

# Containment, grouping and relationships in IFC R2.0

DRAFT 1 – 06.02.2000

In addition to defining actual objects the IFCs provide ways to make the model more intelligent and useful. By adding relationships that define containment, grouping and relationships between specific object types the model becomes much more organized.

## The major categories are:

Containment

'Referencing containment'

Grouping

Relationships between specific object types

IfcRelSeparatesSpaces

IfcRelConnectsPathElements

IfcRelVoidsElement

IfcRelFillsElement

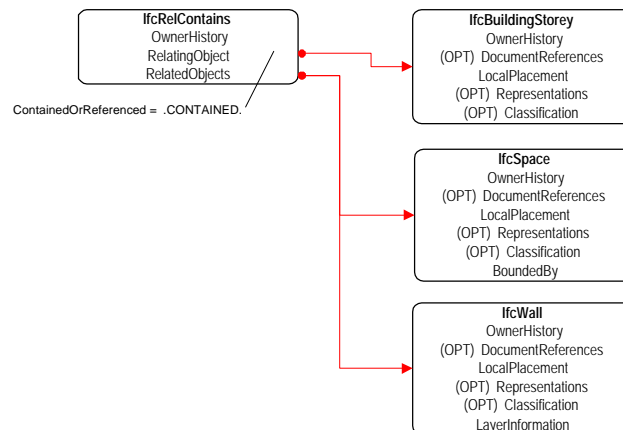
IfcRelAssemblesElements

IfcRelAssignsProperties

## **Containment**

It is in the nature of containment that one 'thing' is contained by only one container, e.g. a building storey is contained by the building, a space is contained by a building storey and furniture is contained by a space. Containment forms a hierarchical structure

In IFCs containment is dealt with the `IfcRelContains` objectified relationship.



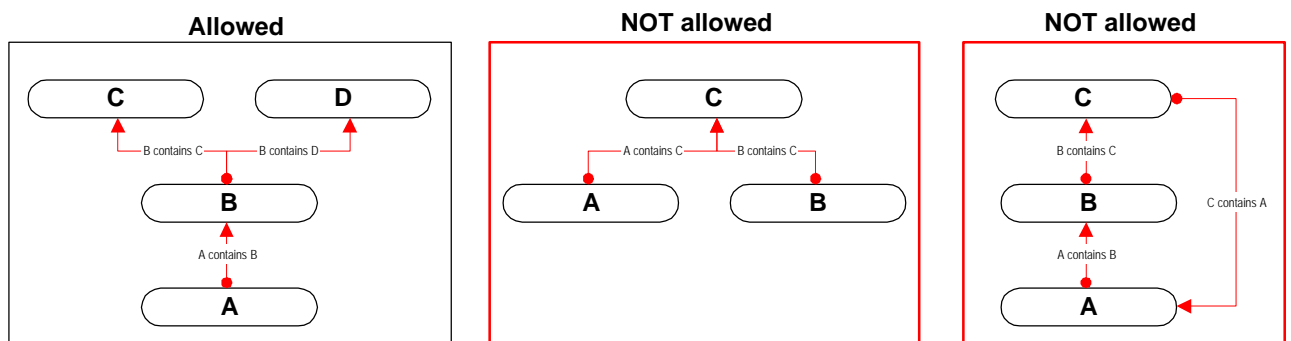
`IfcRelContains` has the attribute `RelationshipType` that gives the allowed values for containment.

- `ProjectContainer`
- `SiteContainer`
- `BuildingContainer`
- `BuildingStoreyContainer`
- `SpaceContainer`
- `ZoneContainer`

- NotDefined

According to the model any sub type of `IfcObject` can act as a container, but containment only makes sense if the objects are actually contained by the container object, e.g. it does not make much sense that an `IfcOccupant` acts as a container.

Each container object can have only one `IfcRelContains` (of type `.CONTAINED.`). It is not possible to divide the contained objects into logical groups, e.g. if a space contains furniture and electric fixtures both have to be referenced through the same `IfcRelContains` instance.

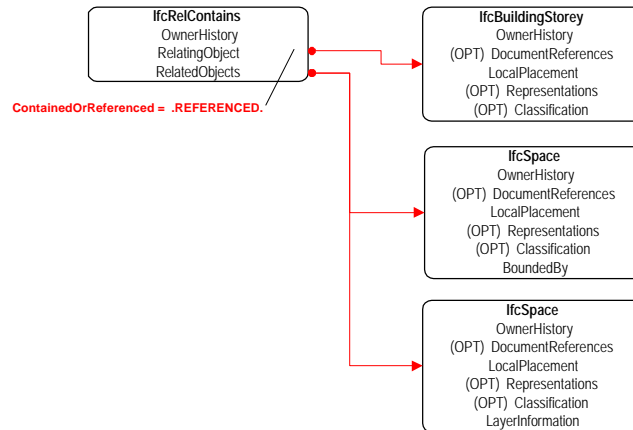


### 'Referencing containment'

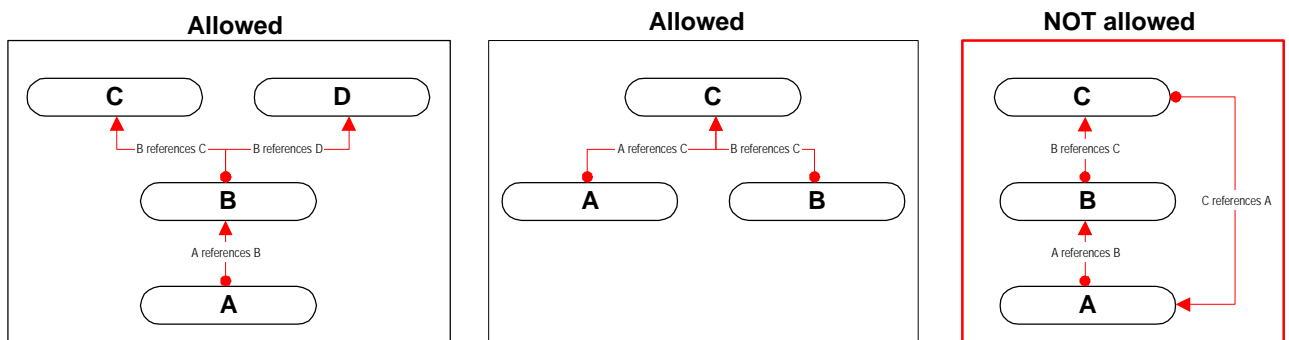
By referencing containment I mean the mechanism by which a group of objects can be related to another object, but the referencing object does not really contain the other objects. The example given by the IFC specifications is a space that spans multiple building stories. The space is only contained by the first building storey, but it is referenced by all the other building stories it spans. Another case would be the assignment of doors and windows to spaces, where the space does not really contain the doors and windows.

A group on the other hand is not related to any object in the model, e.g. a group cannot 'belong' to a space.

In IFCs referencing containment is dealt with the `IfcRelContains` objectified relationship, just as containment. The only difference is, that the attribute `ContainedOrReferenced` has the value `.REFERENCED.` whereas it had the value `.CONTAINED.` when used for containment.



The same relationship types as for containment apply also for referencing containment. Each object can have only one **IfcRelContains** (of type **.REFERENCED.**), all objects that are referenced by the object must be referenced through the same **IfcRelContains** instance. The rules for referencing containment are a bit more relaxed in that the structure does not have to be hierarchical.

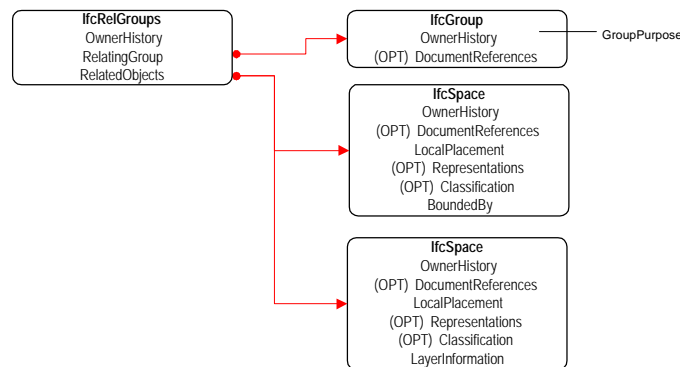


## Grouping

Grouping is a very generic mechanism for grouping together objects of any kind (must be subtype of **IfcObject**). As an idea it is closely related to e.g. the concept of layers in CAD programs, but it is not limited to this functionality.

Objects can be members in any number of groups and groups can be nested, i.e. groups can contain other groups.

Grouping is done using the `IfcRelGroups` objectified relationship and `IfcGroup` or any of its subtypes.



The identification of the group happens with the `GroupPurpose` attribute on `IfcGroup` (which is an optional attribute)

The specialized sub types of `IfcGroup` are:

- `IfcSystem`
- `IfcZone`
- `IfcSpaceProgramGroup`
- `IfcInventory`

The sub types add information or set specific constraints to the grouping, e.g. an `IfcZone` group can only contain `IfcSpace` objects or other `IfcZone` objects. Please refer to the IFC specifications for more detailed definitions of the sub types.

`IfcGroup` has two attributes that can be used to identify a group

- `GroupPurpose`, which is a string containing the 'name' of the group
- `UserDefinedType`, which is a string containing the 'type' of the group.

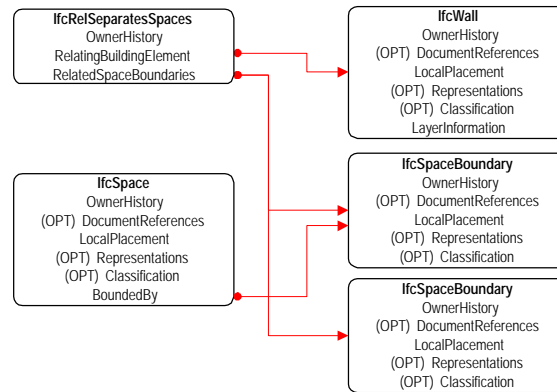
To some extent using the sub types of `IfcGroup` also provide a means to identify the group type, e.g. HVAC systems can be created using the `IfcSystem` object.

## **Relationships between specific object types**

The IFC model provides a number of objectified relationships that are used to relate specific object to each other, there is e.g. `IfcRelSeparatesSpaces` to connect walls and other building elements to space boundaries. I will go through the relationships that are needed by the current scope of implementations. If you want more information just take a look at all the objects that start with `IfcRel`, there are plenty of those.

### **IfcRelSeparatesSpaces**

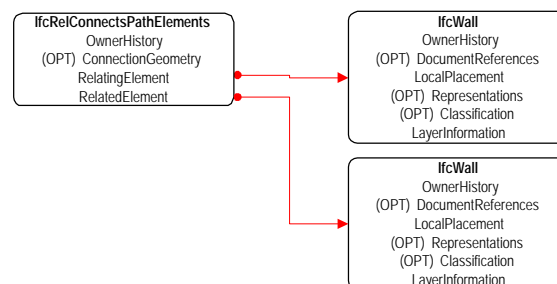
The `IfcRelSeparatesSpaces` objectified relationship is used to relate space boundaries to the physical elements that bound a space.



The relationship from `IfcSpace` to `IfcSpaceBoundary` is one of the few relationships of this type in the model that are not objectified.

### IfcRelConnectsPathElements

The `IfcRelConnectsPathElements` objectified relationship creates a logical connection between two objects (1 to 1 relationship). There is an optional `ConnectionGeometry` attribute, but the connection geometry in this case is a control class that defines how the connection is done.



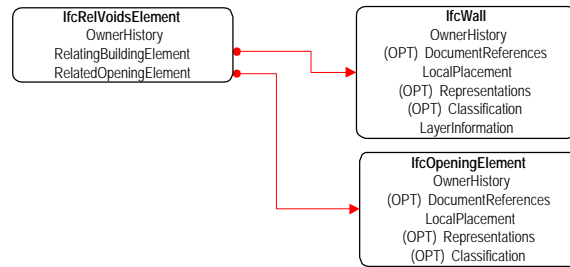
`IfcRelConnectsPathElements` defines the location or type of the connection for both the related and relating element to be:

- `Start`
- `AtEnd`
- `AtPath`
- `NotDefined`

`IfcRelConnectsPathElements` can be used to connect e.g walls or HVAC equipment.

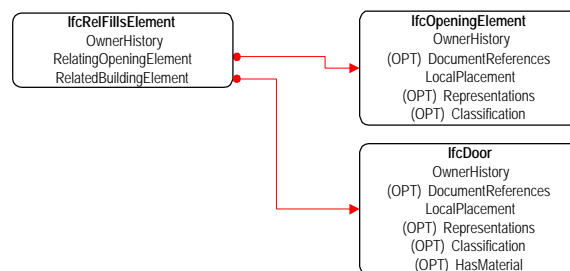
### IfcRelVoidsElement

The `IfcRelVoidsElement` objectified relationship is used to assign voids (`IfcOpeningElement`) to building objects such as walls and slabs.



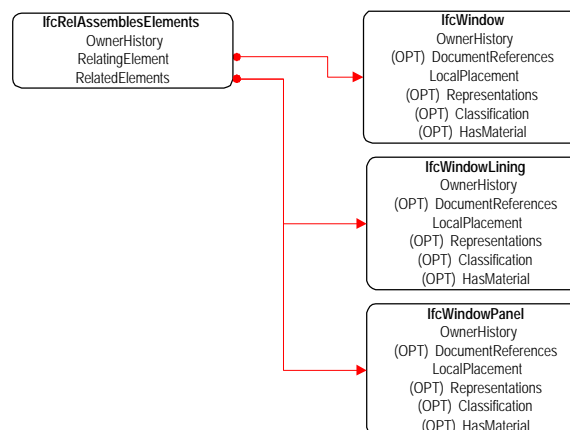
## IfcRelFillsElement

The `IfcRelFillsElement` objectified relationship is used to assign filling elements, such as door and windows, to `IfcOpeningElement` objects.



## IfcRelAssemblesElements

The `IfcRelAssemblesElement` objectified relationship is used to create assemblies of object, e.g. a window can be an assembly of `IfcWindowPanel` and `IfcWindowLining` objects.



## IfcRelAssignsProperties

The `IfcRelAssignsProperties` objectified relationship is becoming one of the most important relationships in the IFC model. This is the relationship that assigns `PropertySets` and property objects to other objects. An example of a `PropertySet` would be `Pset_DoorCommon` and an example of a property class would be `IfcManufactureInformation`.

If you ever wonder why e.g. `IfcDoor` does not have any attributes that really make a door out of the object this is because all the details can be found through the `IfcRelAssignsProperties` relationship objects attached to the `IfcDoor` instance. There should be a separate paper dealing just with this concept.

