

# Industry Foundation Classes (IFC)

## Om IFC-standard

**IFC** (*Industry Foundation Classes*) är en öppen och neutral standard för informationsdelning inom bygg- och anläggningsindustrin. IFC används vid utbyte av information mellan olika programvaror och system, men framför allt som en gemensam informationsmodell för att representera olika aspekter av ett byggprojekt. Modellen definierar strukturen och semantiken för data som kan vara relevanta för byggobjekt, såsom geometrier, egenskaper, relationer och hierarkier. Alla dessa definieras som entiteter – ”företeelser” – oavsett om de har en fysisk representation eller inte.

IFC är utvecklad av intresseorganisationen buildingSMART, och antagen som en internationell standard i form av ISO 16739-1:2024.

IFC kan implementeras i olika filformat för att underlätta informationsutbyte mellan olika programvaror och system. Det vanligaste filformatet som används för att lagra och överföra IFC-data är filformatet som har filändelsen "ifc". Detta är en textfil som följer en specifik syntax och struktur enligt IFC-standardens specifikationer.

IFC är alltså både en standard för informationsmodellering och ett filformat. Informationsmodellen definierar strukturen för data, medan filformatet (IFC-filen) används för att lagra och överföra data som följer den definierade informationsmodellen.

Genom att använda IFC kan programvaror och system kommunicera och dela projekterad information på ett strukturerat sätt. Informationen kan omfatta geometri, egenskaper, relationer och andra relevanta data om byggobjekt såsom väggar, golv, tak, dörrar, fönster, vägar, broar, installationer, material och komponenter. IFC-filen fungerar som en container för att lagra denna information, och den kan sedan användas av olika programvaror för att visualisera, analysera, simulera och hantera byggprojekt.

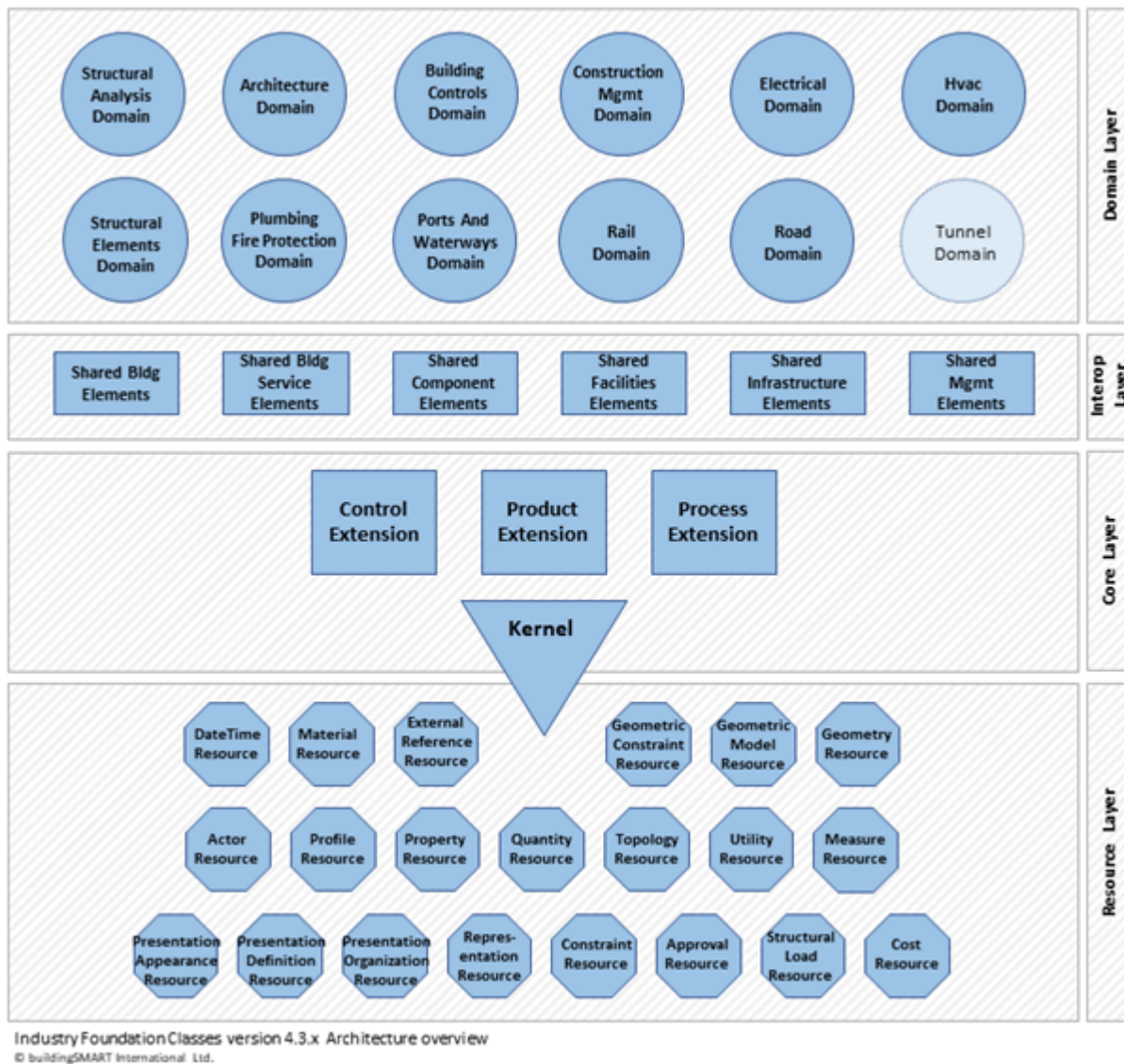
Genom att använda IFC-standardens gemensamma informationsmodell kan aktörer inom bygg- och anläggningsbranschen samarbeta effektivt och dela data oavsett vilken programvara eller system de använder. Det hjälper till att förbättra kommunikationen, minska dubbelarbete och öka precisionen i informationsutbytet mellan olika faser av ett projekt, från design och konstruktion till drift och underhåll.

## Datamodell

Strukturen – schemaarkitekturen – för IFC:s datamodell består av fyra konceptuella lager, där ett lager är en nivå i en hierarkisk struktur. Varje individuellt schema är kopplat till ett specifikt konceptuellt lager. Lagren benämns

- Domain Layer
- Interoperability Layer
- Core Layer
- Resource Layer.

Varje lager är beroende av föregående lager med start utifrån det nedersta lagret. Domain Layer är beroende av Interoperability Layer, som i sin tur är beroende av Core Layer och till sist Resource Layer. Figuren nedan visar de olika lagren och dess innehåll för IFC version 4.3.



Figur 1: De fyra konceptuella lagren för IFCs datamodell.

**Resource Layer** är det lägsta lagret. Det innehåller scheman som beskriver de mest grundläggande delarna av IFC, exempelvis:

- datum och tid (DateTime Resource)
- material (Material Resource)
- roller (Actor Resource)
- egenskaper (Property Resource)
- mängdenheter (Quantity Resource).

Det lägsta lagret inkluderar alla individuella scheman som innehåller resursdefinitioner. Dessa definitioner inkluderar inte en globalt unik identifierare, och ska inte användas oberoende av en definition som deklaras i ett högre lager.

**Core Layer** är det näst lägsta lagret. Det innehåller kärnschemat med de mest allmänna entitetsdefinitionerna. Alla enheter definierade i detta lager eller över har ett globalt unikt ID och eventuellt ägare och historikinformation.

**Interoperability Layer** är det näst högsta lagret. Det innehåller scheman med enhetsdefinitioner, som är specifika för en allmän produkt-, process- eller resursspecialisering som används inom flera discipliner.

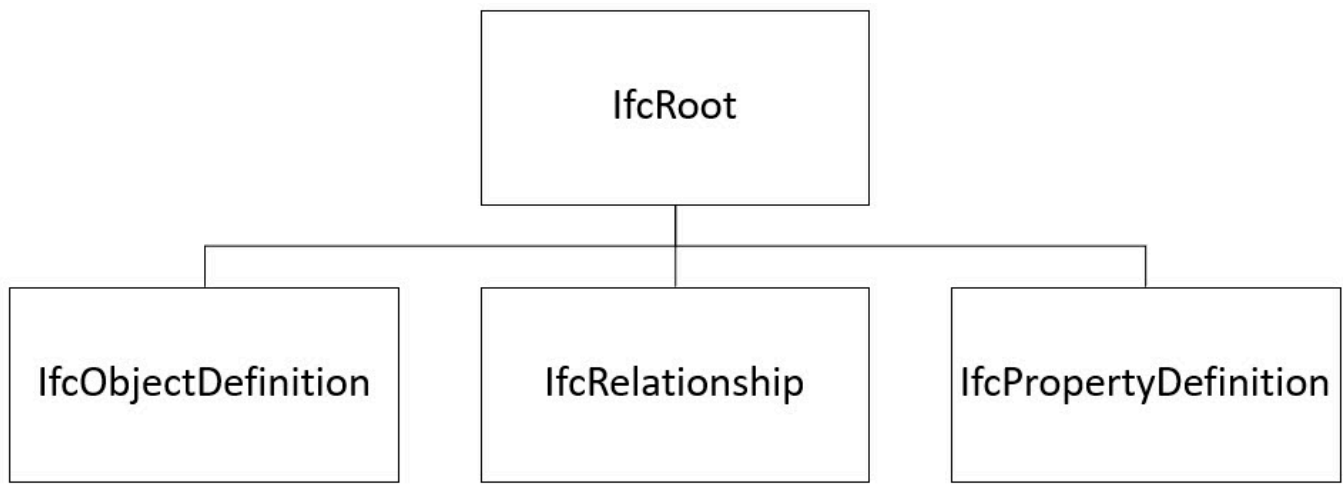
Dessa definitioner används vanligtvis för utbyte mellan domäner och vid utbyte/delning av konstruktionsinformation.

**Domain Layer** är det högsta lagret. Det innehåller enhetsdefinitioner som är specialiseringar av produkter, processer eller resurser som är specifika för en viss disciplin. Dessa definitioner används vanligtvis för utbyte inom domänen och delning av information.

## Struktur

När data levereras med IFC struktureras informationen enligt figuren nedan.

Alla entiteter som placeras inom denna struktur ärver sin överordnade entitets attribut.



Figur 2: Grundläggande struktur för IFC-objekt.

### Nivå 1 - IfcRoot

*IfcRoot* är basen och alla entiteter som definieras under *IfcRoot* tilldelas det obligatoriska attributet *GlobalId* (i form av en globalt unik identitet, GUID) samt följande valfria attribut:

- OwnerHistory (ägare och dess historik)
- Name (entitetens namn)
- Description (entitetens beskrivning).

| #                       | Attribute    | Type                     | Description                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------|--------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IfcRoot (4)             |              |                          |                                                                                                                                                                                                                                                                                                                                                                                        |
| 1                       | GlobalId     | IfcGloballyUniqueId      | Assignment of a globally unique identifier within the entire software world.                                                                                                                                                                                                                                                                                                           |
| 2                       | OwnerHistory | OPTIONAL IfcOwnerHistory | Assignment of the information about the current ownership of that object, including owning actor, application, local identification and information captured about the recent changes of the object.<br><br><div>NOTE Only the last modification is stored - either as addition, deletion or modification.</div> <div>IFC4-CHANGE The attribute has been changed to be OPTIONAL.</div> |
| 3                       | Name         | OPTIONAL IfcLabel        | Optional name for use by the participating software systems or users. For some subtypes of IfcRoot the insertion of the Name attribute may be required. This would be enforced by a where rule.                                                                                                                                                                                        |
| 4                       | Description  | OPTIONAL IfcText         | Optional description, provided for exchanging informative comments.                                                                                                                                                                                                                                                                                                                    |
| IfcObjectDefinition (7) |              |                          |                                                                                                                                                                                                                                                                                                                                                                                        |

Figur 3: De fyra attributen för IfcRoot.

## Nivå 2 – IfcObjectDefinition, IfcRelationship och IfcPropertyDefinition

*IfcObjectDefinition*, *IfcRelationship* och *IfcPropertyDefinition* är de tre delarna där entiteter kan beskrivas: vad det är för typ av objekt, vad det har för relation (relationship) till andra objekt och hur dess egenskaper (properties) kan hanteras.

Följande typer av entiteter placeras under *IfcObjectDefinition*:

- fysiskt påtagliga objekt (exempelvis väggar, balkar, och rör)
- fysiskt existerande objekt (exempelvis utrymmen)
- konceptuella objekt (exempelvis stomlinjer och virtuella gränser)
- processer (exempelvis arbetsuppgifter)
- kontroller (exempelvis kostnadsuppföljning)
- resurser (exempelvis personal, material, och utrustning)
- aktörer (exempelvis personer och företag).

Inom *IfcRelationship* finns följande typer av relationer som används för att hantera kopplingar mellan objekt:

- IfcRelAssigns
- IfcRelAssociates
- IfcRelConnects
- IfcRelDeclares
- IfcRelDecomposes
- IfcRelDefines

**IfcRelAssigns:** Används för att tilldela resurser, uppgifter eller ansvar till objekt.

**IfcRelAssociates:** Kopplar objekt till externa resurser som dokument, klassifikationer eller materialdefinitioner.

**IfcRelConnects:** Hanterar fysiska eller logiska kopplingar mellan objekt, som t.ex. anslutningar mellan byggnadselement.

**IfcRelDeclares:** Används för att deklarera att ett objekt tillhör en viss kontext eller grupp.

**IfcRelDecomposes:** Hanterar hierarkiska relationer där ett objekt delas upp i mindre delar.

**IfcRelDefines:** Används för att definiera egenskaper eller typer för objekt.

Dessa relationer hjälper till att strukturera och organisera informationen på ett effektivt sätt.

Entiteter under *IfcPropertyDefinition* definierar hur egenskaper läses, ändras, och tilldelas till entiteter inom *IfcObjectDefinition*.

Inom IFC 4.3 finns det flera subtyper av *IfcPropertyDefinition* som används för att definiera och hantera egenskaper hos objekt. De huvudsakliga typerna är

- *IfcPropertySetDefinition*
- *IfcPropertyTemplateDefinition*

**IfcPropertySetDefinition:** Detta inkluderar alla typer av egenskapsuppsättningar som kan associeras med objekt.

**IfcPropertyTemplateDefinition:** Detta inkluderar mallar som definierar egenskaper och deras struktur.

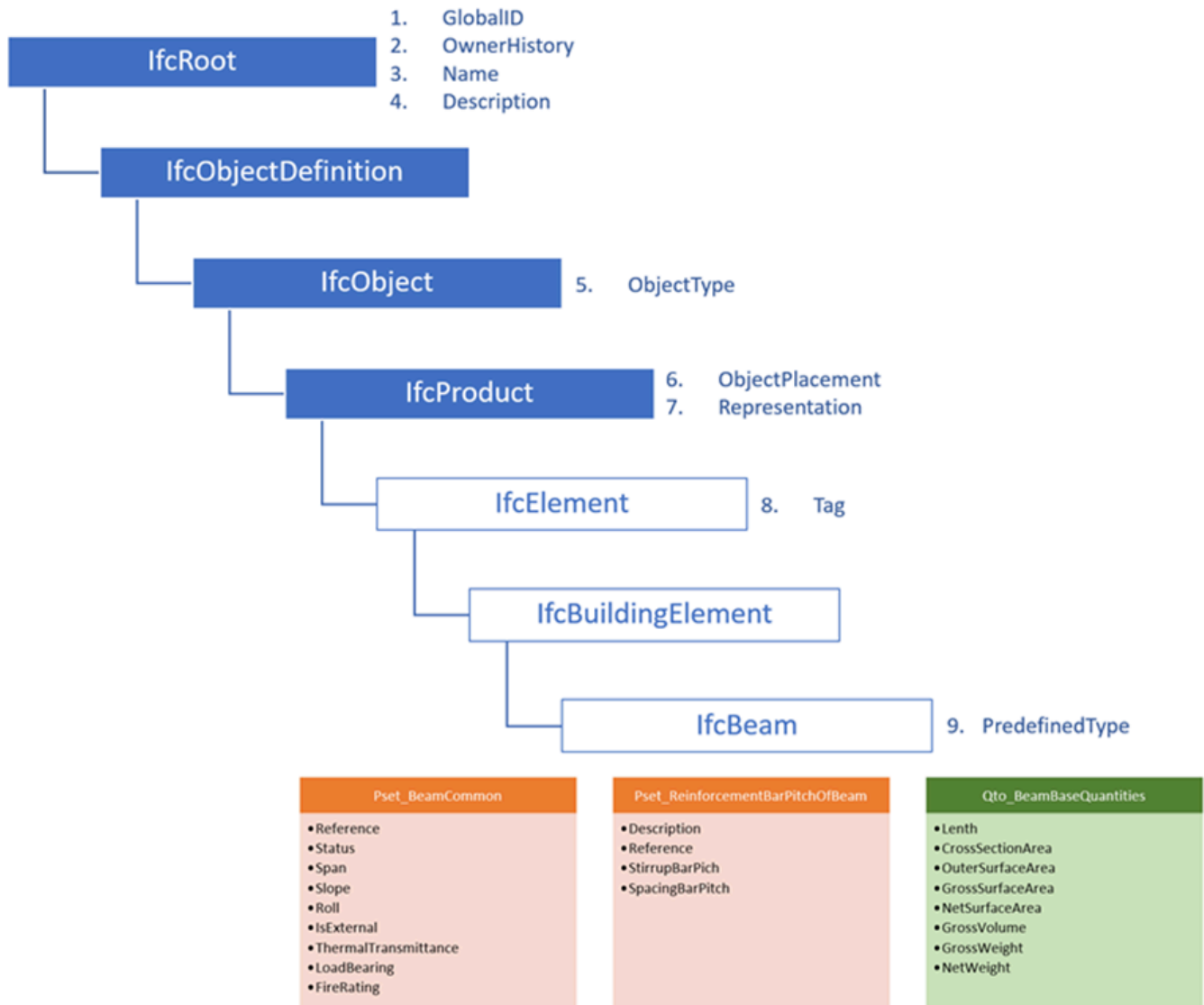
Dessa två subtyper hjälper till att strukturera och standardisera informationen.

## Attribut och egenskaper

Entiteter kan beskrivas ytterligare med hjälp av egenskaper (*IfcProperty*). Dessa egenskaper redovisas i **egenskapsuppsättningar** (*IfcPropertySet*) och **mängduppsättningar** (*IfcQuantitySet*). Attribut ligger alltid direkt på respektive entitet, medan egenskaper alltid ligger i en egenskaps- eller mängduppsättning.

Figuren nedan visar ett exempel på hur attribut (numrerade från 1 till 8) ärvs i IFC-strukturen samt hur egenskaper är samlade i egenskaps- (orange) och mängduppsättningar (grön) för en balk (*IfcBeam*). I IFC benämns alltid egenskaps- och mängduppsättningar med ett specifikt prefix.

Egenskapsuppsättningar börjar alltid med Pset\_... och mängduppsättningar börjar alltid med Qto\_ ...



Figur 4: Exempel på attribut, parametrar, samt parameter- och mängdgrupper.

Attributen kan både vara tvingande och valbara. Av de nio attribut i exemplet ovan är det enbart *GlobalID* som är tvingande. Mappningen av information från originalprogramvaran till dessa attribut styrs av programvaruleverantörens exportfunktion, men kan i vissa fall ändras av användaren. Det kan vara svårt att veta hur denna mappning är definierad, vilket gör det svårt att krävställa information på dessa attribut.

## Egenskaps- och mängduppsättningar

Egenskaper ingår alltid i en eller flera egenskaps- eller mängduppsättningar, och en egenskaps- eller mängduppsättning kan i sin tur beskriva en eller flera IFC-entiteter. Relationen mellan dessa styrs via *IfcPropertyDefinition* som nämnts ovan. För balken i exemplet ovan finns en koppling till 25 egenskapsuppsättningar och två mängduppsättningar, se nedan.

|                                                                                                                                     |                                                                                                                   |                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>Pset_BeamCommon</b><br>Reference<br>Status<br>Scan                                                                               | <b>Pset_ConcreteElementGeneral</b><br>AssemblyPlace<br>CastingMethod<br>StructuralClass                           | <b>Pset_Condition</b><br>AssessmentDate<br>AssessmentCondition<br>AssessmentDescription                                         |
| <b>Pset_ConstructionAdministration</b><br>ProcurementMethod<br>SpecificationSectionNumber<br>SubmissionIdentifier                   | <b>Pset_ConstructionOccurrence</b><br>InstallationDate<br>ModelNumber<br>TagNumber                                | <b>Pset_ElementKinematics</b><br>CyclicPath<br>CyclicRange<br>LinearPath                                                        |
| <b>Pset_EnvironmentalCondition</b><br>ReferenceAirRelativeHumidity<br>ReferenceEnvironmentTemperature<br>MaximumAtmosphericPressure | <b>Pset_EnvironmentalImpactIndicators</b><br>Reference<br>FunctionalUnitReference<br>IndicatorsUnit               | <b>Pset_EnvironmentalImpactValues</b><br>TotalPrimaryEnergyConsumption<br>WaterConsumption<br>HazardousWaste                    |
| <b>Pset_InstallationOccurrence</b><br>InstallationDate<br>AcceptanceDate<br>PutIntoOperationDate                                    | <b>Pset_MaintenanceStrategy</b><br>AssetCriticality<br>AssetFrailty<br>AssetPriority                              | <b>Pset_MaintenanceTriggerCondition</b><br>ConditionTargetPerformance<br>ConditionMaintenanceLevel<br>ConditionReplacementLevel |
| <b>Pset_MaintenanceTriggerDuration</b><br>DurationTargetPerformance<br>DurationMaintenanceLevel<br>DurationReplacementLevel         | <b>Pset_MaintenanceTriggerPerformance</b><br>TargetPerformance<br>PerformanceMaintenanceLevel<br>ReplacementLevel | <b>Pset_ManufacturerOccurrence</b><br>AcquisitionDate<br>BarCode<br>SerialNumber                                                |
| <b>Pset_ManufacturerTypeInfoInformation</b><br>GlobalTradeItemNumber<br>ArticleNumber<br>ModelReference                             | <b>Pset_PrecastConcreteElementFabrication</b><br>TypeDesignation<br>ProductionLotId<br>SerialNumber               | <b>Pset_PrecastConcreteElementGeneral</b><br>TypeDesignation<br>CornerChamfer<br>ManufacturingToleranceClass                    |
| <b>Pset_ReinforcementBarPitchOfBeam</b><br>Description<br>Reference<br>StripedBarPitch                                              | <b>Pset_RepairOccurrence</b><br>RepairContent<br>RepairDate<br>MeanTimeToRepair                                   | <b>Pset_Risk</b><br>RiskName<br>RiskType<br>NatureOfRisk                                                                        |
| <b>Pset_ServiceLife</b><br>ServiceLifeDuration<br>MeanTimeBetweenFailure                                                            | <b>Pset_Tolerance</b><br>ToleranceDescription<br>ToleranceBasis<br>OverallTolerance                               | <b>Pset_Uncertainty</b><br>UncertaintyBasis<br>UncertaintyDescription<br>HorizontalUncertainty                                  |
| <b>Pset_Warranty</b><br>WarrantyIdentifier<br>WarrantyStartDate<br>IsExtendedWarranty                                               | <b>Qto_BeamBaseQuantities</b><br>Length<br>CrossSectionArea<br>OuterSurfaceArea                                   | <b>Qto_BodyGeometryValidation</b><br>GrossSurfaceArea<br>NetSurfaceArea<br>GrossVolume                                          |

Figur 5: Egenskaps- och mängduppsättningar som är kopplade till entiteten IfcBeam.

Precis som för attribut styrs mappningen av information från originalprogramvaran till dessa egenskaps- och mängduppsättningar av programvaruleverantörens exportfunktion. I vissa fall kan de ändras av användaren, men det kan vara svårt att veta hur mappningen är definierad vilket gör det svårt att krävställa information på egenskaperna i dessa egenskaps- och mängduppsättningar.

För varje egenskaps- och mängduppsättning finns en beskrivning av ingående egenskaper, se exempel för *Pset\_BeamCommon* i figuren nedan. Exemplet visar enbart tre av de nio ingående egenskaperna.

| Name                 | Property Type              | Data Type                      | Description                                                                                                                                                                                                                                                                                                                                        |                                                                                       |
|----------------------|----------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Status               | IfcPropertyEnumeratedValue | PEnum_ElementStatus            | Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exists only temporary (like a temporary support structure). |  |
| IsExternal           | IfcPropertySingleValue     | IfcBoolean                     | Indication whether the element is designed for use in the exterior (TRUE) or not (FALSE). If (TRUE) it is an external element and faces the outside of the building.                                                                                                                                                                               |  |
| ThermalTransmittance | IfcPropertySingleValue     | IfcThermalTransmittanceMeasure | Thermal transmittance coefficient (U-Value) of an element, within the direction of the thermal flow (including all materials).                                                                                                                                                                                                                     |  |

Figur 6: Exempel på egenskaper i egenskapsgruppen Pset\_BeamCommon.

Varje egenskaps beskrivs med en benämning (*Name*), egenskapstyp (*Property Type*), datatyp (*Data Type*), och en beskrivning (*Description*).

Det går även att skapa användardefinierade egenskaps- och mängduppsättningar som inte ingår i IFC-standard, exempelvis med egenskaper enligt CoClass ([Gå till Svensk Byggtjänst](#)) eller BIP ([Gå till BIP](#)), eller med företags- eller projektspecifika egenskaper.

I dessa fall ska benämningarna av egenskaps- och mängduppsättningen inte föregås av prefixet *Pset* eller *Qto*. Däremot är det möjligt att skapa egenskaps- och mängduppsättningar med dessa prefix om benämning och ingående egenskaper följer IFC-standard. Detta kan vara aktuellt för att kringgå programvaruleverantörernas fördefinierade mappning av originalprogramvarans information mot IFC-standardens egenskaper.

*Vid utbyte av information via IFC är det viktigt att säkerställa att den kravställda informationen kan levereras från de tilltänkta programvarorna eller att använda programvaror som kan leverera information enligt planerad kravställning.*

## TÄNK PÅ

Vid utbyte av information via IFC är det viktigt att säkerställa att den kravställda informationen kan levereras från de tilltänkta programvarorna eller att använda programvaror som kan leverera information enligt planerad kravställning.

## Historia och versioner

Den första versionen av IFC (IFC1.0) publicerades officiellt i januari 1997. Fram till 2023 har sedan följande sex huvudsakliga uppdateringar publicerats:

- IFC1.5.1 (september 1998)
- IFC2.0 (maj 1999)
- IFC2x (oktober 2000)
- IFC2x2 (maj 2003)
- IFC2x3 (februari 2006)
- IFC4 (februari 2013)

Den första versionen av IFC användes enbart som en prototyp fram till att **IFC1.5.1** publicerades och implementerades kommersiellt. Den versionen omfattar i huvudsak enbart byggdelar inom arkitektur.

**IFC2.0** innehåller stöd för bland annat entiteter för installation, konstnads kalkylering, och produktionsplanering.

**IFC2x** var den första versionen med en bredare implementation och den första versionen med ett certifieringsprogram. Denna version lämnades in till International Organization for Standardization (ISO) för godkännande som en så kallad Publicly Available Specification (PAS) och publicerades 2005 som ISO/PAS 16739:2005.

**IFC2x2** innehåller ännu bättre stöd för entiteter för installation, byggkonstruktion, och fastighetsförvaltning. Versionen innehåller även stöd för tvådimensionellt innehåll, så som linjer, text, och symboler, samt stöd för kulör, mönster, och ytskikt.

**IFC2x3** innehåller till stor del enbart kvalitetsförbättringar från tidigare version. Detta är den version som fram till 2023 har implementerats i flest programvaror.



**IFC4** var den första versionen av IFC som godkändes som en fullvärdig ISO-standard (ISO 16739:2013), det vill säga inte enbart som en PAS. Denna version har bland annat utvecklats med följande:

- Stöd för 4D, 5D, tillverkare, produktbibliotek.
- BIM till GIS-kompatibilitet.
- Länkning till buildingSMART data dictionary (bSDD)
- Stöd för infrastruktur andra delar av den byggda miljön.
- Flerspråkig översättning av schemat.

## Gällande versioner

Ingen av ovan nämnda versioner är längre gällande enligt buildingSMART utan har ersatts av följande officiella versioner:

- IFC2x3 TC1
- IFC4 ADD2 TC1

Sedan juli 2007 är alla tidigare IFC2-versioner ersatta av **IFC2x3 TC1**. Syftet med denna version var att rätta till ett antal mindre tekniska problem och generellt förbättra dokumentationen.

**IFC4 ADD2 TC1** innehåller också en del tekniska förbättringar och förbättrad dokumentation gentemot sina föregångare IFC4, IFC 4 ADD1, och IFC4 ADD2.

Under 2023 publicerades **IFC4.3** vilken även blev godkänd som ISO-standard under 2024. I version IFC4.3 har mycket inom infrastruktur utvecklats. Den har bland annat stöd för följande:

- entiteter för spår, vägar, tunnlar, hamnar, vattenvägar, och broar
- beskrivning av långsträckta byggobjekt
- sektioner
- rumslig nedbrytning i form av anläggningar (IfcFacility) och anläggningsdelar (IfcFacilityPart).

Denna version är alltså av stor betydelse för storskalig implementering av IFC inom infrastrukturuområdet.

För en översikt över IFC-versioner, se [buildingSMART IFC Specification Database](#).

För release notes, se [buildingSMART IFC Release Notes](#).

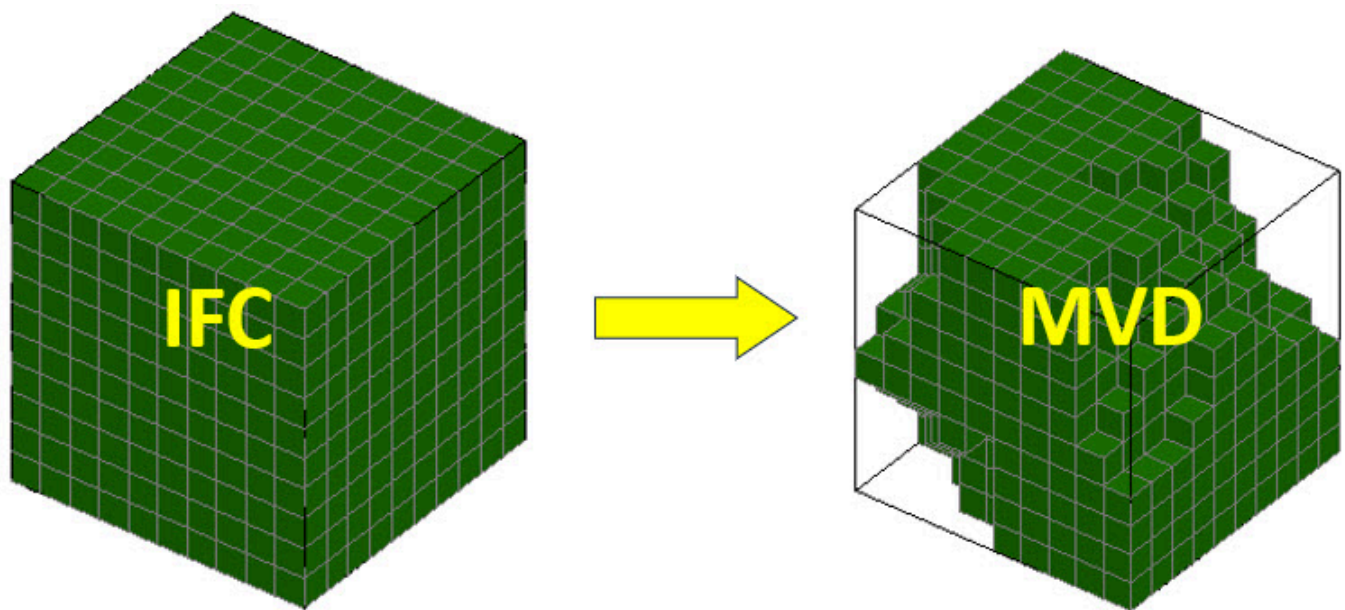
## TÄNK PÅ

Vid utbyte av information via IFC är det viktigt att säkerställa att rätt version används, det vill säga den aktör som beställer en informationsleverans via IFC måste alltid ange gällande version.

## Model View Definitions (MVD) och Exchange Requirements

En **Model View Definition (MVD)** är en specifikation för en delmängd av det övergripande IFC-schemat: en fördefinierad ”vy”. MVD tillämpas i programvaror vid IFC-exporter. Den används för ett specifikt datautbyte som är relevant för ett visst arbetsflöde, i en viss kontext eller för ett specifikt ändamål.

En MVD beskriver alltså enbart en del av innehållet i en specifik version av IFC vilket illustreras i figuren nedan. En mer detaljerad beskrivning av MVD finns i avsnittet ”Model View Definition (MVD)”.



Figur 7: Schematisk bild över skillnaden mellan IFC (vänster) och MVD (höger).  
[Källa: Mark Baldwin, *The BIM Manager*, 2019]

## Implementering och certifiering av programvaror

Programvaruleverantörer har möjlighet att certifiera sina programvaror enligt buildingSMART:s certifieringsprogram och därmed säkerställa kvaliteten på den information som utbyts via IFC.

Programvaror certifieras för en viss version av IFC och tillhörande MVD. År 2023 var det möjligt att certifiera följande versioner av IFC och MVD:

- IFC2x3 Coordination View V2.0 [CV2.0]
- IFC4 Reference View [RV1.2]

Certifieringsprocessen tar ofta lång tid och programvaruleverantörer implementerar därför ofta sina funktioner innan de blivit godkända och certifierade. För att se aktuell implementation av IFC och MVD i programvaror, se [buildingSMART Software Implementations](#).

### TÄNK PÅ

Vid planering av informationsutbyte via IFC är det viktigt att känna till vilka programvaror som har implementerat vilka versioner av IFC och MVD, samt om de är certifierade eller ej. Vid krav på certifierade programvaror minskar möjligheten till val av programvara avsevärt.

## Format för utväxling

Informationsleveranser via IFC kan göras med olika format. De vanligaste är än så länge filformatet **ifc** eller **ifcxml**, men det förekommer även att leveranser görs i exempelvis **json**. Syftet kan vara att filen behöver kunna läsas av olika typer av programvaror. CAD-program, som är grafikbaserade, använder oftast IFC medan webb-applikationer, som huvudsakligen är textbaserade, kanske använder json.

En relativt ny möjlighet är att använda **ifcowl**, som är det språk som används för att publicera länkade data med så kallad semantisk webbt teknik.

För en översikt över möjliga format, se [buildingSMART IFC Formats](#).

*Beställaren av en informationsleverans via IFC måste alltid ange vilket överföringsformat som ska användas.*

---

## Metadata

Namespace: swe-nrb

Paket: openbim

Version: 1.0.0-rc.3

Publiceringsdatum: 2025-08-18

Sökväg: 1\_introduktion\_openbim/ifc.partial.html

Genererad:



QR koden innehåller en länk tillbaka till underlagsfilen