

Graphisoft GDL

Technical Standards

version 1.0

Graphisoft

Copyright © 2000 by Graphisoft, all rights reserved. Reproduction, paraphrasing or translation without express prior written permission is strictly prohibited.

ArchiCAD and ArchiFM are registered trademarks and SalesCAD, GDL and GDL Object Web Plug-in are trademarks of Graphisoft. All other trademarks are the property of their respective owners.

Guidelines for Professional GDL Object Creators

Contents

- 1. Introduction**
 - 1.1 About this Book

- 2. Designing Libraries**
 - 2.1 Specifications
 - 2.2 Time Calculation

- 3. Making Libraries**
 - 3.1 General Requirements
 - 3.2 Directory Structure
 - 3.3 Names

- 4. Making Objects**
 - 4.1 Parameters
 - 4.2 Master Script
 - 4.3 2D Script, 2D Symbol
 - 4.4 3D Script
 - 4.5 Parameter Script
 - 4.6 Interface Script
 - 4.7 Comment
 - 4.8 Preview Picture
 - 4.9 Doors & Windows
 - 4.10 Macros, Macro Objects, Master GDL

- 5. ArchiFM Libraries**
 - 5.1 General
 - 5.2 Objects
 - 5.3 Zones/Rooms
 - 5.4 Examples

- 6. Web Libraries**
 - 6.1 General Requirements
 - 6.2 Library Conversation

- 7. Windows-Macintosh Compatibility**
 - 7.1 Compatibility Limitations
 - 7.2 Changing Platform

- 8. Library Localization**
 - 8.1 Localization Steps

9. Tools for Library Developers

9.1 Special Menu Commands

9.2 Add-Ons

9.3 Drag & Drop

10. Quality Checking

10.1 The Quality Checking Process

Appendix

A: Standard ArchiCAD Object Parameters

B: Standard GDL Script Format

C: Standard ArchiFM Object Parameters

1. Introduction

1.1 About this Book

The release of new ArchiCAD national versions and the growing Graphisoft product line (Graphisoft Collection, ArchiFM and SalesCAD) have dramatically increased the demand for GDL objects and object libraries. As a result, many independent or third party GDL programmers have begun developing libraries for Graphisoft.

To keep these libraries compatible, and to guarantee the standard of quality people expect from Graphisoft products, we published these guidelines for professional GDL creators. This book also includes tips and tricks, examples, and descriptions of previously undocumented ArchiCAD features. This edition applies to libraries made with ArchiCAD version 6.5.

2. Designing Libraries

2.1 Specifications

- First of all, the purpose and the function of the library part must be clear. Therefore, all parameters must be designed together with their hierarchy, to give a perfect overview of the library part's function: the level of abstraction, features, 2D/3D representation, and the section and elevation shape. The most basic question is: what is this library part used for?
- The complexity of the library part should be carefully considered: a very sophisticated object is also very difficult to use. An element can be a composite of smaller parts, which will reduce the number of parameters and/or the size of the library part. A large library part's modeling process can be very slow. The number of parameters is now limited to 125.
- The parametrizing logic must be reflected in the arrangement of the parameters: groups of parameters should be created, and the connection between them must be defined in the Parameter Script using the "parameters," "range," and "lock" commands. To accomplish this, links between different parameters must be carefully analyzed.
- Automatic reset of parameters (using commands like "range", "parameters") should be used carefully. Keep in mind that:
 1. Automatic changes won't always give the result that the user expects.
 2. A "circular" parameter reset must be avoided (e.g., changing X resets Y, which in turn resets Z, which resets X again).
 3. You can automatically set the default value for parameters using the "GLOB_MODPAR_NAME" (last modified parameter's name) global variable
 4. If you annul a parameter in a certain condition, consider giving a default after further modification of this condition; otherwise you can get an error message like "illegal (0) value of variable..."
- When resetting the value of a parameter in a certain condition in the Parameter Script using the "parameters" command, a similar statement must be put into the Master Script, especially if the modification of the second parameter changes the value of a third (or more) parameter. E.g.:

if xx then parameters yy=1 (Parameter script)
if xx then yy=1 (Master script)
- The abstraction level of the same library part can vary depending on its different representation forms. A sophisticated 2D symbol can sometimes be

used as a simple 3D model. These types of decisions must be made before the other phases of specification.

- Detailed specification drawings must be prepared before the creation of a new library is started.
- Specification drawings must contain every version of each unique library part:
 1. They must be drawn in different scales if the 2D/3D models will change depending on the scale.
 2. Independent versions must be drawn for windows/doors, both with and without composite parts (sidelights, transoms, wall joints).
 3. They should contain plan symbol, elevation and section view in a consistent way.
 4. All parameter names and all fixed sizes must be shown on each drawing.
 5. For further modifications, even for fixed sizes, use a variable on the drawing and in the scripts, and set its value in the Master Script to make it available both for 2D and 3D scripts. To avoid unexpected situations, an additional or internal parameter can be made for all attributes (e.g., pen or fill number) that may be changed from one version to the other.
 6. The drawings must always show the model as it should appear. To avoid misunderstanding, sketches that are less detailed than the required result cannot be accepted.
- The specification drawing must contain everything that is important for the client. The programmer will not complete what is not shown on specification drawings. Therefore, good communication is needed between client and programmer to avoid having to make serious modifications in an advanced phase of the library development.
- To represent the requests and possible solutions in an advanced form, a prototype should be created for one or more basic elements of the library.
- Prototypes must be checked very carefully. They should contain all parameters, options and features of the final library part. Very often, this is not possible in an early stage of the development, so library parts must be prepared to easily handle modifications. For this reason, a library part should never be considered as an ultimate solution for the client's requests.

2.2 Time Calculation

The time needed for making libraries can vary widely.

The most important factors are:

- Number of elements in the library
- Complexity of the 3D model
- Complexity of the 2D representation

- Complexity of the special scripts (interface script, property script, parameter script)
- Number of similar elements in the library

The following formula can be used to estimate the number of working days needed to make a library:

$$T_I = \text{SUM}(T_{Oj})$$

T_I : The time needed for making the library

i_{max} : Number of objects in the library

T_O : The time for making an object

$$T_O = (T_{3d} + T_{2d} + T_{vl} + T_{ps} + T_{if} + M_{def}) * m_1$$

Parameters:

T_{3d} = Complexity of the 3D script

- Easy (0.5)

The object is made of basic geometrical elements (cube, cylinder, etc.) with only a few parameters.

Examples: Simplified doors, windows and furniture

- Normal (0.7)

Simple objects with not more than 10 geometrical variations.

Objects made of prism, revolve and cutplane elements.

Examples: Computer, lamp (revolved object), general furniture.

- Complex (1.1...2)

Objects with complex 3D geometry, and highly parametric objects.

Examples: Parametric doors and windows, office chairs, irregular shapes, etc.

T_{2d} = Complexity of the 2D script

- Easy (0.3)

Objects use 2D symbols only. Objects with no more than three plan variations.

The 2D script doesn't contain global variables or difficult calculations.

Examples: Chairs, tables

- Normal (0.5)

Objects with three to six plan variations. The 2D script may contain global variables or some calculations.

Examples: Straight stairs, inclined objects

- Complex (1)

Highly parametric objects with more than six plan variations. Parameters can depend on each other. The 2D script is full of complicated calculations.

Examples: Special shape stairs, parametric doors and windows, parametric building details

T_{vl} = Complexity of the parameters

- Easy (0.1)

Objects that have no more than four or five additional parameters. No parameter script is needed.

Examples: 2D electrical symbols

- Normal (0.2)

Objects that have more than five additional parameters. No more than three parameters can depend on each other. There can be value-listed parameters in the object.

Examples: Simple furniture

- Complex (0.4)

Objects that have more than 20 additional parameters. Several parameters can depend on each other

Examples: Parametric doors and windows, office furniture product lines represented in one object

T_{ps} = Complexity of the property script

- Easy (0.1):

The object calls only predefined components from an existing database and there are no more than 15 of them. The script doesn't contain flow control statements, loops, parameter buffers, etc. The list doesn't use pictures or 3D views of the object

Examples: Article number list of single furniture elements

- Normal (0.2)

The object has more than 15 predefined components or more than three "if" statements or more custom components and descriptors. The list doesn't use pictures or 3D views of the object

Examples: Price list of furniture with different material.

- Complex (0.4):

The object has more than 15 predefined components or more than three "if" statements or more custom components and descriptors. The list contains pictures or 3D views of the object

Examples: Price list of complete office table product line

T_{if} = Complexity of the interface script,

- Normal (0.3):

The object has no more than two interface scripted tab pages.

There are no flow control statements in the script.

Examples: Door panel style setting.

- Difficult ($t_1 = \text{number of pages} \times 0.4$)

The object has more than two interface scripted tab pages.

The script can contain flow control statements, loops, parameter buffers, etc.

Interface pictures have to be created for the object.

Examples: Inclined steel sections with cut at the ends

M_{def} = the time for making materials

- Easy ($0.05*n$):
More than 20 materials have to be created.
- Normal ($0.1*n$):
Less than 20 materials have to be created

m_1 = Correctional factor

This parameter depends on the number of similar elements in the library.

<10: 1

10-20: 0.9

20-: 0.8

Examples: Window library. All objects use the same frame and panel macros.

The interface scripts of the door panels are also the same.

3. Making Libraries

3.1 General Requirements

Libraries released by Graphisoft must:

- be compatible with each other. To ensure compatibility, developers have to follow Graphisoft's library and object standards.
- give the fastest and easiest solutions for the users by using the minimum number of objects and parameters.
- work error free on Windows and Macintosh platforms. Developers have to provide both Windows and Macintosh versions unless otherwise requested by Graphisoft or the customer.

→See chapter 7.

- be organized into logical directory structures. The names of the directories, objects and parameters have to be consistent and have to refer their function.

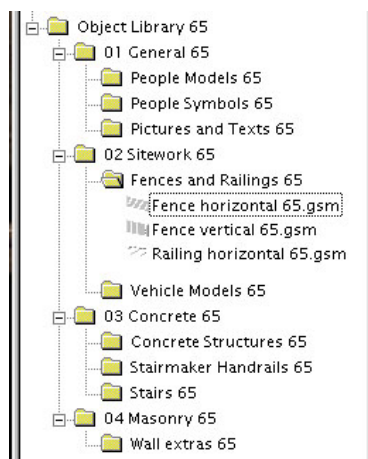
→See chapter 3.2

- work error free under all possible parameter combinations (this applies to the full content). Parameter values and parameter combinations have to be kept between valid limits by parameter scripting.

→See chapter 4.1

3.2 Directory Structure

- Library content has to be kept in a logical and well-arranged directory structure. New ArchiCAD libraries must follow the structure of previous releases, unless Graphisoft advises changes. In other libraries for other products, the directory system has to meet customer's needs and or national standards (e.g., CIS).



- The library system has to be ergonomic. Keep the number of directory levels under six to make library browsing faster.

- Do not mix different types of objects (door, window, macro object, etc.) within the same folder.
- Additional files (macros, macro objects, master GDL scripts, textures, etc.) required by the objects have to be kept in subfolders. When objects in different folders use the same macros or textures, we suggest that you create a „Macro” and a „Texture” folder in the root directory of the library.
- Libraries cannot contain elements with identical names, even if these files have different extensions (e.g., tree.gsm, tree.jpg etc.). Such files have to be renamed.
- Cache files and precompiled binary files (*.gs&, *.do&, *.wi& etc.) must be deleted from final libraries.

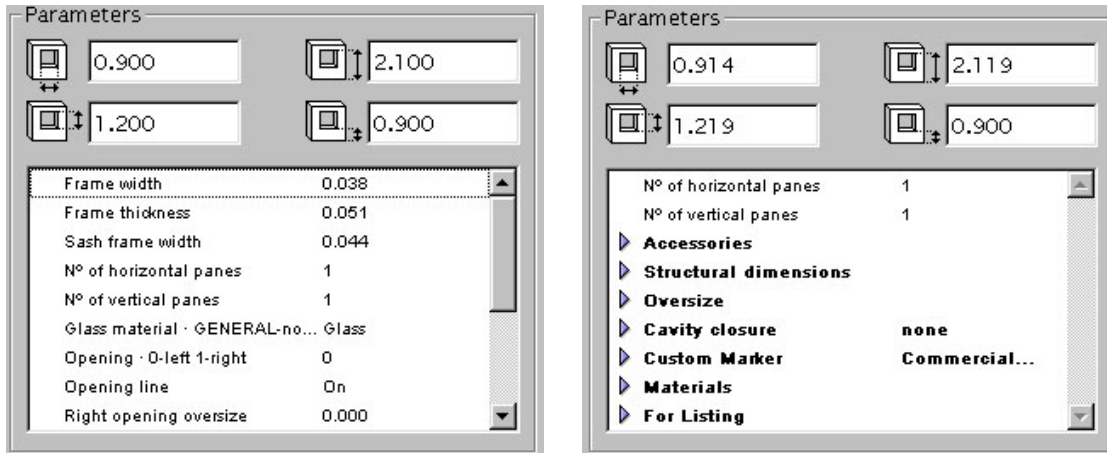
3.3 Names

- File and folder names must refer their function.
- Element or folder names have to be as short as possible to make library browsing faster. Length of the object names cannot exceed 27+3 characters. ArchiCAD won't load files or folders with longer names.
- Do not use special characters in file and folder names (@ # ! \$ % μ ¼ ß etc.).
- All folder and file names in ArchiCAD libraries have to end with a two-digit code that indicates the version of ArchiCAD the library was made for (e.g., Bed 65.gsm).
- All folder and element names in Graphisoft Collection libraries have to end with a three-character code referring the type of the library (e.g., Log Sofa LHL.gsm in Log Home Library).
- In other projects a custom identifying code has to be introduced that relates to the function of the library. If there is a national standard for naming objects or libraries we advice you to use it (e.g., CIS system in the US).
- Use underlines instead of spaces in macro names. This makes macro calling more exact and simplifies further modifications in 3D and 2D scripts when the macro name is changed.

4. Making Objects

4.1 Parameters

- The number of additional parameters in the objects must be reduced to the possible minimum. Users are confused by too many parameters.
- Parameters have to be arranged in a logical system in all library parts. If you have more than 5-10 parameters, arrange them with “title” parameters. See ArchiCAD library parts for reference.



- Identical parameters have to have the same variable and name in all objects to allow modifications on multiple selected objects. Graphisoft’s standard parameter names are used in the international library objects.
- Parameter values and parameter combinations have to be kept between valid limits by parameter scripting in order to avoid error messages caused by invalid parameters.
- Please note that objects that have array value parameters don’t work with the GDL Object Web Plugin.
- We suggest using two to eight characters for variables and no more than 20 characters for parameter names to make the scripts shorter and localization faster. Variable and parameter names should be related with the function of the parameter.

4.2 Master Script

- To reduce file size and to make elements easily modifiable place parameter definitions and calculations used by both 2D and 3D scripts in the master script.
- Place your attribute definitions (material, fill, linetype etc.) in the master script to make further modifications easier.
- Do not put “END” commands in the master script; otherwise, ArchiCAD will not run the parameter script.

- Whenever you redefine “zzyzx” parameter in the master script, be sure to define zzyzx value in the parameter script with the “PARAMETERS” command for correct value display in the object settings window.

Example:

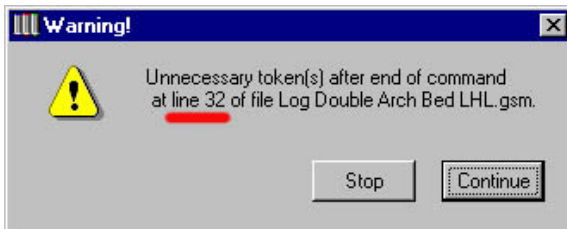
Master Script:

```
tempzzyzx=zzyzx
if zzyzx<35" then zzyzx=35"
```

Parameter Script:

```
if tempzzyzx<35" then parameters zzyzx=35"
```

- Please note that the number of the bad line indicated in ArchiCAD error messages contains the number of the line in the master script.

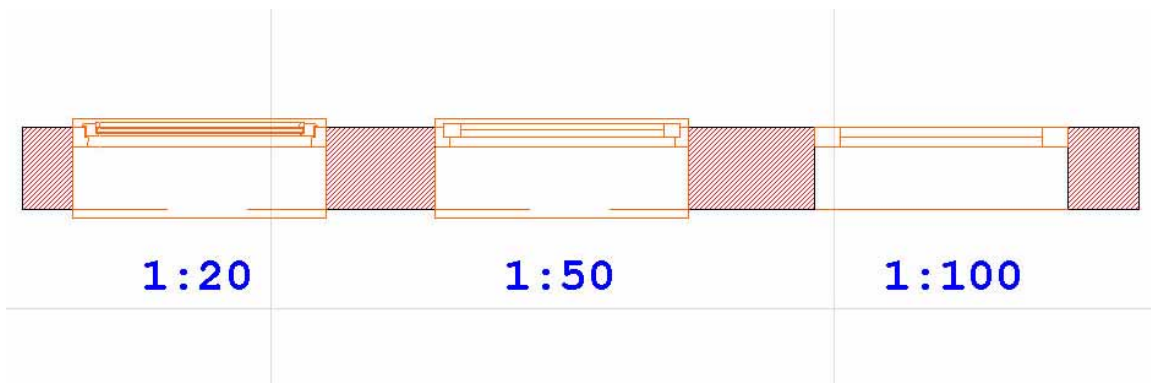


4.3 2D Script, 2D Symbol

- Plan symbols have to be typical of the elements and have to be in compliance with national and international drawing standards.
- Make plan symbols simple and recognizable for faster plan browsing.
- Only the necessary hotspots have to be shown on the symbol. The default hotspot (the first “hotspot2” command in 2D script or the first hotspot placed in 2D symbol) has to be the typical point for placing the element (e.g., the center point of a column).
- Make elements stretchable on plan when it is reasonable (e.g., horizontal beams). When you make stretchable objects, do not put curved fills in the 2D symbol, as they are not displayed correctly on the plan after stretching. In this case, we recommend defining fills in the 2D script.
- Create background fills for all the objects to hide fills and zones under their symbol. Make the background fill type, pen and contour pen adjustable.

↕	Title	☰	Plan symbol	
↕	ms	☒	Minimal space	Off
↕	bft	▨	Background fill type	65
↕	bpn	⌚	Background fill pen	91
↕	cpn	⌚	Contour pen	7

- Plan symbols have to be scale-sensitive whenever possible (e.g., doors and windows).



Custom linetypes, fills and texts can be defined scale sensitive or scale independent. Take this into consideration when making plan symbols.

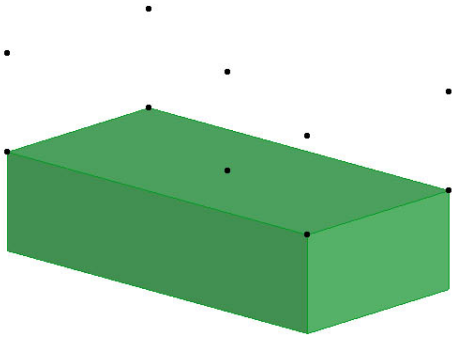
- Try to avoid using the “project2” command as it slows down plan regeneration and you cannot define background fills for the object.
- Try to avoid placing texts in the 2D symbol (especially error messages) in order to make localization easier. Use a 2D script for defining texts in order to make them scale sensitive.
- Add comments (preferably in English) to your scripts to make further changes easier.
- Always check whether the plan symbol covers the 3D model.

4.4 3D Script

- Make 3D models simple and fast. 3D models should not be more detailed than required. Put a 3D on/off switch in all objects to allow users to eliminate unwanted objects from the 3D view.
- Reduce the number of surfaces in your model to the minimum in order to make 3D regeneration faster. Use “RESOL”, “TOLER” and “RADIUS” commands to control segmentation of curved surfaces.
- Note that closed bodies regenerate faster in 3D than open ones (e.g., a cylinder is faster than an open tube).
- Use status codes to control the visibility of the objects in hidden line views. Make the contour lines visible of curved surfaces. Hide unnecessary lines when it is possible.
- Try to avoid using binary format in order to make objects modifiable.
- Make objects stretchable in a vertical direction by using the “zzyzx” parameter when reasonable (e.g., columns). Please note that if you put a “MULZ -1” command before a vertically stretchable object, the object will fall under automatically generated 3D hotspots.

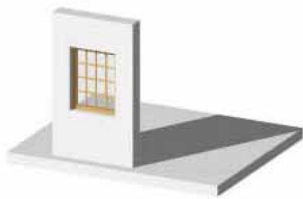
Example:

```
MULZ -1
BLOCK a,b,zzyzx
```



Define hotspots with “HOTSPOT 3” command to avoid this problem.

- Do not use “del top” commands apart from at the end of 3D scripts to make later modifications easier.
- Use the “body -1” command after hollowed 3D shapes to make correct shadow casting if the holes are filled with transparent surfaces (e.g., window sash with grilles).



```

!!!!!!!!!!Frame!!!!!!!!!!!!!!

CALL "ext_rect_frame_65" a,b,c,d
ADDX c
ADDY c
CALL "int_rect_frame_65" a-2*c,b-2*c,e,d

!!!!!!!!!!Grills!!!!!!!!!!!!!!

ADDZ d*3/8
ADDX e
ADDY e
CALL "double_grille_265" a-2*c-2*e,b-2*c-2*e,d/8,f,g
DEL 3
ADDZ d/2
ADDX e
ADDY e
CALL "double_grille_265" a-2*c-2*e,b-2*c-2*e,d/8,f,g

!!!!!!!!!!Glass!!!!!!!!!!!!!!

if h=0 then 30
MATERIAL h
BRICK a-2*c-2*e,b-2*c-2*e,0.01

```



```

!!!!!!!!!!Frame!!!!!!!!!!!!!!

CALL "ext_rect_frame_65" a,b,c,d
ADDX c
ADDY c
CALL "int_rect_frame_65" a-2*c,b-2*c,e,d

BODY -1
!!!!!!!!!!Grills!!!!!!!!!!!!!!

ADDZ d*3/8
ADDX e
ADDY e
CALL "double_grille_265" a-2*c-2*e,b-2*c-2*e,d/8,f,g
DEL 3
ADDZ d/2
ADDX e
ADDY e
CALL "double_grille_265" a-2*c-2*e,b-2*c-2*e,d/8,f,g

BODY -1
!!!!!!!!!!Glass!!!!!!!!!!!!!!

if h=0 then 30
MATERIAL h
BRICK a-2*c-2*e,b-2*c-2*e,0.01

```

- If you use parameter buffers, always empty them with the “GET (NSP)” command after you used all the values of the buffer.

- Do not redefine global variables in your objects, as it affects all other objects using these variables. If it is a necessity, then restore their definitions at the end of the script.
- Always check if texture mapping is applied correctly on your objects. If the default ArchiCAD texture mapping process doesn't produce a good result, use the "COOR" command to set the correct method.



```
TUBE 4,6+nsp/4-2,1+2+16+32,
    1dm/2,0,1,
    0,0,901,
    0,180,4001,
    0,180,4001,

    0,0,-1,0,
    0,0,1cb,0,
    0,0,1ht/n,0,
    get(nsp-8),
    0,0,(n-1)*1ht/n,0,
    0,0,1ht-1ct,0,
    0,0,1ht-1ct+1,0

FOR tem=1 TO nsp
    temp=get(1)
NEXT tem
```

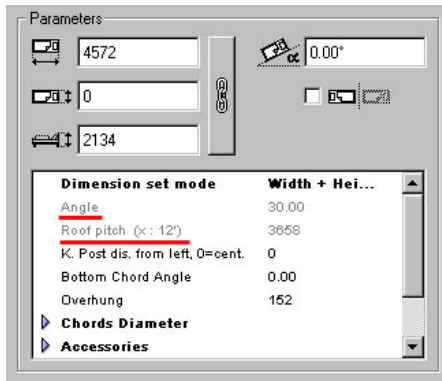
```
BASE
VERT 0, 0, 0
VERT 1, 0, 0
VERT 0, 1, 0
VERT 0, 0, 1
COOR 2+256, -1, -2, -3, -4
BODY -1
```

Please note that if the coordinate box defined by this command is larger than the model, the 3D view of the object will not fit in the preview area in the object settings dialog window.

- Always restore the global coordinate system at the end of the 3D script and follow it with an "END" command to make further modifications on the object easier.
- We suggest the following rules for scripting: Put parameter definitions and calculations at the beginning of the script. Use capital letters for command names. If you make flow control statements, put a tab before each line in the statement. Add comments (preferably in English) to your scripts to make further changes easier. See appendix B for a standard GDL script.

4.5 Parameter Script

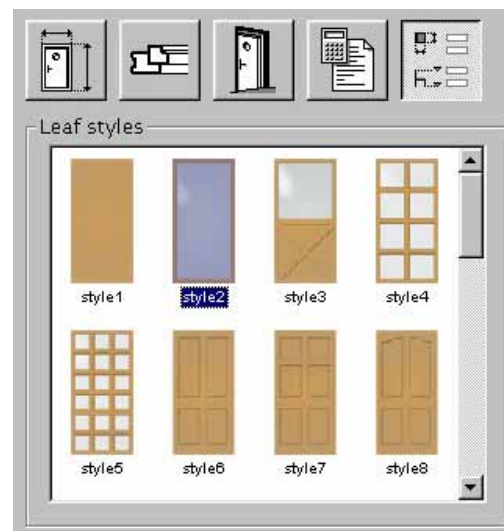
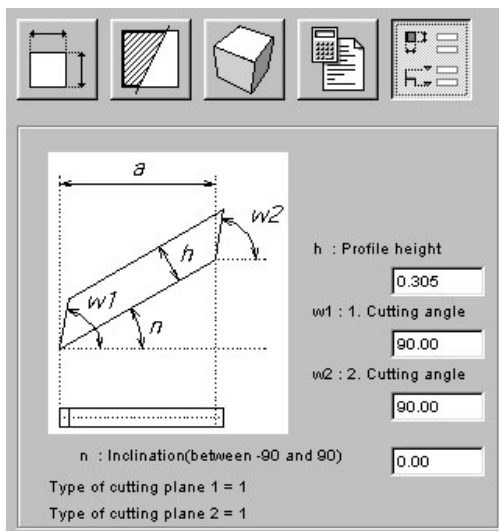
- Use a parameter script to control parameter values and to define value lists. Exclude invalid parameter values and parameter combinations from the parameter list in order to avoid error messages when using the object.
- Lock unusable or fixed parameters with the "LOCK" command. Display real parameter values of the locked parameters using the "PARAMETERS" command.



- When you control parameters in the parameter script, try to follow the order of additional parameters.

4.6 Interface Script

- The Interface Script allows you to define your custom parameter-setting window. You can also add graphical and text type elements to help users in setting parameters. Use this function of ArchiCAD to make library elements user friendly. See the following ArchiCAD library objects for reference: door leaves, steel structures, bitmapped objects.



- Please note that if any character is present in the Interface Script, the GDL Object Web Plug-in will display the interface script window as the default parameter setting dialog.
- If you use the “UI_INFIELD” command to define a thumbnail view field for value lists, be aware of the following. There should be an equivalent thumbnail for all parameter values (including “empty” value). Thumbnails have to be the same size at which they will display, otherwise ArchiCAD will distort them. We advise you to use ArchiCAD’s figure tool for assembling the thumbnails into one picture file. Keep all pictures used by interface scripts in the “Macros” folder.



- If you want the interface script window to be the default parameter setting dialog for the object, push the “Set as Default” button.

4.7 Comment

Fill in every object’s comment window to allow its further identification. The comment window should contain the following information:

File name:	4-Poster Bed LHL.gsm
Version:	ArchiCAD 6.5
Made by:	GDL Factory – Chuck Berry
Date of creation:	April 17, 1999
Modified by:	GDL Factory – James Brown
Date of modification:	June 12, 2000
Notes:	Made for Log home Library

4.8 Preview Picture

- Add preview pictures to all your library elements (including electrical symbols, macro objects) for smart library browsing.
- Picture size has to be 128x128 pixels. Any other size will be displayed distorted in the library browser window.
- Previews have to show typical views of the elements, preferably from the same viewpoint and with the same background.



4.9 Doors, Windows

General

- The same rules apply for doors and windows as for objects.
- 2D and 3D representations of doors and windows have to be based on national standards (opening line, plan symbol in different scales, nominal opening, etc.).
- Make plan symbols scale sensitive.
- Don't forget to fill in the "details" window.
- Doors and windows have to work error free in straight, non-parallel and curved walls as well.
- The result of rotating, mirroring and flipping has to be considered in scripting.
- When you make non-rectangular openings you have to take into consideration the following: Using the "WALLHOLE" command is the easiest way to define holes in walls, but has two limitations. One, the opening cannot be concave. And two, the side surfaces of openings remain parallel in curved walls. The workaround for this problem is to add fill-bodies to rectangular openings. Unfortunately, fill-body definitions in curved walls can be very complicated.

Additional specification problems related to Doors and windows

- The possible contexts of a door or a window go beyond the plan and the 3D model, as may be suggested by the names of the library part editor's scripts. There is no independent script for sections, for example, but by using the GLOB_CONTEXT global variable, you can make a script that will not only be valid for sections/elevations, but also for the library part editor, the setting dialog or for the listing context. For all these versions, independent specifications must be made.

- For the sections, use plane, extrude and poly commands instead of prisms to have a hollow model of the library parts. These shapes will be represented by lines in section, without filling polygons. For eliminating lines that you don't want to see, use masking polygons where boundary lines' mask is set to invisible. This way, you can get a correct elevation and section drawing at the same time.

Positioning of doors and windows:

- There are general placing conventions for windows and doors. The library parts' coordinate system is in a special rotated position: the origin is placed on the bottom center of the wall opening, on the exterior side of the wall, and the x-y plane is vertical. Door/window zero level is on the external surface of the wall. What is outside of this is in the negative (-Z) range in 3D, and in the positive (+Y) range in 2D. A door is well defined if when inserting it in the wall, clicking to the right of the insertion point means that the door leaf is opening to the same side on the right. When a window is placed in the same way, the side that is clicked to indicates the outside.
- There are several combinations used to position a door: three basic positions are mirrored by the y axis (global variable name: SYMB_MIRRORED), mirrored by the longitudinal axis of the wall (=rotated 180 degrees: SYMB_ROTANGLE) and the flipping (WIDO_REVEAL_SIDE). There are many extreme situations where these positions are combined (e.g., a window with unequal jambs that is flipped and mirrored, and inserted into a curved wall), and usually each part of the window should react in a different way to these conditions. The client and the programmer must decide how the library part should, or should not, react to these conditions. To keep the library part consistent, several transformations should be used for these combinations.
- Flipping (changing the reveal side): one must consider whether the library part is sensible for this option, some parts of a door must be flipped when this option is activated (like door leaf, opening line, handle), and other parts must not. For example, wall-filling bodies should never be flipped. These parts should be handled with care in the script: in the case of flipping, changes made automatically by ArchiCAD should be redone.

Solution examples:

2D script:

```
Mul2 1,-1
Add2 0, -f~
```

3D script:

mulz -1
 addz -f~

In the case of windows, though flipping is also possible, for structural reasons it is not a correct solution. So either window shouldn't react to flipping (excluding this possibility by the script), or this option shouldn't be handled in the scripts.

Details

- General rules for using the details' edit boxes:

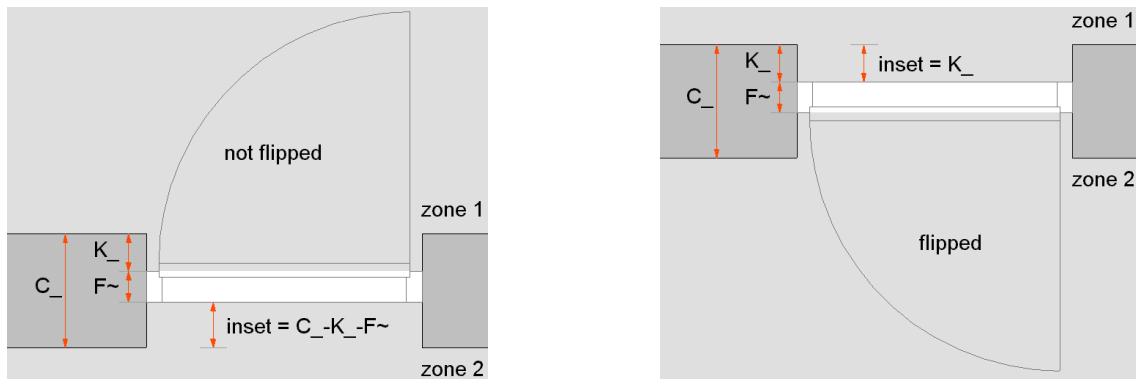
 Oversize, wall inset: only 127 characters can be used here!

 In the definition for oversize and wall inset values, only parameters and global variables can be used as variables.

 In the place of flow control statements (if...endif), you can use logical functions here. For example:

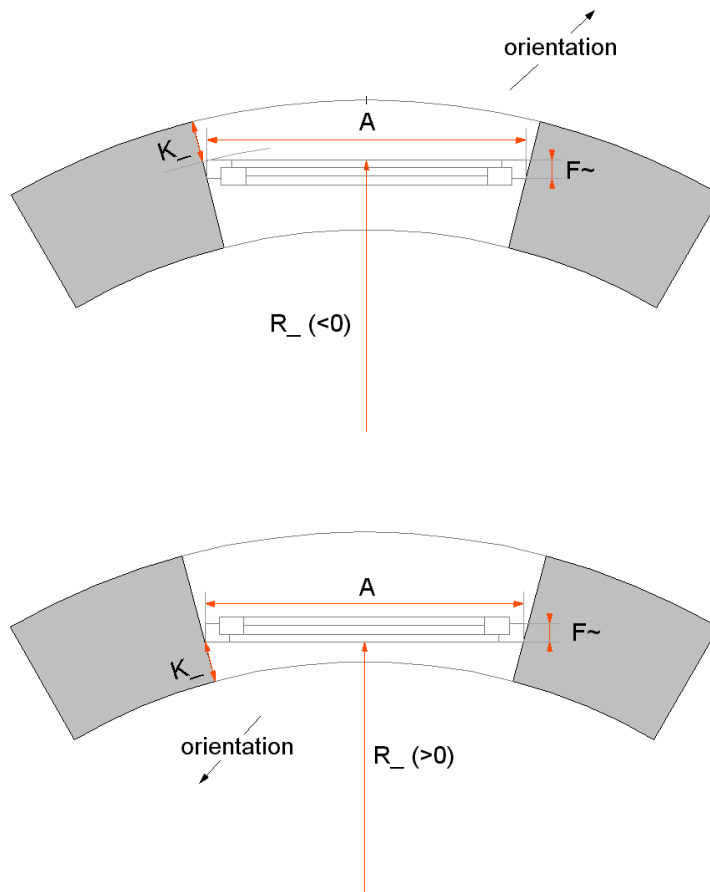
 (a<100)*x means: if a<100 then x
 not(a<100)*x means: if a<100 then x=0

- The parapet wall inset edit box: the value that is set here will define the behavior of the zone fill when the "Update zones" option is activated. If its value is -1, then it has no effect on zones. When its value is $e~*(c~-k~-f~)+not(e~)*k~$ (where $e~=1$ means flipped position, $f~$ means the nominal frame thickness), then the zone will show up on both sides of the frame (see picture below), even in the case of flipping. For windows, it can be used when they are inserted in the wall without any parapet height (GLOB_ELEVATION=0).



- The nominal frame thickness edit box: this value is used in the following contexts:

- When changing the reveal side (flipping), the library part is mirrored and dragged back by the value of the nominal frame thickness.
- When the “Update zones” option is activated, the value for parapet wall inset should contain nominal frame thickness to have a correct result (see picture below).
- Inserting a window/door in a curved wall: nominal frame thickness has an effect on the placing of the window/door (see picture below).



- The oversize edit box: the oversize values have an effect on the size of the rectangular holes cut in the wall automatically by ArchiCAD for doors/windows. It can be used, for example, for placing a sunshade’s box above the window, using scripted jambs instead of ArchiCAD jambs (see below) and so on. These values must be carefully designed during the specification phase to have a satisfactory result.

Scale sensitivity

Very often, windows/doors change according to the project scale. The most common example can be the 2D symbol of these library parts. One can design different symbols for any scale using the GLOB_SCALE variable. However, it is not recommended to define anything for scales less than 1:20, because projects made by ArchiCAD and using the common type of libraries characteristically never require details below this. If for any reason sophisticated details are needed for 1:10, 1:5, 1:1 scales, draw 2D fragments in the 2D symbol window, and use them by fragment2 command in the 2D scripts.

For 3D models and section symbols, a simplified version may be required over 1:100 scale. This option can also be parameterized, even independent of the project scale.

Composite windows and doors

- For optimizing parameter complexity, composite library parts can be made instead of overcomplicated and difficult-to-handle elements.
- Composite library parts are made by using invisible boundary lines on the connecting sides. Such a library part must be functionally perfect by itself, too. Its accessory elements (like sills) must also be special ones to react logically to connections. In the case of corner windows, the oversize must be designed in a way that no wall will show through after connecting two of them.
- Accessory elements of some windows/doors like sunshades, handles, sills should also be involved in the library as independent library parts, not only macros, to make them available for other objects, too. In this way you can put together, for example, a sunshade and a window (that doesn't already have the option of calling this sunshade) by placing them in the same place in the plan.

Transparent materials

Use the "body -1" command after hollowed 3D shapes to ensure correct shadow casting if the holes are filled with transparent surfaces (e.g., window sash with grilles).

Special opening shapes

- Designing a non-rectangular window/door needs wall filling bodies. As ArchiCAD cuts a rectangular hole in the wall, the size of which is A*B (where these values are increased with the oversize), the missing parts must be created in the 3D script. Considerations:
 - Attributes (material, color, pen number etc.) of the filling body must be the same as the wall's adequate attributes.

- In the case of a curved wall, the resolutions must be equal. In ArchiCAD, curved walls are made of segments. To adjust segments perfectly, the zero point of the filling body's coordinate (most commonly the start point of a prism element) must be the same as that of the wall. To have a perfect photorealistic image, the COOR command must be used to define texture coordinates.
- Special care must be paid to the positioning problems: as the wall filling body should never react to the mirrored, rotated, flipped positions in the same way as the window/door itself, these transformations must be neutralized in the 2D/3D scripts.
- Filling bodies should usually contain the option of jambs, even for asymmetrical and curved version.
- Wallholes will create holes in the walls in a different way than the ArchiCAD automatic cut. In addition to the fact that the wallhole can be of any shape, in curved walls the hole's boundaries are not centered but parallel to each other.

Features

- Jamb GDL means that the library part doesn't use the jambs created by ArchiCAD, but in a parameterized form it is an option of the library part. This may be needed, for example, when the user doesn't want to have a jamb in 1:100 scale. This must be handled in the oversize edit box as well. (To keep the user from choosing a window/door with jamb in the toolbox, send a message when the WIDO_SILL global variable is not zero.) Jamb GDL must function for composite wall structures as well, and must react correctly to all positioning, even in asymmetric cases, both in 2D and 3D.
- Zone / fill hiders may be used to hide zone fills under the library part, e.g., under sills. Fill hider usually covers the connecting wall contour line as well, so either these lines must be reestablished by the script or there should be no zone hider used.

Preview pictures

Previews must be made in a uniform way to emphasize the characteristic attributes of the doors and windows. Therefore, show the elevation shape of the library parts on a neutral (white) background. Axonometric or perspective images with wall or floor make the elements difficult to recognize. For previews, use the Special menu / Update library parts command.

4.10 Macros, Macro Objects, Master GDL

- Calling a macro object from many objects can reduce library size. Do not make macro calls from a single object. This complicates further modifications on the object.
- To simplify library checking, avoid calling a macro from another macro.
- Prefer macro objects to macros, as they are more sophisticated library elements (e.g., you can add preview pictures, comments, etc., to your macro objects).
- Do not create a macro whose name is a parameter to avoid missing macros from archive files. ArchiCAD saves only the default macro in the archive file.
- Use Master GDL scripts to define custom attributes (materials, fills and line types) for more objects in the library. This can reduce library size and make further modifications easier. Master GDL scripts have to be named like this: Master_GDL name.gdl (e.g., Master_GDL Materials.gdl).
- When you call a macro, always use the “CALL” command and put the name of the macro between quotation marks (e.g., CALL “leg_65”).
- In GDL, macro files (*.gdl files) always start the script with a parameter definition to make corrections easier.

Example:

```
! door_panel_O /modified by TL 3/1/94/
```

```
! Edited by TL/ 12-15-97
```

```
! internal macro: door_int_frame
```

```
! a=width, b=height, c=thickness, d= panel frame width,
```

```
! e=fill material, f=internal frame width,
```

```
! g=horizontal n. of strips, h=vertical n. of strips
```

```
i= (a-(g+1)*d)/g    ! horiz size of panes
```

```
j= (b-(h+1)*d)/h    ! vertical size of panes
```

- Always use the same variable names for the same parameters in macro objects (*.ism files) and in the file you are calling the macro from.
- Delete all coordinate transformations and empty all parameter buffers at the end of macro scripts. Finish the scripts with the “END” command.

5. ArchiFM Libraries

This chapter explains how to use GDL parameters to create a connection between a graphical object and its database counterpart in ArchiFM

5.1 General

It is possible to extend GDL Objects and Zones with parameters to meet ArchiFM's needs. The main benefit of using GDL Objects and Zones in ArchiFM is to give visual feedback to the information stored in the database.

There are two different types of GDL parameters: one that has no ArchiFM specific characteristics, e.g., Material, Font Size, etc., and the other, which has an exact counterpart in the database or has a special characteristic e.g., Inventory Number, Manufacturer.

If you have a parameter in the GDL library part's parameter list like "Material" and the variable name of this parameter is "material," it can be used as a parameter that affects the 3D object, but it will not be converted into the ArchiFM database. This is because the converter will recognize and convert only those parameters that have an "FM_" prefix.

If you have a parameter created with the prefix "FM_", it will be used by the ArchiFM database. It appears both in an exact data field on a given tab page of the converted object or room, or as an extended attribute in the Extended Attributes tab page. The difference between the two appearances depends on what follows the "FM_" prefix. If the name is the same as one of the Objects or Rooms table's column, then the value of the parameter will appear in the appropriate field. See Appendix C for the list of available Objects table parameter names that have an exact value field.

5.2 Objects

Each object must have a parameter that defines the object's group type, e.g., People, HVAC, Furniture. To this you have to add a parameter whose object's GDL variable name is FM_Type. The GDL parameter name is customizable. The value of this parameter can be the exact name of the required type, which can be found in the Configuration file (*.BFG), e.g.: People, HVAC, Furniture. In this case the GDL variable type is text (abc). Or it can be a unique number, which can also be found in the Configuration file (*.BFG) that defines the group type. In this case the GDL variable type is integer. The following list shows the available group types and their ID:

Default Objects	1
Default Lamp	1

People	3
Elevators	14
Security and Escapes	10
Stairs	11
Work Spaces	12
Electric Facilities	4
Distribution Panels	5
Measuring Places	6
Lamps	8
HVAC Equipment	7
Sanitation	2
Locks	16
Built-in Wardrobes	2
Furniture	2
Ceilings	15
Floors	13
Outdoor Objects	2
Computers	18
Monitors	17
Software	19
Phones	20
Containers	21
Others	9

5.3 Zones/Rooms

The word *Zone* is the name of a space in the graphical part, *Room* is used in the ArchiFM DB part. The Zone GDL object has almost the same behavior as the generic GDL object has. The "FM_" convention also works in case of Zones' parameters, but there are some other specialties.

There are many default zone parameters that can be converted automatically. These parameters are listed in a new zone library part, even if you did not yet edit anything. Examples include *ROOM_NAME*, *ROOM_NUMBER*, *ROOM_AREA*, *ROOM_WALLS_SURF*, etc. Many of these are used also in the

ArchiFM database tab pages. This list and the associated explanation can be found in the Graphisoft GDL Reference Manual.

Usually a zone object has a setting for the *Floor Type*. The Graphisoft Zone' GDL variable name is usually *floor_f*, so if you use this variable name to define a new zone parameter, ArchiFM DB will recognize it. Although the naming convention is also valid: *FM_FloorType* equals of *floor_f*, so the GDL parameter's value will appear in the *Floor Finish* field of the *Attributes* tab page of the selected Room. In this case, the GDL variable type is text (abc).

ArchiFM also converts parameters that have the variable name *cn_zn*. This default Graphisoft parameter variable name defines the *Use Category as Name* parameter. In this case the GDL variable type is boolean (yes or no). This can be very important if you do not want to fill the zone's *Zone Name* field with a value different from the *Zone Category* value. If you check this parameter on, the *Zone Category* field value will appear in the *Zone Name* field.

There are other FM specific parameters that can be converted if the GDL zone object contains the following parameters with the following GDL variable names:

ArchiFM DB parameter name (GDL parameter name)	GDL variable name	GDL variable type
----------------------------------------------------	-------------------	-------------------

Room

Main

Price Multiplier Factor:	(FM_RentAreaPriceMultip)	Real Number
Orientation: (characters)	(FM_Orientation)	Text (max. 20 characters)
Fire Rating: (characters)	(FM_FireRatingLevel)	Text (max. 10 characters)
Room's Rent Type: (characters)	(FM_Rentable)	Text (max. 1 characters)

To use the *Orientation* field properly, it must have a value selected from the next list:

Name	ArchiFM DB Room <i>Orientation</i> data value (GDL parameter value)
north:	N
north-east:	N-E

east:	E
south-east:	S-E
south:	S
south-west:	S-W
west:	W
north-west:	N-W

The *Room's Rent Type* field's values must be one of the following three:

Name	ArchiFM DB Room <i>Room's Rent Type</i> data value (GDL parameter value)
------	-----------------------------------------------------------------------------

Rentable:	R
Not Rentable:	N
Service Area:	S

In the case of the *Price Multiplier Factor*, the default value is "1".

It is possible to control the corresponding Cost Centers of a Room from the Graphics portion of the software. In this case, you have to create a GDL parameter with one of the following GDL variable names:

GDL variable name:	FM_CostCenters
	FM_CostCenters1
	FM_CostCenters2
	FM_CostCenters3
	FM_CostCenters4
	FM_CostCenters5

This means that you can keep track of six different cost centers in a GDL zone at the same time. If you enter a cost center value that does not exist in the ArchiFM database's Cost Centers editing dialog it will be generated during the synchronization between the database and the graphics. The GDL variable type is text (abc).

Units

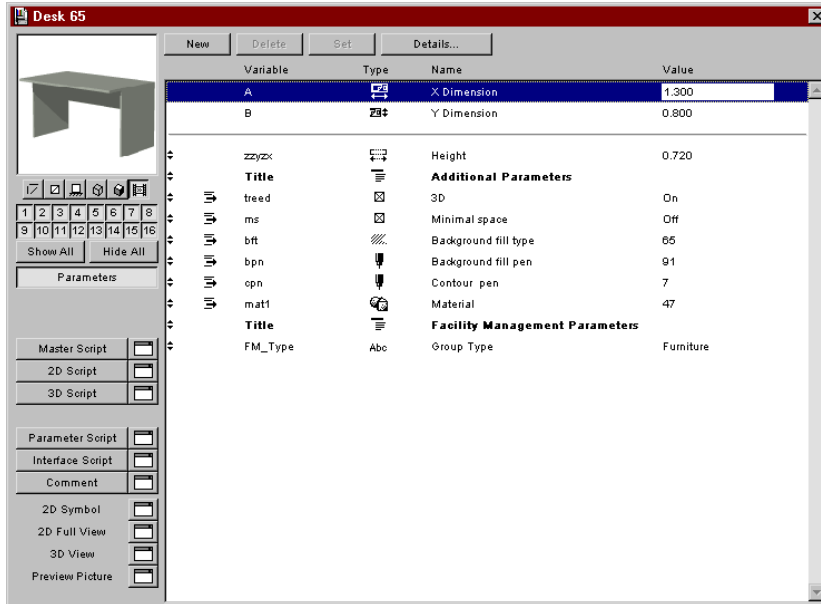
You can also create unit parameters after using a value. In this case, you have to create a text (abc) type parameter, and the value can be the sign of the unit e.g., m, ft, m2, sqft. If you enter a unit value that does not exist in the ArchiFM's Units editing dialog, it will be generated after a synchronization.

5.4 Examples

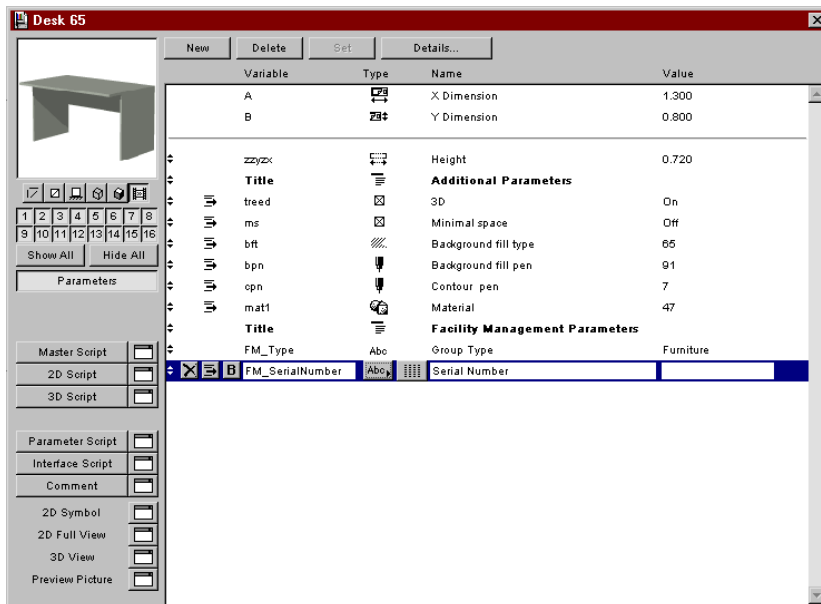
Below there are two examples of how an FM related parameter can be created in terms of objects:

Parameter that has an exact data field counterpart

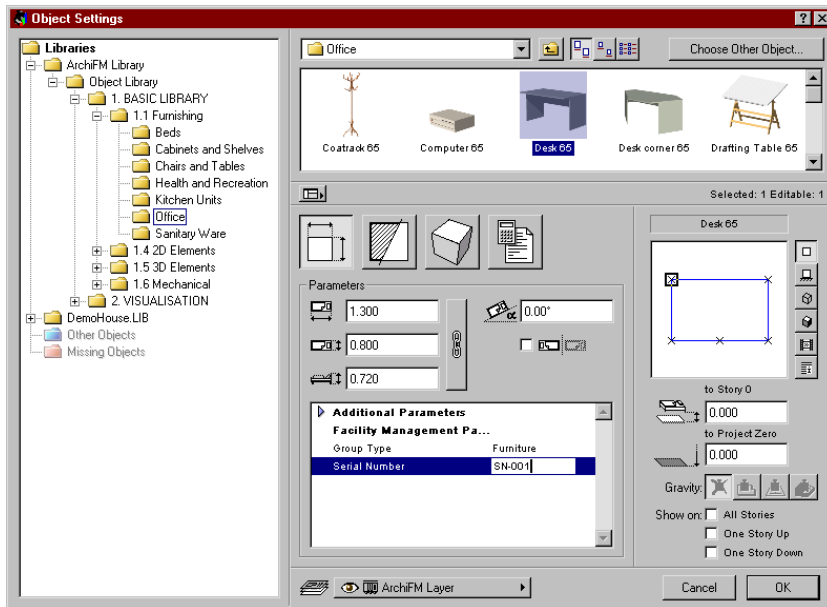
1. Open the Library Part.



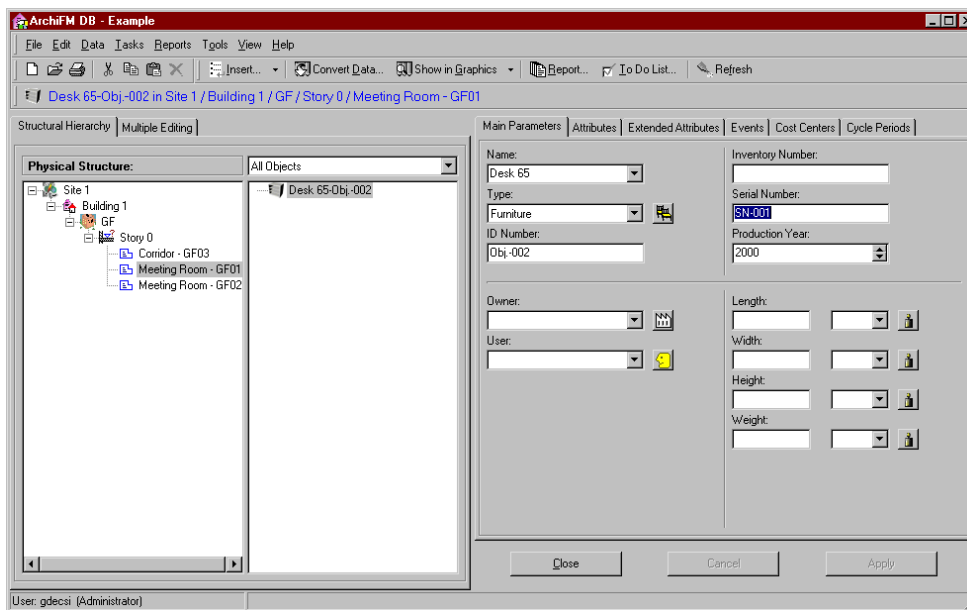
2. Create a new parameter, e.g., variable: *FM_SerialNumber*; parameter name: *Serial Number*; value: Default value or no value. The variable: *SerialNumber* is an existing parameter of some Objects Groups. The parameter's variable must start with "FM_".



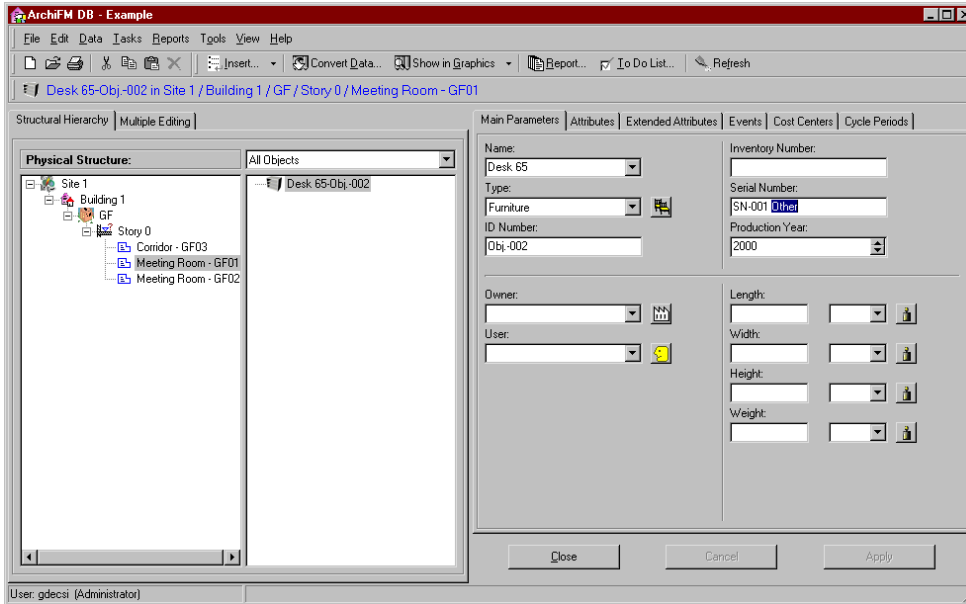
3. When you want to place an object on the floor plan, you have the option to change the parameter's value. Fill in the *Serial Number* value as "SN-001".



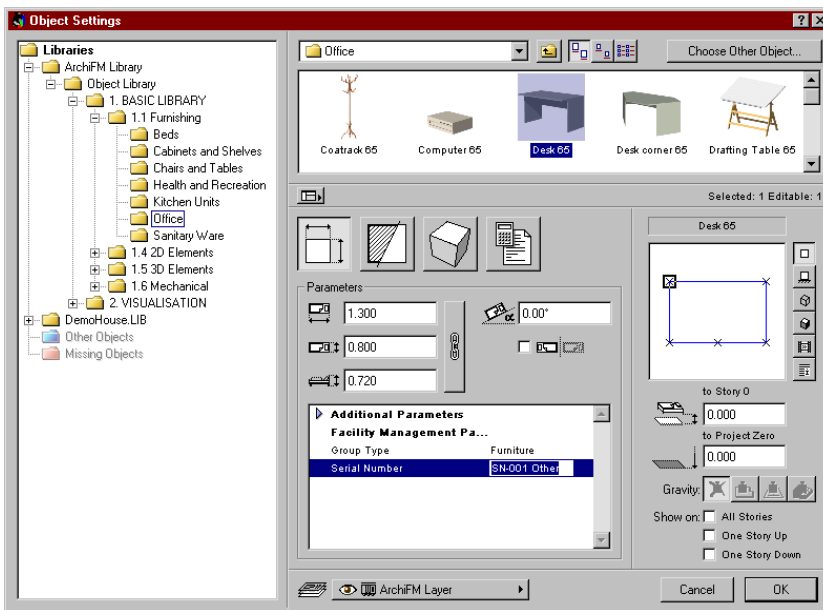
4. After you convert the object from the graphics into the ArchiFM database, the parameter's value of "SN-001" will appear in the *Serial Number* data field.



5. The *Serial Number* value can be modified in the ArchiFM database, e.g., *SN-001 Other*. You must then press *Apply*.

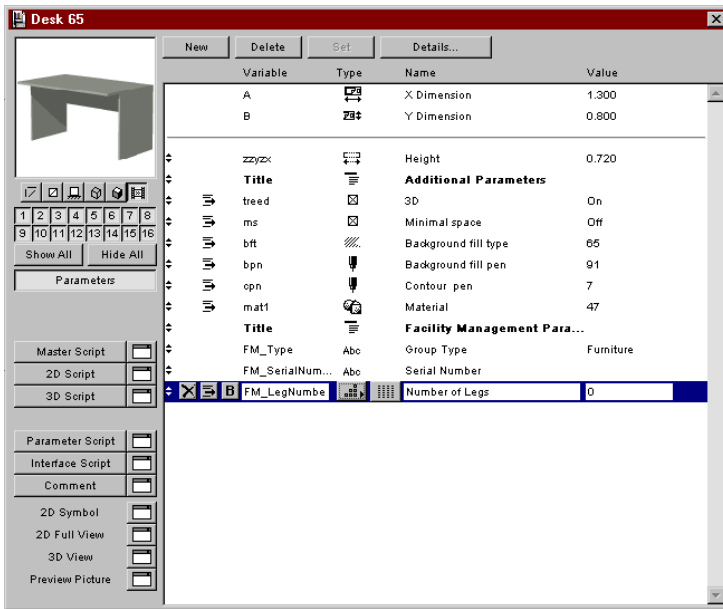


6. Choosing the *Show in Graphics* command will open the Object Settings dialog of the selected object. The modified parameter value will also be updated in the Object Setting dialog's parameter list. This process also works in the reverse order .

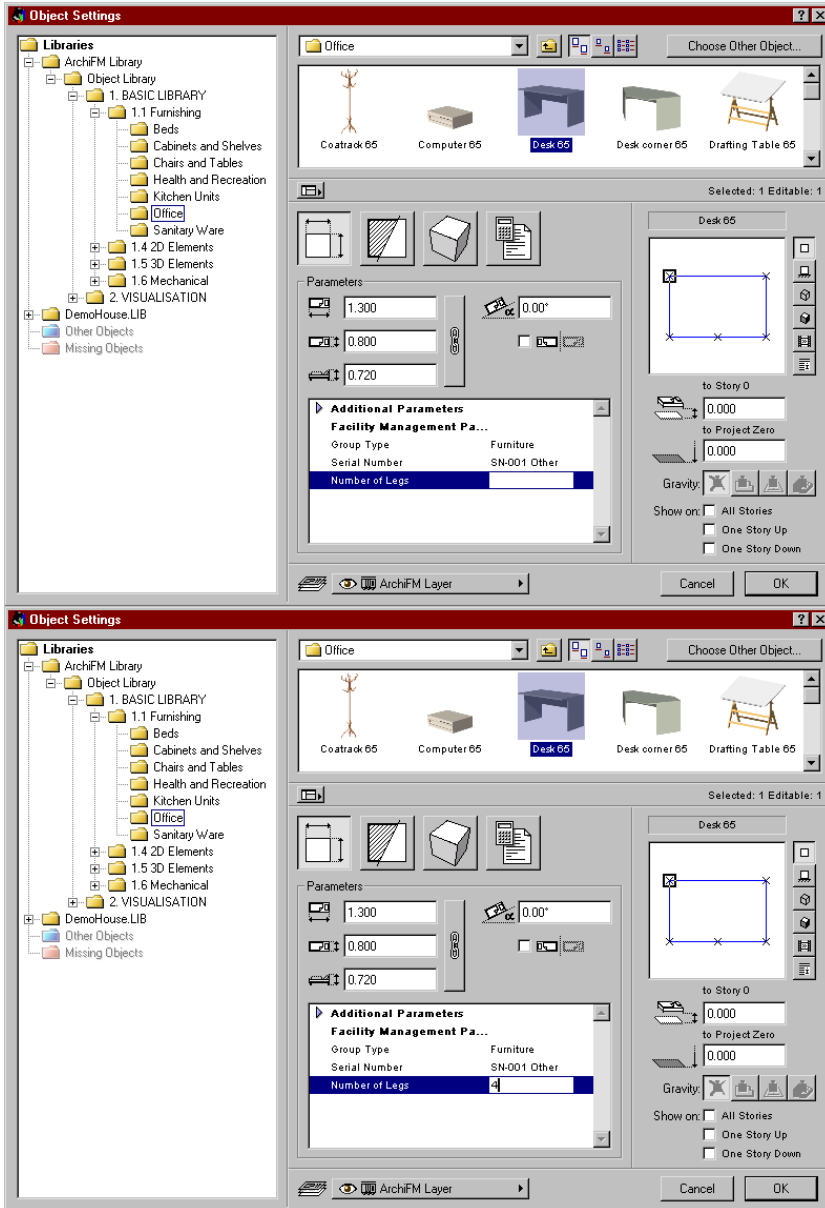


Parameter that has NO exact data field counterpart

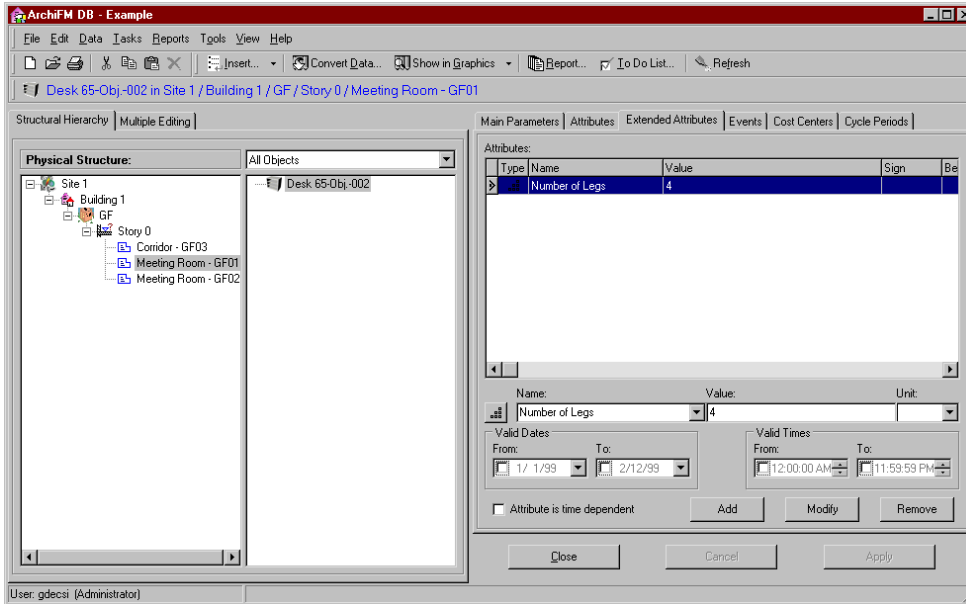
1. Add a new parameter to the parameter list e.g., variable: *FM_LegNumber*, parameter name: *Number of Legs*; value: Default value or no value. The variable: *LegNumber* is not a parameter in any Objects Group. The parameter's variable has to start with "FM_". This parameter can also be used in the 2D and the 3D script with the option of setting the object's leg number.



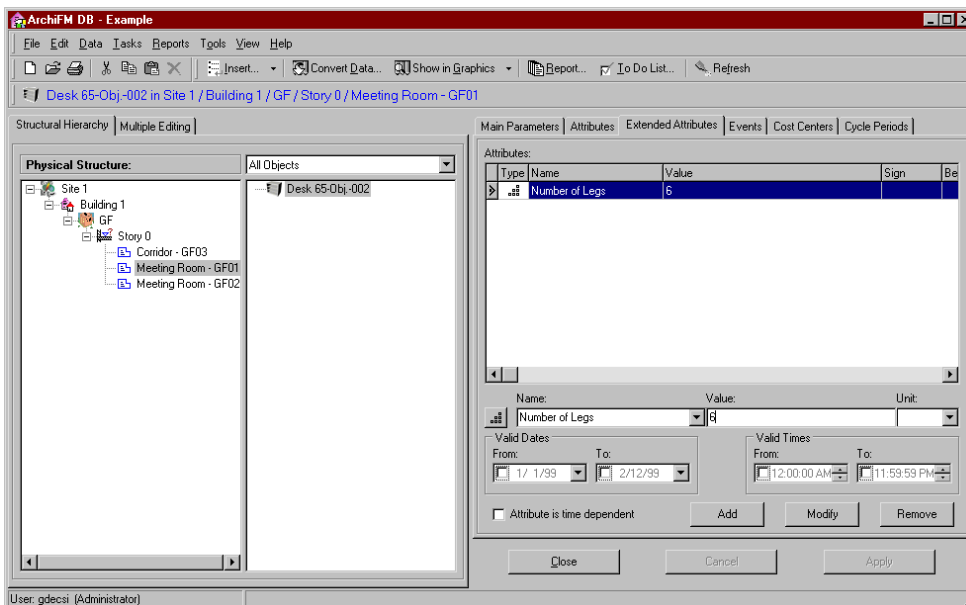
2. When you want to place an object on the floor plan you have the option to change the parameter's value. Fill in the *Number of Legs* value of "4".



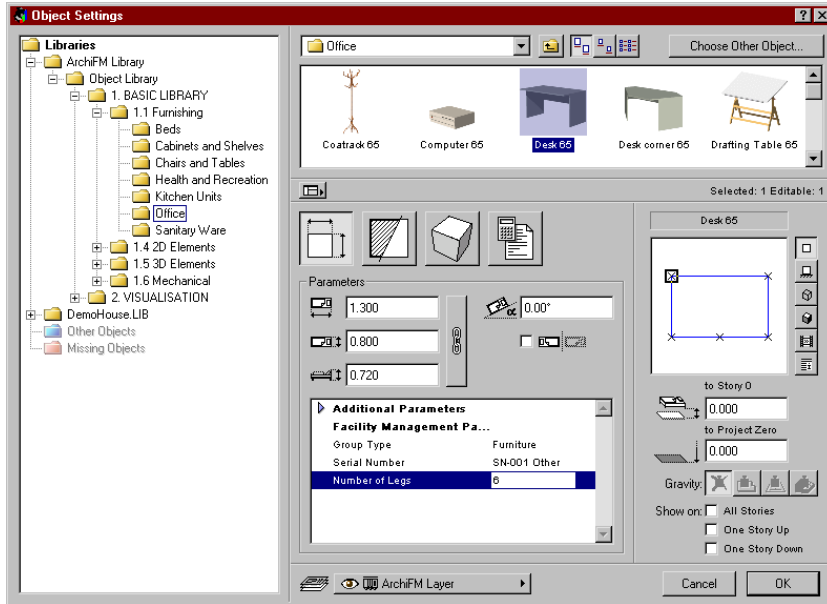
3. 4. After you convert the object from the graphics into the ArchiFM database, the parameter's value of "4" will appear in the object's *Extended Attributes tab page*.



4. The *Number of Legs* value can be modified in the ArchiFM database, e.g., 6. Press *Apply*.



5. Choosing the *Show in Graphics* command opens the Object Settings dialog of the selected object. The modified parameter value will be updated also in the Object Setting dialog's parameter list. This process also works in the reverse order .



Note:

1. When you edit an object that was already used in the floor plan, you must *Reload* the library in the Graphical part by selecting *File Menu\ Load Libraries...*
2. If you also want to use the new parameter in the database, you have to select the required objects and choose *Convert Data...* from the ArchiFM command bar.

6. Web libraries

6.1 General Requirements

If you want to publish your ArchiCAD library on the Web, you must make it compatible with Graphisoft's Object Web Plug-in by following these guidelines:

- There should be a master GDL file in the library that contains all linetypes, fills and materials used by the library and all global parameters used by the Object Web Plug-in. The name of this file has to be "master.gdl".
- The Object's default material, fill and line type parameters have to be set correctly to work without errors in the plug-in.
- The default unit of the length parameters has to be set correctly (m, inch, etc.) according to the national standards.
- The texture and macros folder path has to be deleted from the HTML file.
- All image files without alpha channel should be converted into JPEG format, to make library downloading faster. Files with alpha channel should be saved in TIFF format.

6.2 Library Conversation

- You have to make a correct master.gdl file first. Use the "Save GDL" function from "Special" menu to save ArchiCAD attributes in GDL files. (See chapter 9.1.) Use the "GDL fills, Ltypes" and then the "GDL materials" option from the "Files of type" pull down menu and save the two files.
- Create a new GDL file and copy the content of the two files into it. If you have one master GDL file already, you have to rename and redefine it. Please note that ArchiCAD uses the "master_gdl name.gdl" filename, while in the Object Web Plug-In it has to be called "master.gdl".
 - After creating your master GDL file, you have to add the following global variables, required by the Object Web Plug-In, at the beginning of the script.

```
!!!!!!----- Global Parameters
```

```
glob_scale=100  
symb_linetype=1  
symb_fill=65  
symb_fill_pen=2  
symb_view_pen=1  
symb_fbgd_pen=91  
symb_sect_pen=1
```

```
symb_mat= 1
```

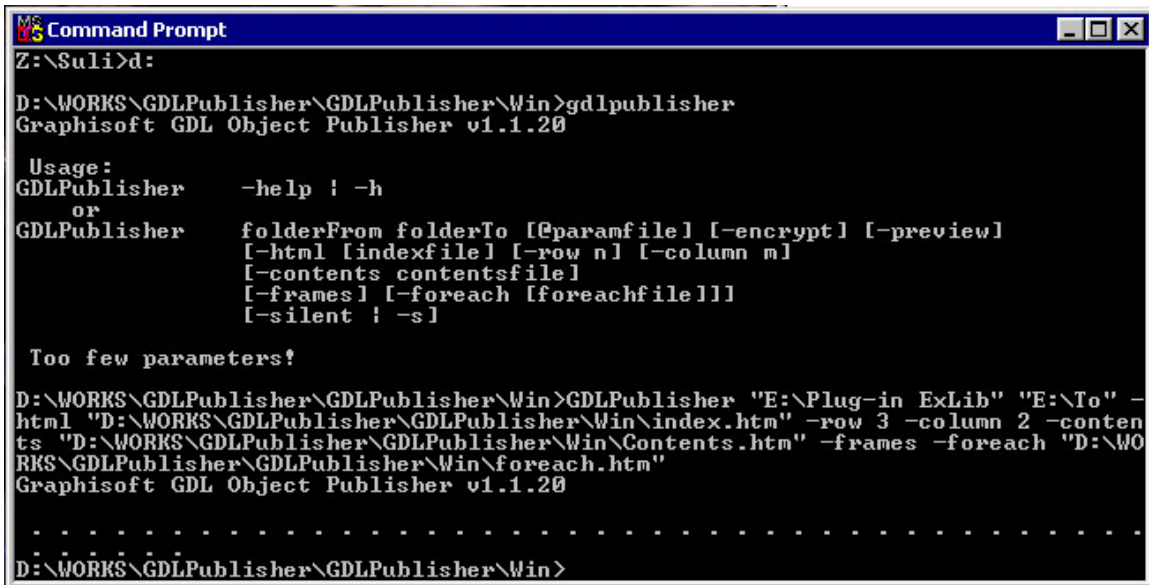
```
Wall_line_type=1  
Wall_Fill=65  
wall_fill_pen=2  
wall_view_pen=1  
wall_fbgd_pen=91  
wall_mat_a=1  
wall_mat_b=1
```

wall_height=2.7
wall_thickness=0.25

mglas=1

!!!!!!-----

- Load the library with the new master.gdl file. Open the Option/materials settings dialog and delete all the materials, fills, and linetypes except the one that is defined by the master.gdl. Reload the library. Now you should only see the materials, fills, and linetypes defined in the master.gdl. Redefine the default parameter values for lines, materials and fills in all objects.
- Run GDL Publisher. You can run it in the DOS window.



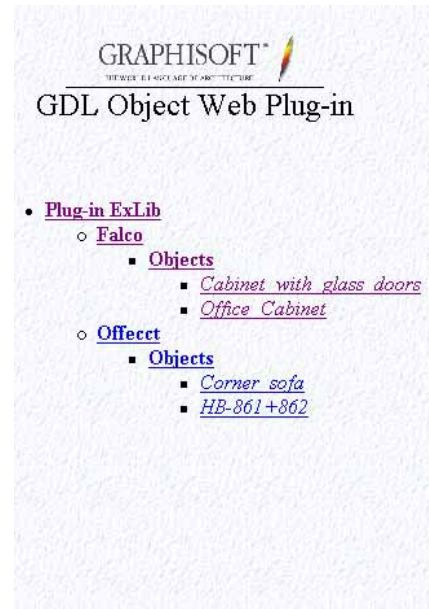
```
MS-DOS Command Prompt
Z:\Suli>d:
D:\WORKS\GDLPublisher\GDLPublisher\Win>gdlpublisher
Graphisoft GDL Object Publisher v1.1.20

Usage:
GDLPublisher      -help | -h
or
GDLPublisher      folderFrom folderTo [@paramfile] [-encrypt] [-preview]
                  [-html [indexfile] [-row n] [-column m]
                  [-contents contentsfile]
                  [-frames] [-foreach [foreachfile]]]
                  [-silent | -s]

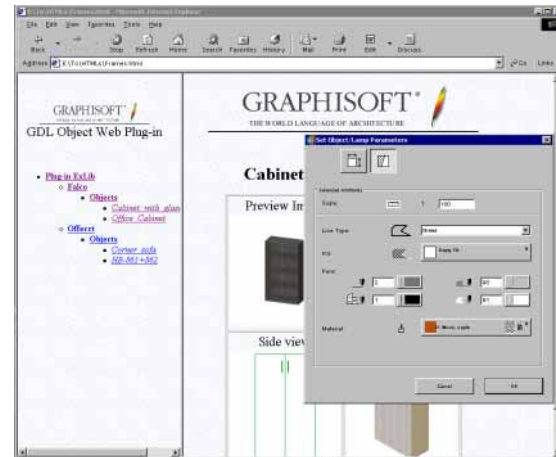
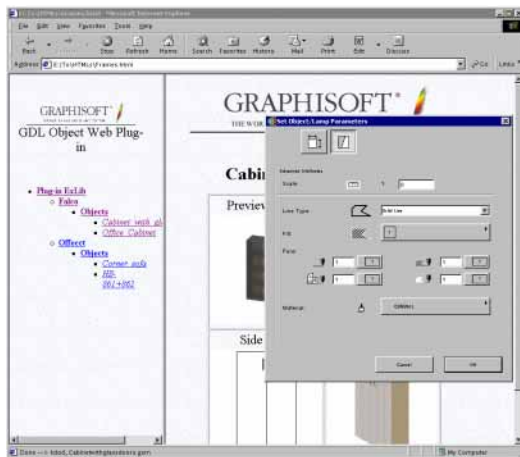
Too few parameters!

D:\WORKS\GDLPublisher\GDLPublisher\Win>GDLPublisher "E:\Plug-in ExLib" "E:\To" -
html "D:\WORKS\GDLPublisher\GDLPublisher\Win\index.htm" -row 3 -column 2 -conten
ts "D:\WORKS\GDLPublisher\GDLPublisher\Win\Contents.htm" -frames -foreach "D:\WO
RKS\GDLPublisher\GDLPublisher\Win\foreach.htm"
Graphisoft GDL Object Publisher v1.1.20
. . . . .
D:\WORKS\GDLPublisher\GDLPublisher\Win>
```

- Put the “title” file near the “Frame.html” and the “white” background file in an “images” subfolder.
- Delete the “textures” and “macros” folder’s path from HTML script.



- Open “Frame.html” and check the default global parameters (pens, fills). If something is missing, check your master.gdl file.



- Your libraries are now compatible with the GDL Object Web Plug-In.

7. Windows-Macintosh Platform Compatibility

7.1 Compatibility Limitations

Though GDL objects and libraries are considered by Graphisoft as platform independent, the following difficulties occurs when you manually move Windows objects to a Macintosh:

- Resource forks will be missing from objects, therefore they loose their icons.
- Windows fonts will be replaced by the default Macintosh font in objects and list templates.
- Text type listing files (listset.txt, listkey.txt, list templates, etc.) could loose line breaks, therefore listing doesn't work.
- Textures moved from Windows to Mac and later saved in an archive file won't work on Windows since file names have been changed by ArchiCAD (e.g., wood.tif will be renamed to wood.tif.tif).

7.2 Changing Platform

- To avoid these problems, save a multiplatform archive file of your library whenever changing platforms. Be sure to turn on the "Save in MultiPlatform Format" checkbox in the Options/Preferences/Miscellaneous menu. Turn on the "Save all Parts of Loaded Libraries" and "Include Properties in Loaded Libraries" options in Save Archive dialog box as well.
- After opening the archive file, manually change non-system fonts in your objects and list templates to system fonts.
- Library developers have to provide both Macintosh and Windows libraries for Graphisoft. Compare the number of elements in these versions.

8. Library Localization

The aim of localization is to create a “local” or national version of the international ArchiCAD library. Library localization has to include the following steps:

8.1 Localization Steps

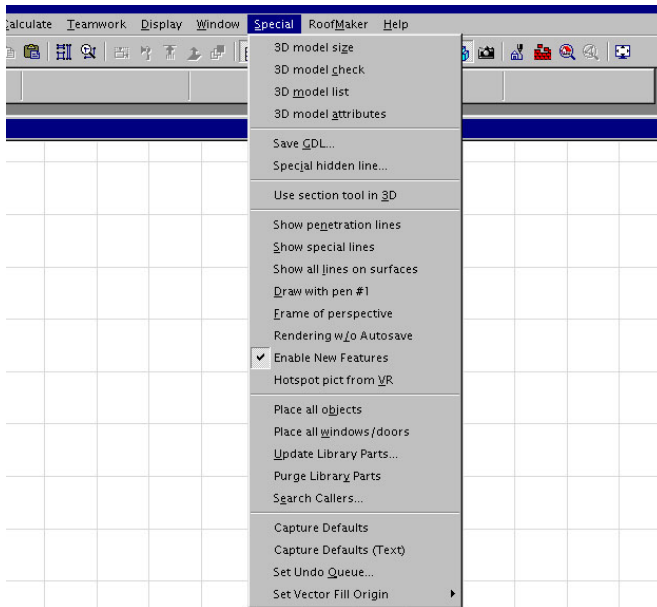
- Files and folders have to be translated and renamed (except macro and texture files) unless otherwise requested.
- Parameter names have to be translated in all library parts with the “Update Library Parts” or the “Search & Replace in Library Part Parameters” tools.
- Value listed parameters, user defined messages and texts have to be translated in parameter, 2D, 3D, interface and master scripts with the above-mentioned tools.
- Font types in objects have to be changed to ones supported by national Macintosh and Windows operating system.
- Default parameter values have to be set according to national standards (metric-imperial units, local ArchiCAD attributes, window door opening direction etc.).
- The library has to be checked for errors. The number of elements in the library has to be checked

9. Tools for Library Developers

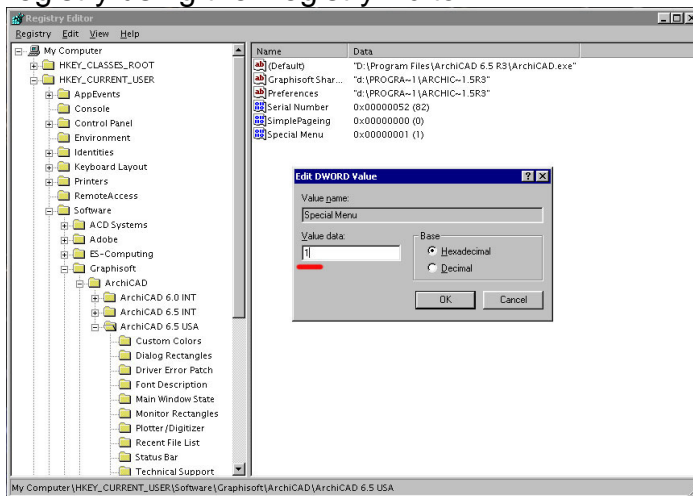
The following commands and add-ons have been developed by Graphisoft to speed up library development, localization and quality checking. We encourage you to use them.

9.1 Special Menu Commands

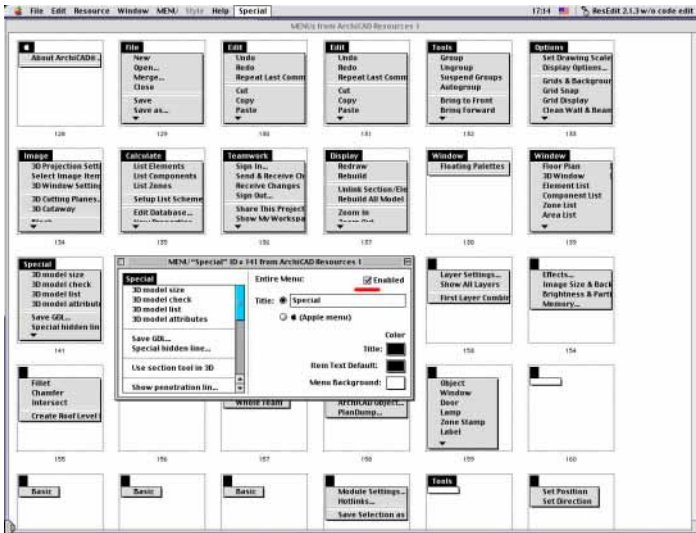
The above mentioned commands are available from ArchiCAD's "Special Menu".



To access this menu, hold down Ctrl+Alt on PC or Alt+Command on Macintosh after starting the program. If you want this menu to be permanently available in your ArchiCAD you have to do the following: On Windows, modify the program registry using the Registry Editor.



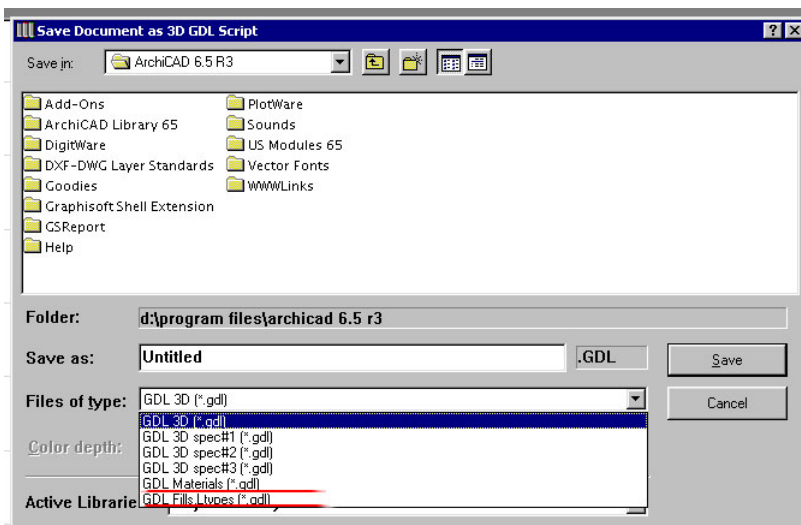
.On Macintosh, you have to modify the “ArchiCAD Resources 1” file with any resource editor program (e.g., Resedit).



- **Save GDL**

This feature allows you to save the attributes of an ArchiCAD file or the 3D model in a GDL file. After choosing the command a dialog comes up where you can set the name, location and type of the GDL file. The file types are:

- GDL 3D ⇒ the 3D model is described with GDL primitives
- GDL 3D spec#1 ⇒ the 3D model is described with GDL primitives
- GDL 3D spec#2 ⇒ the 3D model is described with GDL primitives
- GDL 3D spec#3 ⇒ the 3D model is described with GDL primitives
- GDL Materials ⇒ materials are saved in the GDL file
- GDL Fills, Ltypes ⇒ fills and line types are saved in the GDL file



This tool can be very useful for making master GDL scripts and for defining custom attributes.

- **Place all objects**

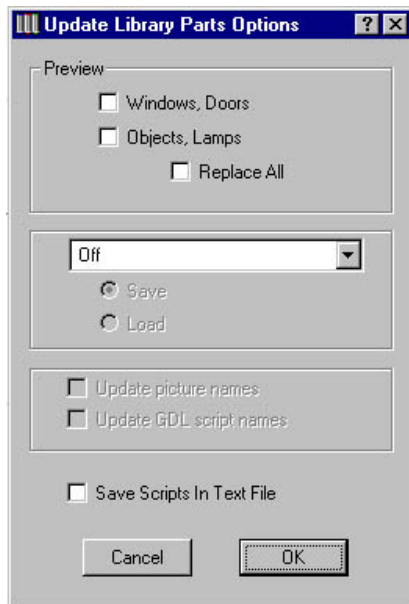
This command places all the objects and lamps in the loaded libraries on the plan. The elements are set with their default parameter settings. This tool can be used for checking complete libraries for errors in 2D and 3D.

- **Place all windows and doors**

This command is similar to the previous one. It places all doors and windows in separate walls.

- **Update library Parts**

With this tool you can automatically modify all elements in the loaded library. The dialog box is divided into three parts.



The upper part contains three checkboxes that control automatic preview generation. If the “Windows, Doors” or the “Objects, Lamps” option is on, this utility generates and saves a preview for the first object, lamp, door or window in the library. If “Replace All” is checked, this process will be done on all elements. The previews are made according to the 3D window and photo rendering setting. Be sure to set a square shape 3D window (e.g., 300x300 pixels) and a 128x128 pixel rendering window in order to generate a correct preview

The middle section of the dialog contains a pull-down menu and two radio buttons. From the pull-down menu you can select the type of script you want to modify. If you check the “Save “ button, the selected scripts from all library parts will be saved in a tabulated text file. The extension of this file is “REP”. This text file can be modified with any word processor program. For example, the error

messages defined in the scripts can be translated. After modifying the “REP” file, check the “Load” button, select the file location and this utility automatically updates the script in all library parts. This tool is essential for library localization.

At the bottom of the dialog you can find the “Save Scripts in Text File” checkbox. If you check it, the program saves all scripts and the additional parameter list of the library parts in a text file, which you can use for finding errors.

- **Purge Library Parts**

This command removes precompiled binary files (*.gs&, *.do&, *.wi& etc.) from loaded libraries on Windows.

9.2 Add-Ons

The “Search & Replace In Library Parts” add-on is part of the GDL Developer Kit. To use this tool you have to load “Search in Library Part.apx” with “Options / Load Add On” command. The “Find & Replace in Library/Library Parts” and the “Find & Replace in Library/Library Part Parameters” commands will appear in the Options menu.

Search & Replace In Library Parts

With this Add-on you can search for a given string in the specified sections of the loaded library parts. The found strings can be replaced with another string. Note: The GDL (plain text file) macros will not be processed.

How to use the Add-On:

- Close any open Library Part in ArchiCAD before using this tool.
- Select the 'Search & Replace In Library Parts...' command from the Tools menu.
- Set the string to be searched.
- Set the string to be replaced if it is necessary.
- Specify the script sections you want to search in. You can select more than one at the same time.
- Select Regular checkbox to search for regular expressions
- Push the 'Find' or 'Replace' button.



Result: Every occurrence and the summary are written to the Report Window of ArchiCAD.

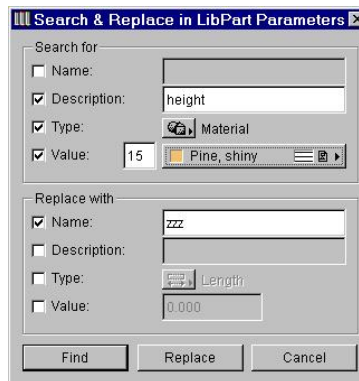
- **Search & Replace In Library Part Parameters**

With this Add-on you can search for a given parameter of the loaded library parts. The found parameter's value, type, description or name can be replaced with the given one.

- Close any open Library Part in ArchiCAD before using this tool.
- Select the 'Search & Replace In Library Part Parameters...' command from the Tools menu.
- **Searching:** library part parameters can be localized in the active libraries, by any of their attributes: *Name*, *Description*, *Type* and *Value*. The search criteria comes from the logical AND of the enabled attribute values. To enable searching by a certain attribute, check the corresponding check box. Every result is written to the ArchiCAD's Report Window.
- **Replacing:** any attribute of a library part parameter can be altered in one step: the attributes you want to replace can be selected by checking the corresponding checkboxes.

Note: In *Name* and *Description* attributes, as well as in text parameter values, only the matching characters (and in the case of multiple occurrences in a text, each) are replaced.

Except for some special behavior on arrays, explained below, use of the tool is straightforward.



Special notes on arrays

Not every combination of searching and replacing attributes is allowed! Array parameters behave in two ways: (i) the whole array behaves as an ordinary parameter having *Name*, *Description* and read-only *Type*, but no *Value* attributes; or (ii) the array elements themselves behave as a parameter having *Type* and *Value* attributes as well.

The following meaningful combinations are allowed:

Functionality	Search In				Replace with			
	Name	Description	Type	Value	Name	Description	Type	Value
Change array name/description				0			0	0
Replace array element values (the 2 types must be the same)			1	1	0	0	1	1
Fill the whole array with a given value (and change its type if necessary)				0			1	1

Legend: 0 - Unchecked, 1 - Checked, Empty - Don't care: can be used for refining the searching criteria

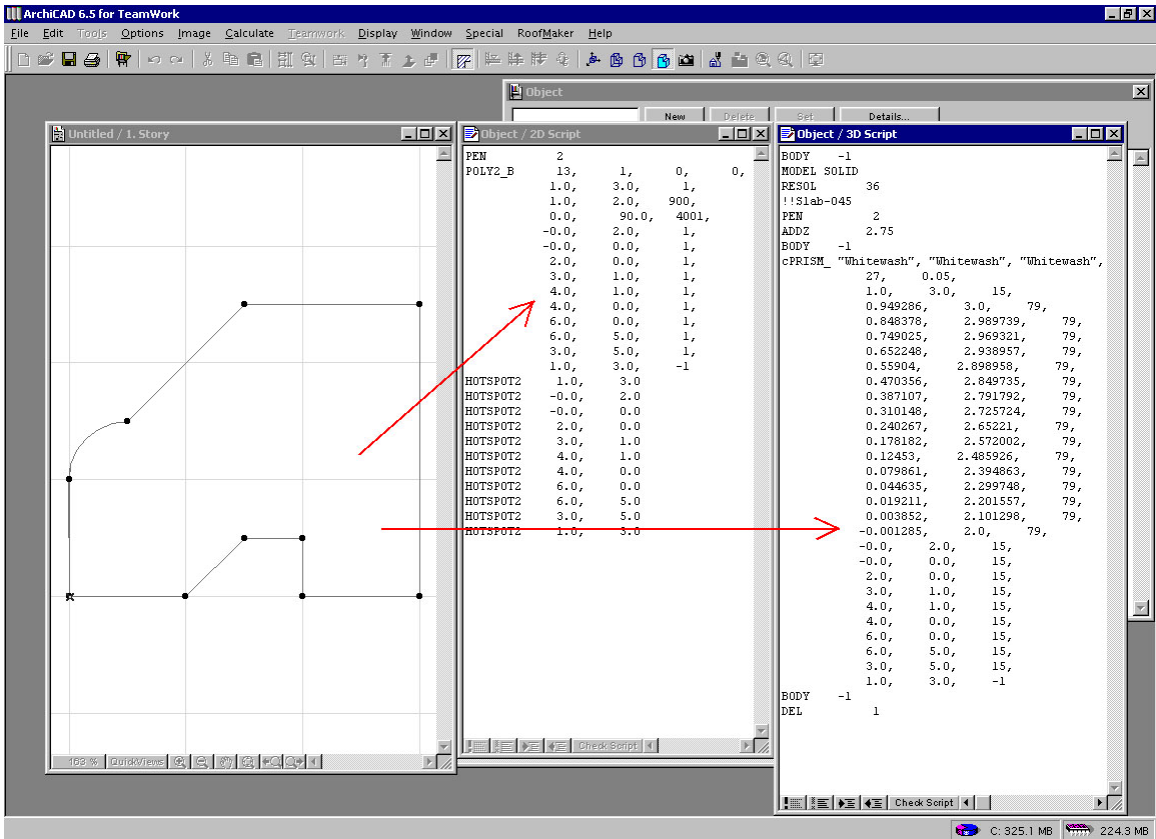
The above rules are applied to ARRAY parameters ONLY. When a simple parameter is found, there are no special restrictions, and exactly those operations are performed that you expect according to the dialog settings.

9.3 Drag and Drop Technique

This ArchiCAD feature allows you to easily define complex 2D and 3D shapes in GDL by drawing them on the plan file and dragging them into the 2D or the 3D script of an object.

How is it work?

- Draw the desired elements on the plan file.
- Open the GDL object's 2D or 3D script. Arrange the windows on your monitor to see both the script window and the plan view.
- Select the objects you want to drag.
- Hold down the left mouse button over the selected elements and move them on top of the script window.
- Release the mouse button.
- The GDL description of the selected elements appears in the 2D or the 3D script window.



Note: Curved polygons (slabs, fills, etc.) will be saved with status codes in the 2D script while in 3D script curves are replaced with line segments.

10. Quality Checking

During library development, regularly check library elements for all valid parameter combinations. Before giving your libraries to Graphisoft, make a final quality check on both Macintosh and Windows platforms.

10.1 The Quality Checking Process

- Load the library into the proper ArchiCAD version. Check if there are any missing library parts.
- Check the language and the length of all library elements and folder names.
- Place all elements from your library on the plan using the “Place all Objects” and “Place all Windows & Doors” commands from the Special menu.
- Check the 2D symbol of all elements on the plan.
- Open the 3D window and check the 3D model. If you get any error messages during this operation, there are errors in your library. Refer to the report file for a description of these errors.
- Browse your library in the object setting dialog and check preview pictures and user interface windows.
- Make sure that you set the correct parameter values in the objects.
- Test all list templates in the library.
- Place all zones on the plan file and check them.
- Delete cache and precompiled binary files from your library.
- Compare directories and library elements on both Macintosh and Windows versions.

Appendix

A: Standard ArchiCAD Object Parameters

Standard Door Parameter Names

3D parameters

RN	Rough Opening. Nominal Opening
DW	Door Width
FRT	Frame Thickness
FRW	Frame Width
LW	Leaf Width
LH	Leaf Height
RLTL	Right Leaf / Total Leaf Width
SLW	Smaller Leaf Width
LFW	Leaf Frame Width
LTH	Leaf Thickness
LWH	Leaf Width on the Hinge Side
LWK	Leaf Width on the Knob Side
LWL	Leaf Width on the Left
LWR	Leaf Width on the Right
LWT	Leaf Width on the Top
LWB	Leaf Width on the Bottom
AH	Arch Height
TRBD	Transom or Sidelight bar Depth
TRBW	Transom or Sidelight bar Width
JB	Jamb Width
JD	Jamb Depth
TRW	Trim Width
EHW	Exterior Head Trim Width
EJW	Exterior Jamb Trim Width
IHW	Interior Head Trim Width
IJW	Interior Jamb Trim Width
HPN	No. of Horizontal Panes
VPN	No. of Vertical Panes
SILL	Sill
WS	Extended Sill
SN	Extended Sill Nosing
LS	Leaf Style
TRS	Transom Style
SLS	Sidelight Style
HAP0	Handle, Postbox
DOP	Open Door
OPA3	Opening Angle in 3D
OP3	Opening in 3D (%)
OPLINES	Opening Line in 3D
XL	Left Opening Oversize
XR	Right Opening Oversize
YL	Lower Opening Oversize
YU	Upper Opening Oversize

2D parameters

OP2	Opening in 2D
ARD	Arch in 2D

DT2 Detailed Trim in 2D
RCFILL Reflected Ceiling Fill
RCS Reflected Ceiling Symbol
DLT Dashed Linetype

Cavity Closer

CLOSURE Cavity Closure
SKINS No. of Skins to turn
MRT Joint
MFILL Joint Fill

Pen

SPEN Swing Pen

Materials

LMAT1 Leaf Material 1
LMAT2 Leaf Material 2 (Glazing)
LMAT3 Leaf Material 3
GMAT Glass Material
GTSMAT Glass Material in Sidelight and Transom
ETMAT Exterior Trim Material
ITMAT Interior Trim Material
SILLMAT Sill Material
HMAT Handle Material

Standard Window Parameters

General

BHM Bottom Height of Horiz. Mullion
BRLT Brickwork Linetype
HPN No. of Horizontal Panes
VPN No. of Vertical Panes
HDH Height of Horizontal Division
TRH Transom Height
OSW Opening Sash Width
SFW Sash Frame Width
UAT2 Used as Transom
LN Lining
OPDIR Opening Direction
WB Window Board
BN Win. Board Nosing
WS Win. Sill
BS Win. Sill Nosing
OPLINES Opening Line in 3D
STYL Style
TRSTYL Transom Style
EHW Exterior Head Width
EJW Exterior Jamb Width
ESW Exterior Sill Trim Width
ETMAT Exterior Trim Material
ETW Exterior Trim Width
IHW Interior Head Width
IJW Interior Jamb Width
ISW Interior Sill Trim Width
ITMAT Interior Trim Material
ITW Interior Trim Width

XL	Left Opening Oversize
XR	Right Opening Oversize
YL	Lower Opening Oversize
YU	Upper Opening Oversize
RSTS	Right Sash / Total Sash Opening (%)

2D parameters

SILPEN	Sill Pen
GPEN	Glass Pen
DT2	Detailed Trim in 2D
TRO	Transom Opening

Cavity Closer

CLOSURE	Cavity Closure
SKINS	No. of Skins to turn
MRT	Joint
MFILL	Joint Fill

Materials

FRMAT	Frame Material
GMAT	Glass Material
WBMAT	Win. Board Material
WSMAT	Win. Sill Material

Special

A2D	Arc in 2D
BHA	Bottom Height of Arc
SLH	Shoulder Height
AH	Arc Height
RAH	Rail Height
RES	Resolution
TF2	Top edge from right
TNOD	Top Node
BANG	Angle on the Bottom
SANG	Angle on the Side
TANG	Angle on the Top

Bay and Bow

AST	Angle at Start
ACR	Angle at Corner
BYA	Bay Angle
BYD	Bay Depth
NUMS	No. of Front Sashes
SSW	Side Sash Width
ROOH	Roof Height
ROOFMAT	Roof Material
ROOTYP	Roof Type

Standard Object Parameter Names

General

TREED	3D
D3D	Detailed 3D
BFT	Background Fill Type
BPN	Background Fill Pen
CPN	Contour Pen

SW	Side Visible
MS	Minimal Space

Sanitary Ware

CTH	Counter Thickness
CTMAT	Counter Material
BSH	Back Splash Height
BSMAT	Back Splash Material
TOH	Toe Height
TOMAT	Toe Material
DTYP	Door Type
DMAT	Door Material
KTYP	Knob Type
KMAT	Knob Material
TATYP	Tap Material
BATYP	Basin Type
BAMAT	Basin Material
BTH	Bath Material
SDMAT	Side Material
DEP	Depth
GMAT	Glass Material
TAMAT	Tank Material
TDIS	Tank Distance from WC

Chairs and Tables

SETMAT	Seat Material
LESGMAT	Leg Material
FRMAT	Frame Material
BAMAT	Back Material
PLMAT	Pleating Material
TDEP	Table Depth
ROLMAT	Roller Material
ARMAT	Arm Material
CVMAT	Cover Material
TOMAT	Tabletop Material
BSMAT	Base Material
PSMAT	Post Material
BSS	Base Size (X,Y)
PSS	Post Size (X,Y)
TTT	Tabletop Thickness
TRMAT	Trim Material

Kitchen Units

UDH	Upper Door Height
LDH	Lower Door Height
DCC	Depth of Connected Cabinets
DRMAT	Drawer Material
DRN	Drawers Number
SHN	No. of Shelves
SHMAT	Shelf Material
SYLT	Symbol Linetype
BOMAT	Box Material
SKMAT	Switch Material
WSMAT	Work Surface Material

Lamps

AFO	Angle Falloff
AIL	Angle of inner Lightcone
AOL	Angle of outer Lightcone
ALA	Angle of Lower Arm
AUA	Angle of Upper Arm
ARL	Arm Length
BD	Bottom Diameter
BRL	Brightener Light
DIL	Diffuse Light
DIST1	Distance 1
DIST2	Distance 2
DIF	Distance Falloff
FXD	Fixture Depth
FXT	Fixture Thickness
FXW	Fixture Width
GSD	Glass Sphere Diameter
GSMAT	Glass Sphere Material
LIN	Lamp Inclination
LAMAT	Lamp Material
LBD	Lampshade Bottom Diameter
LSH	Lampshade Height
LSL	Lampshade Length
LSMAT	Lampshade Material
LRO	Lampshade Rotation
LUD	Lampshade upper diameter
LTMAT	Lampstand Material
LIR	Lightcone in Rendering
LBT	Luminous Body Thickness
PILMAT	Pillar Material
LCR	Lightcone Radius
RZ	Rotation 1 (around Z axis)
RX	Rotation 2
SC	Show Casting
SB	Show Bulb
SLC	Show Light Cone
SM	Socket Material

B: Standard GDL Script Format

! W Top hung 65

```
IF odr="Inside" THEN
  MULZ -1
  ADDZ -0.07
ENDIF
```

!***** CALCULATIONS

```
if vdv<0 then vdv=0
if hdv<0 then hdv=0
```

```
sn = max(0, sn)
bn = max(0, bn)
sd = max(0, k_+sn)
sx = c_-k_-0.07+bn
```

```
IF trah <= 0.18 OR trah >= b-0.16 THEN
  hh=b-0.03
  trc=0
ELSE
  hh=trah-0.005
  trc=1
ENDIF
```

MATERIAL fmat

!***** CUTPLANES

```
IF r_ =0 THEN
  szd=0
ELSE
  IF r_ >0 THEN
    szd=(a/2)*(sd+0.07)/(r_-0.07)
    aw1=atn((r_-0.07)/(a/2))
```

```
    ADD -a/2, 0, 0.07
    ROTY -aw1
    CUTPLANE
    DEL 2
```

```
    ADD a/2, 0, 0.07
    ROTY aw1
    CUTPLANE
    DEL 2
```

```
  ELSE
    szd=(a/2)*(sx+0.07)/(-r_)
    aw1=atn((-r_)/(a/2))
```

```
    ADDX -a/2
    ROTY aw1
    CUTPLANE 180
    DEL 2
```

```

    ADDX a/2
    ROTY -aw1
    CUTPLANE 180
    DEL 2
  ENDIF
ENDIF

|***** FRAME
ADDZ -0.025

PRISM 4, 0.025,
  -a/2-szd, b-0.02,
  a/2+szd, b-0.02,
  a/2+szd, b,
  -a/2-szd, b

ADDZ 0.01
|***** SASH (with bars)
IF vdv=0 THEN
  xdd=a-0.18
ELSE
  xdd=(a-0.18-vdv*0.03)/(vdv+1)
ENDIF

IF hdv=0 THEN
  ydd=hh-0.17
ELSE
  ydd=(hh-0.17-hdv*0.03)/(hdv+1)
ENDIF

ssx=-a/2+0.09
ssy=0.11

|***** HOLES

FOR iv=1 TO vdv+1 step 1

  FOR ih=1 TO hdv+1 step 1

    PUT ssx, ssy, 15,
      ssx+xdd, ssy, 15,
      ssx+xdd, ssy+ydd, 15,
      ssx, ssy+ydd, 15,
      ssx, ssy, -1

    ssy=ssy+ydd+0.03

  NEXT ih

  ssy=0.11
  ssx=ssx+xdd+0.03

NEXT iv

END

```

C: Standard ArchiFM Object Parameters

<u>ArchiFM DB parameter name</u> (GDL parameter name)	<u>GDL variable name</u>	<u>GDL variable type</u>
----------------------------------------------------------	--------------------------	--------------------------

Default Objects

There are no any special parameters.

Default Lamps

There are no any special parameters.

People

Main

First Name:	(FM_FirstName)	Text (max. 32 character)
Last Name:	(FM_LastName)	Text (max. 32 character)
Status:	(FM_EmpPosition)	Text (max. 32 character)
Engagement:	(FM_EmpEngagement)	Text (max. 32 character)
Department:	(FM_EmpDepartment)	Text (max. 32 character)
Phone Number at Work:	(FM_EmpTelExtension)	Text (max. 32 character)
Security Card Number:	(FM_EmpSecurityCardNumber)	Text (max. 32 character)
User Name:	(FM_EmpUserName)	Text (max. 32 character)
E-mail Address:	(FM_EmpEMail)	Text (max. 32 character)
Cellular Phone Number:	(FM_CellularPhoneNumber)	Text (max. 32 character)

Specialties

Admission Date:	(FM_EmpAdmissionDate)	Text (max. 32 character)
Secession Date:	(FM_EmpSecessionDate)	Text (max. 32 character)
Last Transfer Date:	(FM_EmpLastTransferDate)	Text (max. 32 character)
Car:	(FM_EmpHasCar)	Boolean
Type:	(FM_EmpCarType)	Text (max. 32 character)
Production Year:	(FM_EmpCarProductionYear)	Integer
License Plate Number:	(FM_EmpCarLicensePlateNumber)	Text (max. 32 character)
Phone Number at Home:	(FM_PhoneNumber)	Text (max. 32 character)
Identity Card Number:	(FM_IdentityCardNumber)	Text (max. 32 character)
Passport Number:	(FM_PassportNumber)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Elevators

Main

Netto Floor-space:	(FM_FloorSpace)	Real Number
Netto Floor-space Unit:	(FM_FloorSpaceUnit)	Text (max. 32 character)

Speed of Lifting:	(FM_Speed)	Real Number
Speed of Lifting Unit:	(FM_SpeedUnit)	Text (max. 32 character)
Lift Capacity:	(FM_LiftCapacity)	Real Number
Lift Capacity Unit:	(FM_LiftCapacityUnit)	Text (max. 32 character)
Headroom:	(FM_Headroom)	Real Number
Headroom Unit:	(FM_HeadroomUnit)	Text (max. 32 character)

Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer

Gross Weight:	(FM_ObjectWeight)	Real Number
Gross Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)

Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)

Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)

Additional Information:	(FM_Note)	Text (max. 32 character)
-------------------------	-----------	--------------------------

Specialties

Number of Doors:	(FM_Doors)	Integer
Number of Stops:	(FM_Stops)	Integer
Number of Transportable People:	(FM_TransportablePersons)	Integer

Lift Type:	(FM_LiftType)	Boolean
Suitable for Disabled People:	(FM_DisabledPersons)	Boolean

Security and Escapes

Main

Wattage:	(FM_PowerConsumption)	Real Number
Wattage Unit:	(FM_PowerConsumptionUnit)	Text (max. 32 character)
Self Power-supply:	(FM_SelfPower)	Boolean

Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)

ProductionYear:	(FM_ProductionYear)	Integer
Gross Weight:	(FM_ObjectWeight)	Real Number
Gross Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_InstallationDate)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)

Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_InvoiceNumber)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)

Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)

Additional Information:	(FM_Note)	Text (max. 32 character)
-------------------------	-----------	--------------------------

Stairs

Main

Number of Stories:	(FM_Stories)	Real Number
Number of Half-Landings:	(FM_HalfLandings)	Real Number
Height of Levels:	(FM_LevelHeight)	Real Number
Height of Levels Unit:	(FM_LevelHeightUnit)	Text (max. 32 character)
Headroom:	(FM_Headroom)	Real Number
Headroom Unit:	(FM_HeadroomUnit)	Text (max. 32 character)

Bearing Structure:	(FM_BearingStructure)	Text (max. 32 character)
Bearing Capacity:	(FM_BearingCapacity)	Real Number
Bearing Capacity Unit:	(FM_BearingCapacityUnit)	Text (max. 32 character)
Smokefree Staircase:	(FM_SmokeFree)	Boolean

Length of Steps:	(FM_StepsLength)	Real Number
Length of Steps Unit:	(FM_StepsLengthUnit)	Text (max. 32 character)

Specialties

Material of Footstep:	(FM_MaterialFootstep)	Text (max. 32 character)
Material of Stair Railing:	(FM_MaterialStairRailing)	Text (max. 32 character)
Material of Handhold:	(FM_MaterialHandhold)	Text (max. 32 character)

Additional Information:	(FM_Note)	Text (max. 32 character)
-------------------------	-----------	--------------------------

Work Spaces

Main

Area:	(FM_Area)	Real Number
Area Unit:	(FM_AreaUnit)	Text (max. 32 character)
User Name:	(FM_EmpUserName)	Text (max. 32 character)
Function:	(FM_Func)	Text (max. 32 character)
Floor Type:	(FM_FloorType)	Text (max. 32 character)
Floor Quality:	(FM_FloorQuality)	Integer
Additional Information:	(FM_Note)	Text (max. 32 character)

Electric Facilities

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Wattage:	(FM_PowerConsumption)	Real Number
Wattage Unit:	(FM_PowerConsumptionUnit)	Text (max. 32 character)
Egress:	(FM_Egress)	Real Number
Egress Unit:	(FM_EgressUnit)	Text (max. 32 character)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_InstallationDate)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_InvoiceNumber)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Distribution Panels

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Max. Rated Power:	(FM_MaxRatedPower)	Real Number
Max. Rated Power Unit:	(FM_MaxRatedPowerUnit)	Text (max. 32 character)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_InstallationDate)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_InvoiceNumber)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Measuring Places

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Load Shown by the Counter:	(FM_LoadCounter)	Real Number
Load Shown by the Counter Unit:	(FM_LoadCounterUnit)	Text (max. 32 character)
Throughput:	(FM_Throughput)	Real Number
Throughput Unit:	(FM_ThroughputUnit)	Text (max. 32 character)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Lamps

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Number Of Lights:	(FM_NumberOfLights)	Real Number
Wattage Per Light:	(FM_WattPerLight)	Real Number
Mobile Lamp:	(FM_Mobile)	Boolean
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)

Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

HVAC Equipment

Main

Max. Heat Output:	(FM_HeatOutput)	Real Number
Max. Heat Output Unit:	(FM_HeatOutputUnit)	Text (max. 32 character)
Max. Cooling Output:	(FM_RefrigeratingOutput)	Real Number
Max. Cooling Output Unit:	(FM_RefrigeratingOutputUnit)	Text (max. 32 character)
Wattage:	(FM_PowerConsumption)	Real Number
Wattage Unit:	(FM_PowerConsumptionUnit)	Text (max. 32 character)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Sanitation

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Locks

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Lock Number:	(FM_LockNumber)	Text (max. 32 character)
Key Number:	(FM_KeyNumber)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Built-in Wardrobes

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number

Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Furniture

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Ceilings

Main

Area:	(FM_Area)	Real Number
-------	-----------	-------------

Area Unit:	(FM_AreaUnit)	Text (max. 32 character)
Maximum Pipe Diameter:	(FM_MaxPipeDiameter)	Real Number
Maximum Pipe Diameter Unit:	(FM_MaxPipeDiameterUnit)	Text (max. 32 character)
Visible Frame Structure:	(FM_VisibleFrameStructure)	Boolean
Panel Sizes:	(FM_PanelSizeA)	Real Number
Panel Sizes:	(FM_PanelSizeB)	Real Number
Panel Size Unit:	(FM_PanelSizeUnit)	Text (max. 32 character)
Color:	(FM_Color)	Text (max. 32 character)
Material of Frame:	(FM_MaterialFrame)	Text (max. 32 character)
Material of Ceiling:	(FM_MaterialCeiling)	Text (max. 32 character)
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Floors

Main

Area:	(FM_Area)	Real Number
Area Unit:	(FM_AreaUnit)	Text (max. 32 character)
Floor Type:	(FM_FloorType)	Text (max. 32 character)
Floor Quality:	(FM_FloorQuality)	Integer

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)

Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Outdoor Objects

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Computers

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
------------	------------------	--------------------------

Wattage:	(FM_PowerConsumption)	Real Number
Wattage Unit:	(FM_PowerConsumptionUnit)	Text (max. 32 character)
Plug Number:	(FM_PlugNumber)	Text (max. 7 karakter)
HW Type:	(FM_HWType)	Text (max. 7 karakter)
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Service:	(FM_Service)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Specialties

Motherboard Type:	(FM_MotherboardType)	Text (max. 32 character)
Processor Type:	(FM_ProcessorType)	Text (max. 32 character)
Processor Speed:	(FM_ProcessorSpeed)	Real Number
Processor Speed Unit:	(FM_ProcessorSpeedUnit)	Text (max. 32 character)
Memory Type:	(FM_TetryType)	Text (max. 32 character)
Memory Capacity:	(FM_TetryCapacity)	Real Number
Memory Capacity Unit:	(FM_TetryCapacityUnit)	Text (max. 32 character)
HDD Type:	(FM_HDDType)	Text (max. 32 character)
HDD Capacity:	(FM_HDDCapacity)	Real Number
HDD Capacity Unit:	(FM_HDDCapacityUnit)	Text (max. 32 character)
Net Type:	(FM_NetType)	Text (max. 32 character)
Switch Port:	(FM_SwitchPort)	Text (max. 32 character)

Monitors

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Resolution:	(FM_Resolution)	Text (max. 7 karakter)
Special Adapter:	(FM_SpecialAdapter)	Text (max. 32 character)
Screen Size:	(FM_ScreenSize)	Real Number
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Software

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Operating System:	(FM_OperatingSystem)	Text (max. 32 character)
Number of Users:	(FM_NumberOfUsers)	Integer
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number

Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Phones

Main

User Name:	(FM_EmpUserName)	Text (max. 32 character)
Extension:	(FM_Extension)	Integer
Plug Number:	(FM_PhoneDigital)	Integer
Inventory Number:	(FM_InventoryNr)	Text (max. 32 character)
Serial Number:	(FM_SerialNumber)	Text (max. 32 character)
ProductionYear:	(FM_ProductionYear)	Integer
Weight:	(FM_ObjectWeight)	Real Number
Weight Unit:	(FM_ObjectWeightUnit)	Text (max. 32 character)

Attributes

Date of Purchase:	(FM_PurchaseDate)	Text (max. 32 character)
Date of Installation:	(FM_Installation Date)	Text (max. 32 character)
Warranty From:	(FM_WarrantyFrom)	Boolean
Warranty Period:	(FM_WarrantyPeriod)	Real Number
Warranty Period Unit:	(FM_WarrantyPeriodUnit)	Text (max. 32 character)
Purchase Price:	(FM_PurchasePrice)	Real Number
Purchase Price Unit:	(FM_PurchasePriceUnit)	Text (max. 32 character)
DepreciationPerYear:	(FM_DepreciationPerYear)	Real Number
Depreciation Per Year Unit:	(FM_DepreciationPerYearUnit)	Text (max. 32 character)
General Ledger Account Number:	(FM_GeneralLedgerAccNo)	Text (max. 32 character)
Invoice Number:	(FM_Invoice Number)	Text (max. 32 character)
Condition:	(FM_Condition)	Text (max. 32 character)
Manufacturer:	(FM_Manufacturer)	Text (max. 32 character)
Contractor:	(FM_Contractor)	Text (max. 32 character)
Servicer:	(FM_Servicer)	Text (max. 32 character)
Additional Information:	(FM_Note)	Text (max. 32 character)

Specialties

Phone Type:	(FM_PhoneType)	Text (max. 32 character)
PBX Port Number:	(FM_PBXPortNumber)	Text (max. 32 character)

Containers

Additional Information:	(FM_Note)	Text (max. 32 character)
-------------------------	-----------	--------------------------

Others

Subtype:	(FM_SubType)	Text (max. 20 character)
----------	--------------	--------------------------