

PQTO Reference Guide

Release 1.15

July 6th, 2025

1. **PQTO.exe** is a quantity takeoff program, originally for **IFC 2x3** files, developed by Marcelo E. Giacaglia, for academic use only. Professional use is not supported nor recommended, and at your own risk.

- 1.1 the program does this in two stages: (1) from an IFC file to a set of CSV files, one for each Ifc Class; and (2) from the set of CSV files to quantities for building spaces and elements.
- 1.2 Quantity takeoff for **IFC 4** is provided from this release on, although not guaranteed by some BIM authoring software vendors. There is a version of the software configuration files for each IFC version, which can be downloaded separately (even though the executable program is the same).

2. Some features are hard-coded in the software, while others can be customized, as described, as follows.

2.1 What is hard-coded in Stage 1: conversion of IFC into a set of CSV files

File sets listed in Table 1 must exist.

Table 1 – Required files

QTO-Accents.txt	At least one line must exist in this file, e.g., \\X2\\2082\\X0\\ ₂ 2 (refer to 2.1.1)
C2V-IfcClassList.txt as well as any file listed: C2V-<ifcClass>.txt	(refer to 2.1.2 through 2.1.5)

2.1.1 In an IFC (STEP format) file, special characters are coded as escaped Unicode.

PQTO has a functionality to convert from escaped Unicode to HTML coding, then to the actual ANSI coding, on the first stage of its execution.

2.1.2 C2V-...txt files must be placed in the subfolder named **ifc2csv**.

2.1.3 **C2V-IfcClassList.txt** lists IFC Classes that will be converted from the IFC BIModel STEP format file into a CSV file in the output set. Each line in the file represents an IFC Class and can have one or two words. The first is the actual Ifc Class name. The second exits were a Ifc Class is a subtype of a general class, and, thus inherits its data fields description.

For example: in the line, IfcBeam IfcBldgElement, IfcBeam is a particular Ifc Class and IfcBldgElement is a generalization of several Ifc classes. The purpose of such definition is to eliminate the need of a C2V-<ifcClass>.txt file for every class that shares the same fields.

2.1.4 Fields descriptions in each C2V-<ifcClass>.txt files consist of its name and one of the following types:

- text: a character string enclosed in single quotes
- numeric: numeric values, mostly floating-point values
- boolean: an enumeration of, either, .T. or .F.
- cvalue: any of the above, preceded by a Ifc data type, and enclosed in parenthesis
e.g. IfcLabel, IfcText, IfcAreaMeasure, IfcLengthMeasure, IfcVolumeMeasure
- enumerated: a set of terms enclosed in periods (as for the Boolean type)
- id: reference to an instance of another Ifc Class
- idlist: a list of references to instances of another Ifc Class
- cidlist: a list of references to instances of more than one Ifc Class
- latlon: latitude or longitude data, enclosed in parenthesis

2.1.5 Field descriptions do not include the classes' instance IDs, which are always present. All other fields must be listed.

2.2 What is configurable (and to what extent) in Stage 1

2.2.1 Ifc classes not listed in the C2V-IfcClassList.txt will not yield a CSV file, and will not be referenced in the id, idlist or cidlist fields of any CSV file generated. The inclusion of a Ifc Class in this file, and the corresponding C2V-<ifcClass>.txt, doesn't guarantee that data for that class in the IFC file will generate quantity takeoff data. A minimum requirement is the existence of a corresponding QTO-<ifcClass>.txt for that class as well as other entries in other configuration files and the capacity of PQTO to read and process such data. Such inclusion should work for building element classes and their types.

2.2.2 In an IFC (STEP format) file, special characters are coded as escaped Unicode.

PQTO has a functionality to convert from escaped Unicode to HTML, then to the actual ANSI coding on the first stage of its execution.

2.2.2.1 Currently, the QTO-Accents.txt is set is for Brazilian Portuguese.

2.2.2.2 \X2\2082\X0\ and ₂ are codes for subscript two, as in CO₂;

2.2.2.3 Other elements can be added, for other languages, as well as modification to the existing elements, for specific Code Pages;

2.2.2.4 Compound terms must come before their individual letters (ÇÃ comes before Ç or Ã);

2.3 What is hard-coded in Stage 2: Quantity Takeoff from the set of CSV files

File sets listed in Table 2 must exist.

IfcProject.ifc IfcSite.csv IfcBuilding.csv IfcBuildingStorey.csv IfcSpace.csv as well as their configuration files: QTO-<ifcClass>.txt	
IfcRelAggregates.csv IfcRelContainedInSpatialStructure.csv IfcRelDefinesByProperties.csv IfcRelDefinesByType.csv as well as their configuration files: QTO-<ifcClass>.txt	
QTO-col-ifcBldgElementType.txt QTO-col-ifcBldgElementQuantityArea.txt QTO-col-ifcBldgElementQuantityLength.txt QTO-col-ifcBldgElementQuantityVolume.txt QTO-col-BldgElementCommon.txt	(Refer to 2.4.5.1) (Refer to 2.4.5.2) (Refer to 2.4.5.3) (Refer to 2.4.5.4) (Refer to 2.4.5.8)
QTO-ifcElementQuantity.txt	
QTO-BldgElementClasses.txt	(Refer to 2.4.2)
QTO-col-IfcZone.txt QTO-col-SpaceType.txt QTO-col-IfcSpaceQuantityLength.txt QTO-col-IfcSpaceQuantityVolume.txt QTO-col-IfcSpaceQuantityArea.txt QTO-col-SpaceCommon.txt	(Refer to 2.4.6.1) (Refer to 2.4.6.2) (Refer to 2.4.6.3) (Refer to 2.4.6.4) (Refer to 2.4.6.5) (Refer to 2.4.6.8)
QTO-ifcQuantityArea.txt QTO-ifcQuantityLength.txt QTO-ifcQuantityVolume.txt	
ifcQuantityArea.csv ifcQuantityLength.csv ifcQuantityVolume.csv ifcElementQuantity.csv	(Refer to 2.4.4.1) (Refer to 2.4.4.1) (Refer to 2.4.4.1) (Refer to 2.4.4.1)
QTO-col-ifcBldgElementQuantitySynonyms.txt QTO-col-ifcSpaceQuantitySynonyms.txt	(Refer to 2.3.3) (Refer to 2.3.3)
IfcRelAssociatesClassification.csv ifcClassificationReference.csv QTO-ifcClassificationReference.txt	(Refer to 2.4.4.2) (Refer to 2.4.4.2)

QTO-col-IfcBldgElementClassification.txt QTO-col-IfcSpaceClassification.txt	(Refer to 2.3.3 and 2.4.8) (Refer to 2.4.8)
QTO-PSetsToRead.txt QTO-ifcPropertySingleValue.txt IfcPropertySet.csv fcRelDefinesByProperties.csv IfcPropertySingleValue.csv	
Either QTO-IfcZone.txt QTO-IfcRelAssignsToGroup.txt IfcZone.csv IfcRelAssignsToGroup.csv Or IfcSpaceType.csv (*1) QTO-IfcSpaceType.txt	as in ArchiCAD (Graphisoft)

(*1) Refer to 2.3.2

2.3.1 Configuration files, such as QTO-<ifcClass>.txt and QTO-col-<ifcClass>.txt, must reside in the same folder as PQTO.exe

And all data files, i.e., *.csv files, must reside in the same folder as the selected project for building element and space data extraction, ifcProject.csv

2.3.2 IfcSpaceType will only be read if IfcZone.csv and IfcRelAssignsToGroup.csv are not generated from the IFC model exported by the BIM Software. Otherwise, even if present, the file will be skipped as well as its contents.

2.3.3 **Important:** text files can be in UTF-8 or ANSI coding, but if there are accented letters, the file must be saved in ANSI coding, otherwise accented characters will not be interpreted correctly and the software will either not return the desired results or even crash.

QTO-col-ifcBldgElementQuantitySynonyms.txt and QTO-col-ifcSpaceQuantitySynonyms.txt, currently carry Brazilian Portuguese synonyms for English property names exported by some BIM software in Brazilian language implementations. These must be saved in ANSI.

The same for QTO-Accents.txt and QTO-col-IfcBldgElementClassification.txt.

2.4 What is configurable (and to what extent) in Stage 2

2.4.1 (void)

2.4.2 Building elements are those listed in the QTO-BldgElementClasses.txt, along with their Type or Style, and a flag that indicates that only one style or type file is to be read (when there are classes that share the same type).

2.4.2.1 A warning will be included on the on-screen data input Log, for any file listed in QTO-BldgElementClasses.txt and not present in the csv file set;

2.4.2.2 Other building elements can be added to the list file, provided their structure is roughly the same as the above listed elements;

2.4.2.3 Two or more building element classes can share the same type, e.g., IfcWallType is the same for IfcWall and, its specialization, IfcWallStandardCase;

2.4.3 Configuration files QTO-<ifcClass>.txt list the fields in the corresponding <ifcClass>.csv file; Field names are preceded by two parameters: Field used and Field type, separated by a blank character

2.4.3.1 Field used is either: T or F, respectively True or False;

2.4.3.2 Field types are listed in Table 3.

Table 3 – Field Types

XX	a field that will not (and must not) be processed for any purpose
ID	a unique identifier for a row - necessary for data processing within PQTO, but will never be displayed or output by the program
VC	will be displayed as a column heading, and their corresponding values on subsequent rows
1L	reference to another <ifcClass> and a single corresponding ID - necessary for association mapping between <ifcClass>es
mL	reference to another <ifcClass> and a set of corresponding IDs
sL	reference to a set of <ifcClasses> and each one's set of corresponding IDs
AT	content is not a value but a field name for a corresponding value, marked as of type VA
VA	will be displayed as a value for the column heading field defined as of type AT

2.4.3.3 example, for QTO-IfcSite.txt:

T ID ID
F VC GlobalId
F XX OwnerHistory
T VC Name
T VC Description
F VC ObjectType
F XX ObjectPlacement
F XX Representation
T VC LongName
F VC CompositionType
F VC RefLatitude
F VC RefLongitude
F VC RefElevation
F VC LandTitleNumber
F VC SiteAddress

2.4.3.4 example, for QTO-IfcRelAggregates.txt:

F ID ID
F XX GlobalId
F XX OwnerHistory
F XX Name
F XX Description
T 1L RelatingObject
T mL RelatedObjects

2.4.3.5 example, for QTO-IfcSlab.txt:

T ID ID
F VC GlobalId
F XX OwnerHistory
T VC Name
T VC Description
F VC ObjectType
F XX ObjectPlacement
XX Representation
F VC Tag
T VC PredefinedType

2.4.3.6 example, for QTO-ifcQuantityArea.txt:

T ID ID

T AT Name

F VC Description

F VC Unit

T VA AreaValue

2.4.3.7 Fields of the type ID, 1L, mL, sL, AT and VA cannot be changed on the provided configuration files -if changed, the software won't function properly;

2.4.3.8 Fields of the type XX cannot be changed, and their field used value must remain as F;

2.4.3.9 Fields whose field used value is marked as F, and having a field type VC, could be set to T; but shouldn't be changed, because the software wasn't tested extensively for this, and output files would have a great number of columns that would be either empty or not relevant;

2.4.3.10 Field order in the configuration file corresponds to physical order on the corresponding csv file and must not be altered;

If altered, the software won't function properly;

2.4.4 Exceptions (existence is not mandatory)

2.4.4.1 While configuration files QTO-IfcElementQuantity.txt, QTO-ifcQuantityArea.txt, QTO-ifcQuantityLength.txt, and QTO-ifcQuantityVolume.txt are mandatory, Lack of any of the corresponding IfcElementQuantity.csv, IfcQuantityArea.csv, IfcQuantityLength.csv, IfcQuantityVolume.csv data files, will produce a warning on-screen data input Log, and quantities for building spaces and elements takeoffs may be generated, but not from these files (but, from ifcPropertySet.csv and ifcPropertySingleValue.csv files);

2.4.4.2 While configuration file QTO-ifcClassificationReference.txt is mandatory,

Lack of IfcRelAssociatesClassification.csv and fcClassificationReference.csv data files, will produce a warning on-screen data input Log, and information classification data may be generated, but not from these files (but, from ifcPropertySet.csv and ifcPropertySingleValue.csv files – not yet implemented);

For non-English BIM software versions some property values may be written in its native language. Current software package is provided with the English version and the Brazilian Portuguese version, to be saved as QTO-ifcClassificationReference.txt:

- QTO-col-IfcBldgElementClassification - US.txt
- QTO-col-IfcBldgElementClassification - BR.txt

2.4.5 Configuration files QTO-col-IfcBldgElementQuantity<measurement>.txt and QTO-col-BldgElementType.txt, lists fields that present on quantity takeoff listings, even though not displayed on screen grids.

2.4.5.1 Type data fields for building elements, listed in QTO-col-IfcBldgElementType.txt, are:

Name
PredefinedType
ConstructionType
OperationType

2.4.5.2 Area quantity fields for building elements, listed in QTO-col-IfcBldgElementQuantityArea.txt, are:

Area
GrossArea
NetArea
TotalArea
GrossFootPrintArea
NetFootPrintArea
ProjectedArea
GrossSideArea
NetSideArea
NetSurfaceAreaExtrudedSides
CrossSectionArea
GrossSurfaceArea
NetSurfaceArea
OuterSurfaceArea
TotalSurfaceArea

2.4.5.3 Length quantity fields for building elements, listed in QTO-col-IfcBldgElementQuantityLength.txt, are:

Depth
Diameter
Length
Height
Width
Perimeter
Thickness

2.4.5.4 Volume quantity fields for building elements, listed in QTO-col-IfcBldgElementQuantityVolume.txt, are:

Volume
GrossVolume
NetVolume

2.4.5.5 Fields for quantity measurements can be added to each list or removed, provided there is at least one;

2.4.5.6 The above listed Field names for quantity measurements correspond to values of AT field types, as defined in the configuration files:

QTO-ifcQuantityArea.txt, QTO-ifcQuantityLength.txt, and QTO-ifcQuantityVolume.txt;

2.4.5.7 Some building element files carry the same (or it could be similar) field(s) in their corresponding Type or Style data file;

There is a recommendation that the redundant class field only be used when its corresponding type is not defined or has no value attributed to it;

PQTO.exe will display both fields, as originally configured (see examples below), regardless the case;

2.4.5.7.1 If desired, suppression of an element type attribute in its own data file, by changes to the corresponding configuration file, e.g., from T VC to F VC, for slabs:

T ID ID
F VC GlobalId
F XX OwnerHistory
T VC Name
T VC Description
F VC ObjectType
F XX ObjectPlacement
F XX Representation
F VC Tag
F VC PredefinedType (originally T VC PredefinedType)

2.4.5.7.2 The opposite (although not recommended) can be done in IfcSlabType.txt file, by editing settings for PredefinedType, e.g., from T VC to F VC:

T ID ID
F VC GlobalId
F XX OwnerHistory
T VC Name
F VC Description
F XX ApplicableOccurrence
T sL HasPropertySets
F XX RepresentationMaps
F VC Tag
F VC ElementType
F VC PredefinedType (originally T VC PredefinedType)

2.4.5.8 Building elements' Common properties, listed in QTO-col-BldgElementCommon.txt, for IFC 2x3 STEP files, are:

IsExternal
Loadbearing

2.4.5.9 Weight and ComplexPhysical quantity measurements are not currently supported;

2.4.5.10 Listing of materials for building elements, and quantities and materials for parts of building elements are not currently supported.

2.4.6 Configuration files QTO-col-IfcSpaceQuantityArea.txt, QTO-col-IfcSpaceQuantityLength.txt, QTO-col-IfcSpaceQuantityVolume.txt and QTO-col-SpaceElementType.txt, lists fields that are not displayed on the screen grid previews, but will be present on quantity takeoff listings

2.4.6.1 Zone data field for building spaces, listed in QTO-col-IfcZone.txt

2.4.6.2 Type data field for building spaces, listed in QTO-col-SpaceType.txt, is:

Name

2.4.6.3 Examples of Area quantity fields for building spaces, listed in QTO-col-SpaceQuantityArea.txt, are:

Area (for ifcPropertySet*)

GrossFloorArea

NetFloorArea

GrossCeilingArea

NetCeilingArea

GrossWallArea

NetWallArea

2.4.6.4 Examples of Length quantity fields for building spaces, listed in QTO-col-SpaceQuantityLength.txt, are:

Height

ClearHeight

ComputationHeight (for ifcPropertySet*)

NominalHeight

FinishCeilingHeight

FinishFloorHeight

GrossPerimeter

Unbounded Height (for ifcPropertySet*)

Perimiter (for ifcPropertySet*)

GrossPerimeter

NetPerimeter

*Refer to 2.4.7

2.4.6.5 Examples of Volume quantity fields for building spaces, listed in QTO-col-SpaceQuantityVolume.txt, are:

Volume (for ifcPropertySet*)

GrossVolume

NetVolume

*Refer to 2.4.7

2.4.6.6 Fields for quantity measurements can be added to each list or removed, provided there is at least one;

2.4.6.7 The above listed Field names for quantity measurements correspond to values of AT field types, as defined in the configuration files:

QTO-ifcQuantityArea.txt, QTO-ifcQuantityLength.txt, and QTO-ifcQuantityVolume.txt.

2.4.6.8 Spaces' Common properties, listed in QTO-col-SpaceCommon.txt, for IFC 4 STEP files, is currently (and is empty for IFC 2x3 STEP files):

IsExternal

2.4.7 There are quantities that can /should be extracted from ifcPropertySet, depending on the BIM authoring software used. The property sets to look for are listed in the file **QTO-PSetsToRead.txt**, for example:

Dimensions	Pset_RampFlightCommon
BaseQuantities	PSet_RoofCommon
Other	Pset_SlabCommon
Data	Pset_SpaceCommon
Identity Data	Pset_StairCommon
Phasing	Pset_StairFlightCommon
Constraints	Pset_WallCommon
Cotas	Pset_WindowCommon
Outros	Qto_BeamBaseQuantities
Dados	Qto_BuildingElementProxyQuantities
Dados de identidade	Qto_ChimneyBaseQuantities
Fases	Qto_ColumnBaseQuantities
ArchiCADProperties	Qto_CurtainWallBaseQuantities
AC_Pset_RenovationAndPhasing	Qto_DoorBaseQuantities
Pset_BeamCommon	Qto_FootingBaseQuantities
Pset_BuildingElementProxyCommon	Qto_RailingBaseQuantities
Pset_ChimneyCommon	Qto_RampBaseQuantities
Pset_ColumnCommon	Qto_RampFlightBaseQuantities
Pset_CurtainWallCommon	Qto_RoofBaseQuantities
Pset_DoorCommon	Qto_SlabBaseQuantities
Pset_FlowSegmentDuctSegment	Qto_SpaceBaseQuantities
Pset_FlowSegmentPipeSegment	Qto_StairBaseQuantities
Pset_FootingCommon	Qto_StairFlightBaseQuantities
Pset_QuantityTakeOff	Qto_WallBaseQuantities
Pset_RailingCommon	Qto_WindowBaseQuantities
Pset_RampCommon	

2.4.7.1 Note that for foreign languages installations (in this case Portuguese), terminology will vary, as shown above: 'Cotas' ≡ 'Dimensions'; 'Outros' ≡ 'Other'; 'Dados' ≡ 'Data'; 'Dados de identidade' ≡ 'Identity Data'; and 'Phasing' ≡ 'Fases';

*2.4.7.2 Only single value properties listed in:

- QTO-col-IfcBldgElementQuantityArea.txt
- QTO-col-IfcBldgElementQuantityLength.txt
- QTO-col-IfcBldgElementQuantityVolume.txt
- QTO-col-BuildingElementCommon.txt
- QTO-col-SpaceQuantityArea.txt
- QTO-col-SpaceQuantityLength.txt
- QTO-col-SpaceQuantityVolume.txt

, will be accounted for;

2.4.8 QTO-col-IfcSpaceClassification.txt contents are context dependable

For example, if the BIM software is Revit (from AutoDesk) and Omni Class Table 13 are to be used then, QTO-col-IfcSpaceClassification - OmniClass.txt is to be copied as QTO-col-IfcSpaceClassification.txt.

Otherwise, Table 4A from the Brazilian standard NBR-15.965 has been used, whatever BIM software, QTO-col-IfcSpaceClassification.txt is the same as QTO-col-IfcSpaceClassification - NBR.txt.

Note: due to antonyms in Revit (from Autodesk) implementations, Omniclass table 13 and NBR 15.965 table 4A cannot be used concurrently (as can be inferred from the comparison of the contents of both file versions).

Appendix A – Ifc Classes and Relationships among classes (processed in PQTO)

The IFC STEP format file is converted in stage one as a collection of tables, as in the Relational Database Model. Step one will create a table for each class in the C2V-IfcClassList.txt configuration file, e.g., Project, Site, Building, Building-Storey, Space, and, for each Building element class. These files contain attribute values for instances within each class.

There are however, attributes that are common to many instances as well as relationships among instances of different classes, e.g., the building storey of a space. Any class whose name begins with 'IfcRel' are relationship classes, but there are others that do not.

A list with a brief description of the relationship classes processed in PQTO follows.

IfcRelAggregates

For elements within an aggregation¹. Examples are the sites that make up a Project, the buildings and spaces that make up a Site, the spaces that make up a Building, as well as any part of a building element that is also modeled as a building element.

```
IfcProject <----> IfcSite  
IfcSite <----> IfcBuilding  
IfcBuilding <----> IfcBuildingStorey  
IfcBuildingStorey <----> IfcSpace
```

`Ifc<BldgElement> <----> (Ifc<BldgElement>)`

IfcRelAssignsToGroup

For elements within an aggregation. Examples are the association of a building space to a zone in a zoning system.

```
IfcSpace <<----> IfcZone
```

IfcRelAssociatesClassification

For associations of an element to a classification system, such as Omniclass (US), Uniclass (UK), ABNT NBR 15.965 (Brazil).

```
IfcClassificationReference <----> Ifc<BldgElement>  
IfcClassificationReference <----> IfcSpace
```

¹ This should be actually named as a Composition (not an Aggregation), as a collection of parts that cannot exist independently of the whole.

IfcRelContainedInSpatialStructure

For elements contained in a spatial structure, e.g., the building storey of reference of each building element.

`(Ifc<BldgElement>) <----> IfcBuildingStorey`

IfcRelDefinesByProperties

For sets of common properties and their values for instances of elements of a class.

`IfcSpace <----> IfcPropertySet
IfcSpace <----> IfcElementQuantity

Ifc<BldgElement> <----> IfcPropertySet
Ifc<BldgElement> <----> IfcElementQuantity

IfcSite <----> IfcPropertySet
IfcBuilding <----> IfcPropertySet
IfcBuildingStorey <----> IfcPropertySet`

IfcElementQuantity

Used for base quantities of type Area, Length, Volume (among others).

`IfcElementQuantity <----> IfcQuantityArea
IfcElementQuantity <----> IfcQuantityLength
IfcElementQuantity <----> IfcQuantityVolume
IfcElementQuantity <----> IfcPhysicalComplexQuantity`

IfcPropertySet

Used for properties defined through property sets, that are common to a class or to the type of that class.

`IfcPropertySet <----> IfcPropertySingleValue`

IfcRelDefinesByType

The type of building elements or spaces. The association of an element instance to a type instance is a means for generalization of property values shared by a subset of instances of a class.

`IfcSpace <----> IfcSpaceType
Ifc<BldgElement> <----> Ifc<BldgElement>Type | Style`

On step two, PQTO will navigate through these relationships to explicitly indicate these associations in the basic quantities' extraction tables, one for building spaces instances and the other for building elements instances.