

# **IFC 2x Reference Guide**

*Draft – NFP*

v1.6g  
Graphisoft, Sep 2003

# Introduction

## About this Guide

This IFC Reference Guide provides technical explanations of the operations of the IFC 2x Add-Ons for ArchiCAD. For IFC 1.51 and 2.00 see the **IFC 1.51 & 2.00 Reference Guide**

## IFC Add-On

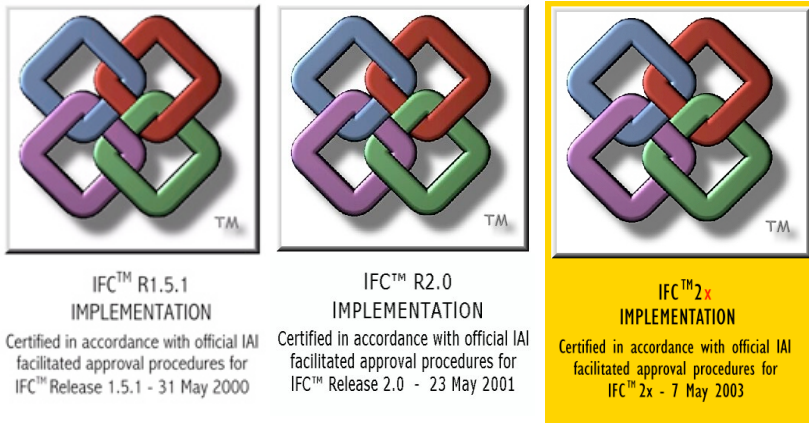
IFC import/export Add-Ons are available for ArchiCAD releases 6.5, 7.0 and 8.0.

The IFC Add-Ons must be in the directory: [ArchiCAD folder]\Add-Ones\Import-Export.

**Note::** For Macintosh users on OS 9.x and earlier, the Add-On uses memory comparable to approximately 2-4 times the size of the IFC file.

## Version Support and Releases

ArchiCAD is certified as follows:



For latest update on IFC capabilities refer to the Graphisoft website <http://www.Graphisoft.com/support/IFC>

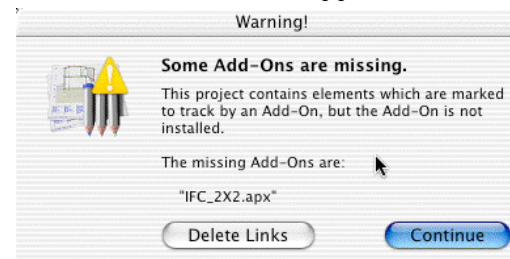
Current availability is:

IFC Version Certification	Add-On		
	Macintosh ( In/Out)	Windows (.apx)	
AC 6.5	AC 7.0	AC 8.0	
<b>1.5.1</b> 21 May 2000	IFC_151	IFC_151	IFC_151 IFC_Common
<b>2.0</b> 25 May 2001	IFC_200	IFC_200	IFC_200 IFC_Common
<b>2x</b> 7 May 2003	NA	IFC_2x	IFC_2x IFC_Common <sup>1</sup>

Table 1: IFC Add-Ons

## Installing IFC Add-ons

In ArchiCAD 8 there is a new method of managing IFC add-ons. In this case you must install IFC Common which now manages the choice of the appropriate IFC version. All IFC version add-ons are located in the Add-ons folder of the ArchiCAD application folder.



ArchiCAD checks for IFC Add-ons and warns you if it is not installed.

### Delete Links

Remove the IFC data related to the missing Add-on. XXX

### Continue

Leave the IFC data in the database.

## IFC2x Add-ons for ArchiCAD 7.0 and 8.x

There are some differences between the 2x add-ons for 7.0 and 8.0 due to

<sup>1</sup> For ArchiCAD 8, A new Add-on Common.apx is required with the all IFC version Add-ons.

ArchiCAD 8.0's improved features and interface.  
The differences are highlighted in the text.

# The IFC Integrated Object Model

## What is an IFC?

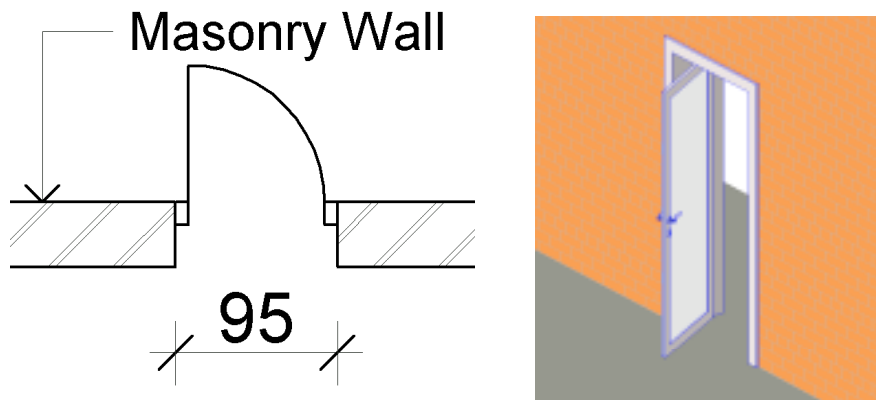
IFC stands for “Industry Foundation Classes”, the set of internationally standardized object definitions for use in the Construction Industry developed by the International Alliance for Interoperability (IAI).

The business objective of the IAI, through its chapters established in some 11 countries around the world (refer <http://iaiweb.lbl.gov/>) is:

To integrate the AEC/FM industry by specifying Industry Foundation Classes (IFC) as a universal language to improve the communication, productivity, delivery time, cost, and quality throughout the design, construction, operation and maintenance life cycle

## Inter-operability Definition

The IFC concept is based on the idea of objects (or *elements* in ArchiCAD terms) brought together in an integrated model (ArchiCAD’s *virtual building*). These objects are defined to support the whole lifecycle of facility development from inception through design, documentation and construction,

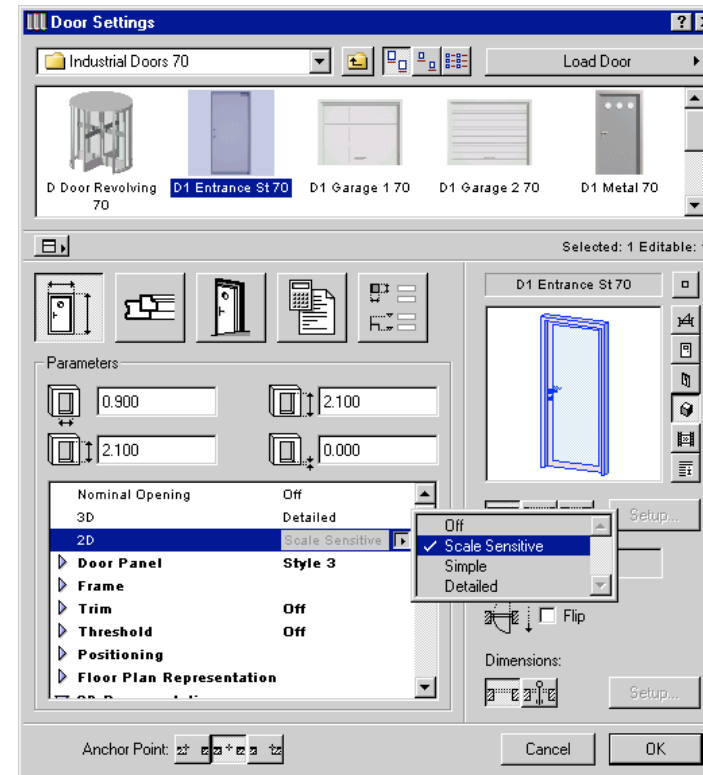


then facility management and finally demolition and or disposal. IFC is to “Project Model” exchange (wall, door, window) what DXF is to graphic entity exchange (line, arc, circle). IFC is available to all participants in the Construction Industry, for use globally, including use by all construction industry software vendors. IFC offers a higher-level “common language” for the sharing of intelligent objects between disciplines across the building lifecycle

## Why use IFC?

In each project, the CAD Manager, with the Project Leader and the party with whom the exchange is to be made, should review why information needs to be shared, what is to be exchanged and how.

Why would you choose IFCs instead of traditional techniques such as 2D files in DWG/DXF or DGN format?



The principal benefit of IFCs is their object description – not only does the IFC protocol preserve the full geometric description in 3D, but it also knows its location and relationships, as well as all the properties (or parameters) of each object such as finish, serial number, material description, thermal conductance, cost etc. ter, b) objects) intelligent data Evolution of 2D graphics to Object integrated models

For ArchiCAD users this offers great potential to access the accurate geometry of building service systems and structural elements, i.e. to improve the information to be coordinated in the model, rather than relying on the

incomplete functions of current 2D CAD.

**Tip:** Note that Graphisoft's GDL object browser technology also allows information to be saved and transmitted in IFC format. See **Accessing GDL Product Data** below

Always remember that a combination of techniques may be more useful than sticking to one solution only. Choose what suits the project resources, skills available and the specific task in hand.

## Planning an IFC Exchange

The first action is to determine that your partner and his software application has an IFC translator. Having validated this, the next important step is to select an appropriate version that both partners can support and understand the functionality offered by that version. ArchiCAD's support of the IFC protocol offers the following IFC compatibility:

IFC Version	ArchiCAD Release	Notes
1.51	6.5, 7.0	The first release of the IFC translator. This version supported the concept of building carcass.
2.0	6.5, 7.0	Substantial extension of functionality over 1.5.1. Increased scope for sharing: <ul style="list-style-type: none"><li>• architectural design model for cost estimating,</li><li>• HVAC design for cost estimating,</li><li>• architectural design for thermal load calculations and/or HVAC system design</li><li>• client brief / space layout for architectural design</li></ul> Additional concepts are provided for special transfers
2x	7.0, 8.0 <sup>2</sup>	The IFC2x platform release will be stable until at least 2005 to allow the industry to deliver mature interface support. Additional capabilities on top of the IFC2x platform may be introduced, but the solutions remain downward compatible, so that previous IFC2x files can be still read by the enhanced IFC2x interfaces.

<sup>1</sup> No commercial release for AC8 until 2003. Contact your distributor for details

Table 2: IFC versions and ArchiCAD support

This task is really no different to planning a .DWG or .DGN transfer, except now

<sup>2</sup> Versions for all three IFC releases will be available 2<sup>nd</sup> Qtr 2003

the quality and richness of object information can radically improve the information to be shared and indeed lead to a re-think of the processes involved in design collaboration and estimation for example

## Determining the Scope of a Transfer

Who are the parties to the transfer and what information should be transferred? Is it one way or two way? How are versions to be managed? While it would normally be preferable to use the latest IFC release, it may be that the one partner has a limited choice only. Once these preliminary decisions have been made, consideration needs to be given to specifying the detailed content of the exchange.

## Views

IFC cover a diverse range of information within building construction and the model does not distinguish who should be exchanging that information or at what point in a project the information is being exchanged.

Software applications are more usually concerned with specific requirements and should not have to implement or use every class that is contained within the IFC model. Therefore, subsets of the model are defined that, when isolated from the complete IFC model, still act as a coherent model. These subsets are called views.

Views are used to support the many different data needs of say architect, structural engineer, HVAC engineer or cost estimator for example.

In the case of the ArchiCAD IFC Add-Ons the default scope for export is to include everything.

**Note:** ArchiCAD does not support the IFC 2.0 BLIS view for HVAC system design to Quantities/cost estimating

More selective functions are provided in the export settings of the Add-Ons to filter the export of data from ArchiCAD. See **Customizing the IFC data – Filter** below.

IFC Version	View	ArchiCAD Functionality
1.5.1	CAD view	The following entities are supported: Beam, Building, Building Storey, Column, Door, Floor, Opening Element, Project, Proxy, Roof Slab, Site, Space, Space Boundary, Wall, Window.  This entity set is comprehensive enough to underpin a wide range of data sharing needs, based on the carcass of a building.
2.0	BLIS <sup>1</sup> view set	Four focused processes are supported: <ul style="list-style-type: none"> <li>• Architectural design &gt;&gt; Quantities take off / cost estimating</li> <li>• HVAC system design &gt;&gt; Quantities take off / cost estimating<sup>2</sup></li> <li>• Architectural design &gt;&gt; Thermal load calculations / HVAC system design</li> <li>• Client brief / space layout &gt;&gt; Architectural design</li> </ul> See the BLIS reference below for detailed description of the view definitions and objects included.
2x		Coordination View – <b>XXX check name</b> Code Checking – Building Plans (Not available until <b>4<sup>th</sup> Qtr 2003 – OK?</b> ).

<sup>1</sup> BLIS is a worldwide consortium of AEC application developers dedicated to developing IFC compliant products. See their website <http://www.blis-project.org/> for more information.

<sup>2</sup>This view is not supported by ArchiCAD

Table 3: IFC Views

## Set up a Pilot or Test.

Once the full scope has been agreed, test the process by doing a pilot. This activity ensures that both parties get what they expected, so that when crucial deadlines occur, the process can be confidently undertaken. The following guidelines may assist you and your team:

- Start with very simple examples
- Validate at both ends
- Progressively expand data set to the full level
- Document procedures at both ends

Report to Project leaders in all disciplines so that they understand and agree to the process, resources and timetables

**Note:** Remember that sharing information is both a technical task and a management process.

## Useful links for Using the IFC Exchange Protocol

To check current versions of **Graphisoft translators** refer to <http://www.graphisoft.com/ifc/>

To understand more about the **International Alliance for Interoperability** (IAI), the worldwide chapters and the IFC development process see <http://iaiweb.lbl.gov>

To understand more about the **BLIS project** and definition of IFC 2.0 BLIS views see <http://www.blis-project.org/>

To join the Graphisoft **IFC User group** see <http://www.graphisoft.com/support/IFCuser>

To find out the current availability of **IFC compliant applications**, IFC development tools and forthcoming products see <http://www.bauwesen.fh-muenchen.de/iai/ImplementationOverview.htm>

## Quickstart

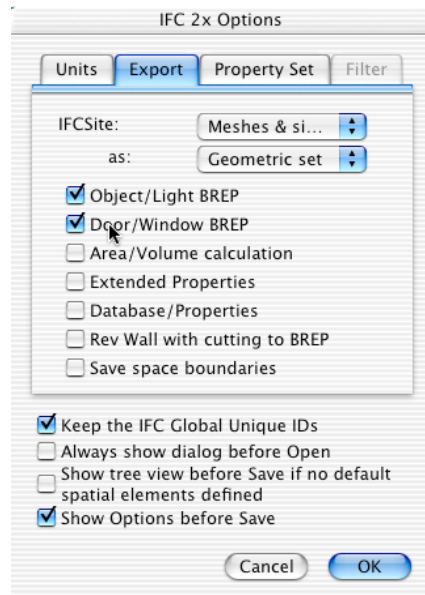
Below is a summary of the default settings for import and export.

The defaults are based on the assumption you are exchanging data with another application that has complete support for the business data needs (or view) that are the subject of the collaboration. For example, if you are an Architect, you are exchanging a model of a proposed building with an HVAC engineer who wants to do a thermal analysis and system design.

## Units

The default is metric. On import the units are set by the sender of IFC file; on export you should define the units required.

## Export



Some export settings can greatly influence the size of the export file; generally if you set BREP as the geometric representation, this leads to much larger files, and more complicated data.

## IFCSite

The default is determined by what the Add-on has already found in the current

database.

## Object/Lamp BREP

Disabled.

## Door/Window BREP

Disabled.

## Area/Volume Calculation

Disabled.

This is only required where the receiving application wants to use the ArchiCAD calculated data – NOT RECOMMENDED.

## Extended Properties

Disabled.

Use only for “round tripping” of IFC data where you want to rebuild the exported model and have the same ArchiCAD element properties etc.

## Database/Properties

Disabled.

## Rev(olved) Wall with cutting to BREP

Disabled.

## Save space boundaries

The default setting is none, that is there is no Custom property set configuration file defined.

## Filter

Generally, ArchiCAD assumes you will export the entire model, but you can select via the Filter tab of the Options dialog, just exactly what you want to export. Filtering the data is a good strategy, firstly by reducing data to a small test example, or later customised to exchange only what is necessary and thus reduce a lot of unnecessary information.

**Note:** For full descriptions of Options see **Customizing the IFC Data – Options.**

# Working with the IFC Translator

## IFC Exchange Processes

### IFC Objects versus 2D Drawings

In relation to the currently common DWG import/export there is a very important difference involved with IFC exchanges. In the DWG 2D case, exchange is predominantly by *layer* i.e. the information to be sent is represented by layer naming conventions which defines the class of usually 2D data being exchanged. This data is not integrated – in fact no building model exists; it is only by visual inspection that coordination of information becomes possible.

In the case of an IFC exchange *objects* are being transferred, and these objects are related to the building model both geometrically in 3D space and by relations such as the room on the 3<sup>rd</sup> Floor, part of a network etc. Objects actually have much more information defined within them – all the parameters defined in the IFC model specification and in ArchiCAD's case many other attributes that describe the object in increasing detail that makes them useful and easy to use.

Figure 2 shows an overview of the processes involved in importing and exporting an IFC file. In this new paradigm, layering is merely a way of ordering and presenting objects, not a proxy for defining information. When undertaking an IFC exchange we must consequently manage the transfer at the **Object level**.

It is important that you understand the requirements for synchronization of objects between you & your partner's databases so that you can manage the versioning of data and audit this process.

### IFC Object Identifiers

The IFC model tracks objects involved in organization to organization exchange by an identifier, the Global Unique Identifier GUID. For every object created, a GUID is assigned that stays with the object for its life. If the object is deleted its GUID is "deleted" too.

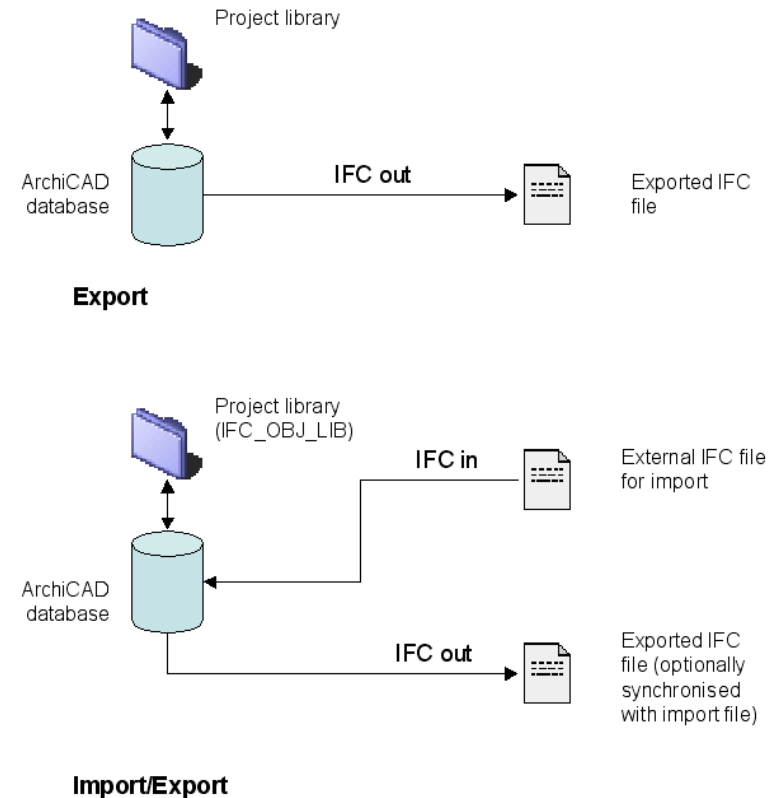


Figure 2: IFC Exchange Processes

## Initiating an IFC Export

In this first case, the user, say an Architect, has decided to issue an IFC file to one of his service engineering consultants. A first consideration is how the translator interprets door and window objects. ArchiCAD has a powerful parametric and intelligent element types for doors and windows; their behaviour is connected to the wall elements they are placed in. The IFC model does not yet support the same level of functionality and as you might expect anyway, all CAD vendors have slightly different approaches to the implementation of these elements as well as other parts of the model. Accordingly, you must understand how the IFC definition relates to ArchiCAD, as these objects have unique considerations involved.

We recommend you use the special library objects for IFC doors and windows, for if you use another door or window object it will be transformed into a proxy object (a general purpose object that carries almost any information a sender wants to exchange) but importantly, will not be recognized when re-imported into ArchiCAD.

The standard IFC door & window types are:

<b>ifcDoors</b>	<b>ifcWindows</b>
IFC2x_DblDoorDblSwing.dor	IFC2x_WindowDoubleHoriz.win
IFC2x_DblDoorFolding.dor	IFC2x_WindowDoubleVert.win
IFC2x_DblDoorSliding.dor	IFC2x_WindowSingle.win
IFC2x_DblDoorSnglSwing.dor	IFC2x_WindowTripleBot.win
IFC2x_DblDoorSnglSwingOpp.dor	IFC2x_WindowTripleHoriz.win
IFC2x_DoorRevolving.dor	IFC2x_WindowTripleLeft.win
IFC2x_DoorRollingUp.dor	IFC2x_WindowTripleRight.win
IFC2x_SnglDoorDblSwing.dor	IFC2x_WindowTripleTop.win
IFC2x_SnglDoorFolding.dor	IFC2x_WindowTripleVert.win
IFC2x_SnglDoorSliding.dor	IFC2x_WindowDoubleHoriz.win
IFC2x_SnglDoorSnglSwing.dor	IFC2x_WindowDoubleVert.win
IFC2x_DblDoorDblSwing.dor	IFC2x_WindowSingle.win

Table4: ArchiCAD IFC Door & Window Library Objects (see Appendix A & D for more details)

These objects have the pre-defined parameters expected and supported in the

IFC model and the GDL objects created by the Add-On have these (and other useful) parameters included. These objects are created normally when you import a new IFC file (but see below for extended options). The Add-On also has a facility to create these objects in your current library (see **Customizing the IFC data – Create IFC library** below). This latter method will ensure your normal project library for instance, includes these objects at project initiation, thus setting the database up for its later IFC export. See **Appendices B & C** for detailed descriptions of these objects. **Appendix D** sets out the Property Sets associated with each ifcObject.

So, having set up as above, the current ArchiCAD database is now saved in the appropriate IFC version format (see **Exporting & Importing IFC Files – Save As IFC file** below) and with any options defined (see **Customizing the IFC data - Options** below) an IFC file is created.

## Importing an IFC file

In this second case, you have been sent a file for the first time on your project. Open the file selecting the appropriate IFC version format, and select a location for the library of IFC objects that will be created (see above). You can either use the default name (IFC\_OBJ\_LIB) and location for the folder or, specify a unique name and location (i.e. Projects/P20642/Libraries/P20642\_IFC.lib).

**Tip:** It is recommended your IFC library be always located in your project specific directory, rather than allowing it to default to the current ArchiCAD library folder, where it may be merged with other project IFC library folders of the same name.

**Note:** Importing an IFC file into ArchiCAD *always* creates a new .pln file. See **Using IFCs on Projects** for techniques to manage IFC exchanges in a working environment

## Exporting a modified imported IFC file

In this third case, you have edited the database created by the import and now want to send back a revised IFC file to your project partner.

A number of decisions must be made about the following:

- treatment of object identifiers (GUID)
- update of the IFC building elements

## Updating IFC Building Elements

If you make changes to the imported elements, for example you change the width of a wall, it will be reflected in the exported model – that is the same

object has a different width. If you keep the original wall the GUID will remain the same; if however you delete the wall and replace it with an identical one, a new GUID is created with the new wall building element.

The synchronization of objects is achieved by “referring” to the IFC file that was responsible for the initiation of the database; i.e. in the case of an imported file, that file, in the case of an exported file the first or most recent file produced.

By choosing the relevant IFC data file, the Add-On is able to reference the object data in the two locations (ArchiCAD and IFC file), carries out a synchronization based on their GUIDs and creates a new export file as follows:

- if the GUID exists in both sources, update the data in the IFC file with the information from ArchiCAD

- if the GUID does not exist, create new objects for the IFC file

- if an object remains in the IFC reference file, but is not in the ArchiCAD data, show those objects as deleted.

Several refinements of this process are possible. See **Exporting & Importing IFC Files – Save As IFC file** below for more information about managing IFC file synchronization.

## Archiving IFC Data

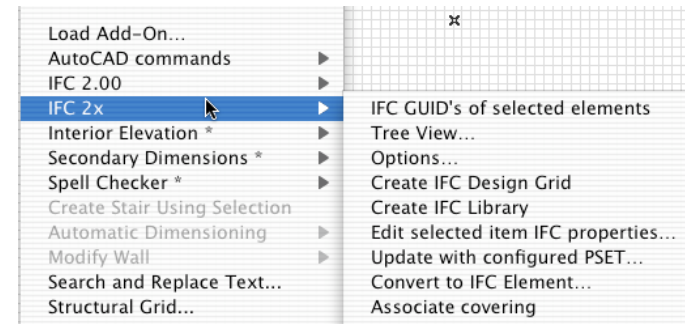
After exporting an IFC file it is recommended to save the .pln file to ensure that the future exports have the same GUIDs for the building elements. Using an archive (extension .pla) will gather all the IFC data and library objects consistently.

For IFC 2x translations there is also a special XML file created (see **Customizing the IFC Data – Edit selected item IFC properties** below) for storing property data. This should also be included in the archive.

## The IFC Tools Menu

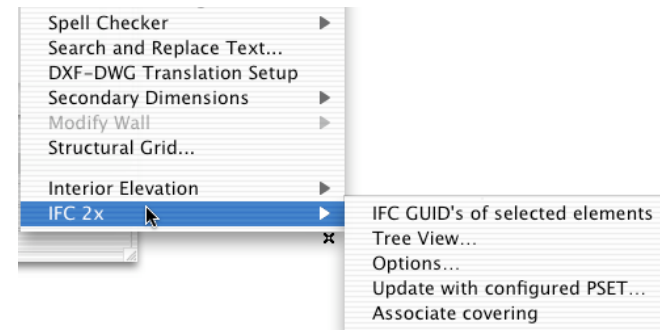
There are several differences between ArchiCAD 7 and 8 versions of the IFC2x add-on.

### ArchiCAD 7.0 IFC 2x Add-on



### ArchiCAD 8.0 IFC 2x Add-on

In ArchiCAD 8 the new handling of libraries, creating IFC properties for GDL objects removes the need for the several functions as shown below.



In the sections following, the ArchiCAD Release specific functions will be highlighted.

### IFC GUID's of selected elements

Choosing this menu will list the selected element's IFC Global Unique ID (GUID)s and the ArchiCAD Unique ID (AC\_UID) in the ArchiCAD Report window.

### Tree View...

The Tree View allows you to examine and edit the IFC project Data, and to edit IFC entities in the model.

See **Understanding IFC Model Data** following.

**Note:** A shorthand keystroke is defined to open the IFC Tree window -

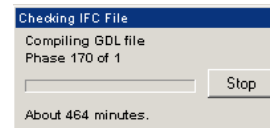
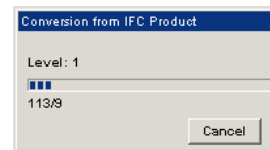
⌘ IFC

## Understanding IFC Model Data

### The ArchiCAD Database and IFC Data

When you import a new IFC file, or export an IFC file from your .pln file ArchiCAD stores special IFC information in the .pln file. This is necessary to ensure that both ArchiCAD and the IFC data are consistent, and for example information that is not used by ArchiCAD on import, can be re-exported. This process also applies to library objects.

Consequently, for IFC import and export, ArchiCAD synchronizes the ArchiCAD database with the IFC stored data. Part of this information is used to create the IFC Tree Dialog called IFC Structure.

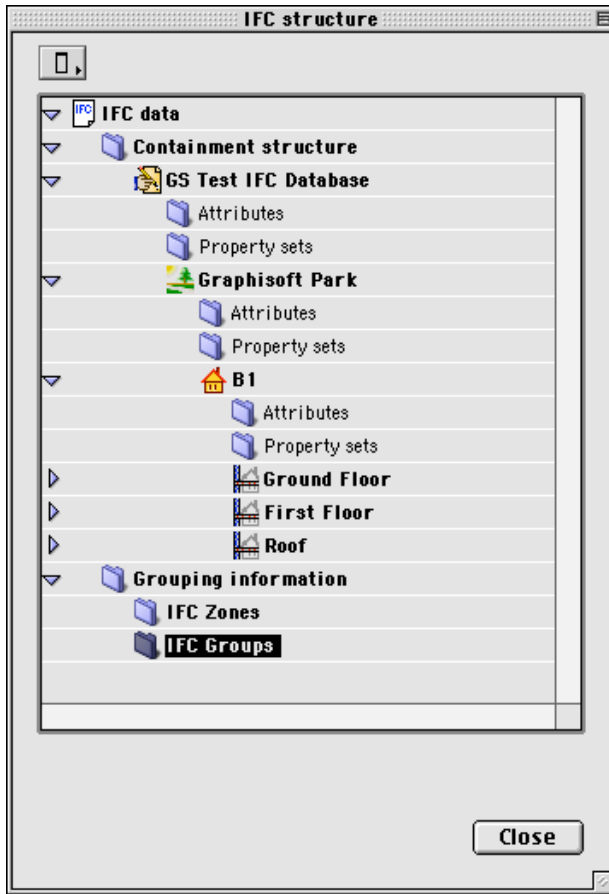


In a larger IFC files and existing ArchiCAD databases the synchronization may take some time.

The Conversion dialog reports progress storey by storey. The Checking IFC file reports on library creation, GDL creation etc.

### The Tree Dialog

The IFC 2x add-on provides a direct interface to view the data in the IFC model; the tree view additionally provides a new mechanism to manage property data, create new entities, select from multiple building data and define ifcZones.







The Tree dialog is displayed in one of two situations:

- An imported IFC file contains more than one building, or
- The user has selected it from the **Tools** menu, **IFC 2x, Tree View...** item

## The IFC Model

The Tree structure follows the rules of the IFC model (**ISO PAS 16739 – IFC 2x Platform Specification**).

The key concepts are:

Icon	Entity	Description	Restrictions (Yes, No)
	<b>Project</b>	The undertaking of some design, engineering, construction, or maintenance activities leading towards a product. The project ( <i>ifcProject</i> ) establishes the context for information to be exchanged or shared, and it may represent a construction project but does not have to.	Yes. A Project must contain at least a Site or a Building. A Project may have many Sites, and many Buildings
	<b>Site</b>	A defined area of land, possibly covered with water, on which the project construction is to be completed. A site ( <i>ifcSite</i> ) may be used to erect building(s) or other AEC products.	No. A Project may not contain any Sites.
	<b>Building</b>	A building ( <i>IfcBuilding</i> ) represents a structure that provides shelter for its occupants or contents and stands in one place. The building is also used to provide a basic element within the spatial structure hierarchy for the components of a building project (together with site, storey, and space).	No. A Project may consist of just a Site.
	<b>Storey</b>	The storey ( <i>IfcBuildingStorey</i> ) has an elevation and typically represents a (nearly) horizontal aggregation of spaces that are vertically bound.	Yes. A Building must have at least one Storey.

The Tree Dialog is the means by which you can navigate the model data.

In addition to the key entities described above there are many other entities which represent the parts of the building carcass (such as *walls*, *beams* and *columns*), or equipment and furniture, or service systems (such as a fan or ductwork).

Many of the IFC entities have a direct counterpart in ArchiCAD i.e. ifcStair is the element Stair. Some don't, such as IfcZone (a collection of Spaces), where a Space is called (confusingly) "Zone" in ArchiCAD.

The Tree Dialog is also the place where ifcEntities that do not map directly to ArchiCAD are viewed and edited. For example you can create an IFC Zone and associate a group of spaces (Zones).

## The IFC Project Data Definition

Whenever an IFC model is created certain data is mandatory. The entities described opposite are termed the IFC Project Data. In summary:

- A Project is mandatory (and must contain at least one Site or Building)
- A Site is optional
- A Building is optional
- Every Building has at least one Story.

Data for this structure or framework is always exported, even if you decide to filter the export to a certain story or exclude geometry data for example.

Note also that an IFC project may have multiple buildings and multiple sites. The tree view shows you always the complete structure, even if you have only imported one of several buildings in an IFC source file (see **Working with Multiple Buildings** following)

In ArchiCAD the familiar view of the model is by Story, and one story also has Site data.

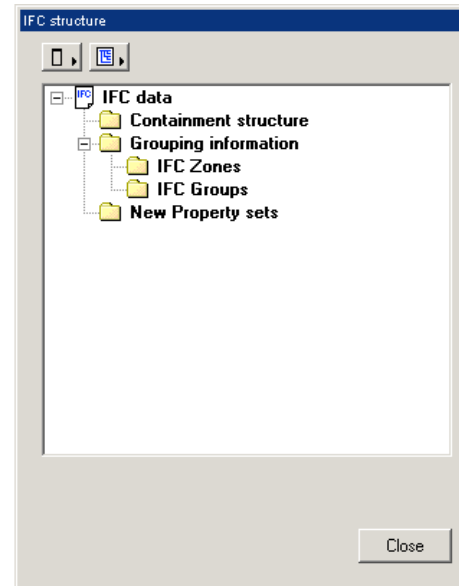
In the IFC model each story has a collection of building elements, described by geometry and property data, very similar to the ArchiCAD model.

We now see there are three types of data:

- The Project Definition
- Geometry Data
- Property Data

## Creating & Editing Models

The Tree View, as we introduced earlier in this section, synchronizes its data with ArchiCAD; if you open a new ArchiCAD session with the IFC2x add-on loaded, a new IFC data set is created.



The IFC data is composed of several structures:

- Containment structure
- Grouping information
- Property sets

### Containment structure

This structure describes the hierarchy of elements in the model, and is where most data is found and edited. See the previous section for key entities.

See **Appendix XX: IFC2x Entities** for the complete list of entities in Platform 2X.

### Grouping information

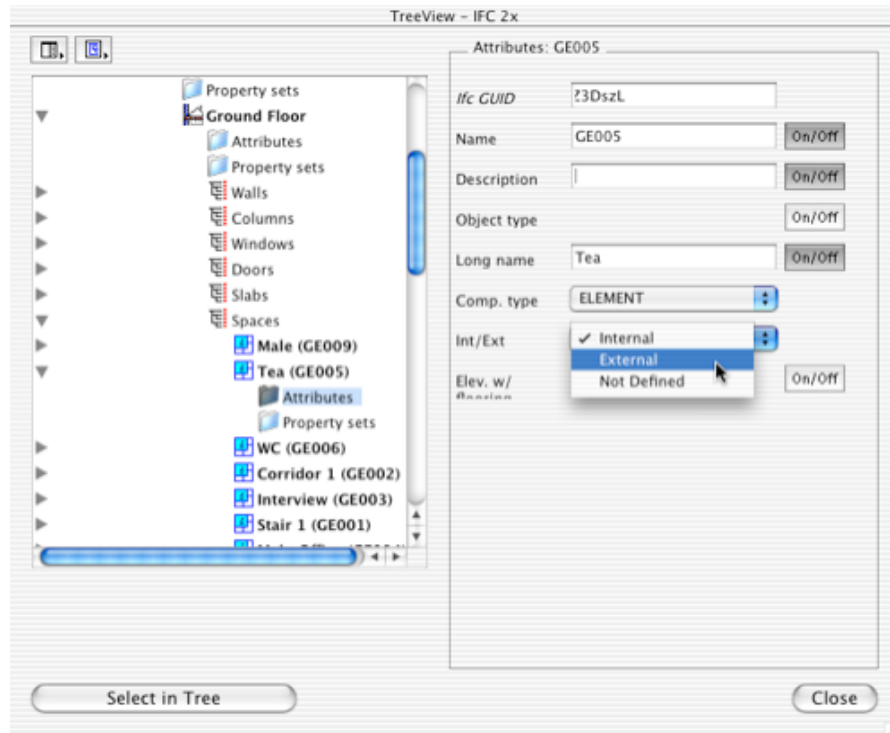
Provides for collections of entities allowed by the model. As ArchiCAD does not yet support the IFC Zone -> Space relationship, this is where you can define them.

## Property sets

The IFC Model provides the assignment of standard PSETS to elements; many elements may share values from a single PSET. This structure allows you to navigate the PSET data sets and assign and edit them for specific entities.

## Editing Models

If you click on the 'Tree view' item in the IFC2x add-on's menu the tree will contain the original IFC file structure but all the items which are not part of the selection path are disabled.

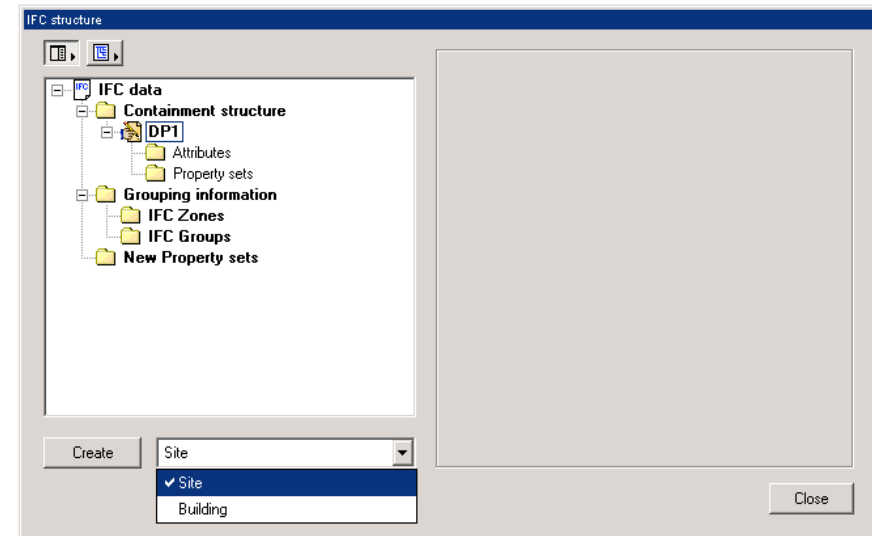


This principle is used through out the model.

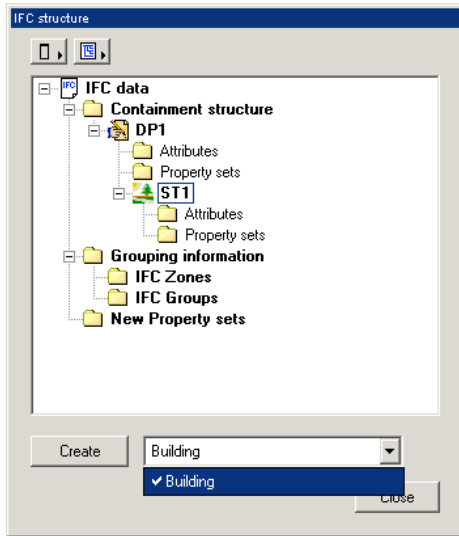
Every entity in the model has two types of data:

- *Parameters* – formal definitions of an entity in the model, considered to be widely applicable (some of which are mandatory such as the GUID – see **XXXX**), or
- *Properties* – additional descriptions of an entity, which are grouped in

structures called property sets PSET (see Configuring PSET Data following). If you expand the tree View Dialog (see previous diagram) it reveals the attributes of the currently selected entities in the tree view hierarchy.



Having created a Project, by selecting it, a **Create** button is presented, offering us one of two choices, a Site or Building. Select Site and the entity is created.



We can continue this process, as far as desired, always subject to the rules of the Model (see **The IFC Model** previous). In the case above, the only option for the Site is a Building.

If you want to change the attributes of the enabled elements just click on the 'Attributes' node on the tree control. You can see the attributes of the disabled elements as well but you can't edit them.

There is an intelligent element creation mechanism in the tree view dialog. Depending on the selected node at the bottom of the dialog a 'Create' button and a popup come up or disappear depending on the node type and other conditions.

For example, if you import only a project object from an IFC file, you can create (add) a site and a building component in the tree. If you import a site from an IFC file you can create only a building (as a spatial element).

The key advantage of the IFC exchange protocol is the rich data that can be defined for entities in the model. This is one of the distinguishing differences with 2D graphic data and formats such as DWG.

Selecting an entity in the Tree for most entities, its attributes are displayed on the right-hand side of the dialog.

You will be allowed to create elements until your 'selection path' is complete. Completeness means that the path contains an object from the following types:  
*project, site (optional) and building.*

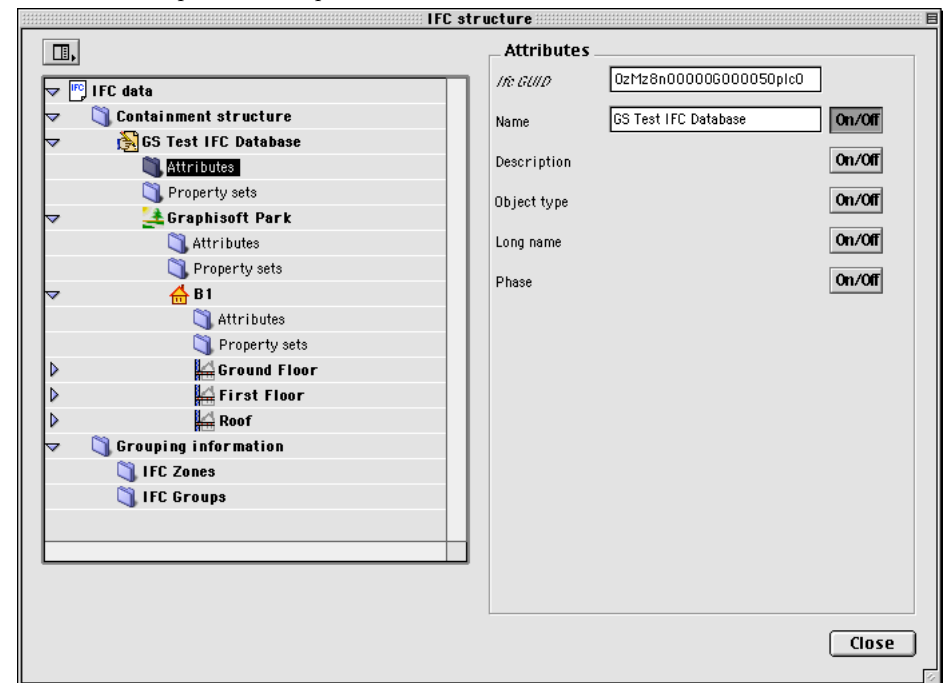
If you start from an empty plan file you have an empty tree view and you can create a project, site, building node in the tree view and set their attributes. So if you create a new project from scratch, always create project, site (optional), building objects otherwise the building elements won't be saved into the exported IFC file.

In the example below we have selected the Project entity, and have displayed its attributes.

## Multiple Building Models

When you open an IFC file containing multiple buildings, the Tree View dialog will be displayed automatically. To continue the import, you need to select a specific building as ArchiCAD can only treat one building at a time.

This means, if for example you select a building, the 'selection path' will contain the building and its parents in the tree (there can be a maximum of two parents (a Project and a Site or a Project)). Only the objects which belong to the selection path are imported from the IFC file.



**Note** You can adjust the dialog to show all attributes and property sets, or

park it out of the way.



If you click on the 'Tree view' item in the IFC2x add-on's menu the tree will contain the original IFC file structure but all the items which are not part of the selection path are disabled. If you want to change the attributes of the enabled elements just click on the 'Attributes' node on the tree control. You can see the attributes of the disabled elements as well but you can't edit them.

## Creating the Building Model

There is an intelligent element creation mechanism in the tree view dialog. Depending on the selected node at the bottom of the dialog a 'Create' button and a popup comes up or disappears depending on the node type and other conditions. **Currently, only the spatial element creation is implemented.**

For example, if you import only a project object from an IFC file, you can create (add) a site and a building entity in the tree. If you import a site from an IFC file you can create only a building (as a spatial element).

Selecting an entity in the for most entities

For each selected IfcElement its attributes are displayed on the right-hand side of the dialog.

You will be allowed to create elements until your 'selection path' is complete. Completeness means that the path contains an object from the following types: project, site (optional) and building.

If you start from an empty plan file you have an empty tree view and you can create a project, site, building node in the tree view and set their attributes.

Note: If you create a new project from scratch, always create project, site (optional), building objects otherwise the building elements won't be saved into the exported IFC file.

## Creating Site Objects

## Creating IFC Compatible Objects

You will see that most of the ArchiCAD elements are direct mappings to entities in the IFC model, eg the ArchiCAD element *Beam* is equivalent to the **ifcBeam**.

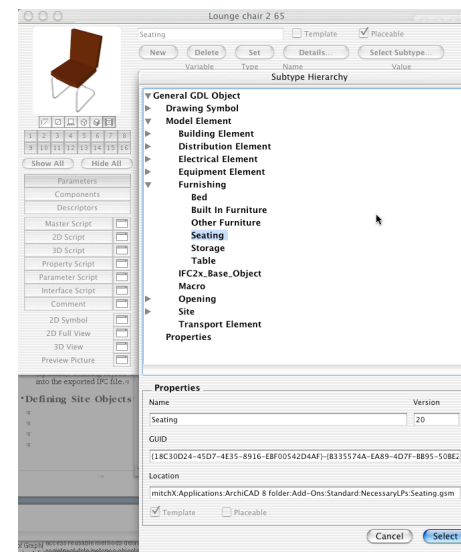
As ArchiCAD's elements are not exhaustive, other IFC model building parts are represented by GDL objects. In ArchiCAD 7, a tool is provided to convert GDL objects into various IFC entities, for example ifcSensor is represented by a GDL Object.

**Note:** Refer to **Appendix A: IFC Support** for a complete list of IFC entities supported by all of ArchiCAD's IFC add-ons.

Importantly, all IFC entities have attributes and properties (see **Editing IFC Entity Properties**).

## Objects in ArchiCAD 8

All GDL objects have a new feature called a subtype, organized in a tree hierarchy based on the IFC model. This new feature brings many other benefits to users, but a particular benefit to IFC users, is that the process of interpreting objects in the ArchiCAD database is automatic, which are now referred by their subtype.

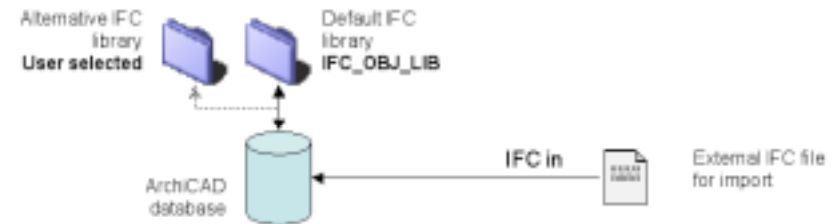


Accordingly when creating new objects in your model you should

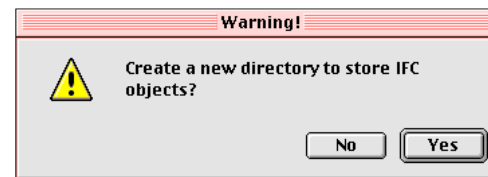
## Exporting & Importing IFC Files

The key IFC operations are performed by the **File Open...** and **File Save as...** menu items.

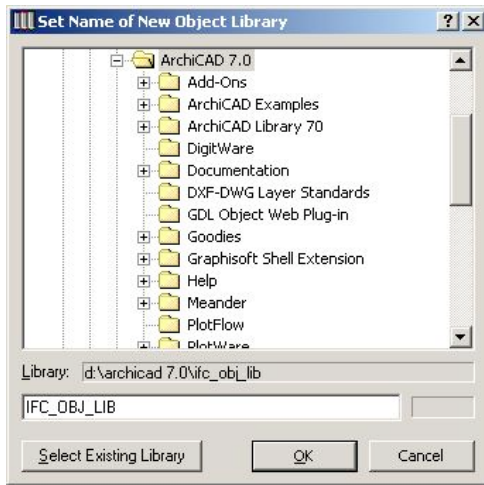
### Open IFC file:



Choose the **File** menu, **Open...**, **Files format: IFC\_2x** and select the IFC file. After selecting the IFC file the Add-On asks if you want to create a new library to store the IFC objects.



If you select **Yes**, you can set the new object library.



The Add-On checks for existing IFC libraries and will offer you options to merge or not.

The result is either an existing library is reused or a default or custom library created. If the used library doesn't contain IFC door and IFC window objects (see **Initiating an IFC Export**), the Add-On creates them.

## Save As IFC file:

Choose the **File** menu, **Save As...**, **File format: IFC\_2x** and set the name of file.

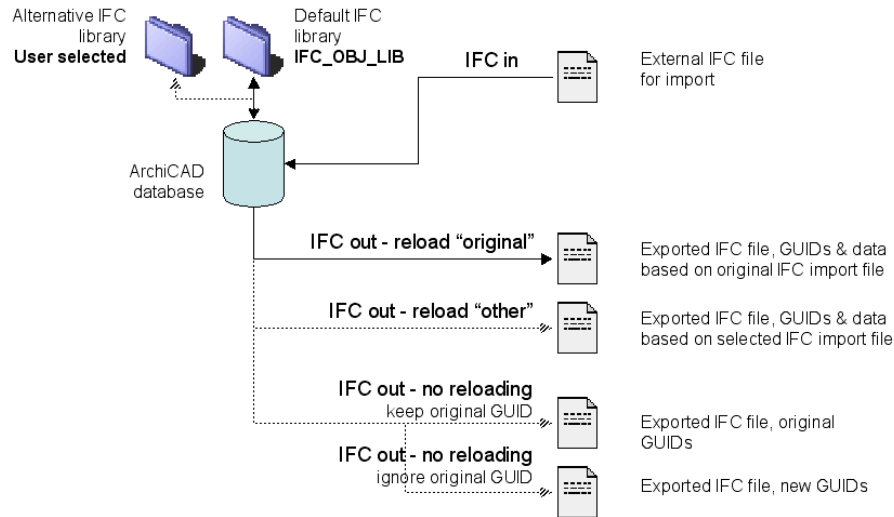
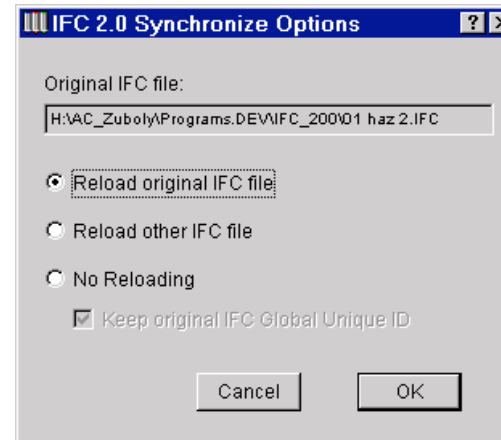


Figure 3: Synchronizing Options for IFC export files

If the ArchiCAD database (.pln file) originated from an IFC file the Add-On asks you to specify how you will treat the new file with respect to the originally imported IFC file.

If you select either of the Reload options the Add-On will update the changes with reference to the specific imported model file chosen and will export the combined result in the destination file. In this way the IFC file will also contain those items which are not understood by ArchiCAD such as extension property sets added by another application or some HVAC or FM related information.



### Original IFC file:

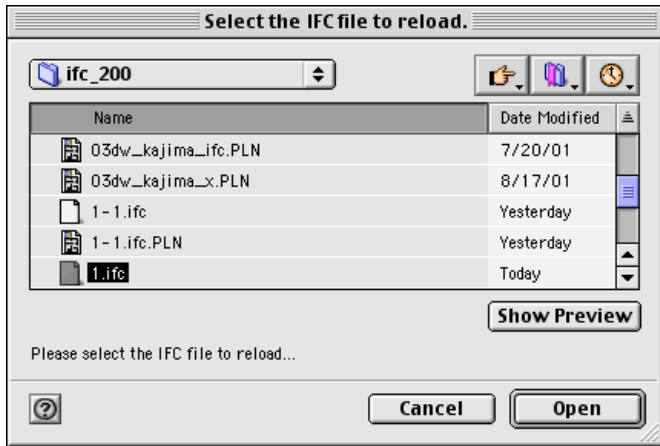
This text box displays the full path of the original IFC file. It cannot be edited in the box, but may be copied (if you have a long or complex path name) to specify a new file path if you change the synchronize option from other than using the original.

### Reload original IFC file

Selecting this option selects the "original IFC file" as the reference IFC import file to be synchronized. It is the file that either created the database, was the initial source of the particular data set or the most recent IFC file exported. Unchanged objects keep all their original parameters (like Global Unique ID, Instance Number, ...). This file is the default name in the dialog.

### Reload other IFC file

If you select to reload another IFC file, the dialog prompts you select it, and uses its entities as the reference IFC export file for synchronization.



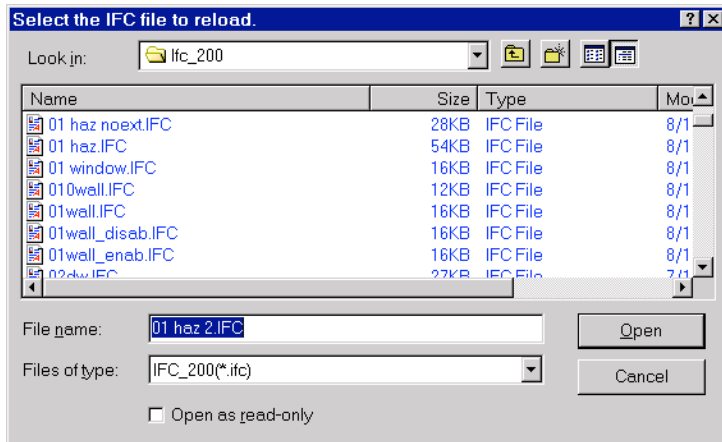
original unique ID (which are stored in the ArchiCAD database). No reference is made for synchronization.

Disabling this option causes ArchiCAD to create a new set of GUIDs.

**Note:** This last option means that the export file has a different set of GUIDs, no longer connected to the original IFC model. This option supports a “template” function and allows new projects to be created based on an original.

See **Customizing the IFC Data – Filter** for the methods customize and filter model data for export.

Note: Filtering will only affect the selected elements; the IFC Project Model Data - Site, Building and Storey structure is always exported. For example if you have a 3 storey building if you select only the top floor for export, you will still have the full IFC structure.



The result is similar as for “original”.

**Note:** The Add-On checks that this referenced file is a valid IFC file for this purpose.

## No Reloading

If you select not to synchronize your new export file, a further option is enabled that allows you to decide between one of two methods for the treatment of IFC GUIDs:

### Keep the original IFC Global Unique ID

Enabling this option causes ArchiCAD elements to be exported with the

## Customizing the IFC data

### Options...

A number of options are provided to filter and customize the information included in the export file.

Four tabs define respectively Units, Export options, Property setup, and on exporting Filter defines the scope of the data.

As some of these settings can have a significant impact of the export file size, effort should be given to ensuring that they are correct for the purpose intended.

### General

There are some globally applicable settings to aid checking of the selection and treatment of data.

#### Keep the IFC Global Unique IDs

Retain the GUIDs ???

#### Always show dialog before Open

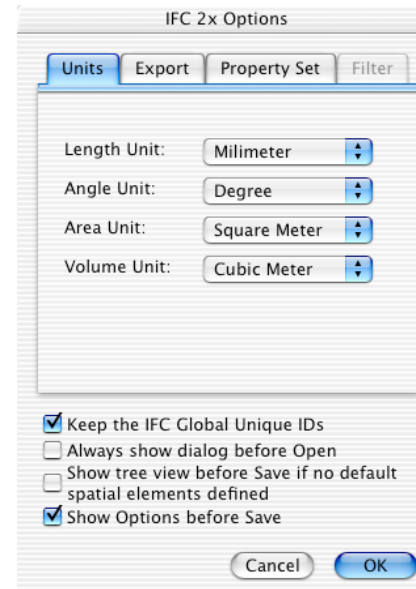
? ?

#### Show tree view before Save if no default spatial elements defined

? ?

#### Show Options before Save

Always show the Options dialog before exporting data

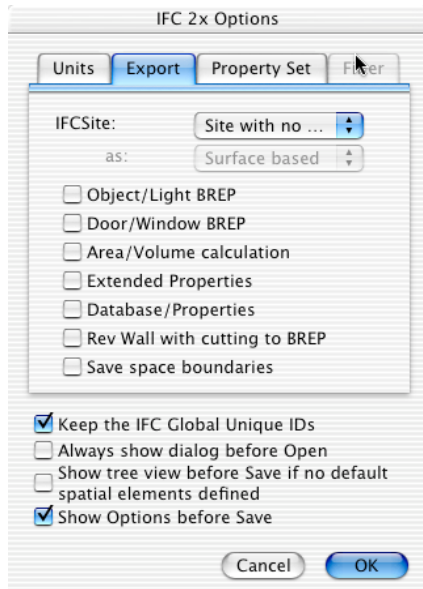


### Units

Specify the units for Length, Angle, Area and Volume

Enabling the **Show Options before Save** ensures that the user confirms these parameters at the time of export.

**Note:** This option only applies if you are creating a new IFC export file.



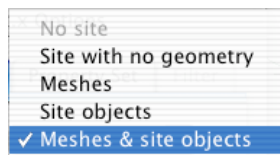
## Export

Specify the way certain objects are handled.

Enabling the **Show Options before Save** checkbox ensures that the user confirms these parameters at the time of export.

**Note:** This option only applies if you are creating a new file.

## IFCSite



Export site geometry data by choosing one of **three** methods:

In all the following options an IFC site is defined in the IFC project data.

### Site with no geometry

Exclude site geometry on export, but include the Site **definition??** and attributes.

## Meshes

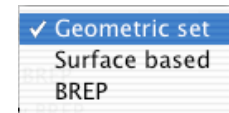
Export any meshes (there may be more than one) as the source of the Site geometry data

### Site object

Use the object **IFC\_SITE.GSM** (must be in uppercase text) as the source of data for the Site

### Meshes & site objects

Use both meshes and site objects as the source of geometry data for the Site



Choose a representation for the site geometry data.

### Geometric Set

Export the geometry as a set of 3D points and, where defined, known connecting lines and the site boundary edges. For meshes, points will be the survey points. For site objects, points will be the vertices of the (GDL) object.

### Surface based

Export a single surface, based on the extent defined by a plan view. Included meshes and site object data will be as noted above for Geometric Set.

### BREP

Export the site as a single body in BREP format.

### Save Proxies as BREP [Object/Lamp BREP]

If this checkbox is enabled, Ifc Proxies (any ArchiCAD element that is not supported by the IFC 2x model) will export its geometry in BREP format.

In this case the exported file size can be significantly bigger than without the geometry. Normally, it is better to disable this flag, because a smaller IFC file is created, and limited usefulness (generally only visualisation) of the BREP geometry in the receiver's application.

### Save Door/Window as BREP [Door/Window BREP]

If this checkbox is enabled ArchiCAD's door and window geometry will be exported in BREP format. In this case the exported file size will be significantly bigger than without the geometry.

Normally, it is better to disable this flag, because a smaller IFC file is created.

## Area/Volume calculation

**Note:** ArchiCAD's Area/Volume calculation data is not required for the IFC certified export/import.

If this checkbox is enabled the Add-On will calculate the elements' area and volume attributes. The area is usually the profile area of the extruded segment but in the case of a wall the area is the average area of the space boundary side of the wall (not the area of extruded profile).

Disable this flag if the calculation is not required.

## Extended properties

**Note:** Extended property data is not required for the IFC certified export/import.

In some situations you can gain an advantage if you export the Extended properties. Because the Add-On saves the ArchiCAD attributes and parameters in the PSET data (for example the ArchiCAD object library name) this setting is useful when you read the IFC file back into ArchiCAD as the Add-On will try to use the same library part as it is defined in the extended properties.

If the same library is loaded in export and import cases then the Add-On can use the original library element and doesn't read the (BREP) geometry. Otherwise, the Add-On creates a new library element from the BREP information or 2DGeometry. The shape will be the same but the color of the object will be different and monochrome.

**Caution:** If the object has changed in the source .pln file, the re-imported object data will be incorrect.

**Note:** Exporting extended property data will significantly increase the file size.

## Database properties

**Note:** ArchiCAD database properties data is not required for the IFC certified export/import.

The Add-On can export the ArchiCAD Database and Properties to the IFC file. However the Add-On cannot reconstruct (read back) the database and properties because the API interface doesn't allow it.

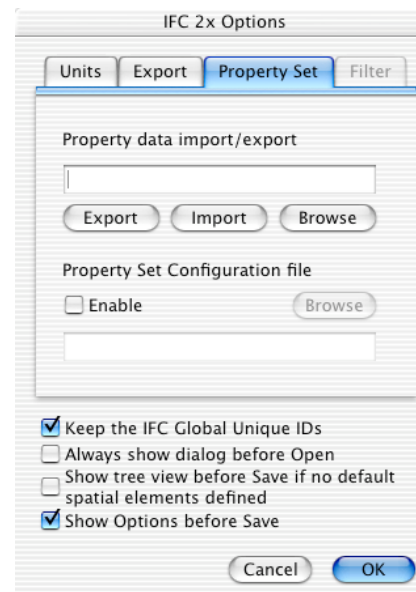
In most cases it should be disabled.

## Revolved wall with cutting to BREP

If this checkbox is enabled the Revolved wall with cutting will be exported as a BREP instead of Advanced: Attribute Driven Clipped Revolved Solid geometry. Some CAD systems can interpret this geometry better.

## Property Set

The IFC 2x Add-On supports configuration of customizable PSET data. See **Using PSET Data** below for a description of the PSET configuration file format and mapping mechanism, and the PSET data operations when this option is enabled.



You can enable or disable the saving of the standard IFC property set in a user defined XML file. See **Appendix E: PSET Data** for details of 2x **XXXXXX**. Refer to IAI International web site <http://iaiweb.lbl.gov/> for details of IFC 2x PSET definitions.

You can enable or disable the configured PSET processes and you can use different configuration file for each plan file.

## Property Set Configuration file

This file stores the ArchiCAD and IFC property parameter mapping (see Using PSET Data below). This file should also be located in your working project directory, as described above.

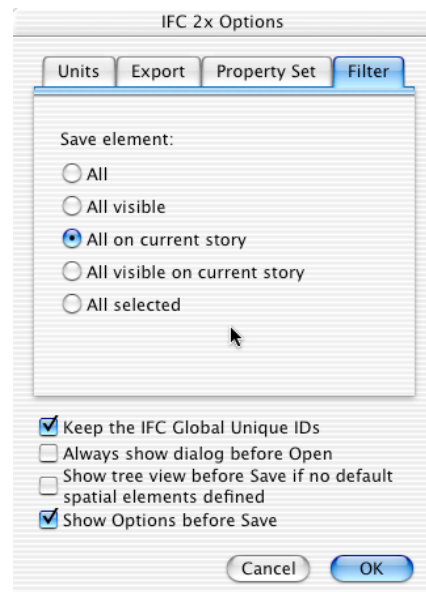
### Enable

Add the custom PSET defined data to the IFC export file

### Browse

Locate the XML configuration file using the Finder.

**Note:** The default location is a folder named XML\_IFC\_WP in your ArchiCAD application folder. If you have multiple projects this may be overwritten so we recommend locating the file in your working project directory.



## Filter

Filter elements in the current ArchiCAD database for export.

**Note:** This dialog is accessible only before the save and if the **Show Option before Save** is set from the **Tools, IFC 2x, Options** dialog box.

The filter option only applies to the creation of “new” IFC files in two cases:

- from a database created in ArchiCAD for its first IFC export, or
- from a working database - where there have been previous IFC exports or the database was created from an IFC import – and where the Synchronize option is **No Reloading**

If the buttons are enabled the following actions occur:

### Site <site\_name>

Export the current IFC Site entity selected in the IFC Project Data

**Note:** IFC data is stored in the ArchiCAD .pln file. The IFC Tree View dialog displays the current state of the IFC project data.

### Building <building\_name>

The name of the currently selected Building in the IFC Project Data (informative only)

### All

Save every building element on every story of the building visible or not.

### All visible

Save every visible building element on every story of the building

### All on current story

Save every building element on the current story of the building visible or not

### All visible on current story

Save every visible building element on the current building story

### All selected

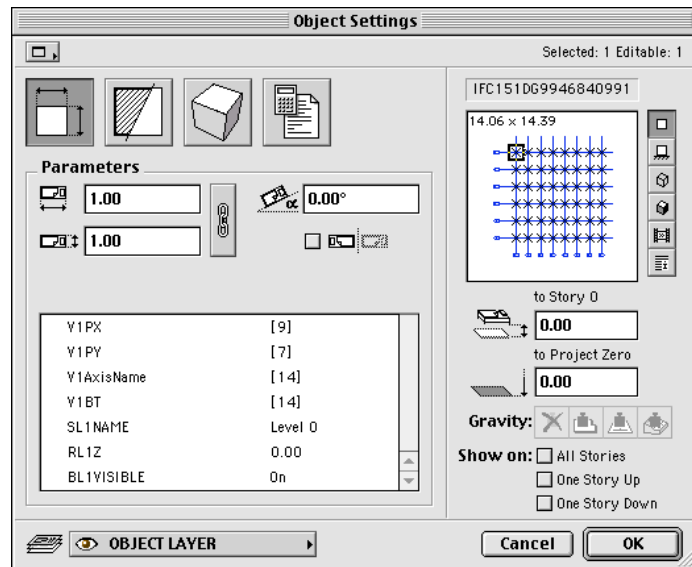
Save all selected elements on the current story

This filtering function ensures that the IFC data file is a consistent body of data. Refer to **Exporting & Importing IFC Files – Save As IFC File** for the synchronization options for the **No Reloading** option. If the **Keep original GUIDs** is enabled exported objects will have IDs based on the database and will be related; if disabled, new GUIDs will be created to establish a new database having now no relation to the original database.

## Create IFC Grid

To exchange an IFC Grid (the Japanese term for a design grid is Torishin) requires creating one with the Tools menu. Follow this sequence:

- Draw lines and arcs of the design grid. Use a different color to draw the crossing lines. The tool can use the line color information to differentiate between the axes.
- Select them (with marquee if you need a multi-level design grid).
- Select the **Tools** menu, **IFC\_1.51, Create IFC Design Grid** to convert the lines to a design grid object named as IFC151DG<ID\_NUMBER>.GSM . You can change the object by adjusting its parameters:



### Variable name Description

<b>V1PX</b>	The distance of cross points in X direction.
<b>V1PY</b>	The distance of cross points in Y direction.
<b>V1AxisName</b>	The name of the axis.
<b>V1BT</b>	The position of the axis name (toggle left, right or top, bottom).
<b>SL1Name</b>	The name of grid level (not visible in ArchiCAD,

appears only in IFC file).

**RL1Z**

The height of the grid level.

**BL1Visible**

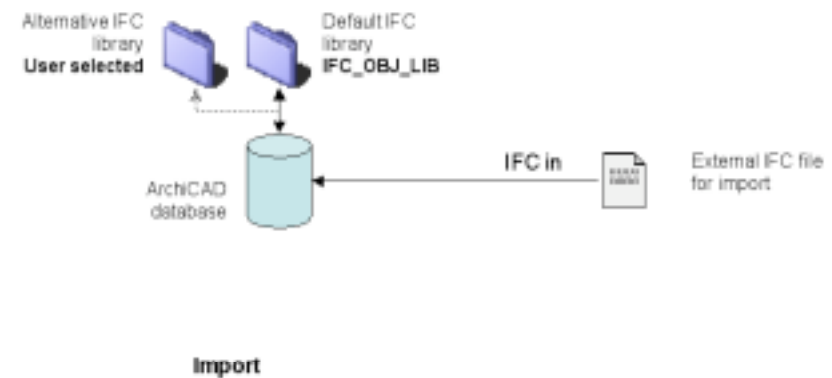
The visibility flag of the grid level.

After saving to an IFC file the object will be converted to IfcDesignGrid.

## Create IFC Library

This menu item allows a user to create the IFC compatible standard library objects for doors and windows. If chosen it places them in the current active library.

**Note:** See **IFC Exchange processes – Importing an IFC File** for discussion of IFC library selection and creation.



## Editing IFC Entity Properties

Editing all the elements in a model can be a large task, yet if we are to achieve the benefits of rich data exchange, we must define the information our collaborator has asked for. For example if you are exporting an IFC file for costing, all building elements need to be identified by a Cost Classification, such as the US Unifomat System. If this data is comprehensively defined then the cost application can quickly and reliably build an estimate.

## Attributes and Properties

In the IFC model IFC entities have two types of properties:

- **Attributes** – properties of the entity that are part of the entity data, and mandatory, and
- **Properties** – properties of an entity that are linked to the entity, and are optional.

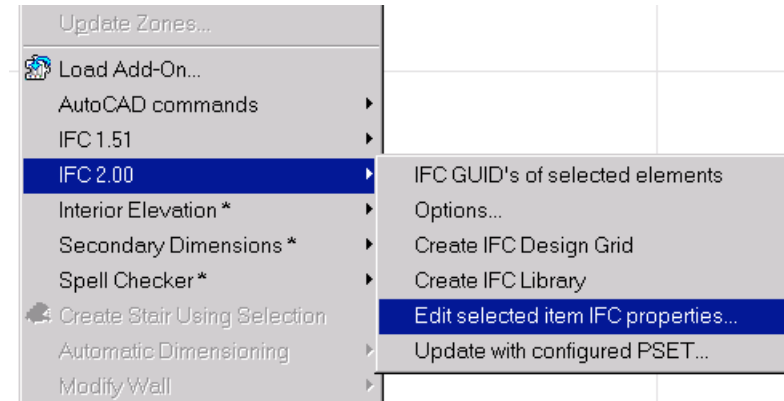
(In ArchiCAD these definitions are very similar, except that an ArchiCAD property object can also be linked to many element types).

The IFC model already has defined groups of properties for certain IFC entities based on industry common practice; these are called **standard PSETS** (property sets) and are documented in **Appendix E: PSET Data**.

From the users perspective both types of data will be used, and particularly for building services and lifecycle analysis, you will use both the standard as well as custom property sets.

Custom property sets are the user definable collections of data for entities. To define custom PSETS see **Configuring PSET Data** following.

**Note:** Custom PSET data is only defined by the PSET configuration file.



## Edit selected item IFC properties...

**Note:** This only applies to IFC version 2.00 for ArchiCAD 7.0

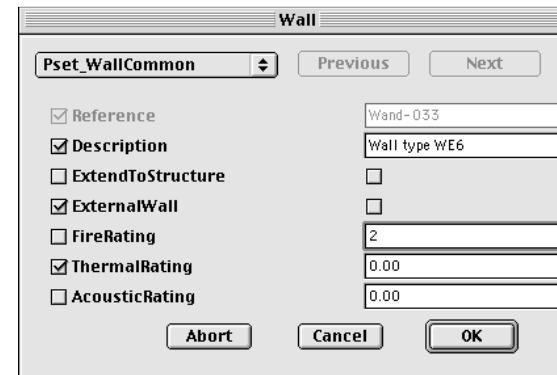
This menu item allows you to review and edit the element's standard PSET IFC properties.

The IFC properties are saved in an XML file which can be found by default in the XML\_IFC\_WP subdirectory of the ArchiCAD application folder.

**Note:** See **Customizing the IFC Data – Options: Property Set** for the for locating the standard property set data file.

The name of the (XML) file is same as the ArchiCAD project except for the extension (.pln -> .xml).

If you select this menu item, the standard Property Sets of the currently selected elements are presented in a dialog.



The first column shows the enable/disable status of the properties. If this

option is selected the property data will be exported.

**Note:** See **Appendix B: Exported Data** and **Appendix C: Imported Data** for detailed specifications of the list of element Property Sets (PSET) and their property inclusions.

The second column is the name of the property and the third is the value. If the current element has more Property Sets then you can select another Property Sets from the list box placed in the upper left corner.

There are some protected Simple Properties which are set according to their ArchiCAD elements. For example: *Reference* is the ID of the element, *Size* is defined by the size of the element.

**Note:** For options on defining the Reference property, see **Customizing the IFC Data – Options: Wall Composite as reference.**

### OK

After changing the parameters selecting OK writes the changes to the XML file

### Cancel

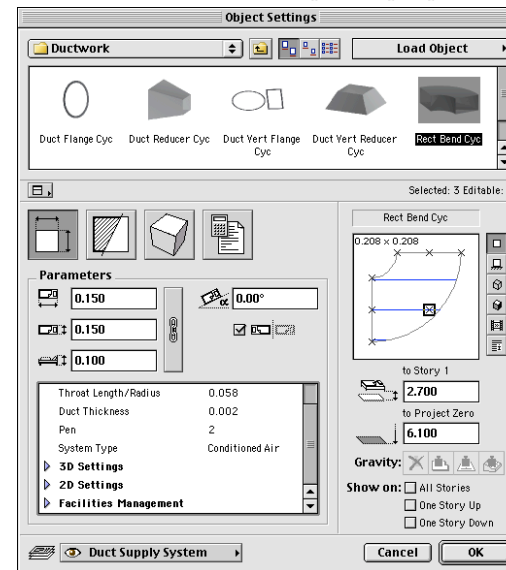
Performs no update for the current element and moves to the next selected element type in the dialog.

### Abort

The Add-On stops writing any changes in the XML file and returns to ArchiCAD.

## Convert to IFC element...

Most objects found in an ArchiCAD library will not be created for IFC usage and will not have the required properties needed for an IFC based exchange.

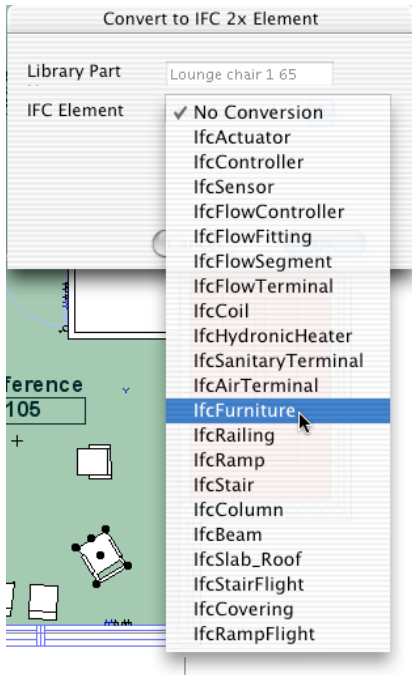


The IFC2x (and IFC 2.00) Add-on overcomes this problem by providing a item in the **Tools** menu to add the required parameters to the GDL object for the currently selected object.

**Note:** This menu item only applies to the ArchiCAD 7.0 Add-on.

**Note:** This tool only works on ArchiCAD elements of type *object*, i.e. GDL objects.

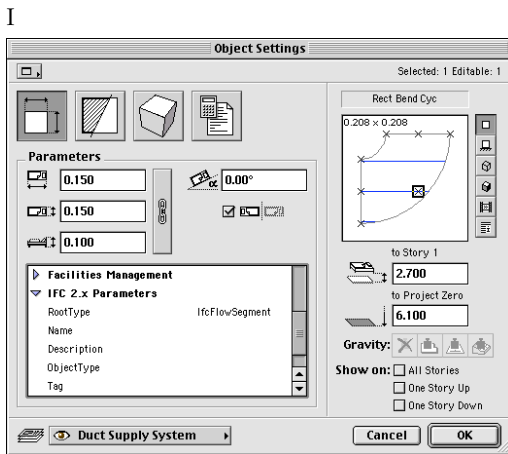
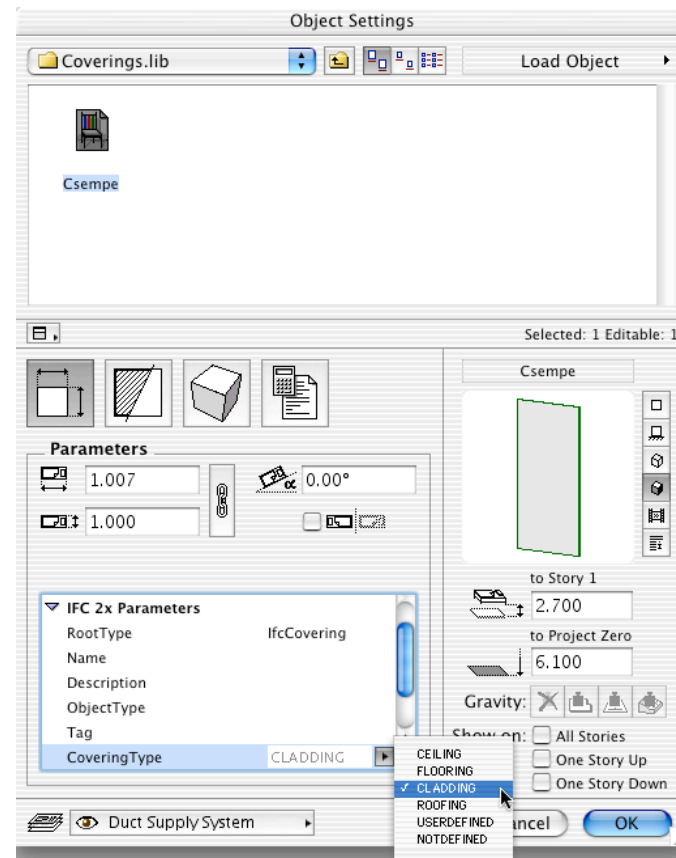
Having selected the library part in the database, the tool will offer you a drop down menu of available conversions: In the case opposite we have selected a fan coil unit library part in the database, and then chosen IfcCoil as the IFC element type to convert to. This results in the following change:



as required by the IFC 2x model. If you require extra property data associated with an object, refer to **Configuring PSET Data - IFC Property Sets** following.

Note that editing the ArchiCAD library part converts all instances of the object in the database. See the next section for editing the properties.

## Associate Covering



The Appropriate IFC properties are created as parameters in the GDL object and the add-on will now recognize this object as the correct IFC entity on exchange. These properties are the defined IFC properties for the element type

## Editing IFC Entity Properties

Editing all the elements in a model can be a large task, yet if we are to achieve the benefits of rich data exchange, we must define the information our collaborator has asked for. For example if you are exporting an IFC file for costing, all building elements need to be identified by a Cost Classification, such as the US Unifomat System. If this data is comprehensively defined then the cost application can quickly and reliably build an estimate.

### Attributes and Properties

In the IFC model IFC entities have two types of properties:

- **Attributes** – properties of the entity that are part of the entity data, and mandatory, and
- **Properties** – properties of an entity that are linked to the entity, and are optional.

(In ArchiCAD these definitions are very similar, except that an ArchiCAD property object can also be linked to many element types).

The IFC model already has defined groups of properties for certain IFC entities based on industry common practice; these are called **standard PSETS** (property sets) and are documented in **Appendix E: PSET Data**.

From the users perspective both types of data will be used, and particularly for building services and lifecycle analysis, you will use both the standard as well as custom property sets.

Custom property sets are the user definable collections of data for entities. To define custom PSETS see **Configuring PSET Data** following.

**Note:** *Custom PSET data* is only defined by the PSET configuration file.

### PSET Element Application

Table 4 below shows the application of Standard and Custom PSETS for each ArchiCAD element. See Appendix XX for the list of all IFC entities supported in IFC 1.51 and 2.0.

In the example opposite

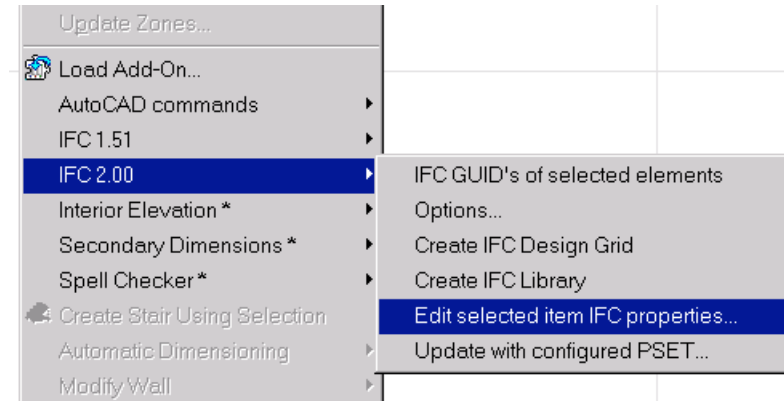
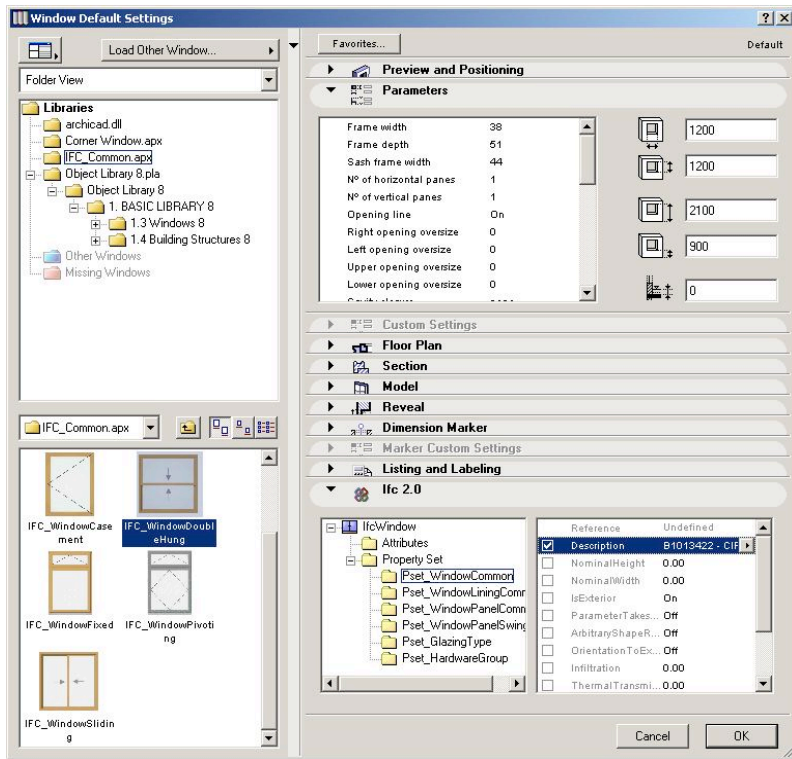
ArchiCAD Element type	PSET		Special Library
	Standard	Custom	
Wall	YES	NO	-
Column	YES	NO	-
Beam	YES	NO	-
Door	YES <sup>1</sup>	YES	YES
Window	YES <sup>1</sup>	YES	YES
Stair	YES	NO	-
Slab	YES	NO	-
Roof	YES	NO	-
Zone (ifcSpace)	YES	YES	-
Object <sup>2</sup>	NO <sup>3</sup>	YES	-

**Table 4:** PSET Element Application

Note that not all elements have the same property set functionality.

#### Notes:

1. In AC 7, normal ArchiCAD .dor or .win elements will only have the respective PSET\_DoorCommon or PSET\_WindowCommon. If the Special IFC Door & Window library is used, the complete set of standards PSETS becomes available. See **Appendix E: PSET Data** for the list of standard PSETS applicable to each ifc entity.
2. In AC8 only, if the object's type matches an IFC type, then all appropriate PSETS will be available for that object. In AC7 if the object name starts with a supported ifcEntity name, eg ifcFIXTURE.gsm then it will be recognized as that IFC entity and have all appropriate PSETS will be available for that object.
3. In AC8 only, if the object's type matches an IFC type, then all appropriate PSETS will be available for that object.



## Edit selected item IFC properties...

**Note:** This only applies to IFC version 2.00 for ArchiCAD 7.0

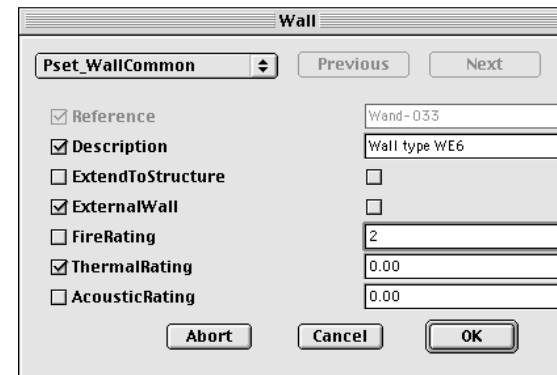
This menu item allows you to review and edit the element's standard PSET IFC properties.

The IFC properties are saved in an XML file which can be found by default in the XML\_IFC\_WP subdirectory of the ArchiCAD application folder.

**Note:** See **Customizing the IFC Data – Options: Property Set** for the for locating the standard property set data file.

The name of the (XML) file is same as the ArchiCAD project except for the extension (.pln -> .xml).

If you select this menu item, the standard Property Sets of the currently selected elements are presented in a dialog.



The first column shows the enable/disable status of the properties. If this

option is selected the property data will be exported.

**Note:** See **Appendix B: Exported Data** and **Appendix C: Imported Data** for detailed specifications of the list of element Property Sets (PSET) and their property inclusions.

The second column is the name of the property and the third is the value. If the current element has more Property Sets then you can select another Property Sets from the list box placed in the upper left corner.

There are some protected Simple Properties which are set according to their ArchiCAD elements. For example: *Reference* is the ID of the element, *Size* is defined by the size of the element.

**Note:** For options on defining the Reference property, see **Customizing the IFC Data – Options: Wall Composite as reference.**

#### OK

After changing the parameters selecting OK writes the changes to the XML file

#### Cancel

Performs no update for the current element and moves to the next selected element type in the dialog.

#### Abort

The Add-On stops writing any changes in the XML file and returns to ArchiCAD.

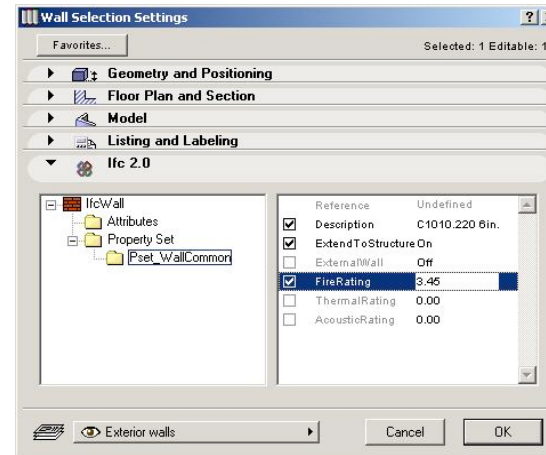
## The ArchiCAD 8.0 Element Settings Dialog

**Note:** This only applies to IFC version 2.00 for ArchiCAD 8.0

In ArchiCAD 8, a user can edit an element's IFC attributes and standard properties in the element's setting dialog box through a custom panel. For example (with the IFC 2.00 Add-on loaded) to set wall properties, open the wall settings dialog box. The custom panel can be seen, named "IFC 2.0". If it is closed click on the panel and edit the attribute or property as required.

**Note:** This feature only works for Standard PSETS; for custom PSETS see **Configuring PSET Data** following.

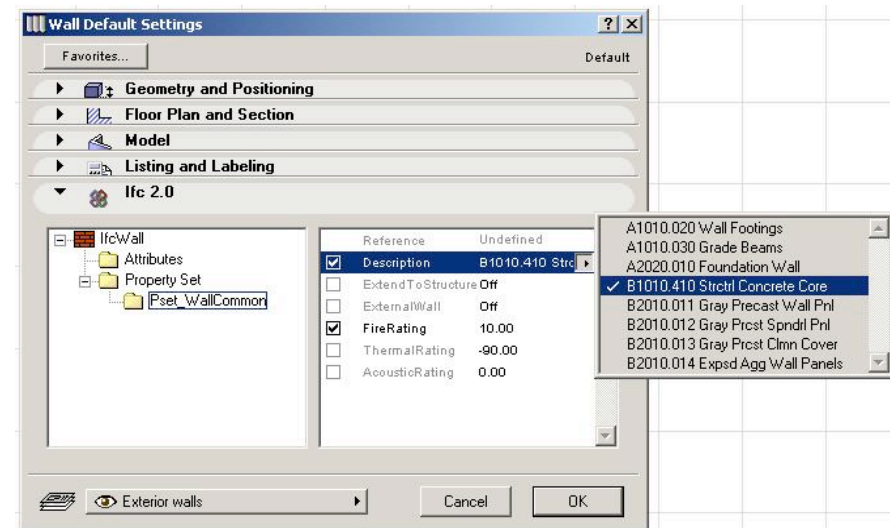
The left sub window of the IFC 2.0 panel is the attribute and properties tree of the element, the right sub window is the parameter list of the selected tree item. The first row of the parameter list indicates if the parameter is enabled for export.



On export, if the flag is enabled, the property data is exported; on importing an IFC file, the flag indicates if the parameter was present in the file and the data imported.

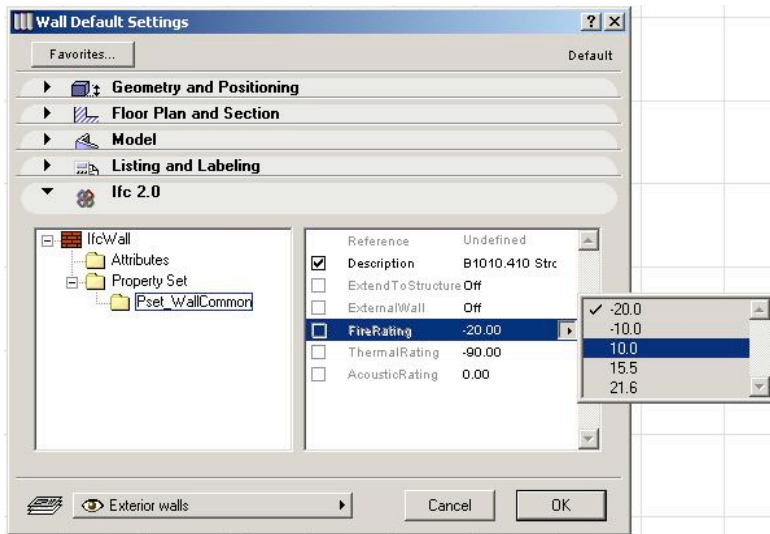
Editing a parameter value field will set the flag to enabled automatically.

## Customizing Element Settings Values



The user can pre-define a set of default values that a string or number type parameter may take. This is enabled by creating an xml file named

“Config\_ifc200\_typeenum.xml” located in the folder “[ArchiCAD folder]\XML\_IFC\_WP” on Windows or “[ArchiCAD folder]:XML\_IFC\_WP” on Macintosh. In the example above the Pset\_WallCommon “Description” parameter displays the predefined string values.



## Using Favorites

To further enhance productivity, Favourites can be used to save template settings with a complete set of parameters in a named favorite (see **Favorite** functionality in ArchiCAD).

To define a favorite:

Select a wall instance, configure all normal ArchiCAD parameters, and importantly, select the chosen IFC 2.0 parameter (in our case “Description”), and finally select the Save Favorites button and save it (with a name to ensure you understand what is meant when you want to use it later on!).



Now, for example suppose we are editing walls and want to set the common typed wall IFC attributes and properties:

Select all the walls to be defined, open the Wall settings dialog box, select the appropriate favorite and choose “OK”.

See Appendix NN for the pictures: ColumnSettingDlg1.jpg ...

## The Element Settings Configuration File

The structure of the Config\_ifc200\_typeenum.xml is:

1. The root of type enumeration:
 

```
<Ifc_Pset_TypeEnums>
```
2. The property Set name:
 

```
<IfcPropertySetName="Pset_ColumnCommon">
```
3. The simple property name:
 

```
<IfcSimpleProperty Name="Description" Type="IfcString">
```
4. The string list item:
 

```
<TypeEnum name="A1010.010 Pad Footings" />
```

The italic text parts may have different values.

## An Example file: Config\_ifc200\_typeenum.xml

```
<?xml version="1.0" ?>
<Ifc_Pset_TypeEnums>
  <IfcPropertySet Name="Pset_ColumnCommon">
    <IfcSimpleProperty Name="Description" Type="IfcString">
      <TypeEnum name="A1010.010 Pad Footings" />
      <TypeEnum name="A2020.020 Pilasters" />
    </IfcSimpleProperty>
  </IfcPropertySet>
  <IfcPropertySet Name="Pset_SlabCommon">
    <IfcSimpleProperty Name="Description" Type="IfcString">
      <TypeEnum name="A1030.010 Slab on Grade" />
    </IfcSimpleProperty>
  </IfcPropertySet>
</Ifc_Pset_TypeEnums>
```

```

        <TypeEnum name="A1030.020 Struct Slab on Grade" />
        <TypeEnum name="C3030.110 2x4 Acstc ClnG by SF" />
    </IfcSimpleProperty>
</IfcPropertySet>
<IfcPropertySet Name="Pset_WallCommon">
    <IfcSimpleProperty Name="Description" Type="IfcString">
        <TypeEnum name="A1010.020 Wall Footings" />
        <TypeEnum name="A1010.030 Grade Beams" />
        <TypeEnum name="B2010.430 Wood Frmd 2x4 wSiding" />
    </IfcSimpleProperty>
</IfcPropertySet>
<IfcPropertySet Name="Pset_DoorCommon">
    <IfcSimpleProperty Name="Description" Type="IfcString">
        <TypeEnum name="B1012420-CIP Door & Salb-One Way" />
        <TypeEnum name="B1012422-CIP Door & Salb-Two Way" />
    </IfcSimpleProperty>
</IfcPropertySet>
<IfcPropertySet Name="Pset_BeamCommon">
    <IfcSimpleProperty Name="Description" Type="IfcString">
        <TypeEnum name="B1010420-CIP Beam & Salb-One Way" />
        <TypeEnum name="B1010430-Composite Beam & Salb" />
    </IfcSimpleProperty>
</IfcPropertySet>
<IfcPropertySet Name="Pset_WindowCommon">
    <IfcSimpleProperty Name="Description" Type="IfcString">
        <TypeEnum name="B101320-CIP Window & Salb-One Way" />
        <TypeEnum name="B1013430-Composite Window & Salb" />
    </IfcSimpleProperty>
</IfcPropertySet>
</Ifc_Pset_TypeEnums>

```

## Configuring PSET Data

### IFC Property Sets (PSET)

The IFC Object Model<sup>3</sup> describes a set of well defined ways of identifying information that defines an object.

However, there are many types of information that users might want to exchange that are not currently included within the IFC Model.

For this purpose the IFC Model provides the Property Definition mechanism. A Property Definition is a generic mechanism that allows model users to define expandable properties for objects.

ArchiCAD's IFC Add-on allows users to define and exchange custom sets of properties for selected ArchiCAD elements.

### IFC 2.00 PSET configuration:

The IFC 2.0 add-on manages custom property data through a PSET *configuration file*, where the user can define the mapping between the Ifc Properties and ArchiCAD door, window, object, lamp and zone element parameters.

The configuration file is an XML<sup>4</sup> format file, named "config\_ifc200.xml". The add-on locates this file in <ArchiCAD folder>:XML\_IFC\_WP folder.

The configuration file contains one <Ifc\_Pset\_Config> structure which contains a mandatory <Usage> structure and may contain optional <Groups>, <Link> or <Mapping> structures.

<sup>3</sup> Refer to the IFC2x Model Implementation Guide for a detailed description of IFC Properties and related concepts.

<sup>4</sup> Refer to <http://www.wc3.org> for details of XML standards.

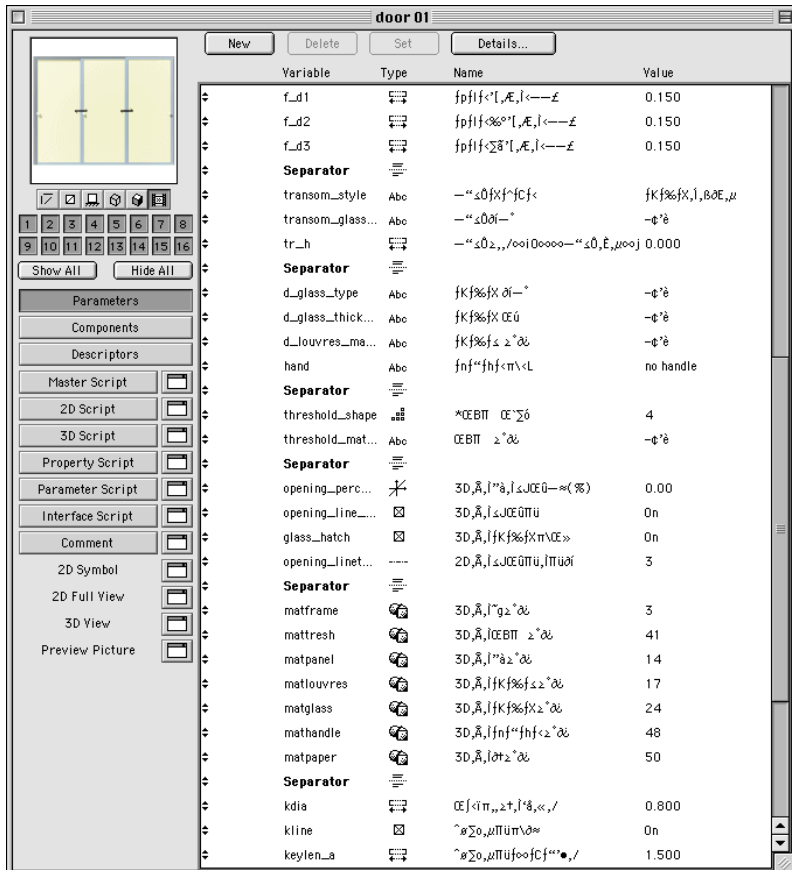


Figure 6: An ArchiCAD element, eg type "Door", with many custom properties can be exchanged in an IFC file using the PSET mechanism.

**Note:** The <Usage> definition must be the first item in the structure

The configuration file may define different PSET mappings for the following three processes:

**Save (export)** - the Add-On saves ArchiCAD library part parameter data in user defined property sets in the new IFC file.

**Load (Open or import)** - the Add-On adds to ArchiCAD library part parameters, the property set data in the selected IFC file.

**Update** - the Add-On updates library part parameters in an existing ArchiCAD plan file using the selected IFC file.

The PSET configuration file format is generally:

```

<?xml version="1.0"?>
<Ifc_Pset_Config>
  <Usage>
    <AbsentWarning Value = "on"/>
  </Usage>
  <Groups>
    < Group Name="Door_set1">
      <LibName Type="Door" Name="door 01"/>
      <LibName Type="Door" Name="door 02"/>
    </ Group>
    ...
  </ Groups >
  <Link>
    <IfcPropertySet Name="Pset_Occurence_Ctrl">
      <IfcSimpleProperty Name="unique_id" Type="IfcString"
ACParamName="@_UID" ACParamType="Integer"/>
    </IfcPropertySet>
  </Link>
  <Mapping>
    <Command Name="Load">
      < Group Name="Door_set1">
        <IfcPropertySet Name="Pset_Door_GS">
          <IfcSimpleProperty Name="hand"
Type="IfcString" ACParamName="hand" ACParamType="String" />
          ...
        </IfcPropertySet>
      </ Group>
    </Command>
  </Mapping>
</Ifc_Pset_Config>
<?xml version="1.0"?>

```

## PSET Configuration Structure

### <Usage>:

The <Usage> structure defines when the Add-on should report errors during processing of IFC files, on import or export.

Errors may occur in either of the following cases; an object has been specified in the PSET configuration but is not in the database, or an object in ArchiCAD has not been included in the PSET configuration.

The <Usage AbsentWarning Value=*onOrOff*> structure, has two possible values for *onOrOff*, “On” or “Off” to report errors

**Note:** See also <Mapping>: **Converting Parameters’ Units of Measure** below for additional processing reporting options.

### <Groups>:

The structure <Groups> contains a list of <Group> definitions. A <Group> is a named set of ArchiCAD library parts sharing a common mapping to a named set of element parameters. See <Mapping> below for details of the mappings.

For example (referring to the example above):

"Door\_set1" contains the doors (door 01, door 02, ... door 16)

"Shutter\_setA" contains the shutters (shutter 01, ...)

"Window\_Contract2" contains all the windows (combination\_1, ..., multiwindow 3x4...), however

"multiwindow 3x4" contains only the multiwindow 3x4 library part. An ArchiCAD element may be a part of more than one group. For example, the multiwindow 3x4 library part is a member of both "Window\_Contract2" and "multiwindow 3x4".

The <Group> definition is:

```
<Group Name=name>
```

where *name* is the user given name of the group

The library part definition is:

```
<LibName Type=type Name=libPartName>
```

where *type* is the ArchiCAD element type (“Door”, “Window”, “Object”, “Lamp” or “Zone”) and *libPartName* is the library part name.

**Note:** *name* and *libPartName* in both structures are case sensitive!

**Note:** ArchiCAD Wall, Beam, Column and Stair elements are not supported for custom property definitions.

```
<?xml version="1.0"?>
  <Ifc_Pset_Config>
    <Groups>
      <Group Name="Door_set1">
        <LibName Type="Door" Name="door 01"/>
        <LibName Type="Door" Name="door 02"/>
        ...
        <LibName Type="Door" Name="door 16"/>
      </Group>
      <Group Name="Shutter_setA">
        <LibName Type="Door" Name="shutter 01"/>
        ...
      </Group >
      <Group Name="Window_Contract2">
        <LibName Type="Window" Name="combination_1"/>
        <LibName Type="Window" Name="combination_2"/>
        ...
        <LibName Type="window" Name="multiwindow
1x4"/>
      </Group >
      < Group Name="multiwindow 3x4">
        <LibName Type="window" Name="multiwindow
3x4"/>
      </Group >
      ...
    </Groups>
    ...
  </Ifc_Pset_Config>
</?xml version="1.0"?>
```

Library part names can also be specified using use wildcard definitions like "door??" or "door ? wide" or "door\*" If you use "?" then the program accepts any character in place of ?, similarly "\*" at the end of the name includes all the names beginning with the defined string (“door”).

For example

"door??" = "door 1", "door12", "doorXY", but not “doortype 1”

"door ? wide" = "door 1 wide", "door 2 wide", "door X wide", but not “door1 wide”

"door\*" = "door", "door1", "doorAnyTailPart", but not “doubledoor”

See the detailed example following.

## <Link>

```
<Ifc_Pset_Config>
...
  <Link>
    <IfcPropertySet Name="Pset_Occurrence_Ctrl">
      <IfcSimpleProperty Name="unique_id" Type="IfcString"
ACParamName="@_UID" ACParamType="Integer"/>
    </IfcPropertySet>
  </Link>
...
</Ifc_Pset_Config>
```

The structure of <IfcPropertySet> is:

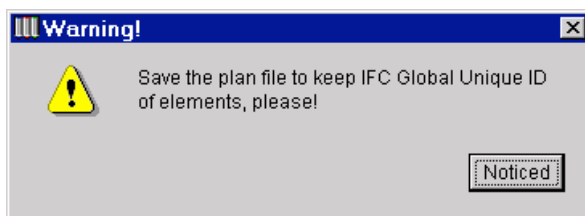
```
<IfcPropertySet Name=name Type=ifcDataType
                ACParamName=parameterName
                ACParamType=parameterType/>
```

This feature defines the link between the IFC entities and ArchiCAD elements. The link is the ArchiCAD unique ID which is saved in the IFC PSET "Pset\_Occurrence\_Ctrl",

- the IFC Simple Property *name* is "unique\_id",
- the IFC data type *ifcDataType* is "IfcString",
- the ArchiCAD parameter name *parameterName* is "@\_UID", and
- the ArchiCAD parameter type *parameterType* is "Integer".

The "@\_UID" string references the ArchiCAD uid but this information is not used, because the Add-On saves the IFC Global unique ID in the appropriate ArchiCAD element's own data section.

The link works automatically if the plan file is saved following the (exporting) saving of the IFC file, for if this is not done the new GUIDs are lost..



A warning Dialog Box reminds the user to save the ArchiCAD database.

## <Mapping>

The structure <Mapping> defines the assignment of the named library part groups to IFC property sets. Mappings are configured for any of the three functions of **Save**, **Load** and **Update**.

### <Mapping>

```
<Command Name="Save">
  <Group Name="Door_set1">
    <IfcPropertySet Name="Pset_Door_GS">
      <IfcSimpleProperty Name="hand" Type="IfcString"
ACParamName="hand" ACParamType="String"/>
      <IfcSimpleProperty Name="opening_angle_3D"
Type="IfcReal" ACParamName="opening_angle_3D" ACParamType="real"
Conversion="Length" Optional="true"
CalcIfcVal="2*(ACParam+10.0)"/>
    </IfcPropertySet>
  </Group>
...
</Command>
```

### </Mapping>

The structure <Command Name=*addonFunction*> defines for a particular function the mapping conversions that will be processed. The values of *addonFunction* can be "Save", "Load" or "Update"

The structure <Group Name=*libraryPartGroupName*> defines the list of property set definitions to be mapped for the named library part group. The *libraryPartGroupName* must match the Group *name* defined in the <Groups> structure.

The structure <IfcPropertySet Name=*psetName*> defines the name of the Ifc PropertySet and contains the list of mapped Ifc Simple Properties.

The structure <IfcSimpleProperty Name=*propertyName* Type=*ifcDataType* ACParamName=*parameterName* ACParamType=*parameterType*/> defines the IFC property to ArchiCAD parameter mapping.

*propertyName* is the name of the IFC Simple Property

*ifcDataType* is the IFC data type and can be: "IfcBoolean", "IfcInteger", "IfcReal" or "IfcString")

*parameterName* is the ArchiCAD parameter name.

**Note:** The *libraryPartGroupName*, *psetName*, *propertyName* and *parameterName* are case sensitive!

There are some reserved strings: “@\_A” and “@\_B” refer to parameter A and parameter B (see **IFC Reference Guide Customizing the IFC Data – Edit selected item IFC properties**, and **Appendix E: PSET Data** for the method to edit PSET data and the default IFC Property Sets).

If the parameter is an array the format is:

*parameterName(index1) [(index2)]*

e.g.: "turn\_width(2)(3)"

*parameterType* is the data type of the ArchiCAD parameter and may be one of "Boolean", "Integer", "Real" or "String")

The above tags are mandatory.

## Optional Tags

### Calculating Parameter Values

CalcIfcVal or CalcACVal are optional parameters that are used to calculate the designated IFC property value by defining a formula and referencing the ArchiCAD parameter with the keyword “ACParam” or “IFCParam”.

You can use fixed point signed numbers, '+', '-', '/', and '\*' operators and '(', ')' parentheses to construct a formula.

**Note:** Use CalcIfcVal and ACParam for the **Save** command and CalcACVal and IFCParam for the **Load** and **Update** commands.

The following expressions are legal:

CalcIfcVal = "2\*(ACParam+10.0)"

CalcIfcVal = "((ACParam-0.52)\*(ACParam+10.0)-34.564)/2"

CalcACVal = "(( IFCParam -0.52)\*( IFCParam +10.0)-34.564)/2"

CalcACVal = "-1"

CalcIFCVal = "-1.0 \* 12"

If you use the Conversion option (see **Converting Parameters' Units of Measure** below) then in the case of each Command function:

**Save:** the calculation is run first and the conversion after!

**Load:** the conversion is run first and the calculation after!

**Update:** the conversion is run first and the calculation after!

### Converting Parameters' Units of Measure

The tag “Conversion” is used to carry out units translation.

Conversion=*unitType*

where *unitType* may be one of “Length”, “Angle”, “Area” or “Volume”. (See the IFC Reference Guide: **Customizing the IFC data- Options...**)

For example:

```
<IfcSimpleProperty Name="opening_width" Type="IfcReal"
  ACParmName="f_w"
  ACParmType="real"
  Conversion="Length"/>
```

**Note:** The Ifc parameters are converted according to the **Tools, IFC 2.00, Options, Units** settings.

### Checking for parameters during processing

The tag “Optional” allows two modes of error reporting when processing property set data.

Optional=*trueOrFalse*

where *trueOrFalse* may be either “True” or “False”

```
<IfcSimpleProperty Name="horizontal_division" Type="IfcInteger"
  ACParmName="horizontal_division"
  ACParmType="Integer"
  Optional="True"/>
```

When Optional=”True”, if the selected parameter is present in the ArchiCAD element or in the IFC Pset its information is processed, otherwise if the parameter is absent then no error message is provided.

When Optional=”False”, or the tag is not defined in the mapping – the default case, a missing parameter causes an error message to be reported (see **Reporting** below).

The structure <Command Name=*addonName*> applies to all three processing options.

## Using PSET Data

The PSET configurations described above modify the actions of the two standard IFC functions of **Save as...** (exporting) and **Open...** (importing) an IFC file. An additional function **Update** provides for editing existing library part parameters in a working ArchiCAD database.

## IFC 2.00 Options – Property Sets

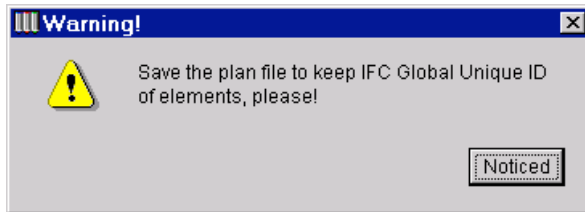
See **Customizing the IFC Data – Options** to configure the settings for XML PSET mapping and data files.

### Save as IFC file...

See **IFC Reference Guide Exporting & Importing IFC Files – Save As IFC file** for the method to export the IFC file.

As the IFC file is created, the Add-On checks for PSET function definitions in the XML configuration file. If both are present, (the Configuration file and a mapping definition for the “Save” function), it adds the library part parameter data to the IFC file according to the mapping specification.

At the conclusion of the command, the user is reminded to save the ArchiCAD database to store the IFC GUID data created on export.



### Load PSET data...

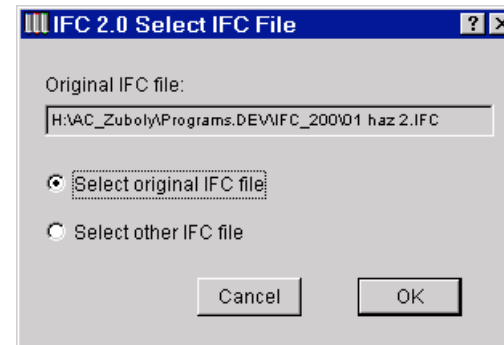
See **IFC Reference Guide Exporting & Importing IFC Files – Open IFC file:** for the method to open (import) an IFC file.

As the IFC file is read, the Add-On checks for PSET function definitions in the XML configuration file. If both are present, (the Configuration file and a mapping definition for the “Load” function), it adds the PSET data to the new ArchiCAD library parts according to the mapping specification.

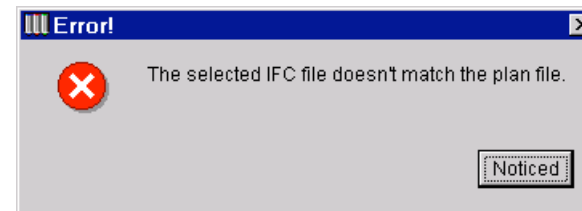
## Update with configured PSET...

When the **Tools, IFC 2.00, Update with configured PSET...** is selected, the Add-On checks for PSET function definitions in the XML configuration file. If both are present, (the Configuration file and a mapping definition for the “Update” function), it adds the PSET data to the existing ArchiCAD library parts according to the mapping specification.

**Note:** This updates the configured parameters only.



If the selected IFC file does not match the GUID data in the ArchiCAD database, an error is advised and the user prompted to select the correct IFC file.



## Reporting:

The Ifc Add-On creates reports for the three PSET processes:

For example: **Update**

```
--Update ArchiCAD Element parameters--  
typeID: 4 uid: 60 Ifc instNum: #8413 Ifc GUID: frG~XdzBH<0RViJs0$MU
```

Error: 'Property not found'! Libpart name: 'shutter 04' Parameter name: 'p\_10' type: 'real' PropertySet name: 'Pset\_ShutterCommon\_Jpn' Property name: 'slat\_panel\_width' type: 'IfcReal'.  
Error: 'Undefined Parameter'! Libpart name: 'window 21' Parameter name: 'jw\_left2' type: 'real' PropertySet name: 'Pset\_WindowCommon\_Jpn' Property name: 'jamb\_width\_side' type: 'IfcReal'.  
Error: 'Parameter type mismatch'! Libpart name: 'multi-corner-window 3x3' Parameter name: 'turn\_width' type: 'real' PropertySet name: 'Pset\_MultiWindowCommon\_Jpn' Property name: 'turn\_width' type: 'IfcReal'.  
Error: 'Parameter dimension failure'! Libpart name: 'window 21' Parameter name: 'jw\_left2' type: 'real' PropertySet name: 'Pset\_WindowCommon\_Jpn' Property name: 'jamb\_width\_side' type: 'IfcReal'.  
Error: 'String length failure'! Libpart name: 'window 21' Parameter name: 'jw\_left2' type: 'real' PropertySet name: 'Pset\_WindowCommon\_Jpn' Property name: 'jamb\_width\_side' type: 'IfcReal'.  
Error: 'Write property failure'! Libpart name: 'window 21' Parameter name: 'jw\_left2' type: 'real' PropertySet name: 'Pset\_WindowCommon\_Jpn' Property name: 'jamb\_width\_side' type: 'IfcReal'.  
Error: 'Parsing function failed: '2\*(ACParam+10.0)'  
IFCParam requested...  
Error: 'Parsing function failed: '2\*(IFCParam+10.0)'  
ACParam requested...  
Error: 'Parsing function failed: '3\*5'  
Syntax error

The Add-On cannot write the property information (program failure).

### **Parsing function failed**

The add-on detected a syntax error in the calculation formula or was missing a parameter value

## **Error messages**

### **Property not found**

The specified property was not found in the IFC PSET definition.

### **Undefined Parameter**

The specified parameter was not found in the ArchiCAD library part.

### **Parameter type mismatch**

String to real conversion was unsuccessful or the parameter was an array and no index was defined (eg "turn\_width(2)(3)") ??

### **Parameter dimension failure**

The index was higher than the limit.

### **String length failure**

String length was longer than 255 byte.

### **Write property failure**

## Example

<?xml version="1.0"?>

```
<Ifc_Pset_Config>
  <Usage>
    <AbsentWarning Value = "On"/>
  </Usage>
  <Groups>
    <Group Name="Door_set1">
      <LibName Type="Door" Name="door 01"/>
      <LibName Type="Door" Name="door 02"/>
    </Group>
    <Group Name="Window_Contract2">
      <LibName Type="window" Name="multiwindow 3x4"/>
      <LibName Type="window" Name="window 01"/>
      <LibName Type="window" Name="window 02"/>
    </Group>
    <Group Name="multiwindow 3x4">
      <LibName Type="window" Name="multiwindow 3x4"/>
    </Group>
  </Groups>
  <Link>
    <IfcPropertySet Name="Pset_Occurrence_Ctrl">
      <IfcSimpleProperty Name="unique_id" Type="IfcString"
        ACPParamName="@_UID"
        ACPParamType="Integer"/>
    </IfcPropertySet>
  </Link>
  <Mapping>
    <Command Name="Save">
      <Group Name="Door_set1">
        <IfcPropertySet Name="Pset_Door_GS">
          <IfcSimpleProperty Name="hand"
            Type="IfcString" ACPParamName="hand"
            ACPParamType="String"/>
          <IfcSimpleProperty Name="opening_angle_3D"
            Type="IfcReal"
            ACPParamName="opening_angle_3D"
            ACPParamType="real" Conversion="Angle"
            Optional="True"/>
          <IfcSimpleProperty Name="glass_hatch"
            Type="IfcBoolean" ACPParamName="glass_hatch"
            ACPParamType="bool"/>
        </IfcPropertySet>
      </Group>
    </Command>
  </Mapping>
</Ifc_Pset_Config>
```

```
<IfcSimpleProperty Name="matframe"
  Type="IfcInteger" ACPParamName="matframe"
  ACPParamType="Integer"/>
<IfcSimpleProperty Name="keylen_a"
  Type="IfcReal" ACPParamName="keylen_a"
  ACPParamType="Real"/>
<IfcSimpleProperty Name="tfont"
  Type="IfcString" ACPParamName="tfont"
  ACPParamType="String" Optional="True"/>
</IfcPropertySet>
<IfcPropertySet Name="Pset_DoorCommon_Jpn">
  <IfcSimpleProperty Name="type_name"
    Type="IfcString" ACPParamName="d_type_name"
    ACPParamType="String"/>
  <IfcSimpleProperty Name="type_no"
    Type="IfcString" ACPParamName="type_no"
    ACPParamType="String"/>
</IfcPropertySet>
</Group>
<Group Name="Window_Contract2">
  <IfcPropertySet Name="Pset_Window_GS">
    <IfcSimpleProperty Name="kline"
      Type="IfcBoolean" ACPParamName="kline"
      ACPParamType="Bool"/>
    <IfcSimpleProperty Name="keylen_a"
      Type="IfcReal" ACPParamName="keylen_a"
      ACPParamType="Real"/>
    <IfcSimpleProperty Name="tfont"
      Type="IfcString" ACPParamName="tfont"
      ACPParamType="String"/>
  </IfcPropertySet>
</Group>
<Group Name="Window_Contract2">
  <IfcPropertySet Name="Pset_WindowCommon_Jpn">
    <IfcSimpleProperty Name="type_name"
      Type="IfcString"
      ACPParamName="w_type_name"
      ACPParamType="String"/>
    <IfcSimpleProperty Name="type_no"
      Type="IfcString" ACPParamName="type_no"
      ACPParamType="String"/>
  </IfcPropertySet>
</Group>
<Group Name="multiwindow 3x4">
  <IfcPropertySet Name="Pset_MultiWindowCommon_Jpn">
    <IfcSimpleProperty Name="thickness_mume_1"
      Type="IfcReal" ACPParamName="th_mume(1)"
      ACPParamType="Real" Conversion="Length"/>
  </IfcPropertySet>
</Group>
```

```

        <IfcSimpleProperty Name="thickness_mume_2"
        Type="IfcReal" ACPParamName="th_mume(2)"
        ACPParamType="Real" Conversion="Length"/>
        <IfcSimpleProperty Name="turn_width_1"
        Type="IfcReal"
        ACPParamName="turn_width(1)(1)"
        ACPParamType="Real" Conversion="Length"/>
        <IfcSimpleProperty Name="open_width_12"
        Type="IfcReal"
        ACPParamName="open_width(3)(4)"
        ACPParamType="Real" Conversion="Length"/>
    </IfcPropertySet>
</Group>
</Command>
<Command Name="Update">
    <Group Name="Door_set1">
        <IfcPropertySet Name="Pset_Door_GS">
            <IfcSimpleProperty Name="glass_hatch"
            Type="IfcBoolean" ACPParamName="glass_hatch"
            ACPParamType="bool"/>
            <IfcSimpleProperty Name="matframe"
            Type="IfcInteger" ACPParamName="matframe"
            ACPParamType="Integer"/>
            <IfcSimpleProperty Name="keylen_a"
            Type="IfcReal" ACPParamName="keylen_a"
            ACPParamType="Real"/>
            <IfcSimpleProperty Name="tfont"
            Type="IfcString" ACPParamName="tfont"
            ACPParamType="String" Optional="True"/>
        </IfcPropertySet>
    </Group>
    <Group Name="Door_set1">
        <IfcPropertySet Name="Pset_DoorCommon_Jpn">
            <IfcSimpleProperty Name="type_name"
            Type="IfcString" ACPParamName="d_type_name"
            ACPParamType="String"/>
            <IfcSimpleProperty Name="type_no"
            Type="IfcString" ACPParamName="type_no"
            ACPParamType="String"/>
        </IfcPropertySet>
    </Group>
    <Group Name="Window_Contract2">
        <IfcPropertySet Name="Pset_Window_GS">
            <IfcSimpleProperty Name="kline"
            Type="IfcBoolean" ACPParamName="kline"
            ACPParamType="Bool"/>
        </IfcPropertySet>
    </Group>
</Command>

```

```

<Group Name="Window_Contract2">
    <IfcPropertySet Name="Pset_WindowCommon_Jpn">
        <IfcSimpleProperty Name="type_name"
        Type="IfcString"
        ACPParamName="w_type_name"
        ACPParamType="String"/>
        <IfcSimpleProperty Name="type_no"
        Type="IfcString" ACPParamName="type_no"
        ACPParamType="String"/>
    </IfcPropertySet>
</Group>
<Group Name="multiwindow 3x4">
    <IfcPropertySet Name="Pset_MultiWindowCommon_Jpn">
        <IfcSimpleProperty Name="thickness_mume_1"
        Type="IfcReal" ACPParamName="th_mume(1)"
        ACPParamType="Real" Conversion="Length"/>
        <IfcSimpleProperty Name="thickness_mume_2"
        Type="IfcReal" ACPParamName="th_mume(2)"
        ACPParamType="Real" Conversion="Length"/>
        <IfcSimpleProperty Name="turn_width_1"
        Type="IfcReal"
        ACPParamName="turn_width(1)(1)"
        ACPParamType="Real" Conversion="Length"/>
        <IfcSimpleProperty Name="open_width_12"
        Type="IfcReal"
        ACPParamName="open_width(3)(4)"
        ACPParamType="Real" Conversion="Length"/>
    </IfcPropertySet>
</Group>
</Command>
<Command Name="Load">
    <Group Name="Door_set1">
        <IfcPropertySet Name="Pset_Door_GS">
            <IfcSimpleProperty Name="hand"
            Type="IfcString" ACPParamName="hand"
            ACPParamType="String"/>
            <IfcSimpleProperty Name="opening_angle_3D"
            Type="IfcReal"
            ACPParamName="opening_angle_3D"
            ACPParamType="real" Conversion="Angle"
            Optional="True"/>
            <IfcSimpleProperty Name="glass_hatch"
            Type="IfcBoolean" ACPParamName="glass_hatch"
            ACPParamType="bool"/>
            <IfcSimpleProperty Name="matframe"
            Type="IfcInteger" ACPParamName="matframe"
            ACPParamType="Integer"/>
        </IfcPropertySet>
    </Group>
</Command>






```

```

                <IfcSimpleProperty Name="keylen_a"
                Type="IfcReal" ACParmName="keylen_a"
                ACParmType="Real"/>
                <IfcSimpleProperty Name="tfont"
                Type="IfcString" ACParmName="tfont"
                ACParmType="String" Optional="True"/>
            </IfcPropertySet>
        </Group>
    <Group Name="Window_Contract2">
        <IfcPropertySet Name="Pset_Window_GS">
            <IfcSimpleProperty Name="kline"
            Type="IfcBoolean" ACParmName="kline"
            ACParmType="Bool"/>
            <IfcSimpleProperty Name="keylen_a"
            Type="IfcReal" ACParmName="keylen_a"
            ACParmType="Real"/>
            <IfcSimpleProperty Name="tfont"
            Type="IfcString" ACParmName="tfont"
            ACParmType="String"/>
        </IfcPropertySet>
    </Group>
</Command>
</Mapping>
</Ifc_Pset_Config>

```

## Appendix A: IFC 2x Model Entities

Icon	ifcEntity	ArchiCAD Element	Conversion		Notes
			AC7	AC8	
	ifcStair	Stair.GSM	Yes	No	
	ifcFlight	Stair.GSM, or Object.gdl	Yes		
	<b>Zone</b>	A zone ( <i>IfcZone</i> ) is an aggregation of spaces ( <i>ifcSpace</i> ), partial spaces or other zones. It is a view based delimited volume for the purpose of analysis and calculation.	No.		
	<b>Space</b>	A space ( <i>IfcSpace</i> ) represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.	No.		
					

### Special Icons



#### Group

A button on the tree dialog which appears when there are more than one item in that Ifc entity class. This is provided to speed up traversing through the data.



#### Toggle

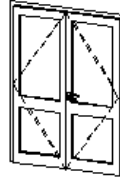
A button to break the current tree view window into two separate dialogs, for easy editing while doing special operations such as creating *ifcZones* for example

There a D D D DJKLLD SKJDLS FJFDS;LF;Ls;lds;jdsdjksjd sksdxsd jnjjd sjsjsjsj

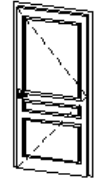
# Appendix A: IFC Door & Window Library

## ifcDoors

IFC\_DOORDBLSWING,



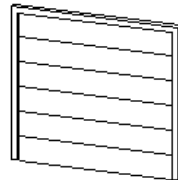
IFC\_DOORSGLSWING,



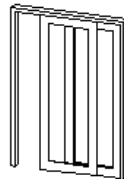
IFC\_DOORREVOLVING,



IFC\_DOORROLLUP,

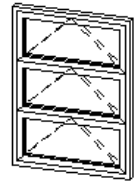


IFC\_DOORSLIDING

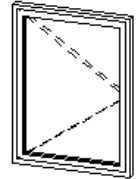


## ifcWindows

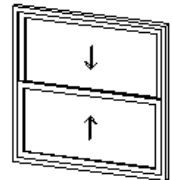
IFC\_WINDOWAWNING



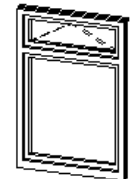
IFC\_WINDOWCASEMENT



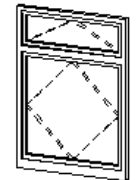
IFC\_WINDOWDOUBLEHUNG



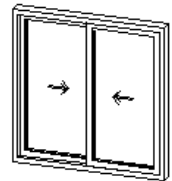
IFC\_WINDOWFIXED



IFC\_WINDOWPIVOTING



IFC\_WINDOWSLIDING



## Appendix C: IFC Entity Support

IFC element	Model Release Version			Notes
	1.51	2.00	2x	
IfcWall	Normal Wall	Normal Wall	Normal Wall or Object	
IfcColumn	Column or Object	Column or Object	Column or Object	
IfcBeam	Beam or Object	Beam or Object	Beam or Object	
IfcDoor	Door Object	Door Object	Door Object or Object	
IfcWindow	Window Object	Window Object	Window Object or Object	
IfcCovering	Object	-	Object	
ifcSlab	-	-	Roof or Object	Type=Roof
“	-	-	Slab or Object	Type=Slab
IfcRoofSlab	Roof	Roof	Roof or Object	
IfcFloor	Slab	Slab	Slab or Object	
IfcSpace	Zone	Zone	Zone	
ifcZone	-	-	Grouping of Zones	The Tree View is used to group spaces to form an ifcZone
IfcBuiltIn	Object	??	-	
IfcDiscreteElement	Object	Object	-	
IfcDistributionElement	Object	Object	-	
IfcElectricalAppliance	Object	Object	-	
IfcEquipment	Object	Object	-	
IfcFixture	Object	-	-	
IfcFurniture	Object	Object	Object	
IfcProxy	Object or Mesh	Object or Mesh		
IfcBuildingElementProxy	-	-	Object or mesh	
IfcDesignGrid	Object	Object		
IfcGrid	-	-	Object	
IfcElectricalAppliance	-	Object	-	
IfcPermeableCovering	-	Object	??	
IfcRailing	-	Object	Object	
IfcRampFlight	-	??	Object	
IfcRamp	-	Object	Object	
IfcStair	-	Stair object	Stair object or Object	
IfcStairFlight	-	Object	IfcStairFlight	
IfcSystemFurnitureElement	-	Object	-	
IfcVisualScreen	-	Object	-	
IfcController	-	Object	Object	
IfcActuator	-	Object	Object	

IfcSensor	-	Object	Object
IfcFlowController	-	Object	Object
IfcFlowFitting	-	Object	Object
IfcFlowSegment	-	Object	Object
IfcFlowTerminal	-	-	Object
IfcCoil	-	-	Object
IfcHydronicHeater	-	-	Object
IfcSanitaryTerminal	-	-	Object
IfcAirTerminal	-	-	Object

---

The additional elements are converted to ArchiCAD objects. The ifc attributes of the above elements are stored in the object's parameters.

## Appendix C: Exported data

### Wall:

**Brick shaped wall** with or without gable(s):

The geometric representation type is **Standard** for Attribute driven extruded segment if there is no cutting needed (without IfcHalfSpace), but it is **Advanced** for Attribute driven extruded segment with IfcHalfSpace(s).

**Revolved wall** with or without gable(s):

The geometric representation type is **Advanced** and described by Attribute driven revolved segment with or without IfcHalfSpace(s).

**Trapezoid or polygon wall** with or without gable(s):

The geometric representation type is **Arbitrary** and that means that a BREP describes the geometry of the object.

The next parameters are saved in the same way for the previous wall types:

The fill of the wall is saved as IfcMaterialLayerSet. Only the thickness and the name of material are saved.

The following parameters are saved as IfcExtendedPropertySet:

LAYERNAME,  
INFO,  
REFMATNAME,  
SIDEMATNAME,  
OPPMATNAME,  
CONTPEN,  
CONTLTYPE,  
CONTPEN3D,  
FILLPEN,  
FILLBGPEN,  
USECOMPENS,  
USECOMPBGPEN

### Wall Connection:

The Add-On stores the wall connection information in the IfcRelConnectsPathElements. This item relates to two wall which are connected, and contains the information of the mode of connection i.e. START, ATEND, ATPATH, and relates to IfcLineConnectionGeometry where the geometry of the common limit is described.

### Column:

Rectangular or Circular column with or without gable(s):

Geometric Representation Item is Standard or Advanced mode of Attribute driven extruded segment with or without IfcHalfSpace.

The fill of the column is saved as IfcMaterialLayerSet. Only the thickness and the name of material are saved.

The following parameters are saved as IfcExtendedPropertySet:

LAYERNAME,  
INFO,  
VENTHICK,  
CORETYPE,  
COREANCHOR,  
COREPEN,  
COREFILL,  
COREFILLPEN,  
COREBGPEN,  
VENPEN,  
VENFILL,  
VENBGPEN,  
CONT3DPEN,  
CONTLTYPE,  
MATER,  
WRAPPING

## Beam:

*Rectangular beam* with or without gable(s):

Geometric Representation Item is Standard or Advanced mode of Attribute driven extruded segment with or without IfcHalfSpace.

The fill of the beam is saved as IfcMaterialLayerSet. Only the thickness and the name of material are saved.

The next parameters are saved as IfcExtendedPropertySet:

LAYERNAME,  
INFO,  
BEAMSTRUCTTYP,  
HOLETYPE,  
HOLECONTUREON,  
PEN,  
LTYPEIND,  
REFPEN,  
REFLTYPE,  
SECTPEN,  
SECTFILLPEN,  
SECTBKGPEN,  
HOLEWIDTH,  
HOLEHEIGHT,  
HOLELEVEL,  
FILLNAME,  
REFMATNAME,  
SIDEMATNAME,  
OPPMATNAME,  
LEFTMATNAME,  
RIGHTMATNAME,  
SHOWALWAYS,  
SHOWABOVE,  
SHOWBELOW

## Door:

If the door is from one of the IFC standard types

IFC\_DOORDBLSWING,  
IFC\_DOORSGLSWING,  
IFC\_DOORREVOLVING,  
IFC\_DOORROLLUP,  
IFC\_DOORSLIDING

then only the opening shape is exported, and the ifc door parameters are exported in the IfcSharedPropertySet and in the IfcOccurrencePropertySet.

If the door is not an IFC standard type then the door shape is exported using the **arbitrary** geometric representation type and that means that a BRep describes the geometry of the object.

If the Opening of the door is rectangle profiled then its geometric representation will be

Standard mode of Attribute driven extruded segment. Otherwise its representation will be advanced mode of Attribute driven extruded with Arbitrary Profile Definition.

The following parameters are saved as IfcExtendedPropertySet:

INFO,  
SILL,  
HEADDEPTH,  
SILLDEPTH,  
JAMBDEPTH,  
JAMBDEPTH2,  
HINGEATSTART,  
SWINGTOINTERIOR

## Window:

If the window is IFC standard type:

IFC\_WINDOWAWNING,  
IFC\_WINDOWCASEMENT,  
IFC\_WINDOWDOUBLEHUNG,  
IFC\_WINDOWFIXED,  
IFC\_WINDOWPIVOTING,  
IFC\_WINDOWSLIDING

then only the opening shape is exported, and the ifc window parameters are exported in the IfcSharedPropertySet and in the IfcOccurrencePropertySet.

If the window is not an IFC standard type then the window shape is exported using the **arbitrary** geometric representation type and that means that a BRep describes the geometry of the object.

If the Opening of the window is rectangle profiled then its geometric representation will be the Standard mode of Attribute driven extruded segment. Otherwise its representation will be the advanced mode of Attribute driven extruded with Arbitrary Profile Definition.

The following parameters are saved as IfcExtendedPropertySet:

INFO,  
SILL,  
HEADDEPTH,  
SILLDEPTH,  
JAMBDEPTH,  
JAMBDEPTH2

## Roof:

If the roof side surfaces are perpendicular to the base surface then a Standard or an Advanced mode of a Attribute Driven Extruded Segment is exported with Arbitrary or Trimmed Curved profile.

Otherwise an Arbitrary Geometric Representation item is exported which means a BRep.

The topMaterial, sideMaterial and bottomMaterial parameters of the roof are saved as an IfcMaterialLayerSet.

The following parameters are saved as IfcExtendedPropertySet:

LAYERNAME,  
INFO,  
SIDETYPE,  
SECTPEN,  
SECTFILLPEN,  
SECTBKG PEN,  
SECTCONTPEN,  
USECOMPPENS,  
USEBKG PEN,  
SECTFILL,  
REFMATNAME,  
SIDEMATNAME,  
OPPMATNAME,  
SHOWALWAYS,  
SHOWABOVE,  
SHOWBELOW

## Slab:

The slab is exported as IfcFloor in a Standard or Advanced Geometric Representation mode using Attribute Driven Extruded Segment where the profile is Arbitrary or Trimmed Curve.

If the slab has any holes, they are exported as IfcOpeningElement.

The topMaterial, sideMaterial and bottomMaterial parameters of the slab are saved as an IfcMaterialLayerSet.

The following parameters are saved as IfcExtendedPropertySet:

- LAYERNAME,
- INFO,
- SECTPEN,
- SECTFILLPEN,
- SECTBKG PEN,
- SECTCONTPEN,
- USECOMPENS,
- USEBKG PEN,
- SECTFILL,
- REFMATNAME,
- SIDEMATNAME,
- OPPMATNAME,
- SHOWALWAYS,
- SHOWABOVE,
- SHOWBELOW

## Object, Lamp:

The objects are generally exported as IfcProxy. The shape representation is Arbitrary meaning a BRep geometric representation item.

The following parameters are saved as IfcExtendedPropertySet:

- LAYERNAME,
- INFO,
- PEN,
- MAT,
- LTYPEIND,
- SECTFILL,
- SECTFILLPEN,
- SECTBGPEN,
- SECTCONTPEN,
- USEOBJMATERIALS,
- ANGLE,
- LEVEL,
- POSX,
- POSY,
- XRATIO,
- YRATIO,
- SHOWALWAYS,
- SHOWABOVE,
- SHOWBELOW

Those objects, which were imported from an IFC file, have an IFC section, where the Add-On registers the original type of the item. These objects are exported as the same kind of elements as they were originally in the IFC file.

## **Zone:**

The Zone is exported as IfcSpace with IfcSpaceBoundary. The IfcSpace geometry is a Standard representation of an IfcAttributeDrivenExtrudedSegment defined by an arbitrary profile. The IfcSpaceBoundary has no geometry if it is a part of a wall. In this case the IfcRelSeparatesSpaces define relation of the wall and the IfcSpaceBoundary. If the IfcSpaceBoundary is logical only (not part of wall), then the IfcSpaceBoundary has an IfcPolyLoop geometry.

## **Appendix D: Imported data**

### **Wall:**

If the IfcWall is IfcAttributeDrivenExtrudedSegment with or without IfcHalfSpace(s) then it will be a straight wall with or without gable(s).

If the IfcWall is IfcAttributeDrivenRevolvedSegment with or without IfcHalfSpace(s) then it will be an arc wall with or without gable(s).

Otherwise the Add-On tries to convert the IfcWall to trapezoid wall. If this is not successful then to polygon wall and if it fails too it converts the wall into an ArchiCAD object registering in the IFC section that the original Ifc type was IfcWall.

The IfcMaterialLayerSet is converted to ArchiCAD fills and composite structure.

If ArchiCAD exported the wall, and the wall contains the original IfcExtendedPropertySet (named 'Graphisoft AC65'), then the Add-On restores these original parameters. The parameters are described above. If the wall doesn't have this kind of IfcExtendedPropertySet then the default parameters remain in the wall item.

The Add-On uses the Wall Connection information to adjust the wall reference line. This feature assures to create real connection between the walls. ArchiCAD connects the wall automatically if the wall reference lines intersect.

### **Column:**

If the IfcColumn is IfcAttributeDrivenExtrudedSegment with or without IfcHalfSpace(s) and the Profile is Rectangle or Circle then it will be a normal column with or without gable(s).

Otherwise the Add-On converts it to an ArchiCAD object item registering in the IFC section that the original Ifc type was IfcColumn.

The IfcMaterialLayerSet is converted to ArchiCAD fills and composite structure.

If ArchiCAD exported the column and the column contains the original IfcExtendedPropertySet (named 'Graphisoft AC65'), then the Add-On restores these original parameters. The parameters are described above. If the column doesn't have this kind of IfcExtendedPropertySet then the default parameters remain in the column item.

## Beam:

If the IfcBeam is IfcAttributeDrivenExtrudedSegment with or without IfcHalfSpace(s) and the Profile is Rectangle then it will be a normal beam with or without gable(s).

Otherwise the Add-On converts it to an ArchiCAD object item registering in the IFC section that the original Ifc type was IfcBeam.

The IfcMaterialLayerSet is converted to ArchiCAD fills.

If ArchiCAD exported the beam and the beam contains the original IfcExtendedPropertySet (named 'Graphisoft AC65'), then the Add-On restores these original parameters. The parameters are described above. If the beam doesn't have this kind of IfcExtendedPropertySet then the default parameters remain in the beam item.

## Door:

If the IfcPropertyTypeDef TypeDefName is a standard IfcDoor type name and the next properties are:

ParameterTakesPrecedence	true
ArbitraryShapeRepresentation	false

then the Add-On uses the standard representation of the door and sets the same parameters to the items as they are in the IfcSharedPropertySet and the IfcOccurrencePropertySet.

The standard IfcDoor types are:

- IFC\_DOORDBLSWING,
- IFC\_DOORSGLSWING,
- IFC\_DOORREVOLVING,
- IFC\_DOORROLLUP,
- IFC\_DOORSLIDING

If the IfcDoor isn't a Standard IfcDoor but the door is exported by ArchiCAD and the IfcDoor contains the original IfcExtendedPropertySet where the Add-On stored the information of the ArchiCAD door item, then the Add-On tries to open the original door object and set the parameters. If the previous action failed but the IfcDoor has geometry, then the Add-On creates a door library part using the geometry and tries to generate a WallHole if the door has a non-rectangular profile.

If the door type is not recognized, and has no geometry (BREP), the IFC\_SimpleDoor is used.

## Window:

If the IfcPropertyTypeDef TypeDefName is a standard IfcWindow type name and the next properties are:

ParameterTakesPrecedence	true
ArbitraryShapeRepresentation	false

then the Add-On uses the standard window item and sets the same parameters to the items as there are in the IfcSharedPropertySet and the IfcOccurrencePropertySet.

The standard IfcWindow types are:

- IFC\_WINDOWAWNING,
- IFC\_WINDOWCASEMENT,
- IFC\_WINDOWDOUBLEHUNG,
- IFC\_WINDOWFIXED,
- IFC\_WINDOWPIVOTING,
- IFC\_WINDOWSLIDING

If the IfcWindow isn't a Standard IfcWindow but ArchiCAD exported the window and the IfcWindow contains the original IfcExtendedPropertySet where the Add-On stored the information of the ArchiCAD window item, then the Add-On tries to open the original window object and sets the parameters. If the previous action failed but the IfcWindow has geometry, then the Add-On creates a window library part using the geometry and tries to generate WallHole if the window has a non-rectangular profile.

If the window type is not recognized and there is no geometry (BREP) the IFC\_SimpleWindow is used.

## Roof:

If the IfcRoof shape can be converted to an ArchiCAD roof then the result will be roof.

Otherwise the Add-On converts it to an ArchiCAD object item registering in the IFC section that the original Ifc type was IfcRoof.

If ArchiCAD exported the roof and the roof contains the original IfcExtendedPropertySet (named 'Graphisoft AC65'), then the Add-On restores these original parameters. The parameters are described above. If the roof doesn't have this kind of IfcExtendedPropertySet then the default parameters remain in the roof item.

## **Slab:**

If the IfcFloor shape can be converted to an ArchiCAD slab then the result will be a slab.

Otherwise the Add-On converts it to an ArchiCAD object item registering in the IFC section that the original Ifc type was IfcFloor.

If ArchiCAD exported the slab and the slab contains the original IfcExtendedPropertySet (named 'Graphisoft AC65'), then the Add-On restores these original parameters. The parameters are described above. If the slab doesn't have this kind of IfcExtendedPropertySet then the default parameters remain in the slab item.

## **Object, Lamp:**

The IfcCovering, IfcBuiltIn, IfcDiscreteElement, IfcDistributionElement, IfcElectricalAppliance, IfcEquipment, IfcFixture, IfcProxy, IfcSite, and IfcDesignGrid are imported as ArchiCAD object registering in the IFC section the original Ifc type.

If the object or lamp was exported by ArchiCAD and it contains the original IfcExtendedPropertySet (named 'Graphisoft AC65'), then the Add-On restores these original parameters. The parameters are described above. If the object or lamp doesn't have this kind of IfcExtendedPropertySet then the default parameters remain in the item.

## **Zone:**

The IfcSpace is imported as an ArchiCAD zone item and the side boundaries will define the limit of the zone.

If ArchiCAD exported the zone and the zone contains the original IfcExtendedPropertySet (named 'Graphisoft AC65'), then the Add-On restores these original parameters. The parameters are described above. If the zone doesn't have this kind of IfcExtendedPropertySet then the default parameters remain in the zone item.



# Contents

CHANGE CONTROL.....	II
FOREWORD.....	III
CONTENTS.....	I
INTRODUCTION.....	2
ABOUT THIS GUIDE.....	2
IFC ADD-ON.....	2
VERSION SUPPORT AND RELEASES.....	2
INSTALLING IFC ADD-ONS.....	2
IFC2X ADD-ONS FOR ARCHICAD 7.0 AND 8.X.....	2
THE IFC INTEGRATED OBJECT MODEL.....	4
WHAT IS AN IFC?.....	4
INTER-OPERABILITY DEFINITION.....	4
WHY USE IFC?.....	4
PLANNING AN IFC EXCHANGE.....	5
DETERMINING THE SCOPE OF A TRANSFER.....	5
VIEWS.....	5
SET UP A PILOT OR TEST.....	6
USEFUL LINKS FOR USING THE IFC EXCHANGE PROTOCOL.....	6
QUICKSTART.....	7
UNITS.....	7
EXPORT.....	7
IFCSite.....	7
Object/Lamp BREP.....	7
Door/Window BREP.....	7
Area/Volume Calculation.....	7
Extended Properties.....	7
Database/Properties.....	7
Rev(olved) Wall with cutting to BREP.....	7
Save space boundaries.....	7
FILTER.....	7
WORKING WITH THE IFC TRANSLATOR.....	8
IFC EXCHANGE PROCESSES.....	8
IFC Objects versus 2D Drawings.....	8
IFC Object Identifiers.....	8

Initiating an IFC Export.....	9
Importing an IFC file.....	9
Exporting a modified imported IFC file.....	9
ARCHIVING IFC DATA.....	10
THE IFC TOOLS MENU.....	10
ArchiCAD 7.0 IFC 2x Add-on.....	10
ArchiCAD 8.0 IFC 2x Add-on.....	10
IFC GUID'S OF SELECTED ELEMENTS.....	10
TREE VIEW... ..	10
UNDERSTANDING IFC MODEL DATA.....	11
THE ARCHICAD DATABASE AND IFC DATA.....	11
THE TREE DIALOG.....	11
THE IFC MODEL.....	12
The IFC Project Data Definition.....	13
CREATING & EDITING MODELS.....	13
Editing Models.....	14
Multiple Building Models.....	15
CREATING THE BUILDING MODEL.....	16
DEFINING SITE OBJECTS.....	16
ARCHICAD ELEMENTS AND IFC ENTITIES.....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
EXPORTING & IMPORTING IFC FILES.....	17
OPEN IFC FILE:.....	17
Reload original IFC file.....	19
Reload other IFC file.....	19
No Reloading.....	20
CUSTOMIZING THE IFC DATA.....	21
OPTIONS.....	21
General.....	21
Units.....	21
EXPORT.....	22
IFCSite.....	22
Save Proxies as BREP [Object/Lamp BREP].....	22
Save Door/Window as BREP [Door/Window BREP].....	22
Area/Volume calculation.....	23
Extended properties.....	23
Database properties.....	23
Revolved wall with cutting to BREP.....	23
PROPERTY SET.....	23
Property Set Configuration file.....	24
FILTER.....	24

CREATE IFC GRID .....	25
CREATE IFC LIBRARY.....	25
EDITING IFC ENTITY PROPERTIES.....	26
ATTIBUTES AND PROPERTIES .....	26
EDIT SELECTED ITEM IFC PROPERTIES ... ..	26
CONVERT TO IFC ELEMENT... ..	27
ASSOCIATE COVERING .....	28
CONFIGURING PSET DATA.....	29
IFC PROPERTY SETS (PSET) .....	33
IFC 2.00 PSET CONFIGURATION:.....	33
PSET CONFIGURATION STRUCTURE .....	35
<Usage>:.....	35
<Groups>:.....	35
<Link>.....	36
<Mapping>.....	36
Optional Tags.....	37
USING PSET DATA.....	38
IFC 2.00 OPTIONS – PROPERTY SETS .....	38
SAVE AS IFC FILE ... ..	38
LOAD PSET DATA... ..	38
UPDATE WITH CONFIGURED PSET... ..	38
REPORTING:.....	38
Error messages .....	39
EXAMPLE .....	40
APPENDIX A: IFC 2X MODEL ENTITIES .....	43
SPECIAL ICONS .....	43
APPENDIX A: IFC DOOR & WINDOW LIBRARY .....	44

IFC DOORS .....	44
IFC WINDOWS .....	44
APPENDIX C: IFC ENTITY SUPPORT .....	45
THE ADDITIONAL ELEMENTS ARE CONVERTED TO ARCHICAD OBJECTS. THE IFC ATTRIBUTES OF THE ABOVE ELEMENTS ARE STORED IN THE OBJECT'S PARAMETERS. ....	2
APPENDIX C: EXPORTED DATA.....	2
Wall:.....	2
Column:.....	2
Beam: .....	3
Door: .....	3
Window: .....	4
Roof:.....	4
Slab: .....	5
Object, Lamp:.....	5
Zone: .....	6
APPENDIX D: IMPORTED DATA.....	6
Wall:.....	6
Column:.....	6
Beam: .....	7
Door: .....	7
Window: .....	7
Roof:.....	7
Slab: .....	8
Object, Lamp:.....	8
Zone: .....	8

## **ArchiCAD IFC 2x Reference Guide**

© 2001-2003 Graphisoft®. All rights reserved. Reproduction, paraphrasing or translation without express prior written permission of Graphisoft is strictly prohibited.

**XXXXXX** Senaati, HUT, CIFE add text

Published by GRAPHISOFT R&D Rt., <http://www.graphisoft.com>

First edition.

Printed in Hungary.

### **Trademarks**

IFC and IAI are trademarks of the International Alliance for Interoperability. ArchiCAD is a registered trademark and PlotMaker, Virtual Building, StairMaker and GDL are trademarks of Graphisoft. Windows is a trademark of Microsoft Inc. Macintosh, Power Macintosh, QuickTime and TrueType are registered trademarks of Apple Computer Inc.

ISBN 963 00 7352 8

## **Change Control**

Draft Version 1.6

# Foreword

The developments in IFC software, functionality and emerging pilots and projects have exceeded expectations. Since the publishing of the first version of this guide, a milestone has been reached for example with the report by CIFE, Stanford on the Helsinki Technical University new Auditorium project, published October 2002. (see <http://www.stanford.edu/group/4D/download/c1.html>).

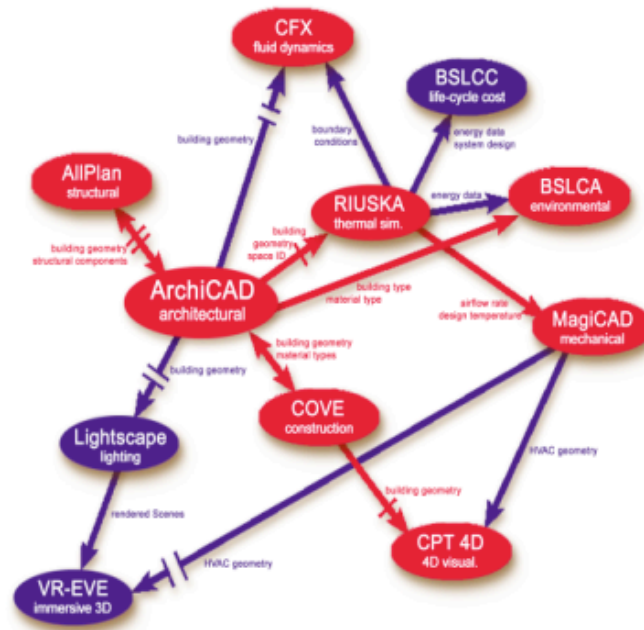
ArchiCAD has been a major supporter of this project; ArchiCAD users have designed and documented the work. **ETC**

The object integrated model, ArchiCAD's innovative concept for over 18 years, puts in place a strategic foundation for a new paradigm of facility development based on the management of information. IFC based exchange now extends this concept to allow many more of the project development participants to create & share comprehensive project information. In contrast with the legacy 2D drawing approach based on a proprietary vendor format, the IFC technology eclipses both of these technology and market limitations by defining an innovative open standard for the description of AEC objects globally available to all in the construction industry.

Companies that have products supporting the IFC protocol enable you to rethink the way you solve common business problems.

The benefits can be any one of the following examples:

- access object data for improved design and coordination in ArchiCAD – getting a building service engineer's HVAC ductwork or water piping layout as 3D objects is one such example.
- export your design for analysis by another specialist package – such as a thermal analysis enabling better understanding of your building performance, or advanced visualization, analysis of the logic of the design etc
- export a model for costing, estimation bidding or procurement – the integrated nature of the IFC information makes your database more valuable because your information is easy to access and analyze by such packages as costing applications
- your model is the definitive source of data for construction planning solutions such as form working applications which identify formed surfaces in the model, apply standard components for formwork assembly access etc and then derive a list of parts and construction schedule.
- extend further your service benefits to clients or users by easily exporting your facility data can into their business systems.



### Legend:

- |  |   |  |   |
|--|---|--|---|
|  | Non IFC-compliant application (Chapter 4.2) |  | IFC-compliant application (Chapter 4.2)         |
|  | One-way info. sharing (Chapter 5.1)         |  | One-way info. sharing through IFC (Chapter 5.2) |
|  | Interventions required (Chapter 5.1)        |  | A few correctable errors (Chapter 6.2)          |
|  | Major errors in sharing (Chapter 6.2)       |  | Two-way info. sharing through IFC (Chapter 5.2) |

Figure 1: Snapshot of major product model applications used by the project team in the PM4D approach (middleware and internal database are omitted).

The figure shows how the project team exchanged product model data between these applications. The figure illustrates clearly the need for the exchange of product model information to support the design of many aspects of a project for many different disciplines and criteria. Note that some of the links that existed at the time of the project (e.g., between ArchiCAD and MagiCAD) were not used by the project team. Furthermore, today some of the links (e.g., between RIUSKA and CFX) are IFC-compliant. *Diagram and text by courtesy CIFE, Stanford.*

Many more opportunities exist to leverage IFC object data for improved design quality, reduction of errors, better coordination and new services to owners and other partners in the development process. See the **IAI Implementer Support Group** website <http://www.bauwesen.fh-muenchen.de/iai/ImplementationOverview.htm>

for descriptions of currently available products.

ArchiCAD's virtual building technology and IFC interface enables you to integrate and coordinate information from these sources; it extends the potential of the Virtual Building concept into real object collaboration with your project partners; and it will improve your level of service and competitive edge in the market place.

We welcome your feedback on several levels – how we have engineered the IFC interface, how you are able to apply it to your business and as this is a partner to partner application, how you make it work in your project team environments.

*Please contact your distributor with any problems or project feedback.*