e.g.



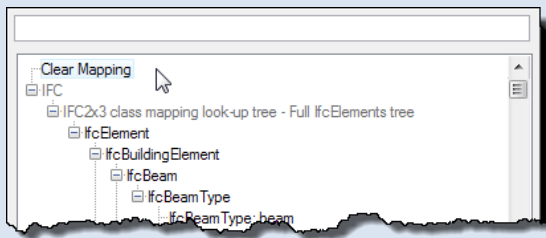
or

- ❑ Double-click on an IfcEntityClass, IfcTypeClass or IfcPredefinedType to add it to the mapping table.

Unsaved mappings are displayed in blue.

- ❑ From the 'Save to:' dropdown menu select 'Project' (default) to save the overridden mappings at the project level, or select 'Dataset' to save the overridden mappings at the dataset level.
- ❑ Click 'Save' to save the unsaved mappings and close the dialog box.
- ❑ Click 'Cancel' to close the 'Map DataGroup to IFC' dialog box without saving.

To remove a mapping, select the mapping and double click 'Clear Mapping in the 'IFC 2x3 Entities Classification' dialog box

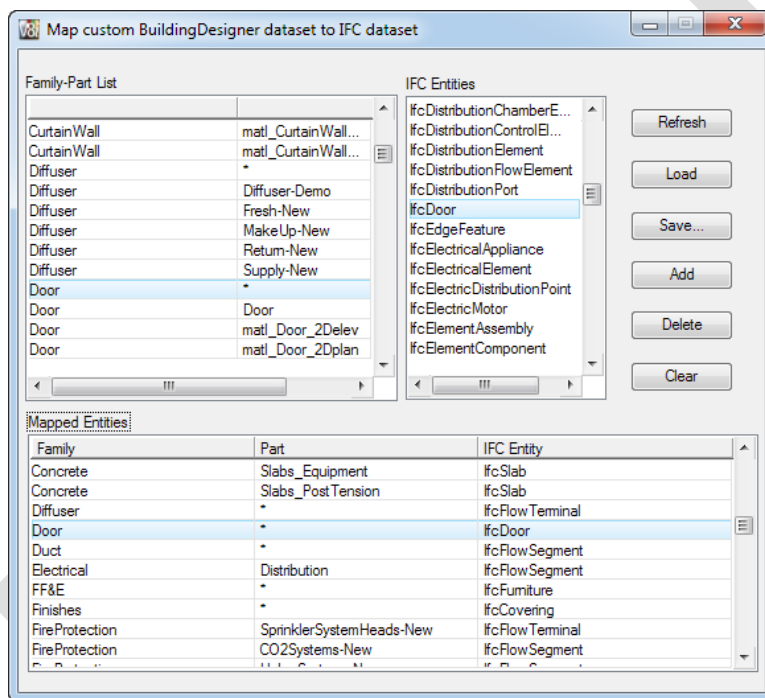


### 4.12.2 Map Parts to IFC Entities/Classes

IFC Entity/Class mapping via Families and Parts in previous versions of AECOSim Building Designer has been superseded by mapping via DataGroup Types. However, for legacy reasons and to map building components without DataGroup instance data, Part Mapping is still available as a secondary method.

☐ In the key-in browse

- key-in 'mdl load tfifc' (only necessary, if the 'IFC Export' dialog box had not been opened previously)
- key 'ifcmap' to open the 'Map custom BuildingDesigner dataset to IFC dataset' dialog box.



☐ In the 'Family-Part List' pane select one or more Parts.

Click the wildcard \* to map all Parts in the selected Family

- ☐ In the 'IFC Entities' pane select the corresponding IFC Entity
- ☐ Click 'Add' to add the mapping to the 'Mapped Entities' pane.
- ☐ Click 'Save' to save the mapping.

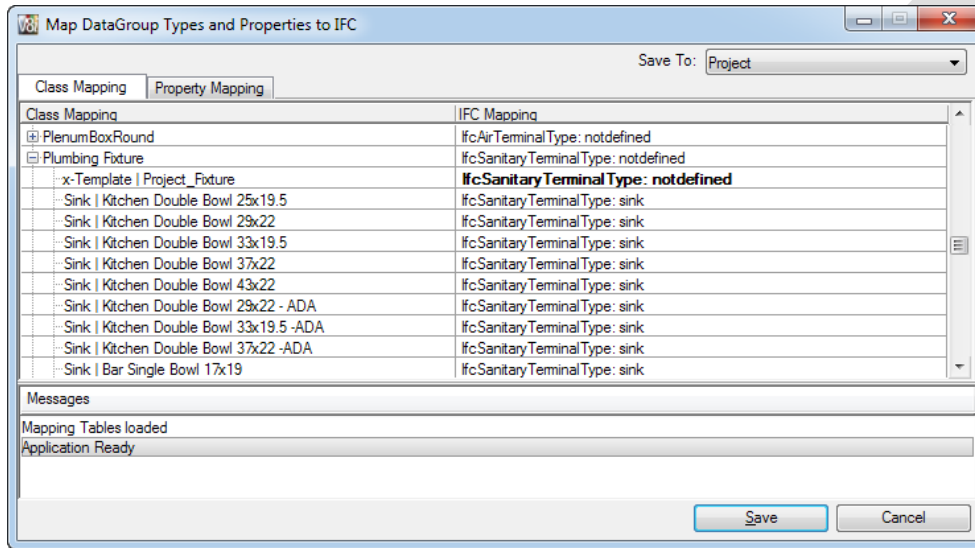
### 4.12.3 Map DataGroup items to IFC Entities/Classes

In most cases, DataGroup catalog items are correctly mapped by the 'parent' DataGroup type; therefore, each DataGroup catalog item inherits the mapping from the DataGroup type mapping of its 'parent' type. However, in some cases it may be

necessary to override the inherited mapping, e.g. to map certain wall catalog items to IfcCurtainWall. This is done in the 'Map DataGroup to IFC' dialog box by expanding the DataGroup type to display the catalog items.

Each localized dataset has different DataGroup catalog item names; hence the mapping overrides for catalog items is dataset-specific.

- ❑ Open the 'Map DataGroup Types and Properties to IFC' dialog box as described above.
- ❑ Expand a DataGroup type, e.g. Plumbing Fixture



Catalog items that 'inherit' the FC entities/class from their 'parent' Catalog type are shown in bold.

- ❑ Use the same method as described above for Type mapping to override the default item mapping to other IFC Class, Type and Predefined Type.

Unsaved mappings are displayed in blue.

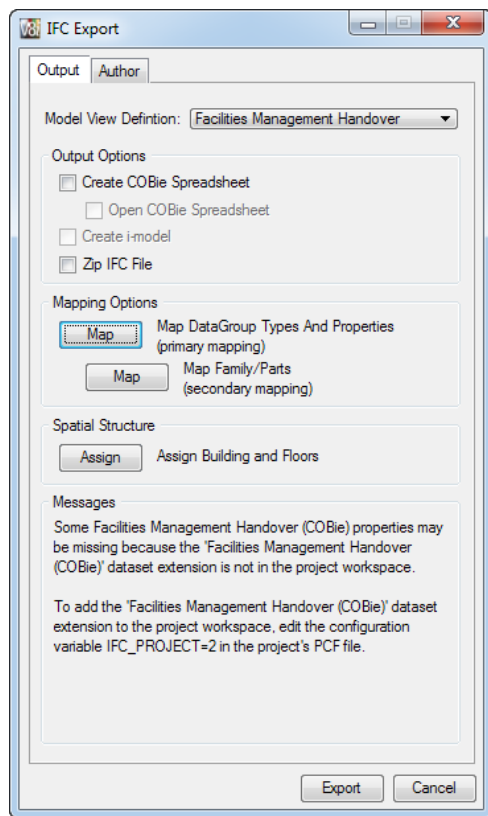
- ❑ Click 'Save' to save the overridden item mappings to either the Project (default) or Dataset.

## 4.12.4 Map duplicate properties

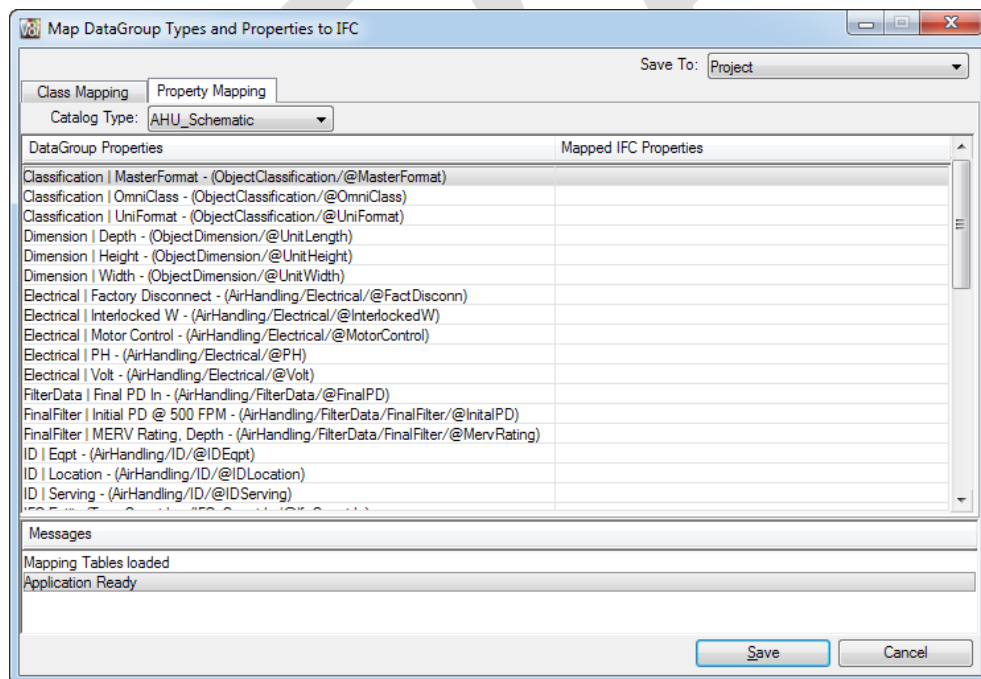
Adding IFC2x3, FM Handover/COBie or user-defined property sets to the workspace may duplicate some properties that are already in the AECOSim Building Designer schema. To avoid that property values need to be entered more than once, duplicate properties can be mapped to AECOSim Building Designer properties. Mapped properties are displayed in grey with a note to which property they are mapped to and the value field is deactivated.

AECOSim Building Designer is delivered with a property mapping table, but further properties can be mapped via the 'Map DataGroup to IFC' dialog box.

- ❑ Open the 'Map DataGroup Types and Properties to IFC' dialog box from the 'IFC Export' dialog  
File > Export > IFC...



- ❑ Click 'Map' (Map DataGroup Types and Properties) to open the 'Map DataGroup to IFC' dialog box.
- ❑ If necessary, click the 'Property Mapping' tab on the 'Map DataGroup Types and Properties to IFC' dialog box.



Note: In the 'DataGroup Properties' column, the AECOSim Building Designer and user-defined properties (if any) for the selected DataGroup type are displayed, except those in property definition files prefixed with a prefix defined by configuration variable IFC\_MAPPED\_PROPERTIES. Default is IFC\_MAPPED\_PROPERTIES=Pset\_, i.e. the

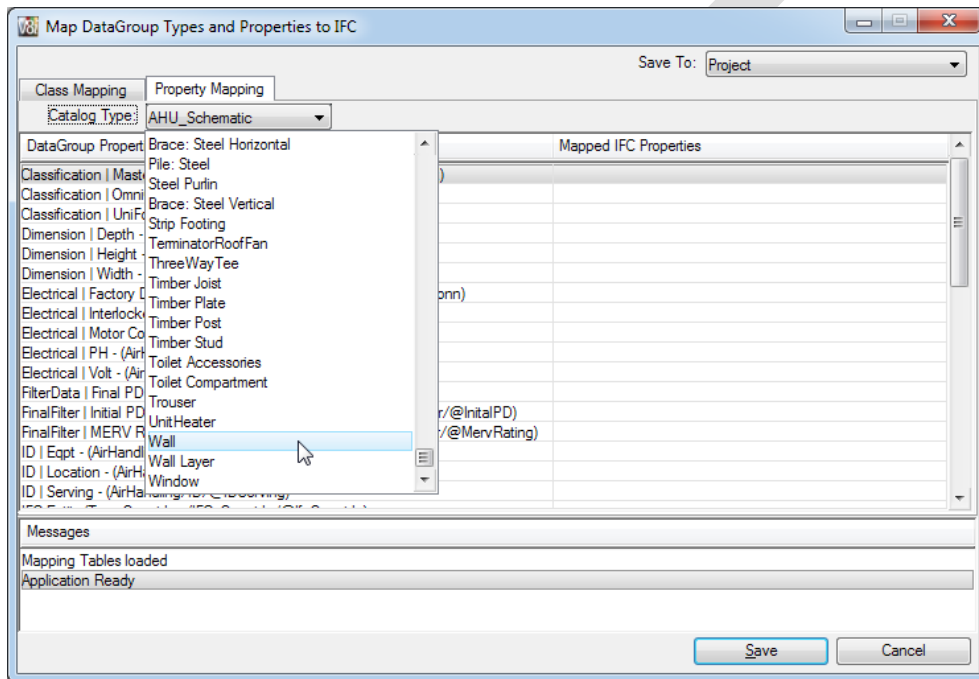
property sets defined in the buildingSMART IFC specification, typically located in  
 ...\\BuildingDatasets\\DatasetExtensions\\IFC2x3\_psets\\datagroupsystem.

☞ Only property sets defined in the buildingSMART IFC specification may be prefixed 'Pset\_'.

Consequently, in the 'Mapped Properties' column, the properties in property definition files prefixed with a prefix defined by configuration variable IFC\_MAPPED\_PROPERTIES are displayed.

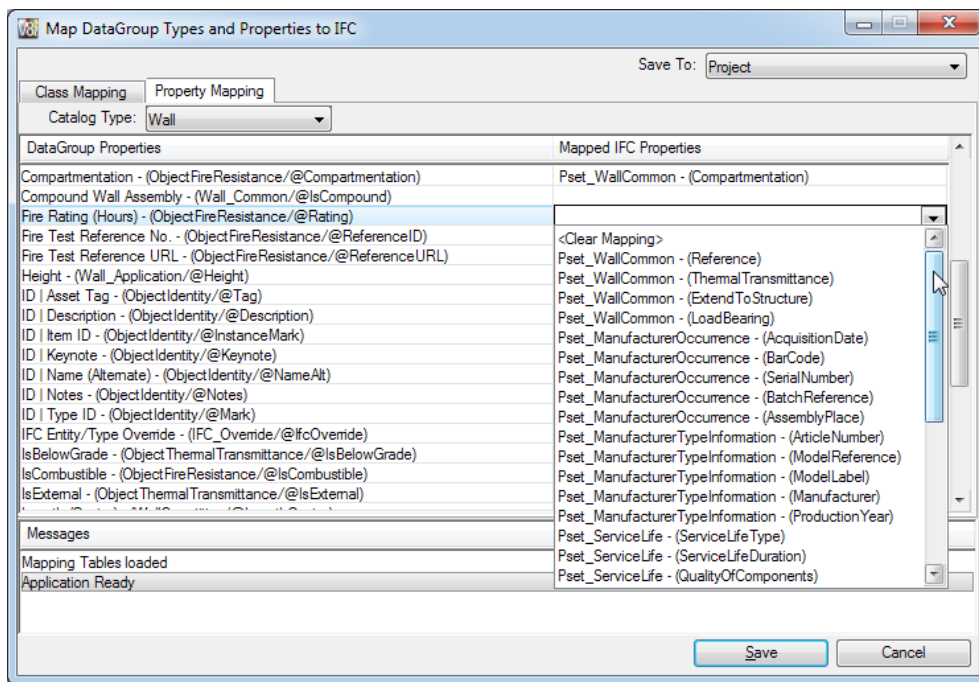
☞ Further prefixes can be added comma-delimited to the configuration variable., e.g.  
 IFC\_MAPPED\_PROPERTIES=Pset\_,ePsets.

- ☐ Select a DataGroup type from the DataGroup Type dropdown menu, e.g. 'Wall', to display all properties attached to wall.

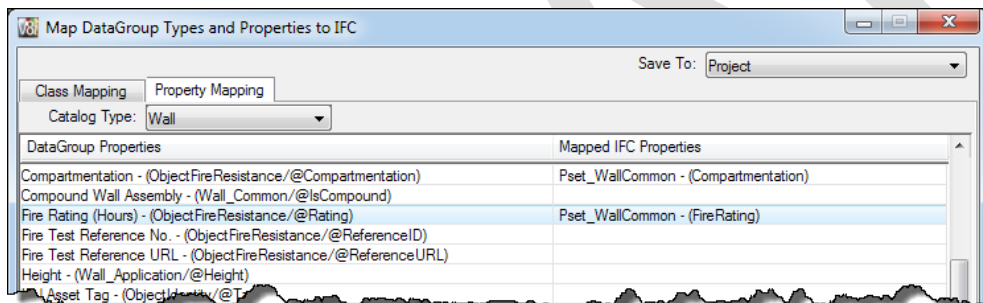


- ☐ In the Destination column cell, click in the same row as a source property to open the dropdown list of available properties in IFC2x3 property sets.





- ❑ Click on the property to transfer it to the selected cell and to close the dropdown list.

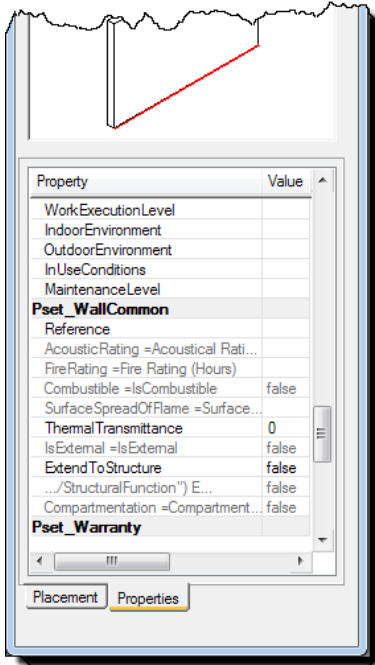


- ☞ Mapped properties are no longer displayed in subsequent dropdown listings

- ☞ To remove a mapping, select <Clear Mapping>

- ❑ Click 'Save' to save the property mappings to either the Project (default) or Dataset.

- ☞ In placement tools, a mapped property is greyed out indicating that a value cannot be added and to which property it is mapped, i.e. where the property value should be entered.

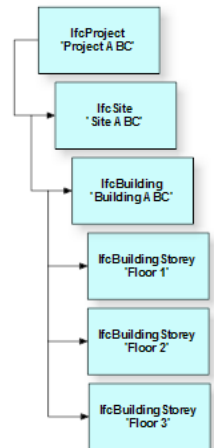
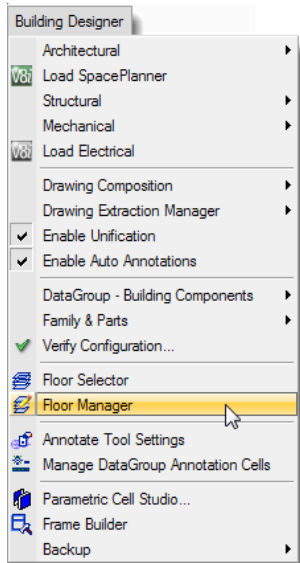


## 5 Building Information Modeling

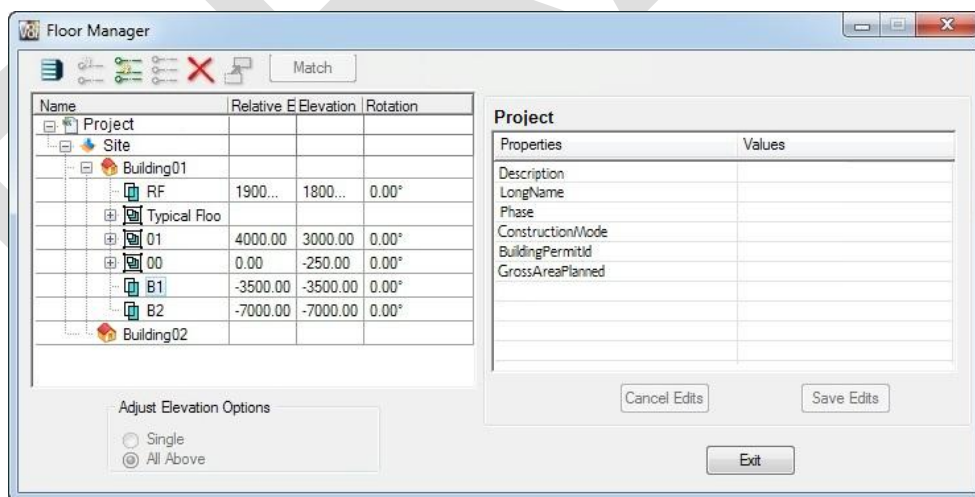
### 5.1 Define the Spatial Structure

IFC requires a clearly defined special containment structure, i.e. Project, Site, Building, (Zones,) Building Storeys, Spaces, Components. This is created in the Floor Manager.



- ❑ Start AECOSim Building Designer and open a DGN in the project
- ❑ Open the Floor Manager from the Building Designer menu




- ❑ In the 'Floor Manager'
    - **Project:** select the Project and enter Project property values in the right pane as required.
    - click 'Save Edits'
- Note: The configuration variable '\_USTN\_PROJECTNAME' defines the displayed project name; only one project is allowed. To change the project name, double-click the name and key-in a new name.



- **Site:** double-click the default site name, rename it and enter Site property values in the right pane as required.
  - click 'Save Edits'
- Note: Only one site is allowed.

- **Building:** if already included in the hierarchy, double-click the default building name, rename it and enter Building property values as required; otherwise select the site and click the  button to create a new building.
- click 'Save Edits'
- Note: Multiple buildings are allowed.
- **Floor:** if already included in the hierarchy, double-click the default floor name, rename it and enter Floor property values as required; otherwise select a building and click the  button to create a new floor.
- click 'Save Edits'
- enter Relative Elevation and Elevation
- optional: change the Rotation angle

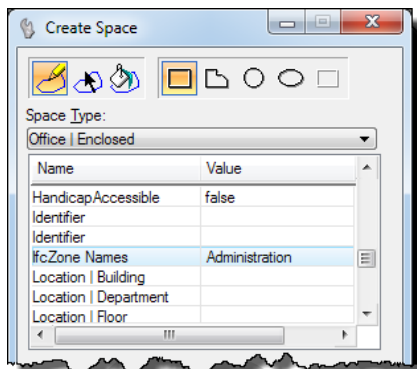
 For spatial containment, the floor range is calculated based on the floor elevation values in the Floor Manager.




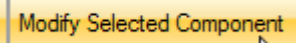
- Click 'Exit' to close the Floor Manager.

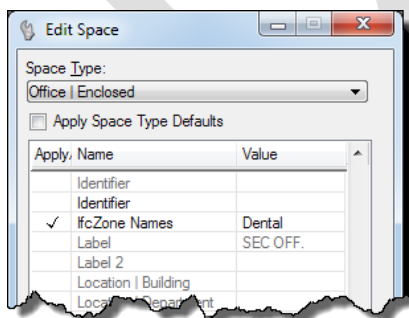
## 5.2 Add Zone information to Spaces


If required, Spaces can be assigned to one or more Zones via the Space property 'IfcZone Name' either when creating spaces or afterwards.

- Open the 'Create Space' tool
- Scroll to the property 'IfcZone Name' and enter one or - comma delimited - more valid Zone names



- After placement: Open the 'Edit Space' dialog box with the 'Modify Existing Instance Data' , 'Modify Properties of Selected Elements'  tool, or right-click option 'Modify Selected Component'  



 The IfcZones Names must be identical to the DataGroup catalog item names.


## 5.3 Assemble the Building Information Model

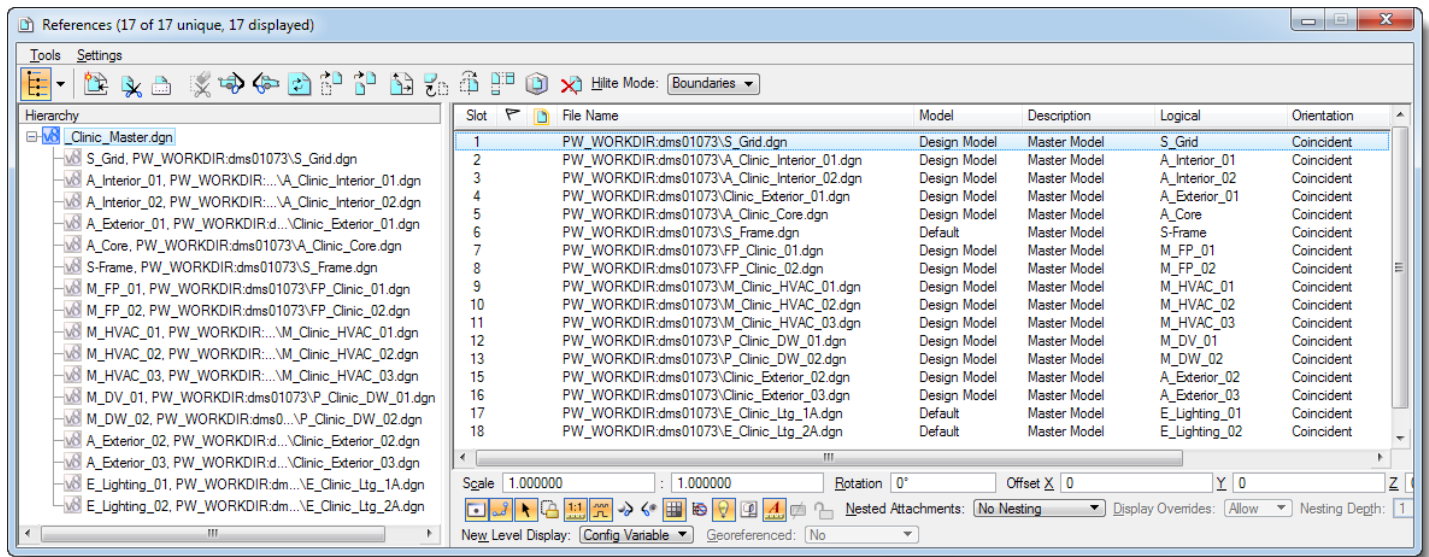
AECOSim Building Designer supports the creation of 'federated' Building Information Models (BIMs), which allows clear separation of responsibilities by discipline while facilitating accurate cross-discipline coordination. In addition, the FM Handover/COBie ModelView Definition requires separation by building storeys. Therefore, it is highly recommended to create separate 'sub-BIMs' for each discipline and floor, which are then assembled via references to the complete BIM as required for the generation of documentation (plans, sections, elevations, details, etc.), design visualization, schedules, clash detection, analyses (structural, thermal, lighting, etc.), IFC export, and more.


While there are many different ways to assemble federated BIMs, a typical strategy is to create an empty 'master' model and then to reference in the 'sub-BIMs', either 'flat', 'by discipline', or 'by floor'.



- ❑ Start AECOSim Building Designer and create a 'Master.dgn'

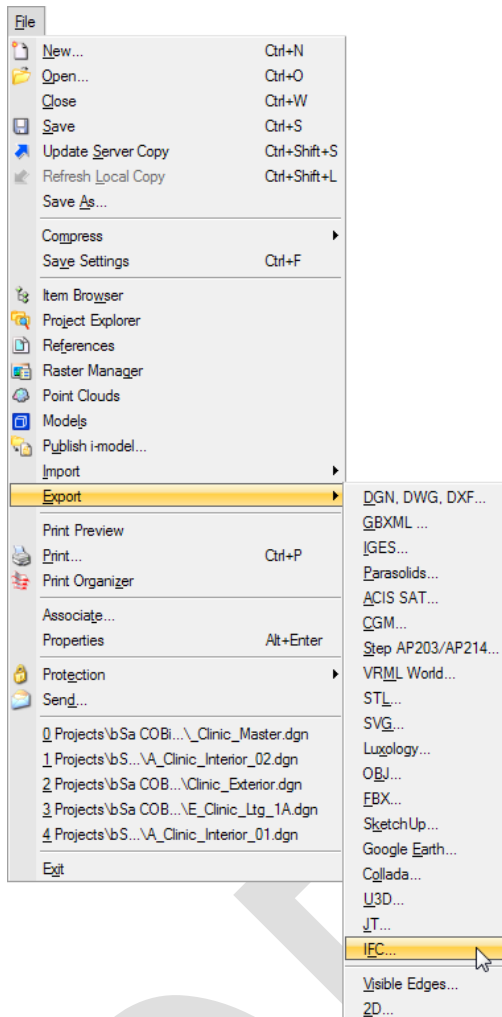
- ❑ Open 'Master.dgn' and click on 'Reference'  to open the 'References' dialog box.



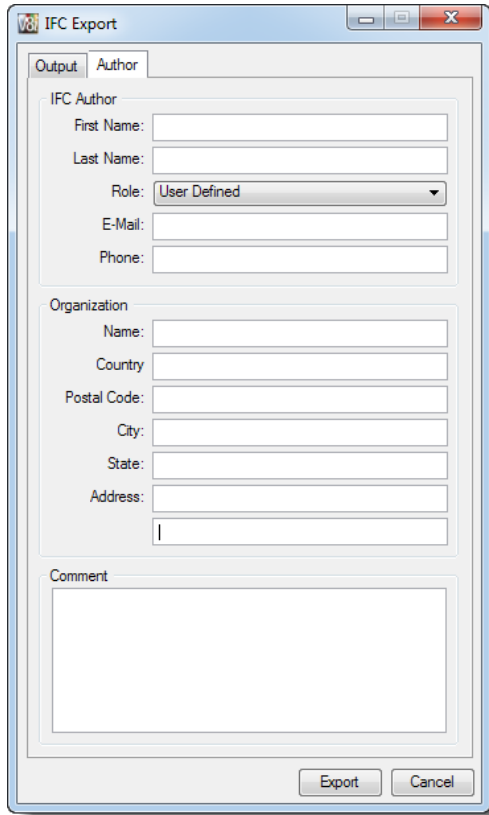
- ❑ Click the 'Attach Reference' button  to attach references.

## 6 IFC Export

- ❑ Click 'IFC' from the 'File' > 'Export' menu list to open the 'IFC Export'



- ❑ In the 'IFC Export' dialog box, click the 'Author' tab.

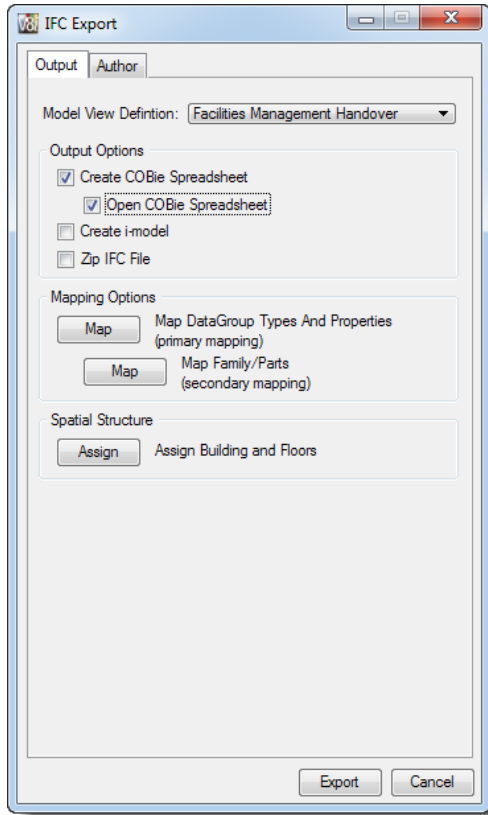


The screenshot shows the 'IFC Export' dialog box with the 'Author' tab selected. The dialog is divided into three main sections: 'IFC Author', 'Organization', and 'Comment'. The 'IFC Author' section contains fields for 'First Name', 'Last Name', 'Role' (a dropdown menu currently showing 'User Defined'), 'E-Mail', and 'Phone'. The 'Organization' section contains fields for 'Name', 'Country', 'Postal Code', 'City', 'State', and 'Address'. The 'Comment' section is a large text area. At the bottom right of the dialog are 'Export' and 'Cancel' buttons.

- ❑ In the 'Author' tab
  - fill in information about the 'IFC Author' and 'Organization'
  - add a comment



- ❑ In the 'IFC Export' dialog box, click the 'Output' tab



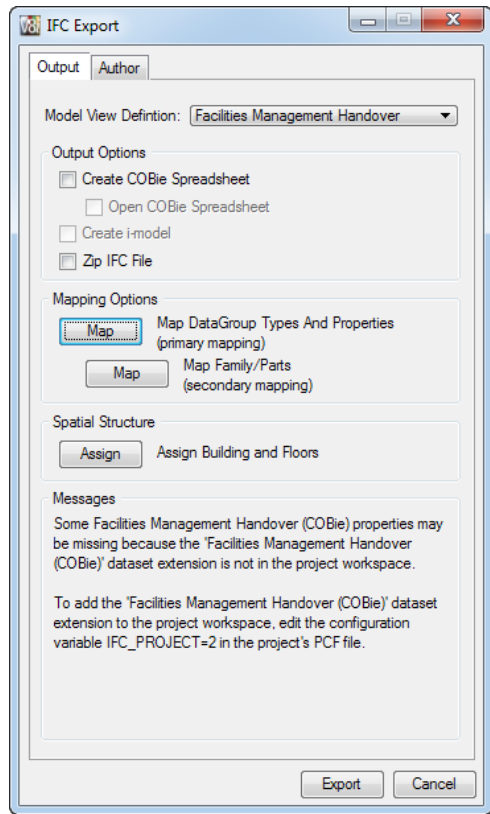
- ❑ Select a Model View Definition (MVD) from the 'Model View Definition' dropdown list.

👉 A message is displayed in the message field if the 'IFC2x3' or FM Handover/COBie' dataset extension is not part of the project's workspace, including information how to add it.

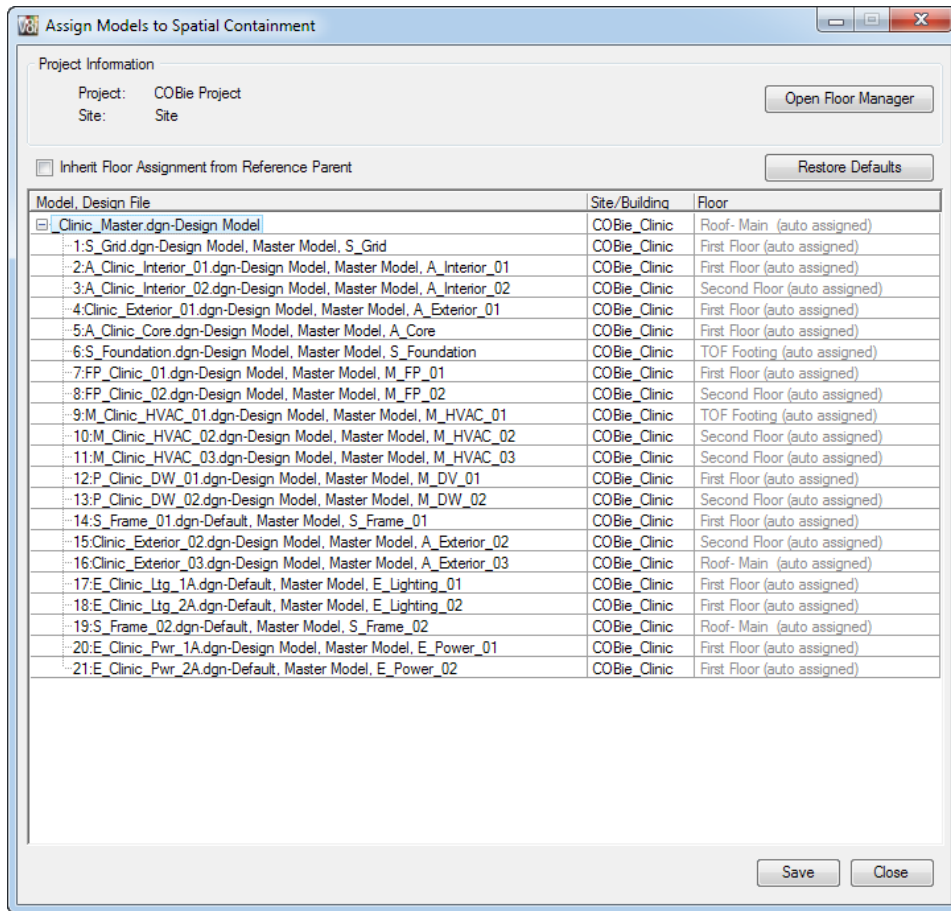
- ❑ If 'FM Handover/COBie' is selected, select 'Output Options' as required
  - check the 'Create COBie Spreadsheet' box to create the COBie spreadsheet
  - check the 'Open COBie Spreadsheet' box to open the COBie spreadsheet once it has been created
  - check the 'Create i-model' box to also create an i-model
  - check the 'Zip IFC file' box to zip the IFC file once it has been created
- ❑ If necessary, click 'Map' Map DataGroup Types and Properties' to open the 'Map DataGroup Types and Properties to IFC' dialog box (see [Map DataGroup types to IFC Entities/Classes](#)).
- ❑ If necessary, click 'Map' Map Family/Parts' to open the 'Map custom Building Designer dataset to IFC' dialog. (see [Map Parts to IFC Entities/Classes](#))  
 Note: this is a secondary option for elements that do not have DataGroup instance data attached.

## 6.1 Assign Buildings and Floors

- ❑ Open the 'Map DataGroup Types and Properties to IFC' dialog box from the 'IFC Export' dialog  
 File > Export > IFC...

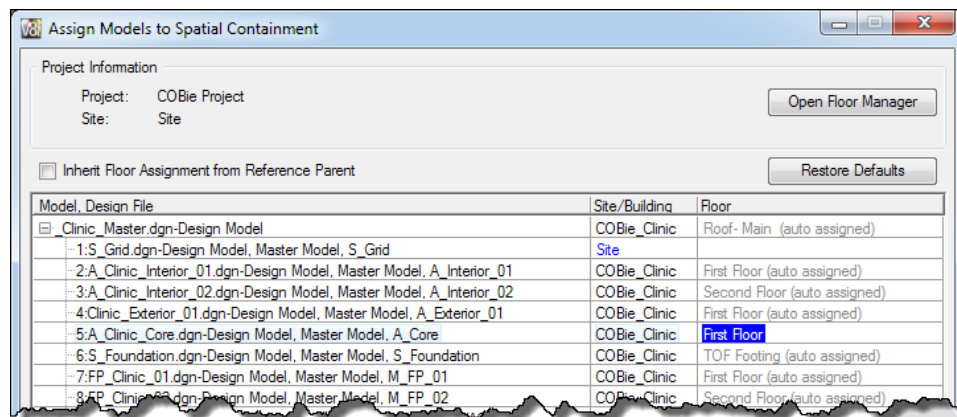


- ❑ Click 'Assign' (Assign Building and Floors) to open the 'Assign Models to Spatial Containment' dialog box.

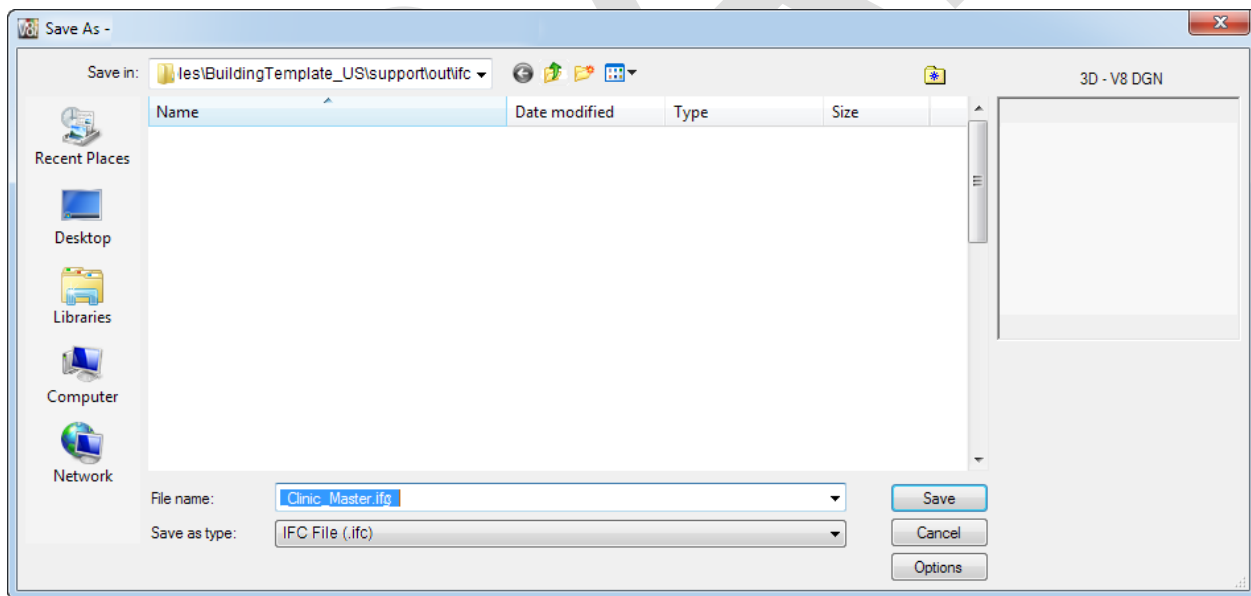


Note: The default 'Site/Building' assignment of the active model and all references is to the first (and typically only) building defined in the Floor Manager, whereas the default 'Floor' assignments are 'Auto Assigned', i.e. they are assigned to the Floor into which range the building components in the model fall. The range is defined by the floor elevation value and the elevation value of the floor above

- ❑ In the 'Assign Models to Spatial Containment' dialog box
  - to override the current 'Site/Building' assignment of a model or reference, click the corresponding 'Site/Building' cell and select another building from the dropdown list;  
Note: if 'Site Name' is selected, the Floor assignment is removed and disabled.
  - to override the current 'Floor' assignment, click the corresponding 'Floor' cell and select a floor, 'Auto Detect', or '(None)' from the dropdown list;  
Note: if '(None)' is selected, the model or reference is assigned to the Building assigned in the 'Site/Building' column.



- check the 'Inherit Floor Assignment from Reference Parent' to assign all nested references to the same Building/Site and Floor as the 'parent' model.
  - Click 'Open Floor Manager' to open the 'Floor Manager' dialog box (see [Define the Spatial Structure](#))
  - Click 'Restore Defaults' to restore the original default assignments to Site/Building and auto-assigned floors
  - Click 'Save' to save the assignments and to close the dialog box
  - Click 'Close' to close the dialog box
- ❑ In the 'IFC Export dialog box, click 'Export' to export the model to IFC (STEP Physical File Format, SPFF).
  - ❑ Optional: In the 'Save As' dialog box, override the 'Save in' folder (default = ...support\out\ifc) and File name (default = name of DGN with .ifc extension).

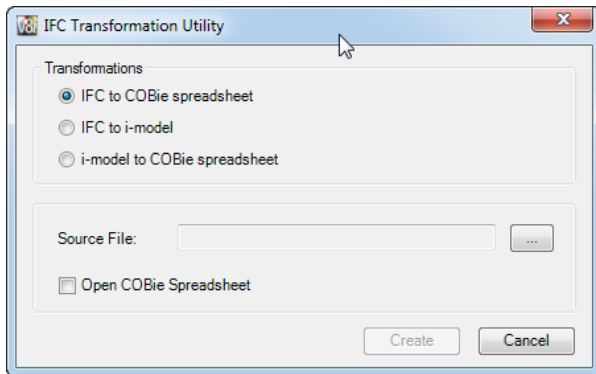
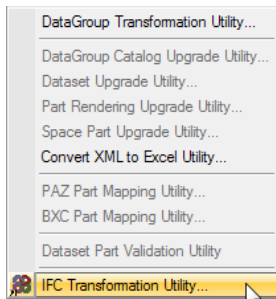


- ❑ Click 'Save' to initiate the export of the IFC file.

Note: if 'Create COBie spreadsheet' is checked, another 'Save As' dialog box opens to override the 'Save in' folder (default = ...support\out\ifc) and File name (default = name of DGN with .xml extension).

## 7 IFC Transformation Utility

- ❑ Open the IFC Transformation Utility from the 'Utilities dropdown menu option 'IFC Transformation Utility...'.



- ❑ Select one of the 'Transformations' buttons to determine the transformation type:
  - IFC to COBie spreadsheet
  - IFC to i-model
  - i-model to COBie spreadsheet
- ❑ Click the 'Source File' button and navigate to the directory to select the source file; the default folder location is the folder defined by TFDIR\_IFC.


☞ If 'Transformation Type' is 'IFC to i-model' or 'IFC to COBie spreadsheet', only files with the extension .ifc are available for selection; if 'Transformation Type' is 'i-model to COBie spreadsheet', only files with extension .i.dgn.

- ❑ Check the 'Open COBie spreadsheet' box to open the COBie spreadsheet after it has been created.

☞ Microsoft Excel needs to be installed, otherwise a notice is displayed: 'Cannot open COBie spreadsheet, because Excel is not installed.'

- ❑ Click 'Create' to start the process.

☞ Note: If the header does not include 'FMHandOverView', a warning message is displayed: Some Facilities Management Handover (COBie) properties may not be included because the source file is not a 'Facilities Management Handover (COBie)' file.

 If the file was not created by 'AECOsims Building Designer', a warning message is displayed or added to the warning message above:  
The COBie spreadsheet may be incomplete because the source file was not created by AECOsims Building Designer.

- ☐ Click 'Continue' on the warning message to close the warning message and start the process, click 'Cancel' to close the warning message without processing.
- ☐ Click 'Cancel' to close the 'IFC Transformation Utility'.