**To convert an .IFC file to multiple .CSV files, NIST IFC File Analyzer style**

**C2V-IfcClassList.txt**

A list of ifc classes to convert, from **.IFC** to **.CSV**.

It has two fields (no heading though): the IfcClass and its generalized type. There are different versions for **IFC2x3** and **IFC4** files.

They are read into an array:

// ifc classes to convert from ifc to csv

var ifcClassItemList : array of ifcClassItem;

of the types

TifcField = record

fieldname: string;

fieldtype : string; [ text, enumerated, numeric, Boolean, cvalue, id, idlist, cidlist ]

fieldcontent : string; values stored for an instance during conversion

end;

TifcInstance = record

Instance\_Id : string;

Instance\_rec : string;

end;

// ifc classes to convert from ifc to csv

ifcClassItem = record

IfcClass: string; // member class

GeneralIfcClass: string; // get field from

fields: array of TifcField;

instances: array of TifcInstance;

end;

Fields’ names and types are read from the corresponding C2V-<GeneralIfcClass>.txt files.

For any building elements (and their types or styles) in the list, the generalized structures are described in the **IfcBldgElement.txt** and **IfcBldgElementType.txt** files. In the case of an **IFC2x3** model version, Windows and Doors are associated to a style rather than a type, in the file **IfcDoorOrWindowStyle.txt**.

For other elements, where a generalized form is not provided in the input file, that classes’ data structure is unique. In this case, the classes’ field structure will populate its **fields** array (as described further). When a generalized form is provided, the classes’ field structure will be read from the corresponding generalized class field list.

Classes that are not of use to this application are not listed in **IfcClassList.txt** and will be skipped when the IFC file is converted to IFC.

**IfcProject – IfcBldgElement – IfcSpace – IfcZone.txt**

Each file contains the set of fields that occur within the corresponding ifc class or classes represented by it.

They have two or mode fields, depending on its data type. Field of type **id** isn’t the actual[[1]](#footnote-1) **id** of an instance of that class, but the **id** of a referenced class. Table 1 shows ifc classes that are referenced by classes listed in **C2V-IfcClassList.txt**.

Names of classes that are referenced but not listed in the above cited file are exhibited in gray. Meaning that, if not found, they will receive null values, i.e., no value between parenthesis.

Table 1 – ifc classes that are referenced by other ifc classes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **IfcOwner History** | **IfcGeometric Representation Context** | **IfcUnit Assignment** | **IfcGrid Placement  IfcLocal Placement** | **IfcMaterial Definition Representation  IfcProduct Definition Shape** | **IfcPostal Address** | **( IfcProperty SetDefinition )** | **IfcRepresentation Map** | **IfcClassification** | **( IfcPhysical Quantity )** | **IfcComplex Property  IfcProperty Single Value** | **IfcProject  IfcSite  IfcBuilding  IfcBuilding Storey  IfcBldg Element** | **IfcSite IfcBuilding  IfcBuilding Storey  IfcSpace IfcMember** | **IfcGroup IfcZone** | **IfcClassification Reference** | **IfcBldg Element** | **IfcBuilding Storey** | **IfcElement Quantity  IfcProperty Set** | **IfcBldg Element Type  IfcSpace Type** |
| **IfcProject** | Owner History | Representation Contexts | UnitsIn Context |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **IfcSite** | Owner History |  |  | Object Placement | Representation | Site Adress |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **IfcBuilding** | Owner History |  |  | Object Placement | Representation | Building Address |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **IfcSpace** | Owner History |  |  | Object Placement | Representation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **IfcSpaceType** | Owner History |  |  |  |  |  | HasProperty Sets | Representation Maps |  |  |  |  |  |  |  |  |  |  |  |
| **IfcZone** | Owner History |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **IfcBuildingStorey** | Owner History |  |  | Object Placement | Representation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **IfcClassificationReference** |  |  |  |  |  |  |  |  | Referenced Source |  |  |  |  |  |  |  |  |  |  |
| **IfcElementQuantity** | Owner History |  |  |  |  |  |  |  |  | Quantities |  |  |  |  |  |  |  |  |  |
| **IfcPropertySet** | Owner History |  |  |  |  |  |  |  |  |  | Has Properties |  |  |  |  |  |  |  |  |
| **IfcRelAggregates** | Owner History |  |  |  |  |  |  |  |  |  |  | Relating Object | Related Objects |  |  |  |  |  |  |
| **IfcRelAssignsToGroup** | Owner History |  |  |  |  |  |  |  |  |  |  |  | Related Objects | Relating Group |  |  |  |  |  |
| **IfcRelAssociatesClassification** | Owner History |  |  |  |  |  |  |  |  |  |  |  | Related Objects |  | Relating Classification |  |  |  |  |
| **IfcRelContainedInSpatialStructure** | Owner History |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Related Elements | Relating Structure |  |  |
| **IfcRelDefinesByProperties** | Owner History |  |  |  |  |  |  |  |  |  |  |  | Related Objects |  |  |  |  | Relating Property Definition |  |
| **IfcRelDefinesByType** | Owner History |  |  |  |  |  |  |  |  |  |  |  | Related Objects |  |  |  |  |  | Relating Type |
| **IfcBldgElement** | Owner History |  |  | Object Placement | Representation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **IfcBldgElementType** | Owner History |  |  |  |  |  | HasProperty Sets | Representation Maps |  |  |  |  |  |  |  |  |  |  |  |
| **IfcDoorOrWindowStyle** | Owner History |  |  |  |  |  | HasProperty Sets | Representation Maps |  |  |  |  |  |  |  |  |  |  |  |

Fields of type **text** are enclosed in quotes, e.g., 'Pset\_WindowCommon'

They are coded in the same way in the CSV file

Fields of type **enumerated** and **boolean** are enclosed in periods, e.g., .ELEMENT. , .T.

They are coded in the same way in the CSV file

Fields of type **numeric** are not enclosed, e.g., 1.9712

They are coded in the same way in the CSV file

Fields of type **cvalue** can be of any of the above types, but they are enclosed in parenthesis, following a SELECT IfcValue, e.g.,

IFCIDENTIFIER('Concreto Armado - Estrutural 190 x 360')  
IFCLABEL('')  
IFCBOOLEAN(.T.)  
IFCTEXT('363.897 (kgCO\X2\2082\X0\)’)  
IFCENERGYMEASURE(3503.63411157)

Only what’s inside the parenthesis is coded in the CSV File, e.g.,

'Concreto Armado - Estrutural 190 x 360'

.T.

'363.897 (kgCO\X2\2082\X0\)’

3503.63411157

**Note**: Quotes take precedence over any other symbol – everything inside is text, including what would be separators, e.g., parenthesis and commas.

Fields of the type **id** are preceded by a hash, e.g., #28

They are coded without the hash in the CSV file, e.g., 28, following the class name, as follows:

… class 28 …

Fields of the type **idlist** and **cidlist** are enclosed in parenthesis, e.g. (#143) (#278,#279,#280)

A referenced class of type **cidlist** is a generalization of several classes. For example, the class **IfcPhysicalQuantity** refrenced in the field **Quantities** of the **IfcElementQuantity** class, can refer to any of a number of subclasses, as shown on Table 2. When the IFC file is read, the actual class being referenced will be determined indirectly by its **id**.

For **m** distinct subclass occurrences of the referenced class, NIST IFC File Analyzer would produce an output of each classes’ **nm** instances’ ids, as follows:

… (n1) class1 id1.1 [ id1.2 … id1.n1 ] [ … (nm) classm idm.1 [ idm.2 … idm.nm ] ] …

Fields of the type **idlist**, despite referencing a single ifc class, may also actually refer to one of its subclasses, for example, one of the subclasses of **IfcPropertySetDefinition**.

Also, there is the case, when the class can be either itself or a subclass, for example, **IfcGeometricRepresentationContext** and its subclass **IfcGeometricRepresentationSubContext**, as shown on Table 3

Table 2 – some referenced abstract classes and their subclasses

|  |  |
| --- | --- |
| **Abstract class** | **Subclasses actually referenced** |
| IfcPropertySetDefinition | IfcPropertySet, IfcElementQuantity, IfcDoorLiningProperties, IfcDoorPanelProperties, IfcPermeableCoveringProperties, IfcReinforcementDefinitionProperties, IfcWindowLiningProperties, IfcWindowPanelProperties |
| IfcPhysicalQuantity | IfcQuantityArea, IfcQuantityCount, IfcQuantityLength, IfcQuantityTime, IfcQuantityVolume, IfcQuantityWeight, IfcPhysicalComplexQuantity |

Table 3 – some classes that have specialization(s)

|  |  |
| --- | --- |
| **Referenced class** | **Subclasse(s)** |
| IfcGeometricRepresentationContext | IfcGeometricRepresentationSubContext |
| IfcRelAssignsToGroup | IfcRelAssignsToGroupByFactor |

Note: except for **boolean**, **enumerated** and **cvalue**, empty values (null) take the form **$** or **\***.

They are coded as null in the CSV file, i.e., no value between commas.

Examples:

**IFC**

#119= IFCPROJECT('0yOsZLRTrE8g8RJluhuq53',#41,'0001',$,$,'Nome do projeto','Status do projeto',(#111),#106);

**C2V-IfcProject.txt**

GlobalId text

OwnerHistory id

Name text

Description text

ObjectType text

LongName text

Phase text

RepresentationContexts idlist

UnitsInContext id

**CSV (NIST IFC File Analyser)**

IfcProject (1)

ID,GlobalId,OwnerHistory,Name,Description,ObjectType,LongName,Phase,RepresentationContexts,UnitsInContext

119,0yOsZLRTrE8g8RJluhuq53,IfcOwnerHistory 41,0001,,,Nome do projeto,Status do projeto,(1) IfcGeometricRepresentationContext 111,IfcUnitAssignment 106

IfcAnalyser only puts strings inside quotes when needed.

**CSV generated by PQTO**

IfcProject (1)

ID,GlobalId,OwnerHistory,Name,Description,ObjectType,LongName,Phase,RepresentationContexts,UnitsInContext

119,0yOsZLRTrE8g8RJluhuq53,,0001,,,Nome do projeto,Status do projeto,,

OwneHistory, RepresentationContexts, and UnitsInContext are not listed in the C2V-IfcClasses.txt file, so they receive no value.

1. Classes’ own ids are a common required property and thus its addition to the field list does not yield any information. [↑](#footnote-ref-1)