National BIM Standard - United States®
an initiative of the National Institute of Building Sciences buildingSMARTalliance®

National BIM Standard - United States® Version 3

4 Information Exchange Standards

4.8 Heating, Ventilation and Air Conditioning information exchange (HVACie) – Edition 2013

CONTENTS

4.8.1 Scope

4.8.1.1 Business Case Description

4.8.1.1.1 Life-Cycle Phase List

Programming and concept HVAC system design
Schematic HVAC system design
Coordinated HVAC system design

4.8.1.1.2 Business Case Description

The US Army Engineer Research and Development Center, Construction Engineering Research Laboratory (ERDC-CERL) has developed a core life- cycle building information model based on three example Army buildings: an Officer Duplex Apartment, a Headquarters Office, and a Clinic. These models were developed inconsistently over time by different modelers, and they reflect different levels of detail and quality of content across disci- plines. One cause for these differences is that an ontology describing the requirements for life-cycle modeling has not been identified for the heat- ing, ventilating, and air-conditioning (HVAC) domain. Current efforts at HVAC modeling typically focus only on those hard physical collisions be- tween ductwork and structural or architectural building elements. One of the difficulties of modeling the HVAC system is the complexity of the sys- tem itself.

Life-cycle information exchanges have previously been identified in the structural steel domain—an analysis model, a design model, and a detailed model. The analysis model reflects the needs of structural engineers to evaluate the requirements of the building and size the system to meet the facility's requirements. The design model shows sufficient detail to allow construction contractors to bid. The detailed model provides fabrication and erection details required to physically construct and connect the sys- tem. A life-cycle model for HVAC systems includes a similar phased set of information needed to effectively support activities over the facility life cy- cle.

Traditional Army HVAC systems include four major subsystems. The first is the circulation of a thermal fluid, typically water that is heated or cooled, depending on the season. The second is the set of equipment needed to transfer energy from the thermal fluid to a thermal transfer fluid, typically air. The third subsystem is the transport mechanism for the thermal trans- fer fluid. This thermal transfer fluid is circulated by pressure differentials in the case of ductwork systems, or through convection currents in the case of radiators and fan coil units. The fourth subsystem is the set of sen- sors that provide a feedback loop to ensure proper delivery of thermal flu-id, heat-transfer rate of equipment, and adequate distribution of thermal transfer fluid.

The work accomplished in this effort complements efforts being conducted by organizations developing and promoting energy modeling tools. It is intended to establish a common minimum standard framework, used on typical Army facilities, to describe the components and topology of HVAC systems from an HVAC engineering design perspective. This work will help to establish the foundation for the delivery of HVAC models during the design stage, thereby easing the requirements on energy modeling tools that currently force the manual entry of higher-order HVAC infor- mation that is not available today.

4.8.1.1.3 Business Case Analysis

4.8.1.2 Participants and Stakeholders

4.8.1.2.1 Participants List

- Robert J. Hitchcock Hitchcock Consulting 6049 Shoo Fly Road Kelsey, CA 95667 [E,D]

- Nicholas Nisbet AEC3 UK Ltd 46 St Margaret's Grove Great Kingshill, High Wycombe, Bucks, HP15 6HP, UK [D]
- Christopher Wilkins and Matthew Tanis Hallam ICS 38 Eastwood Dr., Suite 200 South Burlington, VT 05403 [E]
- Reijo Hänninen and Tuomas Laine Olof Granlund Oy Malminkaari 21 Helsinki, Finland FIN-00701 [E]

4.8.1.2.2 Stakeholders List

- Engineer [E]
- Software Developers [D]

4.8.1.2.3. Stakeholders Coverage Analysis

Each participant in Clause 1.2.1 has role(s) indicated according to abbreviations defined in Clause 1.2.2.

4.8.2 Normative references

4.8.2.1 References and Standards

4.8.2.1.1. Reference Standards List

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 16739, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries

4.8.2.1.2. Reference Standards List (Other)

N/A

4.8.2.1.3. Reference Program and Project List

East, E.W., Danielle Love and Nicholas Nisbet, 2010. A Life-Cycle Model for Contracted Information Exchange. Proceedings of the CIB W78 2010: 27th International Conference –Cairo, Egypt, 16-18 November 2010.

East, E.W., 2012a. Construction Operations Building Information Exchange (COBie), http://www.wbdg.org/resources/cobie.php 6 August 2012]

East, E.W., 2007. Construction Operations Building Information Exchange (COBIE), Requirements Definition and Pilot Implementation Standard, ERDC/CERL TR-07-30, US Army Corps of Engineers, http://www.wbdg.org/pdfs/erdc_cerl_tr0730.pdf August 2007

4.8.3 Terms, definitions, symbols and abbreviated terms

For the purpose of this document, the following term, definitions, symbols and abbreviated terms apply.

4.8.3.1

attribute
unit of information within an entity, defined by a particular type or reference to a particular entity
NOTE There are three kinds of attributes: direct attributes, inverse attributes and derived attributes.

4.8.3.2
direct attribute
scalar values or collections including Set (unordered, unique), List (ordered), or Array (ordered, sparse) as defined in [ISO 10303-11]
NOTE Similar to the term "field" in common programming languages.

4.8.3.3
inverse attribute
unit of information defining queries for obtaining related data and enforcing referential integrity
NOTE Similar to the term "navigation property" in entity-relational programming frameworks.

4.8.3.4
derived attribute
unit of information computed from other attributes using an expression defined in the schema

4.8.3.5
constraints on attributes
data type restricting the values of attributes
NOTE 1 The most general constraint is about the existence of attribute values. There are basically two types: mandatory and optional attributes. Values of mandatory attributes must be provided whereas values of optional attributes may be omitted.
NOTE 2 For aggregation data types such as Set, List, or Array, the existence constraint is often refined by a minimal and maximal number of elements, which is also known as cardinality.

4.8.3.6
entity
class of information defined by common attributes and constraints as defined in [ISO 10303-11]
NOTE Similar to the term "class" in common programming languages but describing data structure only (not behavior such as methods).

4.8.3.7
identification
capability to find, retrieve, report, change, or delete specific instances without ambiguity

4.8.3.8
instance
occurrence of an entity
NOTE Similar to the term "instance of a class" in object oriented programming.

4.8.3.9
object
anything perceivable or conceivable that has a distinct existence, albeit not material

4.8.3.10
type
basic information construct derived from a primitive, an enumeration, or a select of entities
NOTE 1 Similar to the "Type" construct as defined in [ISO 10303-11].
NOTE 2 Similar in concept to "typedef" or "value type" in common programming languages.

4.8.3.11

select
construct that allows an attribute value to be one of multiple types or entities
NOTE 1 Similar to the "Select" construct as defined in [ISO 10303-11].
NOTE 2 Similar to a "marker interface" in common programming languages.

4.8.3.12
enumeration
construct that allows an attribute value to be one of multiple predefined values identified by name
NOTE 1 Similar to the "Enumeration" construct as defined in [ISO 10303-11].
NOTE 2 Similar in concept to "enum" in common programming languages.

4.8.3.13
actor
person, an organization, or person acting on behalf of an organization
NOTE A specialization of the general term object.

4.8.3.14
classification
categorization, the act of distributing things into classes or categories of the same type

4.8.3.15
constraint
restriction for a specified reason
NOTE A specialization of the general term control.

4.8.3.16
control
directive to meet specified requirements such as for scope, time, and/or cost
NOTE A specialization of the general term object.

4.8.3.17
dictionary
collection of words, terms or concepts, with their definition

4.8.3.18
element
tangible physical product that can be described by its shape representation, material representations, and other properties
NOTE A specialization of the general term product.

4.8.3.19
element occurrence
element's position within the project coordinate system and its containment within the spatial structure

4.8.3.20
external reference
link to information outside the data set, with direct relevance to the specific information the link originates from inside the data set

4.8.3.21
feature
parametric information and additional property information modifying the shape representation of an element to which it applies

4.8.3.22

group
collection of information that fulfills a specified purpose
NOTE A specialization of the general term object.

4.8.3.23
library
catalogue, database or holder of data, that is relevant to information in the data set
NOTE It is information referenced from an external source that is not copied into the data set.

4.8.3.24
object occurrence
characteristics of an object as an individual
NOTE Similar to "object", "instance", "individual" in other publications.

4.8.3.25
object type
common characteristics shared by multiple object-occurrences
NOTE Similar to "class", "template", "type" in other publications.

4.8.3.26
process
object-occurrence located in time, indicating "when"

4.8.3.27
process occurrence
conceptual object that may occur at a particular time

4.8.3.28
process type
common characteristics shared by multiple process occurrences

4.8.3.29
product
physical or conceptual object that occurs in space
NOTE It is specialization of the general term object.

4.8.3.30
product occurrence
physical or conceptual object that may have a location in space and shape characteristics

4.8.3.31
product type
common characteristics shared by multiple product occurrences

4.8.3.32
project
encapsulation of related information for a particular purpose providing context for information contained within
NOTE Context information may include default measurement units or representation context and precision.

4.8.3.33
property
unit of information that is dynamically defined as a particular entity instance
NOTE Similar to "late-bound" or "run-time" in programming terminology.

4.8.3.34
property occurrence
unit of information providing a value for a property identified by name

4.8.3.35
property template
metadata for a property including name, description, and data type
NOTE Similar in concept to "extension property" in common programming languages.

4.8.3.36
property set occurrence
unit of information containing a set of property occurrences, each having a unique name within the property set

4.8.3.37
property set template
set of property templates serving a common purpose and having applicability to objects of a particular entity
NOTE Similar in concept to "extension class" in common programming languages.

4,8,3,38
proxy
object that does not hold a specific object type information
NOTE a specialization of object occurrence.

4.8.3.39
quantity
measurement of a scope-based metric, specifically length, area, volume, weight, count, or time

4.8.3.40
quantity occurrence
unit of information providing a value for a quantity

4.8.3.41
quantity set
unit of information containing a set of quantity occurrences, each having a unique name within the quantity set

4.8.3.42
relationship
unit of information describing an interaction between items

4.8.3.43
representation
unit of information describing how an object is displayed, such as physical shape or topology

4.8.3.44
resource
entity with limited availability such as materials, labor, or equipment
NOTE 1 a specialization of the general term object.
NOTE 2 the "resource definition data schemas" section is unrelated to this concept.

4.8.3.45

resource occurrence
entity with inherent financial cost, which may be passed onto processes, products, and controls to which it is assigned

4.8.3.46
resource type
common characteristics shared by multiple resource occurrences

4.8.3.47
space
area or volume bounded actually or theoretically
NOTE a specialization of the general term product.

4.8.3.48
AEC
Architecture, Engineering, and Construction

4.8.3.49
AECFM
Architecture, Engineering, Construction, and Facilities Management

4.8.3.50
BIM
Building Information Modeling

4.8.3.51
GUID
Globally Unique Identifier

4.8.3.52
IFC
Industry Foundation Classes

4.8.3.53
IFD
International Framework for Dictionaries

4.8.3.54
SPF
STEP Physical File

4.8.3.55
STEP
STandard for the Exchange of Product data

4.8.3.56
URI
Uniform Resource Identifier

4.8.3.57
UUID
Universally Unique Identifier

4.8.4 Business Process Documentation

4.8.4.1 Process Models Provided

4.8.4.1.1 Business Process List

Programming
Begin programming
Engage design team
Document spatial, budget, and architectural OPRs
Document HVAC-related owner project requirements
Propose Mechanical Equipment Room (MER) requirements
Program spaces, areas, and budget
Coordinate development of concept design (incl. structural)
Select HVAC system types
Develop HVAC basis of design
Propose HVAC-related space requirements
Estimate energy performance
Document concept design and estimated costs

Schematic
Begin schematic design
Engage design team
Coordinate development of schematic design
Coordinate site plan
Coordinate structural design
Select main HVAC equipment
Coordinate HVAC-related equipment & MER layout
Update HVAC-related space requirements
Zone HVAC systems
Size Main HVAC equipment
Create piping schematics
Create air flow
Document HVAC systems schematic
Estimate energy performance
Estimate HVAC system costs
Document schematic design
Estimate schematic design costs
Proceed to coordinated design

Coordinated
Begin design development
Engage design team
Update HVAC-related space requirements
Update zoning of HVAC systems
Lay out distribution systems
Calculate system loads
Resize main HVAC equipment
Document HVAC design development
Estimate energy performance
Estimate HVAC system costs
Document design development
Estimate design development costs
Document HVAC construction documents
Document coordinated construction documents
Proceed to construction bidding

4.8.4.1.2 Business Process Descriptions

Programming

Begin programming
This is the start of project planning and design. Engage design team Type Intermediate Event Actor Architect Documentation The Architect engages all relevant Design Team members in the Programming process.

Document spatial, budget, and architectural OPRs
The Architect coordinates Programming and documents resulting Owner Project Requirements (OPRs) that focus on functional space requirements and budget constraints as well as higher-level objectives such as the Owner's desired level of energy-efficiency and aesthetic considerations.

Document HVAC-related owner project requirements
The HVAC Designer proposes and documents HVAC- related space requirements and provides input on potential impacts of identified OPRs on HVAC- related issues. This information is documented in the Owner Project Requirements (OPRs) report.

Propose Mechanical Equipment Room (MER) requirements
The HVAC Designer proposes HVAC-related Mechanical Equipment Room (MER) space requirements to be included in the overall project program. This information is documented in the Proposed MER Spaces report.

Program spaces, areas, and budget
The Architect coordinates Programming of spaces, areas, and budget for the overall project. Begin concept design
The Architect initiates the Concept Design phase following initial Programming phase.

Coordinate development of concept design (incl. structural)
The Architect coordinates the Concept Design phase including all relevant Design Team members, particularly the HVAC and Structural Designers. This task includes an initial creation of the er_exchange_building_model[basic](Preliminary Concept) that will be referenced by the HVAC Designer. The level of detail made available to the HVAC Designer by the Architect will determine the level to which the HVAC design is developed during Concept Design

Select HVAC system types
The HVAC Designer proposes HVAC System Types based on information established during Programming.

Develop HVAC basis of design
The HVAC Designer develops and documents a preliminary HVAC Basis of Design & Design Intent report including thought processes and assumptions behind the design decisions made to date to meet known OPRs. The HVAC Basis of Design & Design Intent documentation will be incrementally updated as design proceeds.

Propose HVAC-related space requirements
The HVAC Designer proposes HVAC-related space requirements to meet project OPRs for identified functional space types, based on generalized Industry Space Types Library requirements, and produces a preliminary ER Project Space Types dataset. See ER spatial requirements_aec3_20111109.xlsx and [GSA- 005_MVD]_IFC2x3_Concept_Design_BIM_2010_v7.pdf

Estimate energy performance
The HVAC Designer estimates the whole-building energy performance of the Concept Design model of the building. Energy performance estimation is not within the scope of this project, but is included in the

process map because of its importance in overall design decision making. Estimate HVAC construction costs Type Task Actor HVAC Designer Documentation The HVAC Designer estimates the construction costs (first cost) of Main HVAC equipment and systems for the Concept Design model. Cost estimation is not within the scope of this project, but is included in the process map because of its importance in overall design decision making.

Document concept design and estimated costs
The Architect coordinates documentation of a combined er_exchange_building_model[basic](Concept) including input from all relevant Design Team members. The er_exchange_building_model[basic](Concept) should include as much detail as has been established to date regarding the following: Project, Site, Building, Building Stories, Spaces (Functional), Space Types, Building Elements (General). Proceed to schematic design Type Intermediate Event Actor Architect Documentation Having received approval of Concept Design from the Owner, the Architect directs that the Design Team proceed to Schematic Design.

Schematic

Begin schematic design
This is the start of Schematic design. Engage design team Type Intermediate Event Actor Architect Documentation The Architect engages all relevant Design Team members in the Schematic design process. Coordinate development of schematic design Type Task Actor Architect Documentation The Architect coordinates interactions and communications between all relevant Design Team members in the Schematic Design process. This process builds on information documented in the er_exchange_building_model[basic](Concept).

Coordinate site plan
The Architect coordinates details of the site plan such as overall building orientation and site specifics such as available utilities and relevant codes and standards. Coordinate structural design Type Task Actor Architect Documentation The Architect coordinates interactions and communications related to the structural design between all relevant Design Team members.

Select main HVAC equipment
The HVAC Designer selects main HVAC equipment based on the identified system type. During Schematic Design these selections are often made from generic HVAC equipment libraries to allow sizing and costing based on generalities and uncertainties in building information at this time. This information is documented in the er_exchange_HVAC_model[equipment] (Schematic).

Coordinate HVAC-related equipment & MER layout
The HVAC Designer preliminarily locates HVAC equipment in the building and updates proposed MER spaces layout. This information is documented in the er_exchange_HVAC_model[space] (Schematic). Technical Space information may be combined into the er_exchange_building_model[basic](Schematic) if appropriate.

Update HVAC-related space requirements
The HVAC Designer reviews HVAC-related space requirements and updates them to meet currently specified OPRs for identified functional space types. This information is documented in the ER Project Space Types dataset and may be combined into the er_exchange_building_model[basic](Schematic) through coordination with Architect.

Zone HVAC systems
The HVAC Designer groups identified functional space types into appropriate HVAC zones (e.g., thermal zones) for subsequent HVAC-related calculations and design decisions. This information is documented in the er_exchange_HVAC_model[systems] (Schematic).

Size Main HVAC equipment
The HVAC Designer sizes main HVAC equipment based on the preliminary schematic design. At this stage of design, equipment is likely sized based on aggregated functional space area (“per ft2”) calculations. More detailed sizing calculations will be used if there is sufficient detail in the preliminary schematic design. This information is documented in the er_exchange_HVAC_model[equipment] (Schematic).

Create piping schematics
The HVAC Designer creates preliminary piping schematics for selected HVAC systems. This information is documented in the HVAC Basis of Design & Design Intent report and may ultimately be documented in the er_exchange_HVAC_model[systems] (Schematic) if there is sufficient confidence that subsequent design changes will not force substantial rework.

Create air flow diagrams
The HVAC Designer creates air flow diagrams for selected HVAC systems. This information is documented in the HVAC Basis of Design & Design Intent report and may ultimately be documented in the er_exchange_HVAC_model[systems] (Schematic) if there is sufficient confidence that subsequent design changes will not force substantial rework.

Document HVAC systems schematic design
The HVAC Designer documents HVAC systems information generated in the previous tasks, for which there is sufficient confidence that subsequent design changes will not force substantial rework, in the er_exchange_HVAC_model[systems] (Schematic) for coordination with Architect. Estimate energy performance Type Task Actor HVAC Designer Documentation The HVAC Designer estimates the whole-building energy performance of the preliminary schematic design of the building. Energy performance estimation is not within the scope of this project, but is included in the process map because of its importance in overall design decision making.

Estimate HVAC system costs
The HVAC Designer estimates the construction costs (first cost) of procuring and installing the selected Main HVAC equipment and designed systems for the Schematic Design model. Cost estimation is not within the scope of this project, but is included in the process map because of its importance in overall design decision making.

Document schematic design
The Architect coordinates documentation of a combined er_exchange_building_model[basic] (Schematic) including input from all relevant Design Team members. The er_exchange_building_model[basic] (Schematic) should include the following: Project, Site, Building, Building Stories, Spaces, Space Types, and Building Elements.

Estimate schematic design costs
The Architect coordinates estimation of construction costs (first cost) and life-cycle costs as needed for evaluation of the schematic design before moving on to coordinated design. Cost estimation is not within the scope of this project, but is included in the process map because of its importance in overall design decision making.

Proceed to coordinated design
Having received approval of Schematic Design from the Owner, the Architect directs that the Design Team proceed to Coordinated Design.

Coordination

Begin design development
It is assumed at this point that the Design Team has completed an er_exchange_building_model[basic] (Schematic) that includes building elements and space objects, and that this design has successfully

passed Schematic Design evaluation. This design provides at least a partial proposed building layout including space configuration and placement of other geometric elements. HVAC-related spaces such as Mechanical Equipment Room (MER) technical spaces and chases may not yet be defined by space objects.

Engage design team
The Architect engages all relevant Design Team members in the Design Development process. Coordinate design development Type Task Actor Architect Documentation The Architect coordinates interactions and communications between all relevant Design Team members in the Design Development process. Finalize selection of HVAC equipment Type Task Actor HVAC Designer Documentation The HVAC Designer checks generic HVAC equipment specified in the er_exchange_building_model[basic] (Schematic) and finalizes specification based on Manufacturer HVAC Equipment Libraries. This information is documented in the er_exchange_HVAC_model[equipment] (Design Development).

Update HVAC-related space requirements
The HVAC Designer checks HVAC-related Space Requirements such as thermal conditioning set points and updates information in the ER Project Space Types for Space Types specific to this project. See ER spatial requirements_aec3_20111109.xlsx and [GSA-005_MVD]_IFC2x3_Concept_Design_BIM_2010_v7.pdf Update construction types data Type Task Actor HVAC Designer Documentation The HVAC Designer checks HVAC-related Industry Construction Types Library data and updates information in the ER Project Construction Types for Building Elements specific to this project. See [GSA-005_MVD]_IFC2x3_Concept_Design_BIM_2010_v7.pdf

Update zoning of HVAC systems
The HVAC Designer checks previous groupings of Spaces into Thermal Zones and updates as appropriate. This information is documented in the er_exchange_HVAC_model[systems] (Design Development).

Lay out distribution systems
The HVAC Designer formally models the layout of the HVAC distribution systems and component equipment. This information is documented in er_exchange_HVAC_model[systems] (Design Development) and er_exchange_HVAC_model[equipment] (Design Development).

Calculate system loads
The HVAC Designer calculates the nominal (or design) requirements for the maximum thermal power addition or extraction required to maintain specified conditions in all thermal zone Spaces in the building under suitably chosen assumptions for weather and operation (the design conditions). These values are termed system loads. This information is documented in er_exchange_HVAC_model[systems] (Design Development) and er_exchange_HVAC_model[equipment] (Design Development).

Resize main HVAC equipment
The HVAC Designer recalculates the size of main HVAC plant equipment required to meet the calculated system loads. This information is documented in the er_exchange_HVAC_model[equipment] (Design Development).

Document HVAC design development
The HVAC Designer documents HVAC-related Space, Equipment, and Systems information generated during Design Development in er_exchange_HVAC_model[combined] (Design Development), which is the combined set of er_exchange_HVAC_model[space] (Design Development), er_exchange_HVAC_model[equipment] (Design Development) and er_exchange_HVAC_model[systems] (Design Development)

Estimate energy performance
The HVAC Designer estimates the whole-building energy performance of the Design Development model of the building. This task is not within the scope of this project, but is included in the process map because of its importance in overall design decision making.

Estimate HVAC system costs
The HVAC Designer estimates the construction costs (first cost) of procuring and installing the selected Main HVAC equipment and designed systems for the Design Development model. Cost estimation is not within the scope of this project, but is included in the process map because of its importance in overall design decision making.

Document design development
The Architect coordinates documentation of a combined er_exchange_building_model[basic] (Design Development) and er_exchange_HVAC_model[combined] (Design Development) including input from all relevant Design Team members and should include the following: Project, Site, Building, Building Stories, Spaces (including MER and chases), Space Types, Building Elements, HVAC Equipment with Location, HVAC Systems.

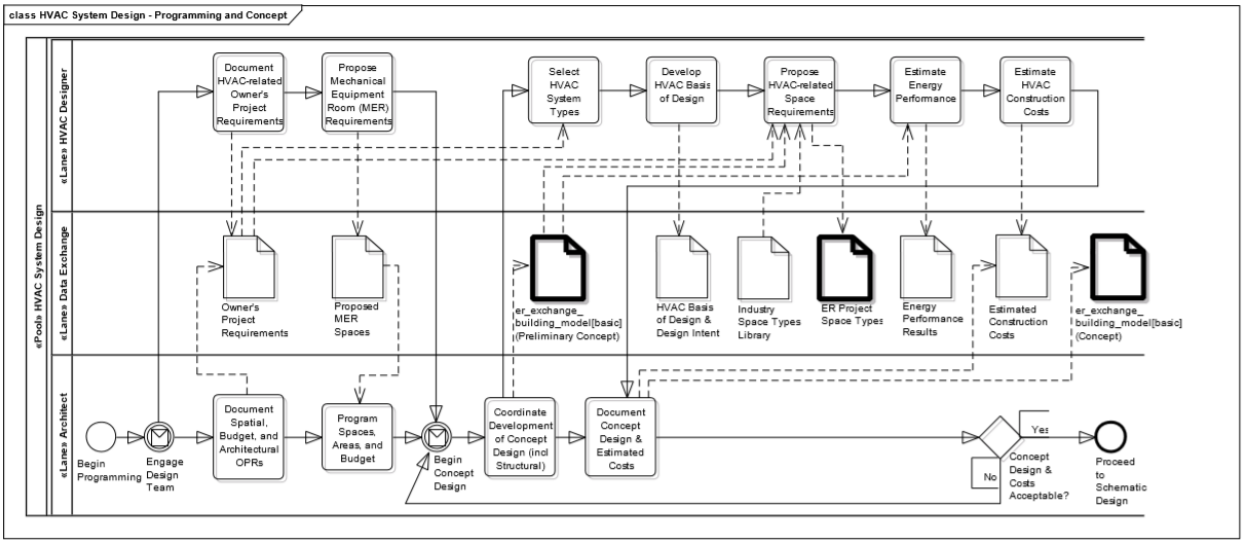
Estimate design development costs
The Architect coordinates estimation of construction and life-cycle costs as needed for evaluation of the building design before moving on to finalization during production of Construction Documents. Cost estimation is not within the scope of this project, but is included in the process map because of its importance in overall design decision making.

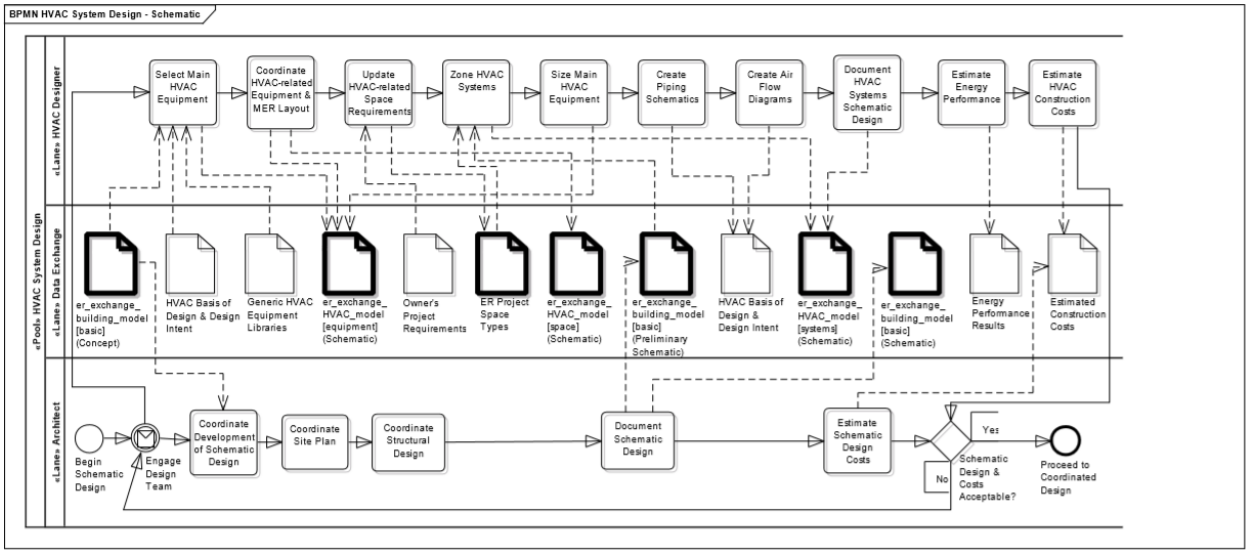
Document HVAC construction documents
The HVAC Designer documents Construction Documents for HVAC-related Space, Equipment, and Systems information in er_exchange_HVAC_model[combined] (Construction Documents).

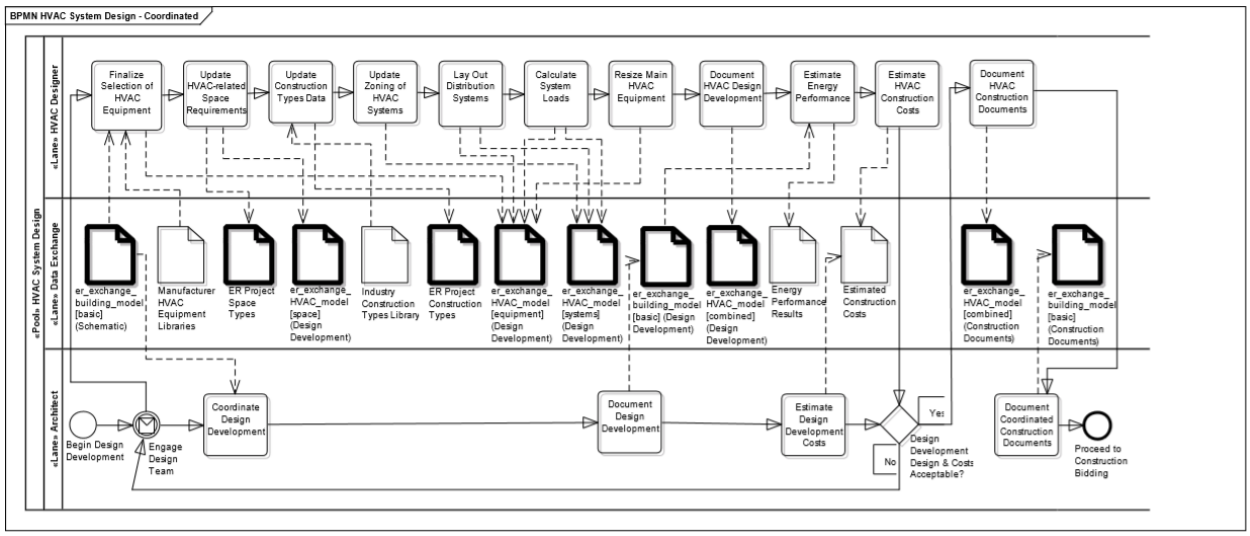
Document coordinated construction documents
The Architect coordinates documentation of a combined er_exchange_building_model[basic] (Construction Documents) and er_exchange_HVAC_model[combined] (Construction Documents) including input from all relevant Design Team members and should include the following: Project, Site, Building, Building Stories, Spaces (including MER and chases), Space Types, Building Elements, HVAC Equipment with Location, HVAC Systems.

Proceed to construction bidding
The Owner directs that the project proceed to construction bidding.

4.8.4.1.3 Business Process Model Diagrams







4.8.4.2 Representative Process Models

4.8.4.2.1 Stakeholder Coverage Analysis

The US Army Engineer Research and Development Center, Construction Engineering Research Laboratory (ERDC-CERL) has developed a core life- cycle building information model based on three example Army buildings: an Officer Duplex Apartment, a Headquarters Office, and a Clinic. These models were developed inconsistently over time by different modelers, and they reflect different levels of detail and quality of content across disci- plines. One cause for these differences is that an ontology describing the requirements for life-cycle modeling has not been identified for the heat- ing, ventilating, and air-conditioning (HVAC) domain. Current efforts at HVAC modeling typically focus only on those hard physical collisions be- tween ductwork and structural or architectural building elements. One of the difficulties of modeling the HVAC system is the complexity of the sys- tem itself.

Life-cycle information exchanges have previously been identified in the structural steel domain—an analysis model, a design model, and a detailed model. The analysis model reflects the needs of structural engineers to evaluate the requirements of the building and size the system to meet the facility's requirements. The design model shows sufficient detail to allow construction contractors to bid. The

detailed model provides fabrication and erection details required to physically construct and connect the sys- tem. A life-cycle model for HVAC systems includes a similar phased set of information needed to effectively support activities over the facility life cy- cle.

Traditional Army HVAC systems include four major subsystems. The first is the circulation of a thermal fluid, typically water that is heated or cooled, depending on the season. The second is the set of equipment needed to transfer energy from the thermal fluid to a thermal transfer fluid, typically air. The third subsystem is the transport mechanism for the thermal trans- fer fluid. This thermal transfer fluid is circulated by pressure differentials in the case of ductwork systems, or through convection currents in the case of radiators and fan coil units. The fourth subsystem is the set of sen- sors that provide a feedback loop to ensure proper delivery of thermal fluid, heat-transfer rate of equipment, and adequate distribution of thermal transfer fluid.

The work accomplished in this effort complements efforts being conducted by organizations developing and promoting energy modeling tools. It is intended to establish a common minimum standard framework, used on typical Army facilities, to describe the components and topology of HVAC systems from an HVAC engineering design perspective. This work will help to establish the foundation for the delivery of HVAC models during the design stage, thereby easing the requirements on energy modeling tools that currently force the manual entry of higher-order HVAC infor- mation that is not available today.

4.8.4.2.2 Process Coverage Analysis

The selection, design, and modeling of Heating, Ventilating, and Air- Conditioning (HVAC) domain equipment and systems involve iterative, evolutionary, collaborative processes. The HVAC Designer should be an active member of the design team, ideally beginning in early project planning, interacting with other team members and the Project Owner pri- marily through the Architect who is coordinating the overall design process.

The process model details in this document focus on the HVAC Designer Activity Tasks and Data Exchanges as part of the overall design process. Activity Tasks and Data Exchanges performed by other design team mem- bers are represented here only through interactions and communications with the Architect, without explicit representation of Tasks and Data Ex- changes performed by other team members.

4.8.4.2.3 Contract Documentary Deliverable List

N/A

4.8.4.2.4 Contract Documentary Deliverable Analysis

N/A

4.8.4.3 Process Models Formatting

4.8.4.3.1 BPMN Usage Description

Models in graphical format are provided in Clause 5.1.3.

4.8.5 Exchange requirements

4.8.5.1 Exchange requirements legibility

4.8.5.1.1 Exchange requirements list

Exchange
Manageable Components
Expected Attributes
Connections
Systems
Zones
Classifications

4.8.5.1.2 Exchange requirement classification list

Notation	Title
31-70 00 00 Handover Phase	31-70 00 00 Handover Phase
31-40 00 00 Design Phase	31-40 00 00 Design Phase

4.8.5.1.3 Exchange requirement coverage analysis

Exchange	Process	Sender	Receiver
Manageable Components	31-70 00 00 Handover Phase	34-20 11 21 Engineer	34-10 11 00 Owner
Expected Attributes	31-70 00 00 Handover Phase	34-20 11 21 Engineer	34-10 11 00 Owner
Connections	31-40 00 00 Design Phase	34-20 11 21 Engineer	34-20 11 21 Engineer
Systems	31-70 00 00 Handover Phase	34-20 11 21 Engineer	34-10 11 00 Owner
Zones	31-70 00 00 Handover Phase	34-20 11 21 Engineer	34-10 11 00 Owner
Classifications	31-70 00 00 Handover Phase	34-20 11 21 Engineer	34-10 11 00 Owner

4.8.5.2 Exchange requirements detail

4.8.5.2.1 Exchange requirements definition

Exchange	Description
Manageable Components	Managable HVAC components include all air distribution elements except duct segments and duct fittings, which are commonplace and so managed through their System.
Expected Attributes	Specific equipment is expected to have the scheduled properties documented in the COBie_Guide.
Connections	Connectivity is enforced by expecting ductwork segments to have at least two ports, ductwork fittings at least one and all ductwork segment and fitting ports to be matched.These three requirements together ensure that the HVAC system is complete, but that other equipment need only have ports if connected to the ductwork, and other ports (electrical or water) need not be satisfied.
Systems	Managable HVAC assets must be assigned to (at least) one system.
Zones	HVAC Zones group spaces with similar heating and venticaltion or air-conditioning requirements. Managable HVAC assets must be assigned to (at least) one space which is assigned to (at least) one Zone.
Classifications	Managable HVAC assets must be assigned to a Type with an industry standard product classification such as Omniclass Table

	23. Managable HVAC assets must be assigned to a System with an industry standard functional classification such as Omniclass Table 21.

4.8.5.2.2 Business rule list

Exchange	Entity	Concept
Manageable Components	IfcAirTerminal	Object Typing
		Space for inspection
		Managable assets in systems
		Managable assets in systems
	IfcAirTerminalBox	Object Typing
		Space for inspection
		Managable assets in systems
	IfcAirToAirHeatRecovery	Object Typing
		Space for inspection
		Managable assets in systems
	IfcChiller	Object Typing
		Space for inspection
		Managable assets in systems
	IfcCoil	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDamper	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDuctSilencer	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporativeCooler	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporator	Object Typing
		Space for inspection
		Managable assets in systems
	IfcFan	Object Typing
		Space for inspection
		Managable assets in systems
	IfcHeatExchanger	Object Typing
		Space for inspection

		Classification expected
		Managable assets in systems
	IfcHumidifier	Object Typing
		Space for inspection
		Managable assets in systems
	IfcUnitaryEquipment	Object Typing
		Space for inspection
		Managable assets in systems
		Classification expected
	IfcDuctSegment	At least two ports expected
	IfcDuctFitting	At least one port
	IfcAirTerminalBoxType	Predefined Type expected
		Classification expected
	IfcAirTerminalType	Predefined Type expected
		Classification expected
	IfcAirToAirHeatRecoveryType	Predefined Type expected
		Classification expected
	IfcChillerType	Predefined Type expected
		Classification expected
	IfcCoilType	Predefined Type expected
		Classification expected
	IfcDamperType	Predefined Type expected
		Classification expected
	IfcDuctSilencerType	Predefined Type expected
		Classification expected
	IfcEvaporativeCoolerType	Predefined Type expected
		Classification expected
	IfcEvaporatorType	Predefined Type expected
		Classification expected
	IfcFanType	Predefined Type expected
		Classification expected
	IfcHeatExchangerType	Predefined Type expected
		Classification expected
	IfcUnitaryEquipmentType	Predefined Type expected
		Classification expected
Expected Attributes	IfcAirTerminal	Object Typing
		Space for inspection
		Managable assets in systems
		Managable assets in systems

	IfcAirTerminalBox	Object Typing
		Space for inspection
		Managable assets in systems
	IfcAirToAirHeatRecovery	Object Typing
		Space for inspection
		Managable assets in systems
	IfcChiller	Object Typing
		Space for inspection
		Managable assets in systems
	IfcCoil	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDamper	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDuctSilencer	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporativeCooler	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporator	Object Typing
		Space for inspection
		Managable assets in systems
	IfcFan	Object Typing
		Space for inspection
		Managable assets in systems
	IfcHeatExchanger	Object Typing
		Space for inspection
		Classification expected
		Managable assets in systems
	IfcHumidifier	Object Typing
		Space for inspection
		Managable assets in systems
	IfcUnitaryEquipment	Object Typing
		Space for inspection
		Managable assets in systems
		Classification expected
	IfcDuctSegment	At least two ports expected

	IfcDuctFitting	At least one port
	IfcAirTerminalBoxType	Predefined Type expected
		Classification expected
	IfcAirTerminalType	Predefined Type expected
		Classification expected
	IfcAirToAirHeatRecoveryType	Predefined Type expected
		Classification expected
	IfcChillerType	Predefined Type expected
		Classification expected
	IfcCoilType	Predefined Type expected
		Classification expected
	IfcDamperType	Predefined Type expected
		Classification expected
	IfcDuctSilencerType	Predefined Type expected
		Classification expected
	IfcEvaporativeCoolerType	Predefined Type expected
		Classification expected
	IfcEvaporatorType	Predefined Type expected
		Classification expected
	IfcFanType	Predefined Type expected
		Classification expected
	IfcHeatExchangerType	Predefined Type expected
		Classification expected
	IfcUnitaryEquipmentType	Predefined Type expected
		Classification expected
Connections	IfcAirTerminal	Object Typing
		Space for inspection
		Managable assets in systems
		Managable assets in systems
	IfcAirTerminalBox	Object Typing
		Space for inspection
		Managable assets in systems
	IfcAirToAirHeatRecovery	Object Typing
		Space for inspection
		Managable assets in systems
	IfcChiller	Object Typing
		Space for inspection
		Managable assets in systems
	IfcCoil	Object Typing

		Space for inspection
		Managable assets in systems
	IfcDamper	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDuctSilencer	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporativeCooler	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporator	Object Typing
		Space for inspection
		Managable assets in systems
	IfcFan	Object Typing
		Space for inspection
		Managable assets in systems
	IfcHeatExchanger	Object Typing
		Space for inspection
		Classification expected
		Managable assets in systems
	IfcHumidifier	Object Typing
		Space for inspection
		Managable assets in systems
	IfcUnitaryEquipment	Object Typing
		Space for inspection
		Managable assets in systems
		Classification expected
	IfcDuctSegment	At least two ports expected
	IfcDuctFitting	At least one port
	IfcAirTerminalBoxType	Predefined Type expected
		Classification expected
	IfcAirTerminalType	Predefined Type expected
		Classification expected
	IfcAirToAirHeatRecoveryType	Predefined Type expected
		Classification expected
	IfcChillerType	Predefined Type expected
		Classification expected
	IfcCoilType	Predefined Type expected

		Classification expected
	IfcDamperType	Predefined Type expected
		Classification expected
	IfcDuctSilencerType	Predefined Type expected
		Classification expected
	IfcEvaporativeCoolerType	Predefined Type expected
		Classification expected
	IfcEvaporatorType	Predefined Type expected
		Classification expected
	IfcFanType	Predefined Type expected
		Classification expected
	IfcHeatExchangerType	Predefined Type expected
		Classification expected
	IfcUnitaryEquipmentType	Predefined Type expected
		Classification expected
Systems	IfcAirTerminal	Object Typing
		Space for inspection
		Managable assets in systems
		Managable assets in systems
	IfcAirTerminalBox	Object Typing
		Space for inspection
		Managable assets in systems
	IfcAirToAirHeatRecovery	Object Typing
		Space for inspection
		Managable assets in systems
	IfcChiller	Object Typing
		Space for inspection
		Managable assets in systems
	IfcCoil	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDamper	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDuctSilencer	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporativeCooler	Object Typing
		Space for inspection

		Managable assets in systems
	IfcEvaporator	Object Typing
		Space for inspection
		Managable assets in systems
	IfcFan	Object Typing
		Space for inspection
		Managable assets in systems
	IfcHeatExchanger	Object Typing
		Space for inspection
		Classification expected
		Managable assets in systems
	IfcHumidifier	Object Typing
		Space for inspection
		Managable assets in systems
	IfcUnitaryEquipment	Object Typing
		Space for inspection
		Managable assets in systems
		Classification expected
	IfcDuctSegment	At least two ports expected
	IfcDuctFitting	At least one port
	IfcAirTerminalBoxType	Predefined Type expected
		Classification expected
	IfcAirTerminalType	Predefined Type expected
		Classification expected
	IfcAirToAirHeatRecoveryType	Predefined Type expected
		Classification expected
	IfcChillerType	Predefined Type expected
		Classification expected
	IfcCoilType	Predefined Type expected
		Classification expected
	IfcDamperType	Predefined Type expected
		Classification expected
	IfcDuctSilencerType	Predefined Type expected
		Classification expected
	IfcEvaporativeCoolerType	Predefined Type expected
		Classification expected
	IfcEvaporatorType	Predefined Type expected
		Classification expected
	IfcFanType	Predefined Type expected

		Classification expected
	IfcHeatExchangerType	Predefined Type expected
		Classification expected
	IfcUnitaryEquipmentType	Predefined Type expected
		Classification expected
Zones	IfcAirTerminal	Space for inspection
		Managable assets in systems
		Managable assets in systems
	IfcAirTerminalBox	Object Typing
		Space for inspection
		Managable assets in systems
	IfcAirToAirHeatRecovery	Object Typing
		Space for inspection
		Managable assets in systems
	IfcChiller	Object Typing
		Space for inspection
		Managable assets in systems
	IfcCoil	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDamper	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDuctSilencer	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporativeCooler	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporator	Object Typing
		Space for inspection
		Managable assets in systems
	IfcFan	Object Typing
		Space for inspection
		Managable assets in systems
	IfcHeatExchanger	Object Typing
		Space for inspection
		Classification expected
		Managable assets in systems

	IfcHumidifier	Object Typing
		Space for inspection
		Managable assets in systems
	IfcUnitaryEquipment	Object Typing
		Space for inspection
		Managable assets in systems
		Classification expected
	IfcDuctSegment	At least two ports expected
	IfcDuctFitting	At least one port
	IfcAirTerminalBoxType	Predefined Type expected
		Classification expected
	IfcAirTerminalType	Predefined Type expected
		Classification expected
	IfcAirToAirHeatRecoveryType	Predefined Type expected
		Classification expected
	IfcChillerType	Predefined Type expected
		Classification expected
	IfcCoilType	Predefined Type expected
		Classification expected
	IfcDamperType	Predefined Type expected
		Classification expected
	IfcDuctSilencerType	Predefined Type expected
		Classification expected
	IfcEvaporativeCoolerType	Predefined Type expected
		Classification expected
	IfcEvaporatorType	Predefined Type expected
		Classification expected
	IfcFanType	Predefined Type expected
		Classification expected
	IfcHeatExchangerType	Predefined Type expected
		Classification expected
	IfcUnitaryEquipmentType	Predefined Type expected
		Classification expected
Classifications	IfcAirTerminal	Object Typing
		Space for inspection
		Managable assets in systems
		Managable assets in systems
	IfcAirTerminalBox	Object Typing
		Space for inspection

		Managable assets in systems
	IfcAirToAirHeatRecovery	Object Typing
		Space for inspection
		Managable assets in systems
	IfcChiller	Object Typing
		Space for inspection
		Managable assets in systems
	IfcCoil	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDamper	Object Typing
		Space for inspection
		Managable assets in systems
	IfcDuctSilencer	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporativeCooler	Object Typing
		Space for inspection
		Managable assets in systems
	IfcEvaporator	Object Typing
		Space for inspection
		Managable assets in systems
	IfcFan	Object Typing
		Space for inspection
		Managable assets in systems
	IfcHeatExchanger	Object Typing
		Space for inspection
		Classification expected
		Managable assets in systems
	IfcHumidifier	Object Typing
		Space for inspection
		Managable assets in systems
	IfcUnitaryEquipment	Object Typing
		Space for inspection
		Managable assets in systems
		Classification expected
	IfcDuctSegment	At least two ports expected
	IfcDuctFitting	At least one port
	IfcAirTerminalBoxType	Predefined Type expected

		Classification expected
	IfcAirTerminalType	Predefined Type expected
		Classification expected
	IfcAirToAirHeatRecoveryType	Predefined Type expected
		Classification expected
	IfcChillerType	Predefined Type expected
		Classification expected
	IfcCoilType	Predefined Type expected
		Classification expected
	IfcDamperType	Predefined Type expected
		Classification expected
	IfcDuctSilencerType	Predefined Type expected
		Classification expected
	IfcEvaporativeCoolerType	Predefined Type expected
		Classification expected
	IfcEvaporatorType	Predefined Type expected
		Classification expected
	IfcFanType	Predefined Type expected
		Classification expected
	IfcHeatExchangerType	Predefined Type expected
		Classification expected
	IfcUnitaryEquipmentType	Predefined Type expected
		Classification expected

4.8.5.2.3 Business rule definition

Business rule definitions are defined in the attached MVDXML file.

4.8.5.3 Exchange requirements reusability

This model view is domain-specific, therefore each exchange requirement is intended to be unique. While re-use of data definitions and concepts provides efficiencies in specification and implementation, re-use of exchange requirements would create redundancy and ambiguity of what should be chosen for a specified data exchange.

4.8.5.3.1 Related business process list

N/A

4.8.5.3.2 Related exchange requirement list

N/A

4.8.5.3.3 Related exchange requirement reuse analysis

N/A

4.8.6 Model view definition

4.8.6.1 Data Definition

4.8.6.1.1 Data definitions list

Entity
IfcAirTerminal
IfcAirTerminalBox
IfcAirToAirHeatRecovery
IfcChiller
IfcCoil
IfcDamper
IfcDuctSilencer
IfcEvaporativeCooler
IfcEvaporator
IfcFan
IfcHeatExchanger
IfcHumidifier
IfcUnitaryEquipment
IfcDuctSegment
IfcDuctFitting
IfcPort
IfcAirTerminalBoxType
IfcAirTerminalType
IfcAirToAirHeatRecoveryType
IfcChillerType
IfcCoilType
IfcDamperType
IfcDuctSilencerType
IfcEvaporativeCoolerType
IfcEvaporatorType
IfcFanType
IfcHeatExchangerType
IfcUnitaryEquipmentType
IfcZone
IfcSystem
IfcSpace

4.8.6.1.2 Data definitions

Entity	Definition
IfcAirTerminal	An air terminal is a terminating or origination point for the transfer of air between distribution system(s) and one or more spaces. It can also be used for the transfer of air between adjacent spaces. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcAirTerminal **SUBTYPE OF** IfcFlowTerminal; PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcAirTerminalTypeEnum.USERDEFINED) OR ((PredefinedType = IfcAirTerminalTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCAIRTERMINALTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions: **CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an IfcAirTerminalType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED. **CorrectTypeAssigned** : Either there is no air terminal type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcAirTerminalType. Inheritance Graph: **ENTITY** IfcAirTerminal **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcObject ObjectType : **OPTIONAL** IfcStrippedOptional; **INVERSE** IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects; **ENTITY** IfcProduct

	ObjectPlacement : **OPTIONAL** IfcStrippedOptional; Representation : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElement Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements; **ENTITY** IfcDistributionElement **INVERSE** HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcFlowTerminal **ENTITY** IfcAirTerminal PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcAirTerminal" type="ifc:IfcAirTerminal" substitutionGroup="ifc:IfcFlowTerminal" nillable="true"/> <xs:complexType name="IfcAirTerminal"> <xs:complexContent> <xs:extension base="ifc:IfcFlowTerminal"/> </xs:complexContent> </xs:complexType>
IfcAirTerminalBox	An air terminal box typically participates in an HVAC duct distribution system and is used to control or modulate the amount of air delivered to its downstream ductwork. An air terminal box type is often referred to as an "air flow regulator". HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcAirTerminalBox **SUBTYPE OF** IfcFlowController**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcAirTerminalBoxTypeEnum.USERDEFINED) OR ((PredefinedType = IfcAirTerminalBoxTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCAIRTERMINALBOXTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcAirTerminalBoxType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no air terminal box type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcAirTerminalBoxType.

Inheritance Graph:

ENTITY IfcAirTerminalBox

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;
Representation : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement;

ENTITY IfcDistributionFlowElement

INVERSE

	ENTITY IfcFlowController **ENTITY** IfcAirTerminalBox PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcAirTerminalBox" type="ifc:IfcAirTerminalBox" substitutionGroup="ifc:IfcFlowController" nillable="true"/> <xs:complexType name="IfcAirTerminalBox"> <xs:complexContent> <xs:extension base="ifc:IfcFlowController"/> </xs:complexContent> </xs:complexType>
IfcAirToAirHeatRecovery	An air-to-air heat recovery device employs a counter-flow heat exchanger between inbound and outbound air flow. It is typically used to transfer heat from warmer air in one chamber to cooler air in the second chamber (i.e., typically used to recover heat from the conditioned air being exhausted and the outside air being supplied to a building), resulting in energy savings from reduced heating (or cooling) requirements. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcAirToAirHeatRecovery **SUBTYPE OF** IfcEnergyConversionDevice**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcAirToAirHeatRecoveryTypeEnum.USERDEFINED) OR ((PredefinedType = IfcAirToAirHeatRecoveryTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCAIRTOAIRHEATRECOVERYTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions: **CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an IfcAirToAirHeatRecoveryType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED. **CorrectTypeAssigned** : Either there is no air-to-air heat recovery type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcAirToAirHeatRecoveryType. Inheritance Graph: **ENTITY** IfcAirToAirHeatRecovery **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition

	INVERSE HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcObject ObjectType : **OPTIONAL** IfcStrippedOptional; **INVERSE** IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects; **ENTITY** IfcProduct ObjectPlacement : **OPTIONAL** IfcStrippedOptional; Representation : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElement Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements; **ENTITY** IfcDistributionElement **INVERSE** HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcEnergyConversionDevice **ENTITY** IfcAirToAirHeatRecovery PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcAirToAirHeatRecovery" type="ifc:IfcAirToAirHeatRecovery" substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/> <xs:complexType name="IfcAirToAirHeatRecovery"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDevice"/> </xs:complexContent> </xs:complexType>
IfcChiller	A chiller is a device used to remove heat from a liquid via a vapor-compression or absorption refrigeration cycle to cool a fluid, typically water or a mixture of water and glycol. The chilled fluid is then used to cool and dehumidify air in a building. HISTORY New entity in IFC4

EXPRESS Specification:

```
ENTITY IfcChiller

 SUBTYPE OF IfcEnergyConversionDevice;

  PredefinedType          : OPTIONAL IfcStrippedOptional;

 WHERE

  CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcChillerTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcChillerTypeEnum.USERDEFINED) AND EXISTS
                          (SELF\IfcObject.ObjectType));
  CorrectTypeAssigned   : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCCHILLERTYPE' IN
                          TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;
```

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcChillerType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no chiller type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcChillerType.

Inheritance Graph:

```
ENTITY IfcChiller

 ENTITY IfcRoot

  GlobalId                : IfcGloballyUniqueId;
  OwnerHistory            : OPTIONAL IfcStrippedOptional;
  Name                    : OPTIONAL IfcStrippedOptional;
  Description             : OPTIONAL IfcStrippedOptional;

 ENTITY IfcObjectDefinition

 INVERSE

  HasAssignments          : SET OF IfcRelAssigns FOR RelatedObjects;
  HasAssociations         : SET OF IfcRelAssociates FOR RelatedObjects;

 ENTITY IfcObject

  ObjectType              : OPTIONAL IfcStrippedOptional;

 INVERSE

  IsTypedBy               : SET [0:1] OF IfcRelDefinesByType FOR RelatedObjects;

 ENTITY IfcProduct

  ObjectPlacement         : OPTIONAL IfcStrippedOptional;
  Representation          : OPTIONAL IfcStrippedOptional;

 INVERSE


 ENTITY IfcElement

  Tag                     : OPTIONAL IfcStrippedOptional;
```

	INVERSE ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements; **ENTITY** IfcDistributionElement **INVERSE** HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcEnergyConversionDevice **ENTITY** IfcChiller PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcChiller" type="ifc:IfcChiller" substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/> <xs:complexType name="IfcChiller"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDevice"/> </xs:complexContent> </xs:complexType>
IfcCoil	A coil is a device used to provide heat transfer between non-mixing media. A common example is a cooling coil, which utilizes a finned coil in which circulates chilled water, antifreeze, or refrigerant that is used to remove heat from air moving across the surface of the coil. A coil may be used either for heating or cooling purposes by placing a series of tubes (the coil) carrying a heating or cooling fluid into an airstream. The coil may be constructed from tubes bundled in a serpentine form or from finned tubes that give a extended heat transfer surface. Coils may also be used for non-airflow cases such as embedded in a floor slab. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcCoil **SUBTYPE OF** IfcEnergyConversionDevice**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcCoilTypeEnum.USERDEFINED) OR ((PredefinedType = IfcCoilTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCCOILTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions: **CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an IfcCoilType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED. **CorrectTypeAssigned** : Either there is no coil type object associated, i.e. the *IsTypedBy* inverse relationship is not

provided, or the associated type object has to be of type IfcCoilType.

Inheritance Graph:

ENTITY IfcCoil

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;
Representation : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement;

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcEnergyConversionDevice

ENTITY IfcCoil

PredefinedType : **OPTIONAL** IfcStrippedOptional;

	END_ENTITY; <xs:element name="IfcCoil" type="ifc:IfcCoil" substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/> <xs:complexType name="IfcCoil"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDevice"/> </xs:complexContent> </xs:complexType>
IfcDamper	A damper typically participates in an HVAC duct distribution system and is used to control or modulate the flow of air. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcDamper **SUBTYPE OF** IfcFlowController**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcDamperTypeEnum.USERDEFINED) OR ((PredefinedType = IfcDamperTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCDAMPERTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions: **CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an IfcDamperType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED. **CorrectTypeAssigned** : Either there is no damper type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcDamperType. Inheritance Graph: **ENTITY** IfcDamper **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcObject ObjectType : **OPTIONAL** IfcStrippedOptional; **INVERSE** IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

	ENTITY IfcProduct ObjectPlacement : **OPTIONAL** IfcStrippedOptional; Representation : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElement Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements; **ENTITY** IfcDistributionElement **INVERSE** HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcFlowController **ENTITY** IfcDamper PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcDamper" type="ifc:IfcDamper" substitutionGroup="ifc:IfcFlowController" nillable="true"/> <xs:complexType name="IfcDamper"> <xs:complexContent> <xs:extension base="ifc:IfcFlowController"/> </xs:complexContent> </xs:complexType>
IfcDuctSilencer	A duct silencer is a device that is typically installed inside a duct distribution system for the purpose of reducing the noise levels from air movement, fan noise, etc. in the adjacent space or downstream of the duct silencer device. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcDuctSilencer **SUBTYPE OF** IfcFlowTreatmentDevice**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcDuctSilencerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcDuctSilencerTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCDUCTSILENCERTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcDuctSilencerType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no duct silencer type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcDuctSilencerType.

Inheritance Graph:

ENTITY IfcDuctSilencer

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;
Representation : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement;

ENTITY IfcDistributionFlowElement

	INVERSE **ENTITY** IfcFlowTreatmentDevice **ENTITY** IfcDuctSilencer PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcDuctSilencer" type="ifc:IfcDuctSilencer" substitutionGroup="ifc:IfcFlowTreatmentDevice" nillable="true"/> <xs:complexType name="IfcDuctSilencer"> <xs:complexContent> <xs:extension base="ifc:IfcFlowTreatmentDevice"/> </xs:complexContent> </xs:complexType>
IfcEvaporativeCooler	An evaporative cooler is a device that cools air by saturating it with water vapor. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcEvaporativeCooler **SUBTYPE OF** IfcEnergyConversionDevice**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcEvaporativeCoolerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcEvaporativeCoolerTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCEVAPORATIVECOOLERTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions: **CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an IfcEvaporativeCoolerType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED. **CorrectTypeAssigned** : Either there is no evaporative cooler type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcEvaporativeCoolerType. Inheritance Graph: **ENTITY** IfcEvaporativeCooler **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition

	INVERSE HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcObject ObjectType : **OPTIONAL** IfcStrippedOptional; **INVERSE** IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects; **ENTITY** IfcProduct ObjectPlacement : **OPTIONAL** IfcStrippedOptional; Representation : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElement Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements; **ENTITY** IfcDistributionElement **INVERSE** HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcEnergyConversionDevice **ENTITY** IfcEvaporativeCooler PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcEvaporativeCooler" type="ifc:IfcEvaporativeCooler" substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/> <xs:complexType name="IfcEvaporativeCooler"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDevice"/> </xs:complexContent> </xs:complexType>
IfcEvaporator	An evaporator is a device in which a liquid refrigerent is vaporized and absorbs heat from the surrounding fluid. HISTORY New entity in IFC4

EXPRESS Specification:

```
ENTITY IfcEvaporator

  SUBTYPE OF IfcEnergyConversionDevice;

    PredefinedType          : OPTIONAL IfcStrippedOptional;

  WHERE

    CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcEvaporatorTypeEnum.USERDEFINED)
                              OR ((PredefinedType = IfcEvaporatorTypeEnum.USERDEFINED) AND EXISTS
                              (SELF\IfcObject.ObjectType));
    CorrectTypeAssigned   : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCEVAPORATORTYPE' IN
                              TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;
```

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcEvaporatorType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no evaporator type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcEvaporatorType.

Inheritance Graph:

```
ENTITY IfcEvaporator

  ENTITY IfcRoot

    GlobalId                : IfcGloballyUniqueId;
    OwnerHistory            : OPTIONAL IfcStrippedOptional;
    Name                    : OPTIONAL IfcStrippedOptional;
    Description             : OPTIONAL IfcStrippedOptional;

  ENTITY IfcObjectDefinition

  INVERSE

    HasAssignments          : SET OF IfcRelAssigns FOR RelatedObjects;
    HasAssociations         : SET OF IfcRelAssociates FOR RelatedObjects;

  ENTITY IfcObject

    ObjectType              : OPTIONAL IfcStrippedOptional;

  INVERSE

    IsTypedBy               : SET [0:1] OF IfcRelDefinesByType FOR RelatedObjects;

  ENTITY IfcProduct

    ObjectPlacement         : OPTIONAL IfcStrippedOptional;
    Representation          : OPTIONAL IfcStrippedOptional;

  INVERSE


  ENTITY IfcElement
```

	Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements; **ENTITY** IfcDistributionElement **INVERSE** HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcEnergyConversionDevice **ENTITY** IfcEvaporator PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcEvaporator" type="ifc:IfcEvaporator" substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/> <xs:complexType name="IfcEvaporator"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDevice"/> </xs:complexContent> </xs:complexType>
IfcFan	A fan is a device which imparts mechanical work on a gas. A typical usage of a fan is to induce airflow in a building services air distribution system. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcFan **SUBTYPE OF** IfcFlowMovingDevice**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcFanTypeEnum.USERDEFINED) OR ((PredefinedType = IfcFanTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCFANTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions: **CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an IfcFanType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED. **CorrectTypeAssigned** : Either there is no fan type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcFanType.

Inheritance Graph:

ENTITY IfcFan

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;
Representation : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement;

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcFlowMovingDevice

ENTITY IfcFan

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

	<xs:element name="IfcFan" type="ifc:IfcFan" substitutionGroup="ifc:IfcFlowMovingDevice" nillable="true"/> <xs:complexType name="IfcFan"> <xs:complexContent> <xs:extension base="ifc:IfcFlowMovingDevice"/> </xs:complexContent> </xs:complexType>
IfcHeatExchanger	A heat exchanger is a device used to provide heat transfer between non-mixing media such as plate and shell and tube heat exchangers. *IfcHeatExchanger* is commonly used on water-side distribution systems to recover energy from a liquid to another liquid (typically water-based), whereas *IfcAirToAirHeatRecovery* is commonly used on air-side distribution systems to recover energy from a gas to a gas (usually air). HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcHeatExchanger **SUBTYPE OF** IfcEnergyConversionDevice**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcHeatExchangerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcHeatExchangerTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCHEATEXCHANGERTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions: **CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an IfcHeatExchangerType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED. **CorrectTypeAssigned** : Either there is no heat exchanger type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcHeatExchangerType. Inheritance Graph: **ENTITY** IfcHeatExchanger **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcObject ObjectType : **OPTIONAL** IfcStrippedOptional;

	INVERSE IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects; **ENTITY** IfcProduct ObjectPlacement : **OPTIONAL** IfcStrippedOptional; Representation : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElement Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements; **ENTITY** IfcDistributionElement **INVERSE** HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcEnergyConversionDevice **ENTITY** IfcHeatExchanger PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcHeatExchanger" type="ifc:IfcHeatExchanger" substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/> <xs:complexType name="IfcHeatExchanger"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDevice"/> </xs:complexContent> </xs:complexType>
IfcHumidifier	A humidifier is a device that adds moisture into the air. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcHumidifier **SUBTYPE OF** IfcEnergyConversionDevice**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE**

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcHumidifierTypeEnum.USERDEFINED) OR ((PredefinedType = IfcHumidifierTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCHUMIDIFIERTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcHumidifierType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.
CorrectTypeAssigned : Either there is no humidifier type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcHumidifierType*.

Inheritance Graph:

ENTITY IfcHumidifier

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;
Representation : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

	HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcEnergyConversionDevice **ENTITY** IfcHumidifier PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcHumidifier" type="ifc:IfcHumidifier" substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/> <xs:complexType name="IfcHumidifier"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDevice"/> </xs:complexContent> </xs:complexType>
IfcUnitaryEquipment	Unitary equipment typically combine a number of components into a single product, such as air handlers, pre-packaged rooftop air-conditioning units, and split systems. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcUnitaryEquipment **SUBTYPE OF** IfcEnergyConversionDevice**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcUnitaryEquipmentTypeEnum.USERDEFINED) OR ((PredefinedType = IfcUnitaryEquipmentTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCUNITARYEQUIPMENTTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions: **CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an IfcUnitaryEquipmentType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED. **CorrectTypeAssigned** : Either there is no unitary equipment type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcUnitaryEquipmentType. Inheritance Graph: **ENTITY** IfcUnitaryEquipment **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional;

```
  Name                : OPTIONAL IfcStrippedOptional;
  Description         : OPTIONAL IfcStrippedOptional;

 ENTITY IfcObjectDefinition

 INVERSE

  HasAssignments      : SET OF IfcRelAssigns FOR RelatedObjects;
  HasAssociations     : SET OF IfcRelAssociates FOR RelatedObjects;

 ENTITY IfcObject

  ObjectType          : OPTIONAL IfcStrippedOptional;

 INVERSE

  IsTypedBy           : SET [0:1] OF IfcRelDefinesByType FOR RelatedObjects;

 ENTITY IfcProduct

  ObjectPlacement     : OPTIONAL IfcStrippedOptional;
  Representation      : OPTIONAL IfcStrippedOptional;

 INVERSE



 ENTITY IfcElement

  Tag                 : OPTIONAL IfcStrippedOptional;

 INVERSE

  ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;

 ENTITY IfcDistributionElement

 INVERSE

  HasPorts            : SET OF IfcRelConnectsPortToElement FOR RelatedElement;

 ENTITY IfcDistributionFlowElement

 INVERSE



 ENTITY IfcEnergyConversionDevice

 ENTITY IfcUnitaryEquipment

  PredefinedType      : OPTIONAL IfcStrippedOptional;

END_ENTITY;

 <xs:element name="IfcUnitaryEquipment" type="ifc:IfcUnitaryEquipment"
substitutionGroup="ifc:IfcEnergyConversionDevice" nillable="true"/>
 <xs:complexType name="IfcUnitaryEquipment">
  <xs:complexContent>
   <xs:extension base="ifc:IfcEnergyConversionDevice"/>
  </xs:complexContent>
 </xs:complexType>
```

IfcDuctSegment

A duct segment is used to typically join two sections of duct network.
HISTORY New entity in IFC4

EXPRESS Specification:

```
ENTITY IfcDuctSegment

  SUBTYPE OF IfcFlowSegment;

    PredefinedType          : OPTIONAL IfcStrippedOptional;

  WHERE

    CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
                              IfcDuctSegmentTypeEnum.USERDEFINED) OR ((PredefinedType =
                              IfcDuctSegmentTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
    CorrectTypeAssigned   : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCDUCTSEGMENTTYPE' IN
                              TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;
```

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcDuctSegmentType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no duct segment type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcDuctSegmentType*.

Inheritance Graph:

```
ENTITY IfcDuctSegment

  ENTITY IfcRoot

    GlobalId                : IfcGloballyUniqueId;
    OwnerHistory            : OPTIONAL IfcStrippedOptional;
    Name                    : OPTIONAL IfcStrippedOptional;
    Description             : OPTIONAL IfcStrippedOptional;

  ENTITY IfcObjectDefinition

  INVERSE

    HasAssignments          : SET OF IfcRelAssigns FOR RelatedObjects;
    HasAssociations         : SET OF IfcRelAssociates FOR RelatedObjects;

  ENTITY IfcObject

    ObjectType              : OPTIONAL IfcStrippedOptional;

  INVERSE

    IsTypedBy               : SET [0:1] OF IfcRelDefinesByType FOR RelatedObjects;

  ENTITY IfcProduct

    ObjectPlacement         : OPTIONAL IfcStrippedOptional;
    Representation          : OPTIONAL IfcStrippedOptional;

  INVERSE
```

	ENTITY IfcElement Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements; **ENTITY** IfcDistributionElement **INVERSE** HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement; **ENTITY** IfcDistributionFlowElement **INVERSE** **ENTITY** IfcFlowSegment **ENTITY** IfcDuctSegment PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcDuctSegment" type="ifc:IfcDuctSegment" substitutionGroup="ifc:IfcFlowSegment" nillable="true"/> <xs:complexType name="IfcDuctSegment"> <xs:complexContent> <xs:extension base="ifc:IfcFlowSegment"/> </xs:complexContent> </xs:complexType>
IfcDuctFitting	A duct fitting is a junction or transition in a ducted flow distribution system or used to connect duct segments, resulting in changes in flow characteristics to the fluid such as direction and flow rate. HISTORY New entity in IFC4 EXPRESS Specification: **ENTITY** IfcDuctFitting **SUBTYPE OF** IfcFlowFitting**;** PredefinedType : **OPTIONAL** IfcStrippedOptional; **WHERE** CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcDuctFittingTypeEnum.USERDEFINED) OR ((PredefinedType = IfcDuctFittingTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType)); CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCDUCTFITTINGTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType)); **END_ENTITY;** Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcDuctFittingType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no duct fitting type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcDuctFittingType*.

Inheritance Graph:

ENTITY IfcDuctFitting

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;
Representation : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedInStructures : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

HasPorts : **SET OF** IfcRelConnectsPortToElement **FOR** RelatedElement;

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcFlowFitting

	ENTITY IfcDuctFitting PredefinedType : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcDuctFitting" type="ifc:IfcDuctFitting" substitutionGroup="ifc:IfcFlowFitting" nillable="true"/> <xs:complexType name="IfcDuctFitting"> <xs:complexContent> <xs:extension base="ifc:IfcFlowFitting"/> </xs:complexContent> </xs:complexType>
IfcPort	An *IfcPort* provides the means for an element to connect to other elements. An *IfcPort* is associated with an *IfcElement*, it belongs to through the objectified relationship *IfcRelNests* if such port is fixed, or *IfcRelConnectsPortToElement* if such port is dynamically attached. Exactly two ports, belonging to two different elements, are connected with each other through the objectified relationship *IfcRelConnectsPorts*. An instance of *IfcElement* may have one or more points at which it connects to other instances of *IfcElement*. An instance of *IfcPort* is located at a point where a connection can occur. The location of the port is determined in the context of the local coordinate system of the element to which it belongs. HISTORY New entity in IFC2x2. **Containment Use Definitions** As a subordinate part being fully dependent on the master element the *IfcPort* shall have no independent containment relationship to the spatial structure. **Geometry Use Definition** The geometric representation of *IfcPort* is given by the *IfcProductDefinitionShape*, allowing multiple geometric representation. **Local Placement** The local placement for *IfcPort* is defined in its supertype *IfcProduct*. It is defined by the *IfcLocalPlacement*, which defines the local coordinate system that is referenced by all geometric representations. The *PlacementRelTo* relationship of *IfcLocalPlacement* shall point to the local placement of the master *IfcElement* or *IfcElementType* (relevant subtypes), which is related to the *IfcPort* by the relationship object *IfcRelNests* for fixed ports, or *IfcRelConnectsPortToElement* for dynamic ports. **Shape Representation** The geometry use definitions for the shape representation of the *IfcPort* is given at the level of its subtypes. EXPRESS Specification: **ENTITY** IfcPort **ABSTRACT SUPERTYPE OF**(IfcDistributionPort) **SUBTYPE OF** IfcProduct**;** **INVERSE** ContainedIn : **SET** [0:1] OF IfcRelConnectsPortToElement **FOR** RelatingPort; ConnectedFrom : **SET** [0:1] OF IfcRelConnectsPorts **FOR** RelatedPort; ConnectedTo : **SET** [0:1] OF IfcRelConnectsPorts **FOR** RelatingPort; **END_ENTITY;** Attribute Definitions: **ContainedIn** : Reference to the element to port connection relationship. The relationship then refers to the element in which this port is contained. IFC4 CHANGE The cardinality has been changed from 1:1 to 0:1. IFC4 DEPRECATION The

	inverse relationship is deprecated for fixed ports due to deprecation of IfcRelConnectsPortToElement for this usage. Use inverse relationship *Nests* instead. **ConnectedFrom** : Reference to a port that is connected by the objectified relationship. **ConnectedTo** : Reference to the port connection relationship. The relationship then refers to the other port to which this port is connected. Inheritance Graph: **ENTITY** IfcPort **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcObject ObjectType : **OPTIONAL** IfcStrippedOptional; **INVERSE** IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects; **ENTITY** IfcProduct ObjectPlacement : **OPTIONAL** IfcStrippedOptional; Representation : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcPort **INVERSE** ContainedIn : **SET** [0:1] OF IfcRelConnectsPortToElement **FOR** RelatingPort; ConnectedFrom : **SET** [0:1] OF IfcRelConnectsPorts **FOR** RelatedPort; ConnectedTo : **SET** [0:1] OF IfcRelConnectsPorts **FOR** RelatingPort; **END_ENTITY;** <xs:element name="IfcPort" type="ifc:IfcPort" abstract="true" substitutionGroup="ifc:IfcProduct" nillable="true"/> <xs:complexType name="IfcPort" abstract="true"> <xs:complexContent> <xs:extension base="ifc:IfcProduct"/> </xs:complexContent> </xs:complexType>
IfcAirTerminalBoxType	The flow controller type **IfcAirTerminalBoxType** defines commonly shared information for occurrences of air terminal boxes. The set of shared information may include: common properties with shared property sets common representations

common materials

common composition of elements

common ports

It is used to define an air terminal box type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcAirTerminalBoxType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of**IfcAirTerminalBoxType** are represented by instances of *IfcAirTerminalBox*. Refer to the documentation at *IfcAirTerminalBox* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

```
ENTITY IfcAirTerminalBoxType

  SUBTYPE OF IfcFlowControllerType;

    PredefinedType          : IfcAirTerminalBoxTypeEnum;

  WHERE

    CorrectPredefinedType : (PredefinedType <> IfcAirTerminalBoxTypeEnum.USERDEFINED) OR ((PredefinedType =
                              IfcAirTerminalBoxTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;
```

Attribute Definitions:

PredefinedType : The air terminal box type.

Inheritance Graph:

```
ENTITY IfcAirTerminalBoxType

  ENTITY IfcRoot

    GlobalId                : IfcGloballyUniqueId;
    OwnerHistory            : OPTIONAL IfcStrippedOptional;
    Name                    : OPTIONAL IfcStrippedOptional;
    Description             : OPTIONAL IfcStrippedOptional;

  ENTITY IfcObjectDefinition

  INVERSE

    HasAssignments          : SET OF IfcRelAssigns FOR RelatedObjects;
    HasAssociations         : SET OF IfcRelAssociates FOR RelatedObjects;

  ENTITY IfcTypeObject

    ApplicableOccurrence    : OPTIONAL IfcStrippedOptional;
    HasPropertySets         : OPTIONAL SET [1:?] OF IfcStrippedOptional;

  INVERSE

    Types                   : SET [0:1] OF IfcRelDefinesByType FOR RelatingType;

  ENTITY IfcTypeProduct

    RepresentationMaps      : OPTIONAL LIST [1:?] OF UNIQUE IfcStrippedOptional;
    Tag                     : OPTIONAL IfcStrippedOptional;

  INVERSE
```

	ENTITY IfcElementType ElementType : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcDistributionElementType **ENTITY** IfcDistributionFlowElementType **ENTITY** IfcFlowControllerType **ENTITY** IfcAirTerminalBoxType PredefinedType : IfcAirTerminalBoxTypeEnum; **END_ENTITY;** <xs:element name="IfcAirTerminalBoxType" type="ifc:IfcAirTerminalBoxType" substitutionGroup="ifc:IfcFlowControllerType" nillable="true"/> <xs:complexType name="IfcAirTerminalBoxType"> <xs:complexContent> <xs:extension base="ifc:IfcFlowControllerType"> <xs:attribute name="PredefinedType" type="ifc:IfcAirTerminalBoxTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcAirTerminalType	The flow terminal type **IfcAirTerminalType** defines commonly shared information for occurrences of air terminals. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define an air terminal type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcAirTerminalType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of**IfcAirTerminalType** are represented by instances of *IfcAirTerminal*. Refer to the documentation at *IfcAirTerminal* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcAirTerminalType **SUBTYPE OF** IfcFlowTerminalType**;** PredefinedType : IfcAirTerminalTypeEnum; **WHERE** CorrectPredefinedType : (PredefinedType <> IfcAirTerminalTypeEnum.USERDEFINED) OR ((PredefinedType = IfcAirTerminalTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType)); **END_ENTITY;** Inheritance Graph: **ENTITY** IfcAirTerminalType

	ENTITY IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcTypeObject ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional; HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional; **INVERSE** Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType; **ENTITY** IfcTypeProduct RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional; Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElementType ElementType : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcDistributionElementType **ENTITY** IfcDistributionFlowElementType **ENTITY** IfcFlowTerminalType **ENTITY** IfcAirTerminalType PredefinedType : IfcAirTerminalTypeEnum; **END_ENTITY;** <xs:element name="IfcAirTerminalType" type="ifc:IfcAirTerminalType" substitutionGroup="ifc:IfcFlowTerminalType" nillable="true"/> <xs:complexType name="IfcAirTerminalType"> <xs:complexContent> <xs:extension base="ifc:IfcFlowTerminalType"> <xs:attribute name="PredefinedType" type="ifc:IfcAirTerminalTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcAirToAirHeatRecoveryType	The energy conversion device type **IfcAirToAirHeatRecoveryType** defines commonly shared information for occurrences of air to air heat recoverys. The set of shared information may include: common properties with shared property sets common representations

common materials

common composition of elements

common ports

It is used to define an air to air heat recovery type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcAirToAirHeatRecoveryType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcAirToAirHeatRecoveryType** are represented by instances of *IfcAirToAirHeatRecovery*. Refer to the documentation at *IfcAirToAirHeatRecovery* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

```
ENTITY IfcAirToAirHeatRecoveryType

  SUBTYPE OF IfcEnergyConversionDeviceType;

    PredefinedType          : IfcAirToAirHeatRecoveryTypeEnum;

  WHERE

    CorrectPredefinedType : (PredefinedType <> IfcAirToAirHeatRecoveryTypeEnum.USERDEFINED) OR ((PredefinedType =
                              IfcAirToAirHeatRecoveryTypeEnum.USERDEFINED) AND
                              EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;
```

Attribute Definitions:

PredefinedType : Defines the type of air to air heat recovery device.

Inheritance Graph:

```
ENTITY IfcAirToAirHeatRecoveryType

  ENTITY IfcRoot

    GlobalId                : IfcGloballyUniqueId;
    OwnerHistory            : OPTIONAL IfcStrippedOptional;
    Name                    : OPTIONAL IfcStrippedOptional;
    Description             : OPTIONAL IfcStrippedOptional;

  ENTITY IfcObjectDefinition

  INVERSE

    HasAssignments          : SET OF IfcRelAssigns FOR RelatedObjects;
    HasAssociations         : SET OF IfcRelAssociates FOR RelatedObjects;

  ENTITY IfcTypeObject

    ApplicableOccurrence    : OPTIONAL IfcStrippedOptional;
    HasPropertySets         : OPTIONAL SET [1:?] OF IfcStrippedOptional;

  INVERSE

    Types                   : SET [0:1] OF IfcRelDefinesByType FOR RelatingType;

  ENTITY IfcTypeProduct

    RepresentationMaps      : OPTIONAL LIST [1:?] OF UNIQUE IfcStrippedOptional;
    Tag                     : OPTIONAL IfcStrippedOptional;
```

	INVERSE **ENTITY** IfcElementType ElementType : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcDistributionElementType **ENTITY** IfcDistributionFlowElementType **ENTITY** IfcEnergyConversionDeviceType **ENTITY** IfcAirToAirHeatRecoveryType PredefinedType : IfcAirToAirHeatRecoveryTypeEnum; **END_ENTITY;** <xs:element name="IfcAirToAirHeatRecoveryType" type="ifc:IfcAirToAirHeatRecoveryType" substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/> <xs:complexType name="IfcAirToAirHeatRecoveryType"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDeviceType"> <xs:attribute name="PredefinedType" type="ifc:IfcAirToAirHeatRecoveryTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcChillerType	The energy conversion device type **IfcChillerType** defines commonly shared information for occurrences of chillers. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a chiller type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcChillerType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcChillerType** are represented by instances of *IfcChiller*. Refer to the documentation at *IfcChiller* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcChillerType **SUBTYPE OF** IfcEnergyConversionDeviceType**;** PredefinedType : IfcChillerTypeEnum; **WHERE** CorrectPredefinedType : (PredefinedType <> IfcChillerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcChillerTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType)); **END_ENTITY;** Attribute Definitions: **PredefinedType** : Defines the typical types of chillers (e.g., air-cooled, water-cooled, etc.).

Inheritance Graph:

ENTITY IfcChillerType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcEnergyConversionDeviceType

ENTITY IfcChillerType

PredefinedType : IfcChillerTypeEnum;

END_ENTITY;

```
<xs:element name="IfcChillerType" type="ifc:IfcChillerType" substitutionGroup="ifc:IfcEnergyConversionDeviceType"
nillable="true"/>
<xs:complexType name="IfcChillerType">
 <xs:complexContent>
  <xs:extension base="ifc:IfcEnergyConversionDeviceType">
   <xs:attribute name="PredefinedType" type="ifc:IfcChillerTypeEnum" use="optional"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

IfcCoilType	The energy conversion device type **IfcCoilType** defines commonly shared information for occurrences of coils. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a coil type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcCoilType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcCoilType** are represented by instances of *IfcCoil*. Refer to the documentation at *IfcCoil* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcCoilType **SUBTYPE OF** IfcEnergyConversionDeviceType**;** PredefinedType : IfcCoilTypeEnum; **WHERE** CorrectPredefinedType : (PredefinedType <> IfcCoilTypeEnum.USERDEFINED) OR ((PredefinedType = IfcCoilTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType)); **END_ENTITY;** Attribute Definitions: **PredefinedType** : Defines typical types of coils (e.g., Cooling, Heating, etc.) Inheritance Graph: **ENTITY** IfcCoilType **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcTypeObject ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional; HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional; **INVERSE** Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType; **ENTITY** IfcTypeProduct

	RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional; Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElementType ElementType : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcDistributionElementType **ENTITY** IfcDistributionFlowElementType **ENTITY** IfcEnergyConversionDeviceType **ENTITY** IfcCoilType PredefinedType : IfcCoilTypeEnum; **END_ENTITY;** <xs:element name="IfcCoilType" type="ifc:IfcCoilType" substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/> <xs:complexType name="IfcCoilType"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDeviceType"> <xs:attribute name="PredefinedType" type="ifc:IfcCoilTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcDamperType	The flow controller type **IfcDamperType** defines commonly shared information for occurrences of dampers. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a damper type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcDamperType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcDamperType** are represented by instances of *IfcDamper*. Refer to the documentation at *IfcDamper* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcDamperType **SUBTYPE OF** IfcFlowControllerType**;** PredefinedType : IfcDamperTypeEnum; **WHERE** CorrectPredefinedType : (PredefinedType <> IfcDamperTypeEnum.USERDEFINED) OR ((PredefinedType = IfcDamperTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType)); **END_ENTITY;**

Attribute Definitions:

PredefinedType : Type of damper.

Inheritance Graph:

ENTITY IfcDamperType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcFlowControllerType

ENTITY IfcDamperType

PredefinedType : IfcDamperTypeEnum;

END_ENTITY;

```
<xs:element name="IfcDamperType" type="ifc:IfcDamperType" substitutionGroup="ifc:IfcFlowControllerType"
nillable="true"/>
<xs:complexType name="IfcDamperType">
 <xs:complexContent>
  <xs:extension base="ifc:IfcFlowControllerType">
   <xs:attribute name="PredefinedType" type="ifc:IfcDamperTypeEnum" use="optional"/>
  </xs:extension>
```

	</xs:complexContent> </xs:complexType>
IfcDuctSilencerType	The flow treatment device type **IfcDuctSilencerType** defines commonly shared information for occurrences of duct silencers. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a duct silencer type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcDuctSilencerType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of**IfcDuctSilencerType** are represented by instances of *IfcDuctSilencer*. Refer to the documentation at *IfcDuctSilencer* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcDuctSilencerType **SUBTYPE OF** IfcFlowTreatmentDeviceType; PredefinedType : IfcDuctSilencerTypeEnum; **WHERE** CorrectPredefinedType : (PredefinedType <> IfcDuctSilencerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcDuctSilencerTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType)); **END_ENTITY;** Attribute Definitions: **PredefinedType** : The type of duct silencer. Inheritance Graph: **ENTITY** IfcDuctSilencerType **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcTypeObject ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional; HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional; **INVERSE**

	Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType; **ENTITY** IfcTypeProduct RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional; Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElementType ElementType : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcDistributionElementType **ENTITY** IfcDistributionFlowElementType **ENTITY** IfcFlowTreatmentDeviceType **ENTITY** IfcDuctSilencerType PredefinedType : IfcDuctSilencerTypeEnum; **END_ENTITY;** <xs:element name="IfcDuctSilencerType" type="ifc:IfcDuctSilencerType" substitutionGroup="ifc:IfcFlowTreatmentDeviceType" nillable="true"/> <xs:complexType name="IfcDuctSilencerType"> <xs:complexContent> <xs:extension base="ifc:IfcFlowTreatmentDeviceType"> <xs:attribute name="PredefinedType" type="ifc:IfcDuctSilencerTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcEvaporativeCoolerType	The energy conversion device type **IfcEvaporativeCoolerType** defines commonly shared information for occurrences of evaporative coolers. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a evaporative cooler type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcEvaporativeCoolerType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of**IfcEvaporativeCoolerType** are represented by instances of *IfcEvaporativeCooler*. Refer to the documentation at *IfcEvaporativeCooler* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcEvaporativeCoolerType **SUBTYPE OF** IfcEnergyConversionDeviceType**;** PredefinedType : IfcEvaporativeCoolerTypeEnum; **WHERE**

CorrectPredefinedType : (PredefinedType <> IfcEvaporativeCoolerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcEvaporativeCoolerTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Defines the type of evaporative cooler.

Inheritance Graph:

ENTITY IfcEvaporativeCoolerType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcEnergyConversionDeviceType

ENTITY IfcEvaporativeCoolerType

PredefinedType : IfcEvaporativeCoolerTypeEnum;

END_ENTITY;

<xs:element name="IfcEvaporativeCoolerType" type="ifc:IfcEvaporativeCoolerType" substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/>

	<xs:complexType name="IfcEvaporativeCoolerType"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDeviceType"> <xs:attribute name="PredefinedType" type="ifc:IfcEvaporativeCoolerTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcEvaporatorType	The energy conversion device type **IfcEvaporatorType** defines commonly shared information for occurrences of evaporators. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a evaporator type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcEvaporatorType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of**IfcEvaporatorType** are represented by instances of *IfcEvaporator*. Refer to the documentation at *IfcEvaporator* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcEvaporatorType **SUBTYPE OF** IfcEnergyConversionDeviceType**;** PredefinedType : IfcEvaporatorTypeEnum; **WHERE** CorrectPredefinedType : (PredefinedType <> IfcEvaporatorTypeEnum.USERDEFINED) OR ((PredefinedType = IfcEvaporatorTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType)); **END_ENTITY;** Attribute Definitions: **PredefinedType** : Defines the type of evaporator. Inheritance Graph: **ENTITY** IfcEvaporatorType **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcTypeObject ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;

	HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional; **INVERSE** Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType; **ENTITY** IfcTypeProduct RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional; Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElementType ElementType : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcDistributionElementType **ENTITY** IfcDistributionFlowElementType **ENTITY** IfcEnergyConversionDeviceType **ENTITY** IfcEvaporatorType PredefinedType : IfcEvaporatorTypeEnum; **END_ENTITY;** <xs:element name="IfcEvaporatorType" type="ifc:IfcEvaporatorType" substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/> <xs:complexType name="IfcEvaporatorType"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDeviceType"> <xs:attribute name="PredefinedType" type="ifc:IfcEvaporatorTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcFanType	The flow moving device type **IfcFanType** defines commonly shared information for occurrences of fans. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a fan type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcFanType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcFanType** are represented by instances of *IfcFan*. Refer to the documentation at *IfcFan* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcFanType **SUBTYPE OF** IfcFlowMovingDeviceType**;** PredefinedType : IfcFanTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcFanTypeEnum.USERDEFINED) OR ((PredefinedType = IfcFanTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Defines the type of fan typically used in building services.

Inheritance Graph:

ENTITY IfcFanType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcFlowMovingDeviceType

ENTITY IfcFanType

PredefinedType : IfcFanTypeEnum;

END_ENTITY;

	<xs:element name="IfcFanType" type="ifc:IfcFanType" substitutionGroup="ifc:IfcFlowMovingDeviceType" nillable="true"/> <xs:complexType name="IfcFanType"> <xs:complexContent> <xs:extension base="ifc:IfcFlowMovingDeviceType"> <xs:attribute name="PredefinedType" type="ifc:IfcFanTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcHeatExchangerType	The energy conversion device type **IfcHeatExchangerType** defines commonly shared information for occurrences of heat exchangers. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a heat exchanger type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcHeatExchangerType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of**IfcHeatExchangerType** are represented by instances of *IfcHeatExchanger*. Refer to the documentation at *IfcHeatExchanger* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcHeatExchangerType **SUBTYPE OF** IfcEnergyConversionDeviceType**;** PredefinedType : IfcHeatExchangerTypeEnum; **WHERE** CorrectPredefinedType : (PredefinedType <> IfcHeatExchangerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcHeatExchangerTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType)); **END_ENTITY;** Attribute Definitions: **PredefinedType** : Defines the basic types of heat exchanger (e.g., plate, shell and tube, etc.). Inheritance Graph: **ENTITY** IfcHeatExchangerType **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcTypeObject

	ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional; HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional; **INVERSE** Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType; **ENTITY** IfcTypeProduct RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional; Tag : **OPTIONAL** IfcStrippedOptional; **INVERSE** **ENTITY** IfcElementType ElementType : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcDistributionElementType **ENTITY** IfcDistributionFlowElementType **ENTITY** IfcEnergyConversionDeviceType **ENTITY** IfcHeatExchangerType PredefinedType : IfcHeatExchangerTypeEnum; **END_ENTITY;** <xs:element name="IfcHeatExchangerType" type="ifc:IfcHeatExchangerType" substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/> <xs:complexType name="IfcHeatExchangerType"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDeviceType"> <xs:attribute name="PredefinedType" type="ifc:IfcHeatExchangerTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcUnitaryEquipmentType	The energy conversion device type **IfcUnitaryEquipmentType** defines commonly shared information for occurrences of unitary equipments. The set of shared information may include: common properties with shared property sets common representations common materials common composition of elements common ports It is used to define a unitary equipment type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcUnitaryEquipmentType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of**IfcUnitaryEquipmentType** are represented by instances of *IfcUnitaryEquipment*. Refer to the documentation at *IfcUnitaryEquipment* for supported property sets, materials, composition, and ports. EXPRESS Specification: **ENTITY** IfcUnitaryEquipmentType

SUBTYPE OF IfcEnergyConversionDeviceType**;**

PredefinedType : IfcUnitaryEquipmentTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcUnitaryEquipmentTypeEnum.USERDEFINED) OR ((PredefinedType = IfcUnitaryEquipmentTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : The type of unitary equipment.

Inheritance Graph:

ENTITY IfcUnitaryEquipmentType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcEnergyConversionDeviceType

ENTITY IfcUnitaryEquipmentType

	PredefinedType : IfcUnitaryEquipmentTypeEnum; **END_ENTITY;** <xs:element name="IfcUnitaryEquipmentType" type="ifc:IfcUnitaryEquipmentType" substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/> <xs:complexType name="IfcUnitaryEquipmentType"> <xs:complexContent> <xs:extension base="ifc:IfcEnergyConversionDeviceType"> <xs:attribute name="PredefinedType" type="ifc:IfcUnitaryEquipmentTypeEnum" use="optional"/> </xs:extension> </xs:complexContent> </xs:complexType>
IfcZone	A zone is a group of spaces, partial spaces or other zones. Zone structures may not be hierarchical (in contrary to the spatial structure of a project - see *IfcSpatialStructureElement*), i.e. one individual *IfcSpace* may be associated with zero, one, or several *IfcZone*'s. *IfcSpace*'s are grouped into an *IfcZone* by using the objectified relationship*IfcRelAssignsToGroup* as specified at the supertype *IfcGroup*. NOTE Certain use cases may restrict the freedom of non hierarchical relationships. In some building service use cases the zone denotes a view based delimited volume for the purpose of analysis and calculation. This type of zone cannot overlap with respect to that analysis, but may overlap otherwise. HISTORY New entity in IFC1.0 IFC4 CHANGE The entity is now subtyped from *IfcSystem* (not its supertype *IfcGroup*) with upward compatibility for file based exchange. EXPRESS Specification: **ENTITY** IfcZone **SUBTYPE OF** IfcSystem**;** LongName : **OPTIONAL** IfcStrippedOptional; **WHERE** WR1 : (SIZEOF(SELF\IfcGroup.IsGroupedBy) = 0) OR (SIZEOF (QUERY (temp <* SELF\IfcGroup.IsGroupedBy[1].RelatedObjects | NOT(('IFCPRODUCTEXTENSION.IFCZONE' IN TYPEOF(temp)) OR ('IFCPRODUCTEXTENSION.IFCSPACE' IN TYPEOF(temp)) OR ('IFCPRODUCTEXTENSION.IFCSPATIALZONE' IN TYPEOF(temp))))) = 0); **END_ENTITY;** Attribute Definitions: Formal Propositions: **WR1** : An IfcZone is grouped by the objectified relationship IfcRelAssignsToGroup. Only objects of type IfcSpace, IfcZone and *IfcSpatialZone* are allowed as *RelatedObjects*. Inheritance Graph: **ENTITY** IfcZone **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional;

	ENTITY IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcObject ObjectType : **OPTIONAL** IfcStrippedOptional; **INVERSE** IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects; **ENTITY** IfcGroup **INVERSE** IsGroupedBy : **SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup; **ENTITY** IfcSystem **INVERSE** **ENTITY** IfcZone LongName : **OPTIONAL** IfcStrippedOptional; **END_ENTITY;** <xs:element name="IfcZone" type="ifc:IfcZone" substitutionGroup="ifc:IfcSystem" nillable="true"/> <xs:complexType name="IfcZone"> <xs:complexContent> <xs:extension base="ifc:IfcSystem"/> </xs:complexContent> </xs:complexType>
IfcSystem	A system is an organized combination of related parts within an AEC product, composed for a common purpose or function or to provide a service. A system is essentially a functionally related aggregation of products. The grouping relationship to one or several instances of *IfcProduct* (the system members) is handled by *IfcRelAssignsToGroup*. NOTE The use of *IfcSystem* often applies to the representation of building services related systems, such as the piping system, cold water system, etc. Members within such a system may or may not be connected using the connectivity related entities (through *IfcDistributionPort*). HISTORY New entity in IFC1.0 EXPRESS Specification: **ENTITY** IfcSystem **SUPERTYPE OF**(**ONEOF**(IfcDistributionSystem, IfcZone)) **SUBTYPE OF** IfcGroup**;** **INVERSE** **END_ENTITY;**

	Attribute Definitions: Inheritance Graph: **ENTITY** IfcSystem **ENTITY** IfcRoot GlobalId : IfcGloballyUniqueId; OwnerHistory : **OPTIONAL** IfcStrippedOptional; Name : **OPTIONAL** IfcStrippedOptional; Description : **OPTIONAL** IfcStrippedOptional; **ENTITY** IfcObjectDefinition **INVERSE** HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects; HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects; **ENTITY** IfcObject ObjectType : **OPTIONAL** IfcStrippedOptional; **INVERSE** IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects; **ENTITY** IfcGroup **INVERSE** IsGroupedBy : **SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup; **ENTITY** IfcSystem **INVERSE** **END_ENTITY;** <xs:element name="IfcSystem" type="ifc:IfcSystem" substitutionGroup="ifc:IfcGroup" nillable="true"/> <xs:complexType name="IfcSystem"> <xs:complexContent> <xs:extension base="ifc:IfcGroup"/> </xs:complexContent> </xs:complexType>
IfcSpace	A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building. A space is associated to a building storey (or in case of exterior spaces to a site). A space may span over several connected spaces. Therefore a space group provides for a collection of spaces included in a storey. A space can also be decomposed in parts, where each part defines a partial space. This is defined by the *CompositionType*attribute of the supertype *IfcSpatialStructureElement* which is interpreted as follow: COMPLEX = space group ELEMENT = space PARTIAL = partial space NOTE View definitions and implementation agreements may restrict spaces with *CompositionType*=ELEMENT to be non-

overlapping.

The *IfcSpace* is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical). The spatial structure elements are linked together by using the objectified relationship *IfcRelAggregates*.

Figure 180 shows the *IfcSpace* as part of the spatial structure. It also serves as the spatial container for space related elements.
NOTE Detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.

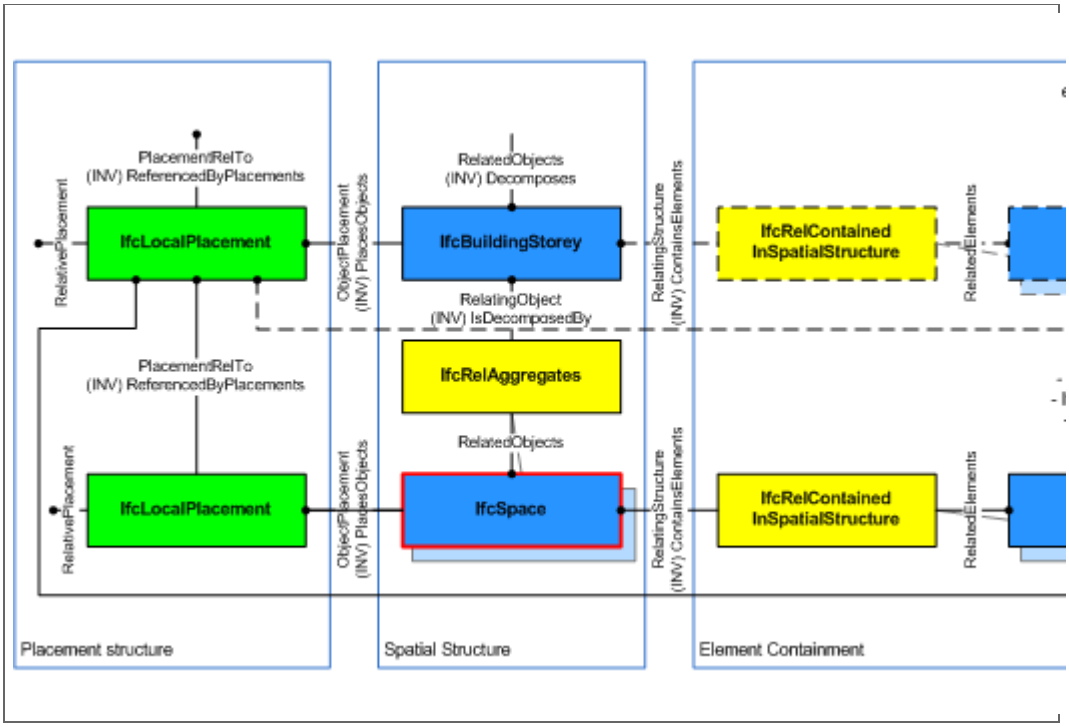


Figure 180 — Space composition

The following guidelines should apply for using the *Name*, *Description*, *LongName* and *ObjectType* attributes.

Name holds the unique name (or space number) from the plan.

Description holds any additional information field the user may have specified, there are no further recommendations.

LongName holds the full name of the space, it is often used in addition to the *Name*, if a number is assigned to the room, then the descriptive name is exchanged as *LongName*.

ObjectType holds the space type, i.e. usually the functional category of the space .
NOTE In cases of inconsistency between the geometric representation of the *IfcSpace* and the combined geometric representations of the surrounding *IfcRelSpaceBoundary*, the geometric representation of the space should take priority over the geometric representation of the surrounding space boundaries.

HISTORY New entity in IFC1.0

Attribute Use Definition

Figure 181 describes the heights and elevations of the *IfcSpace*.

elevation of the space (top of construction slab) equals elevation of storey: provided by *IfcBuildingStorey.Elevation* relative to *IfcBuilding.ElevationOfRefHeight*

elevation of the space flooring (top of flooring on top of slab): provided by *IfcSpace.ElevationWithFlooring* relative to *IfcBuilding.ElevationOfRefHeight*

height of space (top of slab below to bottom of slab above): provided by BaseQuantity with Name="Height"

floor height of space (top of slab below to top of flooring): provided by BaseQuantity with Name="FinishFloorHeight"

net height of space (top of flooring to bottom of suspended ceiling): provided by BaseQuantity with

Name="FinishCeilingHeight"

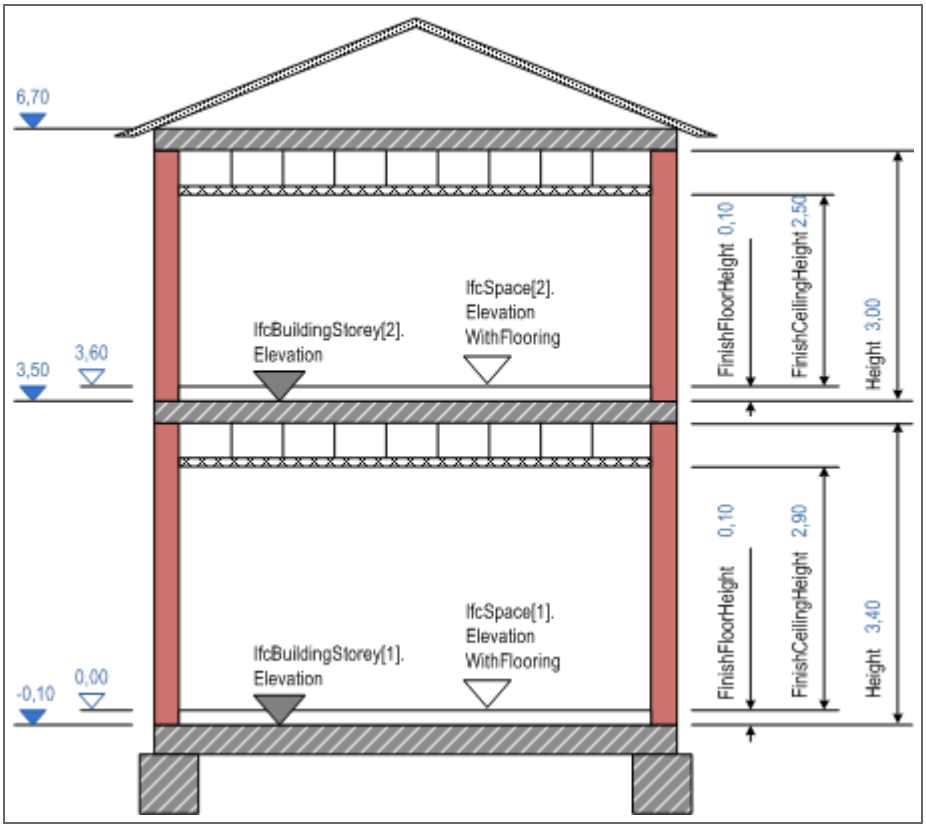


Figure 181 — Space elevations

EXPRESS Specification:

ENTITY IfcSpace

SUBTYPE OF IfcSpatialStructureElement**;**

PredefinedType : **OPTIONAL** IfcStrippedOptional;
ElevationWithFlooring : **OPTIONAL** IfcStrippedOptional;

INVERSE

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcSpaceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSpaceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCPRODUCTEXTENSION.IFCSPACETYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Attribute Definitions:

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcSpaceType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no space type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcSpaceType*.

Inheritance Graph:

ENTITY IfcSpace

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcStrippedOptional;
Name : **OPTIONAL** IfcStrippedOptional;
Description : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcStrippedOptional;
Representation : **OPTIONAL** IfcStrippedOptional;

INVERSE

ENTITY IfcSpatialElement

LongName : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencesElements : **SET OF** IfcRelReferencedInSpatialStructure **FOR** RelatingStructure;

ENTITY IfcSpatialStructureElement

CompositionType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcSpace

PredefinedType : **OPTIONAL** IfcStrippedOptional;
ElevationWithFlooring : **OPTIONAL** IfcStrippedOptional;

INVERSE

	END_ENTITY; <xs:element name="IfcSpace" type="ifc:IfcSpace" substitutionGroup="ifc:IfcSpatialStructureElement" nillable="true"/> <xs:complexType name="IfcSpace"> <xs:complexContent> <xs:extension base="ifc:IfcSpatialStructureElement"/> </xs:complexContent> </xs:complexType>

4.8.6.1.3 Data definition reference schema list

Reference	Description
ISO 16739:2013	Industry Foundation Classes (IFC) for data sharing in the construction and facilities management industries

4.8.6.2 Concept list

Entity	Concept
IfcAirTerminal	Object Typing
	Space for inspection
	Managable assets in systems
	Managable assets in systems
IfcAirTerminalBox	Object Typing
	Space for inspection
	Managable assets in systems
IfcAirToAirHeatRecovery	Object Typing
	Space for inspection
	Managable assets in systems
IfcChiller	Object Typing
	Space for inspection
	Managable assets in systems
IfcCoil	Object Typing
	Space for inspection
	Managable assets in systems
IfcDamper	Object Typing
	Space for inspection
	Managable assets in systems
IfcDuctSilencer	Object Typing
	Space for inspection
	Managable assets in systems
IfcEvaporativeCooler	Object Typing
	Space for inspection
	Managable assets in systems
IfcEvaporator	Object Typing

	Space for inspection
	Managable assets in systems
IfcFan	Object Typing
	Space for inspection
	Managable assets in systems
IfcHeatExchanger	Object Typing
	Space for inspection
	Classification expected
	Managable assets in systems
IfcHumidifier	Object Typing
	Space for inspection
	Managable assets in systems
	Classification expected
IfcUnitaryEquipment	Object Typing
	Space for inspection
	Managable assets in systems
	Classification expected
IfcDuctSegment	At least two ports expected
IfcDuctFitting	At least one port
IfcPort	Ports to be twinned
IfcAirTerminalBoxType	Predefined Type expected
	Classification expected
IfcAirTerminalType	Predefined Type expected
	Classification expected
IfcAirToAirHeatRecoveryType	Predefined Type expected
	Classification expected
IfcChillerType	Predefined Type expected
	Classification expected
IfcCoilType	Predefined Type expected
	Classification expected
IfcDamperType	Predefined Type expected
	Classification expected
IfcDuctSilencerType	Predefined Type expected
	Classification expected
IfcEvaporativeCoolerType	Predefined Type expected
	Classification expected
IfcEvaporatorType	Predefined Type expected
	Classification expected
IfcFanType	Predefined Type expected

	Classification expected
IfcHeatExchangerType	Predefined Type expected
	Classification expected
IfcUnitaryEquipmentType	Predefined Type expected
	Classification expected
IfcZone	Classification expected
IfcSystem	Classification expected
IfcSpace	Classification expected

4.8.6.2.1 Concept definitions

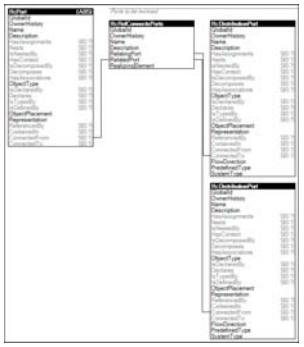
Entity	Concept	Definition
IfcAirTerminal		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
	Managable assets in systems	
IfcAirTerminalBox		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcAirToAirHeatRecovery		

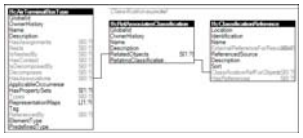
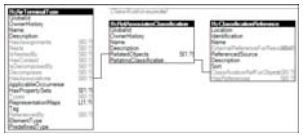
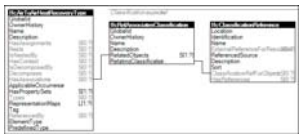
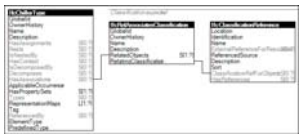
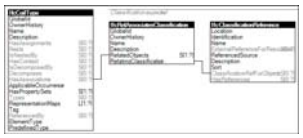
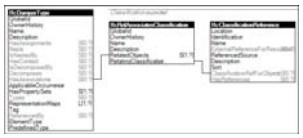
		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcChiller		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcCoil		

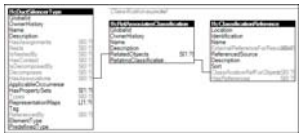
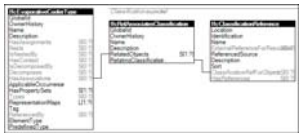
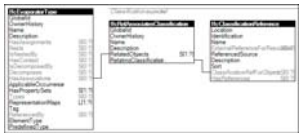
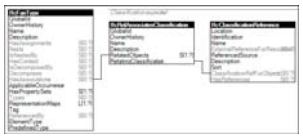
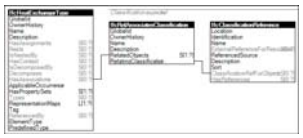
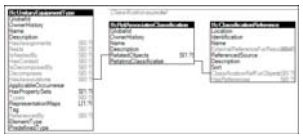
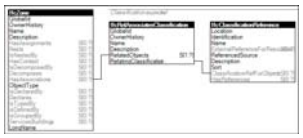
	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcDamper		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcDuctSilencer		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcEvaporativeCooler		

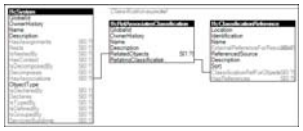
		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcEvaporator		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcFan		

	Object Typing	
	Space for inspection	
	Managable assets in systems	
IfcHeatExchanger		
	Object Typing	
	Space for inspection	
	Classification expected	
	Managable assets in systems	
IfcHumidifier		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
	Classification expected	
IfcUnitaryEquipment		

		
	Object Typing	
	Space for inspection	
	Managable assets in systems	
	Classification expected	
IfcDuctSegment		
	At least two ports expected	
IfcDuctFitting		
	At least one port	
IfcPort		

	Ports to be twinned	
IfcAirTerminalBoxType		
	Predefined Type expected	
	Classification expected	
IfcAirTerminalType		
	Predefined Type expected	
	Classification expected	
IfcAirToAirHeatRecoveryType		
	Predefined Type expected	
	Classification expected	
IfcChillerType		
	Predefined Type expected	
	Classification expected	
IfcCoilType		
	Predefined Type expected	
	Classification expected	
IfcDamperType		
	Predefined Type expected	
	Classification expected	
IfcDuctSilencerType		

		
	Predefined Type expected	
	Classification expected	
IfcEvaporativeCoolerType		
	Predefined Type expected	
	Classification expected	
IfcEvaporatorType		
	Predefined Type expected	
	Classification expected	
IfcFanType		
	Predefined Type expected	
	Classification expected	
IfcHeatExchangerType		
	Predefined Type expected	
	Classification expected	
IfcUnitaryEquipmentType		
	Predefined Type expected	
	Classification expected	
IfcZone		

	Classification expected	
IfcSystem		
	Classification expected	
IfcSpace		
	Classification expected	

4.8.6.2.2 Concept attributes list

Entity	Concept	Attributes	Description

4.8.6.2.3 Concept relationship description

Concept	General Concept

4.8.6.2.4 Concept requirements applicability

Entity	Concept	Exchanges
IfcAirTerminal	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Systems; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcAirTerminalBox	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcAirToAirHeatRecovery	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for	Manageable Components; Expected Attributes ; Connections; Systems; Zones;

	inspection	Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcChiller	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcCoil	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcDamper	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcDuctSilencer	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcEvaporativeCooler	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcEvaporator	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;

	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcFan	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcHeatExchanger	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcHumidifier	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	
IfcUnitaryEquipment	Object Typing	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Space for inspection	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Managable assets in systems	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcDuctSegment	At least two ports expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;

IfcDuctFitting	At least one port	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcPort	Ports to be twinned	
IfcAirTerminalBoxType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcAirTerminalType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcAirToAirHeatRecoveryType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcChillerType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcCoilType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcDamperType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcDuctSilencerType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcEvaporativeCoolerType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcEvaporatorType	Predefined Type	Manageable Components; Expected Attributes ; Connections; Systems; Zones;

	expected	Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcFanType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcHeatExchangerType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcUnitaryEquipmentType	Predefined Type expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
	Classification expected	Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications; Manageable Components; Expected Attributes ; Connections; Systems; Zones; Classifications;
IfcZone	Classification expected	
IfcSystem	Classification expected	
IfcSpace	Classification expected	

4.8.6.3 Concept list

Template		
Definition		
	Object Typing	
Association		
	Classification	
		Classification expected
Connectivity		
	Spatial Structure	
		Spatial Containment
	Element Connectivity	
		Port Connectivity
Product		
	Managable assets in systems	
Product Type		
	Predefined Type expected	

4.8.6.3.1 Related existing concept list

Template		
Project		
	Project Declaration	
		Object Type Definitions
		Property Set Templates
	Project Units	
		SI Units
		Conversion Units
		Derived Units
	Project Context	
	Project Classification Information	
	Project Document Information	
	Project Library Information	
Roots		
	Identity	
	Revision Control	
	Descriptions	
Definition		
	Object Typing	
	Property Sets	
		Property Sets
		Property Sets for Types
		Property Sets for Performance
		Properties on Occurrences
	Quantity Sets	
		Quantity Sets
		Quantities on Occurrences
	Property Set Templates	
Association		
	Classification	
		Classification expected
	Document	
	Library	
	Approval	
	Constraint	
		External Data Constraints
		Parametric Constraints

		Requirement Constraints
	Material	
		Material Solid
		Material Layer Set
		Material Layer Set Usage
		Material Profile Set
		Material Profile Set Usage
		Material Constituents
Assignment		
	Actor Assignment	
	Control Assignment	
	Group Assignment	
	Product Assignment	
	Process Assignment	
	Resource Assignment	
	Product Type Assignment	
	Process Type Assignment	
	Resource Type Assignment	
Composition		
	Aggregation	
		Element Composition
		Element Decomposition
		Spatial Composition
		Spatial Decomposition
	Voiding	
	Nesting	
	Ports	
	Type-Based Ports	
Connectivity		
	Spatial Structure	
		Spatial Container
		Spatial Containment
		Space Coverings
	Space Boundaries	
	Element Connectivity	
		Path Connectivity
		Port Connectivity
	Control Flow	
	Filling	

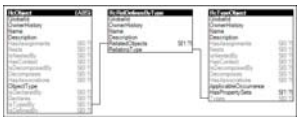
	Structural Activity	
	Structural Connectivity	
	Sequential Connectivity	
	Interference	
	Space Coverings	
	Voiding	
Actor		
	Organization Role	
Control		
	Cost	
	Calendar	
Product		
	Placement	
	Geometry	
		Box Geometry
		Annotation Geometry
		Axis Geometry
		Footprint Geometry
		Profile Geometry
		Surface Geometry
		Body Geometry
		Clearance Geometry
		Lighting Geometry
		Survey Points Geometry
		Mapped Geometry
		Box Geometry
		Row Geometry
	Topology	
		Reference Topology
	Port Types	
	Spatial Naming	
	Site Location	
	Building Location	
	Building Storey Elevation	
	Grid	
	Managable assets in systems	
Product Type		
	Product Type Representation	
		Axis Geometry

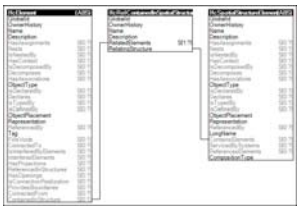
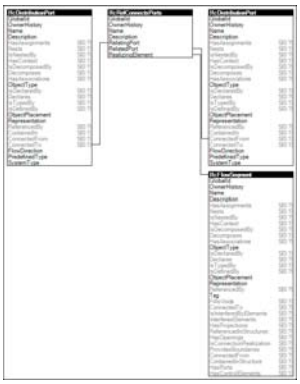
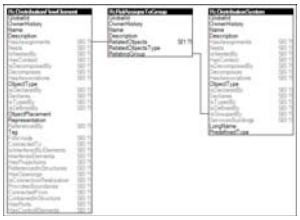
		Body Geometry
		Lighting Geometry
		Clearance Geometry
	Predefined Type expected	
Process		
	Task Scheduling	
	Event Types	
	Event Triggers	
Process Type		
Resource		
	Resource Cost	
	Resource Quantity	
Resource Type		
	Resource Cost Rate	
COBie Metadata		
COBie Contact		

4.8.6.3.2 Concept business rule list

Template	Rules
Project	
Roots	
Definition	
Association	
Assignment	
Composition	
Connectivity	
Actor	
Control	
Product	
Product Type	
Process	
Process Type	
Resource	
Resource Type	
COBie Metadata	
COBie Contact	

4.8.6.3.3 Concept business rule description

Template			Description
Definition			
	Object Typing		Object Occurrences may be defined by a particular Object Type, where such type describes common characteristics. Such characteristics include common properties, shapes, materials, composition, and other concepts described at particular entities. An object occurrence may have similar state as its object type, overridden state for particular characteristics, or have no defined type object. A pair of entities are defined for various object occurrences and object types, where such object occurrence entity may only be defined using a particular object type entity. For example, the *IfcTank* occurrence object entity has a corresponding *IfcTankType* type object entity. Many object occurrence and object type entities have an attribute named *PredefinedType* consisting of a specific enumeration. Such predefined type essentially provides another level of inheritance to further differentiate objects without the need for additional entities. Predefined types are not just informational; various rules apply such as applicable property sets, part composition, and distribution ports. For scenarios of object types having part compositions, such parts may be reflected at object occurrences having separate state. For example, a *wall type* may define a particular arrangement of studs, a *wall occurrence* may reflect the same arrangement of studs, and studs within the wall occurrence may participate in specific relationships that do not exist at the type such as being connected to an electrical junction box. 
Association			
	Classification		Objects, type objects, properties, and some resource schema entities can be further described by associating references to external sources of information. The source of information can be: a classification system; a dictionary server; any external catalogue that classifies the object further; a service that combine the above features. An individual item within the external source of information can be selected. It then applies the inherent meaning of the item to the object or property. 
		Classification expected	Classification is expected to support both analysis and FM activities. 
Connectivity			Objects may participate in various connectivity relationships with other objects.
	Spatial Structure		Spatial structures, such as site, building, storey, or spaces, may contain physical elements, including building elements, distribution elements, and furnishing elements. The containment relationship between the physical elements and the spatial structures is hierarchical, i.e. a physical element shall only be contained within a single spatial structure. EXAMPLE An *IfcBeam* is placed within the spatial hierarchy using the objectified relationship *IfcRelContainedInSpatialStructure*, refering to it by its inverse attribute *SELF\IfcElement.ContainedInStructure*. Subtypes of *IfcSpatialStructureElement* are valid spatial containers, with *IfcBuildingStorey* being the default container. The spatial containment relationship, together with the Spatial decomposition relationship,

			being hierarchical as well, establishes the hiearchical project tree structure in a building information model. EXAMPLE The *IfcBuildingStorey* that logically contains the *IfcBeam* decomposes the *IfcBuilding* using the *IfcRelAggregates* relationship. Therefore the *IfcBeam* is also indirectly contained in the building.
		Spatial Containment	Spatial structures may contain physical elements, including building elements, distribution elements, and furnishing elements. 
	Element Connectivity		Elements may be connected to other elements, where the *RelatingElement* is of equal or higher priority, is generally constructed first, and/or anchors the *RelatedElement*. 
		Port Connectivity	Ports on objects may be connected using elements such as cables, ducts, or pipes.Once Components within a System has some ports, then the connectivity should be complete and continuous. The presence of ports for air, water and electrical connections on complex equipment does not imply that all such connectivity is expected: only that if for example the HVAC segments and fittings have ports, then they will need to connect properly to the equipment's air ports. 
Product			A product is an occurence of a physical or virtual object with finite spatial extent.
	Managable assets in systems		Manageable HVAC assets must be assigned to a *system*. 

Product Type			Product types define explicit product models or parametric product families, that may be instantiated in buildings.
	Predefined Type expected		A predefined type other than *NOTDEFINED* shall be provided

4.8.6.4 MVD Schema Listing

File	Format
HVACie2013.exp	EXPRESS schema definition
HVACie2013.xsd	XML schema definition (XSD)
HVACie2013.mvdxml	MVDXML schema transform

4.8.6.4.1 MVD Format Description

Format	Extension	MIME	Reference
IFC-SPF	.ifc	application/step	ISO 10303-21
IFC-XML	.ifcxml	application/xml	ISO 10303-28

4.8.6.4.2 MVD Dynamic Schema Analysis

File	Format
HVACie2013.ifc	IFC templates

4.8.6.4.3 Non-Applicable Entity Exclusion Analysis

Namespace	Definition
IfcKernel	IfcActor
	IfcComplexPropertyTemplate
	IfcContext
	IfcControl
	IfcGroup
	IfcObject
	IfcObjectDefinition
	IfcPreDefinedPropertySet
	IfcProcess
	IfcProduct
	IfcProject
	IfcProjectLibrary
	IfcPropertyDefinition
	IfcPropertySet
	IfcPropertySetDefinition
	IfcPropertySetTemplate
	IfcPropertyTemplate
	IfcPropertyTemplateDefinition
	IfcProxy
	IfcQuantitySet
	IfcRelAggregates
	IfcRelAssigns
	IfcRelAssignsToActor
	IfcRelAssignsToControl
	IfcRelAssignsToGroup
	IfcRelAssignsToGroupByFactor
	IfcRelAssignsToProcess
	IfcRelAssignsToProduct
	IfcRelAssignsToResource
	IfcRelAssociates
	IfcRelAssociatesClassification
	IfcRelAssociatesDocument
	IfcRelAssociatesLibrary
	IfcRelationship
	IfcRelConnects
	IfcRelDeclares

	IfcRelDecomposes
	IfcRelDefines
	IfcRelDefinesByObject
	IfcRelDefinesByProperties
	IfcRelDefinesByTemplate
	IfcRelDefinesByType
	IfcRelNests
	IfcResource
	IfcRoot
	IfcSimplePropertyTemplate
	IfcTypeObject
	IfcTypeProcess
	IfcTypeProduct
	IfcTypeResource
	IfcPropertySetDefinitionSet
	IfcComplexPropertyTemplateTypeEnum
	IfcObjectTypeEnum
	IfcPropertySetTemplateTypeEnum
	IfcSimplePropertyTemplateTypeEnum
	IfcDefinitionSelect
	IfcProcessSelect
	IfcProductSelect
	IfcPropertySetDefinitionSelect
	IfcResourceSelect
IfcControlExtension	IfcPerformanceHistory
	IfcRelAssociatesApproval
	IfcRelAssociatesConstraint
	IfcPerformanceHistoryTypeEnum
IfcProcessExtension	IfcEvent
	IfcEventType
	IfcProcedure
	IfcProcedureType
	IfcRelSequence
	IfcTask
	IfcTaskType
	IfcWorkCalendar
	IfcWorkControl
	IfcWorkPlan
	IfcWorkSchedule

	IfcEventTriggerTypeEnum
	IfcEventTypeEnum
	IfcProcedureTypeEnum
	IfcSequenceEnum
	IfcTaskTypeEnum
	IfcWorkCalendarTypeEnum
	IfcWorkPlanTypeEnum
	IfcWorkScheduleTypeEnum
IfcProductExtension	IfcAnnotation
	IfcBuilding
	IfcBuildingElement
	IfcBuildingElementType
	IfcBuildingStorey
	IfcCivilElement
	IfcCivilElementType
	IfcDistributionElement
	IfcDistributionElementType
	IfcElement
	IfcElementAssembly
	IfcElementAssemblyType
	IfcElementQuantity
	IfcElementType
	IfcExternalSpatialElement
	IfcExternalSpatialStructureElement
	IfcFeatureElement
	IfcFeatureElementAddition
	IfcFeatureElementSubtraction
	IfcFurnishingElement
	IfcFurnishingElementType
	IfcGeographicElement
	IfcGeographicElementType
	IfcGrid
	IfcOpeningElement
	IfcOpeningStandardCase
	IfcPort
	IfcProjectionElement
	IfcRelAssociatesMaterial
	IfcRelConnectsElements
	IfcRelConnectsPorts

	IfcRelConnectsPortToElement
	IfcRelConnectsWithRealizingElements
	IfcRelContainedInSpatialStructure
	IfcRelFillsElement
	IfcRelInterferesElements
	IfcRelProjectsElement
	IfcRelReferencedInSpatialStructure
	IfcRelServicesBuildings
	IfcRelSpaceBoundary
	IfcRelSpaceBoundary1stLevel
	IfcRelSpaceBoundary2ndLevel
	IfcRelVoidsElement
	IfcSite
	IfcSpace
	IfcSpaceType
	IfcSpatialElement
	IfcSpatialElementType
	IfcSpatialStructureElement
	IfcSpatialStructureElementType
	IfcSpatialZone
	IfcSpatialZoneType
	IfcSystem
	IfcTransportElement
	IfcTransportElementType
	IfcVirtualElement
	IfcZone
	IfcAssemblyPlaceEnum
	IfcElementAssemblyTypeEnum
	IfcElementCompositionEnum
	IfcExternalSpatialElementTypeEnum
	IfcGeographicElementTypeEnum
	IfcGridTypeEnum
	IfcInternalOrExternalEnum
	IfcOpeningElementTypeEnum
	IfcPhysicalOrVirtualEnum
	IfcProjectionElementTypeEnum
	IfcSpaceTypeEnum
	IfcSpatialZoneTypeEnum
	IfcTransportElementTypeEnum

	IfcSpaceBoundarySelect
IfcSharedBldgElements	IfcBeam
	IfcBeamStandardCase
	IfcBeamType
	IfcBuildingElementProxy
	IfcBuildingElementProxyType
	IfcBuildingSystem
	IfcChimney
	IfcChimneyType
	IfcColumn
	IfcColumnStandardCase
	IfcColumnType
	IfcCovering
	IfcCoveringType
	IfcCurtainWall
	IfcCurtainWallType
	IfcDoor
	IfcDoorStandardCase
	IfcDoorType
	IfcMember
	IfcMemberStandardCase
	IfcMemberType
	IfcPlate
	IfcPlateStandardCase
	IfcPlateType
	IfcRailing
	IfcRailingType
	IfcRamp
	IfcRampFlight
	IfcRampFlightType
	IfcRampType
	IfcRelConnectsPathElements
	IfcRelCoversBldgElements
	IfcRelCoversSpaces
	IfcRoof
	IfcRoofType
	IfcShadingDevice
	IfcShadingDeviceType
	IfcSlab

	IfcSlabElementedCase
	IfcSlabStandardCase
	IfcSlabType
	IfcStair
	IfcStairFlight
	IfcStairFlightType
	IfcStairType
	IfcWall
	IfcWallElementedCase
	IfcWallStandardCase
	IfcWallType
	IfcWindow
	IfcWindowStandardCase
	IfcWindowType
	IfcBeamTypeEnum
	IfcBuildingElementProxyTypeEnum
	IfcBuildingSystemTypeEnum
	IfcChimneyTypeEnum
	IfcColumnTypeEnum
	IfcConnectionTypeEnum
	IfcCoveringTypeEnum
	IfcCurtainWallTypeEnum
	IfcDoorTypeEnum
	IfcDoorTypeOperationEnum
	IfcMemberTypeEnum
	IfcPlateTypeEnum
	IfcRailingTypeEnum
	IfcRampFlightTypeEnum
	IfcRampTypeEnum
	IfcRoofTypeEnum
	IfcShadingDeviceTypeEnum
	IfcSlabTypeEnum
	IfcStairFlightTypeEnum
	IfcStairTypeEnum
	IfcWallTypeEnum
	IfcWindowTypeEnum
	IfcWindowTypePartitioningEnum
IfcSharedBldgServiceElements	IfcDistributionChamberElement
	IfcDistributionChamberElementType

	IfcDistributionCircuit
	IfcDistributionControlElement
	IfcDistributionControlElementType
	IfcDistributionFlowElement
	IfcDistributionFlowElementType
	IfcDistributionPort
	IfcDistributionSystem
	IfcEnergyConversionDevice
	IfcEnergyConversionDeviceType
	IfcFlowController
	IfcFlowControllerType
	IfcFlowFitting
	IfcFlowFittingType
	IfcFlowMovingDevice
	IfcFlowMovingDeviceType
	IfcFlowSegment
	IfcFlowSegmentType
	IfcFlowStorageDevice
	IfcFlowStorageDeviceType
	IfcFlowTerminal
	IfcFlowTerminalType
	IfcFlowTreatmentDevice
	IfcFlowTreatmentDeviceType
	IfcRelFlowControlElements
	IfcDistributionChamberElementTypeEnum
	IfcDistributionPortTypeEnum
	IfcDistributionSystemEnum
	IfcFlowDirectionEnum
IfcSharedComponentElements	IfcBuildingElementPart
	IfcBuildingElementPartType
	IfcDiscreteAccessory
	IfcDiscreteAccessoryType
	IfcElementComponent
	IfcElementComponentType
	IfcFastener
	IfcFastenerType
	IfcMechanicalFastener
	IfcMechanicalFastenerType
	IfcBuildingElementPartTypeEnum

	IfcDiscreteAccessoryTypeEnum
	IfcFastenerTypeEnum
	IfcMechanicalFastenerTypeEnum
IfcSharedFacilitiesElements	IfcAsset
	IfcFurniture
	IfcFurnitureType
	IfcInventory
	IfcOccupant
	IfcSystemFurnitureElement
	IfcSystemFurnitureElementType
	IfcFurnitureTypeEnum
	IfcInventoryTypeEnum
	IfcOccupantTypeEnum
	IfcSystemFurnitureElementTypeEnum
IfcSharedMgmtElements	IfcActionRequest
	IfcCostItem
	IfcCostSchedule
	IfcPermit
	IfcProjectOrder
	IfcActionRequestTypeEnum
	IfcCostItemTypeEnum
	IfcCostScheduleTypeEnum
	IfcPermitTypeEnum
	IfcProjectOrderTypeEnum
IfcArchitectureDomain	IfcDoorLiningProperties
	IfcDoorPanelProperties
	IfcDoorStyle
	IfcPermeableCoveringProperties
	IfcWindowLiningProperties
	IfcWindowPanelProperties
	IfcWindowStyle
	IfcDoorPanelOperationEnum
	IfcDoorPanelPositionEnum
	IfcDoorStyleConstructionEnum
	IfcDoorStyleOperationEnum
	IfcPermeableCoveringOperationEnum
	IfcWindowPanelOperationEnum
	IfcWindowPanelPositionEnum
	IfcWindowStyleConstructionEnum

	IfcWindowStyleOperationEnum
IfcBuildingControlsDomain	IfcActuator
	IfcActuatorType
	IfcAlarm
	IfcAlarmType
	IfcController
	IfcControllerType
	IfcFlowInstrument
	IfcFlowInstrumentType
	IfcSensor
	IfcSensorType
	IfcUnitaryControlElement
	IfcUnitaryControlElementType
	IfcActuatorTypeEnum
	IfcAlarmTypeEnum
	IfcControllerTypeEnum
	IfcFlowInstrumentTypeEnum
	IfcSensorTypeEnum
	IfcUnitaryControlElementTypeEnum
IfcConstructionMgmtDomain	IfcConstructionEquipmentResource
	IfcConstructionEquipmentResourceType
	IfcConstructionMaterialResource
	IfcConstructionMaterialResourceType
	IfcConstructionProductResource
	IfcConstructionProductResourceType
	IfcConstructionResource
	IfcConstructionResourceType
	IfcCrewResource
	IfcCrewResourceType
	IfcLaborResource
	IfcLaborResourceType
	IfcSubContractResource
	IfcSubContractResourceType
	IfcConstructionEquipmentResourceTypeEnum
	IfcConstructionMaterialResourceTypeEnum
	IfcConstructionProductResourceTypeEnum
	IfcCrewResourceTypeEnum
	IfcLaborResourceTypeEnum
	IfcSubContractResourceTypeEnum

IfcElectricalDomain	IfcAudioVisualAppliance
	IfcAudioVisualApplianceType
	IfcCableCarrierFitting
	IfcCableCarrierFittingType
	IfcCableCarrierSegment
	IfcCableCarrierSegmentType
	IfcCableFitting
	IfcCableFittingType
	IfcCableSegment
	IfcCableSegmentType
	IfcCommunicationsAppliance
	IfcCommunicationsApplianceType
	IfcElectricAppliance
	IfcElectricApplianceType
	IfcElectricDistributionBoard
	IfcElectricDistributionBoardType
	IfcElectricFlowStorageDevice
	IfcElectricFlowStorageDeviceType
	IfcElectricGenerator
	IfcElectricGeneratorType
	IfcElectricMotor
	IfcElectricMotorType
	IfcElectricTimeControl
	IfcElectricTimeControlType
	IfcJunctionBox
	IfcJunctionBoxType
	IfcLamp
	IfcLampType
	IfcLightFixture
	IfcLightFixtureType
	IfcMotorConnection
	IfcMotorConnectionType
	IfcOutlet
	IfcOutletType
	IfcProtectiveDevice
	IfcProtectiveDeviceTrippingUnit
	IfcProtectiveDeviceTrippingUnitType
	IfcProtectiveDeviceType
	IfcSolarDevice

	IfcSolarDeviceType
	IfcSwitchingDevice
	IfcSwitchingDeviceType
	IfcTransformer
	IfcTransformerType
	IfcAudioVisualApplianceTypeEnum
	IfcCableCarrierFittingTypeEnum
	IfcCableCarrierSegmentTypeEnum
	IfcCableFittingTypeEnum
	IfcCableSegmentTypeEnum
	IfcCommunicationsApplianceTypeEnum
	IfcElectricApplianceTypeEnum
	IfcElectricDistributionBoardTypeEnum
	IfcElectricFlowStorageDeviceTypeEnum
	IfcElectricGeneratorTypeEnum
	IfcElectricMotorTypeEnum
	IfcElectricTimeControlTypeEnum
	IfcJunctionBoxTypeEnum
	IfcLampTypeEnum
	IfcLightFixtureTypeEnum
	IfcMotorConnectionTypeEnum
	IfcOutletTypeEnum
	IfcProtectiveDeviceTrippingUnitTypeEnum
	IfcProtectiveDeviceTypeEnum
	IfcSolarDeviceTypeEnum
	IfcSwitchingDeviceTypeEnum
	IfcTransformerTypeEnum
IfcHvacDomain	**IfcAirTerminal**
	IfcAirTerminalBox
	IfcAirTerminalBoxType
	IfcAirTerminalType
	IfcAirToAirHeatRecovery
	IfcAirToAirHeatRecoveryType
	IfcBoiler
	IfcBoilerType
	IfcBurner
	IfcBurnerType
	IfcChiller
	IfcChillerType

	IfcCoil
	IfcCoilType
	IfcCompressor
	IfcCompressorType
	IfcCondenser
	IfcCondenserType
	IfcCooledBeam
	IfcCooledBeamType
	IfcCoolingTower
	IfcCoolingTowerType
	IfcDamper
	IfcDamperType
	IfcDuctFitting
	IfcDuctFittingType
	IfcDuctSegment
	IfcDuctSegmentType
	IfcDuctSilencer
	IfcDuctSilencerType
	IfcEngine
	IfcEngineType
	IfcEvaporativeCooler
	IfcEvaporativeCoolerType
	IfcEvaporator
	IfcEvaporatorType
	IfcFan
	IfcFanType
	IfcFilter
	IfcFilterType
	IfcFlowMeter
	IfcFlowMeterType
	IfcHeatExchanger
	IfcHeatExchangerType
	IfcHumidifier
	IfcHumidifierType
	IfcMedicalDevice
	IfcMedicalDeviceType
	IfcPipeFitting
	IfcPipeFittingType
	IfcPipeSegment

	IfcPipeSegmentType
	IfcPump
	IfcPumpType
	IfcSpaceHeater
	IfcSpaceHeaterType
	IfcTank
	IfcTankType
	IfcTubeBundle
	IfcTubeBundleType
	IfcUnitaryEquipment
	IfcUnitaryEquipmentType
	IfcValve
	IfcValveType
	IfcVibrationIsolator
	IfcVibrationIsolatorType
	IfcAirTerminalBoxTypeEnum
	IfcAirTerminalTypeEnum
	IfcAirToAirHeatRecoveryTypeEnum
	IfcBoilerTypeEnum
	IfcBurnerTypeEnum
	IfcChillerTypeEnum
	IfcCoilTypeEnum
	IfcCompressorTypeEnum
	IfcCondenserTypeEnum
	IfcCooledBeamTypeEnum
	IfcCoolingTowerTypeEnum
	IfcDamperTypeEnum
	IfcDuctFittingTypeEnum
	IfcDuctSegmentTypeEnum
	IfcDuctSilencerTypeEnum
	IfcEngineTypeEnum
	IfcEvaporativeCoolerTypeEnum
	IfcEvaporatorTypeEnum
	IfcFanTypeEnum
	IfcFilterTypeEnum
	IfcFlowMeterTypeEnum
	IfcHeatExchangerTypeEnum
	IfcHumidifierTypeEnum
	IfcMedicalDeviceTypeEnum

	IfcPipeFittingTypeEnum
	IfcPipeSegmentTypeEnum
	IfcPumpTypeEnum
	IfcSpaceHeaterTypeEnum
	IfcTankTypeEnum
	IfcTubeBundleTypeEnum
	IfcUnitaryEquipmentTypeEnum
	IfcValveTypeEnum
	IfcVibrationIsolatorTypeEnum
IfcPlumbingFireProtectionDomain	IfcFireSuppressionTerminal
	IfcFireSuppressionTerminalType
	IfcInterceptor
	IfcInterceptorType
	IfcSanitaryTerminal
	IfcSanitaryTerminalType
	IfcStackTerminal
	IfcStackTerminalType
	IfcWasteTerminal
	IfcWasteTerminalType
	IfcFireSuppressionTerminalTypeEnum
	IfcInterceptorTypeEnum
	IfcSanitaryTerminalTypeEnum
	IfcStackTerminalTypeEnum
	IfcWasteTerminalTypeEnum
IfcStructuralAnalysisDomain	IfcRelConnectsStructuralActivity
	IfcRelConnectsStructuralMember
	IfcRelConnectsWithEccentricity
	IfcStructuralAction
	IfcStructuralActivity
	IfcStructuralAnalysisModel
	IfcStructuralConnection
	IfcStructuralCurveAction
	IfcStructuralCurveConnection
	IfcStructuralCurveMember
	IfcStructuralCurveMemberVarying
	IfcStructuralCurveReaction
	IfcStructuralItem
	IfcStructuralLinearAction
	IfcStructuralLoadCase

	IfcStructuralLoadGroup
	IfcStructuralMember
	IfcStructuralPlanarAction
	IfcStructuralPointAction
	IfcStructuralPointConnection
	IfcStructuralPointReaction
	IfcStructuralReaction
	IfcStructuralResultGroup
	IfcStructuralSurfaceAction
	IfcStructuralSurfaceConnection
	IfcStructuralSurfaceMember
	IfcStructuralSurfaceMemberVarying
	IfcStructuralSurfaceReaction
	IfcActionSourceTypeEnum
	IfcActionTypeEnum
	IfcAnalysisModelTypeEnum
	IfcAnalysisTheoryTypeEnum
	IfcLoadGroupTypeEnum
	IfcProjectedOrTrueLengthEnum
	IfcStructuralCurveActivityTypeEnum
	IfcStructuralCurveMemberTypeEnum
	IfcStructuralSurfaceActivityTypeEnum
	IfcStructuralSurfaceMemberTypeEnum
	IfcStructuralActivityAssignmentSelect
IfcStructuralElementsDomain	IfcFooting
	IfcFootingType
	IfcPile
	IfcPileType
	IfcReinforcementDefinitionProperties
	IfcReinforcingBar
	IfcReinforcingBarType
	IfcReinforcingElement
	IfcReinforcingElementType
	IfcReinforcingMesh
	IfcReinforcingMeshType
	IfcSurfaceFeature
	IfcTendon
	IfcTendonAnchor
	IfcTendonAnchorType

	IfcTendonType
	IfcVoidingFeature
	IfcFootingTypeEnum
	IfcPileConstructionEnum
	IfcPileTypeEnum
	IfcReinforcingBarTypeEnum
	IfcReinforcingMeshTypeEnum
	IfcSurfaceFeatureTypeEnum
	IfcTendonAnchorTypeEnum
	IfcTendonTypeEnum
	IfcVoidingFeatureTypeEnum
	IfcBendingParameterSelect
IfcActorResource	IfcActorRole
	IfcAddress
	IfcOrganization
	IfcOrganizationRelationship
	IfcPerson
	IfcPersonAndOrganization
	IfcPostalAddress
	IfcTelecomAddress
	IfcAddressTypeEnum
	IfcRoleEnum
	IfcActorSelect
IfcApprovalResource	IfcApproval
	IfcApprovalRelationship
	IfcResourceApprovalRelationship
IfcConstraintResource	IfcConstraint
	IfcMetric
	IfcObjective
	IfcReference
	IfcResourceConstraintRelationship
	IfcBenchmarkEnum
	IfcConstraintEnum
	IfcLogicalOperatorEnum
	IfcObjectiveEnum
	IfcMetricValueSelect
IfcCostResource	IfcAppliedValue
	IfcCostValue
	IfcCurrencyRelationship

	IfcArithmeticOperatorEnum
	IfcAppliedValueSelect
IfcDateTimeResource	IfcEventTime
	IfcIrregularTimeSeries
	IfcIrregularTimeSeriesValue
	IfcLagTime
	IfcRecurrencePattern
	IfcRegularTimeSeries
	IfcResourceTime
	IfcSchedulingTime
	IfcTaskTime
	IfcTaskTimeRecurring
	IfcTimePeriod
	IfcTimeSeries
	IfcTimeSeriesValue
	IfcWorkTime
	IfcDate
	IfcDateTime
	IfcDayInMonthNumber
	IfcDayInWeekNumber
	IfcDuration
	IfcMonthInYearNumber
	IfcTime
	IfcTimeStamp
	IfcDataOriginEnum
	IfcRecurrenceTypeEnum
	IfcTaskDurationEnum
	IfcTimeSeriesDataTypeEnum
	IfcTimeOrRatioSelect
IfcExternalReferenceResource	IfcClassification
	IfcClassificationReference
	IfcDocumentInformation
	IfcDocumentInformationRelationship
	IfcDocumentReference
	IfcExternalInformation
	IfcExternalReference
	IfcExternalReferenceRelationship
	IfcLibraryInformation
	IfcLibraryReference

	IfcResourceLevelRelationship
	IfcLanguageId
	IfcURIReference
	IfcDocumentConfidentialityEnum
	IfcDocumentStatusEnum
	IfcClassificationReferenceSelect
	IfcClassificationSelect
	IfcDocumentSelect
	IfcLibrarySelect
	IfcResourceObjectSelect
IfcGeometricConstraintResource	IfcConnectionCurveGeometry
	IfcConnectionGeometry
	IfcConnectionPointEccentricity
	IfcConnectionPointGeometry
	IfcConnectionSurfaceGeometry
	IfcConnectionVolumeGeometry
	IfcGridAxis
	IfcGridPlacement
	IfcLocalPlacement
	IfcObjectPlacement
	IfcVirtualGridIntersection
	IfcCurveOrEdgeCurve
	IfcGridPlacementDirectionSelect
	IfcPointOrVertexPoint
	IfcSolidOrShell
	IfcSurfaceOrFaceSurface
IfcGeometricModelResource	IfcAdvancedBrep
	IfcAdvancedBrepWithVoids
	IfcBlock
	IfcBooleanClippingResult
	IfcBooleanResult
	IfcBoundingBox
	IfcBoxedHalfSpace
	IfcCartesianPointList
	IfcCartesianPointList3D
	IfcCsgPrimitive3D
	IfcCsgSolid
	IfcExtrudedAreaSolid
	IfcExtrudedAreaSolidTapered

	IfcFaceBasedSurfaceModel
	IfcFacetedBrep
	IfcFacetedBrepWithVoids
	IfcFixedReferenceSweptAreaSolid
	IfcGeometricCurveSet
	IfcGeometricSet
	IfcHalfSpaceSolid
	IfcManifoldSolidBrep
	IfcPolygonalBoundedHalfSpace
	IfcRectangularPyramid
	IfcRevolvedAreaSolid
	IfcRevolvedAreaSolidTapered
	IfcRightCircularCone
	IfcRightCircularCylinder
	IfcSectionedSpine
	IfcShellBasedSurfaceModel
	IfcSolidModel
	IfcSphere
	IfcSurfaceCurveSweptAreaSolid
	IfcSweptAreaSolid
	IfcSweptDiskSolid
	IfcSweptDiskSolidPolygonal
	IfcTessellatedFaceSet
	IfcTessellatedItem
	IfcTriangulatedFaceSet
	IfcBooleanOperator
	IfcBooleanOperand
	IfcCsgSelect
	IfcGeometricSetSelect
IfcGeometryResource	IfcAxis1Placement
	IfcAxis2Placement2D
	IfcAxis2Placement3D
	IfcBoundaryCurve
	IfcBoundedCurve
	IfcBoundedSurface
	IfcBSplineCurve
	IfcBSplineCurveWithKnots
	IfcBSplineSurface
	IfcBSplineSurfaceWithKnots

	IfcCartesianPoint
	IfcCartesianTransformationOperator
	IfcCartesianTransformationOperator2D
	IfcCartesianTransformationOperator2DnonUniform
	IfcCartesianTransformationOperator3D
	IfcCartesianTransformationOperator3DnonUniform
	IfcCircle
	IfcCompositeCurve
	IfcCompositeCurveOnSurface
	IfcCompositeCurveSegment
	IfcConic
	IfcCurve
	IfcCurveBoundedPlane
	IfcCurveBoundedSurface
	IfcCylindricalSurface
	IfcDirection
	IfcElementarySurface
	IfcEllipse
	IfcGeometricRepresentationItem
	IfcLine
	IfcMappedItem
	IfcOffsetCurve2D
	IfcOffsetCurve3D
	IfcOuterBoundaryCurve
	IfcPcurve
	IfcPlacement
	IfcPlane
	IfcPoint
	IfcPointOnCurve
	IfcPointOnSurface
	IfcPolyline
	IfcRationalBSplineCurveWithKnots
	IfcRationalBSplineSurfaceWithKnots
	IfcRectangularTrimmedSurface
	IfcReparametrisedCompositeCurveSegment
	IfcRepresentationItem
	IfcRepresentationMap
	IfcSurface
	IfcSurfaceOfLinearExtrusion

	IfcSurfaceOfRevolution
	IfcSweptSurface
	IfcTrimmedCurve
	IfcVector
	IfcDimensionCount
	IfcBSplineCurveForm
	IfcBSplineSurfaceForm
	IfcKnotType
	IfcTransitionCode
	IfcTrimmingPreference
	IfcAxis2Placement
	IfcCurveOnSurface
	IfcTrimmingSelect
	IfcVectorOrDirection
IfcMaterialResource	IfcMaterial
	IfcMaterialClassificationRelationship
	IfcMaterialConstituent
	IfcMaterialConstituentSet
	IfcMaterialDefinition
	IfcMaterialLayer
	IfcMaterialLayerSet
	IfcMaterialLayerSetUsage
	IfcMaterialLayerWithOffsets
	IfcMaterialList
	IfcMaterialProfile
	IfcMaterialProfileSet
	IfcMaterialProfileSetUsage
	IfcMaterialProfileSetUsageTapering
	IfcMaterialProfileWithOffsets
	IfcMaterialProperties
	IfcMaterialRelationship
	IfcMaterialUsageDefinition
	IfcCardinalPointReference
	IfcDirectionSenseEnum
	IfcLayerSetDirectionEnum
	IfcMaterialSelect
IfcMeasureResource	IfcContextDependentUnit
	IfcConversionBasedUnit
	IfcConversionBasedUnitWithOffset

	IfcDerivedUnit
	IfcDerivedUnitElement
	IfcDimensionalExponents
	IfcMeasureWithUnit
	IfcMonetaryUnit
	IfcNamedUnit
	IfcSIUnit
	IfcUnitAssignment
	IfcAbsorbedDoseMeasure
	IfcAccelerationMeasure
	IfcAmountOfSubstanceMeasure
	IfcAngularVelocityMeasure
	IfcAreaDensityMeasure
	IfcAreaMeasure
	IfcBoolean
	IfcComplexNumber
	IfcCompoundPlaneAngleMeasure
	IfcContextDependentMeasure
	IfcCountMeasure
	IfcCurvatureMeasure
	IfcDescriptiveMeasure
	IfcDoseEquivalentMeasure
	IfcDynamicViscosityMeasure
	IfcElectricCapacitanceMeasure
	IfcElectricChargeMeasure
	IfcElectricConductanceMeasure
	IfcElectricCurrentMeasure
	IfcElectricResistanceMeasure
	IfcElectricVoltageMeasure
	IfcEnergyMeasure
	IfcForceMeasure
	IfcFrequencyMeasure
	IfcHeatFluxDensityMeasure
	IfcHeatingValueMeasure
	IfcIdentifier
	IfcIlluminanceMeasure
	IfcInductanceMeasure
	IfcInteger
	IfcIntegerCountRateMeasure

	IfcIonConcentrationMeasure
	IfcIsothermalMoistureCapacityMeasure
	IfcKinematicViscosityMeasure
	IfcLabel
	IfcLengthMeasure
	IfcLinearForceMeasure
	IfcLinearMomentMeasure
	IfcLinearStiffnessMeasure
	IfcLinearVelocityMeasure
	IfcLogical
	IfcLuminousFluxMeasure
	IfcLuminousIntensityDistributionMeasure
	IfcLuminousIntensityMeasure
	IfcMagneticFluxDensityMeasure
	IfcMagneticFluxMeasure
	IfcMassDensityMeasure
	IfcMassFlowRateMeasure
	IfcMassMeasure
	IfcMassPerLengthMeasure
	IfcModulusOfElasticityMeasure
	IfcModulusOfLinearSubgradeReactionMeasure
	IfcModulusOfRotationalSubgradeReactionMeasure
	IfcModulusOfSubgradeReactionMeasure
	IfcMoistureDiffusivityMeasure
	IfcMolecularWeightMeasure
	IfcMomentOfInertiaMeasure
	IfcMonetaryMeasure
	IfcNonNegativeLengthMeasure
	IfcNormalisedRatioMeasure
	IfcNumericMeasure
	IfcParameterValue
	IfcPHMeasure
	IfcPlanarForceMeasure
	IfcPlaneAngleMeasure
	IfcPositiveLengthMeasure
	IfcPositivePlaneAngleMeasure
	IfcPositiveRatioMeasure
	IfcPowerMeasure
	IfcPressureMeasure

	IfcRadioActivityMeasure
	IfcRatioMeasure
	IfcReal
	IfcRotationalFrequencyMeasure
	IfcRotationalMassMeasure
	IfcRotationalStiffnessMeasure
	IfcSectionalAreaIntegralMeasure
	IfcSectionModulusMeasure
	IfcShearModulusMeasure
	IfcSolidAngleMeasure
	IfcSoundPowerLevelMeasure
	IfcSoundPowerMeasure
	IfcSoundPressureLevelMeasure
	IfcSoundPressureMeasure
	IfcSpecificHeatCapacityMeasure
	IfcTemperatureGradientMeasure
	IfcTemperatureRateOfChangeMeasure
	IfcText
	IfcThermalAdmittanceMeasure
	IfcThermalConductivityMeasure
	IfcThermalExpansionCoefficientMeasure
	IfcThermalResistanceMeasure
	IfcThermalTransmittanceMeasure
	IfcThermodynamicTemperatureMeasure
	IfcTimeMeasure
	IfcTorqueMeasure
	IfcVaporPermeabilityMeasure
	IfcVolumeMeasure
	IfcVolumetricFlowRateMeasure
	IfcWarpingConstantMeasure
	IfcWarpingMomentMeasure
	IfcDerivedUnitEnum
	IfcSIPrefix
	IfcSIUnitName
	IfcUnitEnum
	IfcDerivedMeasureValue
	IfcMeasureValue
	IfcSimpleValue
	IfcUnit

	IfcValue
IfcPresentationAppearanceResource	IfcBlobTexture
	IfcColourRgb
	IfcColourRgbList
	IfcColourSpecification
	IfcCurveStyle
	IfcCurveStyleFont
	IfcCurveStyleFontAndScaling
	IfcCurveStyleFontPattern
	IfcDraughtingPreDefinedColour
	IfcDraughtingPreDefinedCurveFont
	IfcExternallyDefinedHatchStyle
	IfcExternallyDefinedSurfaceStyle
	IfcExternallyDefinedTextFont
	IfcFillAreaStyle
	IfcFillAreaStyleHatching
	IfcFillAreaStyleTiles
	IfcImageTexture
	IfcIndexedColourMap
	IfcIndexedTextureMap
	IfcIndexedTriangleTextureMap
	IfcPixelTexture
	IfcPreDefinedColour
	IfcPreDefinedCurveFont
	IfcPreDefinedItem
	IfcPreDefinedTextFont
	IfcPresentationStyle
	IfcPresentationStyleAssignment
	IfcStyledItem
	IfcSurfaceStyle
	IfcSurfaceStyleLighting
	IfcSurfaceStyleRefraction
	IfcSurfaceStyleRendering
	IfcSurfaceStyleShading
	IfcSurfaceStyleWithTextures
	IfcSurfaceTexture
	IfcTextStyle
	IfcTextStyleFontModel
	IfcTextStyleForDefinedFont

	IfcTextStyleTextModel
	IfcTextureCoordinate
	IfcTextureCoordinateGenerator
	IfcTextureMap
	IfcTextureVertex
	IfcTextureVertexList
	IfcFontStyle
	IfcFontVariant
	IfcFontWeight
	IfcPresentableText
	IfcSpecularExponent
	IfcSpecularRoughness
	IfcTextAlignment
	IfcTextDecoration
	IfcTextFontName
	IfcTextTransformation
	IfcNullStyle
	IfcReflectanceMethodEnum
	IfcSurfaceSide
	IfcColour
	IfcColourOrFactor
	IfcCurveFontOrScaledCurveFontSelect
	IfcCurveStyleFontSelect
	IfcFillStyleSelect
	IfcHatchLineDistanceSelect
	IfcPresentationStyleSelect
	IfcSizeSelect
	IfcSpecularHighlightSelect
	IfcStyleAssignmentSelect
	IfcSurfaceStyleElementSelect
	IfcTextFontSelect
IfcPresentationDefinitionResource	IfcAnnotationFillArea
	IfcPlanarBox
	IfcPlanarExtent
	IfcPresentationItem
	IfcTextLiteral
	IfcTextLiteralWithExtent
	IfcBoxAlignment
	IfcTextPath

IfcPresentationOrganizationResource	IfcLightDistributionData
	IfcLightIntensityDistribution
	IfcLightSource
	IfcLightSourceAmbient
	IfcLightSourceDirectional
	IfcLightSourceGoniometric
	IfcLightSourcePositional
	IfcLightSourceSpot
	IfcPresentationLayerAssignment
	IfcPresentationLayerWithStyle
	IfcLightDistributionCurveEnum
	IfcLightEmissionSourceEnum
	IfcLayeredItem
	IfcLightDistributionDataSourceSelect
IfcProfileResource	IfcArbitraryClosedProfileDef
	IfcArbitraryOpenProfileDef
	IfcArbitraryProfileDefWithVoids
	IfcAsymmetricIShapeProfileDef
	IfcCenterLineProfileDef
	IfcCircleHollowProfileDef
	IfcCircleProfileDef
	IfcCompositeProfileDef
	IfcCShapeProfileDef
	IfcDerivedProfileDef
	IfcEllipseProfileDef
	IfcIShapeProfileDef
	IfcLShapeProfileDef
	IfcMirroredProfileDef
	IfcParameterizedProfileDef
	IfcProfileDef
	IfcProfileProperties
	IfcRectangleHollowProfileDef
	IfcRectangleProfileDef
	IfcReinforcementBarProperties
	IfcRoundedRectangleProfileDef
	IfcSectionProperties
	IfcSectionReinforcementProperties
	IfcTrapeziumProfileDef
	IfcTShapeProfileDef

	IfcUShapeProfileDef
	IfcZShapeProfileDef
	IfcProfileTypeEnum
	IfcReinforcingBarRoleEnum
	IfcReinforcingBarSurfaceEnum
	IfcSectionTypeEnum
IfcPropertyResource	IfcComplexProperty
	IfcExtendedProperties
	IfcPreDefinedProperties
	IfcProperty
	IfcPropertyAbstraction
	IfcPropertyBoundedValue
	IfcPropertyDependencyRelationship
	IfcPropertyEnumeratedValue
	IfcPropertyEnumeration
	IfcPropertyListValue
	IfcPropertyReferenceValue
	IfcPropertySingleValue
	IfcPropertyTableValue
	IfcSimpleProperty
	IfcCurveInterpolationEnum
	IfcObjectReferenceSelect
IfcQuantityResource	IfcPhysicalComplexQuantity
	IfcPhysicalQuantity
	IfcPhysicalSimpleQuantity
	IfcQuantityArea
	IfcQuantityCount
	IfcQuantityLength
	IfcQuantityTime
	IfcQuantityVolume
	IfcQuantityWeight
IfcRepresentationResource	IfcCoordinateOperation
	IfcCoordinateReferenceSystem
	IfcGeometricRepresentationContext
	IfcGeometricRepresentationSubContext
	IfcMapConversion
	IfcMaterialDefinitionRepresentation
	IfcProductDefinitionShape
	IfcProductRepresentation

	IfcProjectedCRS
	IfcRepresentation
	IfcRepresentationContext
	IfcShapeAspect
	IfcShapeModel
	IfcShapeRepresentation
	IfcStyledRepresentation
	IfcStyleModel
	IfcTopologyRepresentation
	IfcGeometricProjectionEnum
	IfcGlobalOrLocalEnum
	IfcCoordinateReferenceSystemSelect
	IfcProductRepresentationSelect
IfcStructuralLoadResource	IfcBoundaryCondition
	IfcBoundaryEdgeCondition
	IfcBoundaryFaceCondition
	IfcBoundaryNodeCondition
	IfcBoundaryNodeConditionWarping
	IfcFailureConnectionCondition
	IfcSlippageConnectionCondition
	IfcStructuralConnectionCondition
	IfcStructuralLoad
	IfcStructuralLoadConfiguration
	IfcStructuralLoadLinearForce
	IfcStructuralLoadOrResult
	IfcStructuralLoadPlanarForce
	IfcStructuralLoadSingleDisplacement
	IfcStructuralLoadSingleDisplacementDistortion
	IfcStructuralLoadSingleForce
	IfcStructuralLoadSingleForceWarping
	IfcStructuralLoadStatic
	IfcStructuralLoadTemperature
	IfcSurfaceReinforcementArea
	IfcModulusOfRotationalSubgradeReactionSelect
	IfcModulusOfSubgradeReactionSelect
	IfcModulusOfTranslationalSubgradeReactionSelect
	IfcRotationalStiffnessSelect
	IfcTranslationalStiffnessSelect
	IfcWarpingStiffnessSelect

IfcTopologyResource	IfcAdvancedFace
	IfcClosedShell
	IfcConnectedFaceSet
	IfcEdge
	IfcEdgeCurve
	IfcEdgeLoop
	IfcFace
	IfcFaceBound
	IfcFaceOuterBound
	IfcFaceSurface
	IfcLoop
	IfcOpenShell
	IfcOrientedEdge
	IfcPath
	IfcPolyLoop
	IfcSubedge
	IfcTopologicalRepresentationItem
	IfcVertex
	IfcVertexLoop
	IfcVertexPoint
	IfcShell
IfcUtilityResource	IfcApplication
	IfcOwnerHistory
	IfcTable
	IfcTableColumn
	IfcTableRow
	IfcGloballyUniqueId
	IfcChangeActionEnum
	IfcStateEnum

4.8.7 Conformance Testing Procedures

4.8.7.1 Format and Content Requirements

4.8.7.1.1. Test Rule List

The list of test rules is defined in the mvdXML file referenced herein.

4.8.7.1.2. Test Rule Definition

The definitions of test rules are included in the mvdXML file referenced herein.

4.8.7.1.3 Test Rules Formatting

Formatting documentation for MVDXML is available at http://www.buildingsmart-tech.org/specifications/mvd-overview/mvd-overview-summary.

4.8.7.1.4 Test Rule Coverage Analysis

Coverage of test rules for a given IFC file may be evaluated by using the mvdXML file referenced herein with the IfcDoc tool available at http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-beta-summary..

4.8.7.2 Examples and Mapping Requirements

4.8.7.2.1 Example File List

The list of example files is available at http://www.nibs.org/?page=bsa_commonbimfiles.

4.8.7.2.2 Example File Description

File descriptions are available at the website identified.

4.8.7.2.3 Common BIM File Reuse

Common BIM files are re-used at the website identified.

4.8.7.2.4 Implementers' Agreements

Implementers agreements are available at http://www.buildingsmart-tech.org/implementation/ifc-implementation/ifc-impl-agreements/ifc-impl-agreements-summary.

4.8.7.2.5 Transformations/Mapping Allowed

Transformations are defined in the MVDXML file referenced herein.

4.8.7.2.6 Transformation/Mapping Documentation

Transformation documentation for MVDXML is available at http://www.buildingsmart-tech.org/specifications/mvd-overview/mvd-overview-summary.

4.8.7.3 Testing Tools and Procedures

4.8.7.3.1 Testing Tool List

IFCDOC is a Windows application that provides functionality for validating files against model view definitions, as well as authoring model view definitions. It is published by BuildingSMART International Ltd and is freely available at http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-beta-summary

4.8.7.3.2 Testing Tool Algorithm

The algorithm for testing files is shown in the following C# source code for IFCDOC.

4.8.7.3.2.1 Model View Validation Algorithm

The core algorithm iterates through selected model views, iterates through concept roots (applying to an entity), finds all instances of the applicable entity, iterates through concepts on each entity, and validates each concept.

```
// iterate through each concept root
foreach (DocModelView docView in this.m_project.ModelViews)
{
  if (docView.Visible)
  {
    foreach (DocConceptRoot docRoot in docView.ConceptRoots)
    {
      Type typeEntity = null;
      if (typemap.TryGetValue(docRoot.ApplicableEntity.Name.ToUpper(), out typeEntity))
      {
        // build list of instances
        List<SEntity> list = new List<SEntity>();
        foreach (SEntity instance in format.Instances.Values)
        {
          if (typeEntity.IsInstanceOfType(instance))
          {
            list.Add(instance);
          }
        }

        foreach (DocTemplateUsage docUsage in docRoot.Concepts)
        {

          bool eachresult = true; // assume passing unless something fails

          // if no template parameters defined, then evaluate generically
          if (docUsage.Items.Count == 0)
          {
            int fail = 0;
            int pass = 0;
            foreach (SEntity ent in list)
            {
              // check with parameters plugged in
              bool? result = true;
              foreach (DocModelRule rule in docUsage.Definition.Rules)
              {
                result = rule.Validate(ent, null, typemap);
                if (result != null && !result.Value)
                  break;
              }

              if (result == null)
              {
                // no applicable rules, so passing
                pass++;
              }
              else if (result != null && result.Value)
              {
                // all rules passed
                pass++;
              }
              else
              {
                fail++;
              }
            }
          }

          foreach (DocTemplateItem docItem in docUsage.Items)
          {
            int pass = 0;
```

```
                int fail = 0;
                foreach (SEntity ent in list)
                {
                  // check with parameters plugged in
                  bool? result = true;
                  foreach (DocModelRule rule in docUsage.Definition.Rules)
                  {
                    result = rule.Validate(ent, docItem, typemap);
                    if (result != null && !result.Value)
                      break;
                  }

                  if (result == null)
                  {
                    // inapplicable; passes
                    pass++;
                  }
                  else if (result != null && result.Value)
                  {
                    // applicable and valid; passes
                    pass++;
                  }
                  else
                  {
                    fail++;
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

4.8.7.3.2.2 Attribute Rule Validation Algorithm

These support routines validate attribute rules.

```
    /// <summary>
    /// Validates an object to meet rule.
    /// </summary>
    /// <param name="target">Required instance to validate.</param>
    /// <param name="docItem">Optional template parameters to use for validation.</param>
    /// <param name="typemap">Map of types to resolve.</param>
    /// <returns></returns>
    public override bool? Validate(object target, DocTemplateItem docItem, Dictionary<string, Type> typemap)
    {
      if (target == null)
        return false;

      // (1) check if field is defined on target object; if not, then this rule does not apply.
      FieldInfo fieldinfo = target.GetType().GetField(this.Name);
      if (fieldinfo == null)
        return false;

      // (2) extract the value
      object value = fieldinfo.GetValue(target); // may be null

      if (value is System.Collections.IList)
      {
        System.Collections.IList list = (System.Collections.IList)value;
        int pass = 0;
        int fail = 0;
        foreach (object o in list)
        {
```

```
            bool? result = ValidateItem(o, docItem, typemap);
            if (result != null)
            {
               if (result.Value)
               {
                  pass++;
               }
               else
               {
                  fail++;
               }
            }
         }

         if (this.CardinalityMin == 0 && this.CardinalityMax == 0)
         {
            return (pass == 0);
         }
         else if (this.CardinalityMin == 0 && this.CardinalityMax == 1)
         {
            return (pass == 0 || pass == 1);
         }
         else if (this.CardinalityMin == 1 && this.CardinalityMax == 1)
         {
            return (pass == 1);
         }
         else if (this.CardinalityMin == 1)
         {
            return (fail == 0);
         }
         else
         {
            return true;
         }
      }
      else
      {
         // validate single
         return ValidateItem(value, docItem, typemap);
      }

   }


   /// <summary>
   /// Checks a value to see if it matches the parameter value.
   /// </summary>
   /// <param name="value"></param>
   /// <param name="docItem"></param>
   /// <param name="typemap"></param>
   /// <returns>True if passing, False if failing, or Null if inapplicable.</returns>
   private bool? ValidateItem(object value, DocTemplateItem docItem, Dictionary<string, Type> typemap)
   {
      // (3) if parameter is defined, check for match
      if (!String.IsNullOrEmpty(this.Identification))
      {
         if (docItem == null)
            return true; // parameter must be specified in order to check this rule

         string match = docItem.GetParameterValue(this.Identification);
         if (value == null && String.IsNullOrEmpty(match))
         {
            return true;
         }
         else if (value is SEntity)
         {
```

```
            if (match != null && value.GetType().Name.Equals(match))
            {
               return true;
            }
            else
            {
               return false;
            }
         }
         else if (value != null)
         {
            // pull out internal value type
            FieldInfo fieldinfo = value.GetType().GetField("Value");
            if (fieldinfo != null)
            {
               object innervalue = fieldinfo.GetValue(value);
               if (innervalue == null)
               {
                  return false;
               }
               else if (match != null && innervalue.ToString().Equals(match.ToString(), StringComparison.Ordinal))
               {
                  return true;
               }
               else if (this.IsCondition())
               {
                  // condition didn't match, so chain of rules does not apply -- return null.
                  return null;
               }
               else
               {
                  // constraint evaluated to false and conditioned applied.
                  return false;
               }
            }
            else
            {
               return false;
            }
         }
         else
         {
            return false;
         }
      }

      // (4) recurse through constraints or entity rules
      if (this.Rules != null && this.Rules.Count > 0)
      {
         foreach (DocModelRule rule in this.Rules)
         {
            // attribute rule is true if at least one entity filter matches or one constraint filter matches
            bool? result = rule.Validate(value, docItem, typemap);
            if (result != null && result.Value)
               return result;
         }

         return false;
      }

      return true;
   }
```

4.8.7.3.2.3 Entity Rule Validation Algorithm

These support routines validate entity rules.

```
/// <summary>
/// Validates rules for an entity.
/// </summary>
/// <param name="target">Required object to validate.</param>
/// <param name="docItem">Template item to validate.</param>
/// <param name="typemap">Map of type names to type definitions.</param>
/// <returns>True if passing, False if failing, or Null if inapplicable.</returns>
public override bool? Validate(object target, DocTemplateItem docItem, Dictionary<string, Type> typemap)
{
  // checking for matching cast
  Type t = null;
  if (!typemap.TryGetValue(this.Name.ToUpper(), out t))
    return false;

  if (!t.IsInstanceOfType(target))
    return false;

  if (target is SEntity)
  {
    foreach (DocModelRule rule in this.Rules)
    {
      bool? result = rule.Validate((SEntity)target, docItem, typemap);

      // entity rule is inapplicable if any attribute rules are inapplicable
      if (result == null)
        return null;

      // entity rule fails if any attribute rules fail
      if (!result.Value)
        return false;
    }
  }

  return true;
}
```

4.8.7.3.3 Testing Tools Sample Files

Sample IFC files are available at the following URL: http://www.nibs.org/?page=bsa_commonbimfiles

4.8.7.3.4 Testing Tool Software Availability

The IFCDOC tool for testing is available at the following URL: http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-beta-summary9

4.8.8 Implementation Resources

4.8.8.1 Implementation Resources list

4.8.8.1.1 Implementation Guides

Implementation guides are available at buildingsmart-tech.org/downloads.

4.8.8.2 Implementation Resources Completeness

4.8.8.2.1 Workflow Coverage Methodology

Implementation resources for various BIM platform workflows are available at http://www.buildingsmart-tech.org/implementation. Additional platforms and workflows may be added as indicated on the web page.

4.8.8.2.2 Workflow Coverage Analysis

A map of software applications, platforms, and supported model views is available at http://www.buildingsmart-tech.org/implementation/implementations.

4.8.9 Revision Plans

4.8.9.1 Revision Plans List

4.8.9.1.1 Revision Management Process

No revisions to this version 1.0 standard have been identified at this time. Revisions will be identified, evaluated and introduced based on initial uses of LCie exchanges that will begin as part of 2015 Alliance Challenge events.

4.8.9.1.2 Revision Management Notification

Revisions will be proposed through a new LinkedIn LCie Group.

4.8.9.2 Proposed Revision Deployment Methods

4.8.9.2.1 Revision Management Process

No revisions to this version 1.0 standard have been identified at this time. Revisions will be identified, evaluated and introduced based on initial uses of LCie exchanges that will begin as part of 2015 Alliance Challenge events.

4.8.9.2.2 Revision Management Notification

Revisions will be proposed through a new LinkedIn LCie Group.

Annex A

The following electronic formats are attached herein:

HVACie2013.exp – schema in EXPRESS format
HVACie2013.xsd – schema in XSD format
HVACie2013.mdxml – model view definition
HVACie2013.ifc – property set templates

Bibliography

ISO 639-1, Codes for the representation of names of languages — Part 1: Alpha-2 code

ISO 639-2, Codes for the representation of names of languages — Part 2: Alpha-3 code

ISO 639-3, Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages

ISO 6707-1, Building and civil engineering — Vocabulary — Part 1: General terms

ISO 8601, Data elements and interchange formats — Information Exchange — Representation of dates and times.

ISO 10303-1:1994, Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles

ISO 10303-11, Industrial automation systems and integration — Product data representation and exchange — Part 11: description methods: The EXPRESS Language Reference Manual

ISO 10303-21, Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure

ISO 10303-28, Industrial automation systems and integration — Product data representation and exchange — Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas

ISO 10303-41, Product data representation and exchange — Integrated generic resource — Fundamentals of product description and support

ISO 10303-42, Product data representation and exchange — Integrated generic resource — Geometric and topological representation

ISO 10303-43, Product data representation and exchange — Integrated generic resource — Representation structures

ISO 10303-46, Product data representation and exchange — Integrated generic resource — Visual presentation

ISO 10303-514, Product data representation and exchange — Application interpreted construct — Advanced boundary representation

ISO 12006-3, Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information

ISO/IEC 8824-1, Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation.

ISO/IEC 14772-1, Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding

ISO/IEC 19775-1, Information technology — Computer graphics and image processing — Extensible 3D (X3D) — Part 1: Architecture and base components

ISO/IEC 81346-12, Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 12: Buildings and building services

CSS-1, Cascading Style Sheets, level 1 — W3C Recommendation

XML Schema Part 2, XML Schema Part 2: Datatypes — W3C Recommendation

RFC 3986, Uniform Resource Identifier (URI): Generic Syntax — Network Working Group NWG Standard

RFC 5646, Tags for Identifying Languages — Internet Engineering Task Force IETF Best Current Practice 47