



Course: BIM introduction. Block 2: Working with BIM data. Lecture 2.4

IFC as a data exchange format

Lecture Notes

Author(s)/Organisation(s):

Anders Östman (Novogit AB)

License



<https://creativecommons.org/licenses/by/4.0/>

Version

Version 2.0

Date: 2025-04-29

Learning outcomes

At the end of this lecture, the learner is expected to be able to

- Explain the role of IFC in the building and construction industry.
- Explain basic concepts used in IFC, such as
 - Objects and entities, inheritance and properties
 - Basic IFC entities such as IfcWall
- Describe the basic structure of an IFC file in STEP format.
- Analyse the content of an IFC file and relate its STEP entities to the IFC standard.

Summary



The aim of this lecture is to briefly describe the structure and content of an IFC file. The background of the standard is described as well as its importance for the development of the BIM sector. The overall content of the BIM schema is then described, as well as the basic structure of the STEP format, which is the de-facto standard used for serializing the IFC model. An example of how to interpret an IFC file is also provided.

Expected competences when entering the lecture

- Basic knowledge about BIM
- Basic knowledge about relational databases

Expected Workload

20 slides with course learning content, 2 classroom hours, 0.2 ECTS (ECVET)

Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



Revision History:

Revision	Date	Author(s)	Status	Description
0.1	2023-05-12	A. Östman	Draft	Table of Contents
0.2	2023-05-30	A. Östman	Draft	Introduction, IFC encoding added
0.3	2023-06-09	A. Östman	Draft	Object inheritance and IFC objects added
0.4	2023-08-25	A. Östman, A. Kubart	Draft	Lecture notes added
0.5	2024-02-06	A. Östman	Draft	Update based on review comments
1.0	2024-03-11	A. Östman	Published	Edited by M. Morbidini
1.1	2024-12-30	A. Östman	Published	Updated based on CR's
2.0	2025-04-29	A. Östman	Published	Updated EU logo and disclaimer. Edited by T. Näslund

Content of the lecture:

Preparation	5
The IFC standard	6
IFC Evolution	7
Historical background	8
The role of IFC in BIM collaboration	9
BIM collaboration – some benefits	10
Interoperability levels	11
Dataset schema	12
IFC Data Schema	13
Object Inheritance	14
Analysing an IFC file using usBIM.viewer	15
Open IFC file in Text Editor	16
STEP file structure	17
Finding entity number	18
Entity inheritance	19
Inherited properties for #27421	20
Analysing an IFC file using FreeCAD	21
Finding entity number using FreeCAD	22
Analysing IFC tree in FreeCAD	23
IFC Model Views and file management	24
Summary	25
References	26

Preparation

Before the lecture, the students and the tutor should have downloaded and installed the following resources.

usBIM.viewer+ (free viewer, <https://www.accasoftware.com/en/freeware/usbim.viewer>)

FreeCAD software (optional). <https://www.freecad.org/downloads.php>

An IFC file (<https://www.ifcwiki.org/index.php?title=File:AC20-FZK-Haus.ifc>)

In addition, the students and tutor should have online access to the following resource

Technical documentation of IFC version 4.0.2.1

(https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/).

The IFC standard

IFC as a data Exchange format



The IFC standard



- IFC (Industrial Foundation Classes) is a standard dealing with processes, data, terms and change management.
- Aims to reduce costs by enabling better collaboration and digital workflows
- Developed by buildingSMART International, <https://www.buildingsmart.org/>
- Open ISO standard ([ISO 16739-1:2018](https://www.iso.org/standard/67391.html))
- Often used for transferring BIM data
- Is meant to be a reference copy of the design
 - "The BIM version of PDF's"
 - <https://youtu.be/9YqXXbdoh00>


5


Industrial Foundation Classes (IFC) is a standardized data schema for the built asset industry. It is an open, international standard (ISO 16739-1:2018) and promotes vendor-neutral and usable capabilities across a wide range of hardware devices, software platforms, and interfaces for many different use cases. The main usage of IFC is for transferring BIM data between organisational entities, between different companies as well as within a single company.

“buildingSMART is the international authority for a set of standards known as the Industry Foundation Class (IFC) which deal with process, data, terms and change management for the specification, management and effective utilization assets in the built asset industry” (<https://www.buildingsmart.org/>). In the GIS domain, the standards are mainly dealing with data and terms. In the built asset industry, the scope of standardisation is wider, also including processes and change management.

IFC Evolution

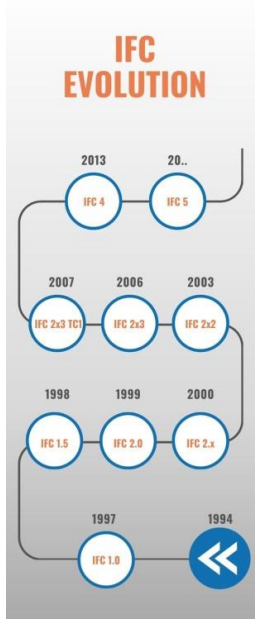
IFC as a data exchange format

 **BIRGIT**

 Co-funded by
the European Union

IFC Evolution

Source: <https://www.buildingsmart.org/>



The diagram illustrates the evolution of the IFC standard from 1994 to 2013. It shows a sequence of versions: IFC 1.0 (1994), IFC 1.5 (1998), IFC 2.0 (1999), IFC 2.x (2000), IFC 2x2 (2003), IFC 2x3 (2006), IFC 2x3 TC1 (2007), IFC 4 (2013), and IFC 5 (20..). A double arrow points from IFC 1.0 to IFC 4, indicating the progression of the standard.

6

In 1994, buildingSMART started the development of the IFC standard. In 1997, its first version (IFC 1.0) was released.

Through the years, new versions have continuously been released and the most recent version now is IFC 4, which was released in 2013. From the very beginning, the IFC standard has been closely associated with the EXPRESS modelling language and the STEP format for file transfer.

We now see an increasing demand for new use cases, such as Digital Twins, Smart Cities and real-time sensor data. To meet new requirements imposed by new type of usages, a new version of IFC (IFC 5) is currently being developed, but a date for its release is not yet scheduled.

Historical background

IFC as a data Exchange format

Historical background

- In object-oriented programming, a class is a specification of a set of objects. A class consists of code (procedures or methods) and data (attributes or properties)
- In 1992, Microsoft introduced Microsoft Foundation Classes (MFC). This is an object-oriented library for developing C/C++ applications for Windows. By using Windows application calls, external software could be made looking like Windows applications. The most popular classes dealt with menus, controllers and all windows-like gadgets.
- In 1994, BuildingSMART started to develop Industrial Foundation Classes, which is an object-oriented library for reusing BIM data objects. The goal was to use IFC-based application calls, when reusing data from other BIM models
- Also in 1994, the Open Geospatial Consortium (OGC) was formed, aiming to improve interoperability in the GI sector. The first OGC standards specified standardised application calls to be used by various GIS software vendors
- A few years earlier, in 1989, the World Wide Web was invented. It was made public in 1991. The usage of WWW was quite limited in 1994, when the development of IFC and OGC standards started. As web technologies became more mature, the interest of data sharing increased and the API calls using internet protocols (HTTP) was more in focus of the developments

7

The term “Industry Foundation Classes” has its origin in the vocabulary used during the 1990’s. The term “class” is used in object-oriented programming, and it is a specification of objects used by the system. A class specifies procedures (or methods) in the form of software code, but also a specification of data associated with an object.

In 1992, Microsoft introduced Microsoft Foundation Classes (MFC). This was an object-oriented library, by which a developer could develop software applications for Windows, that had a look similar to ordinary windows application. The most popular classes dealt with menus, controllers, forms and other windows gadgets.

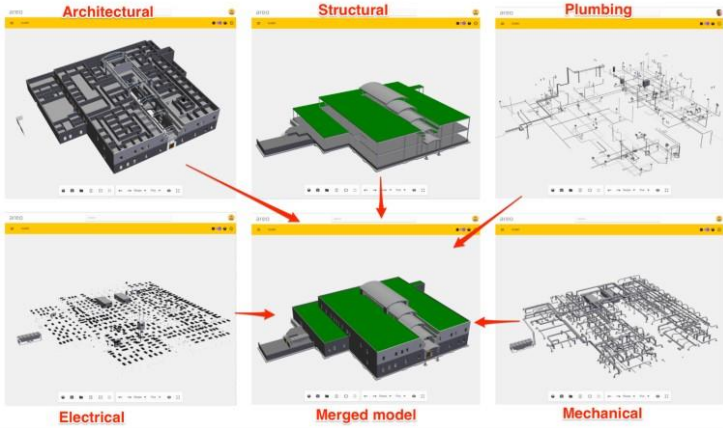
In 1994, buildingSMART started their development of IFC. Due to the success of MFC, buildingSMART decided to adopt a similar approach and the name “Industrial Foundation Classes” was coined. The IFC did not have any executable code in the classes, only a specification of data. But the intention was to use the IFC objects in a similar way as MFC, namely through standardised API’s.

The building industry was not the only sector adapting an MFC-based approach. In the geospatial sector, the Open Geospatial Consortium (OGC) also developed similar standards. However, remote application calls did not always work well, so applications often needed to be executed in one single environment. This of course reduced the flexibility and as web technologies became more widely spread, the industry gradually moved to HTTP-based API’s and file transfer mechanisms.

The role of IFC in BIM collaboration

IFC as a data Exchange format

The role of IFC in BIM collaboration



From <https://blog.areo.io/whatis-ifc/> During design and construction each discipline typically have their own model. The models are merged or referenced for design and production coordination tasks

8

There are usually several different professionals participating in a building and construction project, such as plumbing designers and architects. Each profession is using software tools that are suited to their work. To be able to collaborate, the professionals need to have access to the work carried out by other professionals. IFC is the standard to be used when transferring data between the different professional entities. In principle, one type of professionals should not be able to change the work made by other professionals. If less suitable solutions are detected, they must be discussed and jointly resolved. In that sense, an IFC file being transferred between organisational entities serves as a reference copy of the design before changes are made and documented.

IFC plays a vital role in BIM collaboration by enabling different project stakeholders to share information seamlessly. This is essential for ensuring that everyone is working from the same data and that any changes made to the model are reflected in all downstream applications.

An IFC file is often considered as the BIM equivalent to PDF. An IFC file is typically not intended to be directly edited. It is primarily used as a standardized file format for exchanging BIM data between different software applications and tools. It is often important to save the transactions (IFC files) between different parties in case legal disputes will arise.

BIM collaboration – some benefits

IFC as a data exchange format

BIM collaboration – some benefits

- Improved communication and coordination
 - A common language for exchanging information
- Increased efficiency
 - Streamlining workflows
- Reduced costs
 - Eliminating the need to convert BIM data between different software applications
- Supported by a wide range of BIM software applications.
 - This makes it easy for project stakeholders to exchange BIM data regardless of the software they are using.

9

Some of the benefits to be achieved when using IFC for BIM collaboration include the following:

- Improved communication and coordination: IFC allows project stakeholders to communicate and coordinate more effectively by providing a common language for exchanging information. This can help to reduce errors and omissions, and to improve the overall quality of the project.
- Increased efficiency: IFC can help to increase efficiency by streamlining the workflow for exchanging BIM data. This can save time and resources and allow project teams to focus on more important tasks.
- Reduced costs: IFC can help to reduce costs by eliminating the need to convert BIM data between different software applications. This can also help to reduce the risk of errors and omissions.
- IFC is supported by a wide range of BIM software applications, including Autodesk Revit, Bentley MicroStation, and Trimble Tekla Structures. This makes it easy for project stakeholders to exchange BIM data regardless of the software they are using.

Interoperability levels

IFC as a data exchange format



Interoperability levels

- **Interoperability** is a characteristic of a product or system to work with other products or systems.
- Syntactic interoperability: Common **data formats**. Examples: XML, SQL, STEP
- Semantic interoperability: The systems have a shared understanding of the **meaning** of the concepts
- Organisational interoperability: The output of a process in one organisation can directly **be used in another process** in another organisation

10

To enable collaboration through the BIM lifecycle, a certain level of interoperability is needed. Interoperability may be defined as “a characteristic of a product or system to work with other products or systems”.

The concepts related to interoperability are often described at different levels. At the lowest level, we have syntactic interoperability. This means that an IT system can read data coming from another system (or software package). The two systems can work with a common data format, for instance XML. The most common data format (encoding) for IFC is the STEP format, as described later in this lecture.

At the next interoperability level (semantic interoperability), the systems have a shared understanding of the meaning of the terms being used. In the IFC standard, different building elements are specified, for instance IfcSlab and IfcDoor. The IFC standard defines the meaning of the terms being used, in this case the names of the objects (entities).

Organisational interoperability is the third interoperability level, sometimes also called cross-domain interoperability. This means that the output of a process in one organisation can directly be used in another process in another organisation. We will not deal with this level of interoperability in this lecture, because it is organisational dependent.

A higher level of interoperability provides the opportunities to a higher degree of automation (cost reductions). To apply higher degrees of interoperability, the usage of standardised dataset schemas is required.

Dataset schema

IFC as a data exchange format

Dataset schema

- The **schema of a dataset** is describing the structure of the dataset, often by using a formal language.
- The term "schema" refers to the organization of data as a blueprint of how the dataset is constructed
- A relational database consists of table, where each table has a name and a set of columns. The schema specifies, among other things, the names of the tables and the names, data type and other characteristics of the columns of each table. In many relational database systems, the schema is also stored as tables, but having reserved names and columns and cannot be modified by ordinary users. In such cases, the specification of the schema is made by using SQL statements.
- An XML file consists of elements, sub-elements and attributes. An XML schema (.xsd file) specifies the names of the elements and attributes as well as other properties (cardinality etc). The formal language for specifying an XML schema is also XML.

11

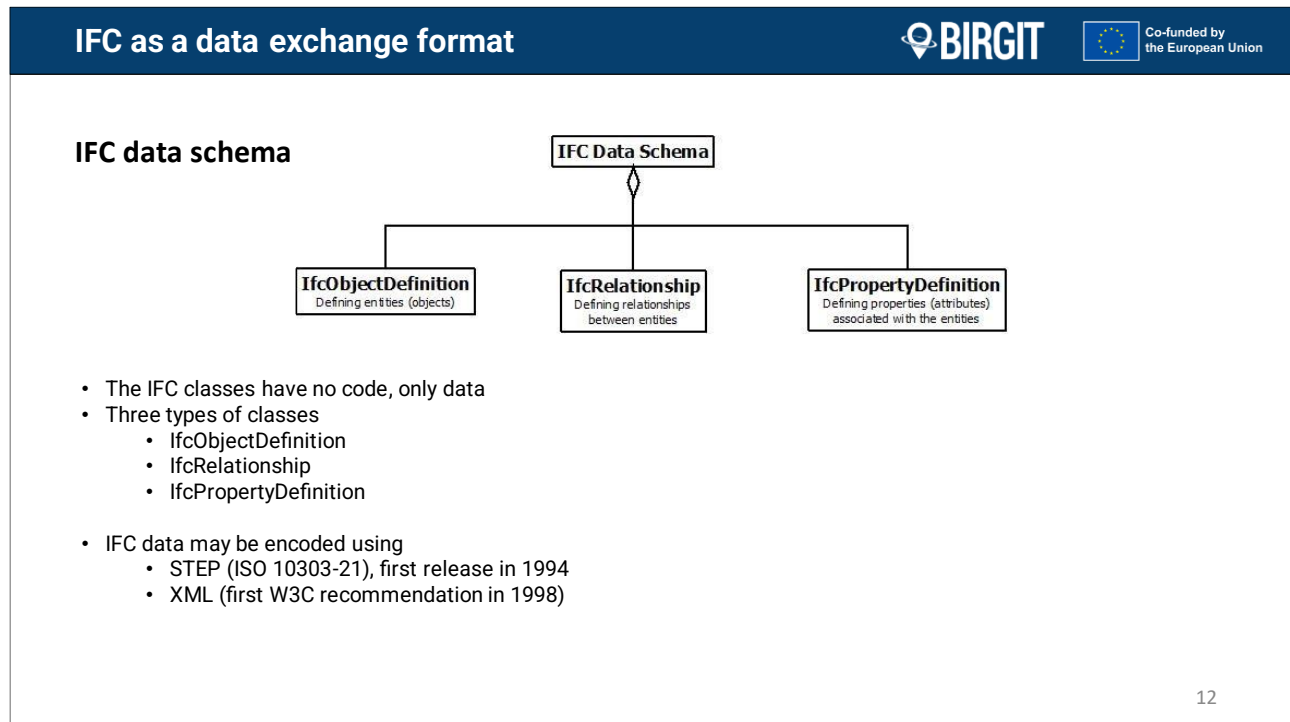
A dataset schema is describing the structure of a dataset. Depending on type of dataset, we have different types of dataset schemas.

A relational database schema is describing the names of the tables as well as the names and properties of their columns. Other information may also be described, for instance views, user access rights etc.

An XML schema specifies the names and properties of the elements, sub-elements and attributes in an XML file.

IFC files are usually encoded using the STEP format (ISO 10303:21) or XML. The IFC schema is specified using the EXPRESS data modelling language (ISO 10303:11). The IFC schema is also described as an XML schema (.xsd) in case XML encoding is preferred.

IFC Data Schema



A data schema describes the structure of a dataset. In the IFC standard, the data schema is described using the EXPRESS standard (ISO 10303-11). The IFC classes don't have any code (methods), only the data structure of each class is specified.

The IFC standard have three types of classes, namely.

- IfcObjectDefinition where different entities like IfcDoor and IfcWall are specified.
- IfcRelationship, which specifies relationships between entities, for instance that a door is mounted on a wall.
- IfcPropertyDefinition, which specify properties (attributes) associated with the entities. Example of properties are type of material, price, supplier etc for a specific door.

As also mentioned, IFC data is usually provided in STEP format, but XML encoding is also supported by the standard.

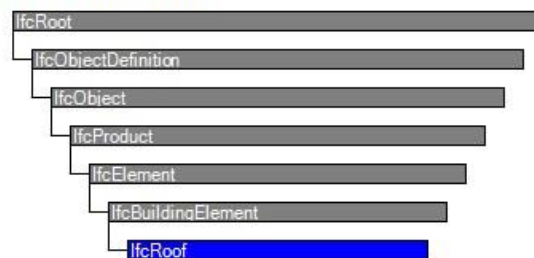
Object Inheritance

IFC as a data exchange format

Object inheritance in IFC

6.1.3.34.2 Inherited definitions from supertypes

Entity inheritance



Source: https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/

13

In order to understand the structure of IFC files, one needs to be familiar with the concept of object inheritance. This image shows the entity inheritance of the class IfcRoof.

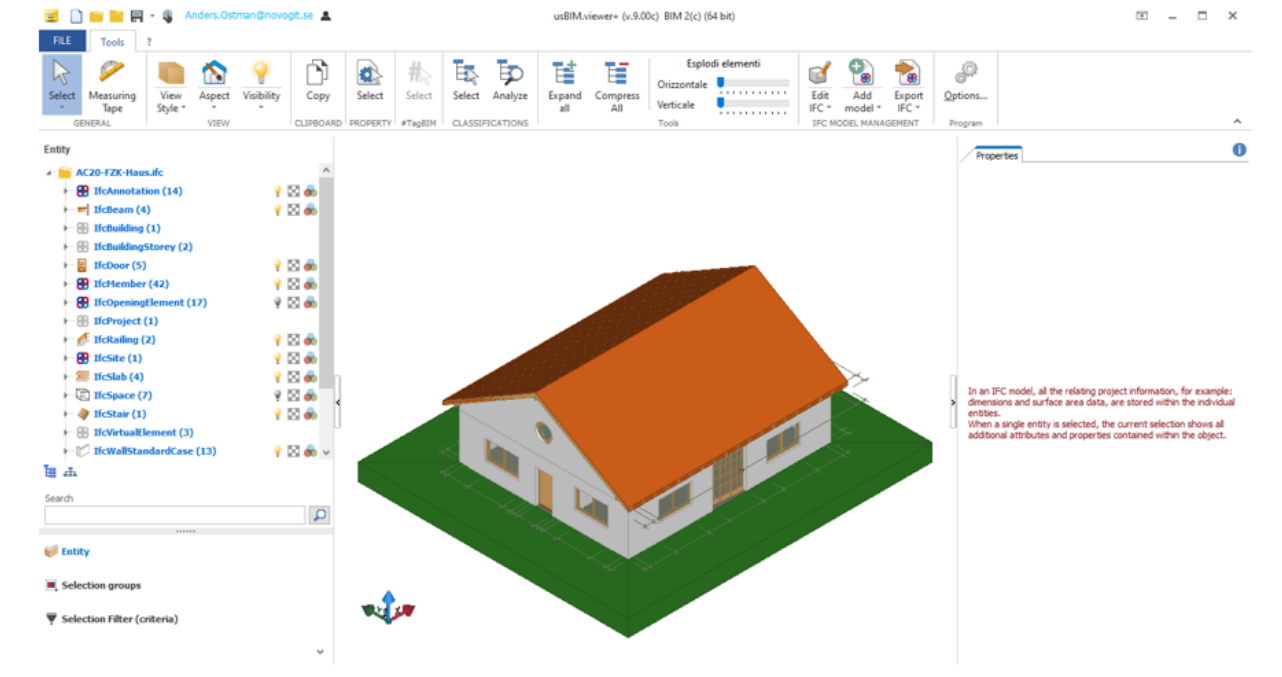
Inheritance is a central concept of object orientation. A class consists of a collection of attributes and methods that determine the state and behaviour of its instances. Using inheritance, attributes and/or methods of one class are passed on to another class. The inheriting class is called the superclass or supertypes – and the inheriting class is called subclass (<https://t2informatik.de/en/smartpedia/uml-inheritance/>).

In the IfcRoof example, the class “IfcRoof” is a subclass of “IfcBuildingElement” and it inherits all attributes related to its superclass. In addition, the class “IfcBuildingElement” is a subclass of the class “IfcElement” and consequently inherits all attributes related to this class. Since new attributes may be added at each level, general attributes are specified at the top of the tree and more specific attributes at the bottom.

The IFC standard is utilizing this type of inheritance to a large extent. The IFC classes form a giant tree, where subclasses inherit the properties (attributes) from its superclass, and where the superclass is a subclass of another superclass. This will be better illustrated later on in this lecture.



Analysing an IFC file using usBIM.viewer





We will now start looking at an IFC file in detail and analyse how it is structured. In this lecture, we will analyse the IFC file “AZ20-FZK-Haus.ifc” by using the IFC viewer “usBIM.viewer” and a text editor. The file extension “.ifc” indicates that the file is in STEP format.

When you open the IFC file in the viewer, you will see a house as indicated in this slide.


Open IFC file in Text Editor

IFC as a data exchange format

Co-funded by
the European Union

Open the IFC file in a text editor

 AC20-FZK-Haus.ifc - Anteckningar

Arkiv Redigera Format Visa Hjälp

```
ISO-10303-21;
HEADER;FILE_DESCRIPTION(('ViewDefinition [ , QuantityTakeOffAddOnView, SpaceBoundary2ndLevelAdd
metry: As boundary representation (BRep)'],'Option [IFC Site Location: At Project Origin]','Op
FILE_NAME('S:\\[IFC]\\[COMPLETE-BUILDINGS]\\FZK-MODELS\\FZK-Haus\\ArchiCAD-20\\AC20-FZK-Haus.i
FILE_SCHEMA(('IFC4')));
ENDSEC;

DATA;
#1= IFCPERSON($,'Nicht definiert',$,$,$,$,$);
#3= IFCORGANIZATION($,'Nicht definiert',$,$,$);
#7= IFCPERSONANDORGANIZATION(#1,#3,$);
#10= IFCORGANIZATION('GS','GRAPHISOFT','GRAPHISOFT',$,$);
#11= IFCAPPLICATION(#10,'20.0.0','ARCHICAD-64','IFC2x3 add-on version: 4009 GER FULL');
#12= IFCOWNERHISTORY(#7,#11,$,ADDED,$,$,$,1482339244);
#13= IFCSIUNIT(*,LENGTHUNIT,$,METRE.);
#14= IFCSIUNIT(*,AREAUNIT,$,SQUARE_METRE.);
#15= IFCSIUNIT(*,VOLUMEUNIT,$,CUBIC_METRE.);
#16= IFCSIUNIT(*,PLANEANGLEUNIT,$,RADIAN.);
#17= IFCMEASUREWITHUNIT(IFCPLANEANGLEMEASURE(0.0174532925199),#16);
#18= IFCDIMENSIONALEXONENTS(0,0,0,0,0,0);
#19= IFCCONVERSIONBASEDUNIT(#18,PLANEANGLEUNIT,'DEGREE',#17);
#21= IFCSIUNIT(*,SOLIDANGLEUNIT,$,STERADIAN.);
#22= IFCMONETARYUNIT('EUR');
#23= IFCSIUNIT(*,TIMEUNIT,$,SECOND.);
#24= IFCSIUNIT(*,MASSUNIT,$,GRAM.);
#25= IFCSIUNIT(*,THERMODYNAMICTEMPERATUREUNIT,'DEGREE CELSIUS');
```

15

We can now also open the IFC file in a text editor. It starts with the header section, which we will not deal with here. Then we have the data section, which consists of a large number of entities and each entity have the structure as explained in the following slide.

Please note that a class is a general specification of objects of a certain type. The entities (objects) are then specific instances of this type of objects, for instance a specific wall.

STEP file structure

IFC as a data exchange format

STEP file structure

- STEP (ISO 10303-21) is the default encoding of IFC files
- Main structure of a STEP file
 - Header
 - Data section
 - #InstanceNumber = EntityType(Links and properties)
 - InstanceNumber must be a positive integer and unique -> Do not merge two IFC data files
 - Unset property values are written as a "\$"-character
- Example (AC20-FZK-Haus)

```
#10= IFCORGANIZATION('GS',GRAPHISOFT',GRAPHISOFT',$,$);  
#11= IFCAPPLICATION(#10;20.0.0';ARCHICAD -64';IFC2x3 add-on version: 4009 GER FULL');
```

16

The general structure of a STEP file is as follows.

The file starts with a header, which we will not pay any attention to right now. Instead, we will focus on the data section,

Each row in the STEP file describes an entity and each entity has a unique number. This means that you cannot simply merge two ifc files, since the uniqueness of the entities will then not be guaranteed.

Anyway, the structure of an entity statement is a hashtag followed by the entity number, equal sign, entity type and a list of properties (attributes) in a parenthesis. In an IFC file, the entity types and associated properties are specified in the IFC standard. Properties that are not set are marked with a dollar-sign.

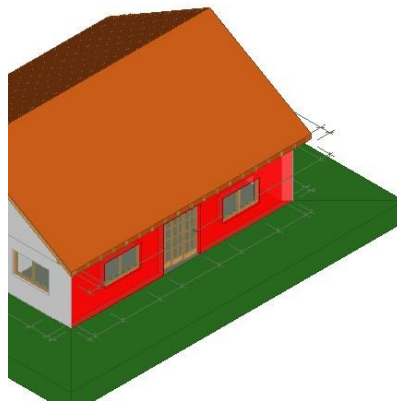
We have earlier talked about inheritance. A link to another entity (superclass) is written by using the hashtag-sign.

In the example in this slide, the class "IfcOrganization" is a superclass of "IfcApplication". The link to the entity of this superclass is the first property in the propertylist.

Finding entity number

IFC as a data exchange format

Finding entity number



Properties	
Characteristics	
ContainedInStructure	IfcBuildingStorey 'Erdgeschoss' (;
General Data	
Class	IfcWallStandardCase
GlobalId	16DNNqzfP2thtfaOfIvsKA
Name	Wand-Ext-ERDG-4
Geometrical Representation	
Axis	Curve2D
Body	SweptSolid
Bounding Box (Altezza)	2.7000 [Meters]
Bounding Box (Lungher)	12.0000 [Meters]
Bounding Box (Spessor)	0.3000 [Meters]
Box	BoundingBox
SurfaceColor	[255, 255, 255, 255]
Volume	6.6049 [Metri Cubi]
Z Max	2.7000 [Meters]
Z Min	0.0000 [Meters]
IfcObjectPlacement	
Axis	[0.0000; 0.0000; 1.0000]
Location	[12.0000; 0.0000; 0.0000] [Metr
PlacementRelTo	IfcBuildingStorey 'Erdgeschoss'
RefDirection	[-1.0000; 0.0000; 0.0000]
IfcOwnerHistory	
ChangeAction	.ADDED.
CreationDate	21/12/2016 16:54:04
OwningApplication	
Identifier	IFC2x3 add-on version: 4009 GEI
Name	ARCHICAD-64
Version	20.0.0

#27421= IFCWALLSTANDARDCASE('16DNNqzfP2thtfaOfIvsKA',#12,'Wand-Ext-ERDG-4',\$,\$,\$,#27374,#27416,'A6C3DE63 -3731-4F6A-94-7E-DE8A8295779F',\$);

17

We will now analyse an IFC wall entity. For this, we need to know the entity number of a specific object. One way to start is by clicking on a wall using the IFC viewer. The properties associated with this wall is then shown. However, the entity number is not shown in the list of properties. Instead, we have to use an unique property, for instance the object name or the global id. I select the global id, which is "16DNNqzfP2thtfaOfIvsKA".

Open then the same ifc file in a text editor and search for "16DNNqzfP2thtfaOfIvsKA". If the global ID's are correct, there is only one occurrence, which it was in our case. We then see that the entity number of this wall is #27421.

When examining entity #27421, we can firstly see that the object type is IfcWallStandardCase. We also notice some links to other ifc elements like #12, #27374 and #27416. We can also see some unspecified properties (the \$-signs) and some additional attributes.

Entity inheritance

IFC as a data exchange format

BIRGIT

Co-funded by
the European Union

Entity inheritance

Search for IfcWallStandardCase in the Index chapter

Formal Propositions

Rule	Description
HasMaterialLayerSetUsage	A valid instance of IfcWallStandardCase relies on the provi

6.1.3.48.2 Inherited definitions from supertypes

Entity inheritance

```

graph TD
    IfcRoot --> IfcObjectDefinition
    IfcObjectDefinition --> IfcObject
    IfcObject --> IfcProduct
    IfcProduct --> IfcElement
    IfcElement --> IfcBuildingElement
    IfcBuildingElement --> IfcWall
    IfcWall --> IfcWallStandardCase
    
```

Attribute inheritance

18

To understand the meaning (semantics) of the object, we must work with the technical documentation of the IFC standard. In its index chapter, we can search for the object type, in our case the IfcWallStandardCase.

In the technical documentation, we can see that an IfcWallStandardCase is a subclass of IfcWall, which is a subclass of IfcBuildingElement, which is a subclass of IfcElement, and so forth up to the IfcRoot element. This means that the objects are formed in a tree structure, also called object tree.

The object inheritance means that our wall (#27421) inherits all properties from the objects above

Inherited properties for #27421

IFC as a data exchange format

Inherited properties for #27421

```

IfcRoot(GlobalId, OwnerHistory, Name, Description)
  IfcObjectDefinition()
    IfcObject(ObjectType)
      IfcProduct(ObjectPlacement, Representation)
        IfcElement(Tag)
          IfcBuildingElement()
            IfcWall()

```

```

IfcWallStandardCase(GlobalId, OwnerHistory, Name, Description, ObjectType, ObjectPlacement, Representation, Tag,
IfcMaterialLayerSetUsage)

```

```
#12= IFCOWNERHISTORY(#7,#11,$,ADDED,$,$,$,1482339244);
```

```
#27374= IFCLOCALPLACEMENT(#477,#27373);
```

```
#27416= IFCPRODUCTDEFINITIONSHAPE($,$,($27399,#27405,#27413));
```

```
#27421= IFCWALLSTANDARDCASE('16DNNqzfP2thtfaOfIvsKA',#12,'WandExt-ERDG-4',$,$,$,#27374,#27416,'A6C3DE633731-4F6A-94-7E-DE8A8295779F',$);
```

19

Our first task now is to determine which properties are associated with the IfcWallStandardCase objects. We start by looking at the superclass on top (IfcRoot) in the technical documentation which has the properties GlobalId, OwnerHistory, Name and Description. The next class (IfcObjectDefinition) do not add any properties, but the next one (IfcObject) add the property ObjectType. And then we can continue reading the technical documentation until we find all properties associated with the wall object.

We can then decode entity #27421 and according to the inheritance of properties, the first property should be GlobalId (16DNNqzfP2thtfaOfIvsKA). The next property is OwnerHistory and it is a link to entity #12. This means that more information about the owner history is described by element #12. We also have two other links, namely #27374 (IfcLocalPlacement) and #27416 (IfcProductDefinitionShape).

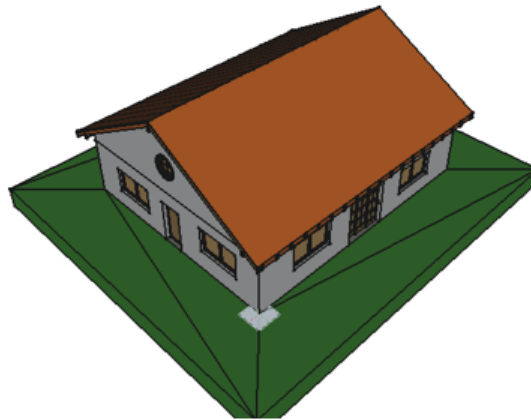
To summarise the analysis:

1. To find the meaning (semantics) of the properties, one has to study the object inheritance tree, as described in the technical documentation of the IFC standard.
2. To find the actual property values, one has to follow the links provided in the IFC file.

Analysing an IFC file using FreeCAD

IFC as a data exchange format

Analysing an IFC file using FreeCAD



20

We may now do a similar analysis using the FreeCAD tool. Open the FreeCAD software, but don't open the IFC file yet. If you do, you will import the entire building as an entity, without the possibility of analysing a single wall or door.

Change the IFC file loading settings as follows.

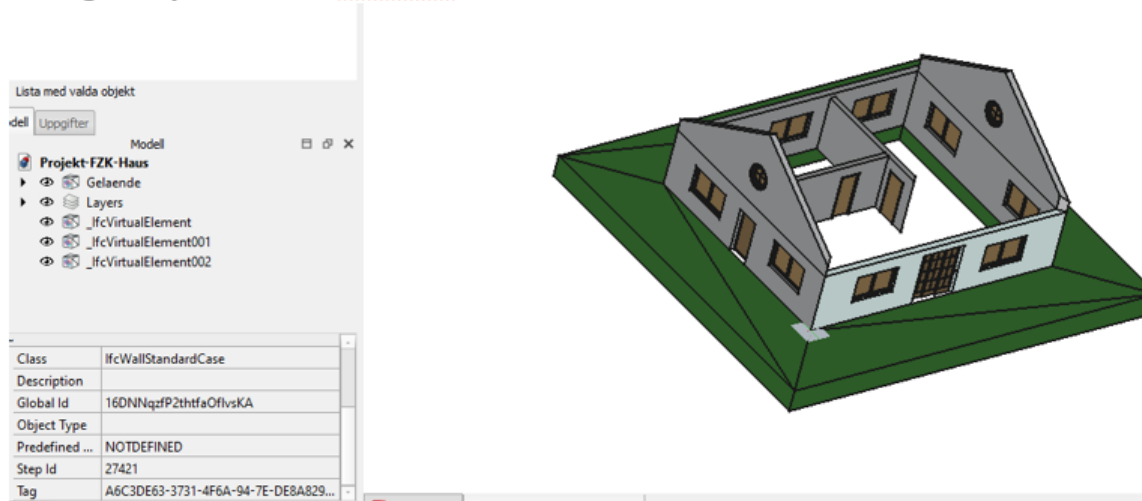
1. Enter Edit – Preferences – BIM – Native IFC
2. Set the field "Initial import" to "All individual IFC objects". In this way you will also load, not only the root object but also the leaf objects.

After this change is made, you may import the IFC file

Finding entity number using FreeCAD

IFC as a data exchange format

Finding entity number in FreeCAD



21

The procedure of finding the entity number of an element is quite easy. Just click at the object and necessary attribute will appear in the panel at the left.

In our case, we can see that the Step Id is 27421 for this specific wall. Surprised?

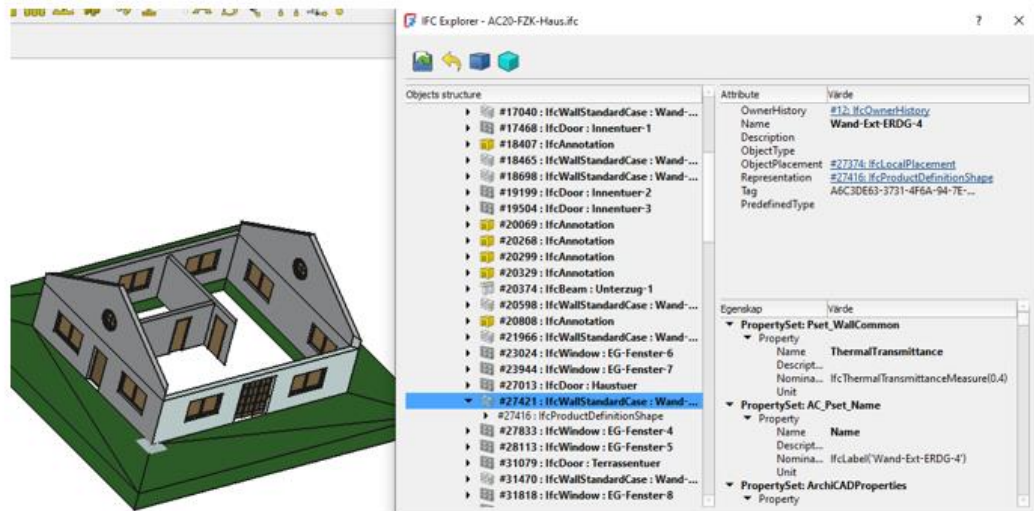


Analysing IFC tree in FreeCAD

IFC as a data exchange format



Analysing IFC file in FreeCAD



22

FreeCAD also have a tool for analysing IFC files. The IFC explorer tool can be accessed from the Utils menu. Just load the IFC file here and you may analyse the entire IFC tree in more detail.

IFC Model Views and file management

IFC as a data exchange format

IFC Model Views and File Management

- Model Views are standardised subsets of the entire IFC model
- In the future, the Model View standards may be superseded by IDS (Information Delivery Specification).
- In large projects or large organizations, a database-oriented management of data is required.

20

Different software tools, for instance Microsoft Project and Archicad, are addressing different workflows. This means that they don't need the same BIM data. Model Views are here a standardised subset of the entire IFC model, addressing a specific use case or workflow. The IFC standard includes a long list of standardised Model Views, for instance Bridge Construction View, Energy Analysis View, LandXML View and so on.

One key objective of the IFC standard is to facilitate communication between different actors in a BIM project. However, in large project or in large organisations, it is a headache to keep track of many files, even if they are properly named and versioned, which they in many cases fail to be. In such cases, a database-oriented solution is preferred, avoiding duplicate information (confusion) and allowing proper data management routines like regular backups etc.

When sharing data among actors in a BIM project, a standardised mechanism is preferred. In case vendor specific solutions are used, the sender and receiver need to have access to the same type of software, which increase the complexity of the processes. In case IFC is used for sharing data, the usage of one of the Model View standards is preferred, for simplicity reasons. In the future, the Model View standards may be superseded by IDS (Information Delivery Specification).

Summary

IFC as a data exchange format

Summary

- IFC is mainly used for transferring BIM data between actors participating in a BIM project
- Basic IFC concepts have been reviewed
 - Objects and entities, inheritance, properties
 - Basic IFC entities, such as IfcWallStandardCase, IfcOwnerHistory etc, are specified in the IFC technical guidelines
- An Ifc file in STEP format consists of
 - A header
 - A set of entities compliant with the IFC standard, with links to other entities and properties assigned.



References

ACCAsoftware, 2020. IFC file structure (part 1): the IfcObjectDefinition.

<https://biblus.accasoftware.com/en/ifc-file-structure-the-ifcobjectdefinition/>

buildingSMART, 2018. Industry Foundation Classes 4.0.2.1.

https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/

Van Berlo L, Krijnen T, Tauscher H, Liebich T, van Kranenburg A, Paasiala P, 2021. Future of the Industry Foundation Classes: towards IFC 5. https://www.buildingsmart.org/wp-content/uploads/2021/06/IFC_5.pdf.