

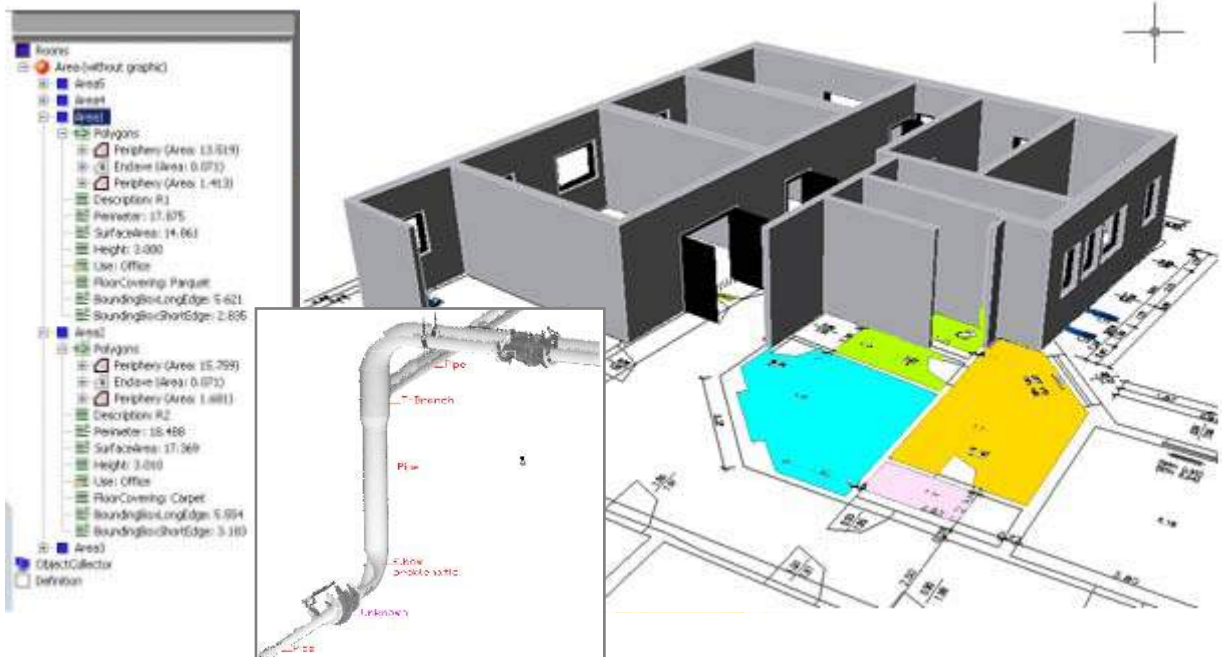


# As-Built

for AutoCAD® Software

Feature Data Capture and Management

Manual



# FARO

**SYNERGY**  
POSITIONING SYSTEMS

For more information contact Synergy Positioning Systems or visit the Synergy Positioning Systems website at [www.synergypositioning.co.nz](http://www.synergypositioning.co.nz)  
All branches: Phone 0800 867 266 Email: [info@synergypositioning.co.nz](mailto:info@synergypositioning.co.nz)



## Version 2018.0

The information and directions in this manual are up to date. FARO Technologies Inc. does not accept liability for the use, violation of patent laws or rights of third parties resulting from the use.

FARO Technologies Inc. reserves the right to carry out product alterations serving the technical progress at any time without previous notice.

No part of this documentation may be reproduced in any form (print, photocopy, or any other procedure) or processed, duplicated or distributed with the use of electronic systems without the previous written license of FARO Technologies Inc.

As at May 2018

### Contact

FARO Technologies, Inc.

[www.faro.com](http://www.faro.com)

Knowledge Base:

<https://knowledge.faro.com/>

### Trademarks

AutoCAD® and AutoCAD LT® are registered trademarks of Autodesk Inc.

## A few Words Concerning this Manual

Dear user,

We would like to thank you for the trust you put into us and wish you much success in the daily work with our software.

This manual is part of the As-Built for AutoCAD user documentation. **Please refer to the [As-Built for AutoCAD manual](#) for installation and system requirements.**

In order to provide an optimal start for the work with the feature data management, this manual contains two tutorials that lead you through typical project runs step by step.

Please do not hesitate to contact our support service if you have questions or in any case of problems. Your questions and suggestions are important to us and are always welcome. Our software gains from the ideas and the creativity of its users.

# Content

<b>1 Introduction</b> .....	<b>1-1</b>
1.1 What is the feature data management for? .....	1-1
1.2 Pattern recognition .....	1-2
1.3 Viewer mode .....	1-2
<b>2 Introductory Tutorial1 (Area Data Management)</b> .....	<b>2-1</b>
<b>3 Data Capture – Tutorial 2</b> .....	<b>3-1</b>
3.1 Preparation .....	3-1
3.2 The structure view .....	3-2
3.3 A new room .....	3-4
3.4 Inherited attributes and links .....	3-7
3.5 Concluding remark .....	3-8
<b>4 The Data Model</b> .....	<b>4-1</b>
4.1 Fundamental terms .....	4-1
4.1.1 Structure definition .....	4-1
4.1.2 Structure .....	4-1
4.1.3 Inheritance .....	4-2
4.1.4 Links .....	4-2
4.2 Creating a structure definition .....	4-5
4.2.1 Classes with special functionality .....	4-5
4.3 Project workflow .....	4-8
<b>5 Modeling and configuration – Tutorial 3</b> .....	<b>5-1</b>
5.1 Preparation and analysis .....	5-1
5.2 Draft of a structure definition .....	5-2
5.3 Realization of the draft in the software .....	5-4
5.3.1 A new structure .....	5-4
5.3.2 New classes and attributes .....	5-5
5.3.3 Define links .....	5-8
5.4 Add objects .....	5-10
5.5 Create consistency .....	5-14
5.6 Layout settings .....	5-16
5.7 Concluding considerations .....	5-17
<b>6 Tutorial Facade Damage Mapping (Part 1)</b> .....	<b>6-1</b>
6.1 Preparation .....	6-1
6.2 The structure view .....	6-2

6.3 Mapping of area data .....	6-3
6.4 Mapping of curve data.....	6-9
6.5 Creating links .....	6-10
6.6 Application of visualizations .....	6-12
<b>7 Tutorial Facade Damage Mapping (Part 2) .....</b>	<b>7-1</b>
7.1 Preparation.....	7-1
7.2 Changing the structure name .....	7-1
7.3 Customize an attribute definition .....	7-2
7.4 A new attribute .....	7-4
7.5 Sorting attributes .....	7-7
7.6 A new visualization.....	7-8
7.7 Configure legend.....	7-13
7.8 Visualization with symbols.....	7-16
7.9 Layout .....	7-18
7.10 Starting a new project .....	7-19
7.11 Effective mapping .....	7-20
7.11.1 Procedures of stone selection.....	7-20
<b>8 Tutorial – Archaeological Excavation Documentation.....</b>	<b>8-1</b>
<b>9 Export of the feature data .....</b>	<b>9-1</b>
9.1 Export of ASCII tables.....	9-1
9.2 Exporting shape files.....	9-5
9.3 Export of XML .....	9-5
<b>10 Data capture for Autodesk Architecture.....</b>	<b>10-1</b>
10.1 Overview of the procedure.....	10-1
10.2 The individual commands .....	10-2
10.2.1 Preparation .....	10-2
10.2.2 Survey .....	10-4
10.2.3 Creating Autodesk Architecture components.....	10-6
<b>11 Command reference feature data management.....</b>	<b>11-1</b>
11.1 Commands of the ribbon tab.....	11-1
Structure → Create structure definition.....	11-2
Structure → Load structure definition... ..	11-3
Structure → Save structure definition... ..	11-6
Structure → Remove structure .....	11-7
Structure → Structure view on.....	11-8

Structure → Structure view off .....	11-9
Structure → Load Autodesk Architecture styles.....	11-10
Structure → Visualize structure definition.....	11-11
Data capture - Fast Capture.....	11-12
Data capture - Repeat command .....	11-17
Data capture - Purge.....	11-18
Data capture - Copy .....	11-19
Data capture - Cut.....	11-20
Data capture - Paste .....	11-21
Data capture - Interpret plan .....	11-23
Data export - table(s).....	11-25
Data export - HTML.....	11-29
Data export - XML.....	11-31
Data export - Shape file.....	11-32
Data export - Configure export.....	11-33
Plan analysis → Find gaps.....	11-35
Plan analysis → Find short lines .....	11-37
Plan analysis → Find double lines.....	11-38
Plan analysis → Trace outline.....	11-39
Visualization → Update order of display.....	11-43
Visualization → Disable all .....	11-44
Visualization → Enable all.....	11-45
Visualization → Material visualization →Display on .....	11-46
Visualization → Material visualization →Display off .....	11-47
Help →Tutorials Feature Data → Tutorial 1/Tutorial 2/Tutorial 3.....	11-48
Help →Tutorials Facade Mapping → Tutorial 1/Tutorial 2.....	11-48
Help →Tutorial Archeological Excavation .....	11-48
Help → Configuration.....	11-50
11.2 Commands of the context menu of the structure view.....	11-53
Check/Create consistency.....	11-54
Find insecure attributes.....	11-55
Highlight "NAME" .....	11-56
Sort "NAME" objects (→ascending, →descending).....	11-57
Add "NAME" .....	11-58
Add "NAME" (Polyline/Line) \ Curve.....	11-59

Add "NAME" (Draw anew) \ Curve.....	11-60
Add "NAME" (Outline tracing) \ Curve.....	11-61
Add "NAME" (Polylines) \ Area .....	11-62
Add "NAME" (Inner point) \ Area.....	11-63
Add "NAME"(Draw anew) \ Area.....	11-66
Add "NAME"(Outline tracing) \ Area.....	11-67
Add "NAME" aligned \ Window, door .....	11-68
Add "NAME" (Pattern recognition) \ Window, door, wall .....	11-70
Export →Blocks.....	11-74
Export →Table .....	11-75
Export →HTML.....	11-76
Export →Shape file .....	11-77
Export →Regions .....	11-78
Export →Enclosing rectangle .....	11-79
Export →Create ADT-... \Room, Wall, Door, Window .....	11-80
Export →Create allplan rooms (XML file) .....	11-81
Import →Blocks.....	11-82
Import →Visualizations.....	11-83
Import →Table.....	11-84
Miscellaneous → Reduce polygon to one polyline each .....	11-86
Miscellaneous → Area proof .....	11-87
Miscellaneous → Replace arcs within polygons by chord polygons.....	11-88
Miscellaneous → Delete unnecessary interpolation points within polygons ....	11-89
Delete all "NAME" objects.....	11-90
Place label.....	11-91
Highlight \ Room, Area, Polygons.....	11-92
Show.....	11-93
Show ID in label.....	11-94
Show ID in header .....	11-95
Change ID .....	11-96
Delete .....	11-97
Assign other graphic.....	11-98
Adopt link.....	11-99
Copy .....	11-100
Dissolve link to "NAME" .....	11-101



## Content

---

Modify (inner point) .....	11-102
Modify (interpolation points) .....	11-103
Assign other polyline .....	11-105
Change.....	11-106
Update.....	11-107
Show in label.....	11-108
Show in header .....	11-109
Filter objects with this value .....	11-110
Visualization... ..	11-111
Link with "NAME" .....	11-113
Add and link "NAME" → Add "NAME" .....	11-114
11.3 Commands of the context menu of the object collector .....	11-115
Add object collector .....	11-116
Remove all object collectors.....	11-117
Add objects .....	11-118
Remove all objects from object collector "NAME" .....	11-119
Rename "NAME" .....	11-120
Delete "NAME" .....	11-121
Delete all objects permanently .....	11-122
Remove from object collector .....	11-123
11.4 Commands of the context menu of the structure definition.....	11-124
Rename structure.....	11-124
Structure wizard .....	11-125
Add class.....	11-126
Insert copy.....	11-137
Sort classes, show/hide.....	11-138
Configure... (class) .....	11-139
Copy class.....	11-140
Remove class.....	11-141
Add base class.....	11-142
Remove base class.....	11-143
Add attribute .....	11-144
Insert from Clipboard.....	11-161
Sort attributes, show/hide.....	11-162
Change attribute definition .....	11-163

Copy attribute definition .....	11-164
Delete attribute definition .....	11-165
Define visualization.....	11-166
Add link.....	11-176
Remove link.....	11-180
Configure... (links) .....	11-181

# 1 Introduction

## 1.1 What is the feature data management for?

The feature data management in the FARO products for AutoCAD is a very powerful tool. It can be used for the highly efficient recording and structuring of building floor plans. With the aim of creating qualified CAFM<sup>3</sup> suitable data structures. Components and pipelines in industrial plants can be labeled and managed or a green space and tree cadastre can be created. The graphic objects present in the drawing are linked to the feature attribute data recorded in the structured tree view.

The structuring concepts of our software exceed typical CAD standards:

- As common in relational data bases, so called **n:m relations** can be used to link objects.
- All **attributes** are captured **typed** (number or text, selection list, key attributes etc.) to assure data consistency and speed up the filling in during capturing.
- **Plausibility** can easily be defined in advance and its consistency with real data can be checked constantly during the capturing.
- Any derived measurements (e.g. wall surface area = room perimeter x room height – openings) can be determined by **calculation attributes**.
- Due to a variety of **visualization** methods, captured objects can be visualized on the basis of their attributes. This allows a visual overview of captured attributes or the review of a complete survey of all data in the plan.

The typical methodology when processing a project with the FARO feature data management is divided into following steps **Modeling, capturing** and **export**.

1. The data **structure of the target system (data base) is modeled**. Thus you define what is to be captured. You determine which objects are to appear with which attributes and in which way the objects are to be related (e.g. you decide whether a door belongs to one room or two rooms and which attributes are to be captured for the door) If requirement specifications for the data capturing exist, they should be included during the data structure definition.
2. The actual **data capture** is run according to the model (see 1). Should ground plans not exist at this stage, our software can also support the creation of the plans significantly (e.g. correctly scaled and aligned symbols for windows and doors). In any case feature objects will be created, attributes will be filled in and relations will be built. Most of the time, derived geometry objects (e.g. room or storey poly lines) will be created as well.
3. **Data are exported**. Mostly, readable ASCII data are used in tabular form. Often the target system also requires special AutoCAD constructions such as blocks or 3D Autodesk Architecture<sup>4</sup> components. The large variety of possible configurations allows designing the export in such a way that exactly those

---

<sup>3</sup> Computer Aided Facility Management

<sup>4</sup> A vertical Solution of AutoCAD for modeling buildings with 3D elements

data that are required by the target system or specification requirements are created without the need of much post-processing.

Our feature data management offers manual, semi-automatic and fully automatic functions for the quick and consistent structuring of CAD plans and drawings. These can be combined, depending on the task.

The software is fully integrated into AutoCAD/AutoCAD LT. This ensures the smooth interaction of non-graphic structures and AutoCAD graphics.

We do not see our feature data management as a complete CAFM system. The software will help you in terms of CAD to capture and structure existing geometry data for such systems.

## 1.2 Pattern recognition

A well-tried technology for pattern recognition is used for the automatic interpretation of ground plans. The system, which is called **Ypsilon**, has been developed at the Technical University Dresden and at kubit in years of research. The system represents cutting edge technology in this field. Through a complex process of optimization, the system determines and puts out the most likely interpretation of a plan or plan section.

This process mostly runs in the background and supports the creation of room and storey polygons, of doors and windows and, if necessary, of walls. User defined patterns (e.g. exploded AutoCAD blocks) are also recognized by the system. The fault tolerance of the pattern recognition can compensate plan faults (e.g. gaps, overlaps, double lines etc.) if necessary. There is, of course, also an option to search for such faults and highlight them.

## 1.3 Viewer mode

Our software is protected against illegal use by a license. You may install the software on as many computers as you like but the full functionality is only available when an active license exists.

If no license is available, the software works in a viewer mode. That means all objects, attributes and classes may be viewed but not be changed. This means that As-Built for AutoAD can be distributed free of charge together with the structured data.

## 2 Introductory Tutorial1 (Area Data Management)

The following [video](#) introduces you to working with area management using a simple example. Not all the details are explained, only the basic principles.

A detailed description of all commands used can be found in the command reference of this manual.

We recommend that you follow the tutorial step-by-step to get a first impression of the possibilities offered by the feature data management.



## 3 Data Capture – Tutorial 2

In this section an easy example will make you familiar with working with the feature data management. At this stage we leave out some details, which will be explained later. For a detailed description of all commands please refer to the →Command reference.

For the time being, this example only deals with **structured data capture**.

We recommend to follow the tutorial step by step in order to get a first impression of the possibilities of the FARO feature data management.

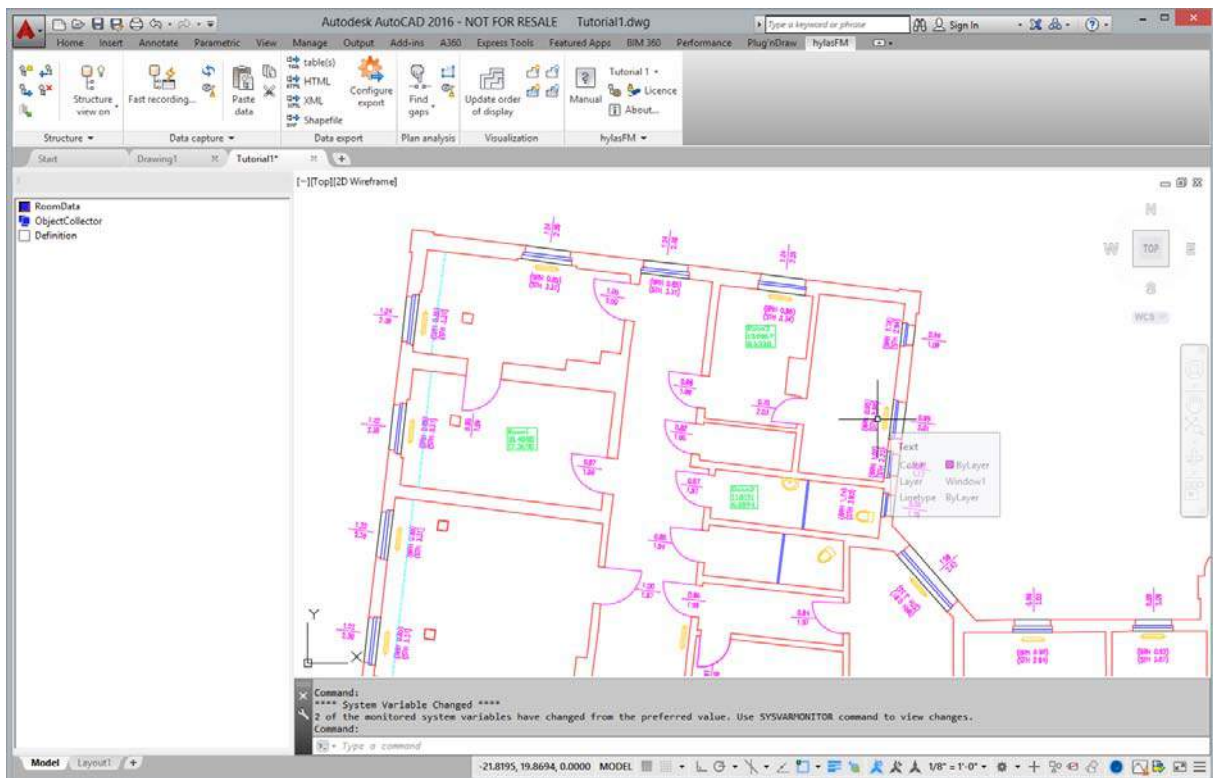
### 3.1 Preparation

To process the tutorial you need a fully installed version of AutoCAD or AutoCAD LT in which As-Built for AutoAD has been integrated.

After starting AutoCAD you will find the entry “Tutorials Feature Data →Tutorial 2” in the Feature Data ribbon group Help. If you select this entry, the two files TUTORIAL2.DWG and TUTORIAL2\_RESULT.DWG will be copied into the directory

```
\\USERS\[NAME]\DOCUMENTS\FARO\AS-BUILT          XXX          EN  
ACXXX\TUTORIALS\FEATURE_DATA\GENERAL\TUTORIAL2
```

The drawing TUTORIAL2.DWG will be opened in AutoCAD automatically.





## 3.2 The structure view


In this drawing you see the ground plan of a building.


You find the structure view of left of the drawing area. In this window all non-graphic data - that the software recognizes or that are added manually - are managed. It is a tree structure similar to the one you know from the Windows Explorer. The structure view is the central control element of the feature data management.

The structure view consists of three parts.

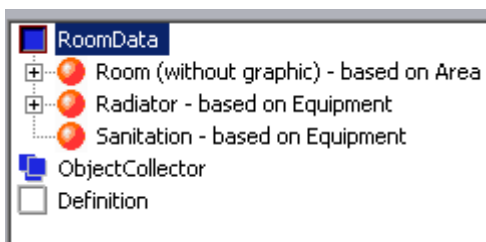
Under the blue symbol "  *RoomData*" there are objects and attributes (sorted according to classes) that have already been inserted into the structure.

Under the symbol "  " there are object collectors. Such object collectors are used to edit several objects or their attributes at the same time. When opening a drawing, there are no object collectors.



Under the grey symbol "  *Definition*" you see definition data for this structure. New classes, their attributes and links can be created and existing ones can be edited there.

In this tutorial we only deal with the **upper part** "  *RoomData*".

Open the tree structure with a double click on that entry. The sub-entries that appear now can again be extended by double clicks. The appearance may be like this.

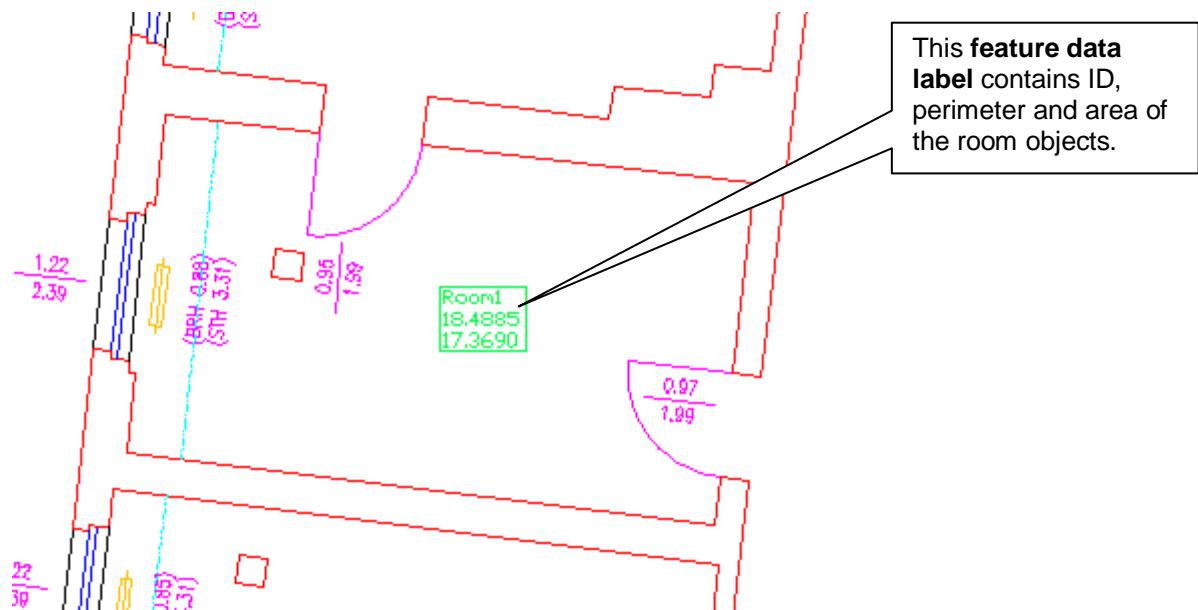


The entries represented by the symbol "  " are **object classes**. For example we have the class "  *Room*" and the class "  *Radiator*".

Now open the entry "  *Room*". You will see three room objects, which are tagged with that symbol "  ".

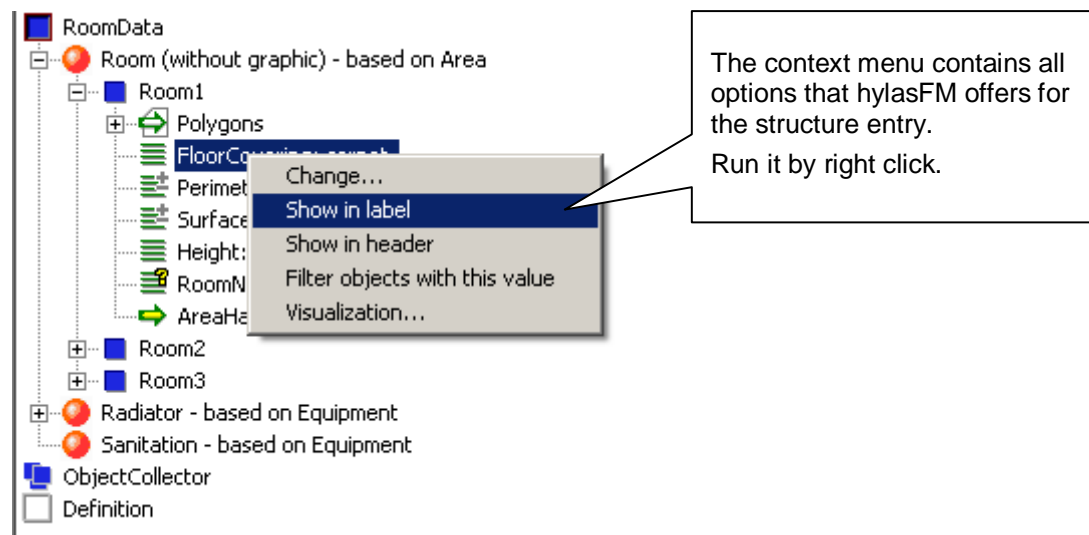
Then have a look at the drawing next to it. You will find the three rooms in the upper left part of the plan. They are marked with a green label, which contains the room ID (e.g. "Room1") as well as the perimeter and area.





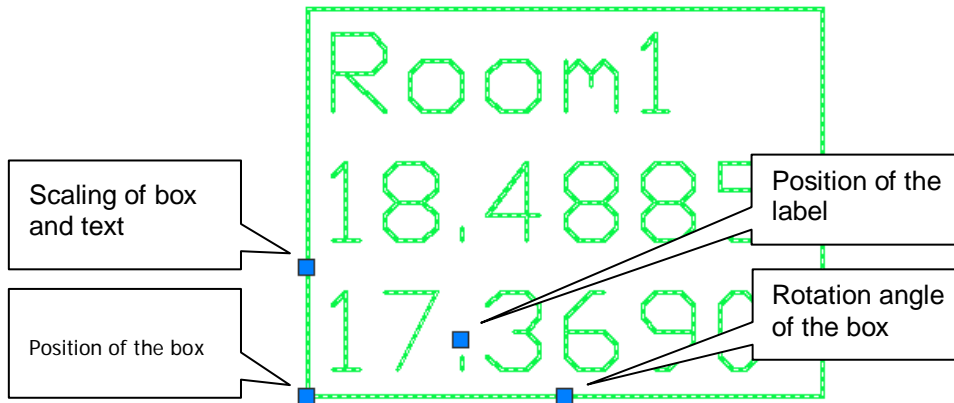
The entries in the structure view are linked with the geometry of the drawing. Every entry in the structure belongs to one label and geometry elements of the drawing. Click on the label "Room1" in the drawing. The corresponding entry in the structure view will open automatically. If you, vice versa, click on the entry "Room1" in the structure view the label and the corresponding polyline will be highlighted as dashed line.

The objects possess assigned **attributes**. Open the object "Room1" by double clicking on it. The attributes that exist for this object ("FloorCovering", "Perimeter", "SurfaceArea", "Height" and "RoomNumber") will appear. You can also make them visible in the label. Mark the sub-entry "FloorCovering: carpet" and open the context menu with a right click.



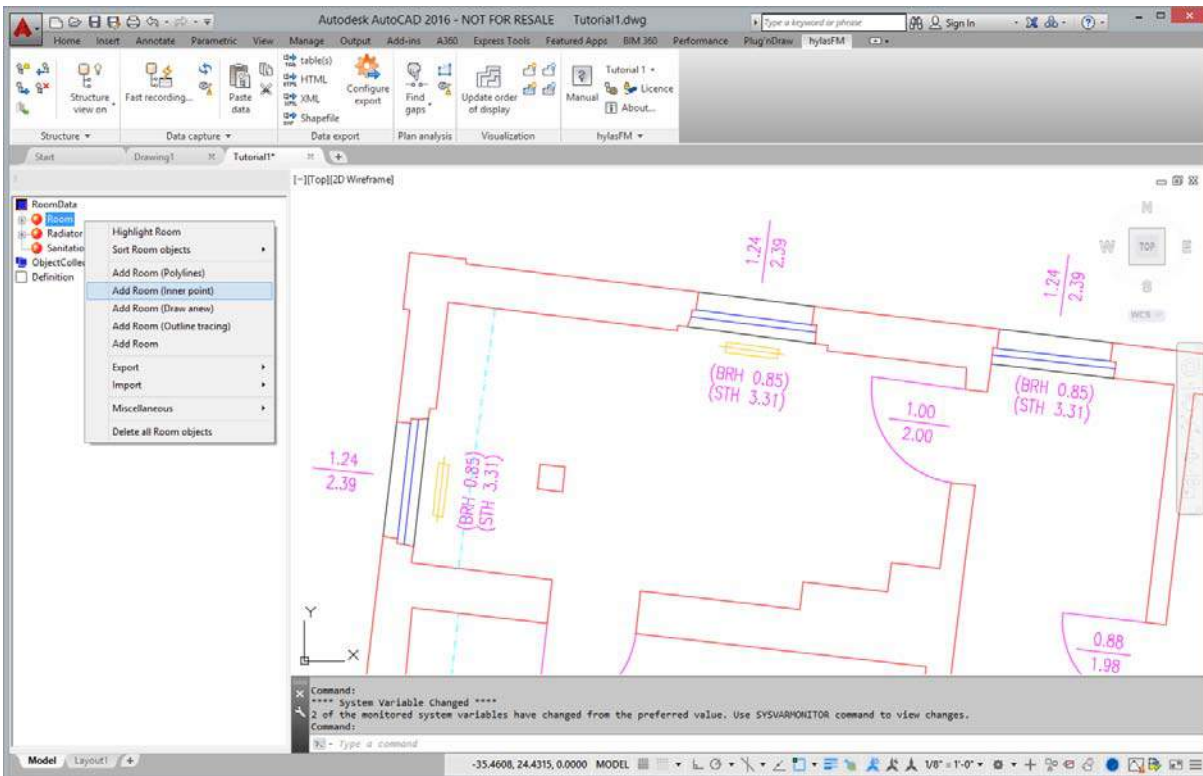
Click on **Show in label**. Thus, the floor covering will be displayed in all room labels. Just try out the settings.

The labels can be manipulated by four handles – just like AutoCAD elements. Just try it out.

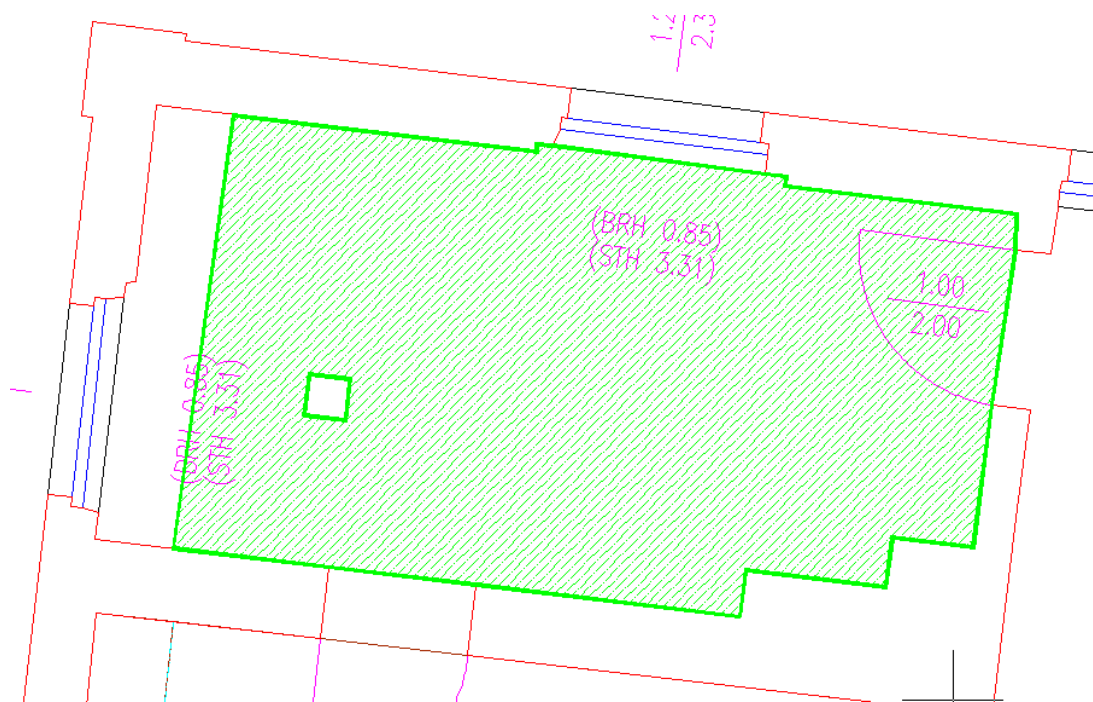


### 3.3 A new room

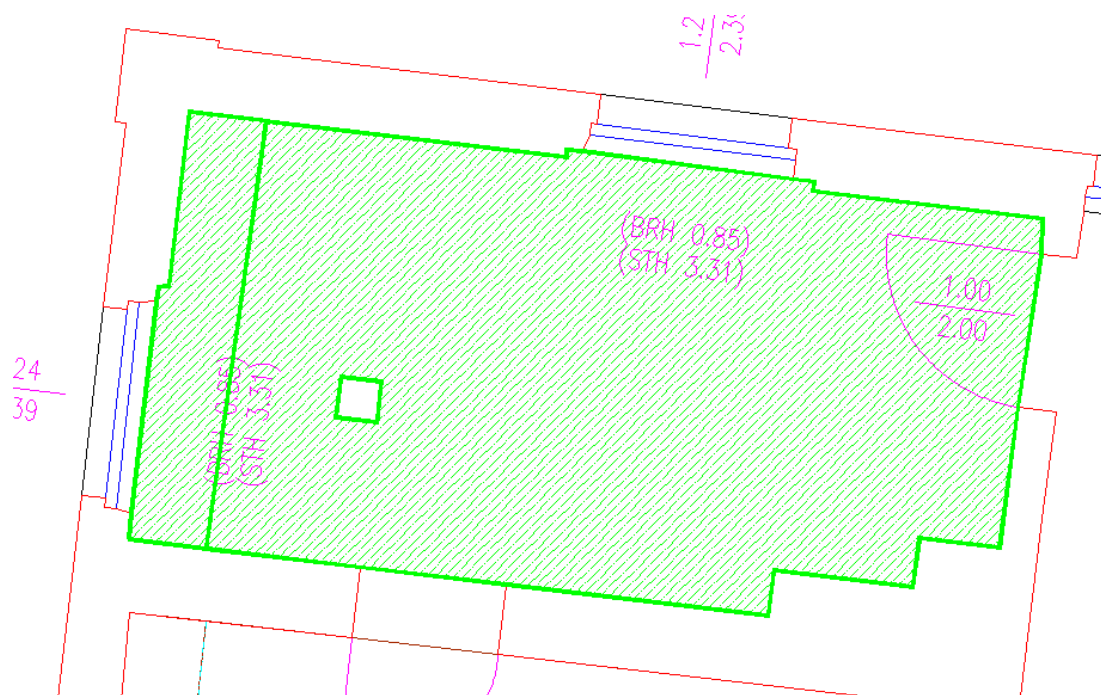
In the next step we want to create a new room. To do so, zoom to the room next to Room1 above it in the drawing. The room must be fully visible on your screen.



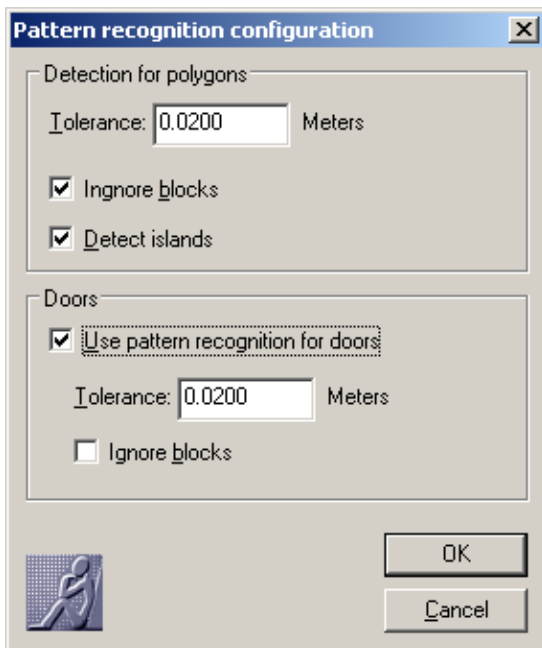
Mark the entry "Room" in the structure view and open the context menu with a right click. Select **→Add Room (Inner point)**. This command automatically determines the wall polygon by clicking inside the room. Now click any point right of the cyan colored dashed line within the room.



After a short waiting time a boundary is recognized and hatched it in green. It encloses the complete room apart from the area next to the window on the left. The rectangular column is omitted due to enclave recognition. Thanks to pattern recognition, the software has automatically recognized that the room ends at the door even though there is no direct reference line. On the other hand, the cyan colored dashed line, which runs across the room, was interpreted as a boundary line. However, this line only represents a bevel. In order to add the missing room area, click into the room area left of the line. Then a boundary and a hatching will be created there, too.



The command has an option “Configure”, that you can enable via the context menu (right mouse click into the drawing) or via the command line. Following dialogue will appear.



- **“Tolerance”** indicates how large the gaps between two lines may be so that the outline tracing skips these gaps.
- **“Ignore blocks”** means that AutoCAD blocks are not analyzed and are not used as room outline.
- **“Detect islands”** means that closed lines within the room are automatically recognized as “enclaves” and thus not taken into consideration when calculating the surface area. The column in this room is therefore not included in the area calculation. Thus, we leave the island recognition switched on.
- **“Use pattern recognition for doors”** means that door symbols are automatically recognized and considered when determining room boundaries. We leave the pattern recognition switched on, too.

Close the configuration dialogue with OK. You will then get back to the drawing.

Now the whole room is captured and you can confirm this with the **Return** key.


After that the software prompts you in the command line to indicate an insertion point for the label. Place it somewhere in the middle of the room. You will now see a new entry "■ Room4" with all its polygons and attributes in the structure view. The ID "Room4" is automatically assigned.

The polygons will be represented within the drawing by closed polylines. In order to make those visible, turn on the layer "YSubarea" via the AutoCAD layer control.

We will now adjust some attributes of the new room. At the moment “carpet” is selected as floor covering. However, the room has been laid out with parquet flooring. To change this mark the entry "■ Room1" – "≡ FloorCovering: carpet" and open the context menu with the right mouse button. Now go to Floor covering → **Change...** and select “parquet” from the drop down menu. Then confirm with **OK**.

Do the same procedure to change the height of the room. Go to the entry **Height** → **Change...** in the context menu. You will then be prompted to enter a new value for the room height.

### 3.4 Inherited attributes and links

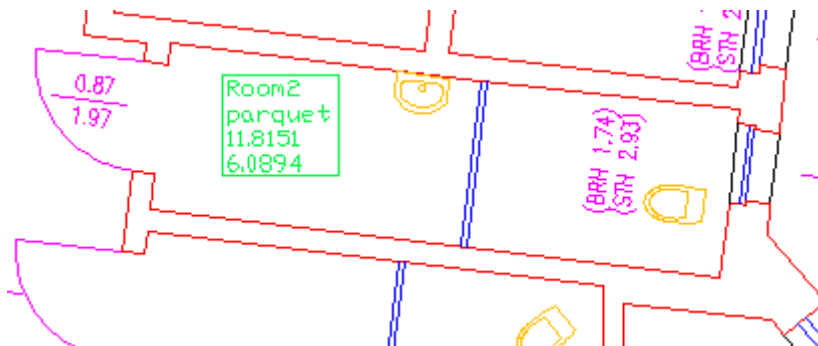
You may already have noticed that there are classes that contain, next to their name the addition "based on XXX". For example "Sanitation – based on Equipment". This means that the class "Sanitation" **inherits** from the class "Equipment". The class "Equipment" is not visible in the upper part of the structure. It only exists as base class and has therefore been hidden. However it is displayed in the lower part under  Definition.

**Inheritance** means that attributes have been transferred from the "base class" to the "derived class". All attributes and links that are defined for "Equipment" are also available in the class "Sanitation".

Which advantages and opportunities does inheritance offer?

Inheritance is used when classes have a lot in common or can be subsumed under one superordinate term. For example objects of the classes "Radiator" and "Sanitation" have in common that both have an inventory number. Both belong to building equipment. Therefore these two classes inherit from the class "Equipment". The attribute "Number" is defined for "Equipment" and is thus passed on to "Radiator" and "Sanitation" by inheritance. One example:

First we create a new sanitation object in the structure. Switch on the layers "Equipment" and "Equipment1". Yellow symbols for radiators and sanitary facilities will appear in some rooms. Some radiator symbols already have a label attached to them and are thus integrated in the structure. The already existing Room2 contains the sanitation "Washbasin". This washbasin is now to be assigned to the structure.



Scroll to the class "Sanitation" in the structure view. Open the context menu with the right mouse button and go to **➔Add Sanitation**. Following request will appear in the command line:

Command: HYLASCONTEXT

Select geometry for Sanitation:

Select object:



Now click on the washbasin in Room2 and confirm with RETURN. The software will now ask for the:




Insertion point for label:



Click on the edge of the washbasin at which the label is to be inserted. As of now, the washbasin "Sanitation1" exists in the structure.

Have a look at the two **attributes**. You can change the “Type” from the standard setting “Toilet” to “Washbasin”. The attribute “Number” has automatically been filled in. This attribute is of the type KEY, i.e. it is a definite key value that cannot exist twice. It will automatically be raised for further entries. If one value has already been assigned, the next possible value will be used.

At this point **inheritance** comes into play. The attribute “Number” is defined for the class “Equipment”. Both, “Sanitation” and “Radiator” inherit from “Equipment”. Therefore both possess the attribute “Number” and the unambiguity of the key value is kept in both classes. Thus, overlaps in different classes can be excluded. The equipment objects, no matter to which class they belong, are all registered by this one definite value.

We will now have a look at the **links** since they too, are handed on from the base class to the derived classes. The class “Room” is linked with the class “Equipment” via a link with the name “AreaHasEquipment”. Due to inheritance, the class “Sanitation” also possesses this link. However, before we can see this, we need to update the newly created object “ Sanitation1”. Go to “ RoomData” and select the entry **Check/create consistency** in the context menu. Confirm the following dialogue with **OK**.

Now open the entry “ Room2” – “ AreaHasEquipment” in the structure. You will then see the just created object “Sanitation1”. That means the software has automatically recognized that “Sanitation1” is located within the polygon that belongs to Room2. Vice versa, you will find a reference to Room2 under “Sanitation1” – “ AreaHasEquipment”.

Do the following test. Move the washbasin with the attached label to another room (e.g. Room3). Now go to **Check/create consistency** under “ RoomData” in the context menu. The washbasin “ Sanitation1” will now automatically be linked with Room3.

The relation “AreaHasEquipment” has the characteristic that objects can be assigned to by their insertion point with areas (and through inheritance with rooms too) using the command **Check/create consistency**.

The first tutorial is herewith finished. Your plan should by now look somewhat like the plan in the file

```
USERS\[NAME]\DOCUMENTS\FARO\ AS-BUILT XXX EN
ACXXX\TUTORIALS\FEATURE_DATA\GENERAL\TUTORIAL2\
TUTORIAL2_RESULT.DWG
```

For further practice you may now try to create and edit some objects (e.g. rooms or equipment) on your own. When adding a room by the method “Inner point”, always keep in mind to zoom in the room as close as possible while it should not stick out of the area of display (before adding it).

### 3.5 Concluding remark

Through this procedure the structure will by and by be filled with data. A data structure that is related with geometry elements is being developed. This structure can be edited or extended at any time.

The captured data can be written into tables using the export functions. This is important if you should later decide to process the data in an external database or spreadsheet analysis.

When capturing data, you have to stick to the predefined data structure. If you want to capture additional information, such as completely new classes (e.g. furniture), you need to indicate this in the structure definition (grey area " *Definition*"). This procedure will be explained in detail in →Chapter 5 Modeling and configuration – Tutorial . However, we recommend to read →Chapter 4 The Data Model first as some of the fundamental concepts of data base theory are dealt with there.





## 4 The Data Model

The following paragraphs deal with relatively abstract matters. Do not worry if not everything seems to make sense right away. Take your time and you will see that it is worth the effort.

### 4.1 Fundamental terms

The following four terms take in a central role in the feature adat management.

<b>Class</b>	A set of similar objects. In a class the shared attributes and the types of links of its objects are defined.	e.g. "Room"
<b>Object</b>	Entity of a class – possesses attributes and CAD graphic	e.g."Room125"
<b>1. Attribute</b>	2. Non-graphic information of an object. (Number, Text, reference to another file); defined in the class	3. e.g. "Usage"
<b>4. Link</b>	5. Link between objects; defined in classes	6. e.g. "AreaHasEquipment"

Using these four terms, the following sections will explain what is meant by **structure definition**, **structure**, **inheritance** and **link** in our software.

#### 4.1.1 Structure definition

**Structure definition** means all information that is not tied to certain objects. These are

- the classes (name; base classes; layer and color allocation; attribute),
- their attributes (name; definition, e.g. number with decimal places) and
- the links (name; linked pair of classes).

In a particular project that consists of many drawings you will always use the same structure definition just like you use a certain layer structure, certain colors and line signatures in AutoCAD for a complete project that consists of several drawings. A structure definition can be saved via **→Save structure definition** or can be assigned to a current drawing via **→Load structure definition**.

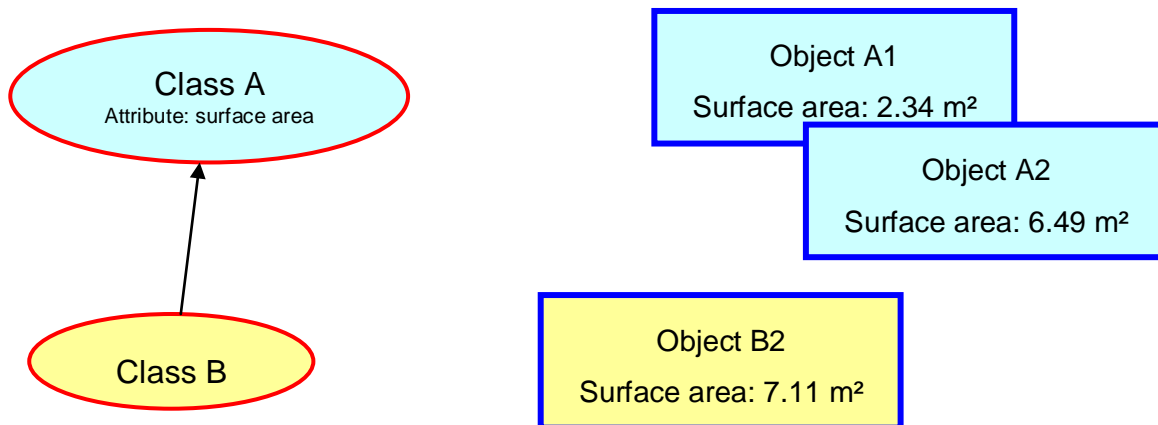
#### 4.1.2 Structure

A **structure** contains the structure definition but also all objects with their specific data and their CAD graphics. The structure (and thus its structure definition) is automatically saved in the AutoCAD drawing. A complete structure can be saved externally and be made available to other applications (database or spreadsheet) with the commands **→Export table(s)**, **Export HTML** and **→Export XML**.

Two drawings of a project will thus always have the same structure definition but different structures.

### 4.1.3 Inheritance

A very important term in the feature data management is inheritance, which means the transfer of attributes and characteristics from a base class to a derived class. If a class A possesses a surface area, a class B that inherits from class A will also possess this attribute. All objects of classes A and B will be assigned a surface area. In the following illustration classes are represented by ellipses, objects by rectangles and inheritance (B inherits from A) by an arrow. By the way, the imagery of the structure view follows this concept.



For each class you determine in the structure definition whether and which base class(es) it possesses.

Inheritance can express similarities between classes. If you change an attribute that is used in several classes, you only have to do so once.

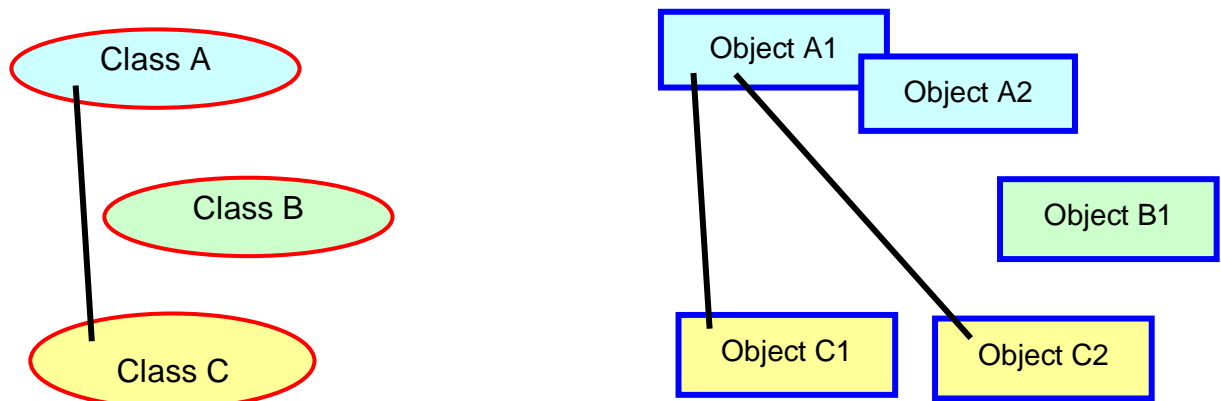
### 4.1.4 Links

Links connect objects with each other. Should you, for example, want to know who the tenant of a room object room17 is and how the room is equipped later, there need to be links between the room object room17 and the tenant object as well as the room object room17 and the inventory object(s).

In the structure definition you determine, to which classes the objects that can be linked need to belong. In our example there would be

- a link “uses” between the classes “Tenant” and “Room” (TenantUsesRoom),
- a link “has” between the classes “Room” and “Inventory” (RoomHasInventory)

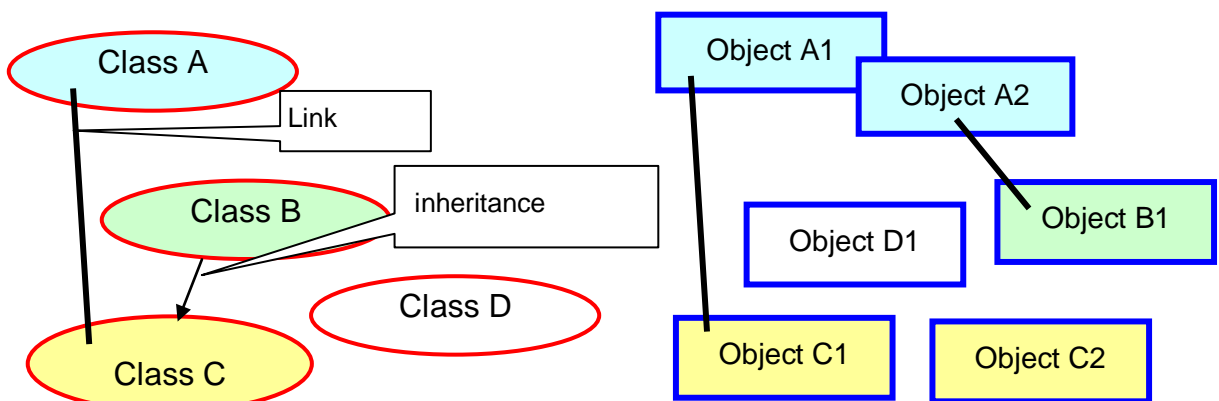
Following figure illustrates the definition of entities and relations:



In this figure a link between A and C has been created in the definition (on the left: classes). Therefore it is possible to link A1 with C1 and C2 in a particular structure (on the right: objects of a structure ). A relation between A1 and B1 is not possible. Not all objects that could be related have to be related (A2).

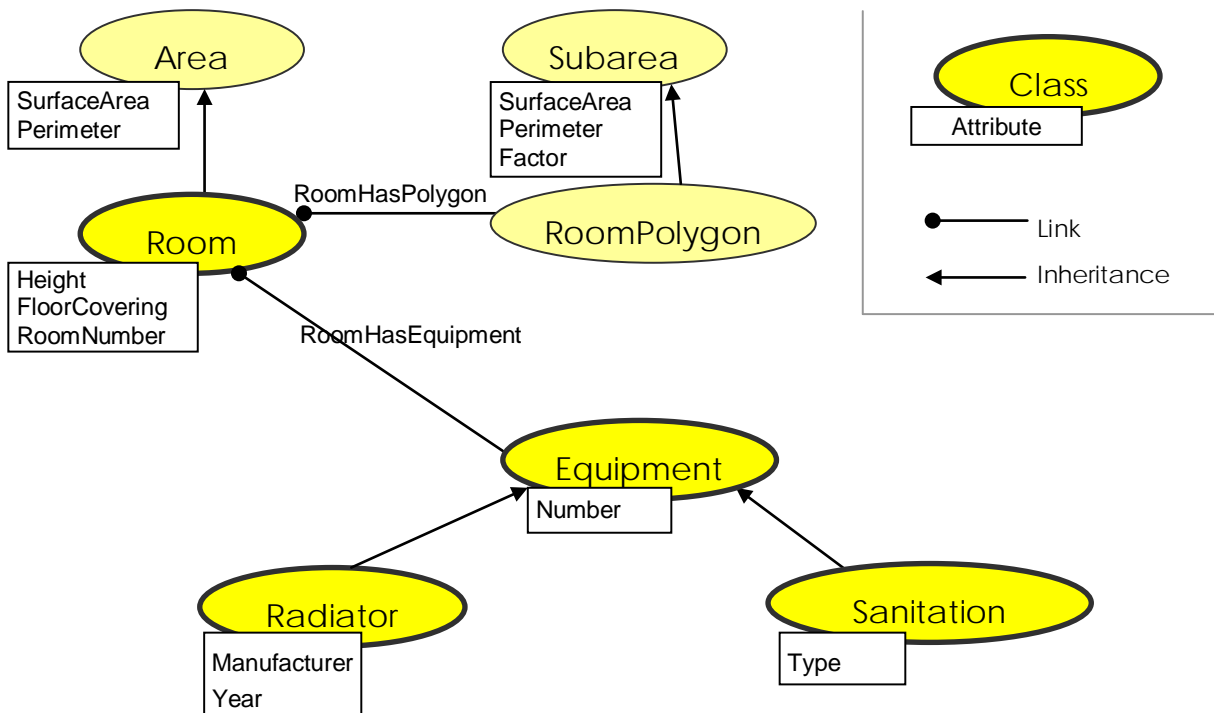
**Correlation of link and inheritance**

Inheritance also influences links. Have a look at following figure:



In the class definition there is a link between class A and class C. Additionally, class B inherits from class C. Now object A2 can be linked with B1, although a direct link between class A and class B has not been defined. The link between A and C is handed on to B. A link between A1 and D1 is not allowed.

The following illustration shows the class graph from tutorial 2.



There are special relations for certain applications:

### Link characteristic “Equipment”

A useful characteristic can be assigned to links between the class “Area” and any other class – the characteristic “Equipment”.

*Every object of the class “Area” knows which equipment it contains.*

More precisely: If an object lies within the subarea polygon that belongs to an “Area”, it will automatically be linked with it. Decisive for the automatic generation of a link are following cases

- The object has a label. A link is generated if the insertion point of the object label lies within the area polygon.
- The object has been derived from the class “Area”. A link is generated when both areas overlap.
- The object has been derived from the class “Curve”. A link is generated when curve and areas intersect.

To use this characteristic, go to "**Links**" in the definition part of the structure (" **Definition**") and run the configuration dialogue **Configure...** (links). Activate the option

“Link ‘class1’ on the basis of position with ‘class2’ (Equipment).“

Such links are identified by a yellow arrow. Since updating these links can be time-consuming, you have to initiate it manually. To do so use the command **Check/Create consistency...** in the context menu of the structure.

### Link characteristic “Subarea”

If one class (e.g. “Room”) inherits from “Area” the property

*“Consider ‘RoomSubarea’ when calculating the surface area for ‘Room’ (Subarea)”*

can be assigned to the link. The link will be tagged with a white arrow .

This characteristic has the effect that the surface area and perimeter of the “Area” objects result from the weighted sum of all related polygons. In the example a “Room” will, thus, consist of several “RoomSubareas”, that are all summed up. For further explanation refer to →4.2.1

## 4.2 Creating a structure definition

In order to edit a drawing with feature data tools, this drawing needs to get a structure definition first. In this definition you define *what* is to be included. I.e. you determine classes and their attributes as well as links.

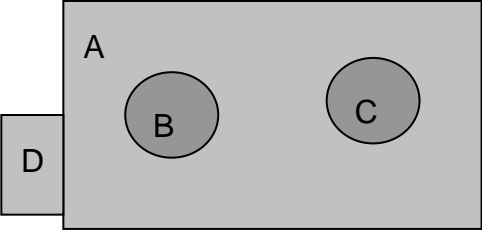
In an open drawing a structure definition can be created with the command →**Create structure definition** or a stored one can be opened with →**Load structure definition**. The current drawing will then contain this structure definition.

If a new structure is created, the structure wizard opens immediately. The structure wizard supports you when creating predefined classes. The next section will explain these classes and the structure wizard.


### 4.2.1 Classes with special functionality

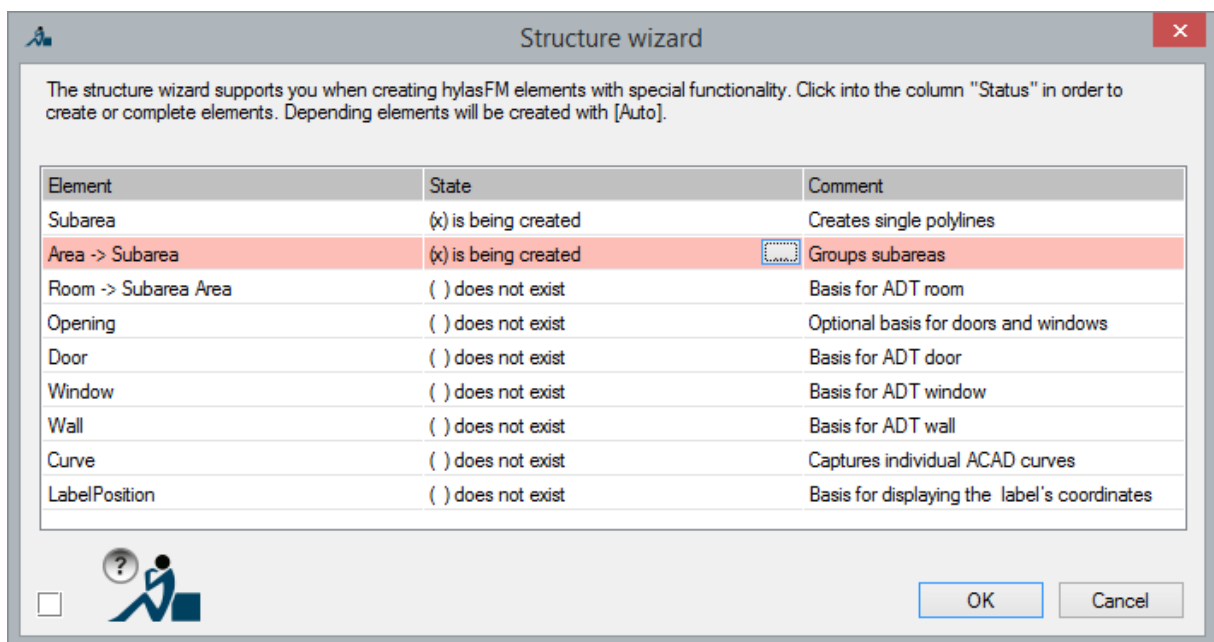
Some classes in the feature data management possess special properties and can be created with the structure wizard:

Subarea / Polygon	<p>Generally, feature data objects consist of a label and any associated AutoCAD graphic. Objects of the class “Subarea”, however, can only be related to <b>exactly one 2D polyline</b>. The software calculates their surface area, multiplies it by a factor, that can be configured, and passes on this value to the corresponding subarea object in the structure. Within the structure the subareas can be found under "Polygons" within the linked area classes.</p> <p>The software offers an intelligent outline recognition for the creation of subareas. In the context menu of the class "Subarea" (respectively in the context menu for the "Polygons" within the area class) there are following entries</p> <p>→<b>Add subarea (Inner point)</b>. Here you have to indicate a point inside the area. The boundary of this area is automatically recognized and a polyline and a subarea object created. Also door symbols are recognized and the polyline is drawn correspondingly.</p> <p>→<b>Add subarea (Outline tracing)</b>. Here you have to click near a set of lines/polylines. The software will follow the outline as far as possible and will create a polyline and subarea object, too. Click near a line/polyline.</p>
-------------------	--

	Drawing faults like small gaps between the lines will be skipped (depending on the set fault tolerance)
Area	<p>The automatic outline recognition is also available for the creation of areas.</p> <p>The difference to the subarea is that enclaves can be recognized here, too. Area objects themselves do not have a graphic. The recognized polyline(s) will rather be converted into one or more subarea/polygon object(s) that will then automatically connect with the area object. The surface area of the "Area" object is calculated from the sum of its related subareas.</p> <p><b>Example:</b> The area of a room A with two columns B and C is to be captured exactly. A recess D is only to be considered with the factor 0.5. The total area results from the following sum:</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 20px;"> <math display="block">A - 1xB - 1xC + 0.5xD</math> </div>  </div> <p>In order to allow this mechanism to run smoothly, a class "Subarea" and a link between the two classes must exist. The "Subarea property" must also be set for the link (Command reference → <b>Configure...(Relations)</b>).</p> <p>If you select the option "Area" on the tab "Type" while defining a class, a polygon/subarea class as well as an according link to it will automatically be generated for the new class.</p>
Room	Basis for AutoCAD Architecture room; inherits from Area;
Opening	Optional basis for door and window
Door	<p>There are special creation commands for the "Door" class. With the aid of pattern recognition, a door in the ground plan is detected, a corresponding object is created and some attributes (width, opening direction) are automatically filled in.</p> <p>Basis for AutoCAD Architecture door; attribute "Width" is automatically taken over from drawing when a door is being created</p>
Window	<p>There are special creation commands for the "Window" class. With the aid of pattern recognition, a window in the ground plan is detected, a corresponding object is created and some attributes (width, opening direction) are automatically filled in.</p> <p>Basis for AutoCAD Architecture window; attribute "Width" is automatically taken over from drawing when a window is being created</p>
Wall	There are special creation commands for the class "Wall". Pattern

	<p>recognition makes it possible to create a wall with just two clicked points. The correct alignment and thickness of the wall are determined, a corresponding feature data object is created and some attributes (length, wall thickness) are automatically filled in.</p> <p>Basis for AutoCAD Architecture wall; attribute "Width" is automatically taken over when from the drawing when a wall is being created</p>
Curve	<p><b>Exactly one AutoCAD Curve object</b> can be linked with an object of the type "Curve" (e.g. line, polyline, arc, spline...) The attribute "Length" is automatically filled in when a curve object is being created and contains the length of the selected curve.</p> <p>The automatic outline tracing is also available for the creation of curves. It is called up with the following context menu command:</p> <p>➔ <b>Add curve (Outline tracing)</b>. Click close to a set of curves and the software will create an approximated polyline for the curve object.</p>
Label-position	Basis for the display of label coordinates

All these classes can be created or completed with the structure wizard (Context menu of  *Definition* ➔ **Structure wizard...**)



Typically, these special classes are not directly used for data capturing but only as base class for others. The features of these classes are handed on to the derived classes by inheritance. The special classes will mostly be hidden in the upper part of the structure tree so that they are invisible during the data capturing. An example for such a construction is to be found in tutorial 2.

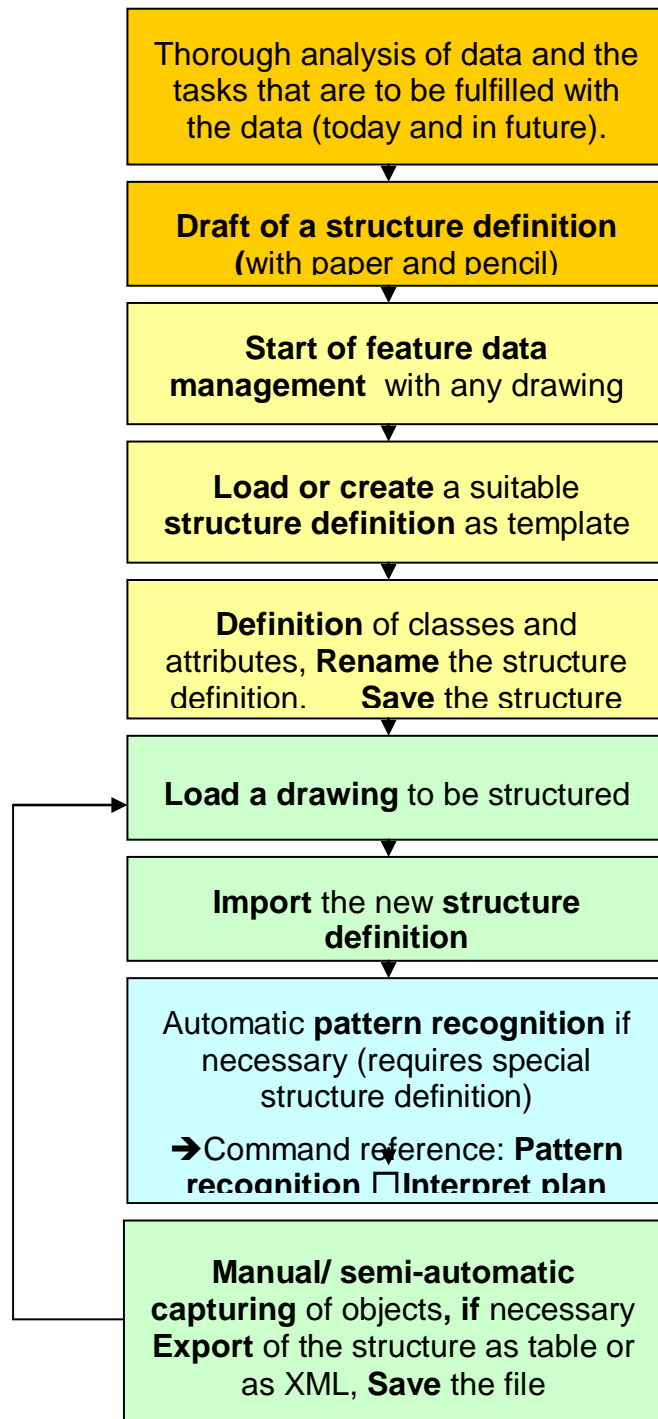
### 4.3 Project workflow

We recommend to **plan** the **change of a structure definition**, i.e. editing or adding classes and attributes **carefully**, although or even since this process is very simple in our software. To undo a change is often not possible without data loss if objects have already been captured. Therefore, the determination of a structure definition has a long-ranging impact: Tediously captured objects and links are only as good as their underlying structure. If in doubt, please do not hesitate to contact our support team.

If you create a new structure definition by editing or adding classes, always choose a new name. You can enter a new name either when you are creating the structure or later. To do so use the context menu command "Rename structure" under " Definition".

" *Unnamed*" thus becomes e.g. " *Office*".

The schema on the right illustrates the recommended workflow of a project.





## 5 Modeling and configuration – Tutorial 3

This section introduces you to data modeling for the feature data management by means of an easy example. At the same time, different ways of creating and relating objects will be explained. You will also learn the most important ways to a custom-fit configuration.

We recommend to follow tutorial step by step. Even if you do not develop own models but choose to use precast structure definitions, important functions for the data capturing will be explained in this tutorial, too.

You will find a detailed description of all commands in the chapter →**Command reference**.

### 5.1 Preparation and analysis

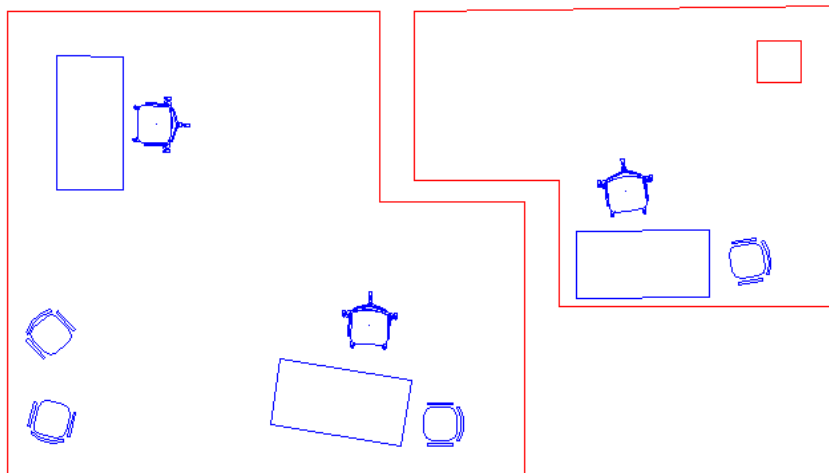
In order to work with this tutorial, you need a fully installed version of AutoCAD or AutoCAD LT into which As-Built for AutoCAD has been integrated.

You should also have read the previous section →**4 The Data Model**, which provides the theoretical base for the following example.

After starting AutoCAD you will find the entry “Tutorials Feature Data →Tutorial 2” in the Feature Data ribbon group Help. If you select this entry, the two files TUTORIAL3.DWG and TUTORIAL3\_RESULT.DWG will be copied into the directory

```
\\USERS\[NAME]\DOCUMENTS\FARO\ AS-BUILT XXX EN  
ACXXXX\TUTORIALS\FEATURE_DATA\GENERAL\TUTORIAL3
```

The drawing TUTORIAL3.DWG will automatically be opened in AutoCAD.



The drawing contains two stylized rooms with office furniture and a pillar. There are three types of furniture: tables, chairs and chairs for guests. There are following informal requirements that the structure definition to be created needs to meet.

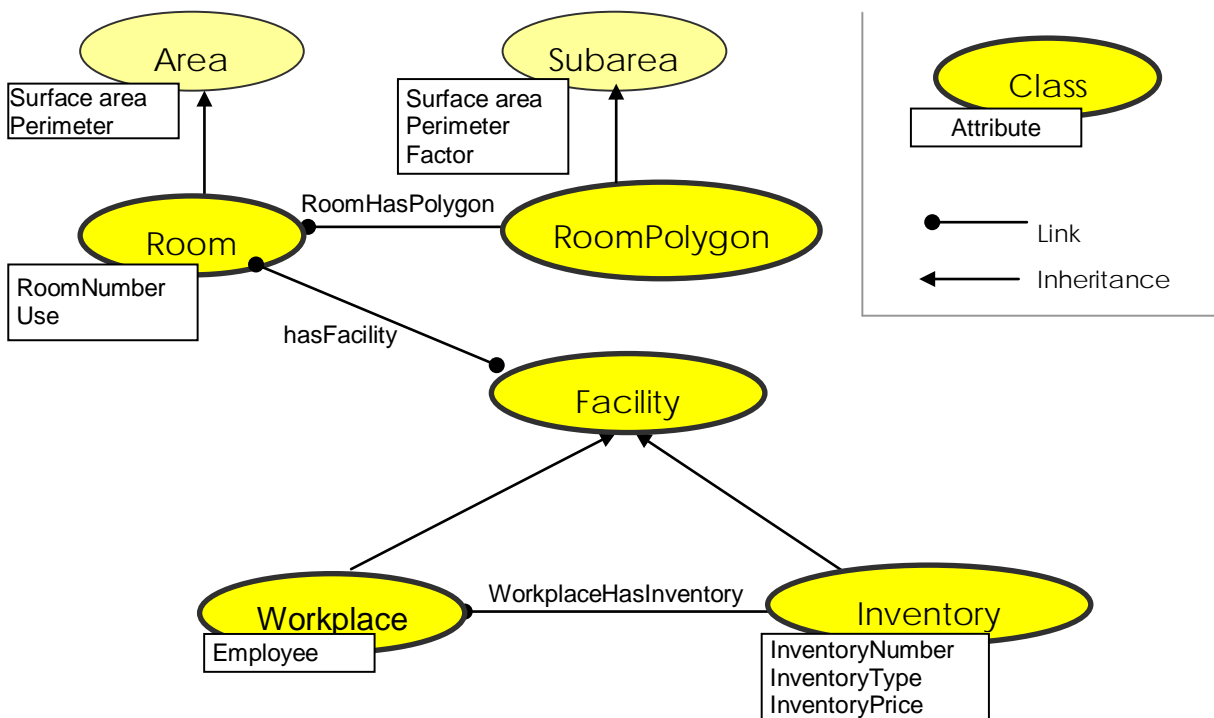
- Recording of rooms with surface area, room number and use,
- Recording of furniture and inventories with inventory number, price and type of furniture or facility,
- Grouping of some furniture to workplaces,

- Allocation of furniture and inventories to rooms,
- Assigning employees to work stations and
- there are many plans of this kind...

For practice you can try to design a structure definition similar to the illustration in section →4.1.4 on a piece of paper!

## 5.2 Draft of a structure definition

After due consideration you might get to following structure definition.



The predefined classes “Area” and “Subarea” have been used as basis for the classes "Room" and "Roompolygon" since the surface areas of the rooms need to be calculated and the facilities are to be assigned to rooms. Therefore, “Room” inherits from “Area” and "Roompolygon" from "Subarea". “Workplace” and “Inventory” inherit from “Facility” and are linked with the rooms by “hasFacility”.

The remaining information has been assigned to the respective classes as attributes. Attribute definitions could look as follows

- Employee: Text
- InventoryNumber: Key
- InventoryType: Selective list
- InventoryPrice: Decimal number
- RoomNumber: Key
- RoomUse: extendable selective list

The data type is indicated after the attribute name, with following meanings

**Text:** any character string that has to be entered when changing an attribute  
**Key:** a unique key value, i.e. every object receives another value for this attribute

**Selective list:** enumeration, i.e. a given value can be selected from a list for the attribute value ( **extendable selective list** = enumeration extendable by the user)

**Decimal number:** floating point number.

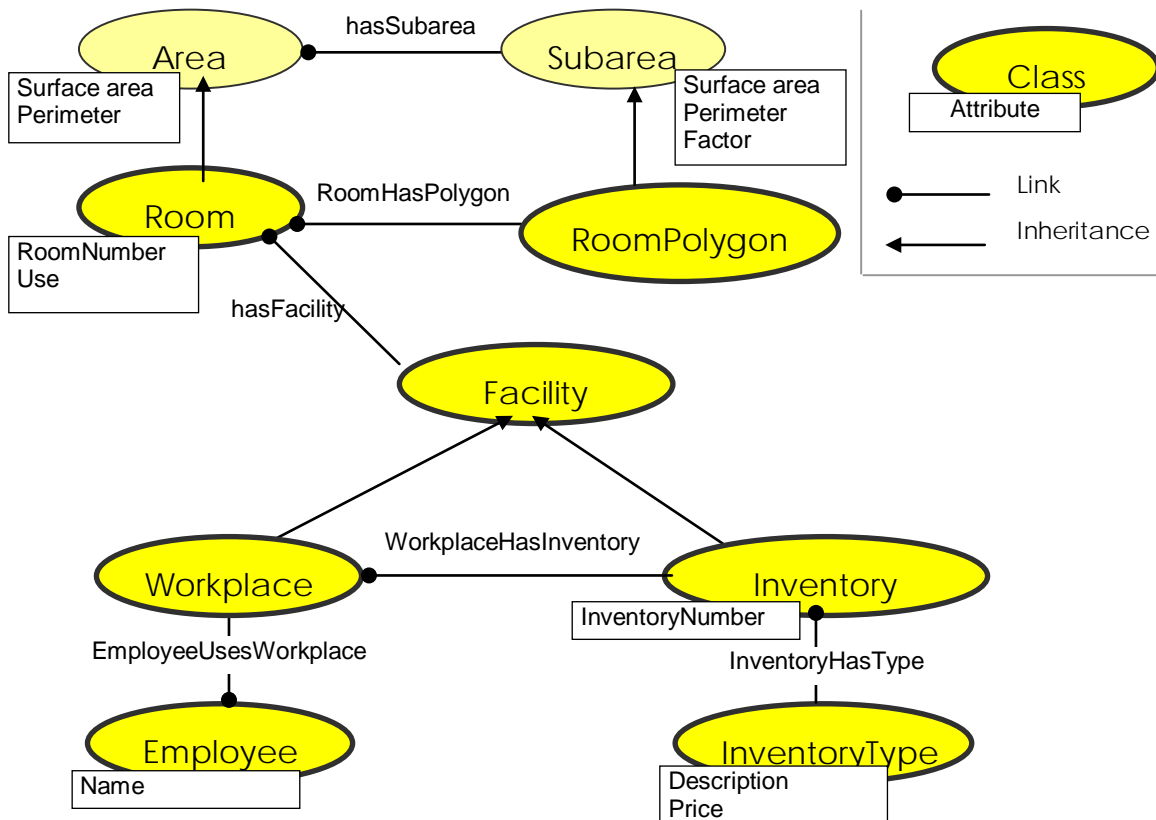
The default values for the attribute and momentarily not important information about the appearance of the attributes in the label stand after the type identification.

**Criticism of the first draft**

After further due consideration two main points of criticism of the first draft arise.

- The price of a piece of furniture or other inventory should be the same for all pieces of inventory of the same type. This property is not enforced by the first draft.
- Sometimes employees share a workplace. This fact cannot be implemented using the first draft.
- We also expect to capture more information on employees later (other than the name).

The following second draft eliminates all the deficiencies of the first draft.




Thus, the attribute definitions here can be:

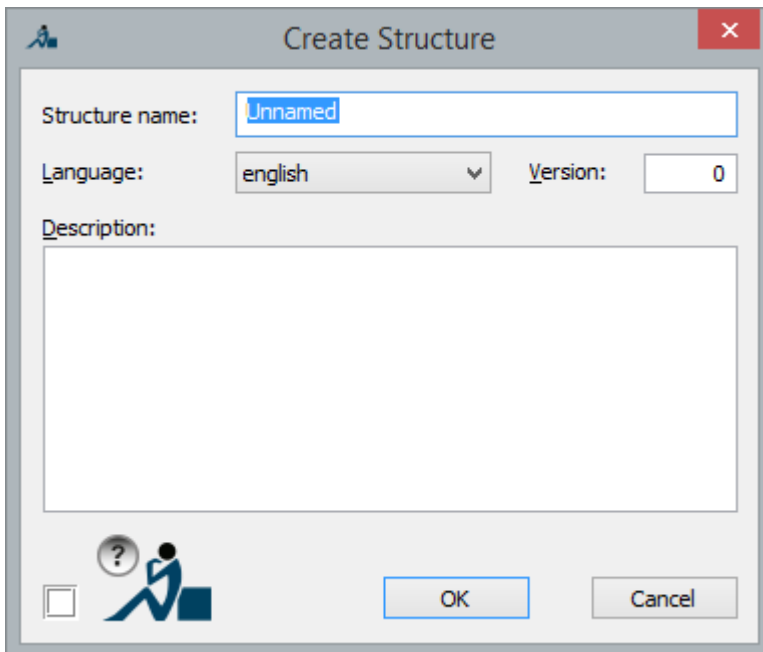
- Name: Text
- InventoryNumber: Key
- Description: Selective list
- Price: Decimal number
- RoomNumber: Key
- Use: Selective list

## 5.3 Realization of the draft in the software

### 5.3.1 A new structure

The second draft for the new structure shall be realized now.

First of all, we create the structure. For this call up the command → **Structure - Create structure definition**  from the feature data ribbon. Following dialogue will appear:




Enter the name of the structure under “Structure name”, e.g. “Office”. You can also add a short note under description. When you load the structure in future, the note will also be displayed as selection aid. Confirm your input with OK.

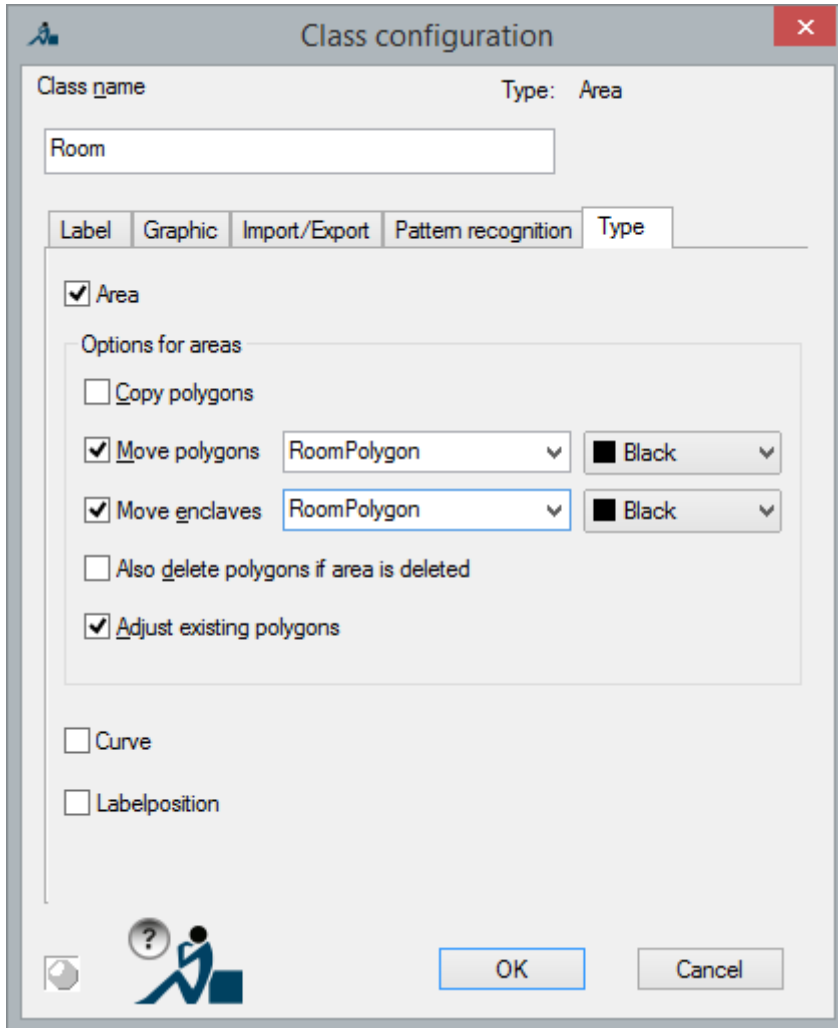
Then the **structure view** appears on the left. The structure view is subdivided into three parts. The lower grey part " Definition" contains all definitions and the upper colored part " Office" contains all concrete objects. Object collectors stand under "". Such object collectors are used to change several objects or their attributes at the same time. When opening a drawing, there are no object collectors.

Now we want to create a new structure definition and will therefore only be dealing with the **lower grey part** " Definition".

### 5.3.2 New classes and attributes





In the next step the new office classes are created according to our draft from section → 5.2. Start with the class “Room”.

For this go to the structure and open the subentries by double clicking "  Definition". For the subentry "Classes" open the context menu with the right mouse button and select → **Add class...** Now enter the class name “Room” and activate the checkbox „Area“ on the tab „type“. Now the class “Room” gets all properties from “Area”.



Confirm your entries with **OK**.

In the next step the attributes “RoomNumber” and “Use” of the class “Room” are defined.

Select the entry → **Add attribute...** from the context menu of "  Definition" → "  Classes" → "  Room" → "  Attribute". Fill in the dialogue that appears as follows:

**Attribute definition**

Atribute name  
RoomNumber

Description  
Description

Atribute type  
key

Fill-in help

Offer standard  
 Offer next in size value  
 Offer last value

Display attribute on label

Prefix Value Suffix

Note  
The attribute definition is consistent.  
Showing and hiding the attribute can be controlled via the context menu entry "Sort attributes...".

key  
Base value  
R01

key.{R01}.Next,{'', ''}

OK Cancel

Select **“Key”** as attribute type. Since all room numbers should be unique in a plan, this attribute type is especially suitable. Thus, it is guaranteed that no room number doubles.

Create another attribute “Usage”. The procedure is similar to the previous one:

Select "Extendable selective list" as attribute type. Write some default values into the extendable selective list, e.g. office, kitchen, corridor, bathroom.

The so defined attribute can get a value for every room from the given set of terms or a value that has been defined by the user<sup>3</sup>. “Offer last value” means that the value that has been selected or entered last will be offered next time. The modeling of the class “Room” is completed by confirming this attribute definition with OK.

Use the schema of the second draft from section 5.2 to create all other classes likewise “Inventory”, “InventoryType”, “Workplace”, “Employee”, “Facility”. For the time being keep the default settings in the “Class configuration”. Do not forget about the base classes and attributes. For the attributes you can take over the default settings but following settings have to be changed

- InventoryNumber (Attribute type: Key) of the class Inventory receives the base value “I01”
- Description (Attribute type: Selective list) of the class InventoryType receives the following values “OfficeChair”, “Chair” and “Table”.

<sup>3</sup> If you do not want that the data collector can add new terms later, select the attribute type “selective list”. Thus, a better data consistency can be enforced.

- Name (Attribute type: Text) of the class Employee. Delete the text “Comment” and leave the entry empty.
- Price (Attribute type: Decimal number) of the class InventoryType. Additionally set the display accuracy in the combobox from the default value <like AutoCAD> to 0.01.

The classes “InventoryType” and “Employee” do not have graphic representations in the drawing. Thus, they should only appear as entries in the structure. Therefore, call up the entry →**Configure...** in the context menus of “InventoryType” and “Employee”.

Disable the option “Object has label” on the tab “Label”. Confirm with OK.

The class "Inventory" shall get additional graphic. Therefore, activate the option “Link object with graphic (Relation with AutoCAD drawing elements)” on the tab “Graphic”.

At the bottom of the configuration dialogue on the tab “Label” there is the option “Prompt attributes immediately after object creation”. If you enable this box, all attributes are prompted upon the creation of an object. If you disable this box, the attributes are automatically filled in with the default values. This does not apply for the predefined attributes “SurfaceArea” and “Perimeter” of the Area classes as well as for the so-called calculation attributes (Attribute type "Numeric calculation attribute" and "Text calculation attribute"). The values of these attributes are always calculated automatically and therefore not prompted.

### 5.3.3 Define links

As of now, the links are still missing. Click on →**Add link...** in the context menu of "→ Links". Fill out the dialogue for the link “InventoryHasType” as follows.

The screenshot shows the "New type of link" dialog box. The "Name of link" field is filled with "InventoryHasType". Under "classes to be linked", "Inventory" is selected in the first dropdown and "InventoryType" in the second. In the "Properties of the link" section, the "Monitor number of objects to be linked (Cardinality)" checkbox is checked. Below it, the first dropdown is set to "{0,+∞}" and the second to "1". At the bottom, both "Also show linked objects" and "Also delete linked objects" are unchecked. The "OK" and "Cancel" buttons are visible at the bottom right.



The fields “Name of link” and “linked classes” are obligatory.

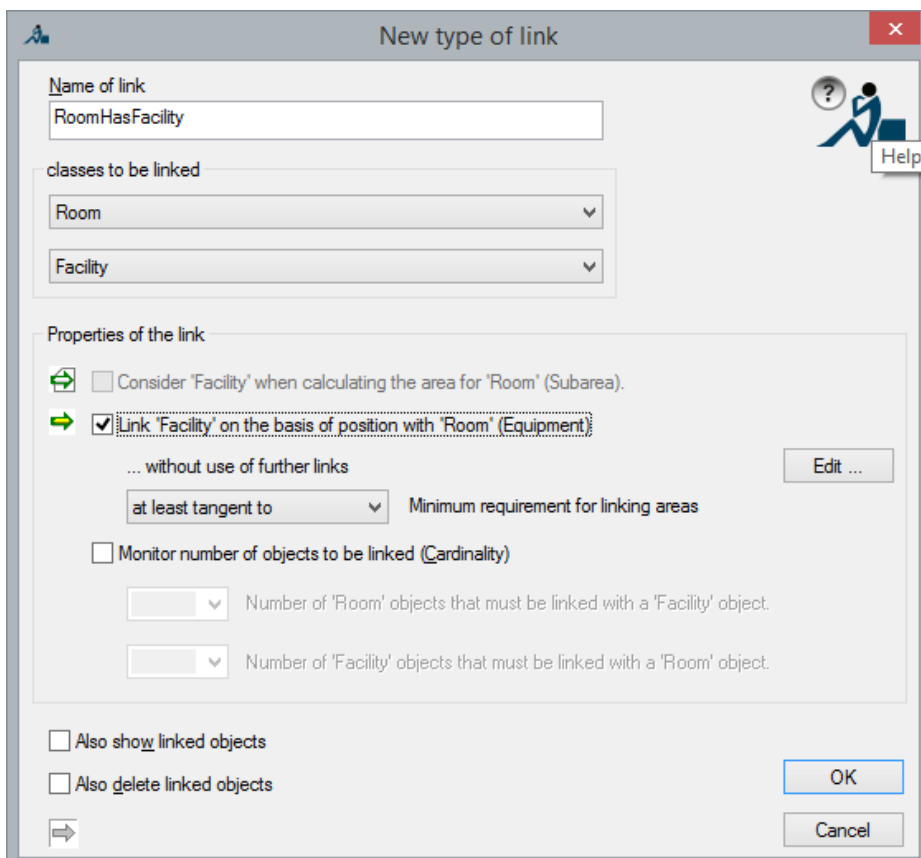
The option “Monitor number of objects to be linked (Cardinality)” helps to ensure the quality of data capture, by applying further restrictions. The default settings allow to link every object with any number of other objects. For example this behavior makes sense for a link “TenantUsesRoom” - Every tenant can use several rooms and every room can usually be assigned to several tenants. However, this standard behavior is unwanted for most links. For example with the link “InventoryHasType” every piece of “Inventory” should only have exactly one “InventoryType”. This is expressed by the interval {1,1}. This setting can ensure that every piece of inventory is assigned its type during capturing. An inventory type, on the other hand, can be assigned to any number of or even no piece of inventory. In the interval notation this is expressed by {0,+∞}. +∞ stands for *infinite*.

Our next step is to create the link “hasFacility” between the classes “Room” and “Facility”. Since “Room” inherits from “Area”, the option (marked by a yellow arrow)

“Link 'Facility' on the basis of position with 'Room' (Equipment) “

is offered. That means that a facility whose insertion point lies within a room (respectively its polyline) will be assigned to this room automatically (also see. → Section 4.1.4 Links)

Following figure shows the situation before closing the dialogue with OK.



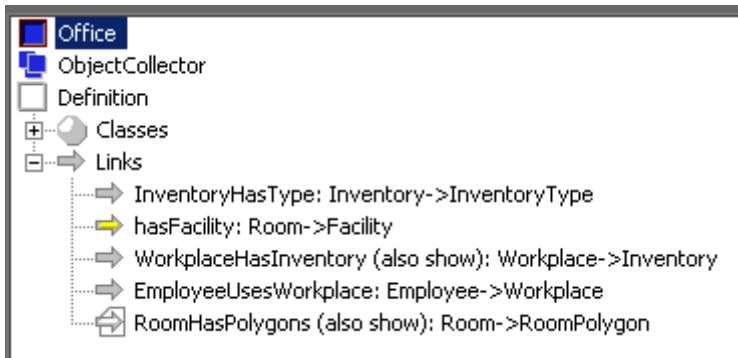
You may wonder what the white arrow and the option

“Consider 'Facility' when calculating the area of 'Room' (Subarea)”

means. In this example this option would only be available if “Room” inherited from “Area” and “Facility” from “Subarea”. If this option was enabled, the surface areas of


facility objects would then contribute to the surface areas of room objects. For more information refer to **→Configure... (links)**.

Please also create the still missing links “WorkplaceHasInventory” and “EmployeeUsesWorkplace”. As mentioned above, a workplace should display the graphic of the linked inventory when marked. Therefore, the option “Also show linked objects” is to be switched on. The link types are depicted in the structure view as follows.



The modeling is herewith completed. Your drawing should look similar to the file `..\TUTORIALS\FEATURE_DATA\GENERAL\TUTORIAL3\TUTORIAL3_MODELLING.DWG`. The structure definition shall now be made available for other plans. For this purpose, use the command **→Structure - Save structure definition...** and save the file as



`..\TUTORIALS\FEATURE_DATA\GENERAL\TUTORIAL3\OFFICE.TXT`

Now comes the work of sedulity. The objects of the sample plan need to be anchored in the structure. Thus we switch to the upper part of the structure view (“ Office“).



## 5.4 Add objects

### Rooms

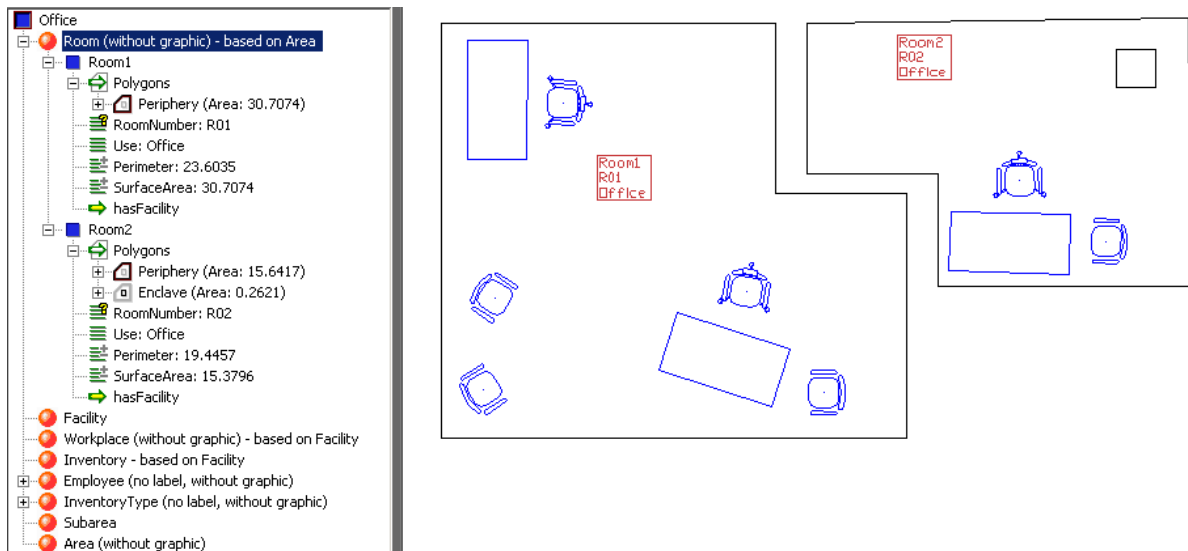
First we create the two rooms. You already know the general procedure from the previous tutorials.

Just a reminder: Open the context menu and select “Add room (Interior point)” in “ Office” → “ Room”. Disable the enclave recognition (Detect islands) with the command line option “Configuration”. Click into the room and the feature data management will create the respective subarea(s). Finally place the room label.


The room on the left is made up of one polygon, while the room on the right is made up of two polygons due to the square column. Since you have deactivated island detection, the software ignores the column at first.

To subsequently insert the island, please open the context menu “ Room 2” → “ polygons”. There select the entry “Add room polygon (inner point)”. As it is dealing with an island, and that the area must be subtracted from the total room area, is automatically regognised.

After displaying the attributes the following image is displayed:



## Inventory


Now create the inventory objects. Via " *Inventory*" → Add Inventory" you will be prompted the following

Select geometry for Inventory:

Select objects: *(select any inventory block and confirm with Enter)*

Then the software asks for an

Insertion point for label: *(Select a point on or close to the inventory)*

A new label and the corresponding entry appear in the structure view " *Inventory1*". This way you create all objects of the class inventory. Since pieces of inventory have an attribute "Number" with key properties (attribute type "Key"), this one is automatically counted up to ensure unambiguousness.

## Inventory types





Now we are creating three objects of the invisible class "InventoryType". These have – as configured before – no graphic and no label. Therefore the dialogue to change the first attribute "Description" will appear right away (due to the setting "Check attribute immediately after creation"). You will get to the next attribute "Price" with the [ $\gg$ ] button.

Name the three inventory types "OfficeChair", "GuestChair" and "Table". You can determine the prices as you like or leave them at the default setting "0". The three objects now serve as a kind of "catalogue" for the available types of inventory in the project.


## Workplaces

The three "Workplaces" are created the same way. Workplaces as well do not have a graphic but they have a label. Place the label of the workplaces close to the desk. No attributes are defined for the class "Workplace".



## Links InventoryHasType

Now we link the InventoryTypes with the pieces of inventory that exist in the plan. There are two ways to do this: Either you use the so called **forward relation** via  *Inventar1* →  *InventarHasType* or the **return relation** via  *InventoryType1* →  *InventoryHasType*.

In this case we recommend the return relation as the pieces of inventory have labels in the drawing and, thus, the return relation is easier to use.




Before we begin we replace the IDs (InventoryType1, InventoryType...) with the descriptions (OfficeChair, Table...) in the object header. To do so go to one of the attributes  *Description* and select "Show in header" from the context menu. Now the attribute "Description" serves as ID





Now select  "Chair" →  *InventoryHasType* → "Link with Inventory" via the context menu. Following message appears:

Select Inventory label:

Select objects: (Select the chairs)



The corresponding inventory objects can now either be selected in the structure (the object will then appear in bold print " ***Inventar1***") or you can click on the labels in the drawing. The latter is easier here since the graphic tells you which inventory object is of which type. Click onto the labels of the four guest chairs and confirm your selection. In order to check, have a look at the structure under  *Inventory1* →  *InventoryHasType*. The inventory should now be mentioned here.

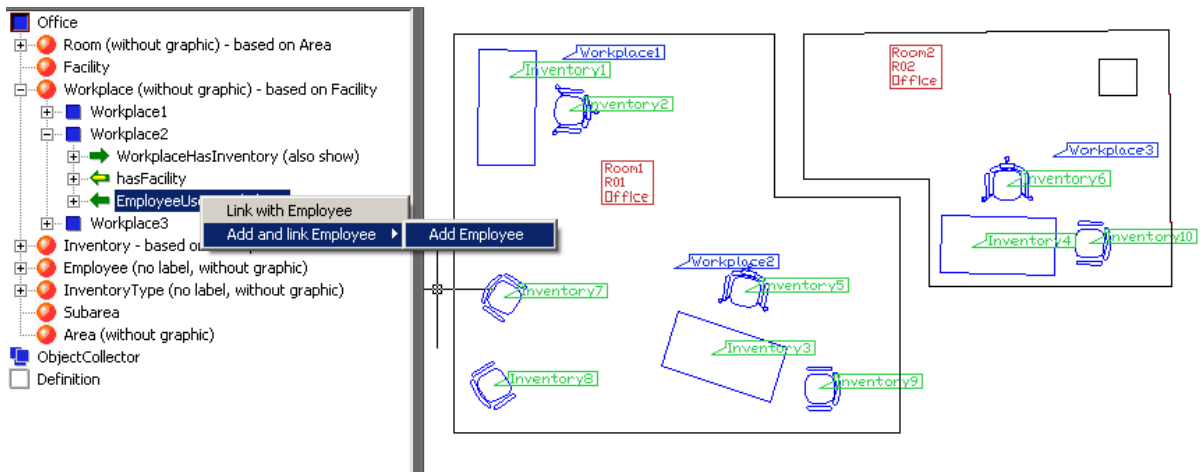
### Links WorkplaceHasInventory

The forward link via  *Workplace1* →  *WorkplaceHasInventory* seems most suitable here. The procedure is the same as the one described above. Click on the respective inventory labels for each workplace.

### Links EmployeeUsesWorkplace

Sometimes it is of advantage to create and link an object at the same time instead of creating it first and then linking it. This procedure shall be illustrated by means of the example EmployeeUsesWorkplace.

Go to  *Workplace 2* →  *EmployeeUsesWorkStation*. Assign an employee to workplace2, which you add to the structure simultaneously. Use the context menu entry "Add and link Employee → Add employee" as following figure illustrates.



The attribute “Name” is prompted if the respective option has been enabled.

**Important remark**

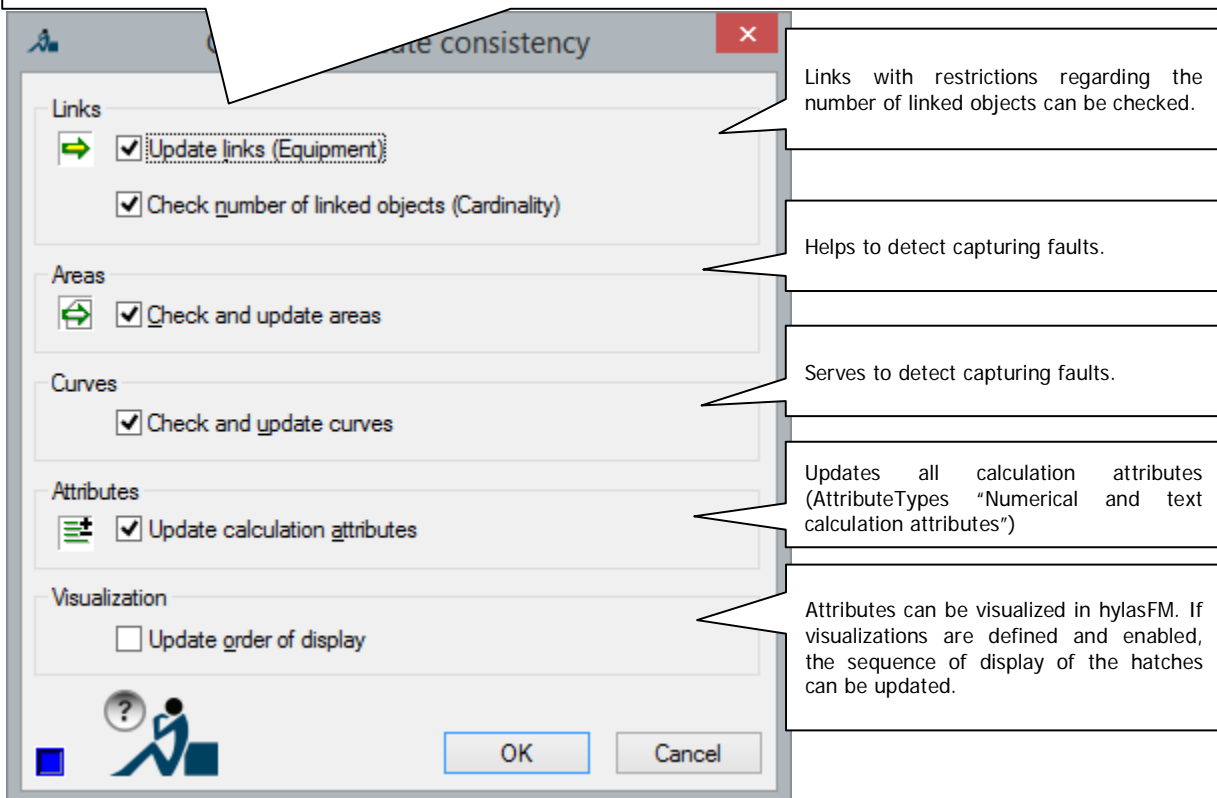
Should an employee “Mrs. Diligent” use several workplaces, you do not have to create a new employee “Mrs. Diligent” for the second workplace but only create the link. Select the entry “Link with employee“ in the context menu of the relation “EmployeeUsesWorkStation”. Since employees do not have labels – and can thus not be selected in the graphic – you have to select them in the structure view. This again is easier if you choose to display the attribute “Name” in the header (Employee1 → Name → Show in header). Thus you can look for “Mrs. Diligent” instead of “Employee1” in the structure. If you want to select an employee in the structure view, click on the entry in the tree structure with your left mouse button and then click with the right mouse button somewhere into the drawing area. The latter serves to take over the employee.

You may wonder when the links between the rooms and facility (base class of workplaces and inventories) are created. The next section shows how this task is done automated.

## 5.5 Create consistency

After many data have been captured, a consistency check should be run. To do so enable the entry →**Check/Create consistency...** in the context menu of "Office". In the following dialog one can determine which activities should be run.

- 1) If a link exists between objects that have been derived from "Area" (e.g. rooms) and other objects that have a label (e.g. inventory), these will be linked with each other if the insertion point of the label lies within the area (polygon).
- 2) If a link exists between two objects that have both been derived from the class Area, these will be linked with each other when the boundaries of both areas of the objects intersect.
- 3) If a link exists between two classes of which one has been derived from "Area" and the other one from "Curve", these will be linked with each other when the (out)lines of objects intersect.



A report will be created after the check. Unclear objects will be made available for editing as references in on or more object collectors.

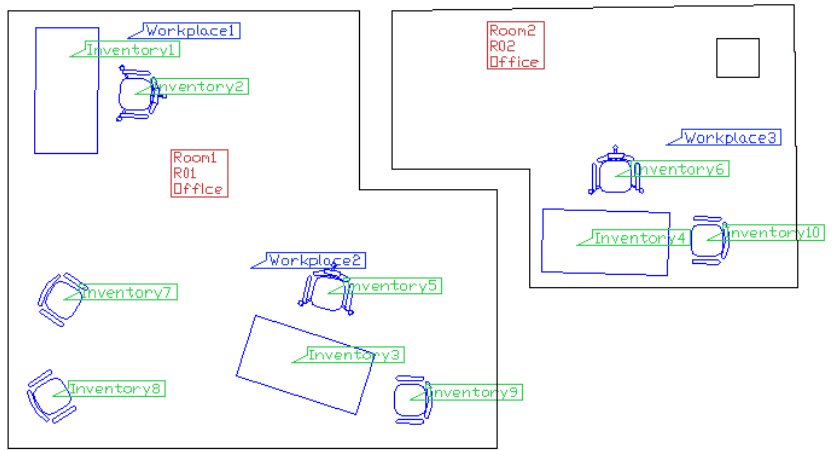
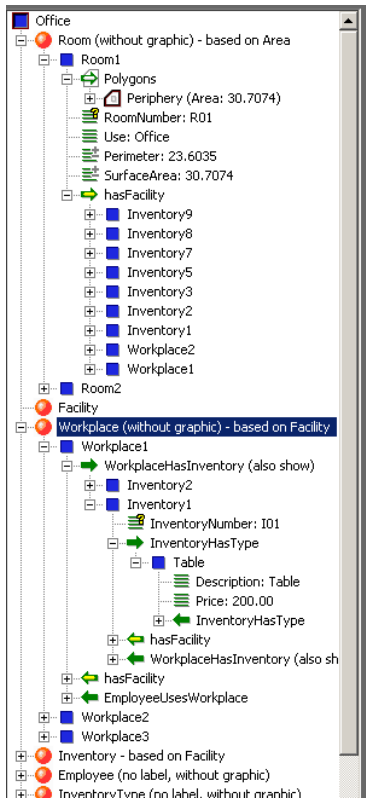
When this step has been completed, the desks and their pieces of inventory will be linked with the respective rooms, too.

**Remark:** The characteristics equipment and cardinality of the links can be adapted in the definition part of the links (e.g. Links→ hasFacility) under the context menu entry →**Configure...**

The resulting drawing will look as follows.

..\TUTORIALS\FEATURE\_DATA\GENERAL\TUTORIAL3\TUTORIAL3\_RESULT.DWG

# Modeling and configuration – Tutorial 3 Tutorial Facade Damage Mapping (Part 15-15)



## 5.6 Layout settings

Now you can make some modifications that give the plan a better and clearer layout:

You find setting options for the label in the definition in the context menu of the class “Configure...”: Font, font size, layer, appearance. By choosing your own layer for a label, you can give the layer its own color.

The screenshot shows the 'Class configuration' dialog box for the class 'Inventory'. The 'Label' tab is active, showing the following settings:

- Object has label
- Insert label on certain layer
- Layer for label: Inventory (dropdown), 47,208,74 (color swatch)
- Label size: 0.2 AutoCAD units
- Label compact
- Text alignment for Label: Bottom left (dropdown)
- Text style for label: Standard (dropdown)
- Prompt attributes immediately after object creation
- Adjust existing labels

The 'Layout' section contains three radio button options:


- Label with Frame
- Leader at text
- Only text

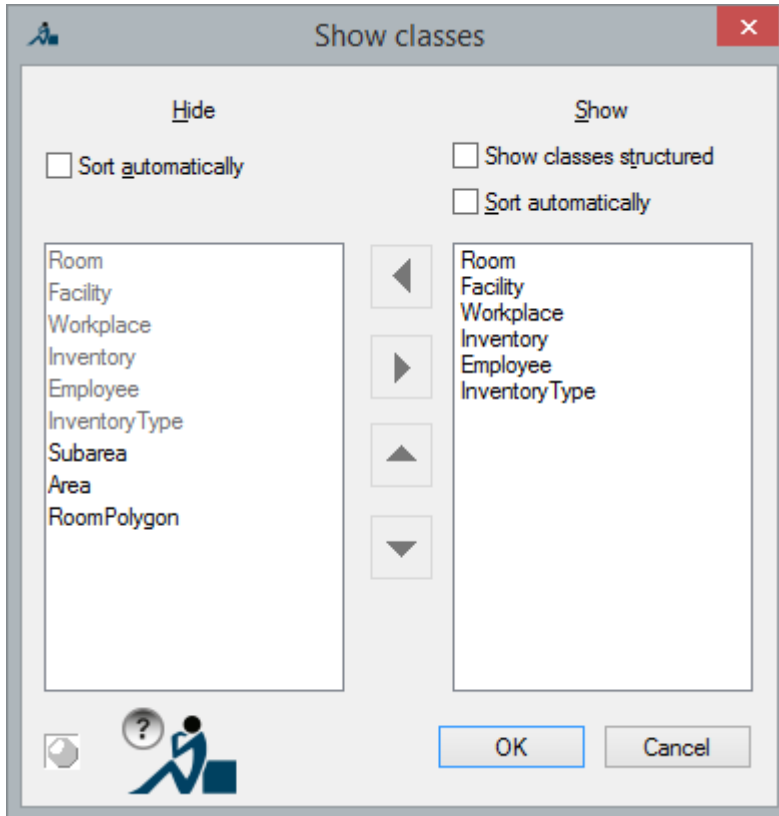
Callouts on the right side of the dialog box illustrate the visual results of these settings:

- Label with frame:** Shows the text 'Inventory2' and '102' enclosed in a rectangular box.
- Leader at text:::** Shows the text 'Inventory2' and '102' with a leader line pointing to the text.
- Only text:** Shows the text 'Inventory2' and '102' without any frame or leader.
- Label compact:** Shows the text 'Inventory2' and '102' stacked vertically within a rectangular box.















In order to give the structure a clearer layout for the data capturing, you can hide the classes that are not needed. “Facility” and “Area” only serve as utility classes for inheritance, but objects will not be captured there.

Go to the entry  Classes and select the context menu entry **→Sort classes, show/hide....** A dialogue with the two fields “Hide” and “Show” will appear.



You can move the classes from one list to the other with the arrow buttons. Mark the class “Area” and click on “<“. Do the same for “Facility”, "Roompolygon" and "Subarea". Confirm with OK and you will see that the classes will no longer be displayed in the upper part of the structure:

-  Office
-   Room (without graphic) - based on Area
-   Workplace (without graphic) - based on Facility
-   Employee (no label, without graphic)
-   Inventory - based on Facility
-   InventoryType (no label, without graphic)
-  ObjectCollector
- Definition

## 5.7 Concluding considerations

Modeling data structures is a challenging process. This section communicates some experience regarding this topic.

Following question should always be the beginning of all considerations.

Which inquiries need to be answered using the data structure today and in future?

Our realized draft, for example, allows following queries (for example in a data base via structured query language [SQL])

- What is the total value of the work place used by employee X?
- In which room can we find inventory of the type “Chair”?
- Is there at least one workplace in every room?

Following query cannot be answered

- How much area (in square meters) does workplace Y occupy?

Following abstract quality criteria of a structure definition can be named

- **The structure should be free of redundancy**

Every piece of information is contained in the structure only once. Thus, ambiguity can be avoided. Knowledge that can be derived from other information is not to be captured (e.g. the wall surface area of a room can be calculated from  $Room.Perimeter \times Room.Height - Opening.Area$ ).

- **Level of detail**

The draft should come as close as possible to the natural order of things. The classes, links and attributes receive meaningful names.

- **Simplicity**

The draft should be easy to understand and should not contain elements that appear constructed.

- **Flexibility**

Future additions (new classes, attributes and links) should be integrated into the current draft without the need to delete classes and links. Inheritance relations should be maintained.

Concrete clues are presented in the following paragraphs.

### **Inheritance or link?**

If a class B is to be understood as specialization of another class A, inheritance is used. I.e. all attributes in class A also exist in class B. Class B can possess additional attributes. “A” is called a base class of B (e.g. “tree” is the base class of “conifer”). The litmus test should always be the examination of the following statement:

- A “B” is a special “A”. (“A conifer is a special tree.”)

A counter-example: “A car is *not* an engine”. Therefore, a class “Car” should not inherit from “Engine”, even though the possible transfer of attributes (Power, Fuel consumption) may suggest it. Instead, there should be a link “CarHasEngine”.

### **Attribute or class?**

The question whether an issue S should be modeled as attribute or class is often not easy to answer. If one of the following questions is answered with yes, modeling the issue as class seems preferable:

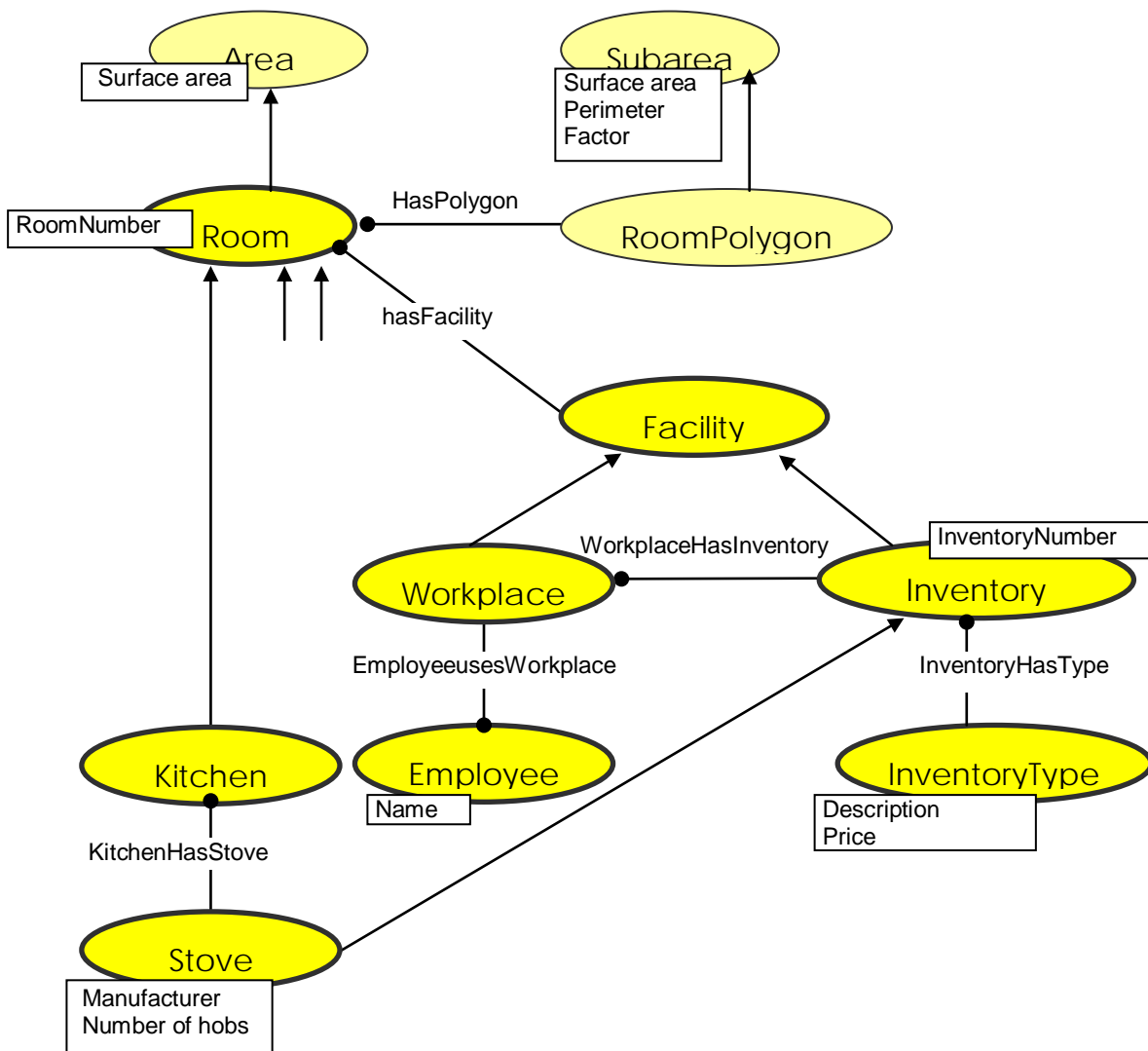
- Does a class A (sometimes) have several entities of S?
- Does S have own attributes?
- Are there specializations of S (means S is a potential base class of another class)?

- Are there links between S and other classes?

The following (incomplete) draft shows that modeling is by all means a creative process and that you have to balance between the quality criteria **level of detail** and **simplicity**. This draft is not intended to be realized in this tutorial. It is only food for thought and should illustrate the long-term importance of the modeling phase.

If one considers the question “Are there specializations of an issue?”, one can come to the conclusion that in this tutorial the class “Room” with its attribute “Use” has turned out a bit too simple.

For every possible use of the room special links and attributes could unfold. Following figure depicts this at the example of the usage “kitchen”. Similar effects are also possible for the usage “corridor” and “office”...



It is beneficial to put a structure definition draft to the test. Take a representative section of the plan and create the structure definition according to your draft. Import the structure into the target application (e.g. a database, CAFM system) and go through the query scenarios (SQL queries, reports). Improve your draft by and by. Only migrate all plans into the structure when the complete data flow for the plan section works properly.

### Direction of a link

The direction of a link can be set into any direction when modeling. It often only describes how to read the links name. Only if a class is linked with itself (e.g. “Polygon” encloses “Polygon”), we have to consider the direction when relating concrete objects (of the class “Polygon”).

The commands of the feature data clipboard **→Copy** and **→Cut** have an option that allows the implicit copying of objects of the forward link. Should, for example, the associated “InventoryType” always be created in the new file, too, when moving “Inventory” to another file, this option can be used. A “Workplace” associated with “Inventory”, however, would remain in the original file since it is related via the return link. When moving a workplace, its “Inventory” and “InventoryTypes” will be moved too.

Should you plan to move or copy objects often, you should consider the direction of a link under this aspect.

## 6 Tutorial Facade Damage Mapping (Part 1)

This tutorial and its structure definition is primarily based on a suggestion for an area- or stone-oriented damage mapping for approximate determination of measures for preservation.

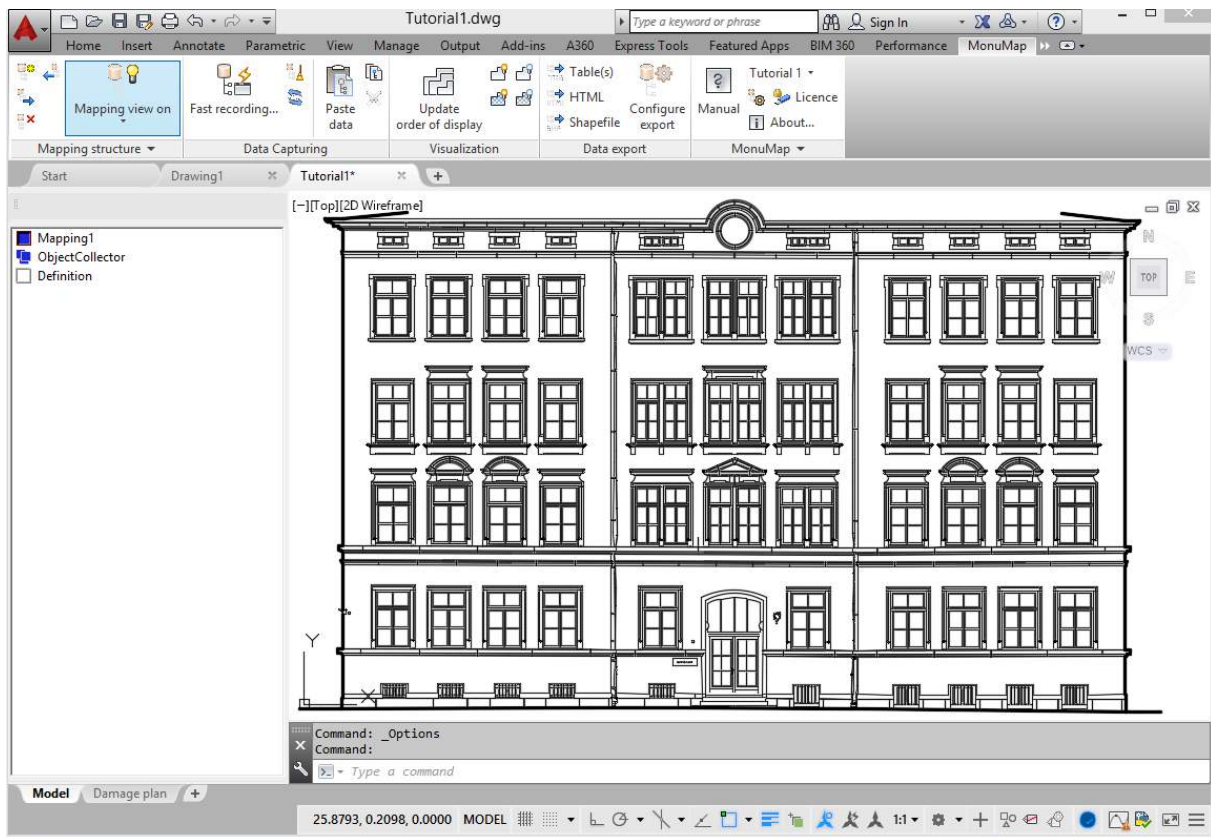
### 6.1 Preparation

In order to be able to work on this tutorial, you need a fully installed version of AutoCAD or AutoCAD LT into which As-Built for AutoCAD has been embedded.


After having started AutoCAD, you find the "Tutorials Facade Mapping → Tutorial1" entry within the As-Built Feature Data ribbon tab under Help. Two files, TUTORIAL1.DWG and TUTORIAL1\_RESULT.DWG, are copied into the directory

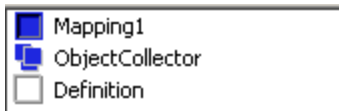
```
\USERS\[NAME]\DOCUMENTS\FARO\AS-BUILT          XXX          EN
ACXXX\TUTORIALS\FEATURE_DATA\HISTORICAL_FACADE_MAPPING\TUT
ORIAL1
```




when you select this entry. The drawing TUTORIAL1.DWG is automatically opened within AutoCAD.



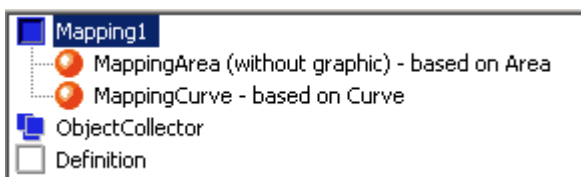
## 6.2 The structure view

To the right within the drawing area you see the façade view of a school. Left of the drawing area you see the structure view. The structure view consists of three parts. You gain access to the **mapping data** when double clicking the symbol  "Mapping1" at the top. The complete mapping and data capture is done via this part of the structure view.



Underneath you find the  "ObjectCollector", it may be used to change several object data at the same time. That is useful, when for example a large number of descriptions or attributes are to be changed in the same way. Also, when the software detects errors, those rejected data will be gathered in an object collector. The bottom symbol  "Definition" grants access to the change of the mapping structure: that means which data have to be captured and how they are linked. After double clicking  "Mapping1" the structure view opens. The following entries appear:

- "MappingArea" and
- "MappingCurve".



In mapping, two cases are of particular interest. At first that are the data, which refer to laminar structures and second that are data which refer to linear structures like cracks. In the present case, the "MappingArea" combines all data which have a laminar structure. These may be linked to closed AutoCAD polylines for example. "MappingCurve" combines all data that have a linearstructure. These may be linked to AutoCAD curves. By their appendix "based on Area" or "based on Curve" you can recognize the recording of data according to their geometrical structure.

### Excursus: Mapping – Elements

In our software "MappingArea" and "MappingCurve" are called **classes**. These two classes have characteristics of predefined special classes called "Area" and "Curve". The assignment of the characteristics is expressed by "based on Area" and "based on Curve".

#### *Characteristics of "Area":*


supports the link to an AutoCAD polyline  
 automatic calculation of the surface area  
 automatic calculation of the perimeter

#### *Characteristics of "Curve":*

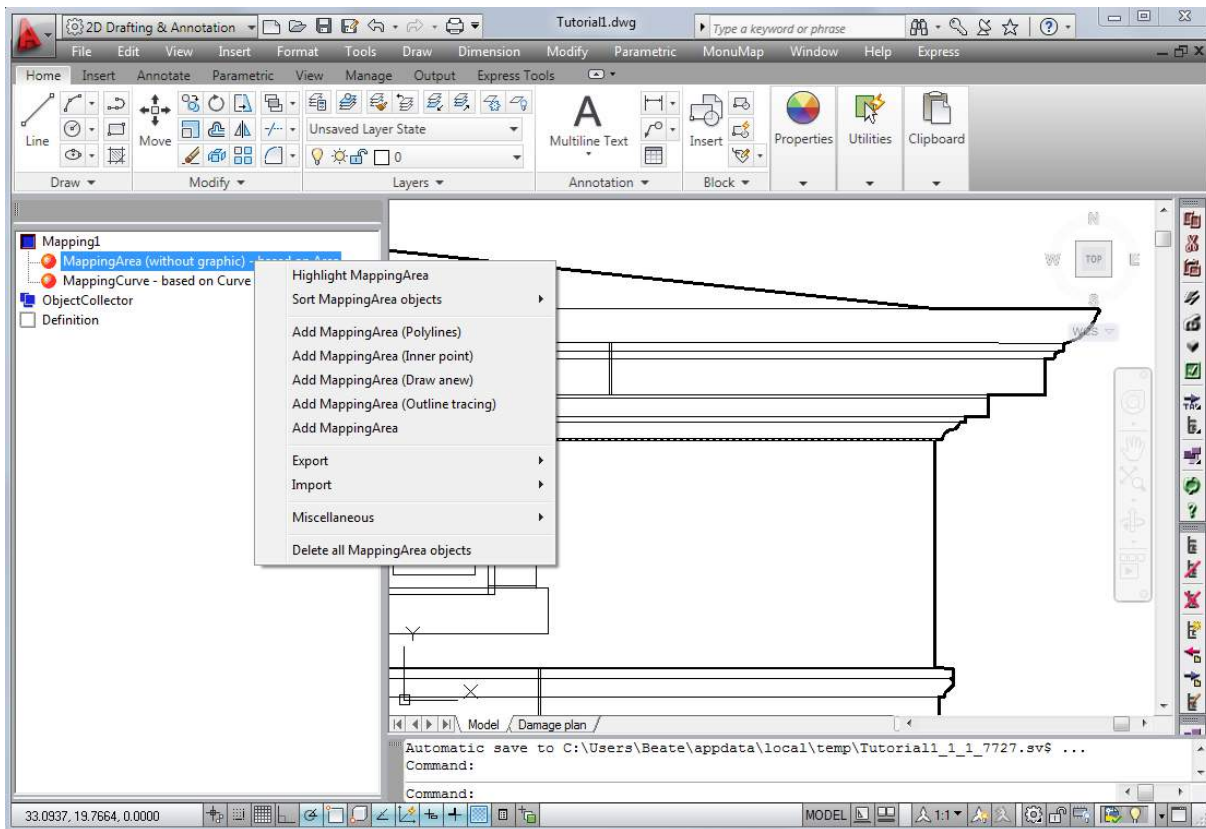
supports the link to an AutoCAD line element  
 automatic calculation of the length

## 6.3 Mapping of area data

In the present case it has been assumed that three different kinds of materials have been used for the façade (stone, plaster, brick). The damages are only estimated according to their damage intensity (little damage, medium damage, heavily damaged and destroyed). Also the measures are only differentiated by their rough form of action (preservation, repair, replace).

Let us assume that the plastering of an area in the upper right corner of the building is peeling off cavernous. This damage shall be repaired by the injection of binder. Zoom in to the top right corner of the building. As this concerns an area damage, the option  "*MappingArea (without graphic)*" is applied. By clicking the right mouse button on the entry the context menu will open. It contains several options to create a "MappingArea".

- "*(Polyline)*" assumes that a polyline is already existing.
- "*(Inner Point)*" determines an area bounded by several lines
- "*(Draw anew)*" activates the polyline command, the boundary will be created at that moment.
- "*(Outline tracing)*" adopts different existing drawing elements.



Since the area to be mapped is an isolated area, it is advisable to use the option **➔Add MappingArea (Draw anew)**. You select it by clicking the left mouse button. The command line will read as follows:

Command: HYLASCONTEXT

Specify first point or [Arc] <Arc>: *(click first point or change mode)*

Specify next point or [Arc/Undo] <Arc>: *(click second point or change mode or delete last point)*

Specify next point or [Finish/Undo/Arc] <Finish>: *(click third point or change mode or delete last point or finish)*

... *(click further points ... finish or ESC)*

Insertion point for label: *(click a point e.g. inside the boundary)*

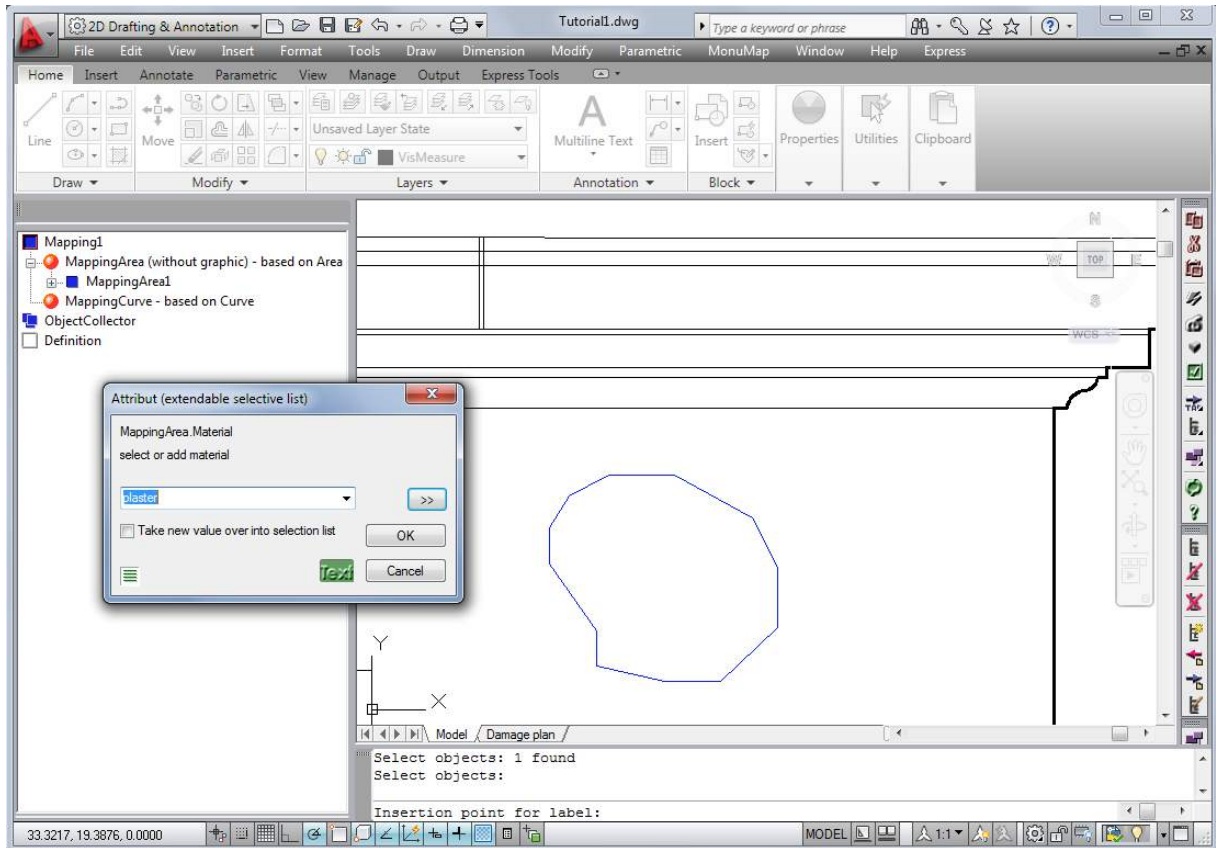
A polyline is to be drawn, which defines the damage area. After finishing the line, the software prompts for the insertion point for the label of the mapped area. Now click on the position within the damage boundary, where you want to place the label.

### Excursus: Labels

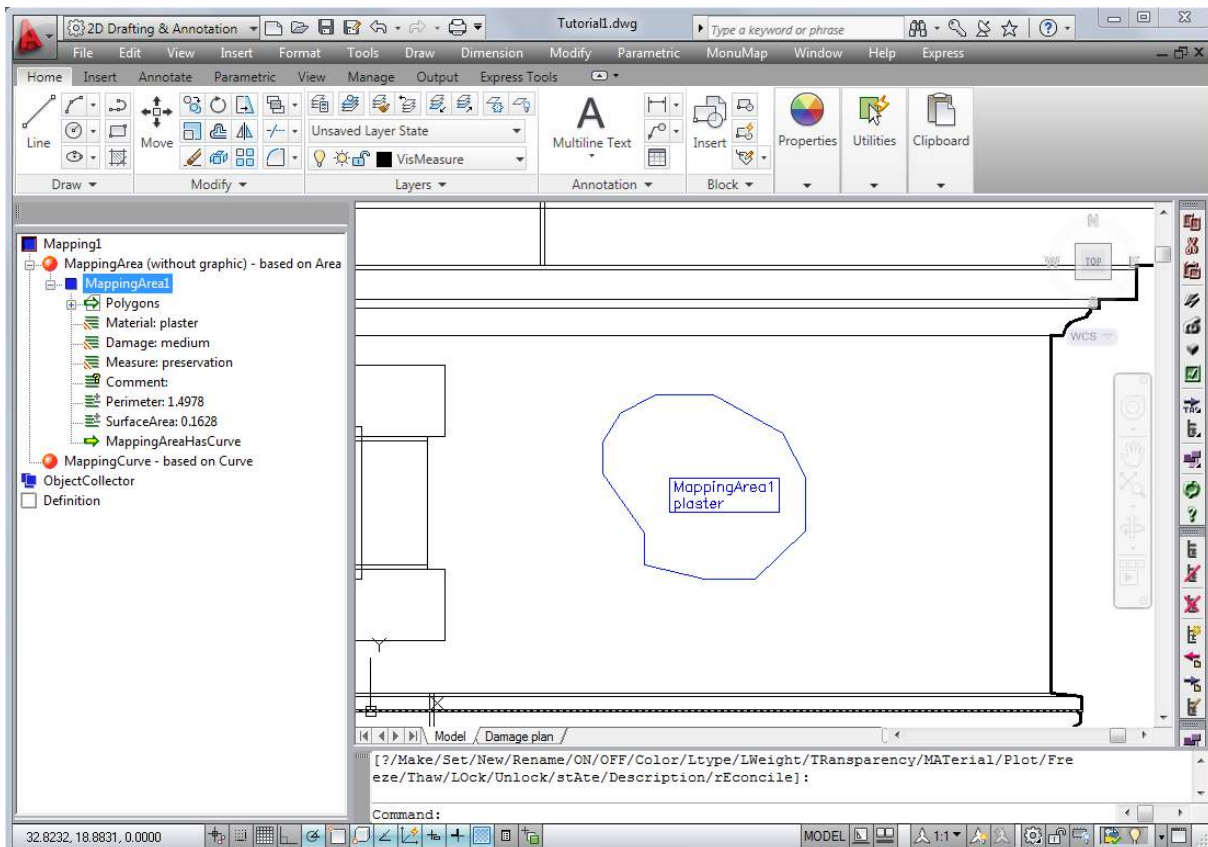
A **label** is assigned to every mapping/feature data element. If you click on the label, the structure view will open the according dataset. That way you have quick access to the data. The name of the class and the sequential, automatically assigned object number are standard contents of the label. You may also show other mapping/feature contents within the label. In our example this is the material. The labels are placed on an extra layer, which may be switched on or off as needed.



A window opens, where you are prompted to enter your data successively. At first the material has to be entered. You open the selection list by clicking on the arrow symbol on the right of the selection box. In case you are using a wheel mouse, you may scroll through the list by using the wheel. Select "plaster".



By clicking the double arrow button (>>) right of the selection box, you get to the prompt for the degree of damage. The default setting is "no" (for no damage). Analog the first step select the degree "medium". Next to the double arrow that points right >> (next) a second double arrow to the left << (back) now appears. For corrections you may click that button to get back to the previous selection. By clicking >> you now get to the selection for the repair task. The default setting is "no measure". Select the measure "preservation" by opening the list or scrolling through it. When you click on the >> button again, you can enter a text comment if you like. This is the last input option, which is indicated by the fact that the double arrow pointing right is not shown within this dialog any more. Confirm your entries by clicking **OK**.

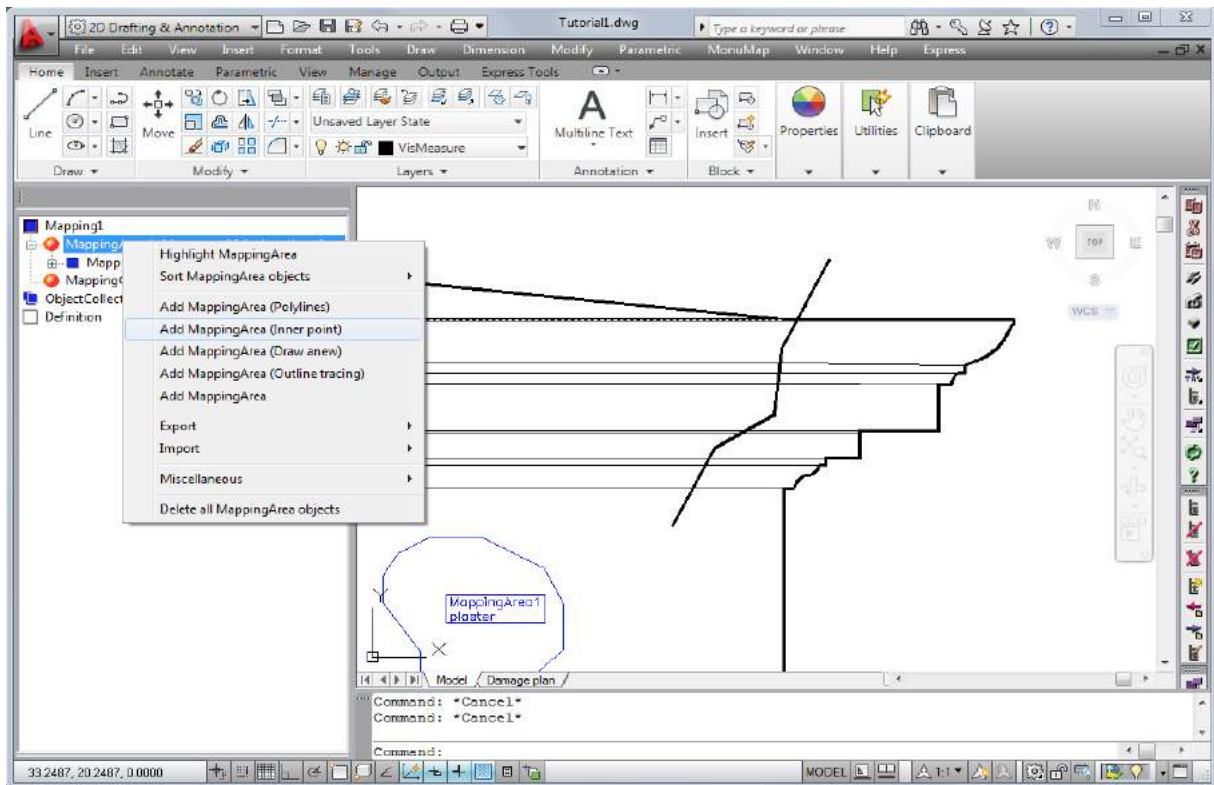


The boundary that you have drawn previously on the current layer has now been transferred to the predefined layer "AreaBoundary". The Label with the text "MappingArea1" and the material description has been inserted at the selected position and placed on the layer "LabelMappingArea". The name of the layer can be configured within the definition section  "Definition", but this is subject of the Tutorial 2.

#### Excursus: Structure view

After the first mapping a square with a "+" has been generated within the structure view in front of the entry "MappingArea". This indicates that data have now been stored for mapping areas. When you double-click on "MappingArea" or a single click on the "+" symbol a structure tree opens, which shows the particular mapped areas. By clicking on the single "+" symbols shown there you can display the properties of the individual damage areas. The values may be changed by double-clicking on "material" for example.

With the next step you will generate an additional "mapping area". Let us assume that in the top right corner there is a damage on the entablature. You may now use AutoCAD drawing elements for the definition of the damage. At first draw a diagonal polyline in order to delimit the damage area to the left. The line has to intersect the upper and the lower edge of the entablature. The overlap is not important, so just be generous. Zoom the mapping area in closely. But make sure that the complete area to be mapped is visible. After clicking the right mouse button on "MappingArea" select **➔ Add MappingArea (Inner point)**.



Now you are asked to click a point inside the area to be mapped:

Click inner point of a polygon for MappingArea or

[Configure]:

*(inner point of the area)*

You do so by clicking the left mouse button within the zone of the lower profile element. The software now looks for the boundary of the area. This boundary is composed of the temporary line and the profile lines or the edge of the building within the drawing. The identified area is hatched, please verify the area. Extend the selection by clicking the two profile areas above.

Click inner point of a further polygon for MappingArea or

[Undo/Analysis/Configure]:

*(point inside first top profile element)*

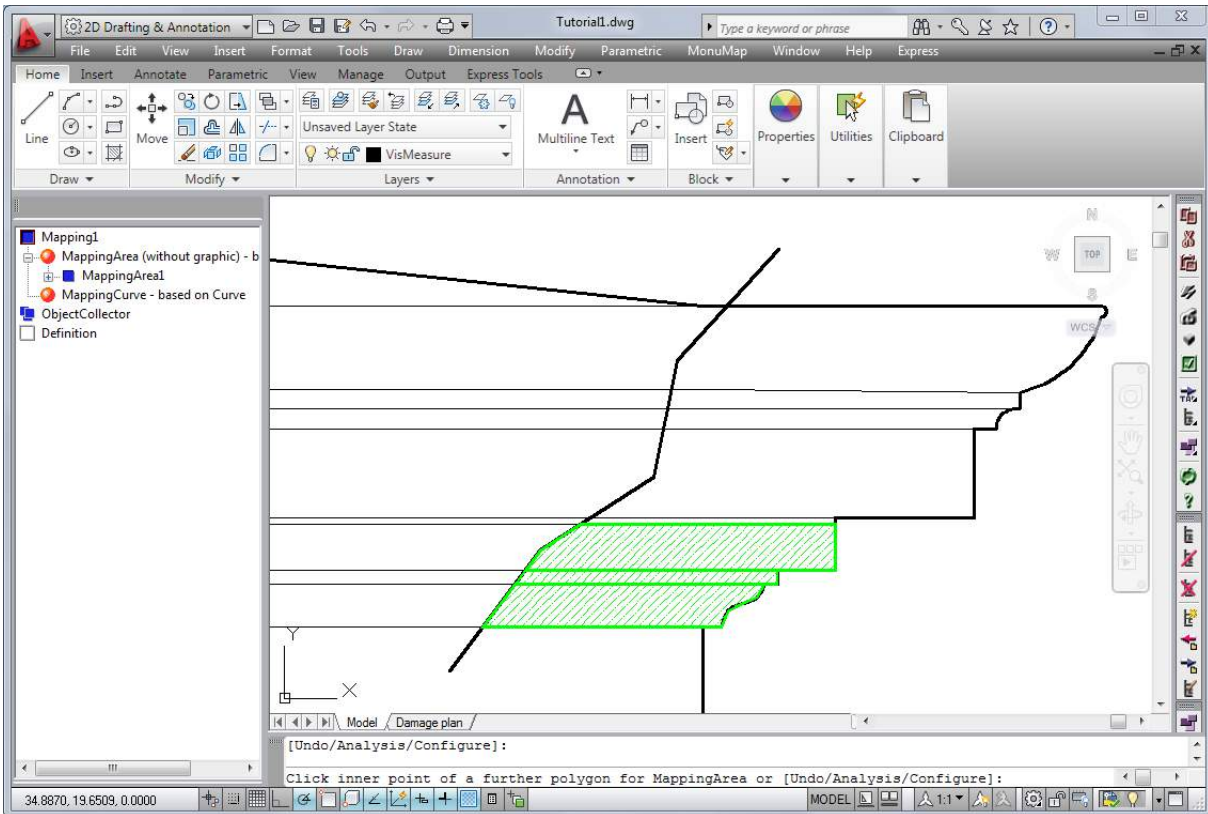
Click inner point of a further polygon for MappingArea or

[Undo/Analysis/Configure]:

*(point inside second top profile element)*

If your are satisfied, confirm your selection by pressing the <Return> key or <Space>

By pressing the "U" key for undo you discard the complete selection and go back to the previous state.

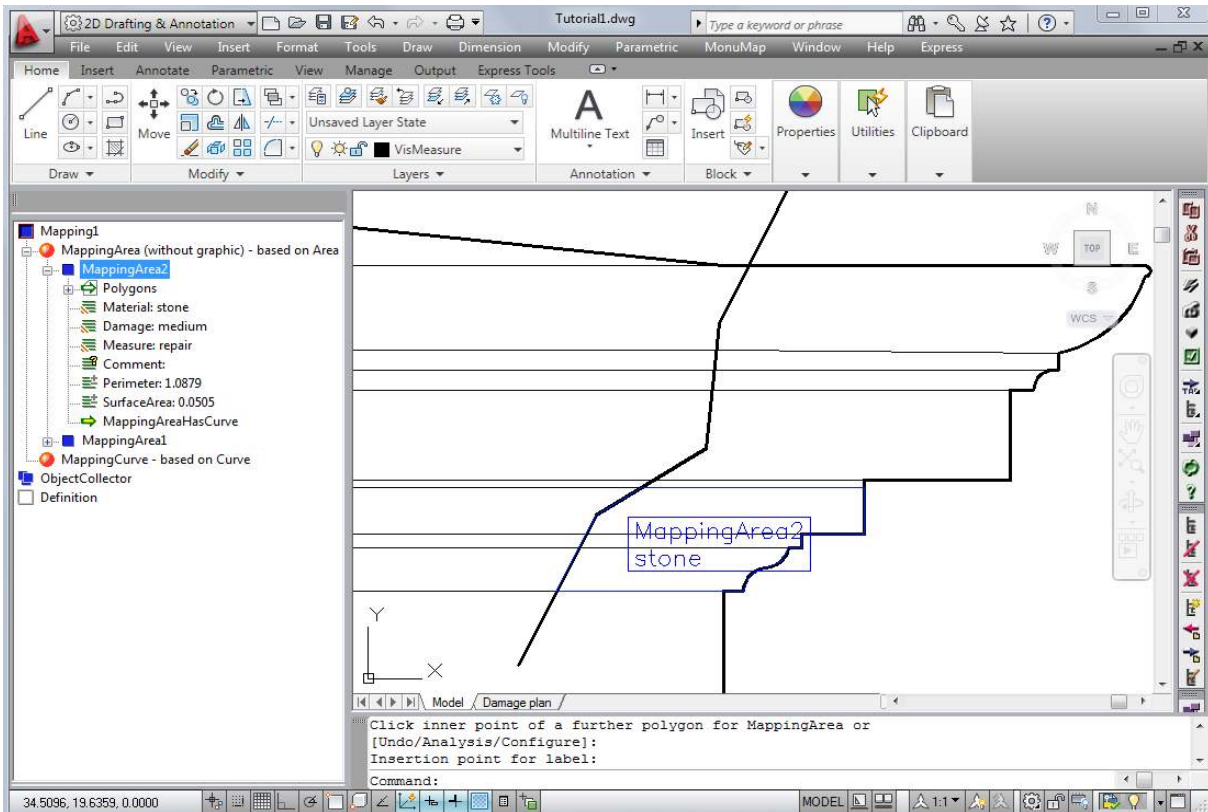


Finally you are prompted for the insertion position of the labels again:

Insertion point for label:

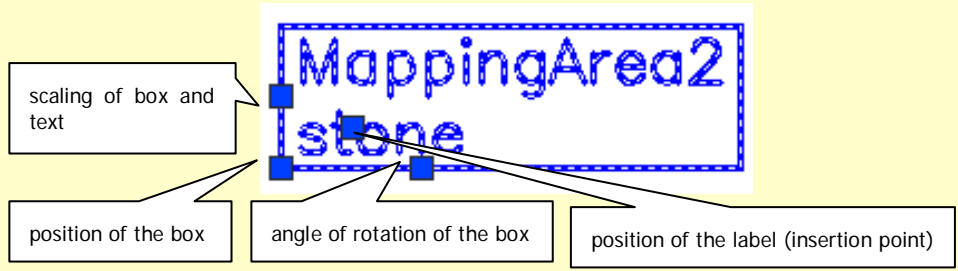
*(position of point label)*

Insert the label at an appropriate position and select "stone" for material, "medium" for the degree of damage and "repair" for the measure. Confirm your selections by pressing **OK**.





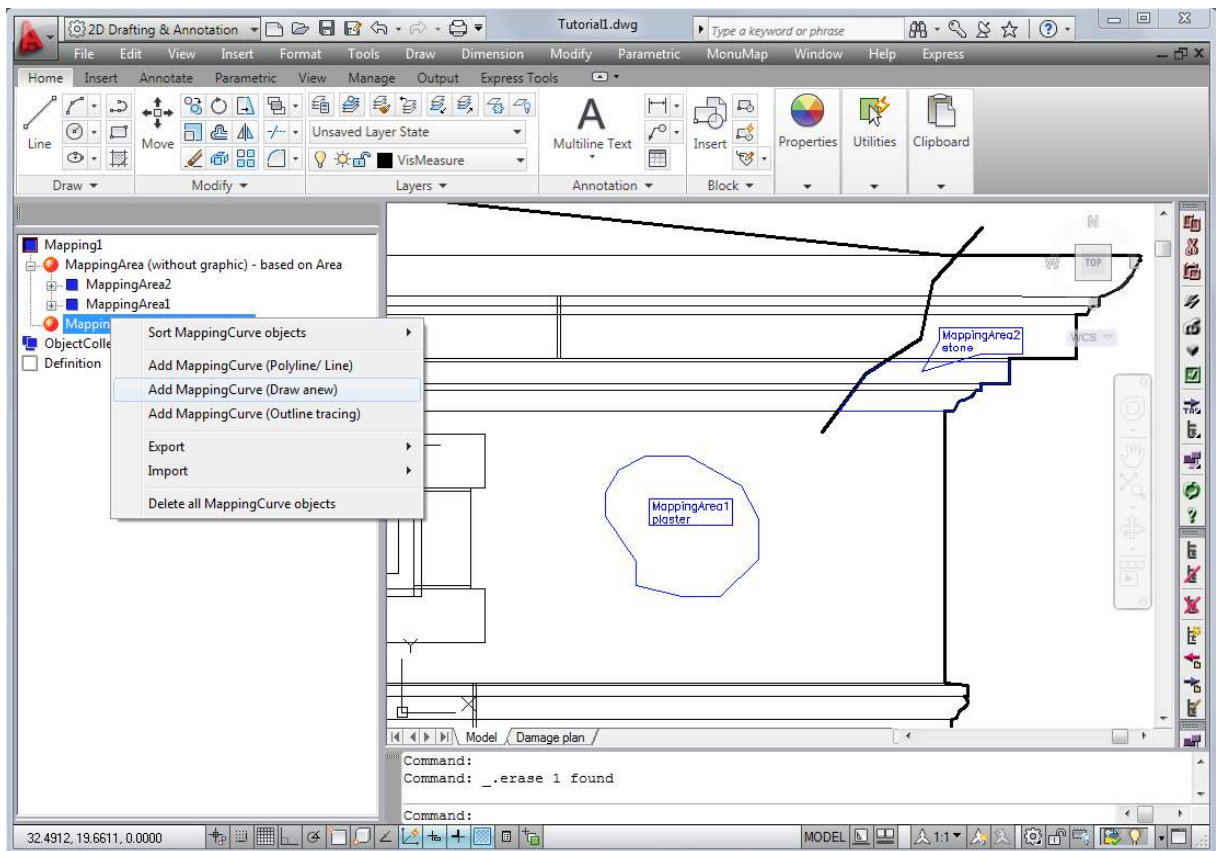
Excursus: Label 2

When you select a label, four grips will appear. With the grip inside you can move the whole label. This is the insertion point of the label as well. The lower left grip is for moving the box. The reference to the insertion point is kept. Thus you can move the label to a place where it does not disturb. With the other nodes you may rotate or scale the label.



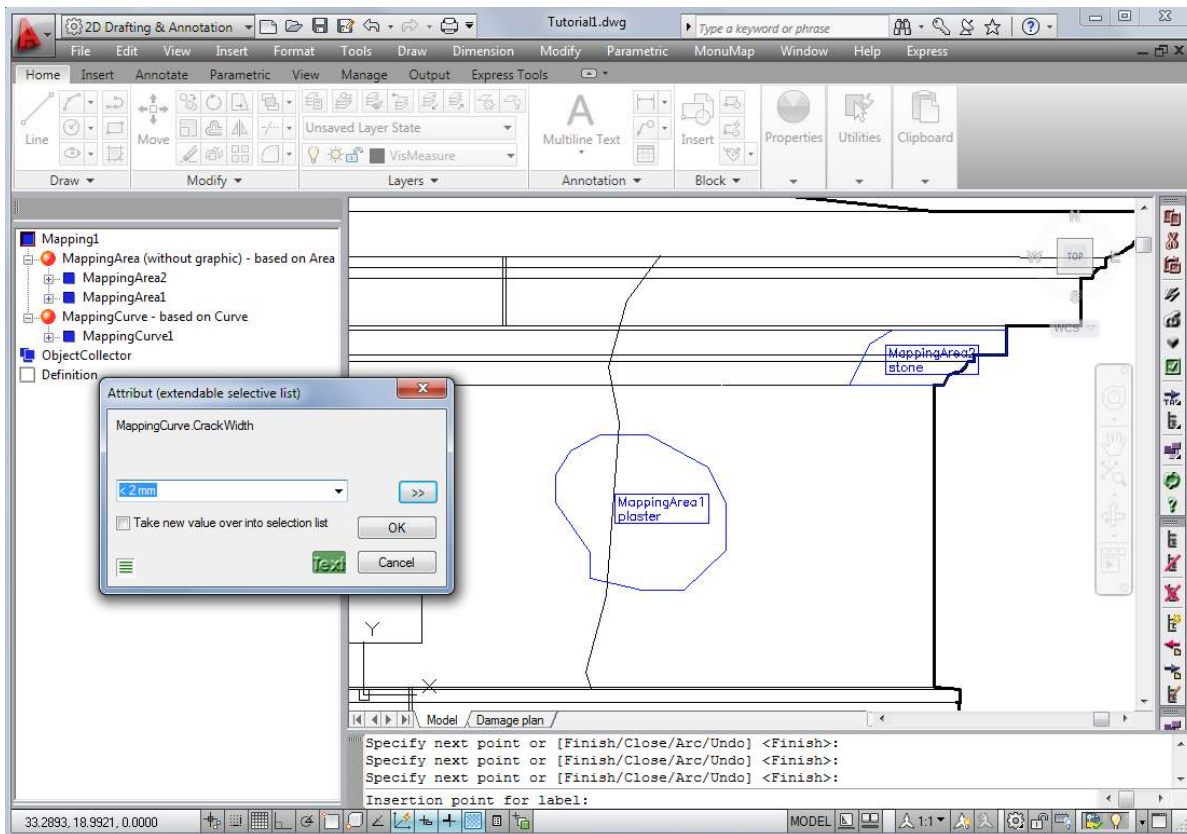
### 6.4 Mapping of curve data


Now a crack in the façade is to be mapped. This line-like element is recorded via the entry  "MappingCurve". By clicking the right mouse button on  "MappingCurve" the context menu is opened.




In our case a line is to be drawn, therefore you have to select the second option: **→Add MappingCurve (Draw anew)**. You are now prompted to draw a polyline, this works analog to the recording of a "MappingArea". Make sure, that the polyline intersects MappingArea1 (the reason will be explained in the next chapter). After

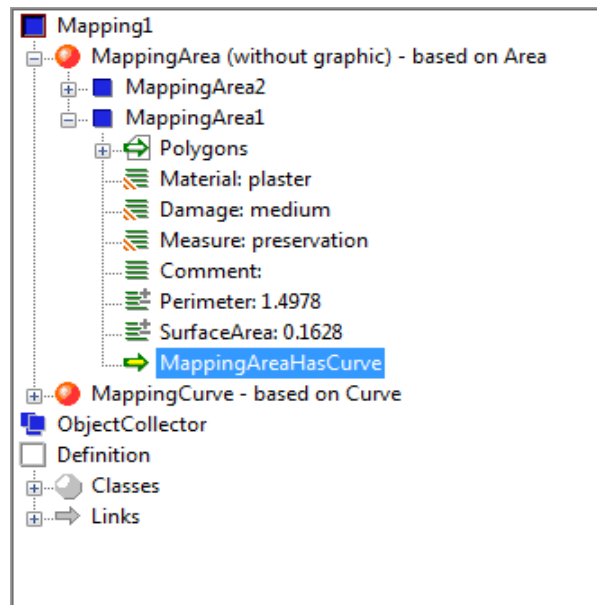
finishing the line, a selection window appears and you are prompted to characterize the crack. Select for example: crack width " $\geq 2$  mm" and measure "injection".



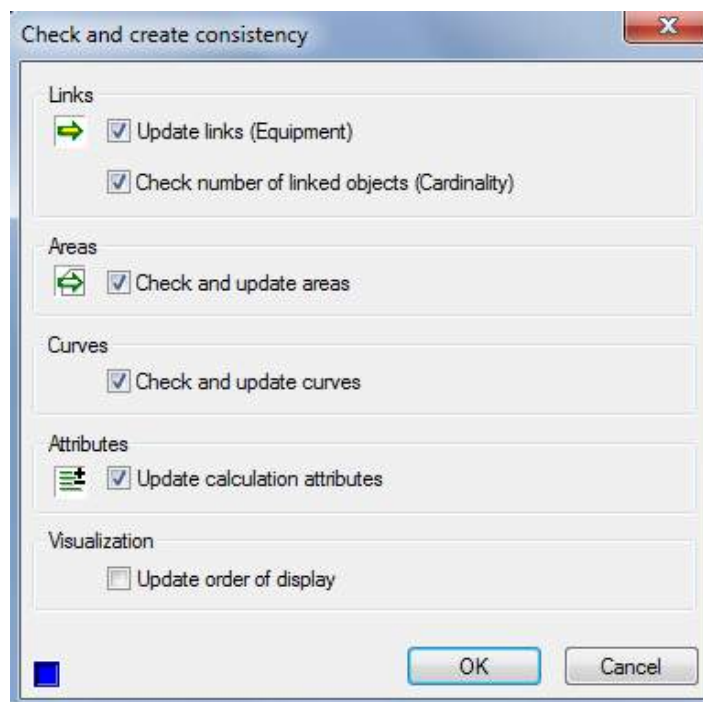
In this structure view there will now be a square with a "+" in front of "MappingCurve" as well. This symbol indicates contents for this category. By clicking the label you get to the correct position within the structure view quickly. You may repeat the last feature data command and create another crack object by clicking the command icon . For the fast and efficient creation of mapping objects, the so-called **→Fast capture** is very helpful (see command reference).

## 6.5 Creating links

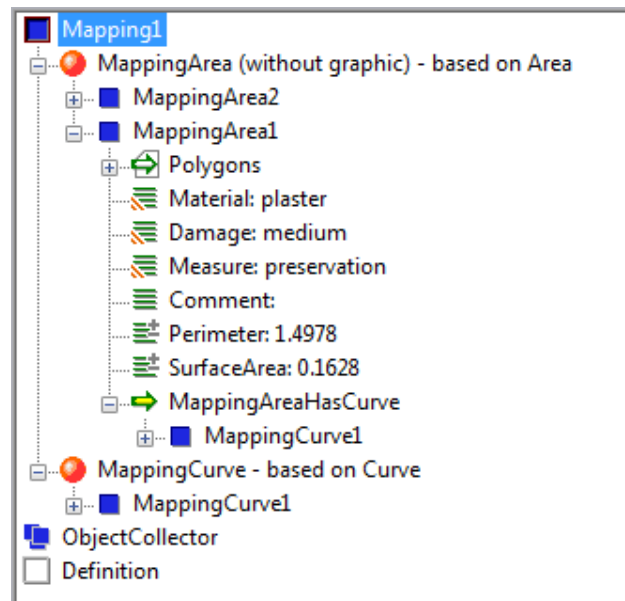
By looking at the drawing it is obvious, that a crack runs through MappingArea1. You will not find any information regarding this fact within the recorded data sets though. If you open the node MappingArea1 you will find the entry  "MappingAreaHasCurve".



This is called a link, its name indicates its meaning. A mapping area shall point to all mapping curves that cross this mapping area. Such an assignment can be carried out automatically or may be done manually. The former will be explained here. Move to **Mapping1**. By clicking the right mouse button you get to the command **Check/Create consistency** within the context menu. The following window opens.




Here different options can be chosen. According to the amount of data the consistency check may take several seconds. In this case especially the checkmark in front of "Update links (Equipment)" is important. After the consistency check has been executed, the links are updated. This means for our case that the *linked MappingCurve1* can now be found under the entry **MappingAreaHasCurve** of *MappingArea1*.

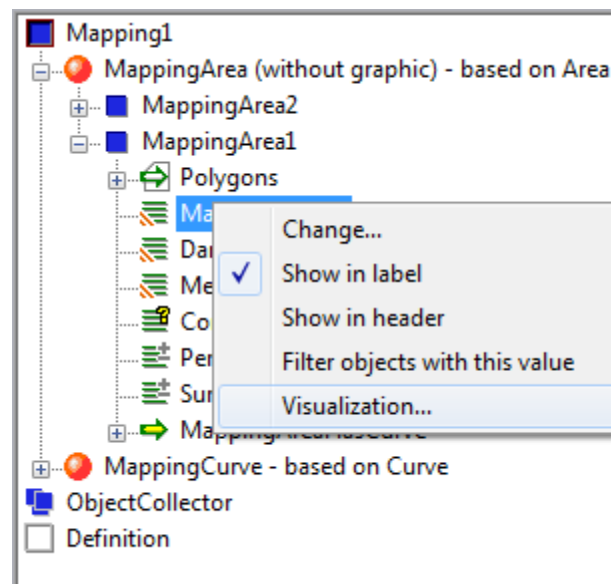


The structure view may be folded completely by clicking the  symbol.

## 6.6 Application of visualizations

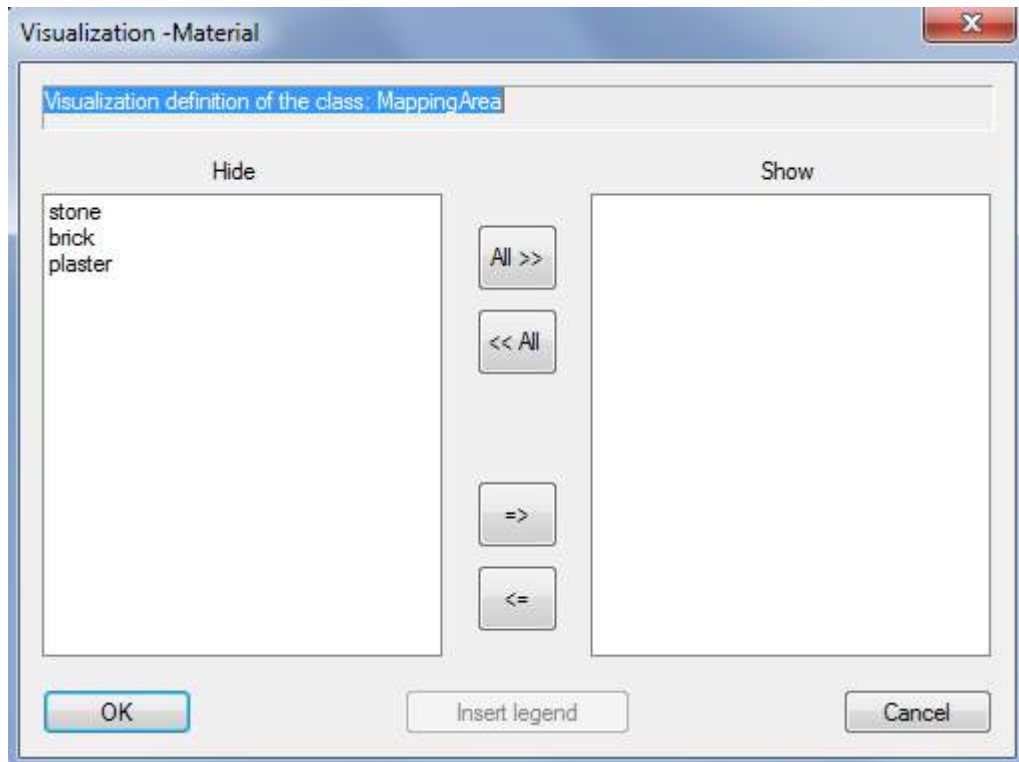
The mapped data are not only to be stored within the structure view, but are also to be displayed as coloured mapping plan.

In order to do so open the data set of any mapping area, e.g. by clicking the label of MappingArea1. The symbol next to "Material: plaster" has two diagonal grey stripes in the bottom left corner . That indicates that a visualization has been defined for the attribute "Material". By clicking the right mouse button on "Material: plaster" the context menu opens and at the very bottom there is the entry **→Visualization....**

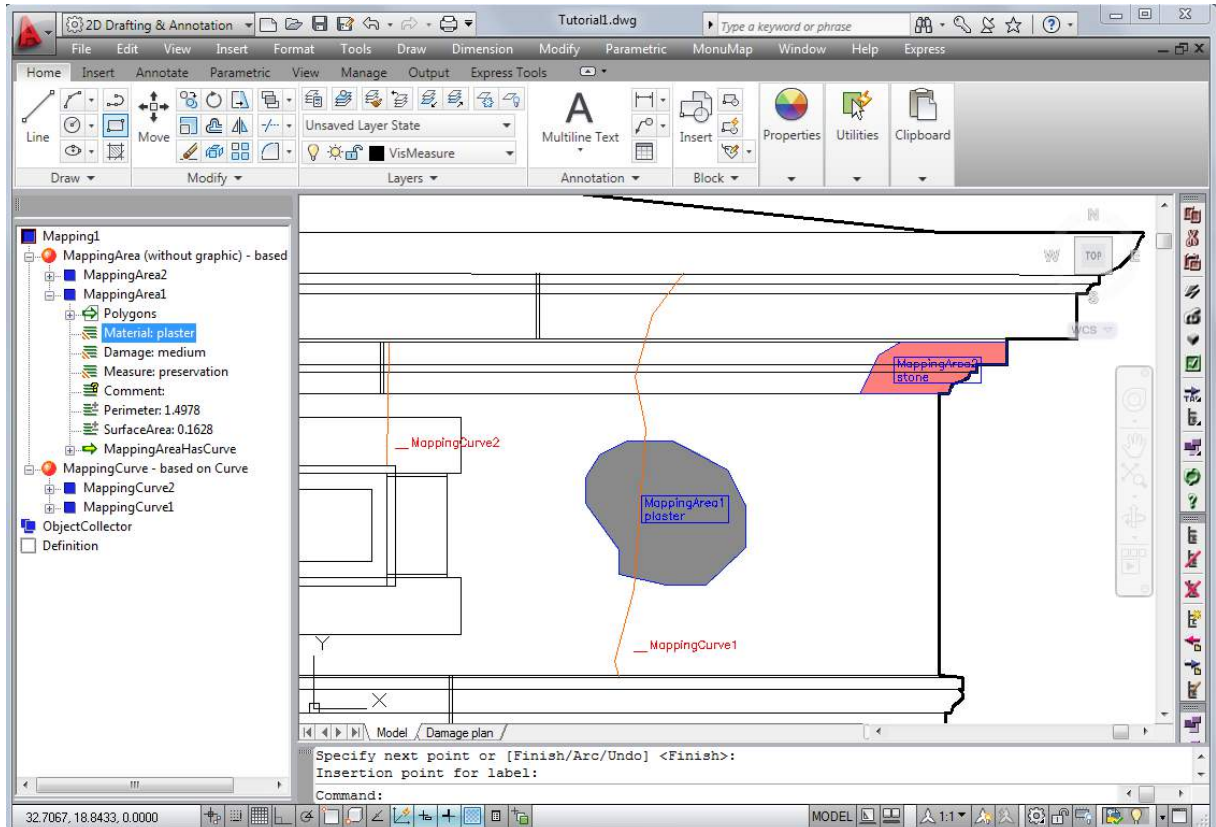


If you select this menu option, the window "Visualization – Material" will open.






On the left hand side the hidden materials are listed. If you click the button **All >>**, all materials are moved to the field "Show". Thus the visualization of the materials is activated. In this case the different materials are visualizes by a coloured solid hatching. Close the window by clicking **OK**.



The visualization areas for the materials are displayed on the layer "VisMaterial". The other mapping themes are displayed accordingly on the layers "VisDamage" and

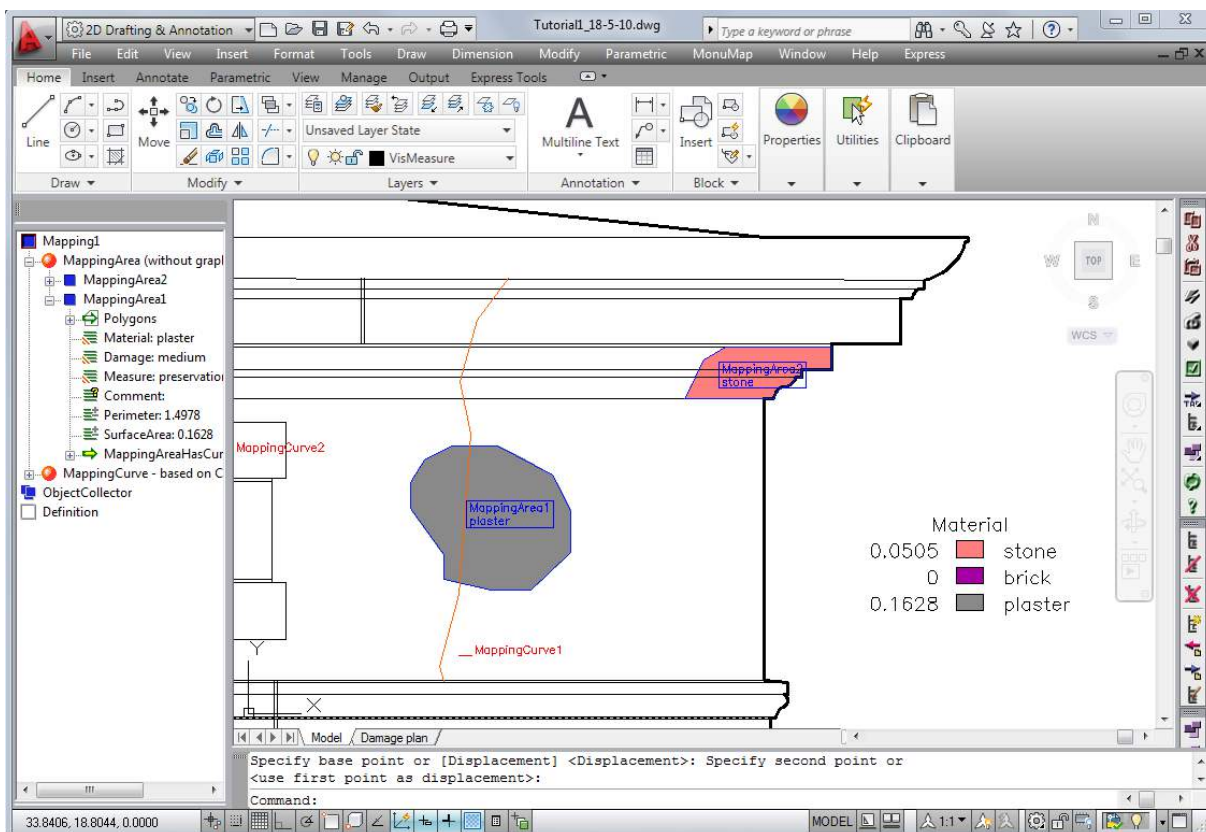
"VisMeasure". Cracks are displayed on the layer "VisDamage" as well. Visualizations may be controlled via the corresponding layers. Due to the time of regeneration it is advisable for large amounts of data though to control the visualizations using the mapping tools.

Keep the visualization of the "Material" activated and create a new mapping area now. The software will arrange for a good display order within the screen area itself, so that for example solid hatchings are placed behind line elements. In this connection the order of AutoCAD elements lying outside of the screen area may change. In that case you may call up the command → Visualization – Update order of display . Thereto the complete drawing should be visible on the screen.

## Inserting the legend

You find the button "Insert legend" within the window where you can show or hide visualizations. Repeat the steps described above until you get to the dialogue "Visualization – Material" by running the command → **Visualization** from the context menu of the attribute "Material" again.

If the visualization is switched on (the right part of the window is not empty) press the button Insert legend and set the insertion point. After you have closed the window the legend will be inserted at the wanted position. Additionally the added up areas are inserted left of the colored areas. Please note, that the legend and numerical values are not updated automatically when changes are done. Therefore it is advisable not to generate the final legend until right before the printout. You may repeat this process as often as you like, inserting the legend will neither have an impact on the drawing nor on the data sets.



## 7 Tutorial Facade Damage Mapping (Part 2)

In tutorial 2 you will learn how to customize the structure definition to your own requirements. You will learn to create new attributes as well as to assign new values to existing attributes. Visualizations will be customized.

This tutorial requires the basic mapping knowledge, as was put across in tutorial 1.

### 7.1 Preparation

In order to be able to work on this tutorial you need a completely installed version of AutoCAD or AutoCAD LT, into which As-Built for AutoCAD has been embedded.

In the As-Built Feature Data ribbon tab under Help under the entry "Tutorials Facade Mapping →Tutorial2". If you select this item, two files TUTORIAL2.DWG and TUTORIAL2\_RESULT.DWG will be copied to the directory

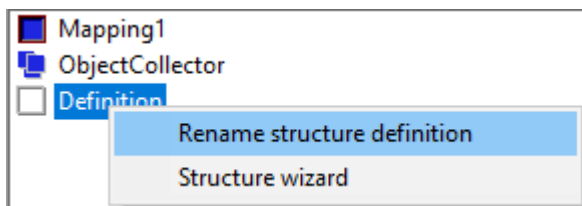
```
\USERS\[NAME]\DOCUMENTS\FARO\AS-BUILT XXX EN
ACXXX\TUTORIALS\FEATURE_DATA\HISTORICAL_FACADE_MAPPING\TUT
ORIAL2
```

The drawing TUTORIAL2.DWG is automatically opened in AutoCAD. Here you find again the structure definition of tutorial 1, but now the mapping task will be to record a natural stone façade.

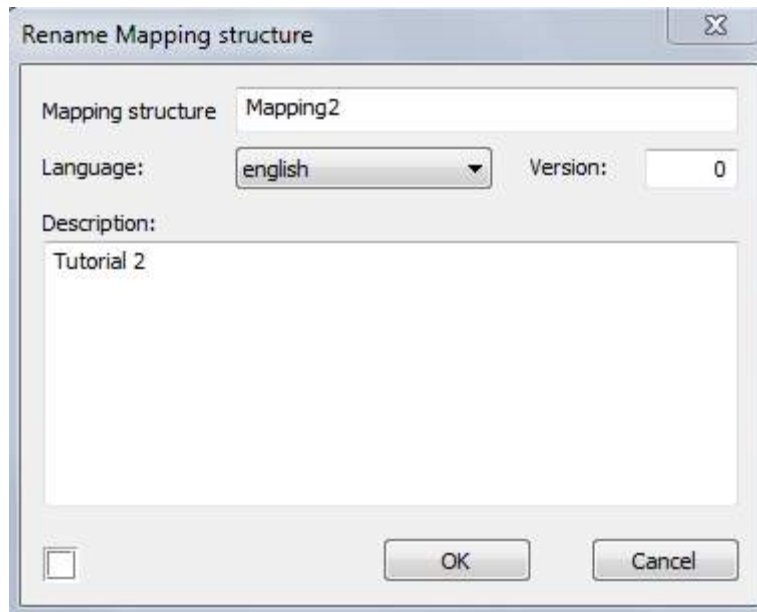
### 7.2 Changing the structure name

You will change the structure definition when working on this tutorial. Therefore we are working in the lower part  "Definition" of the structure view.

At first you will change the name of the mapping structure within the drawing. Therefore you find the entry →**Rename Structure definition** within the context menu of  "Definition".



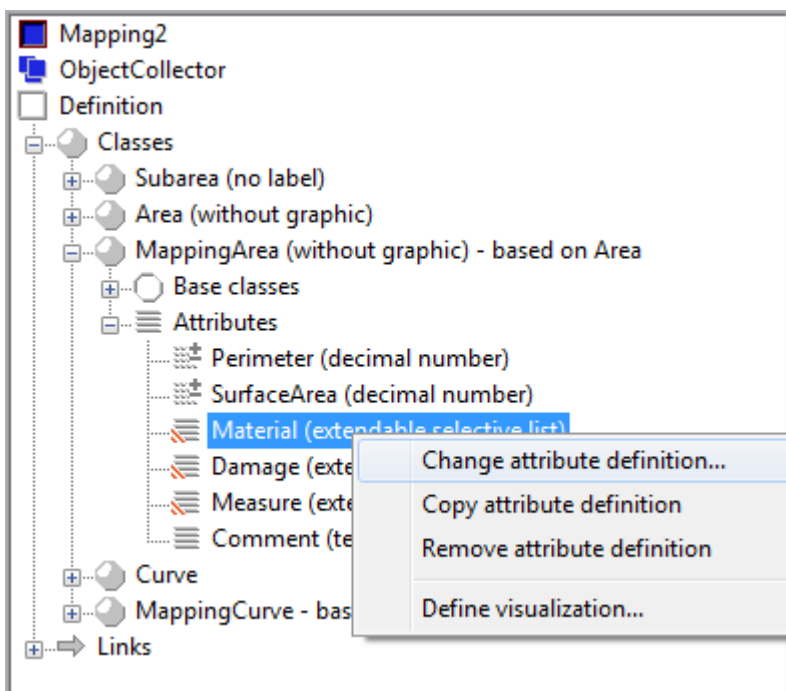
In the following window change the mapping structure name of "Mapping1" to "Mapping2" and enter a meaningful description, e.g. Tutorial 2.



### 7.3 Customize an attribute definition

Next, you will change the default values for the material, since several kinds of natural stone are to be distinguished.

Open the entry "Definition" within the structure view by double clicking the symbol. The classes and links are defined and managed here. When you click on the "+" next to the entry "Classes" all the defined classes are shown. Here you also find the class "MappingArea (without graphic) – based on Area", which you already know from tutorial 1. Via the plus symbol next to "MappingArea" you get to the entry "Attributes" where you find all attributes that have been defined. Move the cursor to the attribute "Material (extendable selective list)" and open the context menu.



The attribute characteristics and the default entries may be accessed via the option **→Change attribute definition**.

The screenshot shows the 'Attribute definition' dialog box. The 'Attribute name' is 'TypOfStone'. The 'Description' is 'Type of stone, select or add material'. The 'Attribute type' is 'extendable selective list'. Under 'Fill-in help', 'Offer last value' is selected. The 'extendable selective list' contains 'indefinite', 'sandstone', and 'limestone'. A callout box points to this list with the text: 'The default entries for the attributes are in this list. When filling in, they will be offered for selection.' The 'Show selection sorted' checkbox is checked. At the bottom, there is a text field containing a userenum configuration string and 'OK' and 'Cancel' buttons.


Enter the new attribute name "TypeOfStone" (note: no space allowed in an attributes name) and change the default values to "indefinite", "sandstone" and "limestone". Under "Fill-in help" select the option "Offer last value". That means that during mapping always the last attribute used will be offered. Confirm your selection with **OK**.

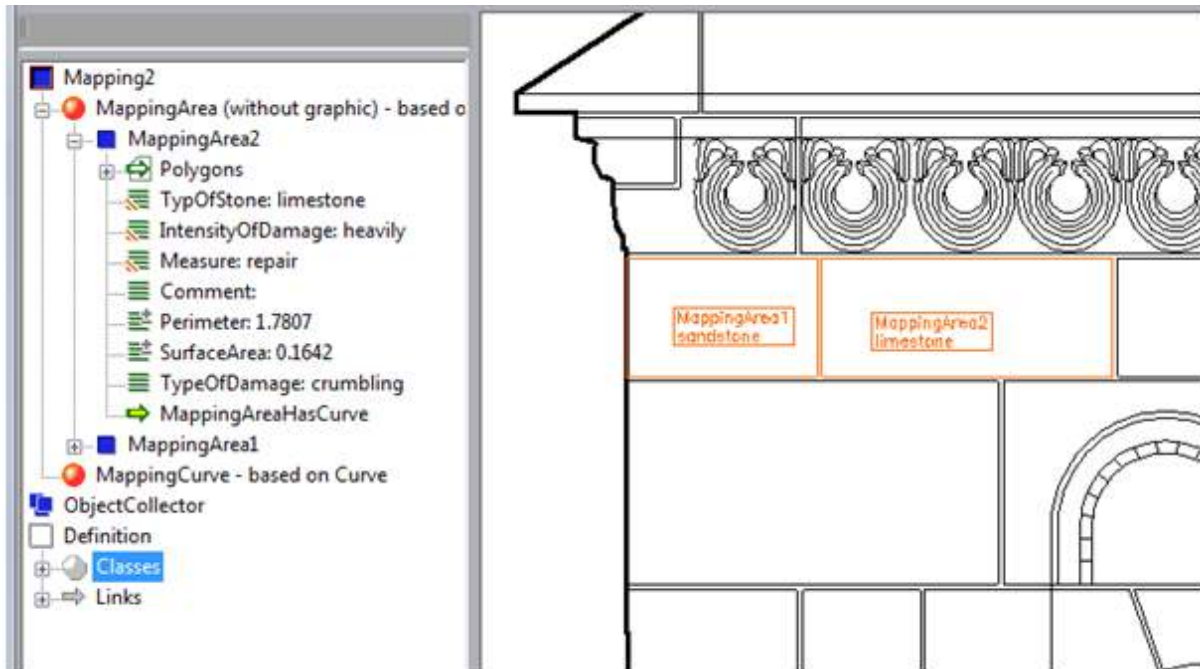
#### Excursus: Attributes

If you want to allow to add more types of stone while mapping, then select the attribute type *"extendable selective list"*. But if you want to limit the selection to existing entries, then use *"selective list"* for attribute type. You should also think twice whether you need neutral entries like "indefinite" or "not specified" for example.

The fill in help enables you to determine which value will be offered first when capturing data. The option "Offer standard" means, that always the first value of the (extendable) selective list is offered. The option "Offer last value" means that the attribute value of the object mapped before is offered.

Now you will test the new structure changes.

Zoom in the top left corner of the building and map the stone underneath the cornice with the characteristics: sandstone – medium degree of damage – measure preservation. In order to do so open the context menu of  "MappingArea" and select the method **→Add MappingArea (Inner point)** or **→Add MappingArea (Polyline)**. Take a look at the default values of the attribute "TypeOfStone". The attribute values that have been defined before should be in the list.







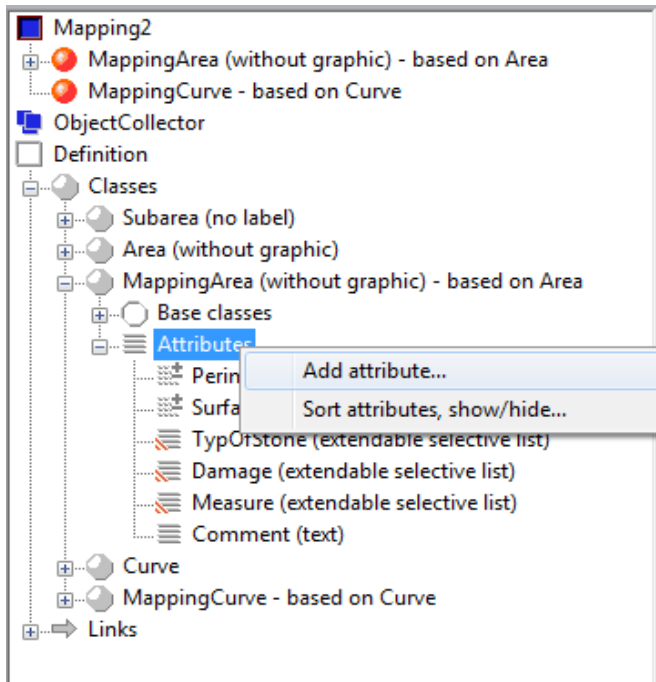
## 7.4 A new attribute

Now you want to display the damages at the stones more differentiated. You are not satisfied with only stating the degree of damage any more. The information regarding the kind of damage shall also be added to the mapping. Your considerations result in the following types of damage to be distinguished:

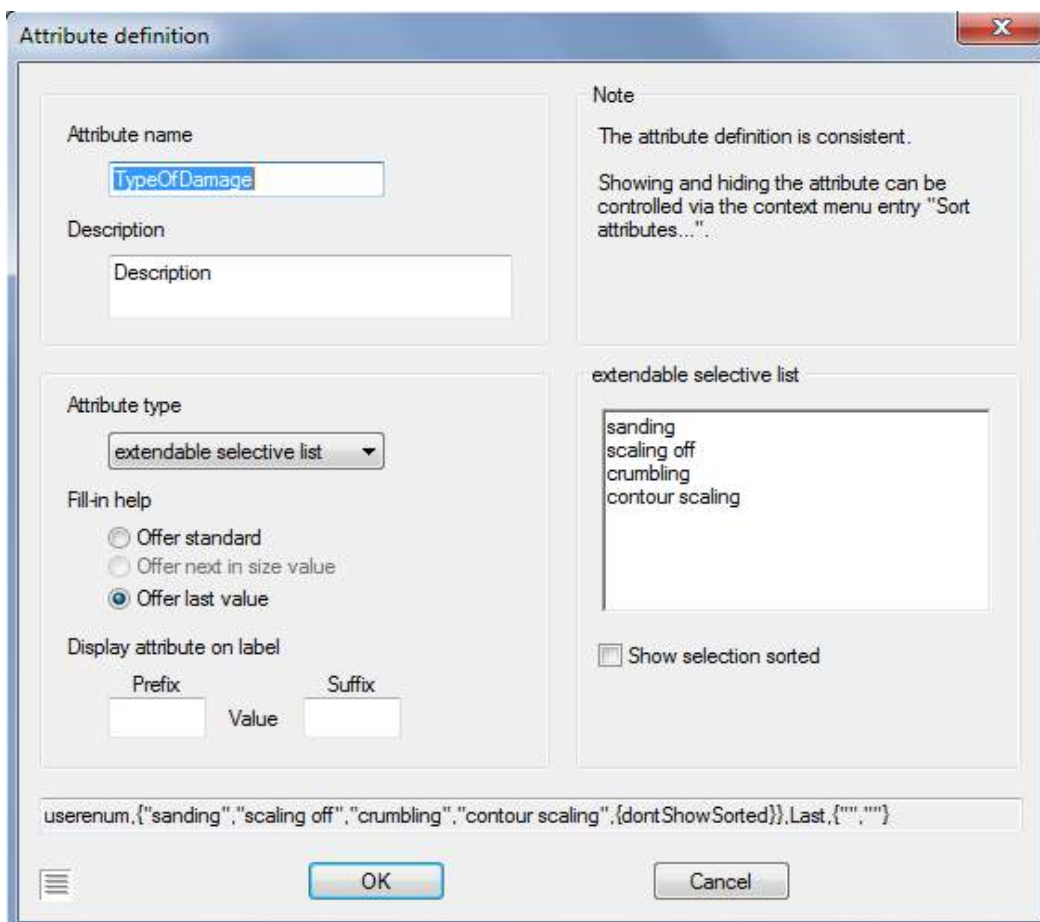
- sanding
- scaling off
- crumbling
- contour scaling

These types of damage may be used in any combination with the existing degrees of damage.

Open the structure view entry  "Definition" in the left part of the screen by double clicking the symbol. Move to the entry  "Attributes" of the class  "MappingArea". Within the context menu of  "Attributes" select the command **→Add attribute....**



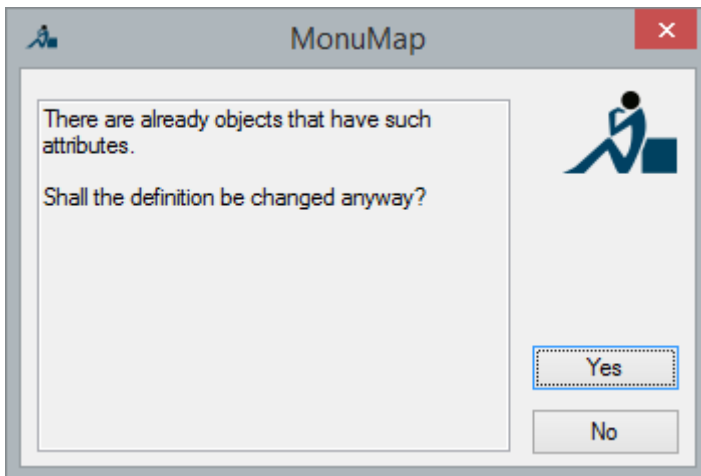
The window "Attribute definition" opens.



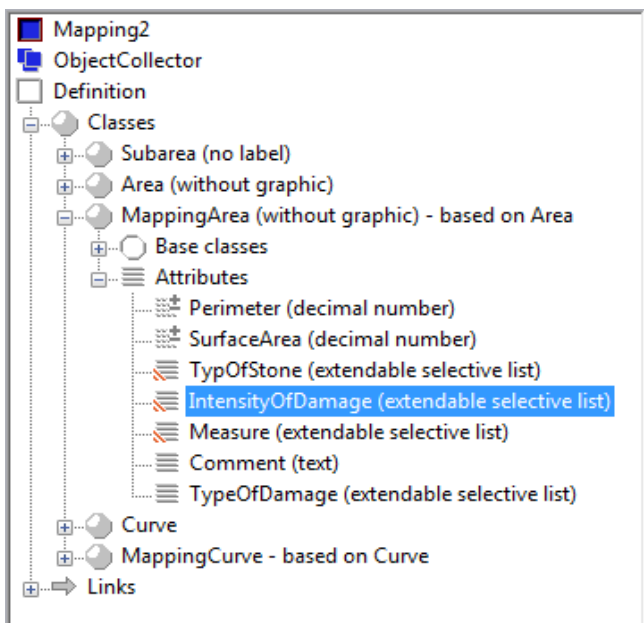
Enter "TypeOfDamage" as attribute name and a description that makes sense and is easy to recognize. You may use "extendable selective list" or "selective list" as attribute type again, depending on whether you want to permit additional entries or not. Select "extendable selective list". Some default values are displayed within the "extendable selective list" window. These should of cause be changed. Here it is

important again, to decide whether a neutral description as for example "indefinite" is to be inserted. After confirming with **OK** you see that the newly created attribute "TypeOfDamage" has been inserted at the last position under "Attributes".

Now, the attribute name "Damage" as a description for the degree of damage is confusing. In order to make it easier to distinguish the degree of damage from the type of damage this attribute shall be named "IntensityOfDamage". The procedures to do so are as described in chapter **7.3 Customize an attribute definition**. Please run the command **→Change attribute definition** from the context menu of "Damage". You find the attribute "Damage" within the attribute list of the class "MappingArea". You can answer **YES** to the query below since all you want to do is change the attribute name.



Change the attribute name in the appearing window to "IntensityOfDamage" and confirm your entry with **OK**. The attribute name is changed for the already mapped datasets automatically.

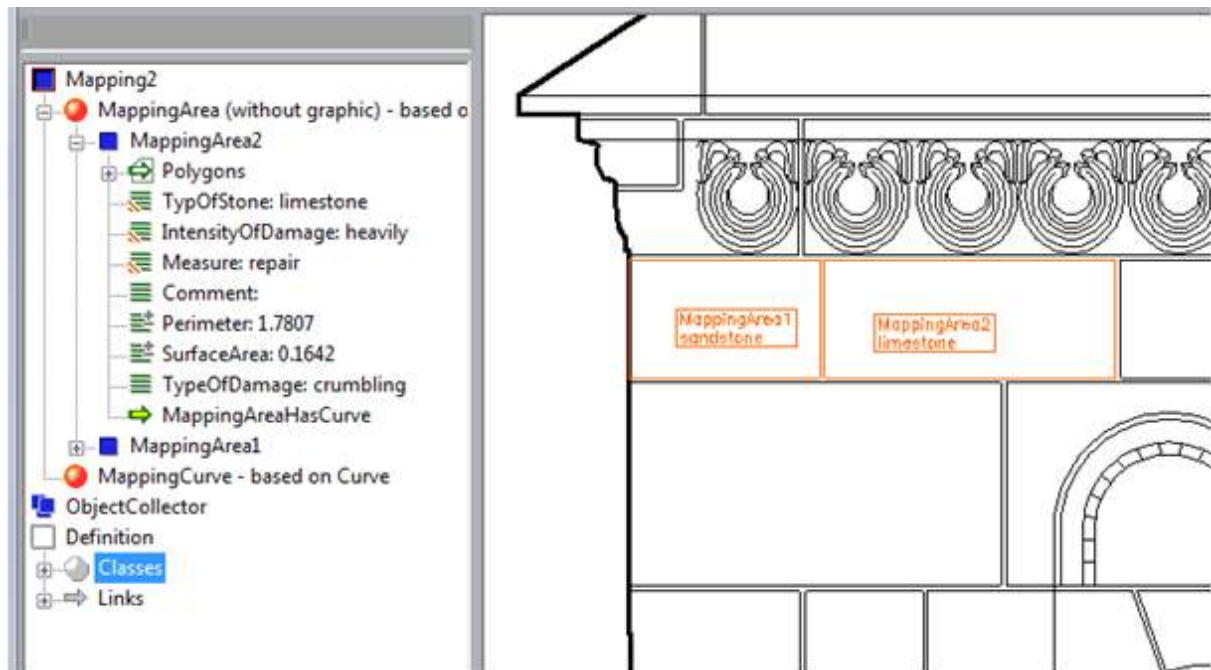


For checking purposes map the next stone using the new definition and following attribute values:

- limestone



- crumbling
- heavily damaged
- repair.



#### Excursus:

#### Behavior of old data sets when adding attributes afterwards

In our example the attribute "TypeOfDamage" has automatically been generated for the existing stone "MappingArea1". It was filled with the standard value, which in this case is sanding. By the question mark next to the entry "TypeOfDamage: sanding" you can see that no value has been assigned explicitly but a default value has been assumed. Therefore, when large datasets already exist, it is advisable to place a neutral value like "not specified" to the first position in the value list when adding attributes.

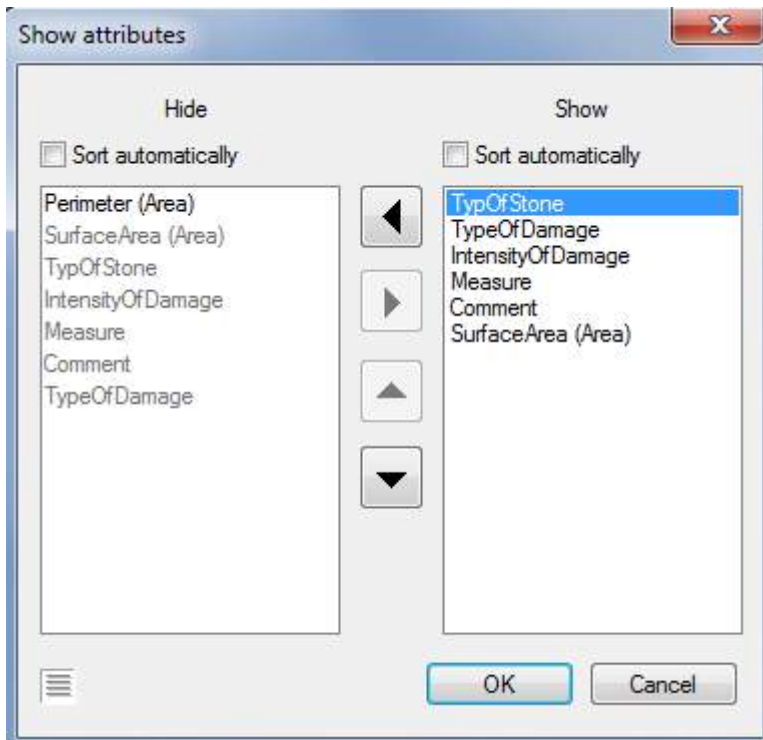
**Important remark:** You should avoid changing a structure definition after having recorded data already. Always think the structure definition through **before** capturing data.

## 7.5 Sorting attributes

The order of the query for the attributes corresponds to the order of their generation. The attribute added last is prompted last as well. Now you would like the query to be executed in a reasonable context, that, for example the type of damage is prompted before the degree of damage.

You may control the visibility and order of the attributes by the command **→Sort attributes, show/hide...** from the **☰ "Attributes" context menu**.

Use the arrow keys between hide and show in order to rearrange the attributes.



The "Perimeter" attribute is rather uninteresting in this connection and therefore it stays hidden.

## 7.6 A new visualization




The new attribute "TypeOfDamage" does not have a visualization yet. The existing visualization of the type of stone does have to be changed as well, due to the change of name and the connected change of contents.

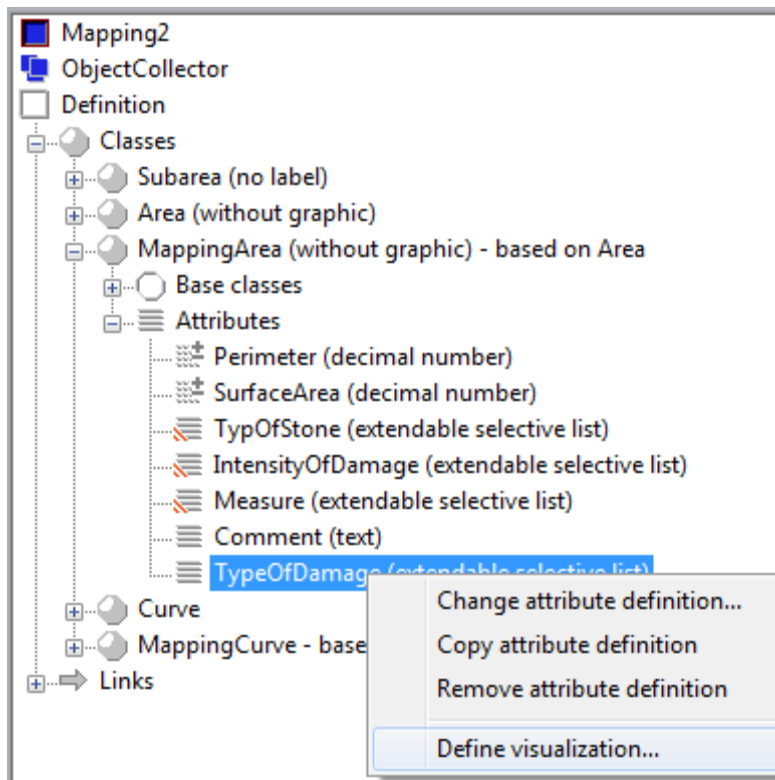
### Excursus: Visualization

Visualizations refer to attributes. The definition of the visualization, this is the specification how to visualize (hatching, solid hatching, block, color...), is carried out in the definition section of the structure view.

Whereas switching a visualization on or off is done in the upper part of the structure view via the context menu of the according attribute.

In the viewer mode (without license) only the visibility of the visualization can be controlled but not its definition.

The definitions for the visualizations are created under  "Definition". Open the context menu of  "TypeOfDamage". You find it within the attribute list of  "MappingArea".



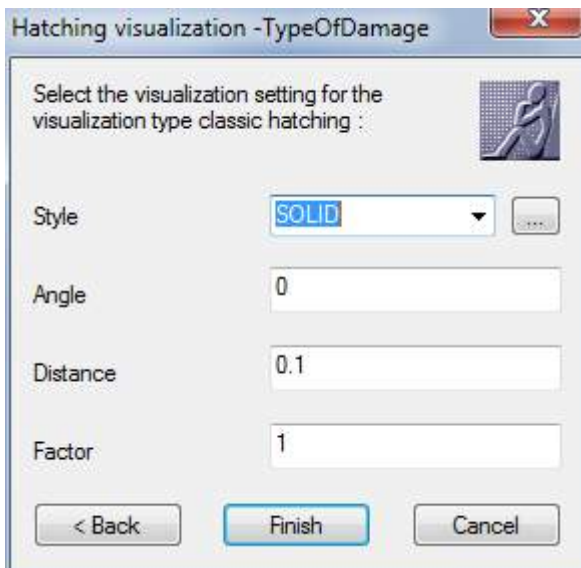
You reach the window "Visualization definition" via **→Define visualization....** Once the window is open press the button **Generate automatically**. An assistant for the automatic generation of value ranges and the according visualization settings opens.

At first you are prompted for the visualization type as well as the layer and colour. Keep "Hatching" as visualization type. Do the same with the "Vary colour" option. Thus the assignment of colours is done automatically for the "value ranges of the type of damage". In order to create a well arranged layer structure, select the layer "VisDamage" for the current case from the selection box. The damage intensity is already visualized on this layer. You could also create a completely new layer. Press the button **Next>**.

### Excursus: Layer structuring

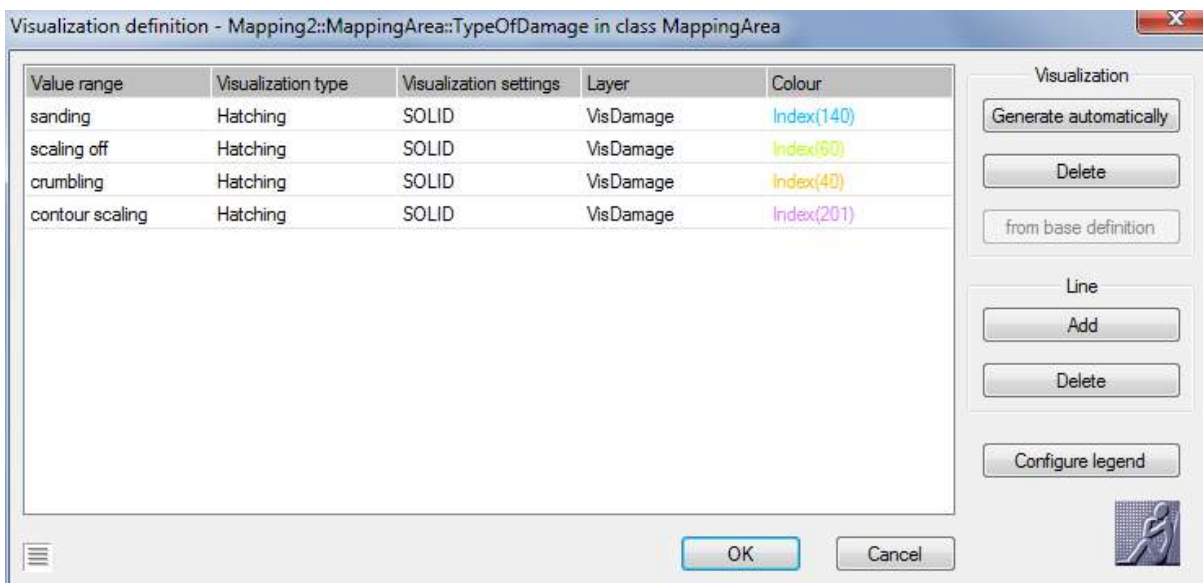
It is advisable to put different attributes on different layers when working with the Layer Properties Manager of the AutoCAD layout for a printout. For the final drawing you may activate all visualizations. The visibility within the layouts is determined by the button "Current VP Freeze" within the AutoCAD layer manager. VP stands for view port.

The prompt for the hatching style etc. is the second step. Just leave this as it is. The default setting is "SOLID", which corresponds to an area filling hatching.

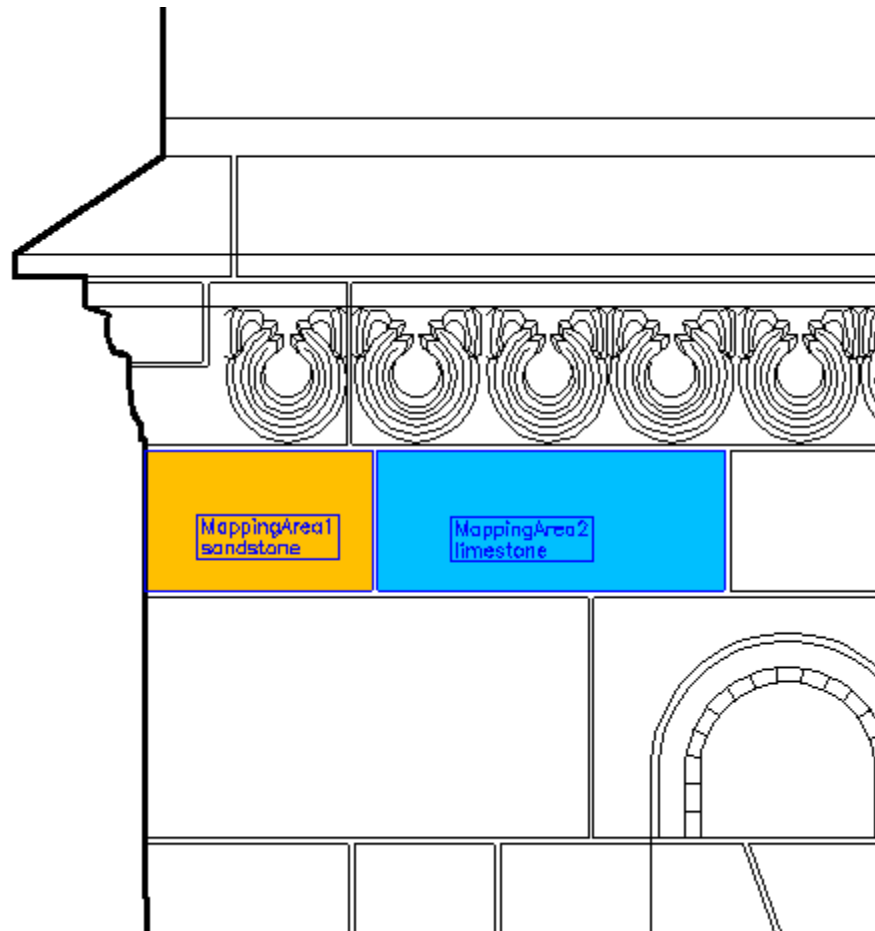


Here you may change and customize the characteristics of the hatching in detail. For this case leave the settings as they are and close the assistant by pressing the button **Finish**.



The colours with the numbering of the 256 AutoCAD colors may not meet your requirements. To change them proceed as follows: Click on the colour entry to be changed, e.g. **Index(201)** and an AutoCAD selection box for the colours opens. There you can select a new colour.



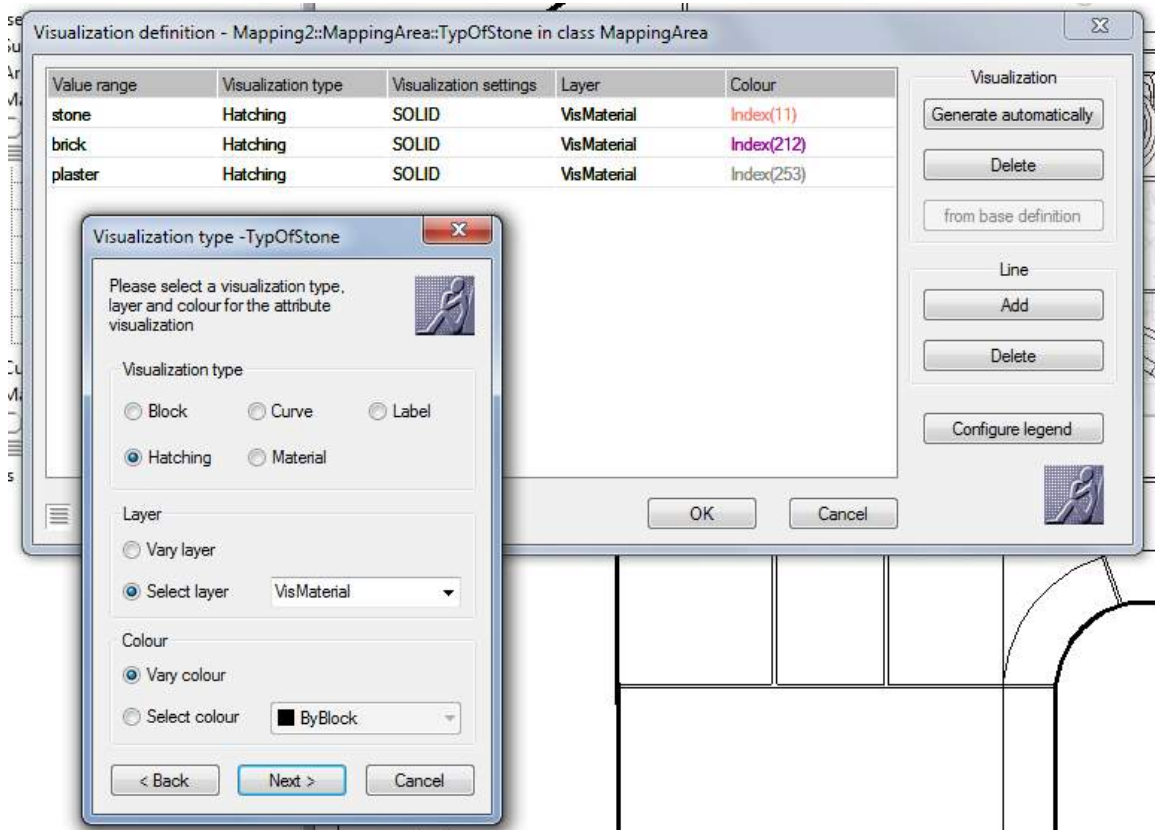
After finishing the definition, the visualization is activated right away and you see the following result on your screen.



The structure definition for the stone types has to be customized as well since the entries have been changed compared to the original version. Now you may not use a solid hatching, since it would interfere with the type of damage. Therefore you will define a line hatching.

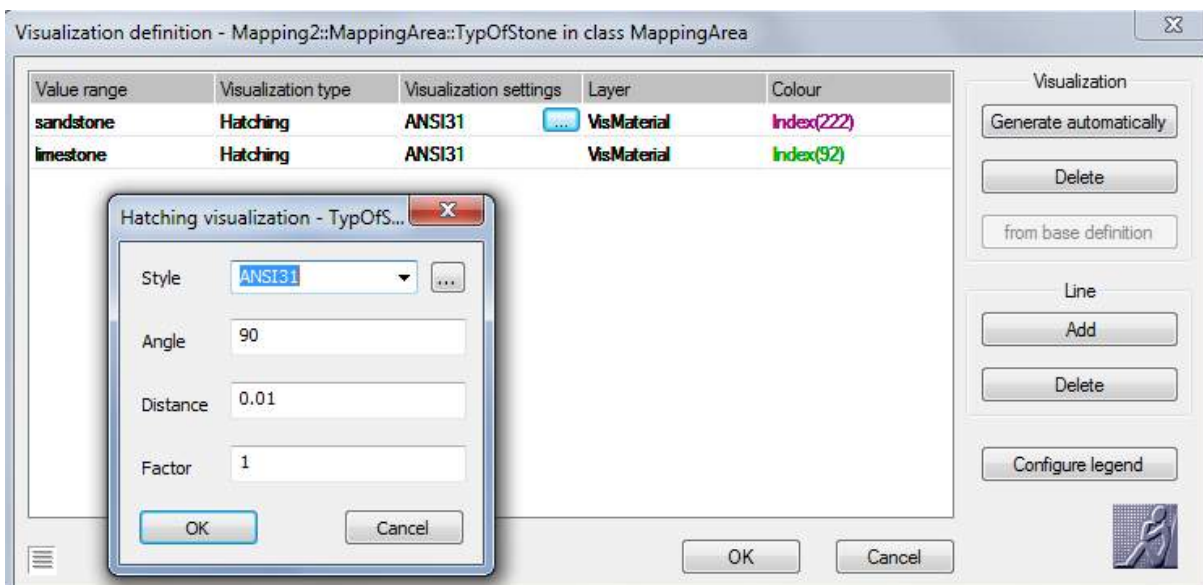
Open the window for the visualization definition for the type of stone as described above (context menu:  "TypeOfStone" under  "MappingArea"). Here you still find the old entries for the value ranges like "stone", "brick" and "plaster". The original definition, which is not valid any more due to the changes, has to be revised completely. If you select the button **Generate automatically** from the visualization window the existing entries are deleted and replaced by new ones. The wizard will guide you through the windows step by step.

Within the first window set the visualization layer to "VisMaterial" and then click **Next>**



In the first dialogue box set the visualisation layer to “VisMaterieal” and then click **Next>**

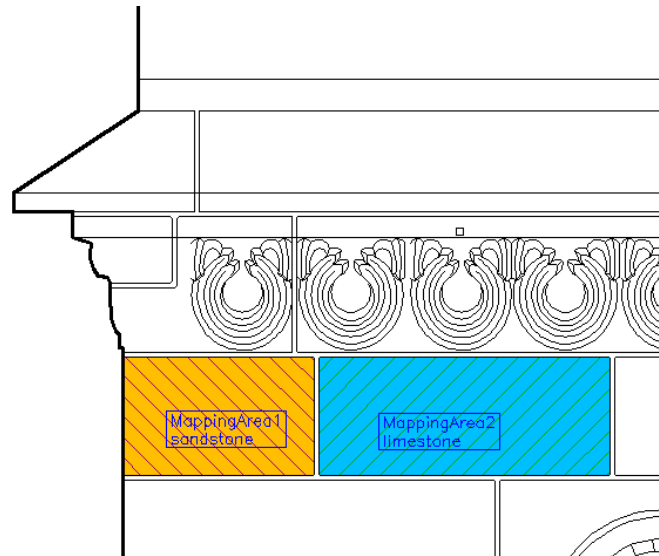
Then the window "Hatching visualization – TypeOfStone opens. The hatching pattern is selected here. Select ANSI31 under "Style", it is equivalent to a 45° line hatching. Leave the angle at "0" and set the factor to 0.01, thus the hatching will be clearly visible. Quit the assistant with "**Finish**".



In doing the above you have created a visualization for the value "indefinite". You do not want to have this value displayed though. You may delete the line by moving the cursor over an entry of this line. Then click the button **Delete** in to the right part of the window in the section "Line". Also select colours for "sandstone" and "limestone". In

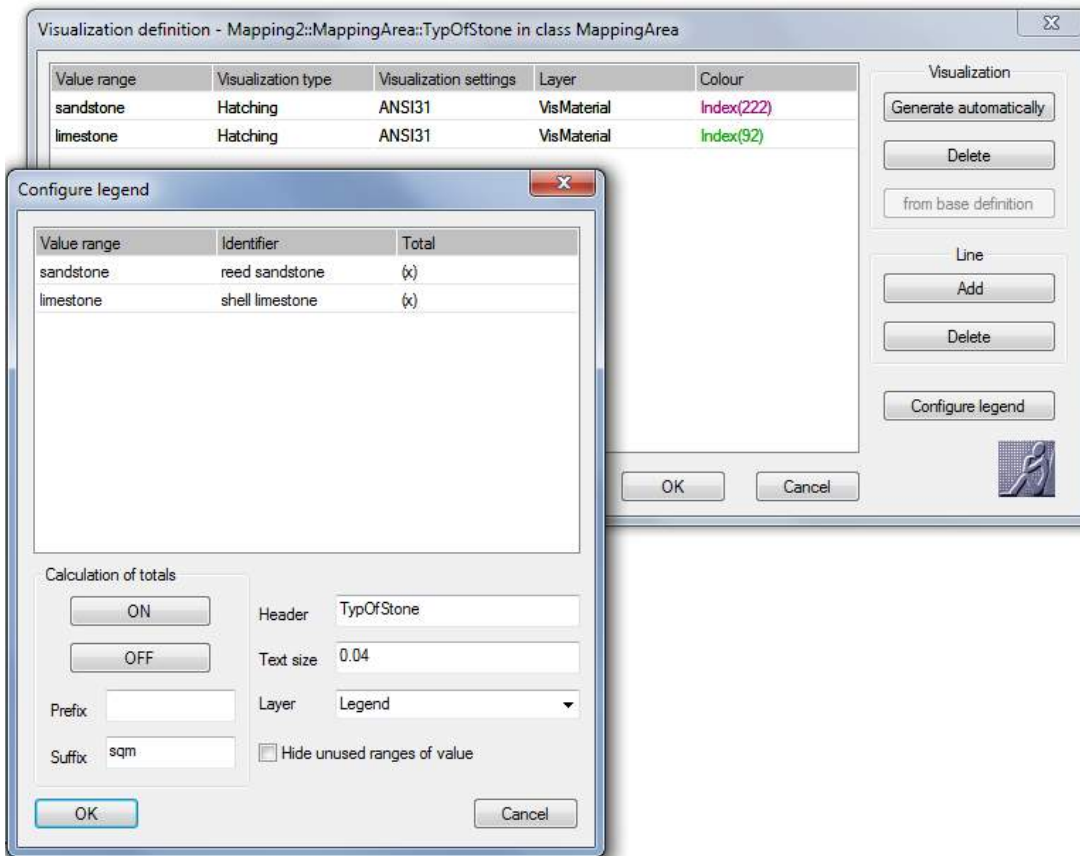
order to be able to recognize the sandstone more easily you should assign a new angle to the hatching. Therefore click the "ANSI31" entry of the line of sandstone. The window for the settings of the hatching characteristics opens. Set the angle to 90° and close the window by pressing the **OK** button.

The following mapping has been generated:





## 7.7 Configure legend

Now the legend has to be adapted to the new material. Within the window of the visualization definition for the attribute `TypeOfStone` you find the button **→Configure legend** in the lower right. Go to the visualization definition again and click the button. Within the opening window "Configure legend" you find the types of stone listed.



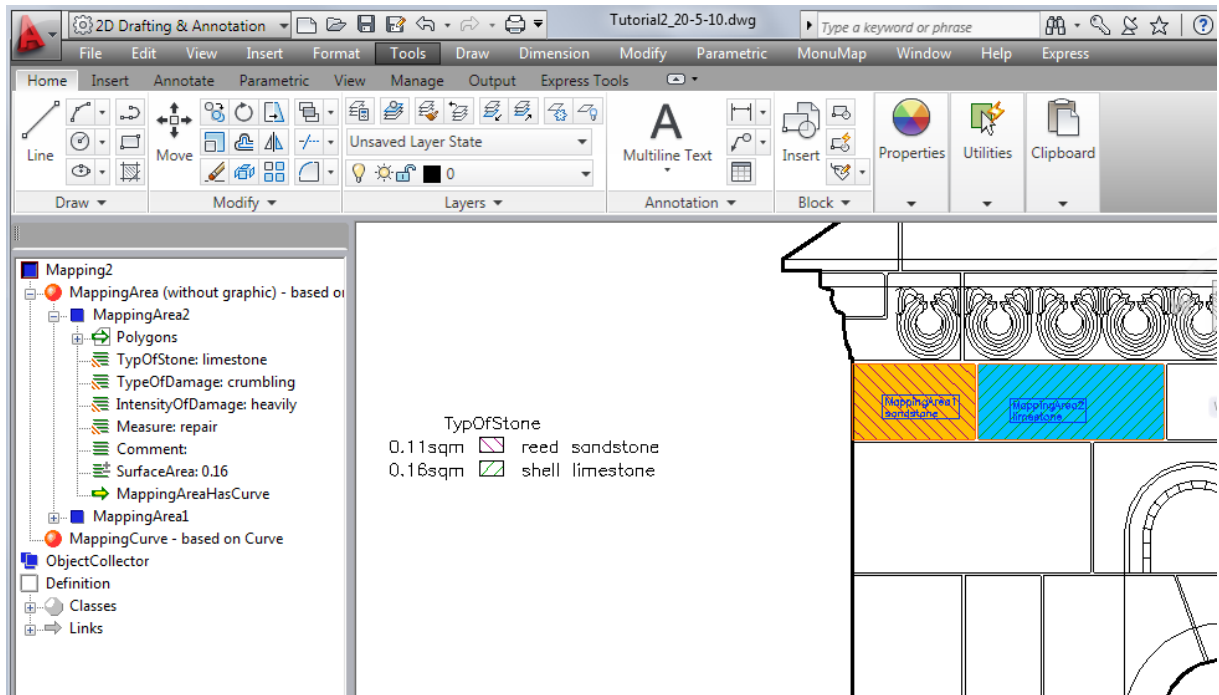
A name differing from the attribute name may be defined within the column "Identifier". That name then is displayed within the legend. Change the identifiers of "sandstone" to "reed sandstone" and "limestone" to "shell limestone".

- **Calculation of totals:** The cross-sign for the totals means, that the area values are added up and shown in the legend. Via the buttons ON and OFF you are able to control the display. Keep the totals calculation activated.
- **Hide unused value ranges:** Values that are not used in the mapping may be omitted in the legend via this setting option.
- **Text size:** This value controls not only the height of the text but also the size of the rectangle with the legend display. Keep these settings as they are.

Keep the remaining settings and confirm your entries with **OK**. You then get back to the superior window of the visualization definition. Confirm with **OK** again. For testing purposes insert the new legend. Proceed as described in chapter →**6.6 Application of visualizations**. Open the context for  "TypeOfStone" of any of the datasets  "MappingAreaX" and select →**Visualization....**





## Tutorial Facade Damage Mapping (Part 2) Modeling and configuration – Tutorial 7-15



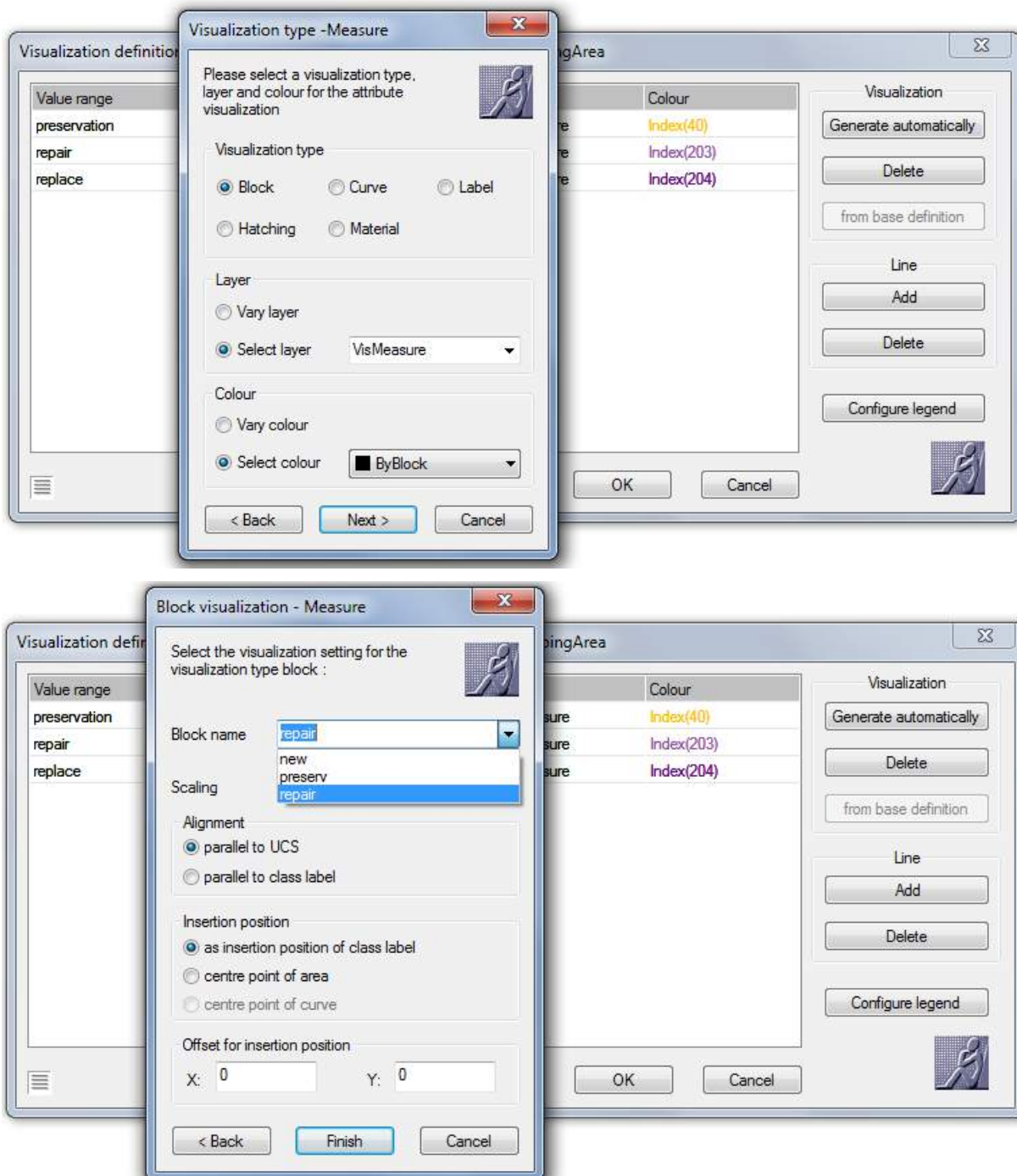
The number of decimal places for the area values are conform to the set grade of accuracy of AutoCAD. You may control these settings under menu option "Format – Units".

## 7.8 Visualization with symbols

The measures are to be visualized by symbols. For that purpose the drawing contains the AutoCAD blocks "repair", "new" and "preserv".

Open the visualization definition of the attribute  "Measure" within the class  "MappingArea". Select **→Generate automatically**. Set the option "Block", the layer "VisMeasure" and "byBlock" as colour within the appearing window.

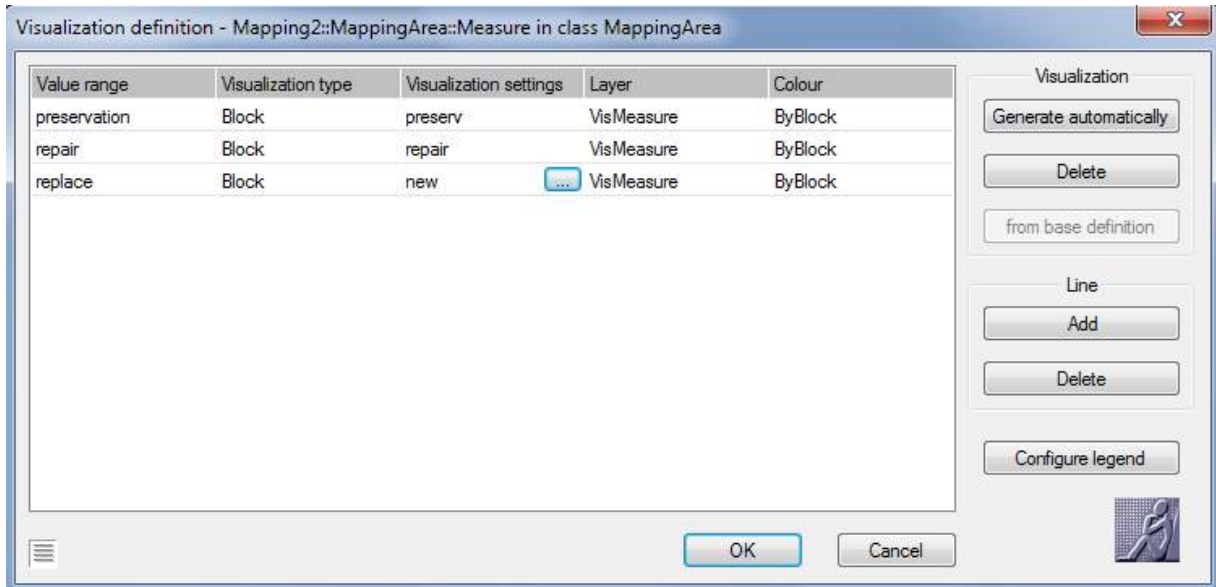
By clicking **Next>** you get to the window for the block assignment.



For the block select the entry "repair" at first and click **Finish**. In the following you will be prompted to confirm the settings, although they do not vary. Confirm the above, the variation of the settings is done later.

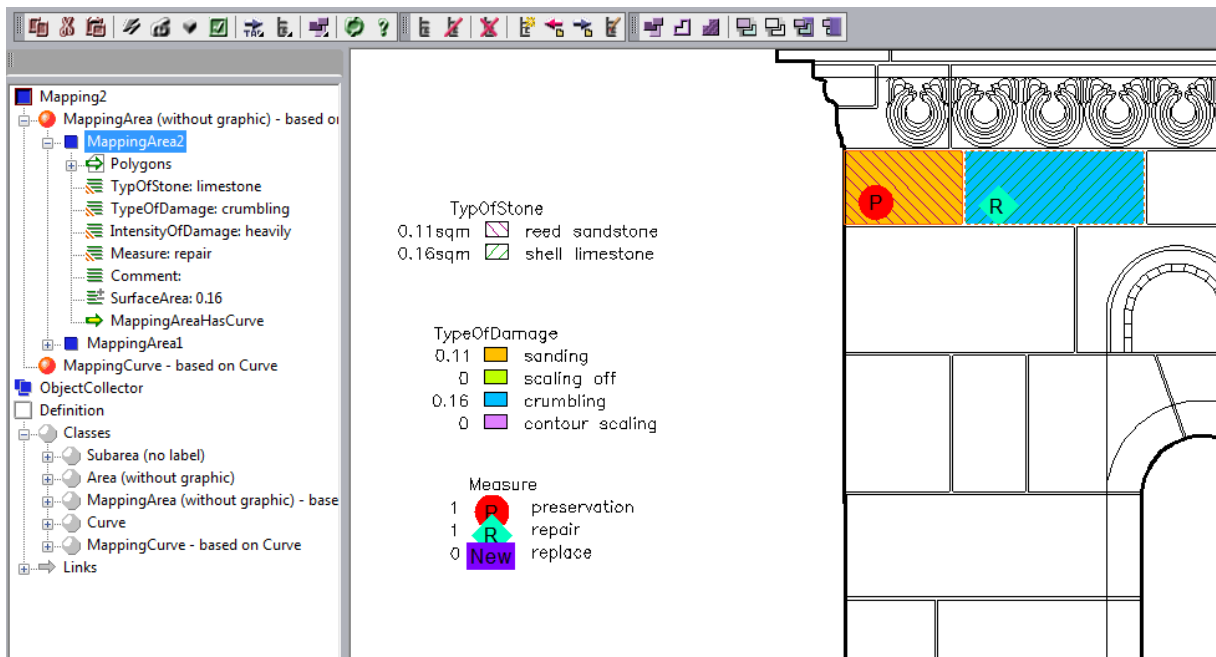
The window "Visualization definition" opens again containing the new entries. At first delete the line "no measure".

Then in the column "Visualization settings" click into the line "preservation" and select the block name "preserv" within the appearing window. Close the window with **OK**. Do the same for the line "replace" and select "new". Finally your definition window has to look like the following graphic.




When you confirm with **OK** the blocks will appear within the drawing right away.

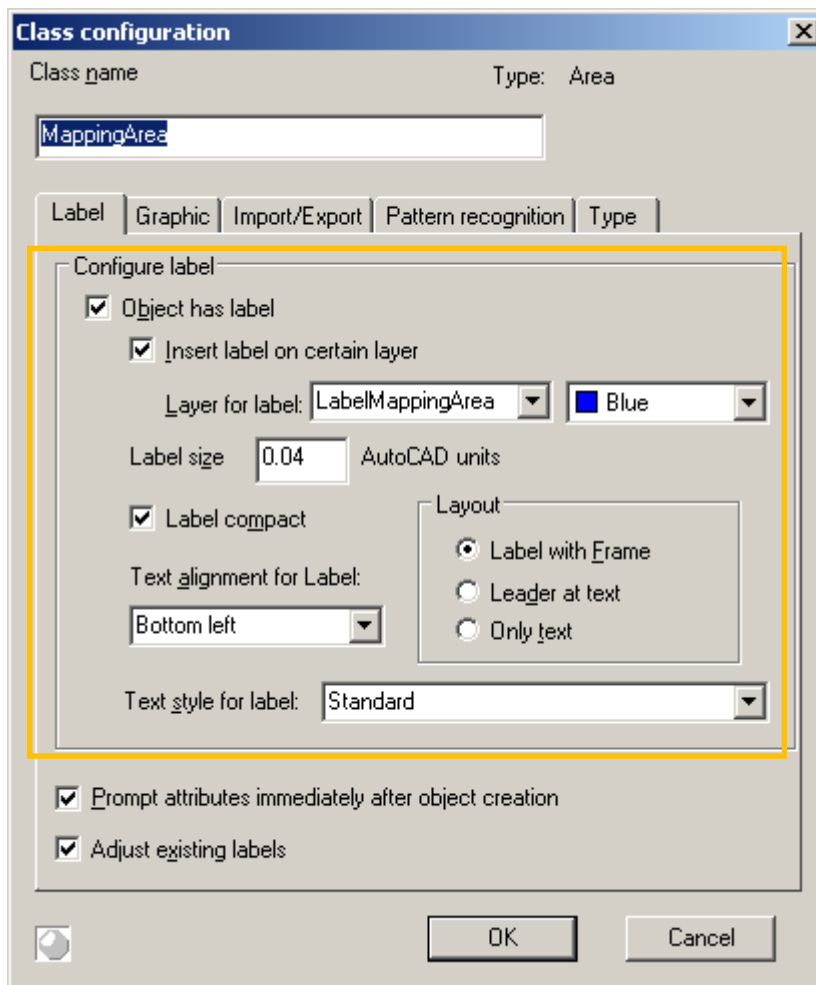
For a printout of the mapping it is advisable to switch off the layer of the labels. If you insert the legend for the type of damage and the measures afterwards, you may get the following mapping.



## 7.9 Layout

Now a few changes for the layout. Larger labels are desired and the layer on which the label is placed shall get another name.

For this you stay in the definition area under  "Definition". Move to the entry  "MappingArea". Open the context menu and select **→Configure...**. A window with several setting options opens. Only the ones in the middle under "Configure label" are of interest now.



Layer for label: Instead of the so far used layer name "LabelMappingArea" enter the new name "MyTutorial" for the layer.

Change the label size from 0.04 AutoCAD units to 0.06 units.


At last uncheck the option "Label compact".


If you confirm your entries with **OK** now, all labels of the according class are adapted to the new display settings. The labels are moved to the new layer. When creating new objects now, the according labels will get a pointer tip:

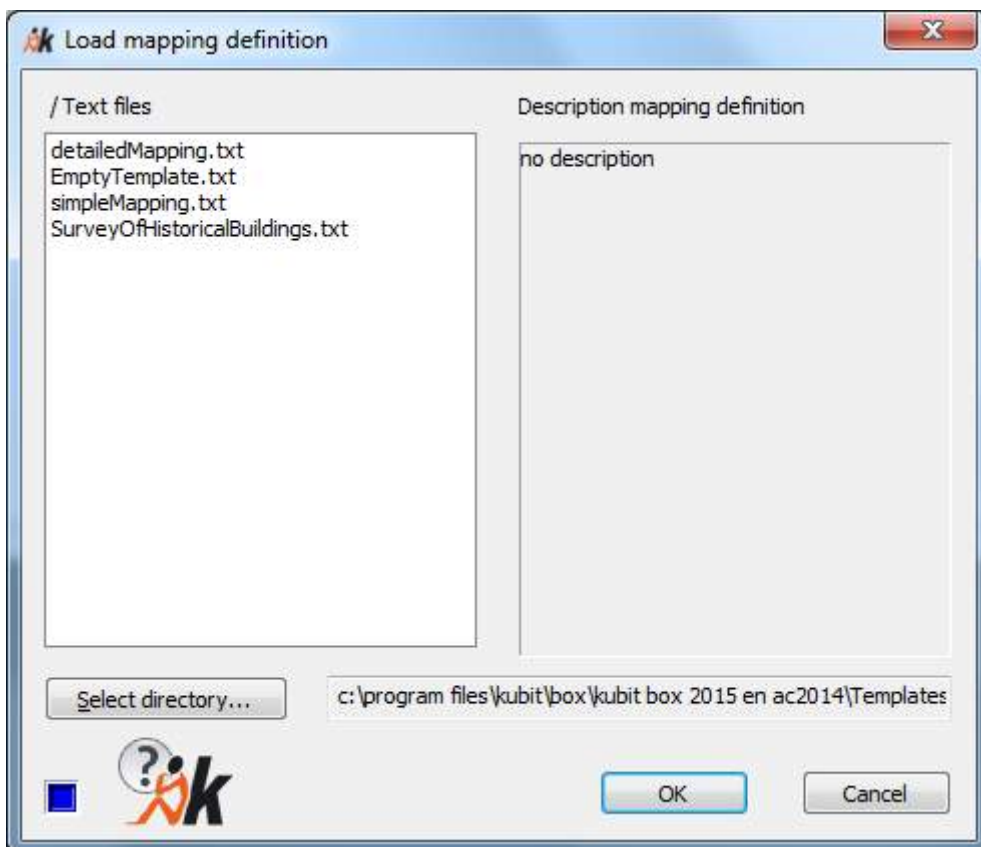


## 7.10 Starting a new project

You have finished the definition of the mapping structure and now you want to open it for additional projects or drawings.

In order to do so run the command  → **Save structure definition** and save the structure definition with a distinct file name, e.g. "TUTORIAL2\_RESULT.TXT".

For a new drawing without structure view, you may insert the saved structure into your drawing by running the command  → **Load structure definition**. The following window opens:



Use the button **Select directory...** in order to browse to the directory to which you have saved the structure definition. All structure definitions of the directory are displayed in the window. If you now mark your structure definition TUTORIAL2\_RESULT.TXT, you will see the description for the structure definition which you entered at the beginning of this tutorial. If you press **OK** the structure view is loaded into the current drawing.

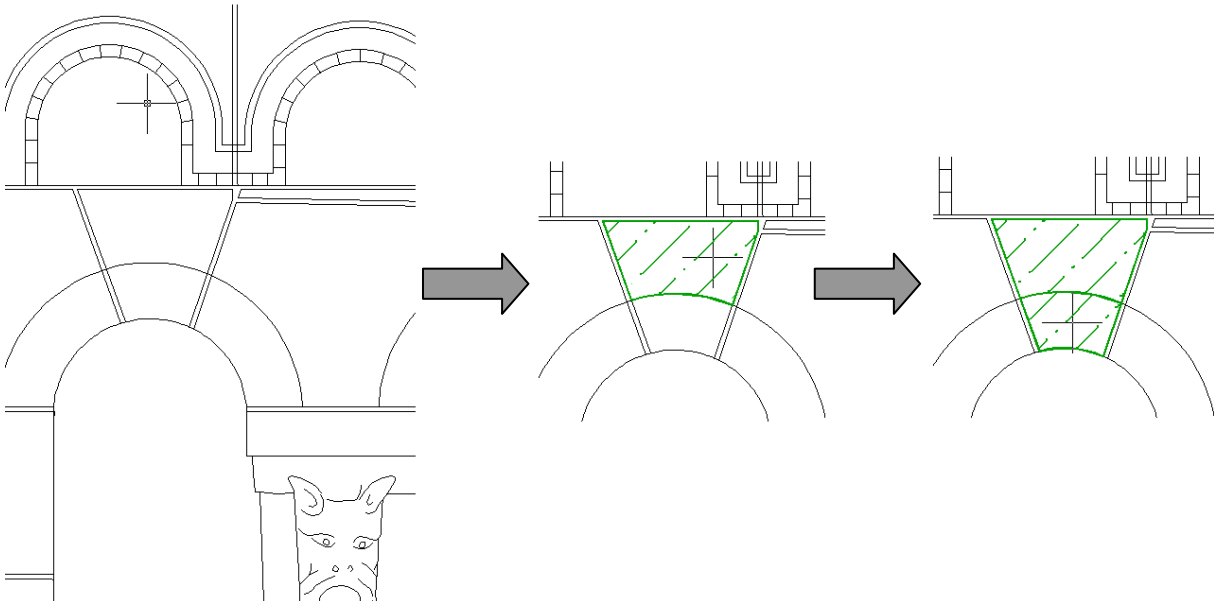
The structure definitions used for the tutorials can be found within the folder installation directory of the As-Built for AutoCAD in the folder Templates\Heritage\_Features. When loading a structure definition the window, displayed above automatically points to that directory.

## 7.11 Effective mapping

### 7.11.1 Procedures of stone selection

#### "Inner point" with several subareas

You want to map the top left keystone of the arcade opening. The drawing has been laid out in a way, that the stone boundary is not a closed polyline. It rather is build of the line of the joint and the boundary of the opening. If you choose the option "Inner Point" for the mapping you have to click into the upper and lower area.

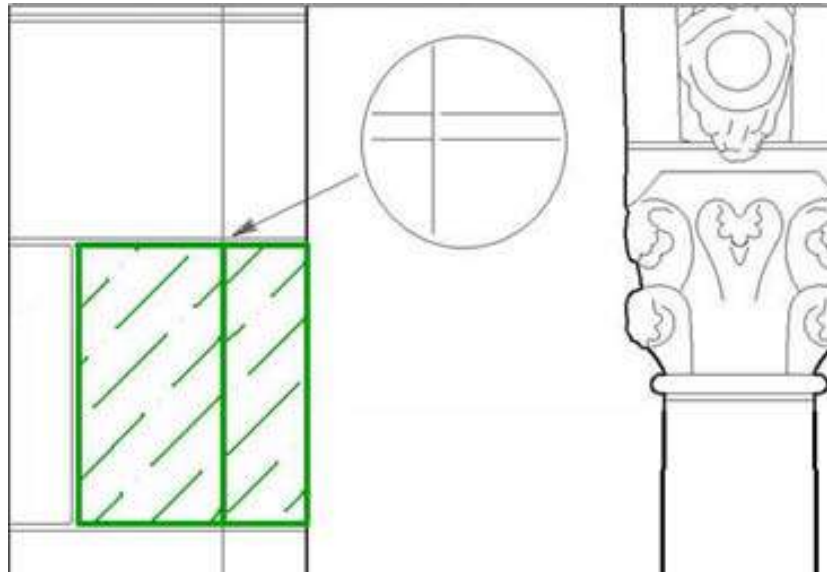


The result is a closed polyline which encloses the complete area that has been hatched in green.

To accelerate collecting and mapping stones of similar types (e.g. same type of stone) use the **→Fast Capture** option. This tool allows you to create, list and execute commands that are reapplied. Of course the Fast Capture is not limited to stones but possible for all kinds of objects. Further details are described at the command reference.

#### Gaps in the drawing

In drawings that have been created photogrammetrically there are sometimes lines of joints and edges of the building which do not intersect, like in the field left of the arcades for example.

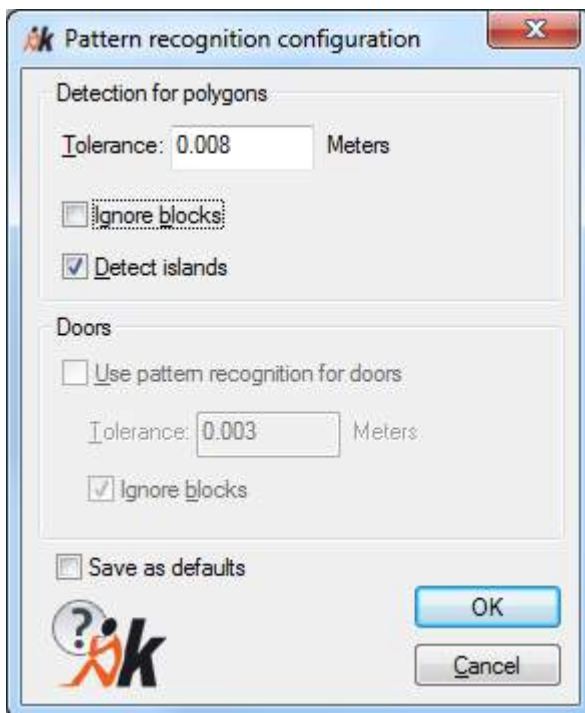


Such small gaps complicate the mapping, especially if hatchings are to be applied. The software offers a function to skip gaps in order to be able to map despite such drawing errors.

Start the command "Inner point" in order to create a mapping area. Select the option "Configure" by entering "c" into the command line.

Click inner point of a polygon for MappingArea or [Configure]: c

Within the opening window you may configure the pattern recognition.



Within this window you may define a tolerance. It states how large the gaps may be in order to be skipped during the outline recognition. Because of that option a mapping is possible even for slightly inaccurate drawings. Do not set the value for this tolerance too high since real gaps shall be recognized as gaps.





## 8 Tutorial – Archaeological Excavation Documentation

By means of an easy example this chapter inducts you to the work with the feature data management. Only the basic principles will be explained here.

We recommend that you follow the tutorial step by step in order to get a first impression of the possibilities which are offered by the feature data management.

### Preparation

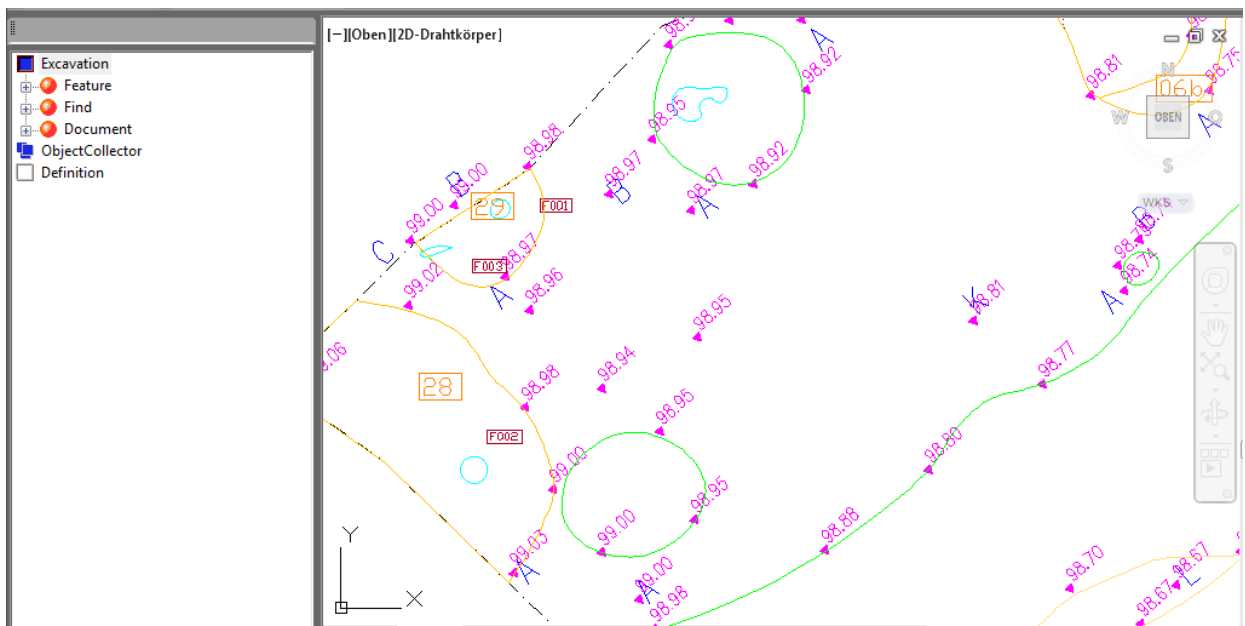
In order to work on the tutorial you will need a completely installed version of AutoCAD (or AutoCAD LT), into which As-Built for AutoCAD has been embedded. After you have started AutoCAD please open the tutorial file by going to the As-Built Feature Data ribbon tab and clicking "Help – Tutorial Excavation Data"

The file "FeatureTutorial.dwg" will open from the following folder:

```
\USERS\[NAME]\DOCUMENTS\FARO\AS-BUILT XXX EN
ACXXX\TUTORIALS\FEATURE_DATA\ARCHAEOLOGICAL_EXCAVATION
```

### Start

Please open the structure view first. In order to do so select →**Feature data** – **Structure** – **Structure view on**



### A glance at the Excavation structure view

The above drawing shows an excavation plan.

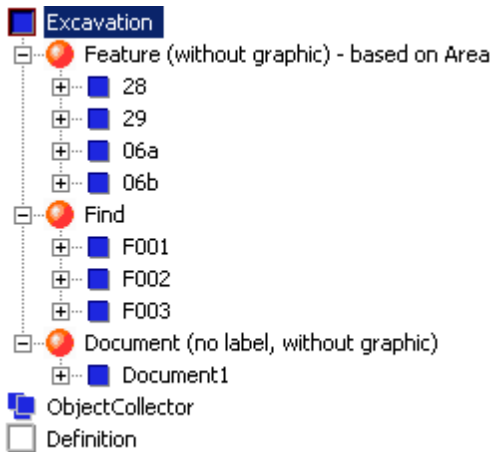
To the left of the drawing area you can see the **structure view**. Within this window all non-graphical data is managed. It is shown as structure view, similar to the Windows Explorer view.


The structure view consists of three parts: The blue symbol "Excavation" shows, sorted by "Feature", "Find" and "Document", all feature objects, their polygons and



their attributes, which have already been integrated into the list. Below the gray "Definition" symbol the definition data for this list is shown. Here new attributes may be created or already existing ones may be changed. Collected data may be filtered or edited using the "ObjectCollector".

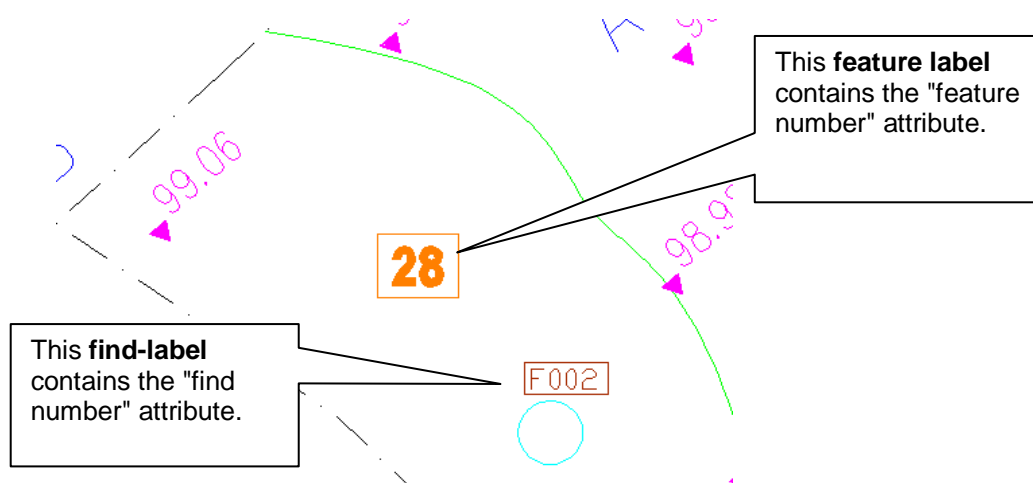
This tutorial will only explain the upper part "Excavation".

Open the structure view by double clicking the icon. The subentries may be extended by a double click as well. The following structure can be seen



The entries marked by the  symbol are **classes**. Available are the classes "Feature", "Find" and "Document". Not shown in the area list but existing in the definition part there are three more classes "Area", "Subarea" and "FeaturePolygon". Those serve as auxiliary classes for the polygons of the features and should not be deleted!

For the entry " Feature" you can see four **objects**, which show the  symbol. Please take a look at the adjoining drawing. You will find the four features in the left part of the plan. They have been marked with an orange **label**, which contains the feature number (e.g. "28").



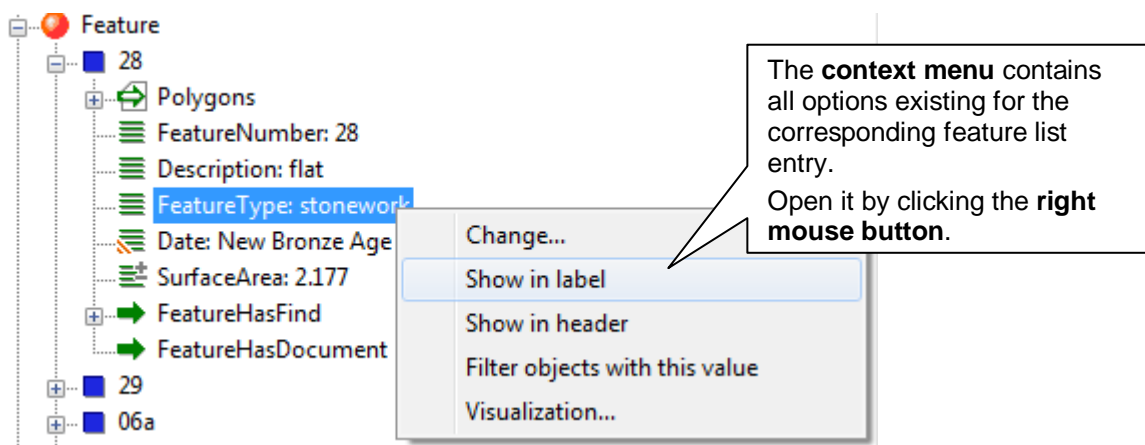
The entries of the structure view have been linked to the geometry of the drawing. Every entry of the list has a label and a geometry element within the drawing. Click onto the label "28" within the drawing. The corresponding entry of the structure view

will open automatically. The same happens when clicking on a closed polyline that belongs to the feature. It also works the other way round. If you click the right mouse key on the entry "■ 28" of the list and select "Show" within the context menu, the corresponding label will be indicated shortly by an arrow.

**Attributes** have been assigned to the objects. Open the object "■ 28" by double clicking it. The existing attributes for the object will appear

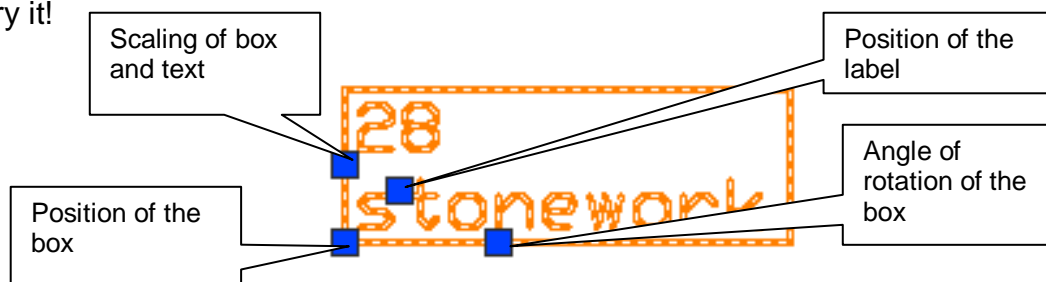
- Feature number
- Description
- Feature type
- Dating
- Surface area

You may also make the attributes visible within the label. To do so mark the subentry "☰ FeatureType: stonework" for example and open the context menu by clicking the right mouse button.



Then click →**Show in label**. After that the feature type will be shown in all feature labels. Just try out those settings.

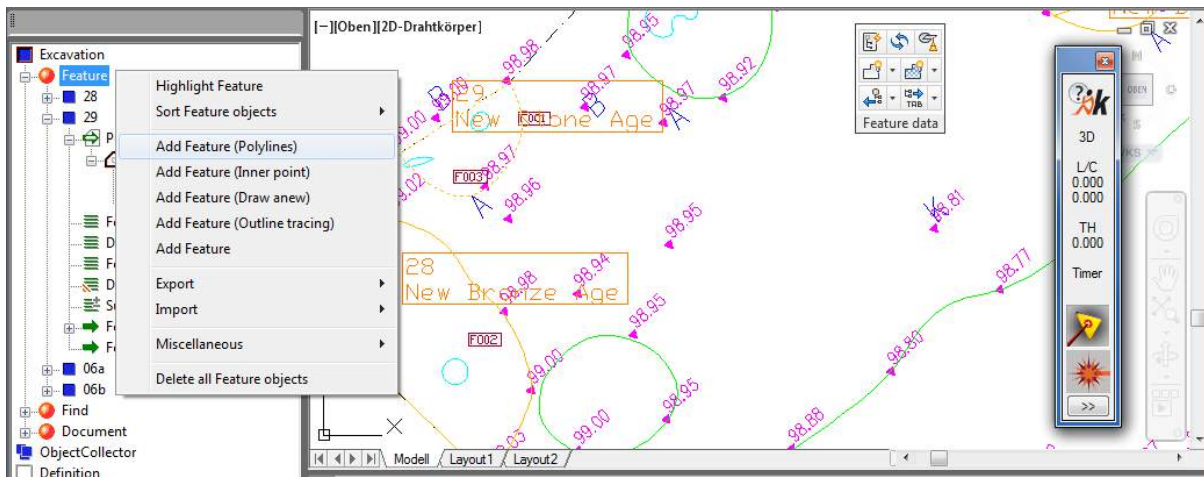
The labels can – just as common AutoCAD elements – be manipulated by four grips. Just try it!




Further interesting information regarding the objects have been saved within the so called **links**. In order to get this information open the *subentry* „■ 28 – "☑ FeatureHasFind". You will see that feature 28 is linked to find F002. If you, vice versa, open "■ F002" and ask for the (backwards) link "☑ FeatureHasFind", feature 28 will be noted there.

### A new feature

The next step will show you how to create a new feature. To do so move to the following location (in the left part of the drawing)



Mark the "  Feature" entry in the structure view and open the context menu by clicking the right mouse button. Select **→Add feature (Polylines)**. The following will be shown in the AutoCAD command line

command: **HYLASCONTEXT**

Polylines that are to be transferred into areas:

Select objects:

Select the green, almost circular polyline. It is the feature boundary of a pit. Confirm your selection by pressing Return. In the command line you will now be prompted for



Insertion point for label:

Click on a point within the green boundary in order to position the label.

A window opens and you will be prompted for the attribute values (feature number, feature type ...) of the new feature. The standard values will be adopted when you confirm with OK. You may also enter your own values. You can scroll through the attributes by using the >> key. Finally confirm your entries by pressing OK.

You will now find a new entry with all its attributes within the structure view.

The "Feature" class has been configured in a way that the surface area will automatically be calculated on basis of the polyline. If there is no polyline for an area (yet) or if the surface area is not relevant for the project, it is possible to generate features otherwise. You will find further information regarding this topic in the As-Built command reference chapter **→Feature data – Context menu commands of the structure view - Add "NAME"**

Attributes may be adapted afterwards. An example: The "Description" for feature 28 is "flat" right now. In order to change this mark entry "  28" – "  Description: flat" and open the context menu by clicking the right mouse button. Select **Change**. Then enter anything in the input window. Confirm your entry by clicking OK.

In order to change the feature mode proceed as described above. You will find the **Change** option within the context menu of the attribute. Select another value from the opening list or enter a new value.

You may proceed as well with all other attributes, except for "SurfaceArea" and "Perimeter". These values are linked to polylines and therefore a manual change is not suggestive.

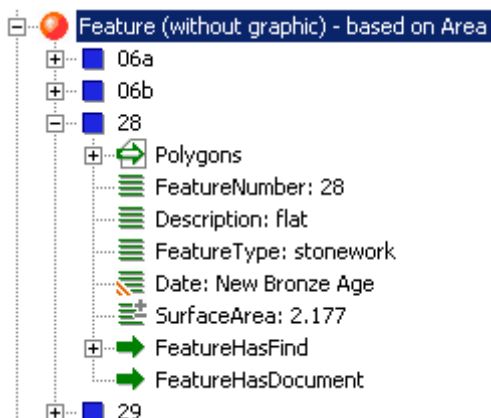
## Links

Now, you will create a find the same way:

Mark the "Find" entry in the structure view. Open the context menu by clicking the right mouse button and select **→Add Find**. Insert the label close to the cyan coloured polyline within the feature. Select the cyan coloured polyline. This line symbolizes the place of finding. Assign a find number, e.g. "F004". Now you will find the new entry "F004" within the structure view.

This find is now to be assigned to the previously created feature – that means it has to be linked. Select "F004" – "FeatureHasFind" from the list. Select "Link with feature" from the context menu. Now select "[FeatureNumberXX]" from the structure view in order to link this find. It will be highlighted. Confirm your selection by pressing the Return key.

You may have noticed that for feature objects there is (besides the attributes) an entry 'Polygons'. This is because features are defined as objects of the type 'Area'. If a feature is created by assigning one or more polylines or by using the method 'Inner point' As-Built automatically generates an entry 'Polygons' for each polyline.



Because of this concept it is possible, that a feature can be made up of several contours and that their surface areas (and perimeters) will be added up. If further polygons shall be assigned to existing features this can be done via the context menu which is accessible by making a right click on the entry 'Polygons'.

For training purposes you may create and edit more features and finds.

## Visualizations

Visualizations may be defined for the individual attribute values. That means, that according to its attribute value a find or feature will be hatched or a symbol will be assigned (AutoCAD block) automatically.

For this example a visualization was defined for the "Date" attribute. Please unfreeze the layer "Date" within the layer list in order to show this. A picture similar to the one below will be created.

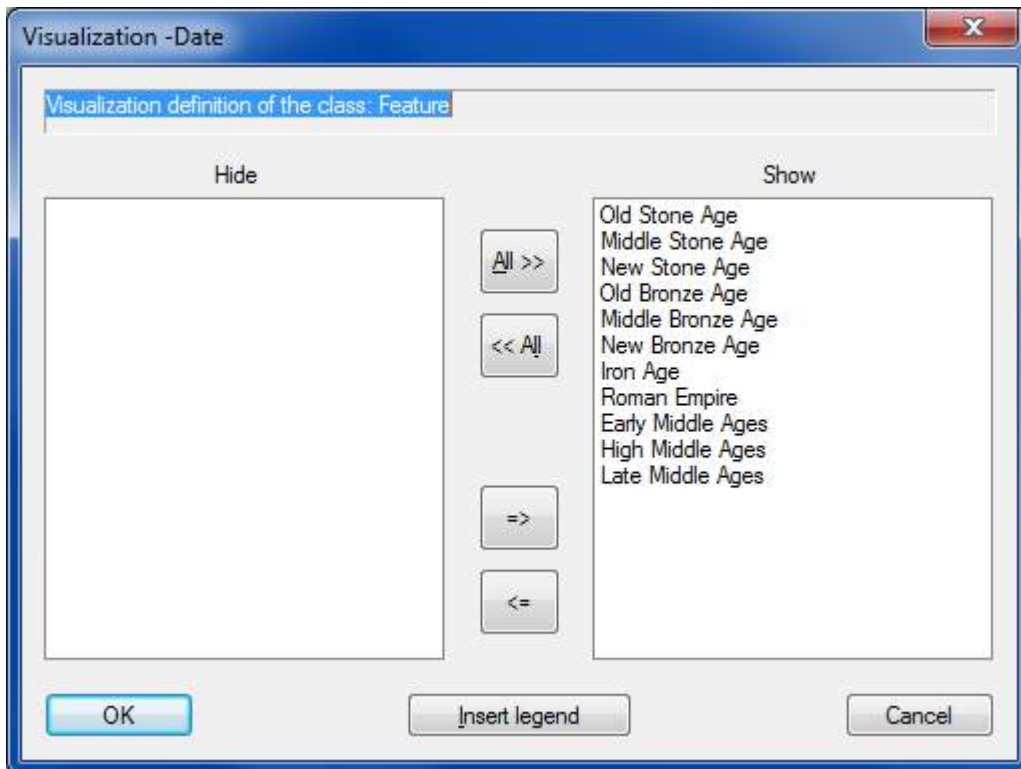


If some area hatchings lie in front you may use the command **→Feature data – Visualization – Update order of display**. Thus, the order of display is arranged in a way the labels and geometry are not covered by hatchings.

You can see, that the closed polylines, assigned to the features, have been provided with colored area fillings. The dark blue filling stands for "New Bronze Age", gray for "New Stone Age", etc. Depending on how many own features you have created and which dating attributes you have entered for these your drawing will have more hatched areas.

Open any feature within the structure view and change the value of the "Date" attribute. You will notice that the area filling of the drawing will automatically be adapted to the new value.

In order to be able to work with visualizations, click on the "Date" attribute. (By the little red triangle within the symbol you will recognize, that there is a visualization for this attribute.) Open the context menu by clicking the right mouse button and select **→Visualization....** The following window will open



The hatchings may be switched off by highlighting the values and clicking the arrow buttons. This makes an easy and layer-independent adaptation of the drawing possible. For example switch off the " New Stone Age " value and confirm with OK. When you take a look at the drawing, you will notice that the corresponding gray hatching is missing.

In order to insert a legend into the drawing open the window described above again. Select "Insert legend" and enter the insertion point within the drawing. Confirm your entry with OK.

That will do as introduction. Visualizations are very flexible concerning the definition of colours, visualization type and layer. Please refer to the As-Built command reference for further information.

### Final remarks

By doing the above, the structure view is filled step by step with object data. A data structure, linked to geometry elements, is created which may be extended or edited at any time.

Via export functions the collected data may be written into tables. This is important in order to be able to further process the data in an external database or spreadsheet.

Refer to the command reference

#### → Feature data – Export tables...

It is also possible to convert the labels into regular AutoCAD blocks, e.g. for the circulation to the end customer. Refer to the command reference

#### → Context menu commands of the structure view – Export - Blocks

If you would like to collect additional data, e.g. the find date, you will have to state this in the structure definition (grey area " Excavation"). The procedure to do so will be explained in the As-Built command reference.

**→ Context menu commands of the structure definition**

If you want to create a completely new structure view you will have to use the command **→ Feature data – Structure – Load structure definition.**



## 9 Export of the feature data

In order to process captured data outside of FARO software – for example in a database or spreadsheet – we offer several export methods:

- Export as Microsoft Excel file ((→Data export - table(s)...) )
- Export of ASCII tables (→Data export - table(s)...) )
- Export as HTML file (→Data export - HTML...) )
- Export as XML file (→Data export - XML...) )
- Export as Shape file ((→Data export - XML...) )

### 9.1 Export of ASCII tables

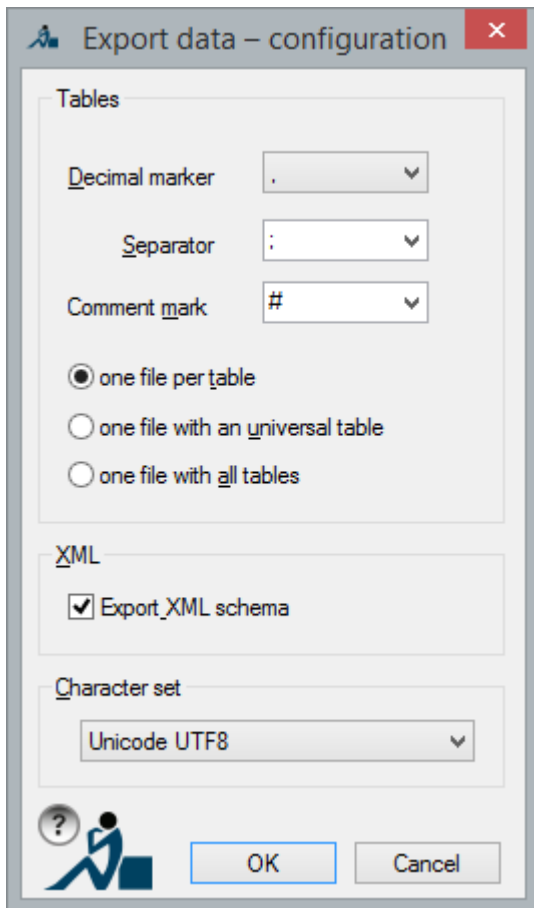
The command **→Data export - Configure export...** allows to determine the format of the exported tables. For further editing of data in Office programs in English speaking countries, the decimal marker needs to be set to “.” (point). Mostly, you will export each table into a separate file.

In the following the export to tables is explained using the example from tutorial 2.

Load the file

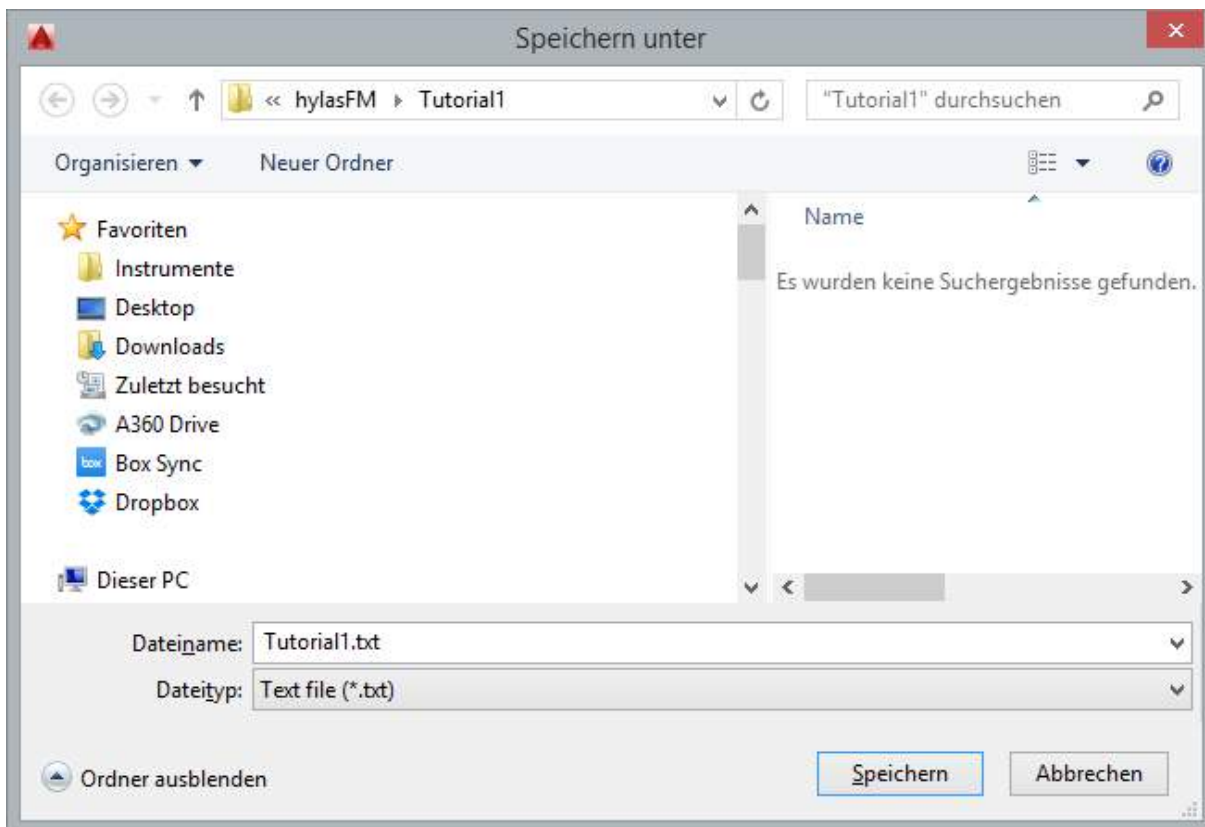
```
..\TUTORIALS\FEATURE_DATA\GENERAL\TUTORIAL2\TUTORIAL2_RESULT.DWG.
```

Select the command **→Data export - Configure export...** from the feature data ribbon. Following dialogue appears



Please set everything as shown above and confirm with OK.

Now run the command → **Data export - table(s)...** Following selection dialogue appears

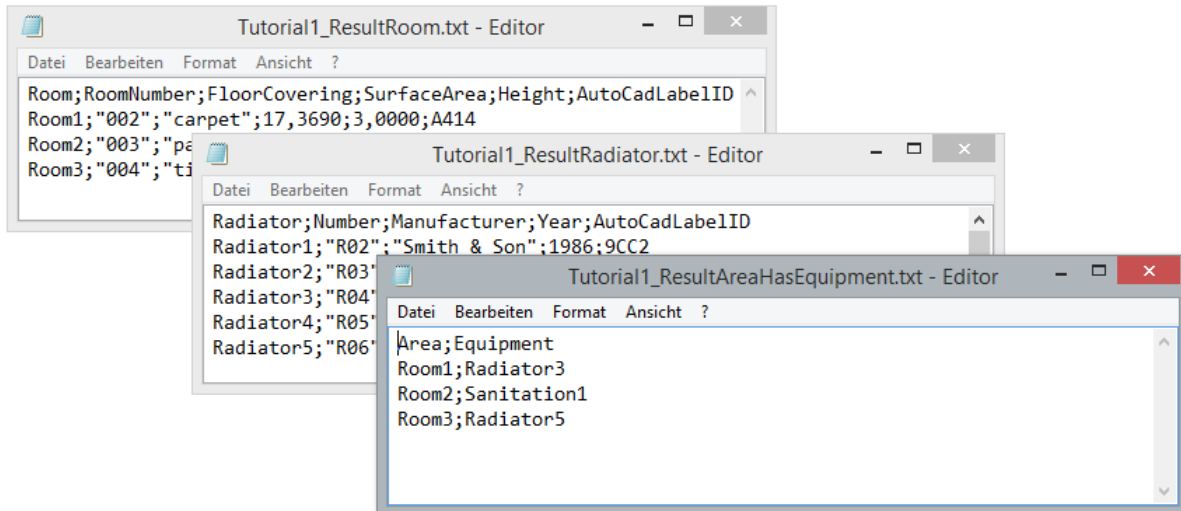


Select a directory (e.g. “FeatureDataExport”) into which the data are to be written and enter the global prefix (e.g. “tut1”) for all files in the field “File name”. Under “file type” you may choose between Microsoft Excel (XLS), text file (TXT) and comma separated format (CSV). Now select “Save” and following files will be created.

File name	Content
TUT1AREA.TXT	{ Classes }
TUT1OPENING.TXT	
TUT1DOOR.TXT	
TUT1WINDOW.TXT	
TUT1SANITATION.TXT	
TUT1ROOM.TXT	
TUT1SUBAREA.TXT	
TUT1EQUIPMENT.TXT	
TUT1RADIATOR.TXT	
TUT1HASEQUIPMENT.TXT	{ Links }
TUT1HASSUBAREA.TXT	

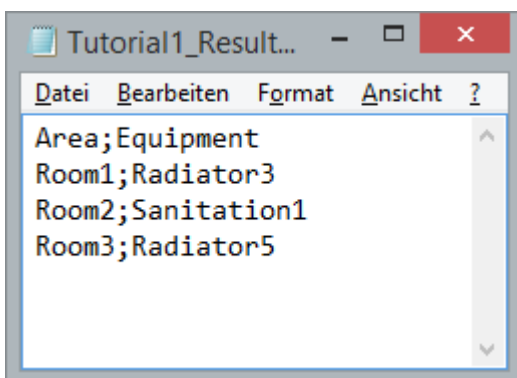
TUT1RAUMHASOPENING.TXT	} Links with AutoCAD graphic
TUT1AUTOCADLINKS.TXT	

Here are three examples from the tables mentioned above:



The first line is always the header. It contains the class names and attribute names. The individual objects, attribute values and links are contained in the following lines. The graphic linked with the feature data object is saved in the column "AutoCadLabelID" as well as in the table "TUT1AUTOCADLINKS.TXT". This information can be used by external AutoCAD based applications (e.g. CAFM systems) to create the connection between the feature data objects and their graphic representations in the CAD drawing. These links are of no concern for non-graphic data.

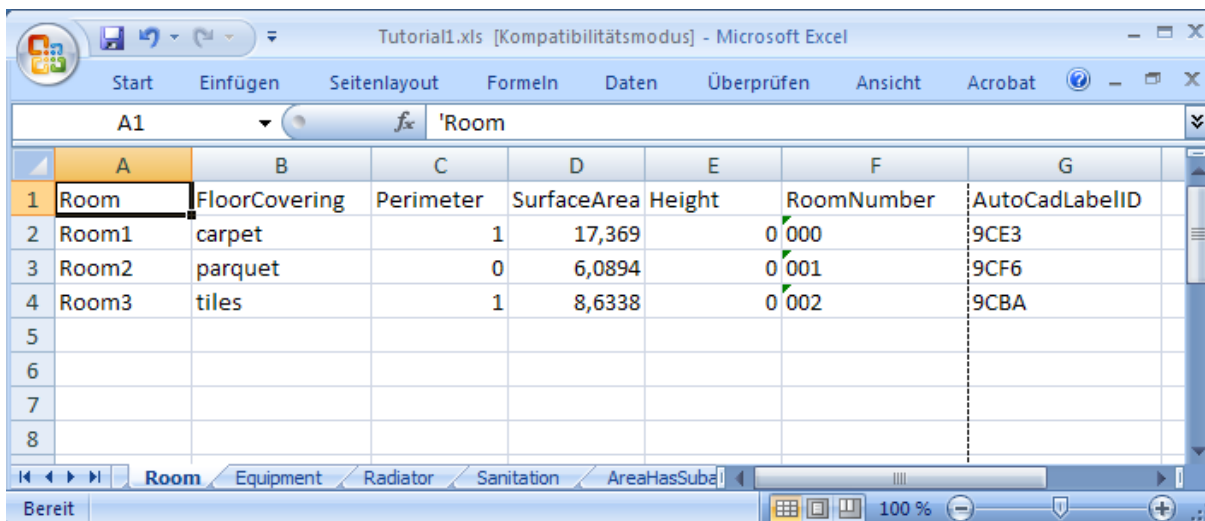
A link file contains a header with both linked classes. Then all concrete links between objects are listed.



Please consider that even objects of different classes can stand in one link file when using inheritance.

The order of the columns of attributes in the exported tables depends on the sorting of the attributes under **→Sort attributes, show/hide....** Hidden attributes will not be exported.

In case you export to Excel, you will get a single Excel file where the individual tables have been separated on different work sheets.



## 9.2 Exporting shape files

The shape file format from ESRI is widely used in the field of GIS as the data exchange format, as it displays a relatively simple and in relation to the data quality not very sophisticated format.

A shape file is not a single file, but is made up of at least three files with the same name:

1. .shp - is used to save the geometric data,
2. .dbf - attribute data in dBASE format,
3. .shx - is the index for linking the geometry with the attribute data.

A shape file is generated for each class. The name of the shape file is made up of a freely selectible prefix and the class name. After starting the command the name prefix is given in the “Save As...” dialogue box that is then displayed.

## 9.3 Export of XML

XML is *the* universal data exchange format on the internet. XML can write and display all structured documents. Our software offers to export the automatically and manually captured data in this format.

### XML Schema Definition (XSD)

Two files are created when exporting to XML

1. one XML file with the feature data (extension XML)
2. one description of the structure of the XML file as XML schema definition (extension XSD).

The XDS file (short: schema) describes the structure of the XML file and defines the used element or attribute data types. By means of the schema the validity of the XML document can be checked.

Following table gives an overview over the mapping of the feature data structure definition in an XML schema. However, XML knowledge is required to understand it!

Fixed element and	Mapping of FARO feature data concepts on XML element types
-------------------	--

element type definitions (independent of concrete feature data structure definition)	Object ID	Element <DcId> of type "DcIdType"
	Reference to an (existing!) Object ID (for links)	Elements <First> and <Last> of type "DcIdRefType"
	Base type of all objects <ul style="list-style-type: none"> <li>Object ID</li> <li>AutoCADLabel ID</li> <li>linked graphic</li> </ul>	"DcObjectType" with elements <ul style="list-style-type: none"> <li>&lt;DcId&gt;</li> <li>&lt;CadAnker&gt;</li> <li>&lt;CadGrafik&gt;(List of &lt;CadId&gt;)</li> </ul>
	Links (1:n mapping)	Element <Related> of type "DcRelationType" with the elements <ul style="list-style-type: none"> <li>&lt;First&gt;</li> <li>&lt;Lasts&gt; (List of &lt;Last&gt;)</li> </ul>
	AutoCAD ID's	Elements <CadId> and <CadAnker> of type "CadIdType"
	Mapping of feature attribute types	
	double	xs:double
	int	xs:long
	ENUM	xs:string (limited)
	STRING	xs:string
	KEY	xs:ID
	Element and element type definitions that depend on the feature data structure definition	for every class "NAME"
for every link "NAME"		Element definition <"NAME"Relation>
Structure definition name "NAME"		Root element <"NAME">

**Remarks**

- The concept of inheritance is neither depicted in a XML schema, nor in the XML file.
- The type security of links is not guaranteed.

**XML file**

The FARO feature data structure is written into a XML document the element structure and element data types of which are defined by the schema.

Objects of the class "NAME"	<p><b>Elements &lt;"NAME"&gt;:</b>  are sub-elements of the element &lt;"NAME"Class&gt;  have the element type &lt;"NAME"Type&gt;  have following sub-elements:</p> <ul style="list-style-type: none"> <li>• &lt;DcId&gt;</li> <li>• &lt;CadAnker&gt;</li> <li>• &lt;CadGrafik&gt; (List of &lt;CadId&gt;)</li> <li>• Attributes</li> </ul>
Links	<p><b>Elements&lt;Related&gt;</b>  of type "DcRelationType"</p>

The data structure created in tutorial 2 looks as follows in the MS Internet Explorer:

```

<?xml version="1.0" encoding="ISO-8859-1"?><!--?xml-stylesheet type="text/xsl" href="Tutorial11_Result.xslt"?-->
<RoomData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Tutorial11_Result.xsd">
  <AreaClass>
  </AreaClass>
  <SubareaClass>
  </SubareaClass>
  <EquipmentClass>
  </EquipmentClass>
  <RoomClass>
  <Room>
    <DcId>Room1</DcId>
    <CadAnker>A414</CadAnker>
    <CadGrafik>
      <CadId>NoId</CadId>
    </CadGrafik>
    <Area_Perimeter>18.4885</Area_Perimeter>
    <Area_SurfaceArea>17.3690</Area_SurfaceArea>
    <Area_BoundingBoxLongEdge>5.5543</Area_Bounding
    <Area_BoundingBoxShortEdge>3.1830</Area_Boundi
    <FloorCovering>carpet</FloorCovering>
    <Height>3.0000</Height>
    <RoomNumber>002</RoomNumber>
  </Room>
  <Room>
    <DcId>Room2</DcId>
    <CadAnker>A43F</CadAnker>
    <CadGrafik>
      <CadId>NoId</CadId>
    </CadGrafik>
    <Area_Perimeter>12.1331</Area_Perimeter>
    <Area_SurfaceArea>6.0776</Area_SurfaceArea>
    <Area_BoundingBoxLongEdge>2.0783</Area_BoundingBoxLongEdge>
    <Area_BoundingBoxShortEdge>1.3514</Area_BoundingBoxShortEdge>
    <FloorCovering>parquet</FloorCovering>
    <Height>3.0000</Height>
    <RoomNumber>003</RoomNumber>
  </Room>
  <Room>
    <DcId>Room3</DcId>
    <CadAnker>A454</CadAnker>
    <CadGrafik>
      <CadId>NoId</CadId>
    </CadGrafik>
    <Area_Perimeter>13.0867</Area_Perimeter>
    <Area_SurfaceArea>8.6338</Area_SurfaceArea>
    <Area_BoundingBoxLongEdge>3.6137</Area_BoundingBoxLongEdge>
    <Area_BoundingBoxShortEdge>2.5250</Area_BoundingBoxShortEdge>
  </Room>

```

Header

Comment (Example for application of a style sheet)

Empty elements (classes without objects)

Root element with reference to the corresponding schema

Object of the class Room with following attributes:

- Object ID (<DcId>)
- AutoCAD ID of label (<CadAnker>)
- linked AutoCAD graphic (<CadGrafik>)
- Area\_Perimeter (inherited from base class Area)
- Area\_SurfaceArea (inherited from base class Area)
- Area\_BoundingBoxLongEdge (inherited from base class Area)
- Area\_BoundingBoxShortLongEdge (inherited from base class Area)
- FloorCovering
- Height
- RoomNumber

Other Room objects

```

<RoomHasPolygonsRelation>
  <Related>
    <First>Room1</First>
    <Lasts>
      <Last>RoomPolygon1</Last>
      <Last>RoomPolygon2</Last>
    </Lasts>
  </Related>
  <Related>
    <First>Room2</First>
    <Lasts>
      <Last>RoomPolygon3</Last>
      <Last>RoomPolygon4</Last>
    </Lasts>

```

Link "RoomHasPolygons"

Linked objects



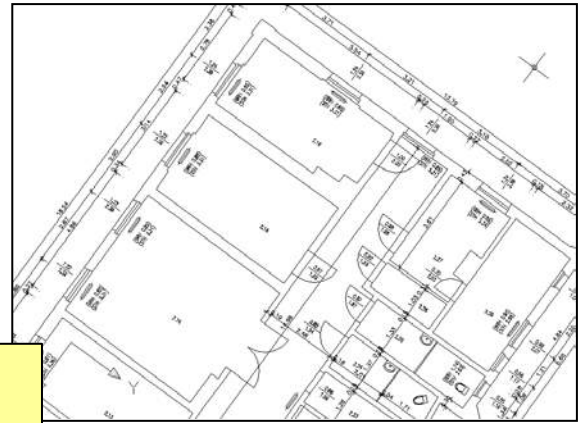
## 10 Data capture for Autodesk Architecture

The FARO feature data management can, in combination with Autodesk Architecture, be used as a tool for modelling 3D components. Special pattern recognition functions support the creation of the following Architecture components in floor plans:

- Window,
- Door,
- Wall,
- Room and
- AEC Areas (as an alternative to room).

### 10.1 Overview of the procedure

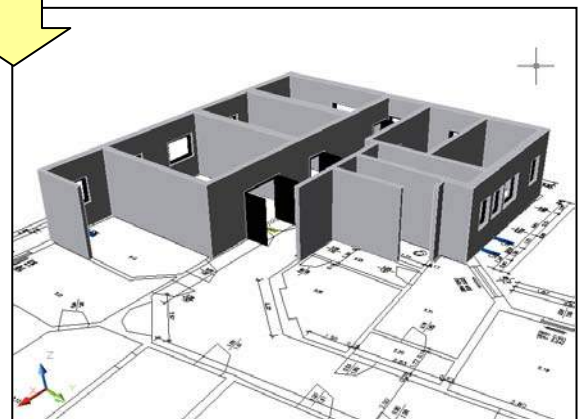
(1) After loading an AutoCAD drawing the featured data structure definition is created or loaded from a template.



(2) Now the above named components will be successively captured as feature data objects and where necessary manually given attribute values. It can be height information that is added or architectural styles can be defined.



(3) Finally with the help of the export function, the architectural components will be created from the FARO feature data objects.

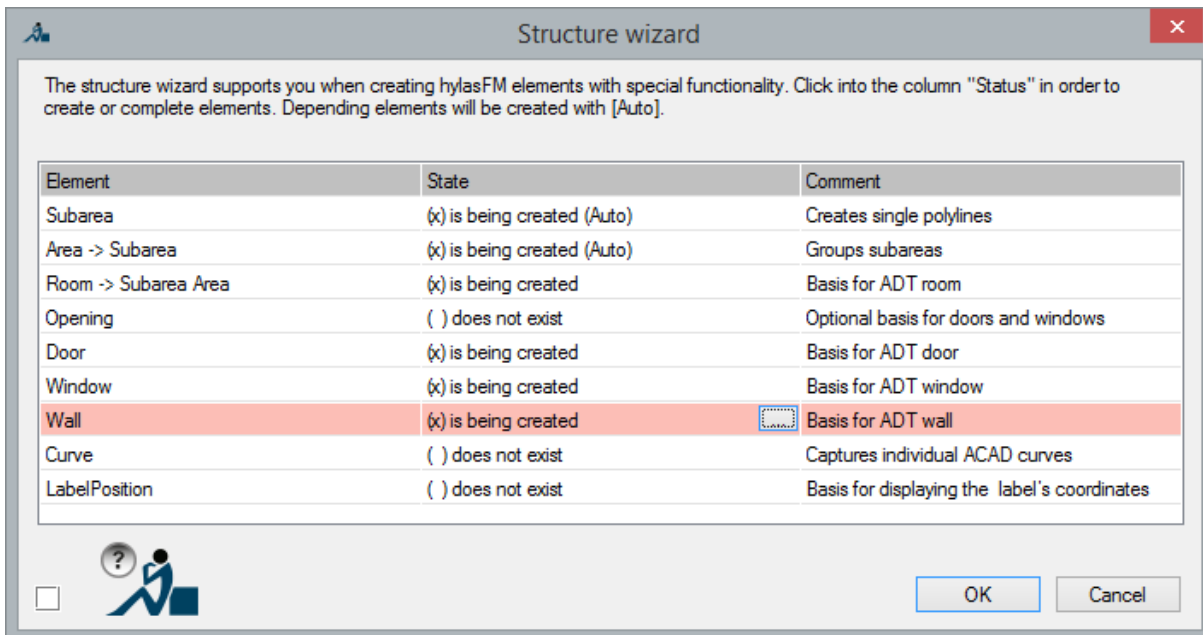


## 10.2 The individual commands

### 10.2.1 Preparation

After loading any AutoCAD floor plan drawing a feature data structure definition is created. To do so use the command → **Create Structure definition...** (See **Command Reference**).

With the help of the Structure Assistant ( → see Command Reference “Structure Assistant” activate the classes room, door, window and wall.



The following attributes are created. The attributes describe the geometry of the objects and are used when creating Architecture objects.

Class	Attribute	Explanation
Window	Width	Width of the window
	Height	Height of the window from breast to lintel
	MeasureType	Decides if the width and height of the window in Architecture should be interpreted as clear openings or construction measurements.
	BreastHeight	
	AdtWindowStyle	If the style exists in the drawing, it will be used when converting to Architecture components.
	StopAlignement	Only relevant for windows with an opening casement. The position of the casement is coded with the help of the feature data label and can take the value “left” or “right”.
	WallThickness	The thickness of the wall at the position of the window. It is only used when exporting windows as Architecture components. Is used when exporting the walls as Architecture components, so this width is ignored and the width of the surrounding wall is used.
Door	Width	Same as for window (threshold height instead of breast height)

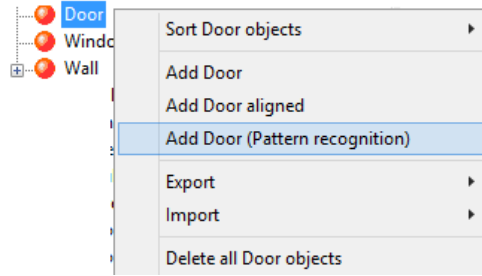
	<b>Height</b> <b>MeasurementType</b> <b>ThresholdHeight</b> <b>AdtDoorStyle</b> <b>StopAlignement</b> <b>Wall thickness</b>	
<b>Wall</b>	<b>Length</b> <b>WallThickness</b> <b>Height</b> <b>AdtWallStyle</b>	
	<b>Alignment</b>	Defines where the axis of the Architecture wall should lie (centred, left or right). Important, if an Architecture style with a fixed wall thickness is used and it varies from the entered FARO feature data wall thickness, but the position of one side of the wall should remain constant.
<b>Room</b>	<b>Height</b> <b>AdtRoomStyle</b>	
	<b>RoomNumber</b>	It is added to the Architecture object as a description

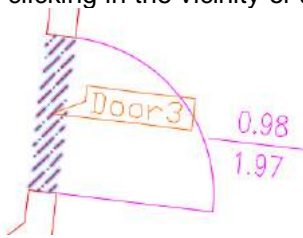
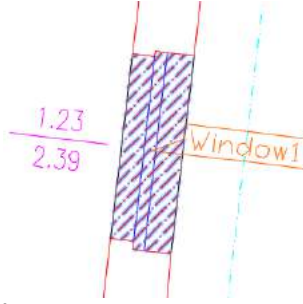
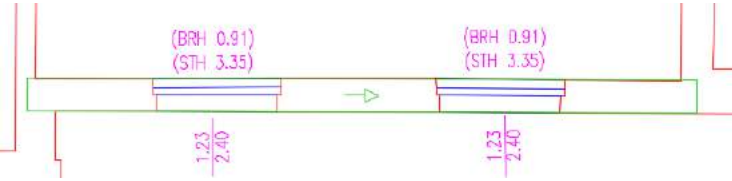
If your drawing contains Architecture styles, it is recommended to note this in the FARO feature data classes, so that later you will be able to easily select it from a list.

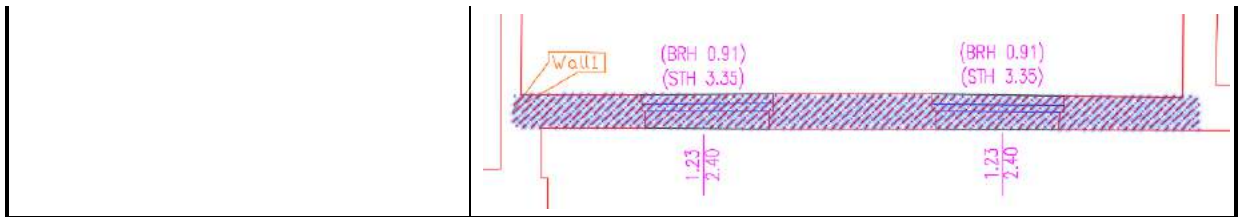
To do this use the command →  **“Load Autodesk Architecture styles”**.

### 10.2.2 Survey

Using the context menu of each particular class you can now with the following commands create doors, windows and walls:



Command	Explanation
Add door (pattern recognition)	<p>Creates doors in the feature data by clicking in the vicinity of a door. Should the command not find a door, you can also place it manually. For that use the context menu function "Insert aligned door".</p> <p>The attributes "Width", "Hinged side" and "Wall thickness" will be automatically filled in.</p> 
Add window (pattern recognition)	<p>Creates windows in the feature data by clicking in the vicinity of a window.</p> <p>Should the command not find anything, you can also place it manually. For that use the context menu function "Add window aligned".</p> <p>The attributes "Width", "StopAlignment" and "Wall thickness" will be automatically filled in.</p> 
Add wall (pattern recognition)	<p>Creates walls in the feature data by picking two points on the estimated wall axis (within the inside of the wall).</p> <p>The software automatically recognises the precise wall form as well as the wall thickness.</p>  <p>The wall thickness can be later changed by picking a point or entering a fixed value.</p> <p>Click point for new width or OK/Centered/alignedLeft/alignedRight/Undo] &lt;OK&gt;:</p> <p>With the options "Centred", "Aligned Left" or "AlignedRight" is the wall axis defined and prompted for the wall thickness.</p> <p>You can thereby define on which side of the wall any possible variances in the wall thickness should be visible (see the dependent attribute "Alignment").</p>



Rooms can be created with a command in the context menu of the structure view in the room class.

The context menu entry reads: “Add room (inner point) (→Command Reference).

In the structure view the created feature data objects can be edited quite normally (e.g. the Architecture styles and heights) (→ Manual).

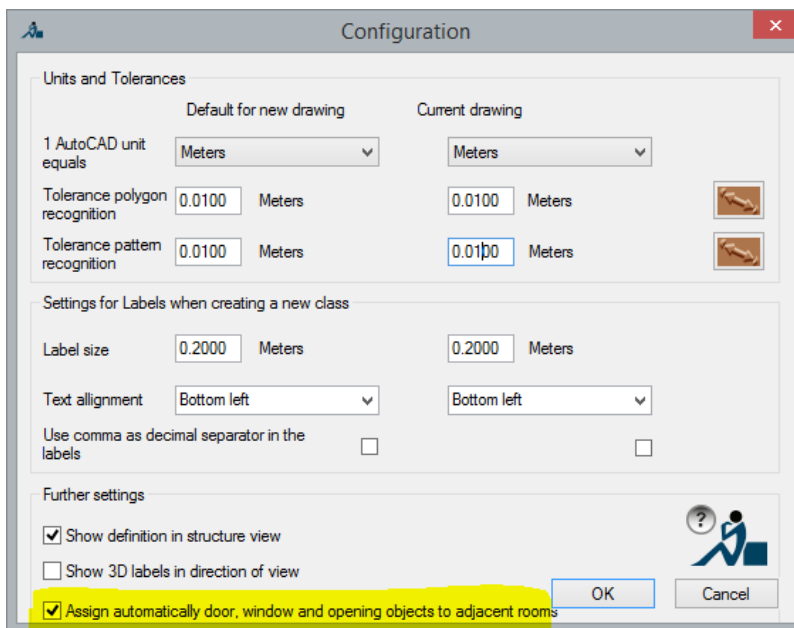
## Links

The feature data management can automatically build links between rooms and/or windows/doors/openings. Thereby doors/openings are automatically associated with both of the rooms that they connect and windows to the room in which they are found.

It is required that the link between the classes where the structure definition is and that the Configuration option

“Assign automatically door, window and opening objects to adjacent rooms”

is switched on.



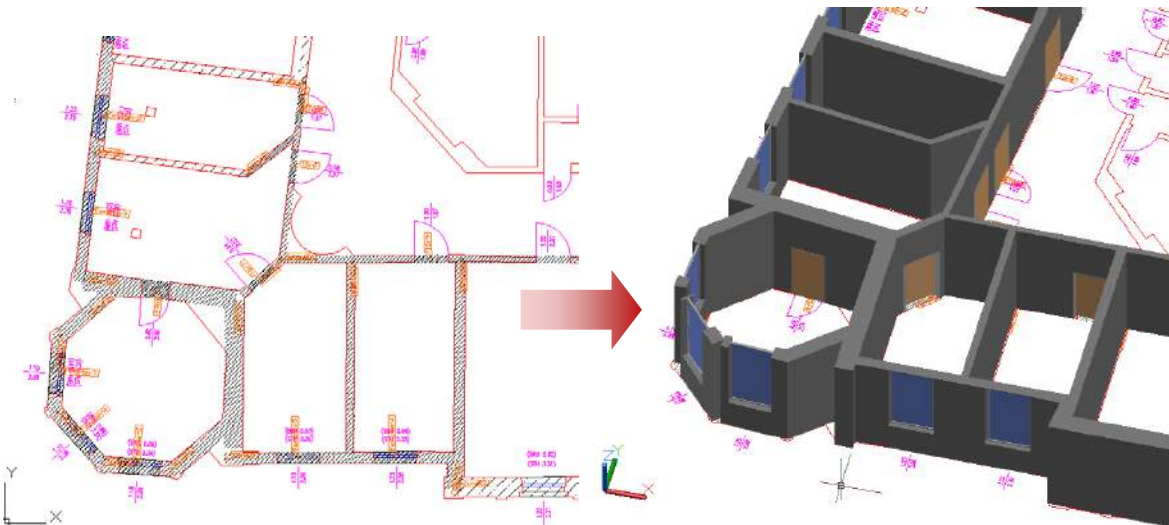
### 10.2.3 Creating Autodesk Architecture components

All of the functions described so far can be used in As-Built for AutoCAD that has been installed in AutoCAD or AutoCAD LT.

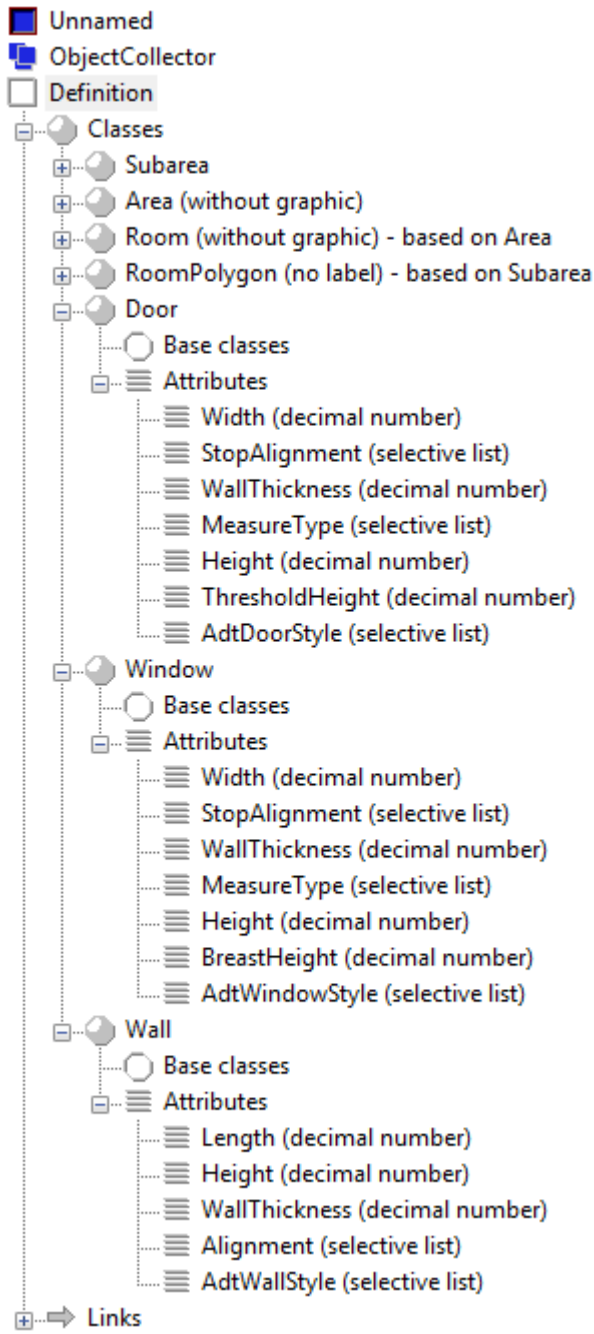
To be able to export to AutoCAD Architecture it is imperative that our software is installed in AutoCAD Architecture.

Command	Explanation
Rooms→Export→Create Architecture rooms	Creates Architecture rooms or AEC areas from FARO feature data rooms (if applicable also with islands).
Walls→Export→Create Architecture walls	Creates Architecture walls from FARO feature data walls. If there are any FARO feature data doors, windows or openings, then they will also be converted into the corresponding Architecture elements. Doors and windows must not then be exported separately.
Doors→Export→Create Architecture doors	Creates individual Architecture windows, which are in short lengths of wall.
Windows→Export→Create Architecture windows	Creates individual Architecture windows, which are in short lengths of wall.

To be able to create Architecture components with these commands the feature attributes such as the information in the particular labels (the alignment) will be used. The CAD graphics will not be considered.



A structure definition can be extended with any of its own classes, attributes and links. Such an extended definition can be of use in other drawings. That is done by using HYLASEXPORT (→Command Reference).







# 11 Command reference feature data management

## 11.1 Commands of the ribbon tab

These are the commands that can be accessed via the menu, the ribbon tab or the command line. The context menus of the structure view will be explained in detail in the section 11.2 and following.

All commands in the command reference are in the same order as in the feature data ribbon tab. The explanations of every command start with a summary in the following form.

### Menu entry

Button	Command:	
	COMMAND	

### Menu entry

You will find the command under this name in the menu.


### Button

You can run the command by clicking in this symbols the ribbon tab.

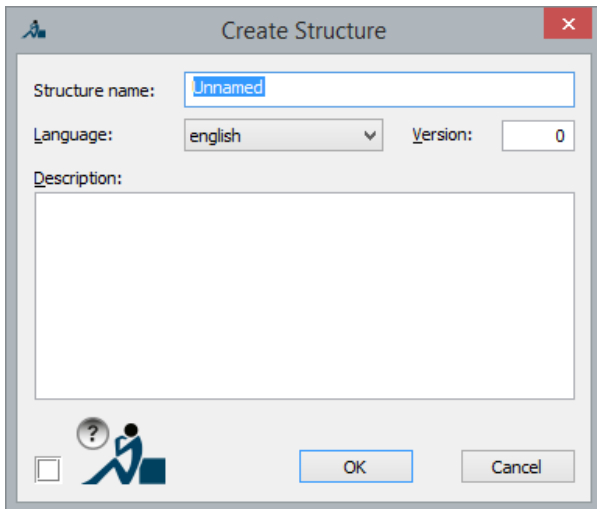
### Command

Here you find the command that you can enter in the command line of AutoCAD.

## Structure → Create structure definition...

	Command:	
	HYSTRUCTURNEW	

The command "hystructurnew" creates a new data structure. Following window opens.



It is advisable to assign a name to a new structure since you will probably add attributes and classes – according to your current project. You will later save this specialized structure and use it for structuring of other drawings. A good name would be one that is not derived from the concrete drawing but from your project. With a meaningful description of the structure you will support the selection of the most suitable structure definition → **Load structure definition** in future projects.

### Language:

By selecting the language, the structure definitions can be recognized in versions with other languages as well. For structure templates, which have been delivered with the English version, "English" has been preset and it should stay like this. If you complement your definition and for example apply a German version, this version will, because of the entry "English", recognize, that it is a structure definition generated with an English version.


### Version

For large and long term projects, the version of the structure definition offers the possibility to keep control over the structure definitions included in the drawings of a project. With the help of the version it is easy to determine if the structure definition of a drawing is up to date.

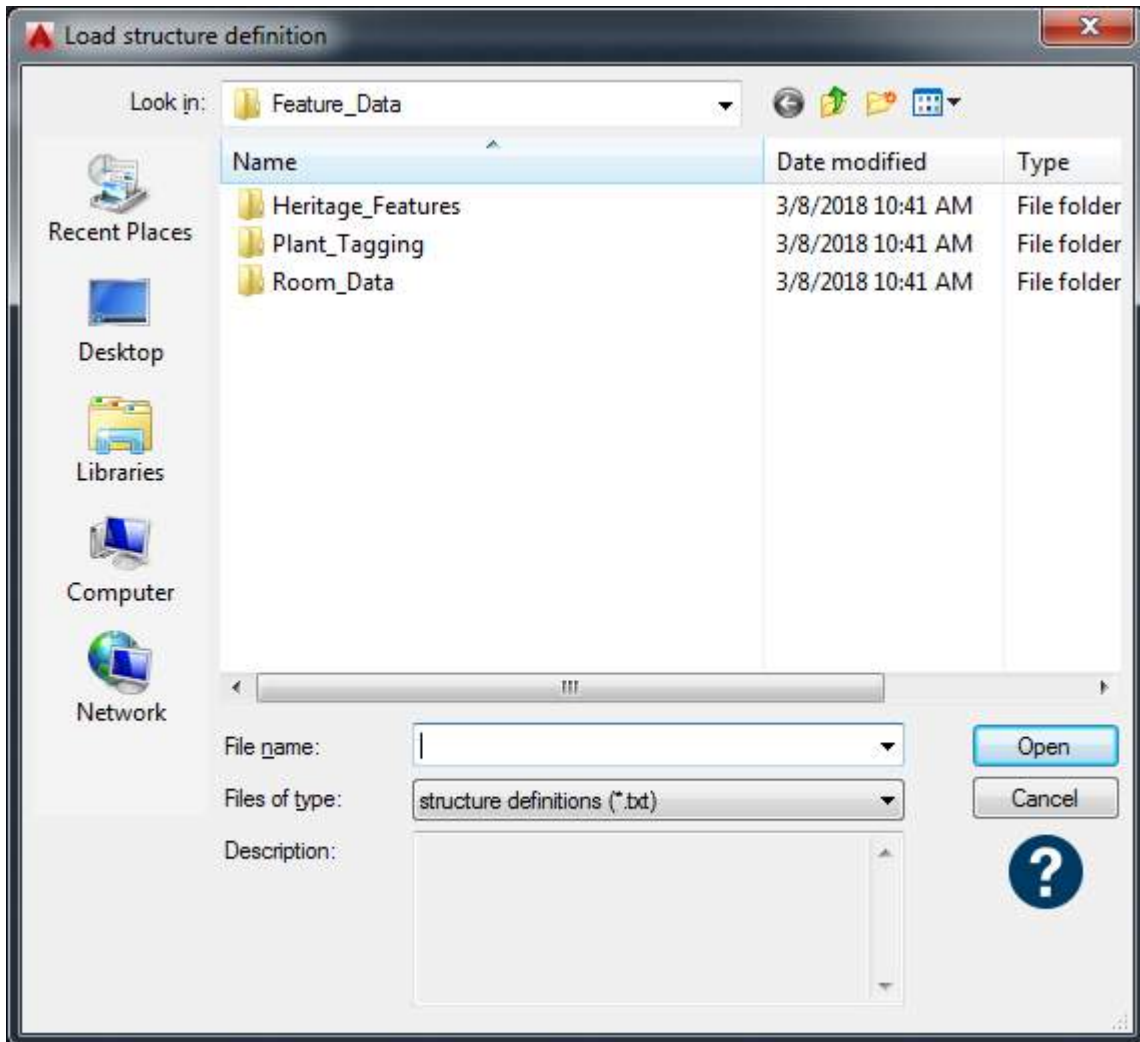
This will also be checked when reloading a structure definition. This way you avoid to load an older version unintentionally.

The wanted version has to be assigned manually as integral number.

**Structure → Load structure definition...**

	Command:	
	HYIMPORT	

The command "HYIMPORT" loads a previously saved structure definition from a file. A file selection dialog opens.



Choose the directory containing the required structure definition templates. If a structure definition file is selected, the description of this structure appears below. After you have selected a template file (that contains a structure definition), it is loaded and the structure view is updated.

**Please note:** When loading a new structure, a data structure that may already exists in this drawing will get lost!

also see:     **→Save structure definition**  
                   **→Create structure definition**

As-Built provides several templates.

Templates for the calculation of room surface areas are saved in the folder ...\\FEATURE\_DATA\\ROOM\_DATA.

For the tagging of components and pipelines in plants, you will find templates under ...\\FEATURE\_DATA\\PLANT\_TAGGING.

Templates for historical building studies and archaeological excavations are stored under ... \\FEATURE\_DATA\\HERITAGE\_FEATURES.

For some of the structure definition templates there are already AutoCAD drawing templates which can be found in the Templates\\DWT folder. These provide, next to the structure definition template, a predefined layer structure.

### **EmptyTemplate.txt**

This structure definition does not contain any classes or attributes. It serves as basis for the generation of entirely new data structures.

### **Excavation.txt**

This structure definition may be used for capturing data about finds and archaeological features. You find it in TEMPLATES\\FEATURE\_DATA\\HERITAGE\_FEATURES.

### **simpleMapping.txt**

This structure definition may be used for a damage and measures mapping of (natural) stone buildings with lower accuracy. It will be used for simple, sectional or stone related mappings.

A class "MappingArea" is used for the classification of parts of the façade or stone contours. Information regarding material, damage, measures and size will be recorded within the attributes.

The class "MappingCurve" may be used for the mapping of line-like structures like cracks for example. Information regarding the length of the line/curve as well as crack width and measure can be found within the attributes. As said before, the attributes are just suggestions. If you record other lines then cracks with the template, you may easily redefine or delete the attributes.

### **detailedMapping.txt**

This structure definition may be used for a damage mapping of (natural) stone buildings with high accuracy, that means for the detailed mapping and the allocation with high position accuracy. There are separate mapping runs for the inventory, damage and measure.

Stone contours are recorded by means of the class "**Stone**".

- There are separate classes for agglomerations and losses. These features are displayed by means of closed ploylines.
- There are three classes for the recording of the different damages. **DamageArea** records laminar damages by means of a contour. **DamageCrack** and **DamageJoint** correspondingly show linear damages.
- The classes **MeasureRestoration** and **MeasureInstauration** are used for the differentiated recording of the necessary measures.

The advantage of many individual classes is, that more detailed information can be captured for every single object through the attributes. An object MeasureRestoration for example has completely different attributes than an object MeasureInstauration. Besides that multiple assignments are possible. Therefore a stone may be linked not only to one but to several measures. The linking of feature, damage and measure objects to the stone objects is done automatically. The link is created if the contours of the objects touch or intersect the outline of the stone or if they lay within the outline of the stone.

### **SurveyOfHistoricalBuildings.txt**


This structure definition has been developed especially for the inventory data recording for (natural) stone buildings. Only the current state is recorded, damages or measures are not taken into consideration.

With the classes stone or plaster different parts of the façade are recorded laminary. The attributes provide further information on material, surface and size.

Punctual characteristics are occurrences on the façade like stone cutter's marks or gripper holes. The attributes provide further information according the shape.

Linear characteristics are line-like appearances on the façade like joints or scratches. Type and length are saved via the attributes.


Paintings are laminar appearances and they describe stone mountings. The era of origin, colour and size are recorded in the attributes.

Structure → Save structure definition...		
	Command:	
	HYEXPORT	

The command "hyexport" saves the current structure definition into a template file.

A window for file selection opens. You can select any name for the template file. The suggested file extension is ".txt" since the template files are readable text files (ASCII). The default name is "*structurename.txt*".

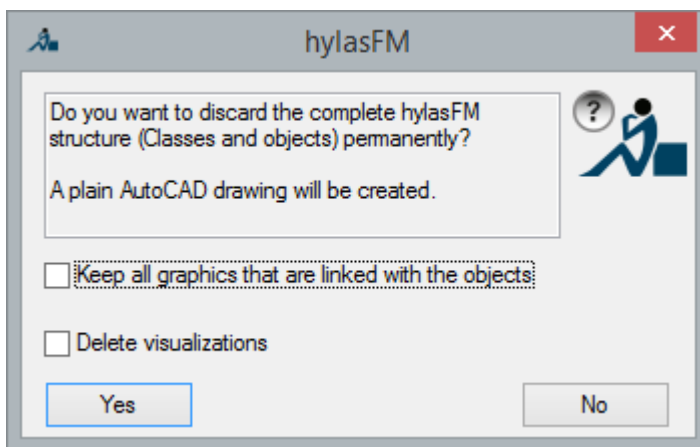
**Structure → Remove structure**

	Command:	
	HYREMOVE	


This command deletes all specific FARO feature data permanently from the current drawing and leaves a pure AutoCAD drawing file.

When running the command a window opens which offers the following options.

- “Keep all graphics linked with the objects” - all AutoCAD drawing elements are kept. If this option is disabled, all objects and their linked graphics will be deleted according to the class configuration settings.
- “Delete visualizations“ deletes displayed visualizations that are defined in the FARO feature data management.



We recommend that you always keep a **backup copy with FARO feature data** even if the final drawing (to be handed to the client) should not contain FARO feature data labels.

Structure → Structure view on	
	Command:
	STRUCTURSHOW

The command "structureshow" displays the structure view.

The structure view is automatically displayed if

1. an AutoCAD drawing that already contains a data structure is loaded,
2. a structure definition is loaded with the command →Load structure definition or is created with the →Create structure definition or
3. the drawing is interpreted with the command →Interpret plan.


If a file does not have a structure, this command has the same effect as →Load structure definition.

If the structure view is already visible, the command "structureshow " has no effect or reduces the branches of the unfolded structure view so only the three main items are visible.

also see: →Structure view off



**Structure → Structure view off**


	Command:	
	STRUCTUREHIDE	

The command "structurehide" hides the structure view.

The object data within the structure view will be maintained, they are just not visible.

also see: → **Structure view on**

**Structure → Load Autodesk Architecture styles...**

	Command:	
	HYLASADTSTYLES	

The command transfers the ADT styles in the current drawing for

- Doors,
- Windows,
- Walls and
- Rooms

into the selection lists of the corresponding attributes of the corresponding FARO feature data class (also see predefined classes → **Structure wizard...**). Already existing entries are kept.

Thus, feature data objects of the predefined classes can be assigned an individual style that is used with the context menu command → **Export** → **Create ADT-...** .

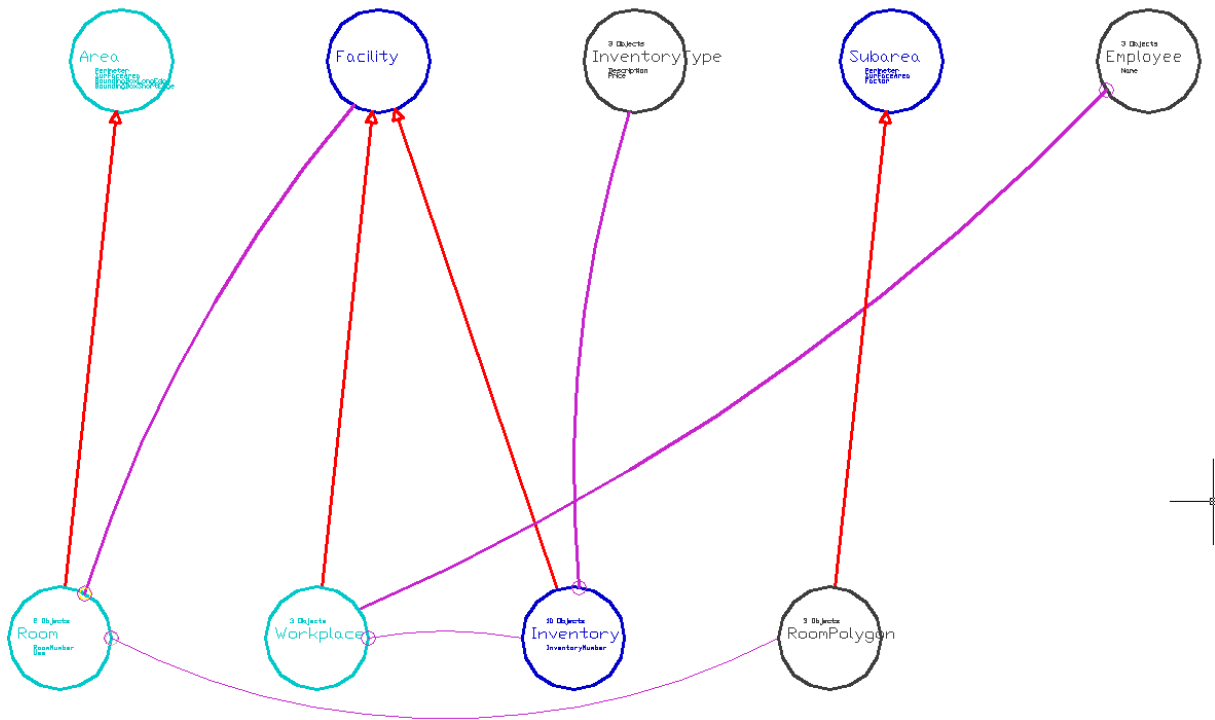
These commands only work if the FARO feature data management has been installed within the Architectural Desktop (ADT) or AutoCAD Architecture.

**Structure → Visualize structure definition...**


	Command:	
	KUBT_HY_SDPLLOT	

With this command the structure definition will be displayed as class diagram within the drawing. The integration point of the diagram can be chosen freely.

In case there are more than 15 classes in the structure definition, an optimizing algorithm will be started. By doing so the display will be much clearer. The algorithm may be aborted at any time with ESC. After you have done this, the result achieved so far will be presented.

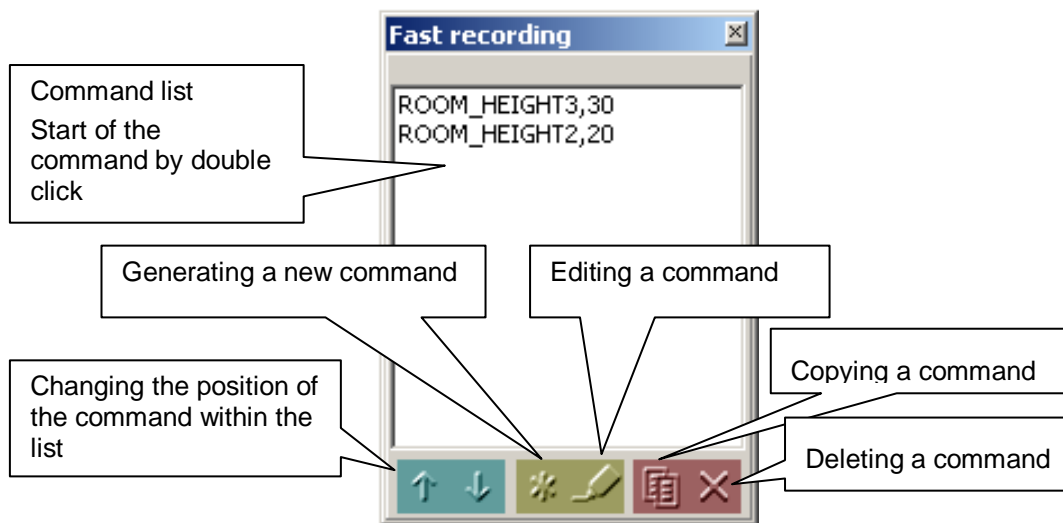


## Data capture - Fast Capture

	Command:	
	HYLASFASSTREC	

It is possible to generate your own command for very often used actions for the generation of feature objects using the fast capture option. By doing so you avoid having to call up the command from the context menu of the structure view. Instead you can start the function by a double click.

With the command "HYLASFASSTREC" a window will open, where the defined fast entry commands are listed. The commands may be relisted, deleted, edited and started.



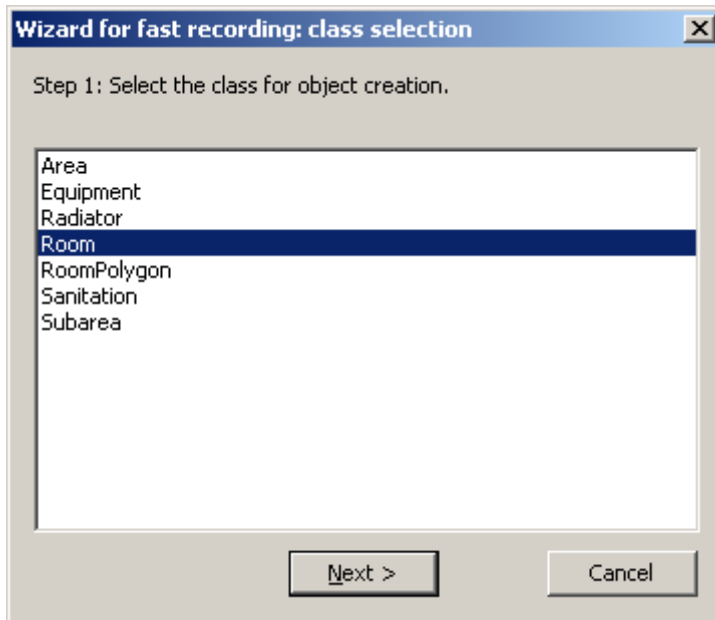
### Executing a fast entry command

In order to start a command from the list double click the command. You may start it alternatively by entering it into the command line. It is also possible to integrate the command as button into the AutoCAD menu. In order to do so us the AutoCAD function for the adaptation of the user interface (command "CUI").

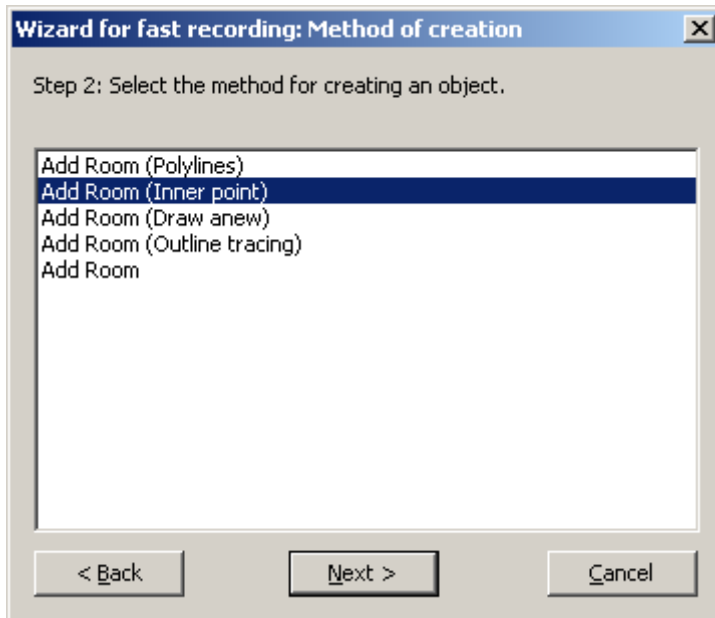
## Generating a new command

When you click the star symbol, the wizard that leads you through the command generation will open.

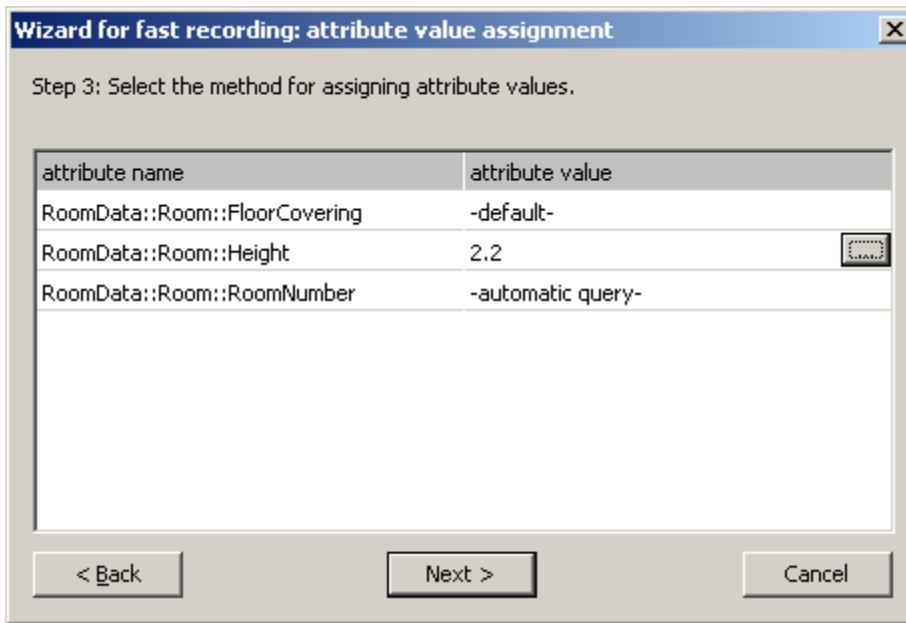
First you will have to select a class.



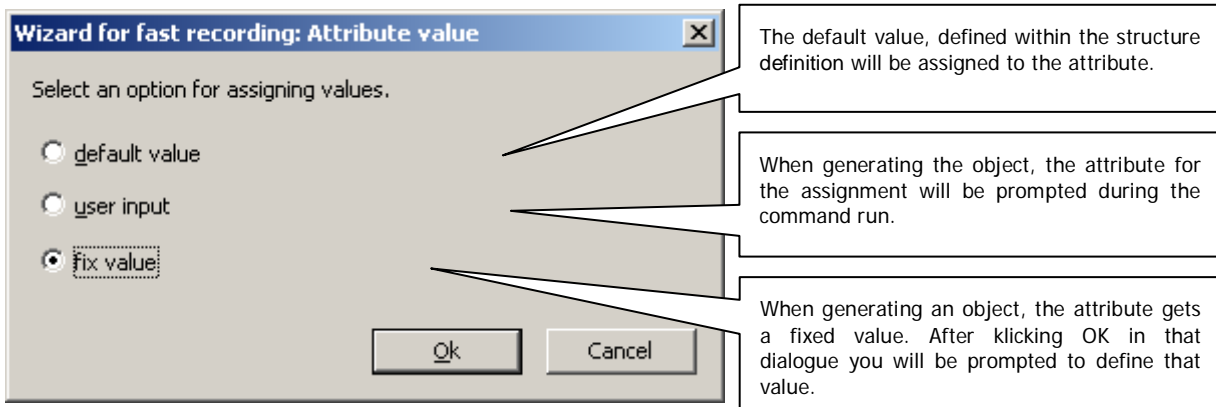
Following that the statement of the generation method is expected.



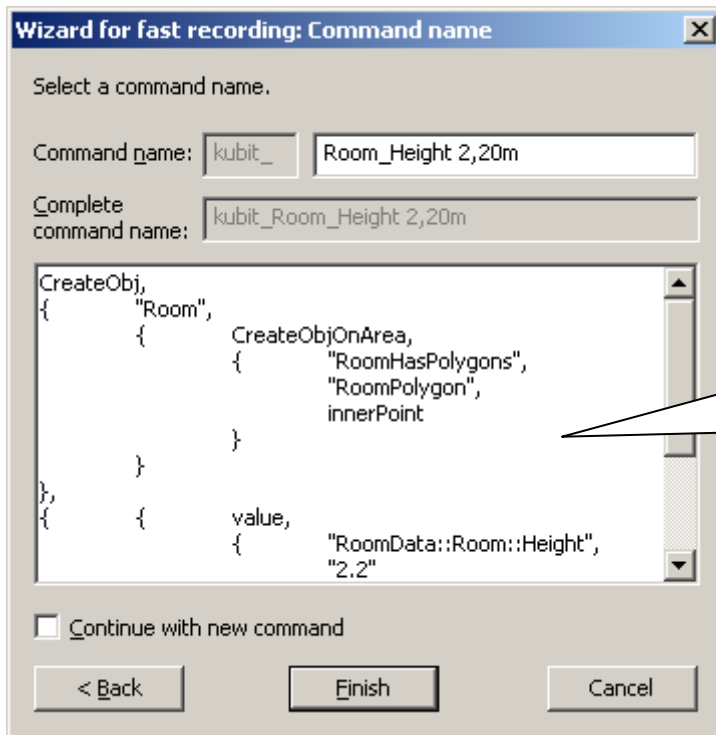
In the following step the attribute assignment will be displayed.



Every single attribute of the class may be predefined by clicking on the entry:



In the last step enter a clear command name:

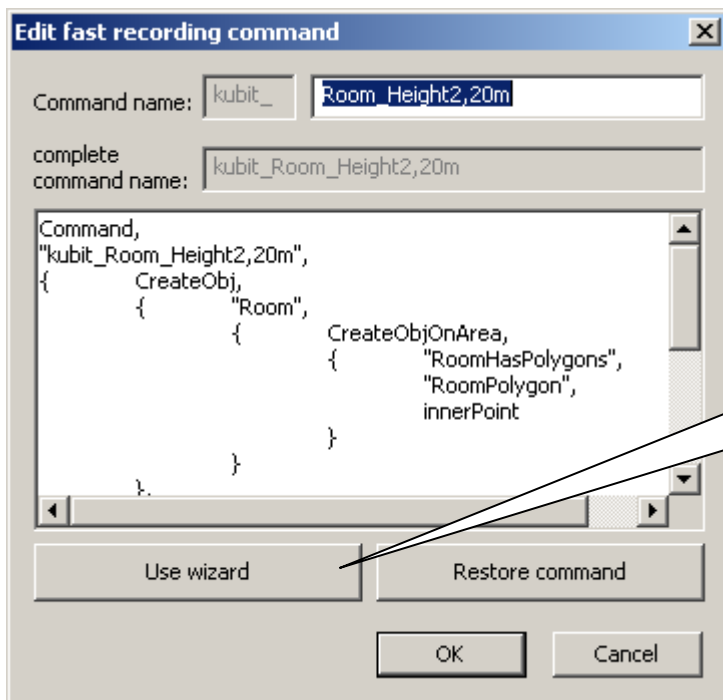


This field shows the command syntax of the generated command. It is only of interest for the experienced user when editing the command

### Editing of a fast entry command



Select the fast entry command to be edited from the list and then click on the pen symbol. You may edit a command with the help of the fast entry wizard or directly within the command syntax window.




Here you can open the wizard and go through the command settings step by step.

The grammar for the generation of a command is as follows:

<b>Symbol</b>	<b>Expression</b>
CreateObjectExpr	<b>Command</b> , Commandname { CreateObj, {Classname, {CreateMethodExpr}}, {AttributExprList } }
CreateMethodExpr	CreateObjectSimpleExpr   CreateObjectOnCurveExpr   CreateObjOnSubareaExpr   CreateObjOnOpeningExpr   CreateObjOnAreaExpr
CreateObjectSimpleExpr	<b>CreateObjSimple</b>
CreateObjectOnCurveExpr	<b>CreateObjOnCurve</b> ,{MethodCurve}
CreateObjOnSubareaExpr	<b>CreateObjOnSubarea</b> ,{MethodSubArea}
CreateObjOnAreaExpr	<b>CreateObjOnArea</b> ,{Relationname, MethodSubarea}
AttributExprList	{{(AttributeExpr, )*, AttributeExpr}}{ }
AttributeExpr	{ <b>query</b> ,{Attributename}}  { <b>value</b> ,{Attributename, Attributevalue}}
SpecialBaseclass	"ID of a feature class with methods"
MethodSubarea	<b>interiorPoint</b>   <b>draw</b>   <b>contour</b>   <b>polyline</b>
MethodCurve	<b>draw</b>   <b>contour</b>
Relationname	"ID of a feature link"
Classname	"ID of a feature class"
Attributename	"global ID of an attribute"
Attributevalue	<b>num</b>   <b>string</b>
Commandname	"Command name" starting with "kubit"



**Data capture - Repeat command**

	Command:	
	HYLASCONTEXT	

This command is used to run the previous command of the context menu of the structure view again. This is very convenient if you want to create several objects of the same class in succession for example.

Alternatively, you can also run the previous command by a right click into the graphic window. Select the top entry of the context menu: "Repeat HYLASCONTEXT".

**Data capture - Purge**


	Command:	
	HYLASPURGE	

The command "HYPURGE" deletes all AutoCAD objects that have temporarily been created by the FARO feature data management. Such objects are created during the Plan analysis commands (→ **Find gaps, Find short/double lines**).

Object marks (dashed lines) that are created when an object is clicked on in the structure view are also deleted.

Possibly existing inconsistencies within the drawing and the structure view are removed as well.

### Data capture - Copy

	Command:	
	HYLASCOPY	

The command “hycopy” allows you to copy feature data objects and regular AutoCAD graphics from one AutoCAD drawing into another. The process is similar to using the Windows or AutoCAD clipboard. To the output in the command line

Select objects to be copied

Select objects: *(Select objects)*

you respond by clicking on the labels of feature data objects and/or entries of the structure view. Thereby, these objects, together with their AutoCAD graphic, are copied to a special clipboard. If you also select regular AutoCAD graphic objects (lines, blocks etc.), these are copied to the clipboard as well.

Then you are asked to indicate a reference point. This point is later used to position objects in the other drawing. If you want to place the objects at exactly the same position in the other drawing, select the option “Origin”.

Modes: Also copy forward linked objects (On), Also copy linked objects without label (On)

Base point[Origin/Linked/WithoutLabel] <Origin>:

At the same time, you can change the modes of the implicit object selection.

Also copy forward-linked objects

In addition to the manually selected feature data objects, all objects that are linked forward ( ➡ ) to these objects will also be selected.  
 Example: If a room with this option is copied from the drawing ...\\FEATURE\_DATA\\GENERAL\\TUTORIAL2\\TUTORIAL2\_RE SULT.DWG, its subareas, its belonging equipment like radiators will also be copied. Whilst if one only selects radiators the linked rooms will not be copied since they are return related ( ← ).


Also copy linked objects without label

In addition to the manually selected objects, all objects that have a link to these objects and **do not** have a label, will be selected.  
 Example: If a piece of inventory is copied from the drawing ...\\FEATURE\_DATA\\GENERAL\\TUTORIAL3\\TUTORIAL3\_RE SULT.DWG, its inventory type will also be copied.

After the indication of the reference point the command is finished. Now you can go to the target drawing and paste the objects ➔**Paste**.

Also see ➔**Cut**.

## Data capture - Cut

	Command:	
	HYLASCUT	

The command "hycut" is used to move both FARO feature data objects and ordinary AutoCAD graphic from one AutoCAD drawing to another. The process is similar to using the Windows or AutoCAD clipboard. To the output in the command line

Select objects to be cut

Select objects: (Select objects)



you respond by clicking on the labels of feature data objects and/or entries in the structure view. Thereby these objects, together with their AutoCAD graphic, are moved to a special clipboard. If you also select normal AutoCAD graphic (lines, blocks etc.), these are copied into the clipboard as well.

Then you are asked to indicate a reference point. This point will later be used to position objects in the other drawing. If you want to place the object at exactly the same position in the other drawing, select the option "Origin"

Modes: Cut forward linked objects (On), cut linked objects without label (On)

Base point [Origin/Linked/WithoutLabel] <Origin>:

At the same time, you can change the modes of the implicit object selection.


Cut forward linked objects	<p>In addition to the manually selected feature data objects, all objects that are linked forward (  ) to these objects will also be selected.</p> <p>Example: If a room with this option is cut from the drawing ...\\FEATURE_DATA\GENERAL\TUTORIAL2\TUTORIAL2_RESULT.DWG, its subareas, its belonging equipment like radiators will also be cut. Whilst if one only selects radiators the linked rooms will not be cut out since they are return related (  ).</p>
Cut linked objects without label	<p>In addition to the manually selected objects, all objects that have a link to these objects and <b>do not</b> possess a label, will be selected.</p> <p>Example: If a piece of inventory is copied from the drawing ...\\FEATURE_DATA\GENERAL\TUTORIAL3\TUTORIAL3_RESULT.DWG, its inventory type will also be cut.</p>

After the indication of the reference point the command is finished. Now you can go to the target drawing and paste the objects → **Paste**.

**Remark:** The settings done in the configuration of a class regarding deleting linked objects do also apply when cutting objects.

Also see → **Copy**.

## Data capture - Paste

	Command:	
	HYLASPASTE	

The command “hypaste” allows you to paste feature data objects and ordinary AutoCAD graphic from the special feature data clipboard into the current drawing. The process is similar to using the Windows clipboard.

The objects must have been transported into the clipboard either with **→Copy** or **→Cut** before. Links that may exist between the objects are created in the target drawing too.

The software expects a reference point when running this command.

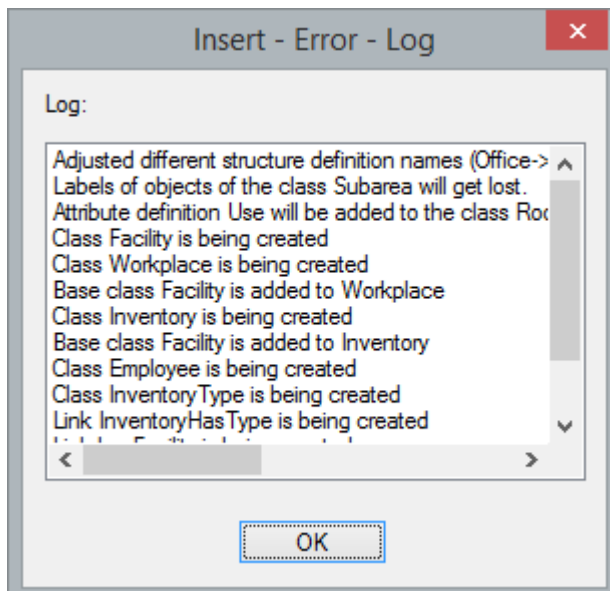
Base point [Origin] <Origin>:

If objects should be placed at exactly the same position as in the original drawing, please select the option “Origin”.

The software is then trying to paste the objects. For this, the structure definition of the selected set of objects must be compatible with the structure of the target file. If necessary, attributes, classes and links are created. Should the need arises, object IDs and Key attributes of the newly pasted objects are also altered so that the unambiguousness remains. If the process was successful, a protocol of the changes is displayed.

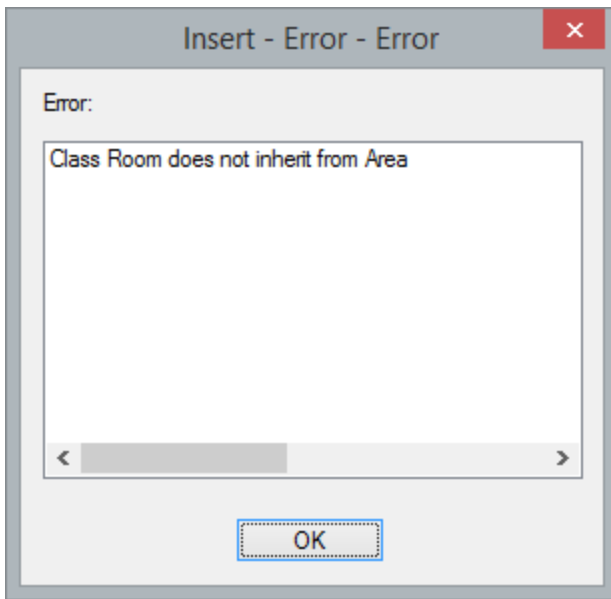
Example: The following protocol is created when copying two rooms and their linked equipment from file

...\FEATURE\_DATA\GENERAL\TUTORIAL2\TUTORIAL2\_RESULT.DWG into a new empty drawing and if the option “Also copy forward linked objects” is enabled:




If no FARO feature data structure exists like in the above example, a minimal data structure is created which can take up the new objects and, if necessary, links that exist between this objects.

If the data structure of the selected objects is incompatible with the structure in the target file, an error report will appear, e.g.:



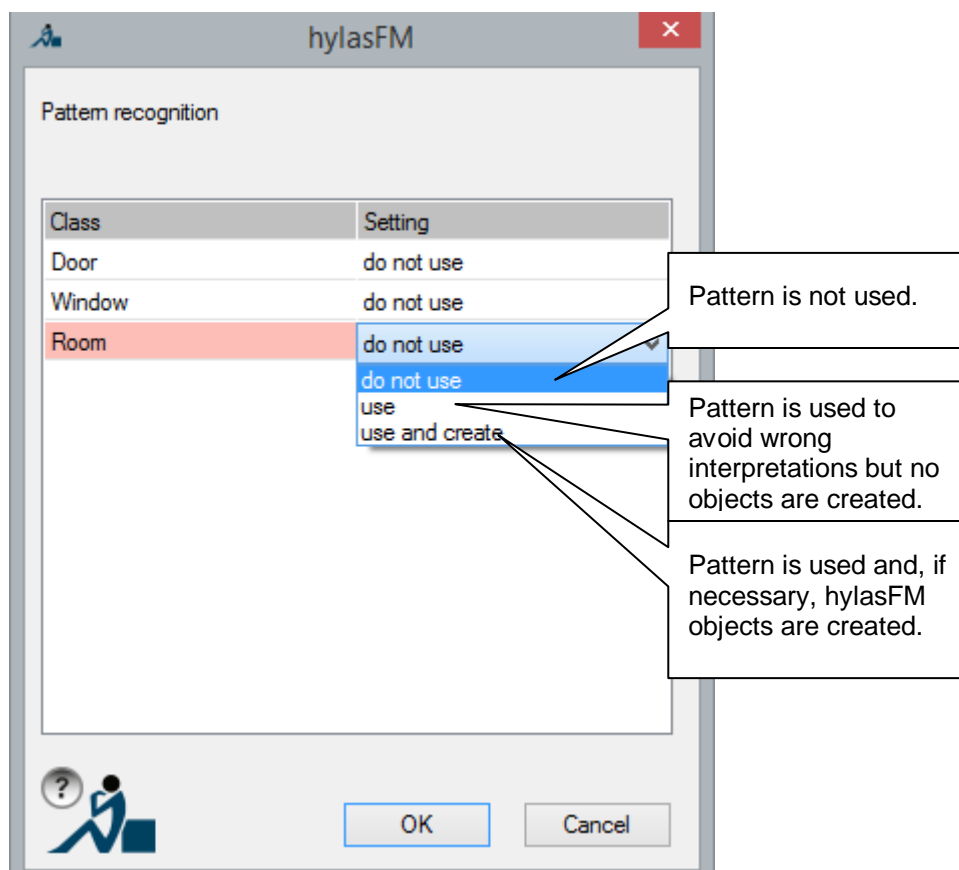
We speak of incompatibility if objects cannot be transferred without losses, e.g. when an object of the class "Radiator" with an integer attribute "Number of ribs"=11 is to be copied into a drawing in which a class "Radiator" with an attribute "Number of ribs" also exists but where the attribute has been defined as selective list,{"10","15","20"}. The structure definition of the target drawing will always only be extended by complete attributes. Attributes will never be changed or deleted.

### Data capture - Interpret plan

	Command:	
	INTERPRET	

The command "interpret" starts the algorithm **Ypsilon** for the automatic interpretation of a complete drawing or a section.

First, a window for configuration opens.



In this dialogue you set which patterns are to be recognized.

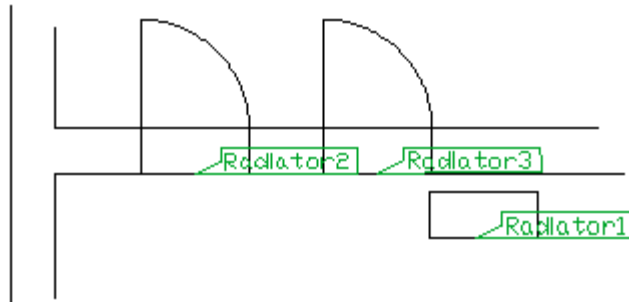
In any case the patterns of the classes door, window and room are offered. If a user defined pattern exists for a class (in the above figure this is the class "Radiator" [see →**Configure class...**, tab Pattern recognition]), this pattern is offered, too.

### Settings

The algorithm **Ypsilon** sometimes creates alternative interpretations. These are evaluated and inferior interpretations are deleted. An example: Let us assume the pattern radiator is a simple square. However, a door in a plan can also contain a square, which represents a threshold. If you want the radiators to be recognized but not the doors, one could set the following:

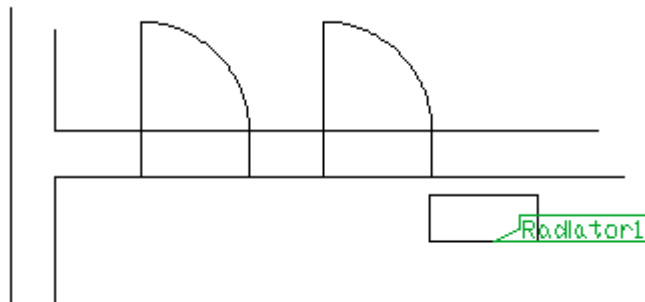
- Door – do not use
- Radiator – use and create

The result would be disappointing



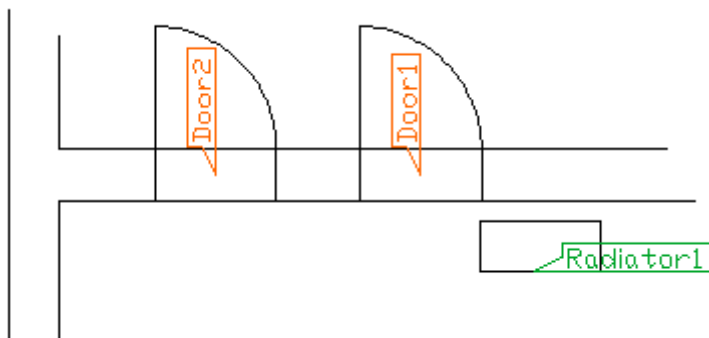
In order to avoid that radiators are created at the doors, the pattern “door” must be made available to **Ypsilon**. Following settings are better.

Door – use  
Radiator – use and create



If both doors and radiators are to be recognized, use

Door – use and create  
Radiator – use and create



After closing the configuration with **OK**, the following appears in the command line.

Interpretation - Object selection for plan recognition

Select objects: *(Select the AutoCAD objects to be interpreted)*

The plan interpretation itself is started after the object selection (e.g. by drawing up a window).

**Ypsilon** runs a complex analysis of the CAD graphic. Therefore, the process may take a couple of seconds (depending on the selected plan section).


You can cancel the command with **ESC**.

If you run the command several times for the same plan section, it can happen that an object is created several times in a plan. However, this only occurs with objects whose labels have been moved manually in the meantime.

The software recognizes the identity of two objects at identical insertion positions of their labels and thus avoids multiple creations.



## Data export - table(s)...

	Command:	
	HYLASEXPORTTXT	

The command "hyexporttxt" saves all data of the structure as tables in one or more files. You can choose between the following formats:

- Textfile (TXT)
- Comma separated file (CSV)
- Microsoft Excel 97-2003 workbook (.XLS)

This function allows you to make the data captured with the feature data management available for further processing in databases, spreadsheets etc.

After running the command a file selection window for the selection of path and name of the export file(s) opens.

The selected file name is put in front of the class name - that has been generated automatically- as a prefix (Example: file name in the dialogue *TUT1.txt* and object table of the class *Room* → File *TUT1Room.txt*).

Depending on the settings made under →**Configure export...**, the tables will be put out in on or more files. Following options are available.

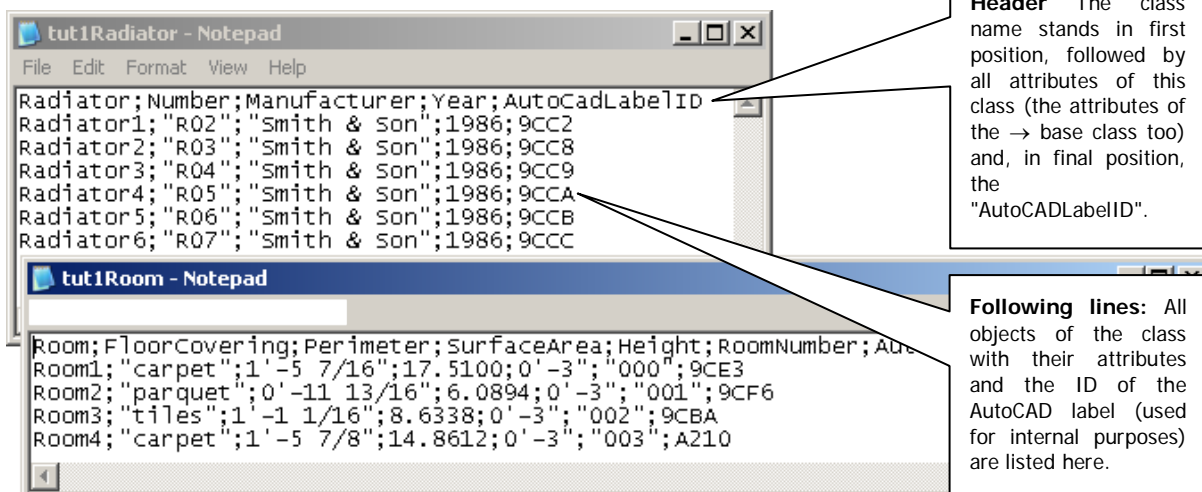
- one file or spreadsheet per table
- one file or spreadsheet with a universal table
- one file or spreadsheet with all tables

### Option "One file per table"

Several files will be generated in the TXT and CSV format. Only one file will be created for the XLS format but the different tables will be separated on the different spreadsheets. The following files (or spreadsheets) are generated:

#### ▪ Object tables

One table (one file) is created for every feature class. It contains all objects of one class with all its attributes.



**Header** The class name stands in first position, followed by all attributes of this class (the attributes of the → base class too) and, in final position, the "AutoCADLabelID".

**Following lines:** All objects of the class with their attributes and the ID of the AutoCAD label (used for internal purposes) are listed here.

```

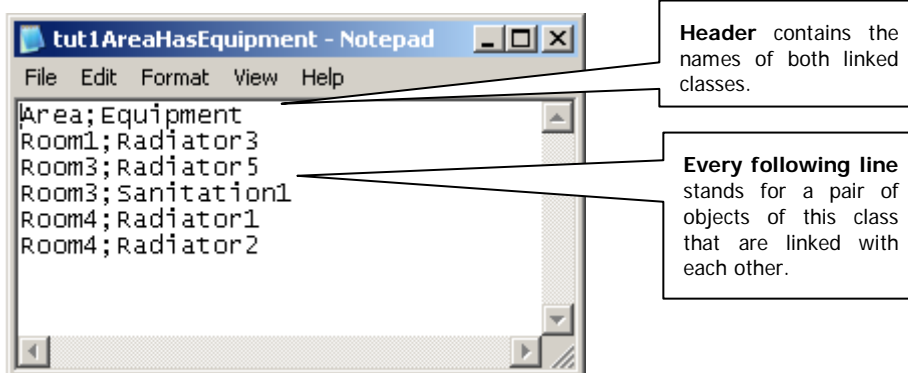
Radiator; Number; Manufacturer; Year; AutoCadLabelID
Radiator1; "R02"; "Smith & Son"; 1986; 9CC2
Radiator2; "R03"; "Smith & Son"; 1986; 9CC8
Radiator3; "R04"; "Smith & Son"; 1986; 9CC9
Radiator4; "R05"; "Smith & Son"; 1986; 9CCA
Radiator5; "R06"; "Smith & Son"; 1986; 9CCB
Radiator6; "R07"; "Smith & Son"; 1986; 9CCC

Room; FloorCovering; Perimeter; SurfaceArea; Height; RoomNumber; AutoCadLabelID
Room1; "carpet"; 1'-5 7/16"; 17.5100; 0'-3"; "000"; 9CE3
Room2; "parquet"; 0'-11 13/16"; 6.0894; 0'-3"; "001"; 9CF6
Room3; "tiles"; 1'-1 1/16"; 8.6338; 0'-3"; "002"; 9CBA
Room4; "carpet"; 1'-5 7/8"; 14.8612; 0'-3"; "003"; A210

```

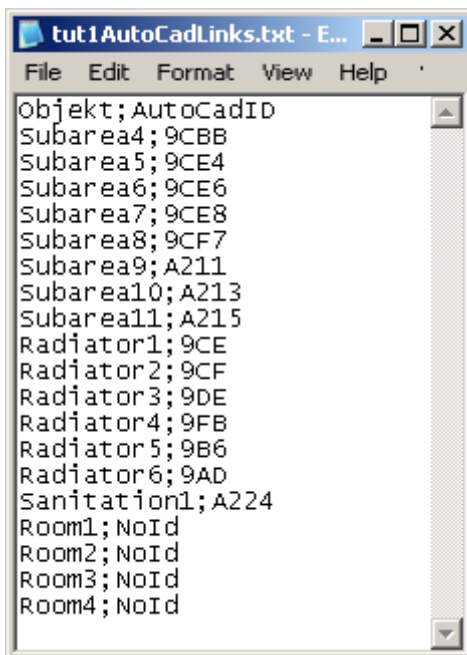
### ▪ Link tables

A table (one file) is created for every link. It contains information about which objects are related via the link.



### ▪ AutoCAD links table

One such file is created per export. It contains the IDs of the graphic, AutoCAD objects that are assigned to FARO feature data objects.



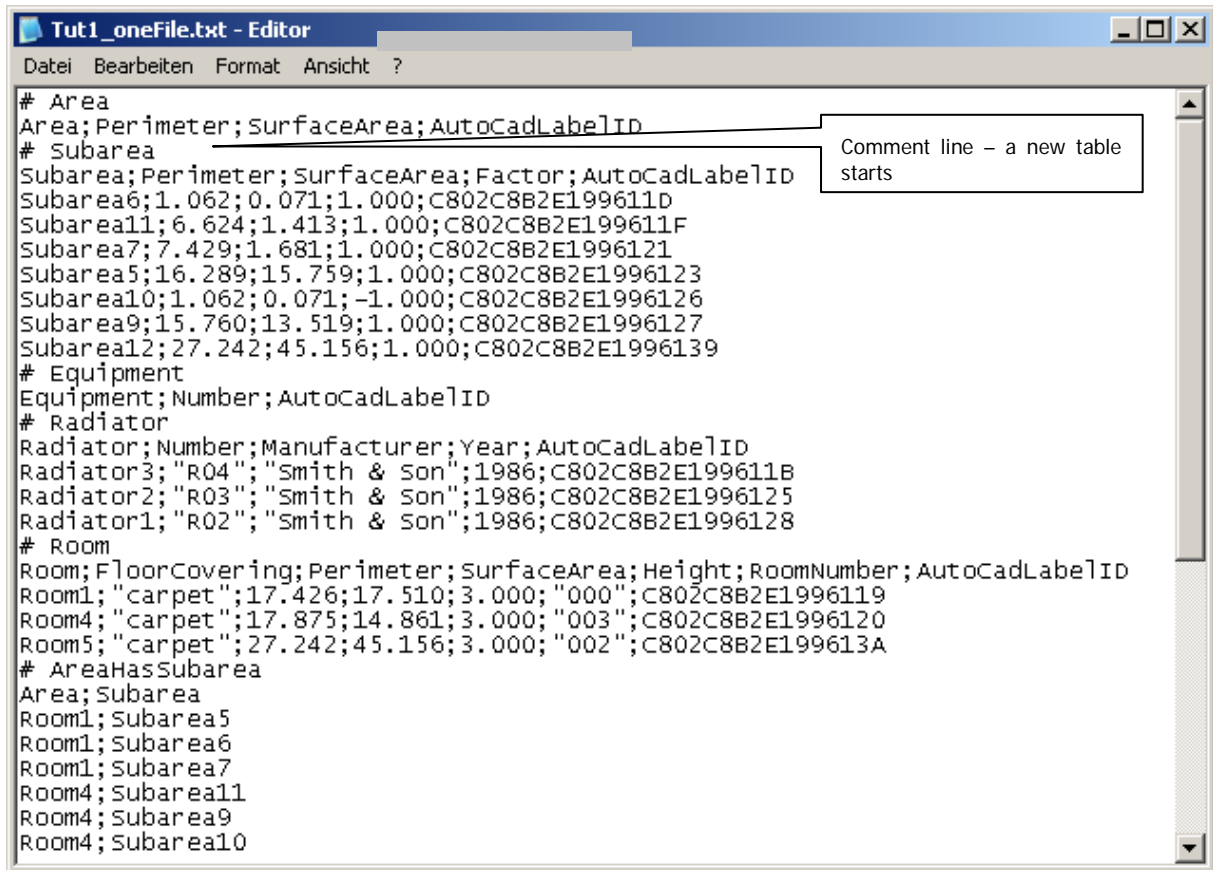
#### Option “One file with a universal table”

Only one file with one large table is created here. It contains all objects of all classes and their links. This particular way of exporting is necessary for forwarding data to certain FM systems (e.g. FAMOS).

#### Option “One file with all tables”

Only one file is created. However, this file contains all tables. The individual tables are separated by a comment line.

The tables correspond to those that have been described under the option “One file per table”.



```

# Area
Area;Perimeter;SurfaceArea;AutoCadLabelID
# Subarea
Subarea;Perimeter;SurfaceArea;Factor;AutoCadLabelID
Subarea6;1.062;0.071;1.000;C802C8B2E199611D
Subarea11;6.624;1.413;1.000;C802C8B2E199611F
Subarea7;7.429;1.681;1.000;C802C8B2E1996121
Subarea5;16.289;15.759;1.000;C802C8B2E1996123
Subarea10;1.062;0.071;-1.000;C802C8B2E1996126
Subarea9;15.760;13.519;1.000;C802C8B2E1996127
Subarea12;27.242;45.156;1.000;C802C8B2E1996139
# Equipment
Equipment;Number;AutoCadLabelID
# Radiator
Radiator;Number;Manufacturer;Year;AutoCadLabelID
Radiator3;"R04";"Smith & Son";1986;C802C8B2E199611B
Radiator2;"R03";"Smith & Son";1986;C802C8B2E1996125
Radiator1;"R02";"Smith & Son";1986;C802C8B2E1996128
# Room
Room;FloorCovering;Perimeter;SurfaceArea;Height;RoomNumber;AutoCadLabelID
Room1;"carpet";17.426;17.510;3.000;"000";C802C8B2E1996119
Room4;"carpet";17.875;14.861;3.000;"003";C802C8B2E1996120
Room5;"carpet";27.242;45.156;3.000;"002";C802C8B2E199613A
# AreaHasSubarea
Area;Subarea
Room1;Subarea5
Room1;Subarea6
Room1;Subarea7
Room4;Subarea11
Room4;Subarea9
Room4;Subarea10

```

Every table has a header with the column headings. The individual entries are separated by separators. The type of separator (comma, semicolon, tabulator) can be configured (→ **Configure export...** ). The order of the columns of the attributes to be exported depends on the order of the attributes in the structure view (→ **Sort attributes, show/hide**). Hidden attributes will not be exported.

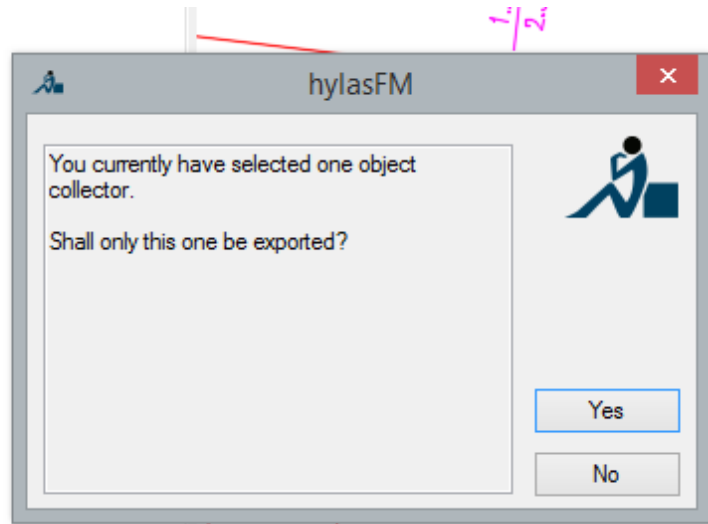
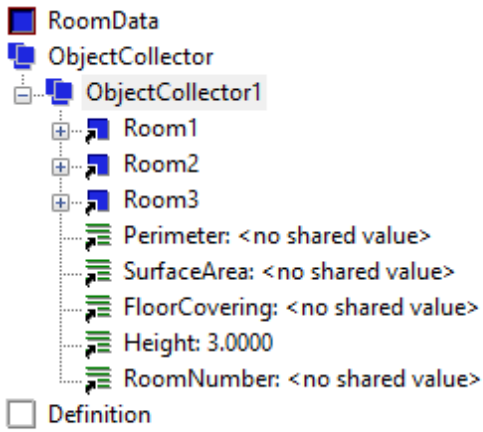
The generated tables can be imported and processed by database applications like MS ACCESS®.

### Export object collector

If you want to export only specific objects with very specific properties (e.g. all objects with an area >100 m<sup>2</sup>) you could proceed as follows:

Create an object collector in which all objects with these properties can be placed.

Highlight the object collector and start HYEXPORTTXT. In the subsequent window you will be prompted if you want to export only the object collector or all of the objects.



## Data export - HTML...

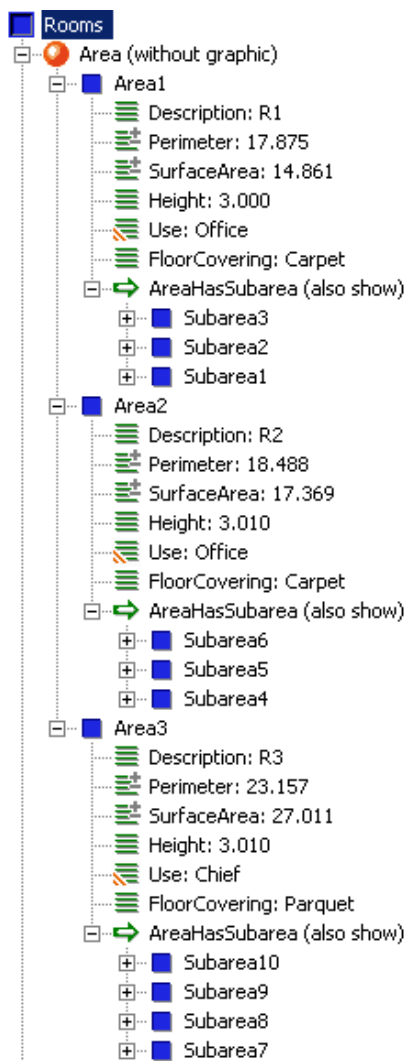
	Command:	
	HYLASEXPORTHTML	

This command saves all data of the structure view in an HTML file that can be read by a web browser.

After the start of the command a dialogue for the selection of the name of the HTML file will open.

In the emerging HTML file all objects are listed in separate tables. Besides the ObjectID all attributes as well as the relations between objects are also listed. Hyperlinks between the objects make the navigation in the file easier.

Here an example of an area list and the resulting exported HTML file:



### Class overview test

#### Area

Object ID	Description	Use	Height	FloorCovering	Perimeter	SurfaceArea	AreaHasSubarea
Area1	R1	Office	3	Carpet	17.875	13.5188963124653 x 1 0.0705186092525555 x -1 1.41282379795 x 1	<ul style="list-style-type: none"> <li>• <a href="#">Subarea1</a></li> <li>• <a href="#">Subarea2</a></li> <li>• <a href="#">Subarea3</a></li> </ul>
Area2	R2	Office	3.01	Carpet	18.488	15.7589238992238 x 1 0.0705186092525555 x -1 1.68060456151957 x 1	<ul style="list-style-type: none"> <li>• <a href="#">Subarea4</a></li> <li>• <a href="#">Subarea5</a></li> <li>• <a href="#">Subarea6</a></li> </ul>
Area3	R3	Chief	3.01	Parquet	23.157	24.7061783637725 x 1 0.0705186092525555 x -1 0.0705186092525555 x -1 2.44567354362157 x 1	<ul style="list-style-type: none"> <li>• <a href="#">Subarea7</a></li> <li>• <a href="#">Subarea8</a></li> <li>• <a href="#">Subarea9</a></li> <li>• <a href="#">Subarea10</a></li> </ul>

#### Subarea

Object ID	Perimeter	SurfaceArea	Factor	AreaHasSubarea(back)
Subarea1	15.76	13.519	1	• <a href="#">Area1</a>
Subarea2	1.062	0.071	-1	• <a href="#">Area1</a>
Subarea3	6.624	1.413	1	• <a href="#">Area1</a>
Subarea4	16.289	15.759	1	• <a href="#">Area2</a>
Subarea5	1.062	0.071	-1	• <a href="#">Area2</a>
Subarea6	7.429	1.681	1	• <a href="#">Area2</a>
Subarea7	20.033	24.706	1	• <a href="#">Area3</a>
Subarea8	1.062	0.071	-1	• <a href="#">Area3</a>
Subarea9	1.062	0.071	-1	• <a href="#">Area3</a>
Subarea10	10.716	2.446	1	• <a href="#">Area3</a>

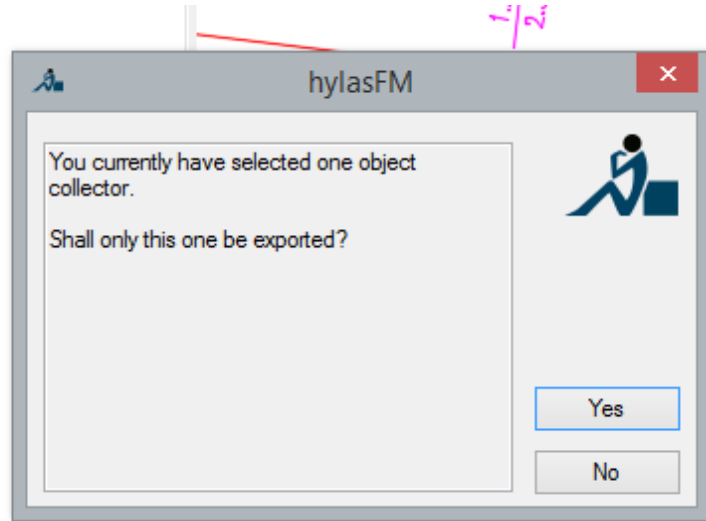
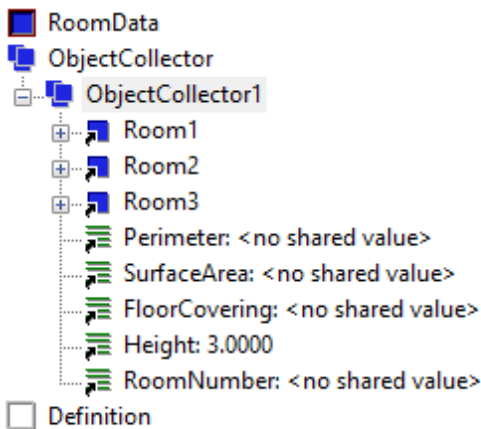


## Export object collector

If you want to export only specific objects with very specific properties (e.g. all objects with an area >100 m2) you could proceed as follows:


Create an object collector in which all objects with these properties can be placed.

Highlight the object collector and start HYEXPORTTXT. In the subsequent window you will be prompted if you want to export only the object collector or all of the objects.



also see: →Export →HTML

**Data export - XML...**

	Command:	
	HYLASEXPORXML	

The command "exportxml" saves all data in an XML file and creates an XML structure definition (XSD file – not to be mistaken with the FARO feature data structure definition).

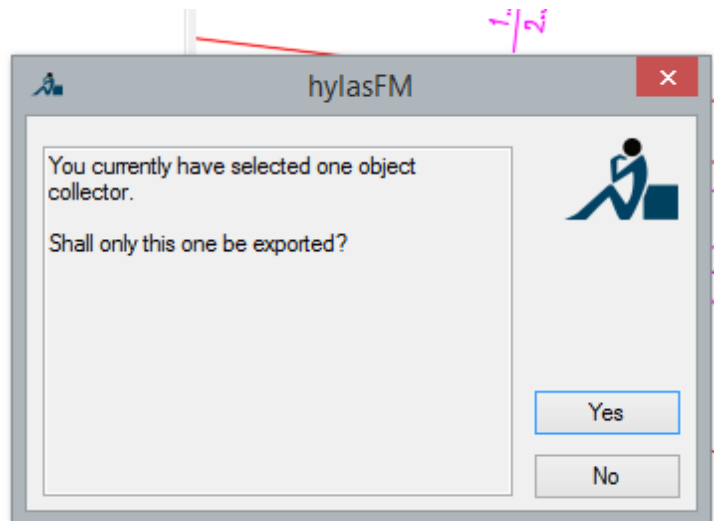
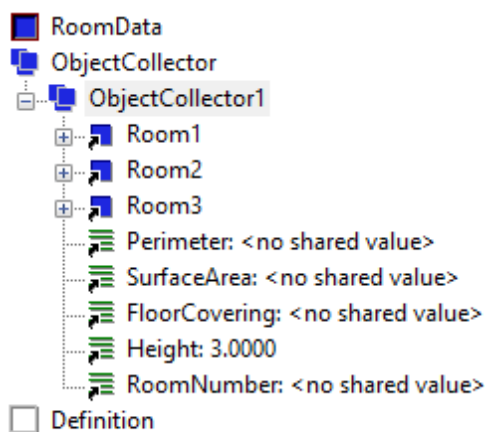
XML was developed as a data exchange format for the internet. XML allows you to present documents of all structures.

**Export object collector**

If you want to export only specific objects with very specific properties (e.g. all objects with an area >100 m2) you could proceed as follows:


Create an object collector in which all objects with these properties can be placed.

Highlight the object collector and start HYEXPORTTXT. In the subsequent window you will be prompted if you want to export only the object collector or all of the objects.



also see: →Data export - Configure export...

**Data export - Shape file**

	Command name:	
	HYEXPORTSHAPE	

The command HYEXPORTSHAPE saves all of the data in the structure in the shape file format from ESRI.

Thereby further processing of the captured feature data is possible in a large number of GIS systems.


After starting the command a file picker dialogue box opens up for selecting the path and name of the export file(s).

The selected file name is used as the prefix for the automatically generated class names (Example: The file name in the selection dialogue box is *TUT1.shp* and the object table of the class is *Room*→File *TUT1Raum.shp*).

If when running the command there is currently an object collector selected in the tree, you will be offered the option of just exporting the objects in the object collector. The objects can belong to different classes.

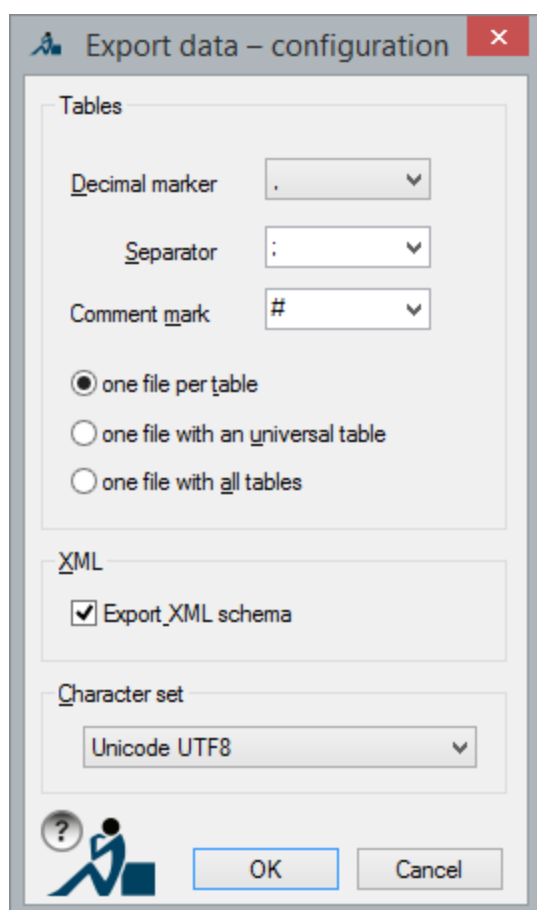


## Data export - Configure export...

	Command:	
	EXPORTCONFIG	

The command "exportconfig" is used to set the different export parameters.

For the export itself use the commands → **Data export - table(s)...**, **Data export - HTML...** and **Data export - XML...**



In the **upper frame "Tables"** you can configure the appearance of the export tables.

- In the list field "**Decimal marker**" you decide for a comma or point as decimal marker (**hint**: If you plan to further edit data in English applications, you should select "." [point]).
- The list field "**Separator**" offers you ";" (semicolon), "," (comma) and "TAB" (tabulator) for selection. However, you can also enter almost any other character or a string of characters.
- A "**Comment mark**" can also be any set of characters. Here too, you have a choice from suggested characters ("#" (hash mark), "\$" (dollar) or "\*" (asterisk) ).

**Note:** Please make sure that all format characters or sets of characters are unique!

Below these list fields you set

- whether you want to write every table into a separate file (many files could be created),


- whether the software should create only one universal table (with all attributes of all classes in every line) or
- a file that contains all tables should be created.

The **lower frame “XML”** contains switches by the help of which you can turn the XML structure definition (also called XML schema, XSD) on and off.

By confirming the settings with **OK** they are adapted for the following export runs and when you (duly) finish the application they are saved permanently.

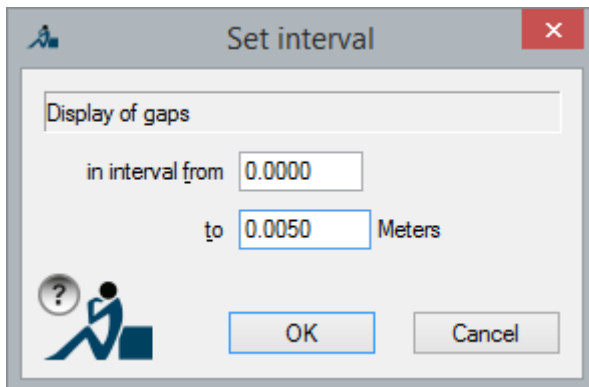
**Cancel** keeps the old settings and finishes the command.

## Plan analysis → Find gaps

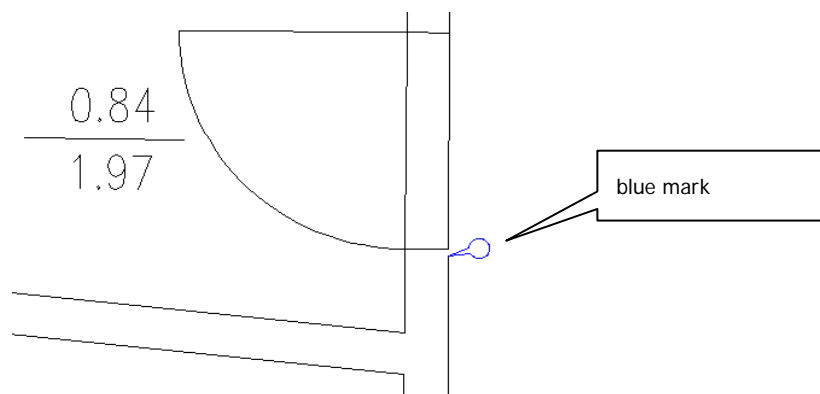
	Command:	
	ANAGAP	

Often there are small gaps between lines that result from a wrong or disabled object snap during drawing. This command allows to find these gaps.

First, this window opens.



Here you enter a length interval. All gaps inside this interval will be displayed. After confirming with OK, you are prompted to indicate the objects to be analyzed. After that the gaps are marked in the drawing by a blue symbol. Now you can correct the errors in the plan.



The size of the blue mark depends on the zoom level.

When running another command for the plan analysis, the marks will be deleted. Alternatively, you can initiate the deleting process manually with → **Data capture - Purge**.

Why does this command require an interval for the gap size and not a maximal gap size?

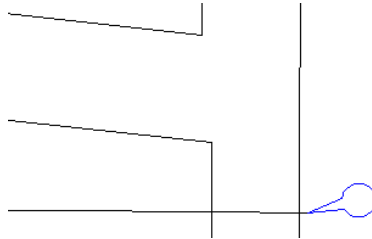
The answer results from the two reasons that could make you search for gaps in the plan.

1. You simply want a clean plan. Then you will indeed set the lower limit of the search interval to 0.
2. You want to create polygons with the feature data management (e.g. for rooms) or you want to use the automatic → **Interpret plan**. All gaps that are smaller than a certain value will be skipped with the procedures for the outline tracing and

pattern recognition. You can set this tolerance yourself with the command **→Configuration**. Thus, you do not have to correct errors in the plan below the tolerance – at least for the processing in the feature data management.


Therefore, you can set the lower interval limit of the gap search to the tolerance value and thus only display and correct errors that are larger than the tolerance.

The gap search also finds small errors as in the following section:

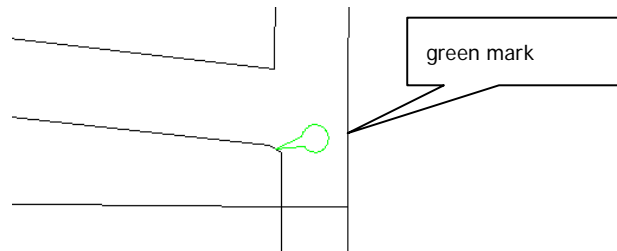


The gap search runs a complex analysis of the CAD graphic. Therefore, the process may take a couple of seconds (depending on the selected plan section).

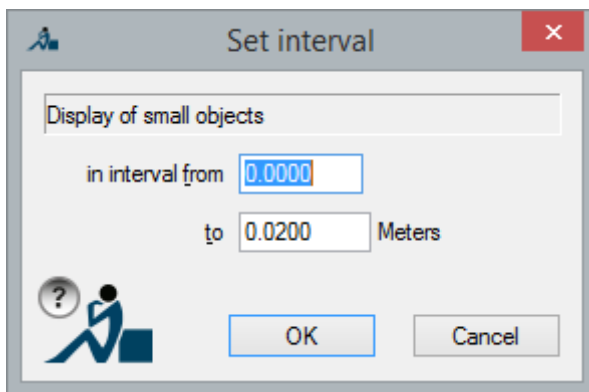
**Plan analysis → Find short lines**

	Command:	
	ANASHORT	

This command allows you to find short lines. These will be marked with a green mark.



The length of the lines to be marked is prompted in the following dialogue.




The size of the green symbol depends on the set zoom level.

When running another command for the plan analysis, the marks will be deleted. Alternatively, you can initiate the deleting process manually with → **Data capture - Purge**.

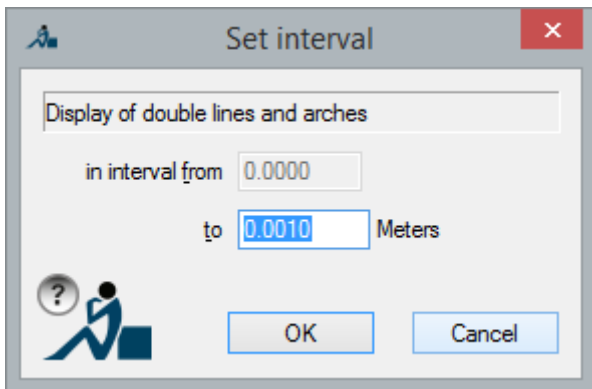
The search for short lines function runs a complex analysis of the CAD graphic. Therefore, the process may take a couple of seconds (depending on the selected plan section).

## Plan analysis → Find double lines

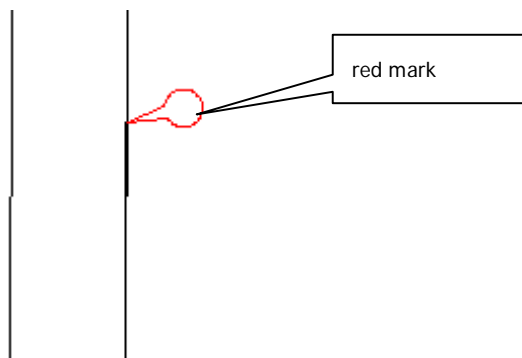
	Command:	
	ANADDOUBLE	

Sometimes plans contain double lines, which this command can detect. They will be marked with a red mark.

First, following window opens.



Here you enter a length interval. All lines that have a maximum distance within this interval will be displayed. After confirming with OK, you are prompted to indicate the objects to be analyzed. After that the gaps are marked with a red symbol. Then you can correct the errors in the plan.




The size of the red symbol depends on the set zoom level.

When running another command for the plan analysis, the marks will be deleted. Alternatively, you can initiate the deleting process manually with → **Data** capture - Purge.

The search for double lines function runs a complex analysis of the CAD graphic. Therefore, the process may take a couple of seconds (depending on the selected plan section).

## Plan analysis → Trace outline

	Command:	
	ANAPATH	

This command runs an outline tracing. It can be used to detect gaps and for creating polylines.

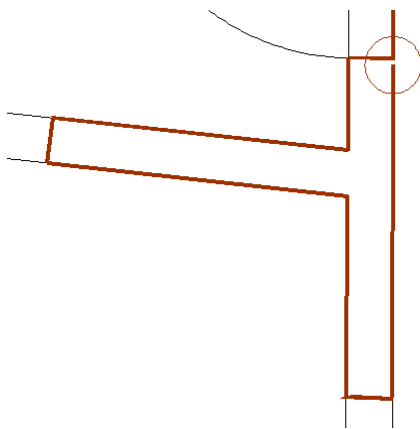
First, following is prompted in the command line.

Plan analysis -

Click point next to line or [Configure]:

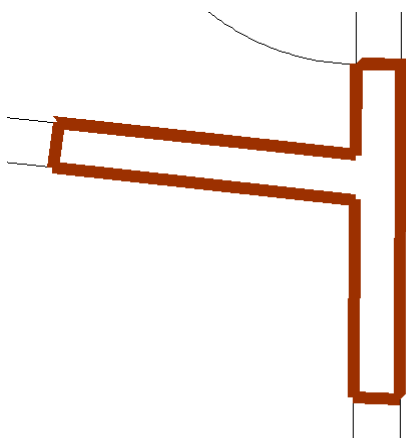
*(click a point)*

Click on a point next to a line. Now the outline of this line will be traced and highlighted in red. At intersections the side will be preferred that has been selected while clicking the point.

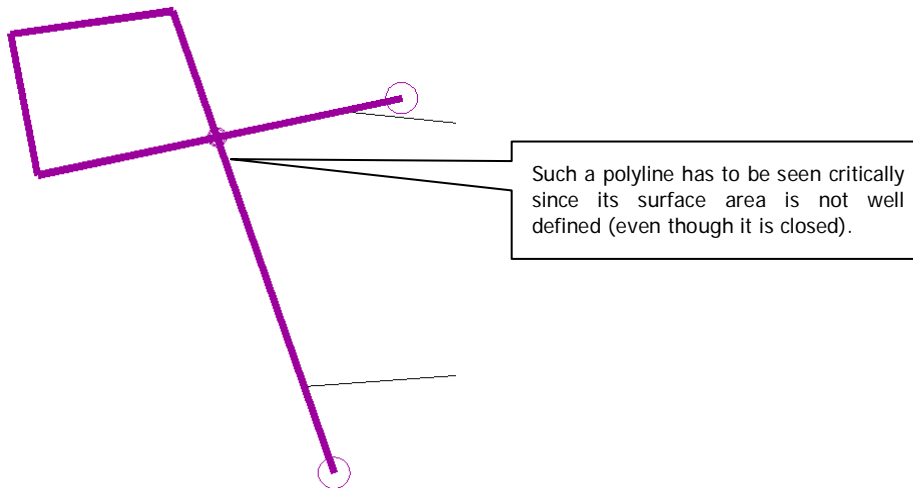


The ends of an outline are highlighted by circles.

Possible gaps within the line are ignored, provided they are smaller than a certain tolerance value. You can determine this value using the command line option “Configure” or the menu command → **Configuration**. The width of the line marking also depends on this tolerance. In the following figure a larger tolerance value has been set and, thus, the outline tracing finds a closed polygon. When there is a closed polygon, no circles are displayed:



If the outline tracing results in a so called self-overlapping, the intersection points are marked by concentric circles. The outline line is also highlighted in purple.

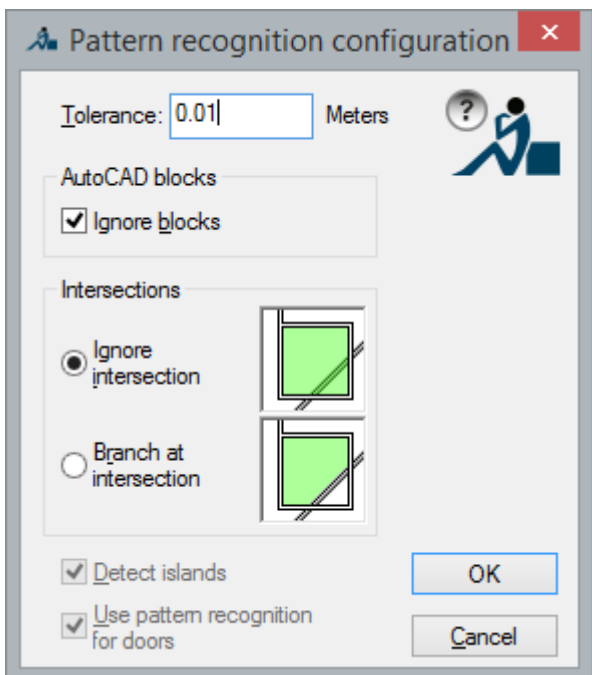


The outline tracing function runs a complex analysis of the CAD graphic. Therefore, the process may take a couple of seconds (depending on the selected plan section).

After you have traced an outline, you can run another analysis or take this outline over as AutoCAD 2D polyline. Following prompt will appear repeatedly in the command line until you cancel the command.

Click point next to line or [insertPolyline/Configure]:

The option “Configure” opens the following dialogue.



- **Tolerance:** Enter the value when to skip a gap
- **Ignore blocks:** Decide to ignore AutoCAD blocks or not

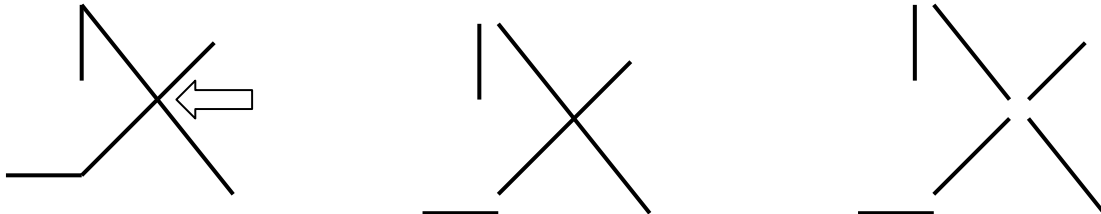
The “Intersection” options changes the mode of outline tracing in a fundamental way:

- **Ignore intersection:** outline tracing will NOT turn at true X-intersections
- **Branch at intersection:** outline tracing will also turn at true X-intersections

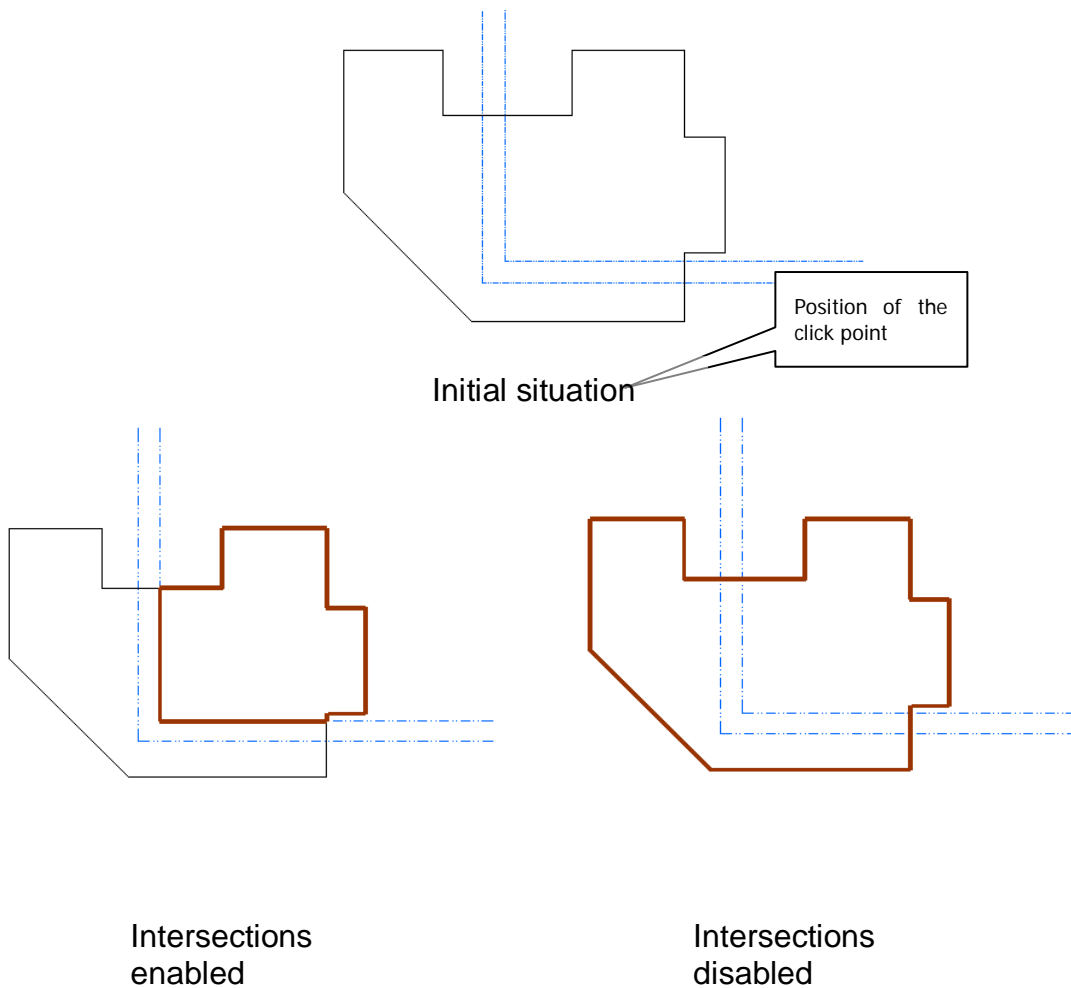


We speak of true X-intersection when the AutoCAD elements intersect instead of just meeting.

Intersection situation in the drawing (4 or 6 lines?)	Exploded view	
	true X-intersection (4 lines)	pseudo X-intersection (6 lines)




Here an example for the effect of the two modes with the same click point.



The reason for the two modes is: In most CAD plans pseudo X-intersections are used for objects that are connected with each other by content (e.g. adjoining walls and windows or doors). Things that overlay the actual plan (e.g. center lines and building grids) are often simply drawn through. Then real X-intersections are created. Thus, it is mostly better to ignore X-crossing lines when creating room polygons and the like.

## Visualization → Update order of display

	Command:	
	VISORDERSORT	

This command allows to revise the order of display of the graphic elements of a drawing. This is especially important when solid hatchings are used in feature data visualizations. Often, AutoCAD puts these in the foreground and thus the other graphic elements are hidden.

The order of display will only be updated on the currently displayed screen section. If you want to update the order of display for the whole drawing, you should run the AutoCAD command “Zoom Extents” beforehand.

Following rules apply for updating the order of display:


at the bottom lie	:	images
		SOLID hatchings
		non-solid hatchings
on top lie	:	any other AutoCAD elements

↓

It is a rule for hatchings, that larger hatching areas will be placed beneath smaller areas.

The definition of visualizations is done via → **Define visualization**. You can switch visualizations on and off via → **Visualization – Enable all / Disable all**.

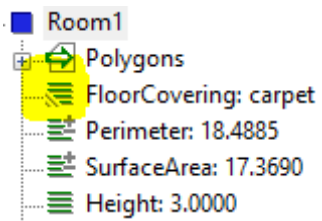
**Visualization → Disable all**

	Command:	
	VISUALOFF	


This command allows to hide all displayed FARO feature data visualizations.

If you want to switch off the visualization of individual attributes, use the command **→Visualization...** within the context menu of an attribute.

If the visualisation of an attribute is switched off, it is marked grey in the structure tree:



**Visualization → Enable all**

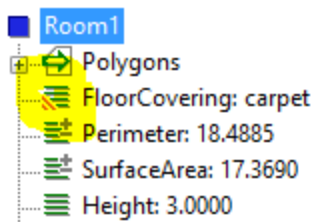
	Command:	
	VISUALON	

This command allows to display all FARO feature data visualizations.

**Note:** If the drawing contains much data and several visualized attributes, this function could require considerable time. Your system performance also plays a role here.

If you want to switch on the visualization of individual attributes, use the command **→Visualization...** in the context menu of an attribute.

If the visualisation of an attribute is switched on in the drawing, it is marked red in the structure tree:



**Visualization → Material visualization → Display on**

	Command:	
	MATERIALVISUALIZATIONON	

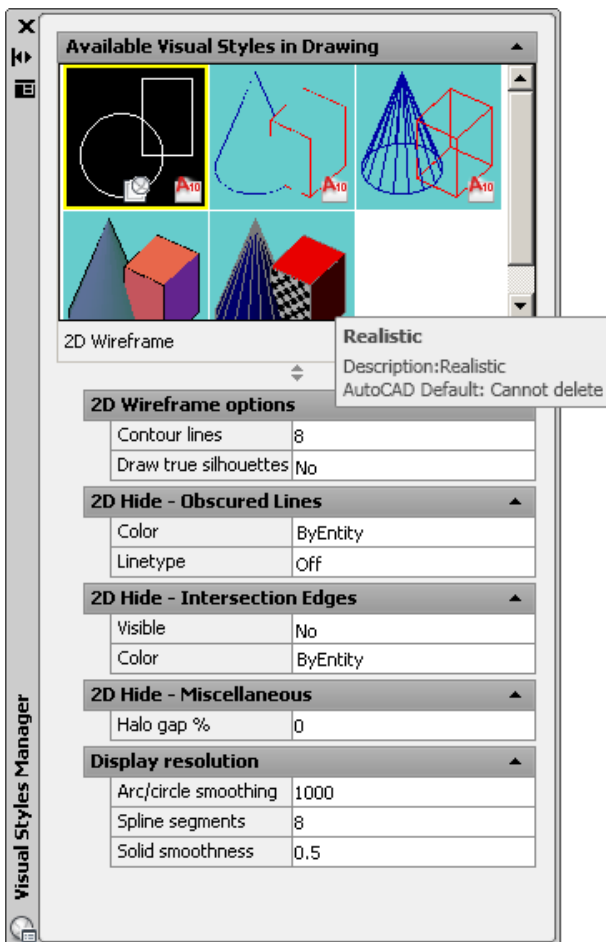
With this command you are able to show a feature data material visualization within the drawing.

It switches the display of your drawing window into the visual style "realistic". AutoCAD is only able to display the materials if the visual style has been set to "realistic". For other visual styles (AutoCAD uses "2D wire frame by default) the fillings of the materials will be empty or solid.

Additionally the software updates the order of display of your visualizations.

With **→Visualization**  **Material visualization**  **Display off** you may switch the visual style back to "2D wire frame".

**Hint:** With the AutoCAD command "Visualstyles" (command line entry) you can call up the current visual style.



**Visualization → Material visualization → Display off**

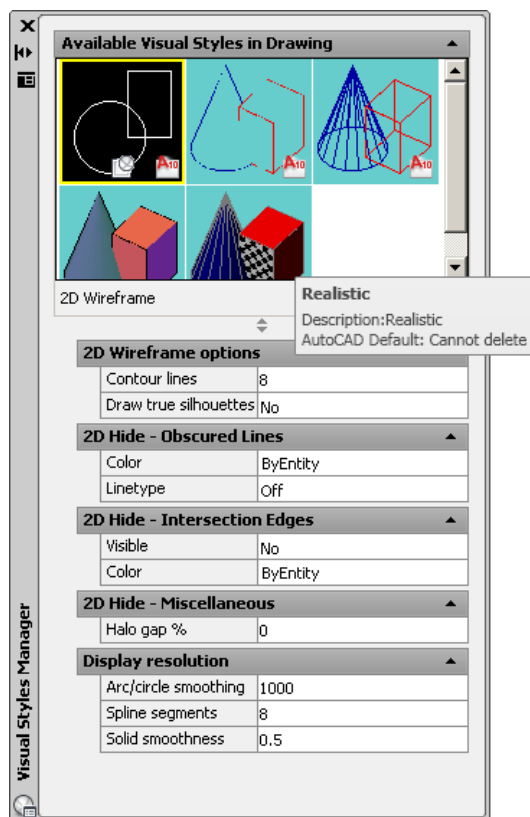
	Command:	
	MATERIALVISUALIZATIONOFF	

This command switches the display of your drawing window into the visual style "2D wire frame". AutoCAD can only display materials, when the visual style is set to "realistic". When being in the "2D wire frame" mode, the fillings will be empty.

Additionally the software updates the order of display of your visualizations.

With **→ Visualization**  **Material visualization**  **Display on** you can make the material visualization visible again.

**Hint:** With the AutoCAD command "Visualstyles" (command line entry) you can call up the current visual style.



**Help → Tutorials Feature Data → Tutorial 1/Tutorial 2/Tutorial 3**

**Help → Tutorials Facade Mapping → Tutorial 1/Tutorial 2**

**Help → Tutorial Archeological Excavation**

Command:	
TCADTUTORIALFD	
HYLASFMTUTORIAL1	
HYLASFMTUTORIAL2	
HYLASMMTUTORIAL1	
HYLASMMTUTORIAL2	
TCADTUTORIALBD	

By calling one of these commands, you open the corresponding sample drawing that is required for working on the respective tutorial.

A corresponding subfolder is created in the user data folder under \TUTORIAS, e. g.

```
\USERS\[NAME]\DOCUMENTS\FARO\AS-BUILT          XXX          EN
ACXXXX\TUTORIALS\FEATURE_DATA\TUTORIAL1
```

All files needed for the respective tutorial are saved there. For example:

- TUTORIAL1.DWG
- TUTORIAL1\_RESULT.DWG

The corresponding tutorial drawing (e. g. TUTORIAL1. DWG) is opened automatically at the same time.

The tutorials use simple examples to introduce you to working with FARO Feature Data Management.

TCADTUTORIALFD provides an introduction to the **area data management for Facility Management**. You find a description of this tutorial in chapter 2 **Introductory Tutorial1 (Area Data Management)**.

HYLASFMTUTORIAL1 deals with structured data capture. You find a complete description of tutorial 1 in chapter 3 **Data Capture – Tutorial**.

HYLASFMTUTORIAL2 introduces you to data modeling with FARO's feature data management. There are various possibilities for creating objects and linking objects. The most important possibilities for the precise configuration of the FARO data management are also presented. You find a complete description of tutorial 2 in chapter 5 **Modeling and configuration – Tutorial**.

### **Façade (Damage) Mapping**

HYLASMMTUTORIAL1 deals with data structuring for the purpose of damage mapping on historical facades. You find a complete description of tutorial 1 in chapter 6 of the manual.


HYLASMMTUTORIAL2 explains how you can customise the structure definition to meet your own requirements. You learn how to create new attributes and how to enter new values for existing attributes. The visualisations will be adjusted. You find a complete description of tutorial 2 in chapter 7 of the manual.



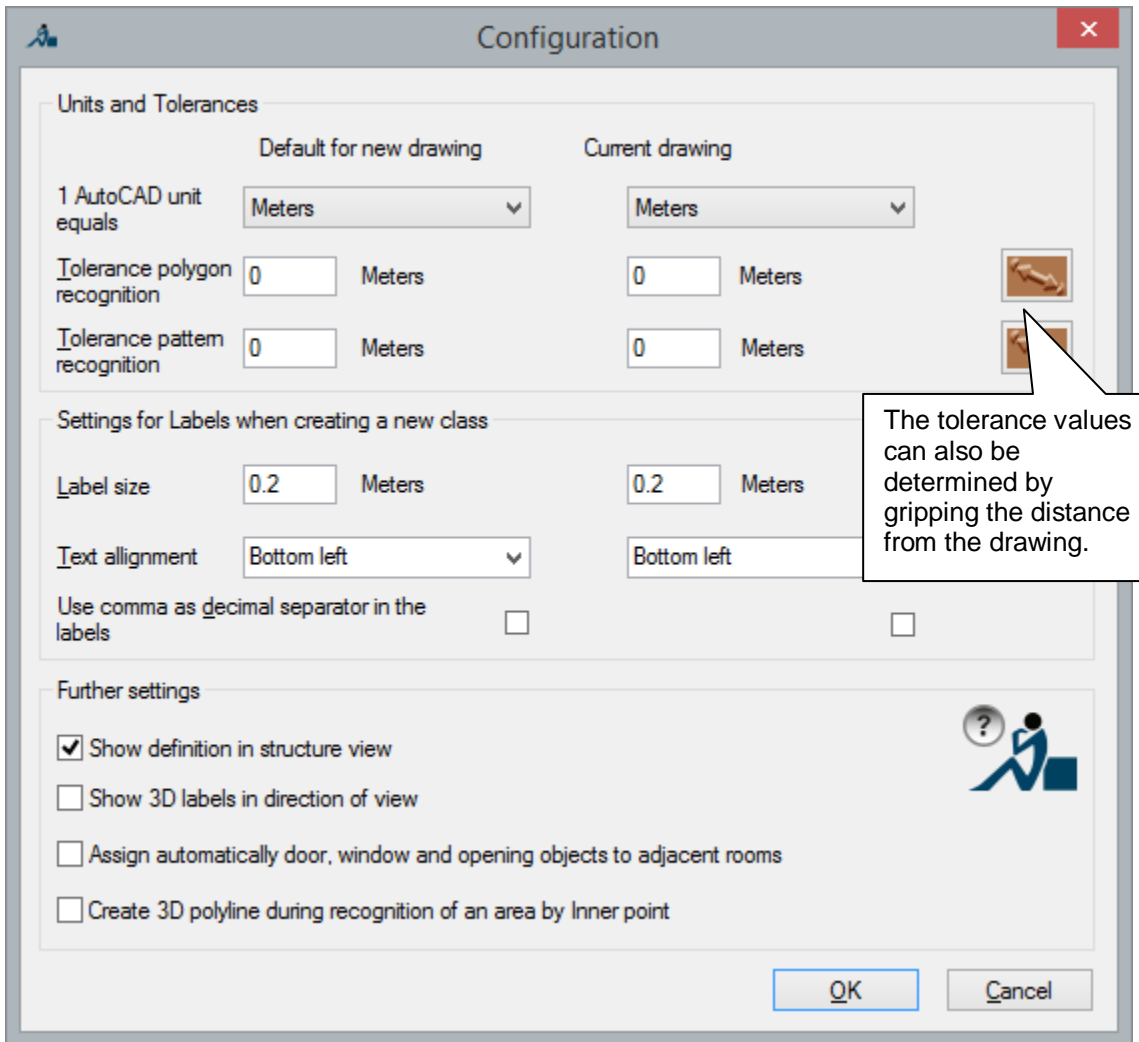
**Archeological Excavation Dokumentation**

TCADTUTORIALBD shows how data management can be used for the documentation of archaeological excavations. You find a description of this tutorial in chapter 8 Tutorial – Archaeological Excavation Documentation.

## Settings → Settings... (Configuration)

	Command:	
	HYCONFIG	

The command "hyconfig" allows you to configure the FARO feature data management.



### Units and tolerances

You may configure the used unit of length and the tolerance measure.

You may set two values for each of these parameters:

- The *default value* is the value that is to be used in every new drawing (not processed by FARO feature data management yet).
- The value for the *current drawing* that is to be effective immediately.

The correct setting of **the length unit** is important for the proper function of the pattern and polygon recognition (e.g. command "inner point" for creating areas). Enter, to which length unit one AutoCAD unit corresponds.

The **tolerance polygon recognition** indicates the allowed size of gaps in the plan that are skipped by the outline tracing. It is important for the proper function of the polygon recognition (create area by “inner point”, “Trace outline”)

The **tolerance pattern recognition** indicates the allowed deviation from the defined pattern graphic during pattern recognition.

### Settings for labels when creating a new class

The indication of the **label size** refers to a label with one line and is expressed in AutoCAD units.



Moreover the text alignment of a label can be set. The default setting is “bottom left”. When adding attributes to be displayed in the labels they grow from bottom to top. A second option is the alignment “top left“. Accordingly the labels grow from top to bottom when adding attributes.

### Further settings

**Show definition in structure view** Here you can enter whether the structure definition (the grey region in the structure view) should be shown or hidden. During data acquisition there should, wherever possible, be no more changes be made to the definition. The definition part can then be hidden.

**Show 3D labels in direction of view:** Switch to 3D labels. If you activate this option, then the feature data labels will always be so spatially aligned that they are always readable. If you are working in a 3D model and view the model from various directions (AutoCAD command 3DORBIT), then the labels will always be readable.

**Create 3D polyline during recognition of an area by Inner Point:** This option allows, when using the internal point method for area creation, to retain the original height information of the base objects, such as lines or 3D polylines, and to attach for each so created polygon a 3D polyline as a graphic object. The 3D polyline is in the X-Y plane congruent with the associated polygon. When activating this option the area class must be configured, for the management of any drawing elements.

**Assign automatically doors, windows and openings objects to adjacent rooms:** Should the classes room, door, window or opening be defined in the structure definition, then an automatic link can be created. The requirement for this is an associated configured link between the class room and the classes door, window and/or opening. The link must be defined with cardinality of {1,1} or {1,2}. for door/window/opening:

**New type of link**

Name of link  
RoomHasDoor

classes to be linked  
Room  
Door

Properties of the link

Consider 'Door' when calculating the area for 'Room' (Subarea).

Link 'Door' on the basis of position with 'Room' (Equipment)  
... without use of other relations Edit ...

at least tangent to Minimum requirement for linking areas

Monitor number of objects to be linked (Cardinality)

{1,2} Number of 'Room' objects that must be linked with a 'Door' object.

{1,+∞} Number of 'Door' objects that must be linked with a 'Room' object.

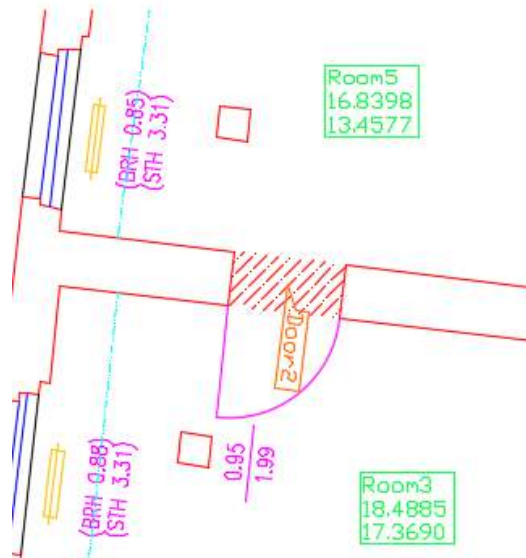
Also show linked objects

Also delete linked objects

OK Cancel

With the command “Add aligned door” the door will be created and the label will be placed within the wall between the two rooms.

With a cardinality of {1,1} the door will only be associated with the room into which the door opens (room 3 in figure), but {1,2} will however associate both of the rooms which the door connects.



## 11.2 Commands of the context menu of the structure view

Most functions for data structuring are not accessible via the AutoCAD menu or toolbars but are available as commands of a context menu in the structure view (right mouse button).

### Context menu command

Symbol	Type of structure entry
	Restrictions

#### Context menu command

defines the precise entry within the context menu. "**NAME**" stands for a concrete structure entry.

Example:

Within the context menu of the structure entry **Room12** the command "Show **Room12**" stands in first position. You find the description of this command in the command reference under **Show "NAME"**

#### Symbol

graphic symbol within the structure view.

#### Type of structure entry

specifies to which type of structure entries the command can be applied, e.g. for classes, objects or attributes.

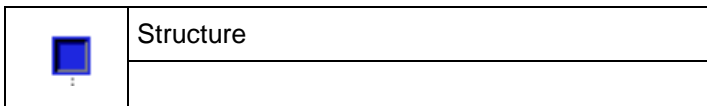
#### Restrictions

Certain context menu commands can not be applied to all structure entries of a kind. For example, the command →**Add "Area" (Polylines)** exists only for objects that inherit from the class Area.

For clarity, the restrictions are stated within the header of the context menu command with a backslash "\":

Add "NAME" (Polylines) \ Area

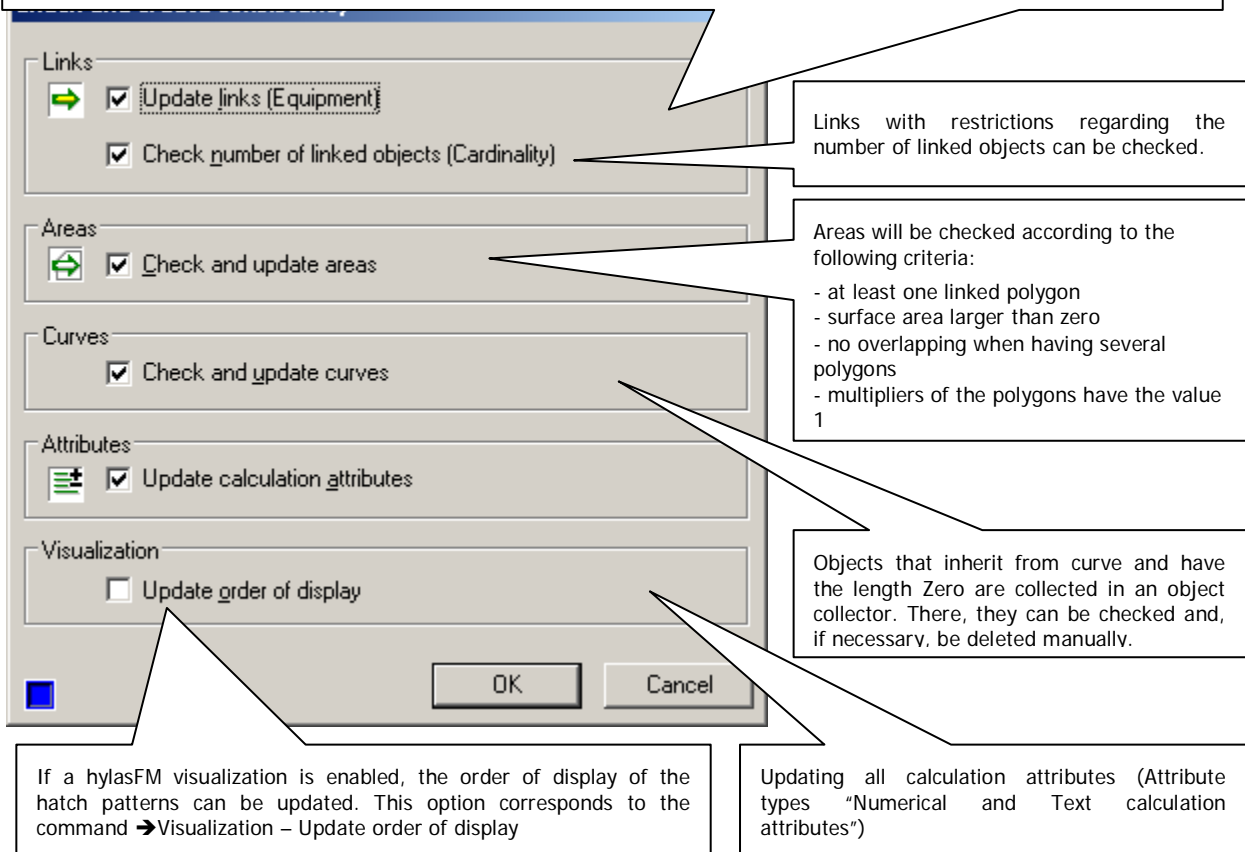
## Check/Create consistency...



This command analyses and updates the complete drawing as well as all its data objects. By doing so different inconsistencies within the data structure will be localized and part of them will be corrected. We recommend to run this command before closing the drawing or if many data have been entered.

A window with different setting options opens.

- 1) If a link exists between objects that have been derived from the class area (e.g. rooms) and other objects (e.g. equipment) that have a label, these are automatically linked with each other, if the insertion point of the label lies within the area (polygon).
- 2) If a link exists between two objects that both have been derived from the class area, these are automatically linked with each other if both areas (polygons) of the objects intersect.
- 3) If a link exists between objects of which one has been derived from the class area and the other one from the class curve, these are linked with each other if both outlines of the objects intersect.




After the check a report is created. Unclear objects are made available in one or more object collectors for further editing.



### Remarks

The properties Equipment and Cardinality of the links can be customized in the definition part of the links under the context menu entry → **Configure...**

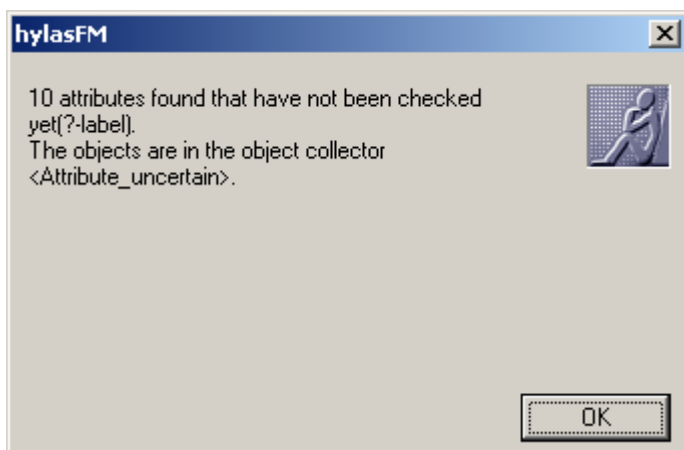
Attributes of the type "Numeric calculation attribute" or "Text calculation attribute" are created like regular attributes within the definition part of the respective class using the command → **Add attribute...**

**Find insecure attributes**

	Structure


This command is for the control of the capturing. With it you may start an automatic search for insecure attributes. Those are attributes, which have not been confirmed by the user during the capturing (  ) of which had to be changed automatically by the software in order to keep the consistency (  ).

In case insecure attributes have been found, the software will store the affected objects in an object collector and show the following message:

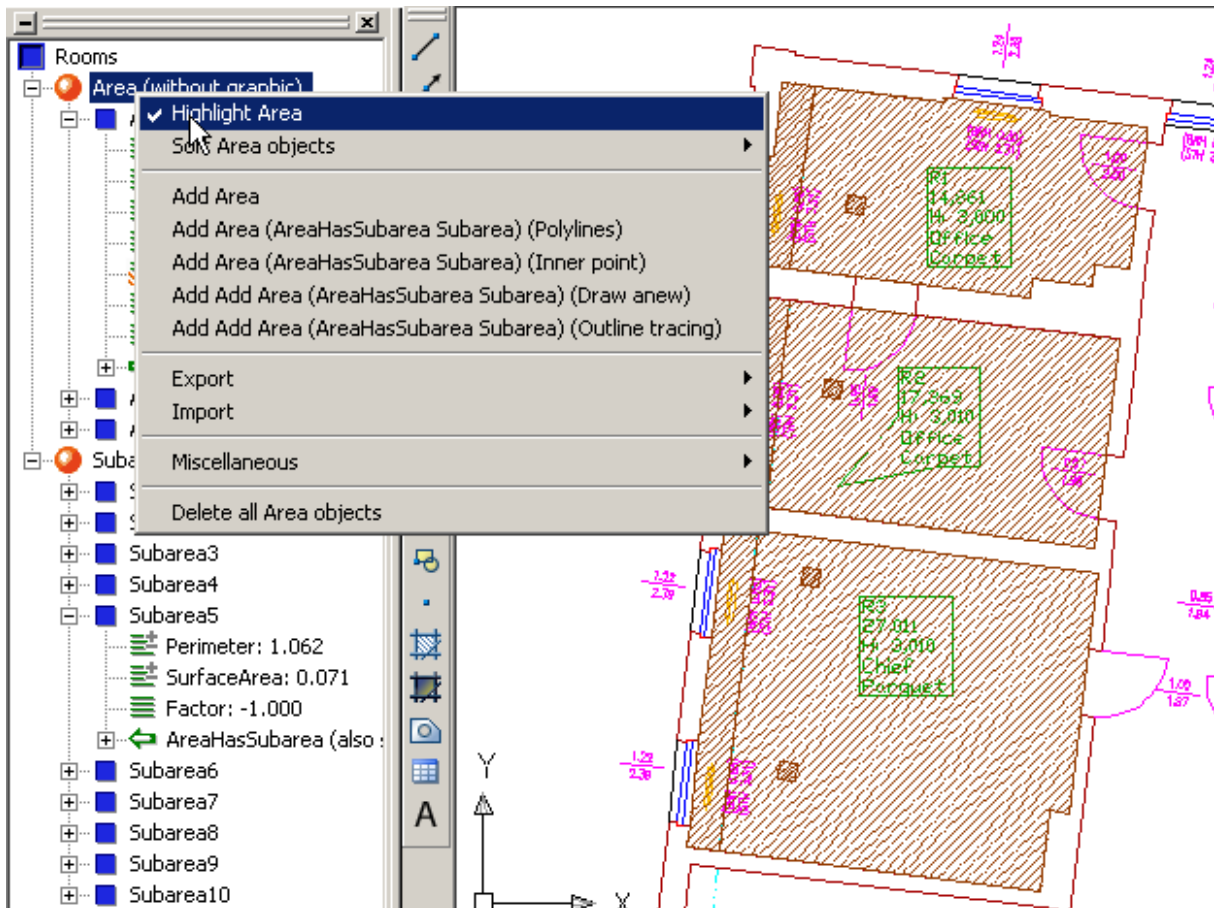


The attribute values may then be checked within the object collector and be edited manually where necessary. You will find further information regarding the object collectors within the command reference under → **11.3 Commands of the context menu of the object collector.**

## Highlight "NAME"

	Class
	only for classes "Area" and derived classes

This command can be used to highlight laminar objects. All objects of the selected class will be hatched. By running this command again, the hatch pattern is switched off. The current hatch pattern state is indicated by a check mark in front of the command.



### Remark

All hatchings that result from highlighting are deleted before saving! The hatchings are intended for the temporary visualization of objects that have been recognized by the feature data management and are therefore **not** saved. If you need a permanent hatching, use the command → **Define visualization...**




**Sort "NAME" objects (→ascending, →descending)**

	Class

This command allows you to sort all objects of a class in the structure view in ascending or descending order.

The sorting is carried out according to the entry in the object header. By default the object ID stands in the object header. However, you can display any other attribute (→ **Show in header**). Thus, you could sort areas by their surface area for example.

## Add "NAME"

	Class

The command "**Add "NAME"**" creates a new object of the class "NAME".

Depending on the configuration of the class "Name" (→**Configure...** (class)), different things are prompted in the command line. Here the default case

Select geometry for "Name":

Select objects:

*(Select graphic)*

If you have selected the graphic belonging to the object (any AutoCAD objects), finish the input by **RETURN** or by clicking with the right mouse button. If you do not want the object to contain a graphic, just press RETURN or ESC right away. You are then prompted to state a point for the insertion of the label.

Insertion point for label:

*(Select point)*


After the selection of the insertion point, the object is created and an according label is inserted in the drawing.

If you have configured the class accordingly (→**Configure...** (class), tab label: the option "Prompt attributes immediately after object creation" is enabled), all attributes of the object are prompted.

### Specifics


- If the class "NAME" does not have a graphic or label (→**Configure...**, tab Label and tab Graphic) you do, of course, not have to indicate a graphic or an insertion point when creating an object of this class.
- If an AutoCAD block is assigned to the class "NAME" (→**Configure...**, tab Graphic option "Block" is enabled), the command works slightly different. Depending on the chosen insertion method ("Request insertion point", "Align at object" or "Fit block"), you are prompted for entries. The block is inserted and linked with the feature data object. Thus, you may save a lot of time when generating a plan.

**Add “NAME“ (Polyline/Line) \ Curve**

	Class
	only for the class “Curve” and derived classes

This command allows you to create objects of the class “Curve” or of a derived class. Per object one AutoCAD curve is indicated as graphic. If you select several curves, several objects of the class “Name” are created.

## Add "NAME" (Draw anew) \ Curve

	Class
	for the class "Curve" and derived classes

This command allows you to draw a 2D polyline and at the same time convert it to a feature data curve object. You are prompted to indicate a set of polyline interpolation points and finally the insertion point of the label.

One can select between a linear and a curved run of a polyline. The command line options "arc" and "line" serve as switch. The curved run corresponds with the display of "Fit" in the AutoCAD command **PEDIT**.

Drawing a polyline is finished by using the option "Finish" or by pressing **ESC** or **RETURN**.

Command: HYLASCONTEXT

Specify first point or [Arc] <Arc>:

*(click first point or change mode)*

Specify next point or [Arc/Undo] <Arc>:

*(click second point or change mode  
or delete last point)*

Specify next point or [Finish/Arc/Undo] <Finish>:

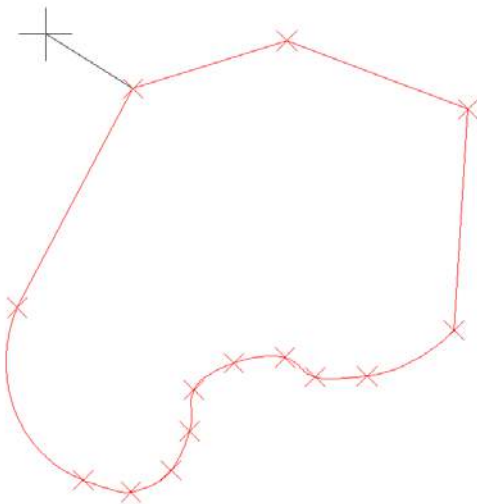
*(click third point or change mode  
or delete last point  
or finish)*

Specify next point or [Finish/Arc/Undo] <Finish>:...


*(click further points ... finish or ESC)*

Insertion point for label:

*(click a point)*



**Add “NAME“ (Outline tracing) \ Curve**

	class
	for the class “Curve” and derived classes

With this command you can start an outline tracing of a geometry in your plan. As a result, a new 2D polyline is created that is converted into a feature data curve object automatically. The new polyline is created by scanning the selected outline.

You are prompted to click on a point next to the graphic to be analyzed. If you select further points, the course of the outline will change. After that the insertion point of the object label is prompted.

Command: HYLASCONTEXT

Click point next to line or [Configure]:

*(Point next to the graphic to be analyzed)*

Click point next to line or [Configure]:


*(further points or Return)*

Insertion point for label:

*(Indicate insertion point of the label)*

The curve object is linked with the new 2D polyline that has been created by outline tracing. For further information on outline tracing, please refer to **→Plan analysis → Trace outline.**

## Add "NAME" (Polylines) \ Area

	Class
	only for the class "Area" and derived classes

With this command you can convert one or more 2D or 3D polylines into feature data area objects.

Command: HYLASCONTEXT

Polylines that are to be transferred to areas.

Select objects: *(select the polyline(s) to be transferred)*

Every selected polyline is automatically closed and transferred to a polygon. Polygons that lie within another polygon are recognized as "enclaves" (enclaves). That means the factor -1 is assigned to the enclaves and they build an area together with their polygon. The enclave polygons are subtracted from the total surface area.

The polylines are kept when using this command. They are linked as graphics with the polygon objects (subareas).

When you have selected a 3D polyline, a 2D polyline is automatically created from the projection onto the X-Y plane of the current coordinate system. This new 2D polyline is then linked as a graphic object to the polygon object.

If the area class is configured for the management of all AutoCAD graphic objects, then the 3D polyline can be used as the graphic object. The decision is made on the command line.

- When there is precisely one area:

Select geometry <use 3d polyline>:


*(select a further graphic object or with the Enter key use the option "3D polyline")*

- When there are several new areas, the 3D polylines can be appended as managed graphic objects:

Any selected 3D polylines are to append as graphic [Yes/No] <Yes>:

*(select an option)*

## Add "NAME" (Inner point) \ Area

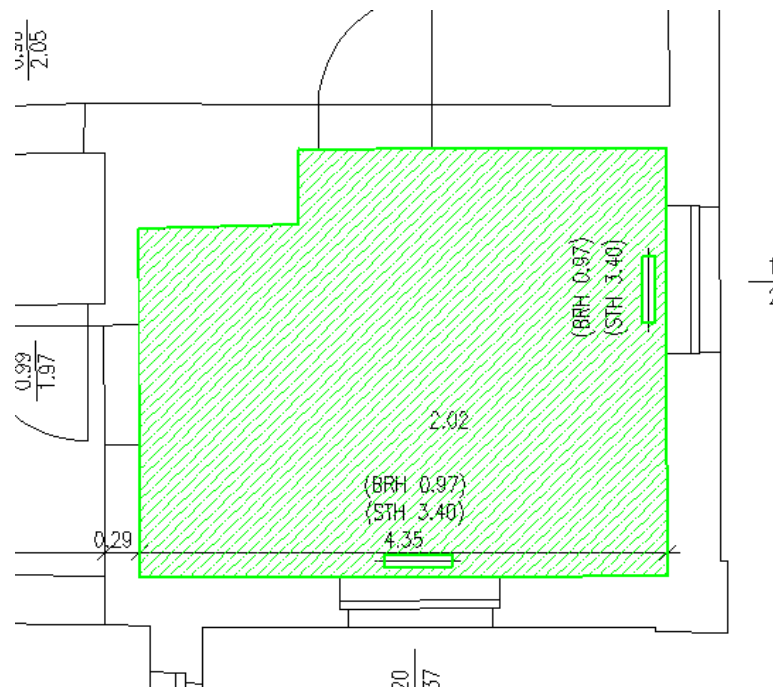
	Class
	only for the class "Area" and derived classes

With this command you create an object of the class "Name" without a previously existing polyline. The visible plan section is analyzed by an innovative outline tracing and one or more 2D polyline(s) are aligned at existing geometry (lines, arches, blocks etc.). Small gaps are skipped.

Following prompt appears in the command line.

Click inner point of a polygon for "NAME" or [Configure]:

Click an interior point of the area to be captured. A green hatching appears that is defined by a green frame. The width of the frame depends on the set tolerance value and represents the ability of the process to skip gaps.

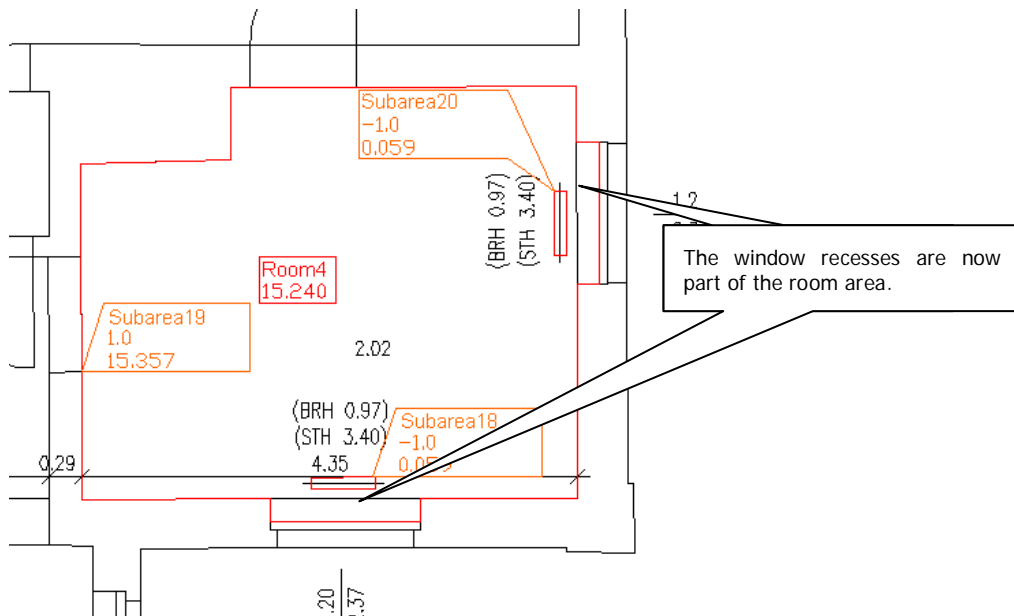


If further polygons are to be added, click into further (not necessarily adjoining) areas after the prompt:

Click inner point of a further polygon for Room or [Undo/Analysis/Configure]:

Thus, you could, for example, include the two window reveals.

An area according to the following figure will be generated:



The area (here an object of the class “Room” derived from “Area”) itself does NOT have polylines. However, it is connected with three polygons by an according link and sums up their surface areas. Every polygon is linked with exactly one 2D polyline. If you have configured the class (here: room) accordingly (→ **Configure...**, "Prompt attributes immediately after object creation" is enabled), you are now prompted to enter the further attributes for the new object (Room4).

**Option “Configure”**

Following settings can be made with this command by selecting the option “Configure”.

The screenshot shows the 'Pattern recognition configuration' dialog box with the following settings and callouts:

- Detection for polygons:** Tolerance: 0.02 Meters. Callout: Enter a dimension value for the maximum size of a gap. Gaps of maximum this size will be overlapped during the outline tracing.
- Ignore blocks. Callout: Here you decide, whether AutoCAD blocks will be used or not for outline tracing.
- Detect islands. Callout: Here you decide, whether enclaves in the outline will be detected (details see below)
- Doors:**  Use pattern recognition for doors. Callout: Here you decide, whether the pattern recognition for doors should be used or not. (details see below)
- Tolerance: 0.05 Meters
- Ignore blocks

Buttons: OK, Cancel

▪ **Tolerance**

Enter the size of the gaps to be skipped.

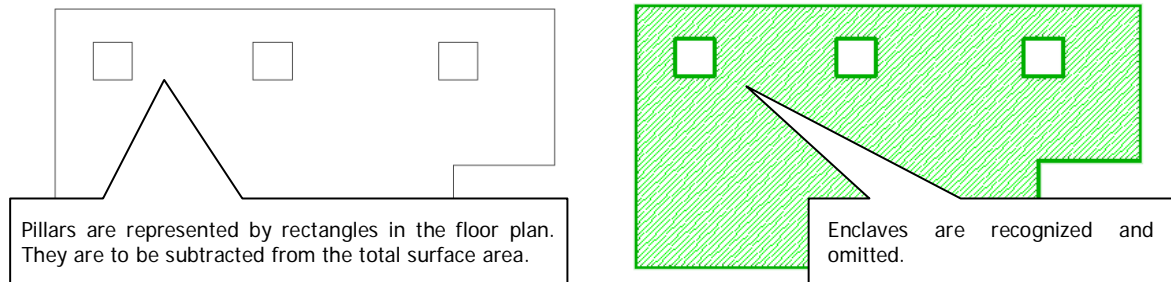


- **Ignore Blocks**

Here you can decide, whether AutoCAD blocks will be used for outline tracing or not.

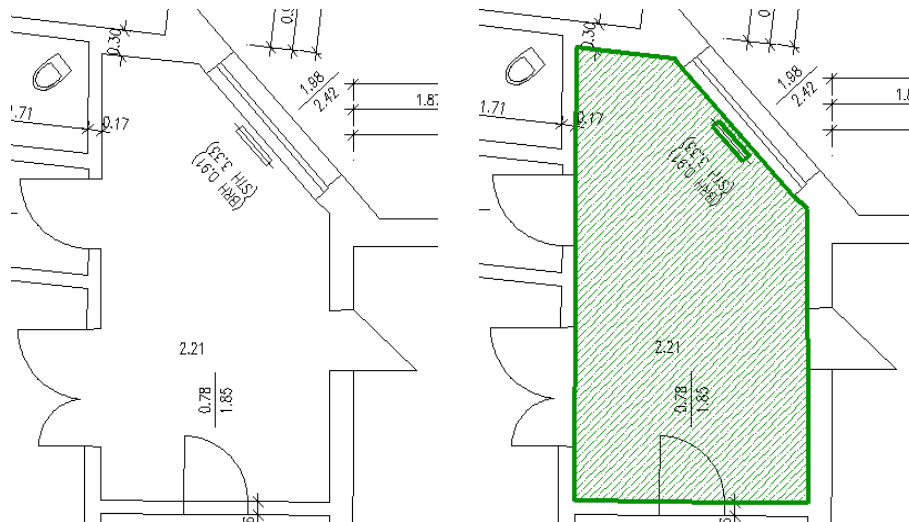
- **Detect islands**

If this mode is enabled, the outline of closed lines inside an area are recognized as “enclaves” and automatically are considered in the surface area. A separate polygon is created for every enclave. Their surface areas are subtracted from the total surface area.




- **Use pattern recognition for doors**

If this mode is enabled, the software will recognize door symbols during outline tracing and will ignore them when creating polylines.



## Add "NAME"(Draw anew) \ Area

	Class
	only for the class "Area" and derived classes

This command allows you to draw a 2D polyline and at the same time convert it to a feature data area object. You are prompted to indicate a set of polyline interpolation points and the insertion point of the label.

One can select between a linear and a curved run of a polyline. The command line options "arc" and "line" serve as switch. The curved run corresponds with the display of "Fit" in the AutoCAD command **PEDIT**.

Drawing a polyline is finished by using the option "Finish" or by pressing **ESC** or **RETURN**.

Command: HYLASCONTEXT

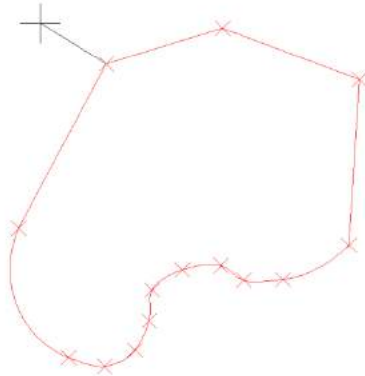
Specify first point or [Arc] <Arc>: *(click first point or change mode)*

Specify next point or [Arc/Undo] <Arc>: *(click second point or change mode  
or delete last point)*

Specify next point or [Finish/Arc/Undo] <Finish>: *(click third point or change mode  
or delete last point  
or finish)*


... *(click further points ... finish or ESC)*

Insertion point for label: *(click a point)*



An area object and a polygon that is linked with the area are created. The newly drawn polyline will be assigned as graphic to the polygon.

## Add “NAME“(Outline tracing) \ Area

	Class
	for the class “Area” and derived classes

With this command you can start an outline tracing of a geometry within your plan. As a result, a new 2D polyline is created that is automatically transferred into a feature data area object. For further information on outline tracing, please refer to **→Plan analysis → Trace outline.**

You are prompted to click a point next to the graphic to be analyzed. If you select further points, the run of the outline will change. After that, the insertion point of the object label is prompted.

Command: HYLASCONTEXT

Click point next to line or [Configure]:

*(Point next to the graphic to be analyzed)*

Click point next to line or [Configure]:


*(further points or Return)*

Insertion point for label:

*(Indicate insertion point of the label)*

An area object and a polygon that is linked with the area are created. The polyline, created through contour tracking, will be assigned as graphic to the polygon.

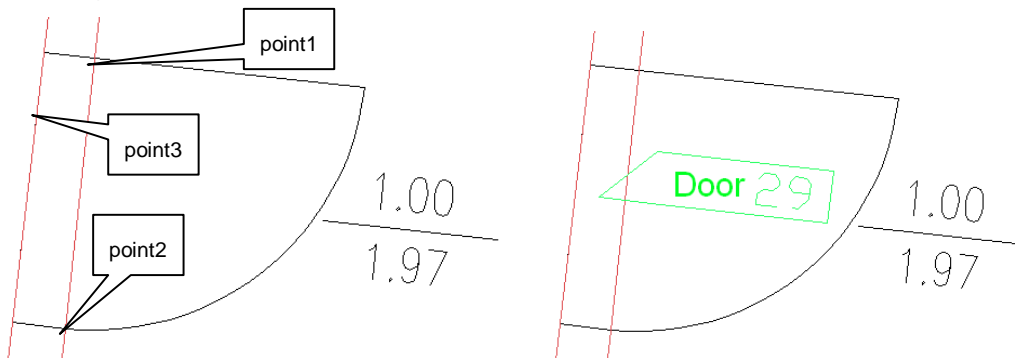
### Add "NAME" aligned \ Window, door

	Class
	only for the classes "Window", "Door" and derived

If doors and windows are to be converted into 3D elements of the Architectural Desktop (ADT) or AutoCAD Architecture later (**→Export→Create ADT door**), it is important that not only width and height but also the position in the room is saved. For this the software uses the alignment of the label. In order to create a door aligned, the command prompts three points in the command line:

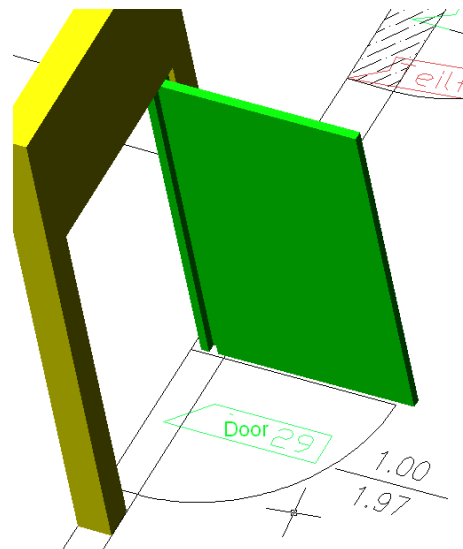
- first point (inside, right) (Snap point1)
- second point (inside, left) (Snap point2)
- Point on other wall side (outside) (Snap point3)

Following points in the environment of the door symbol should be clicked:



The third point serves the determination of the wall thickness and the opening direction of the door. It can be arbitrarily selected on the other side of the wall.

If later on an AutoCAD Architecture door is created, it has the correct alignment.




Adding aligned windows is run analogous. Three points are prompted in the command line:

first point (inside, right)	<i>(Snap point1)</i>
second point (inside, left)	<i>(Snap point2)</i>
Point on other wall side (outside)	<i>(Snap point3)</i>

If a window opens to the inside, the first two points should be selected on the side of the room, the third on the outline of the building.

Another advantage of the aligned insertion is that a possibly existing attribute "Width" is filled in automatically.

## Add "NAME" (Pattern recognition) \ Window, door, wall

	Class
	only for the classes "Window", "Door" and "Wall" and derived and for classes that have been assigned a user defined pattern

The command behaves differently with different classes:

### Classes "Door" and "Window"

The prompt in the command line for the class "Door" reads:

Interpretation - click near a door:

*(Click on point)*

The FARO pattern recognition searches for a door or window pattern around the click point and, if applicable, creates a new object.

The door and window labels are aligned like in the previous command → **Add "NAME" aligned** so that a later creation of 3D Architecture objects is possible. If an attribute "Width" exists, it is filled in automatically.

Should the creation fail, please use → **Add "NAME" aligned**.

Also see → **Export** → **Create ADT doors**.

### Class „Wall“

For most FM projects it is not necessary to create walls. It does, however, become necessary when 3D elements are to be created. This command is of utmost importance when the goal of the stocktaking (what exactly is to be recorded?) is a model for the architectural desktop (ADT) or AutoCAD Architecture. Before we go into the details of this very complex command, at first some remarks on the basic functions:

Before calling up the command, please disable the **object snap**.

The following appears in the command line

Current settings: Aligned=off

Click first point or [Align/Reference] <Align>:

*(Click point1 inside the wall)*

Click second point or [Align/Reference] <Align>:

*(Click point2 inside the wall)*

Width: 0.376

Click point for new width. or

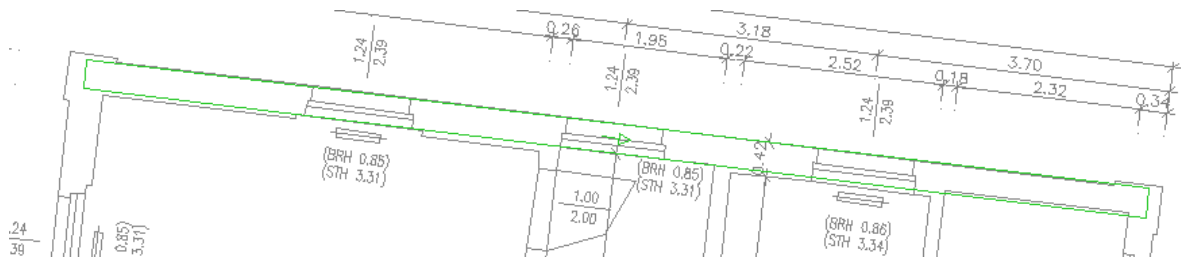
[OK/Centered/aligned left/aligned right/Undo] <OK>:

*(RETURN)*

Just before clicking point2, the screen will look like this:

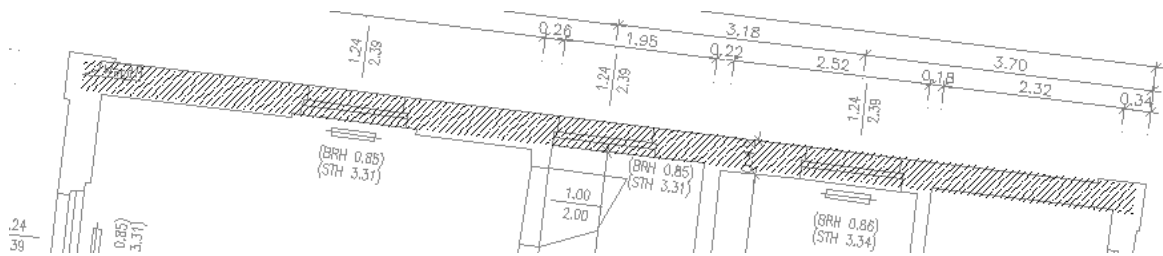


The wall line is approximately defined by the two points. The software analyzes the surrounding graphic and determines the correct line as well as the wall width. It then writes the width into the command line and the wall is represented by a green rectangle:



Width and line orientation are the result of the minimization of the deviation of the model “parallel wall” at the lines existing in the drawing.

If you finish the command with the option “OK” or by pressing RETURN, a label with a hatch pattern appears in the plan and the FARO feature data wall object appears in the structure view.



This object is the basis for creating 3D walls in the Architecture later (→Export→Create ADT walls).

Besides the illustrated standard procedure of creating walls, there is also the possibility to align walls at each other and/or to enforce exact wall width. Here the explanations of the further options of the command:

**Options “Align/Reference”**

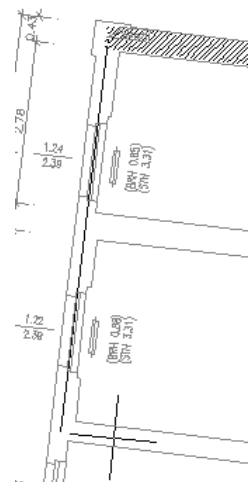
Often walls are to be aligned to each other in an orthogonal way. To achieve such a behavior, first select a reference object that defines the alignment. If you select the option “Reference” on the query of point1 or point2, the command line reads as follows

Select label, line or curve for alignment:

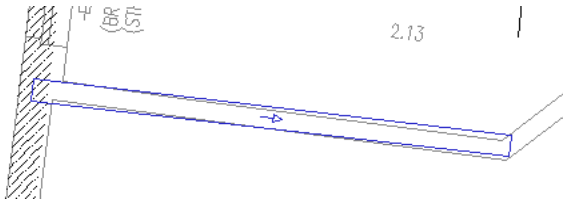
After the reference object has been indicated, the wall currently to be created can only be created orthogonal to this object.

Walls will be aligned orthogonally to object.

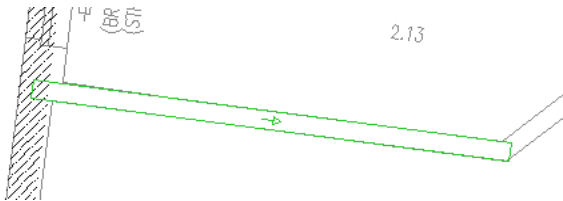
In the figure on the right, the label of the upper, horizontally running wall has been selected as reference object and a wall that is orthogonal to it is just being created.



For a following horizontal wall, orthogonality is enforced as well. The wall is – as in the figure – drawn in blue and be slightly oblique.



Here an illustration of the same wall without alignment



With the option “**Align**” you can switch between the two modes. The reference object is saved until a new reference is determined (with the option “Reference”).

### Options “Centered/Aligned left/Aligned right”

Shall a wall get values other than those suggested, there are two possibilities with the following situation in the command line:

Width: 0.456100

Click point for new width or

[OK/Centered/aligned left/aligned right/Undo] <OK>:

1. By clicking the wall boundary on one side, the wall width is varied.
2. When selecting one of the options “Centered”, “Aligned left” or “Aligned right”, you are prompted to enter a numerical value for the width. Depending on the chosen option, the number is interpreted differently. “Aligned left” leaves the left wall side at the position determined by the software, and only changes the right wall side. The other options behave analogous. The suggested value is the last value that had been entered manually (!) for the wall width. An example

After following dialogue in the command line

Current settings: Aligned=off

Click first point or [Align/Reference] <Align>:

Click second point or [Align/Reference] <Align>:

Width:0.231087

*(width determined by the software)*

Click point for new width or

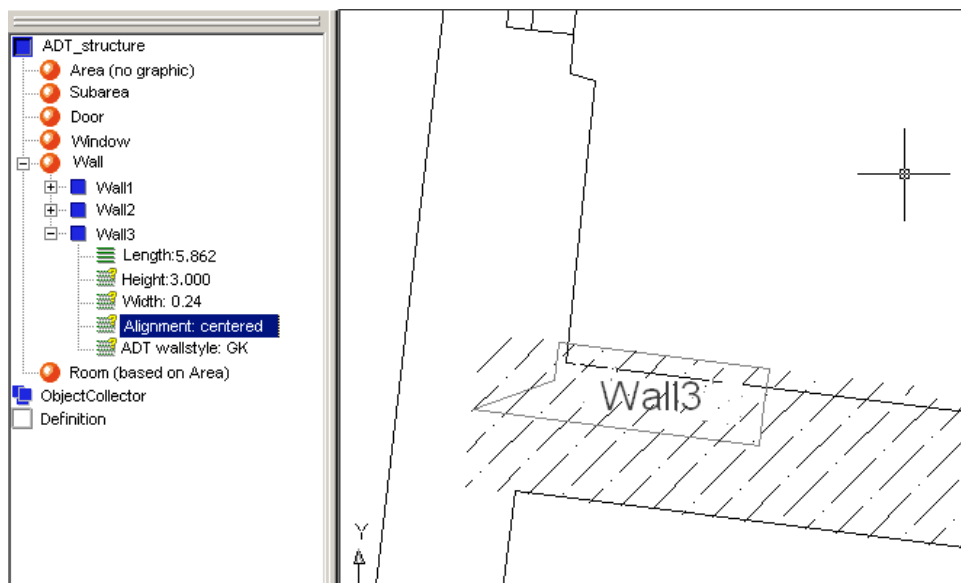
[OK/Centered/aligned left/aligned right/Undo] <OK>: right *(Option Aligned right selected)*

Width <0.48>: 0.24

*(Enter desired width)*

the screen will show this image:





The wall has been brought right aligned at a width of 0.24. **Note.** The wall attribute “Alignment” (here: centered) is in no way involved in this process. The attribute “Alignment” is later used when creating ADT walls (→**Export**→**Create ADT walls**) as value for the ADT wall property with the same name.

### Comment


For the wall styles of the Architecture model (independently of FARO feature data management) you can decide whether you want the wall thickness to be variable or fixed. Therefore, the wall thickness created by the feature data management can only influence styles with variable wall thicknesses. If you want an Architecture model with fixed wall thicknesses, you should use styles with fixed wall thicknesses (and only use FARO feature data management for the construction of the axes).

Also see →**Export**→**Create ADT walls**.

### Classes with user defined patterns

If a user defined pattern has been assigned to the class (→**Configure** (class), tab pattern recognition), you have to click near the pattern for object creation.


**Export → Blocks**

	Class

This command is used to create AutoCAD blocks. At every label of the class an AutoCAD block reference is created. The block reference is aligned according to the alignment of the label. Block attributes can be filled in with FARO feature data attributes.

Before this command can be put to use, the class has to be related with an export block. To do so, use the command **→Configure...** in the context menu of the class definition.

**Export → Table**

	Class

This command is used to create a Microsoft Excel table or an universally readable ASCII table. Every object is represented by one line in the table. The columns are assigned to the attributes of a class. The order of the attribute columns depends on the set order of sorting → **Sort attributes, show/hide....** Not visible attributes are not exported. When calling up the command, a window will open, where you will have to enter the name and the format of the table file. You can choose between these formats:

- Text file (TXT),
- Microsoft Excel (XLS) or
- Comma separated values (CSV)

The configuration (e.g. separator) can be carried out with → **Data export - Configure export...** from the feature data ribbon tab.

Shall the information of all classes be saved as table in one step, use the menu command → **Data export - table(s)....**


**Export → HTML**

	Class

This command is used to create a HTML table that can be read by your web browser. The table will contain a column for every attribute of the class and a line for every object of the class. When running the command, you are prompted for the name of the file to be created.

Should the information of all classes be saved as HTML table in one step, use the command → **Data export - HTML....**

**Export → Shape file**

	Class

This command is used to save the attribute data and geometry data of a class in the shape file format from ESRI. When invoking the command you will be prompted for the name of the file to be created (the drawing name + class name is suggested).


Thereby further processing of the data captured in FARO feature data management is possible in a large number of GIS systems.

To carry out a global export of all of the classes in one step the command **→Data→Export shape file ...** can be used.

When exporting classes and it is dealing with area classes, then during the export it will be checked to see if a 3D polyline has been appended as a graphic object. If yes, this will be exported instead of a 2D polygon.

For the required settings please see **→Configuration ...**


**Export → Regions**

	Class
	only for the class "Area" and derived classes

With this command you may generate AutoCAD regions from the polygons of all objects of a class of the "Area" type.

One region per object will be created on the current layer.


**Export → Enclosing rectangle**

	Class
	only for the class "Area" and derived classes

With this command you may display the smallest, all polygons of an object enclosing, rectangle (polyline). The values of the created rectangle equal in their dimensions the values in the attributes "BoundingBoxLongEdge" and "BoundingBoxShortEdge " from the class "Area".

One rectangle per object will be created on the current layer.

**Export → Create ADT-... \Room, Wall, Door, Window**

	Class
	only for the classes "Room", "Wall", "Door", "Window" and class derived from it


This command can be used when the FARO feature data management is installed and in use within AutoCAD Architecture.

Depending on the class, the specific ADT objects are created. Per object one style can be indicated in the attribute Autodesk Architecture style (with doors: Autodesk Architecture door style). To load the styles from the drawing into the default list of the attributes, use the command:

**→Data → Autodesk Architecture styles...**



**Export → Create allplan rooms (XML file)**

	Class
	only for the class "Room" and classes derived from it

This command is used for the export of all rooms into the CAD system ALLPLAN of the company Nemetschek.

A name for the XML file is requested. This file can be dragged into ALLPLAN by Drag & Drop. There, 3D rooms with the height of the room attribute are created.


**Import → Blocks**

	Class

This command is used to create feature data objects at the positions of AutoCAD blocks that already exist in the drawing. A label is created at every AutoCAD block reference of a certain block definition. The alignment of the labels is done corresponding to the alignment of block references. Feature attributes can be filled in with block attributes.

Before this command can be used, the class has to be linked with an import block. To do so, use the command **→Configure... (class)** from the context menu of the class definition.

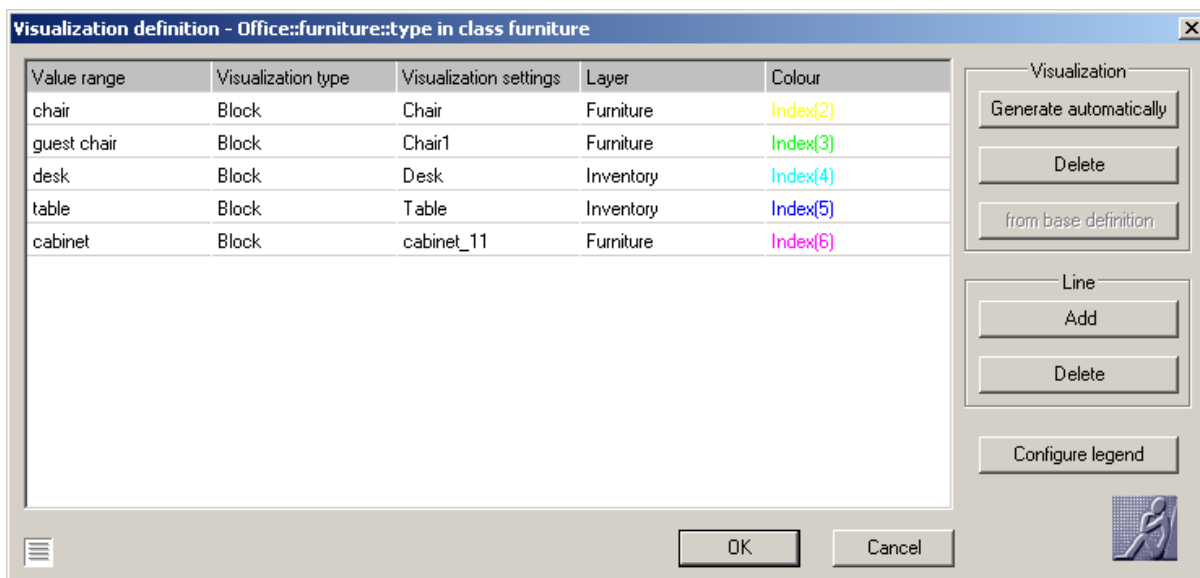
## Import → Visualizations

	class

This command is used to create feature data objects at the position of AutoCAD blocks that already exist in the drawing. This way one can quickly create feature data objects in large drawings. A label is created at each AutoCAD block reference of **different** block definitions. The labels are aligned feet to bottom. Feature attributes are filled in automatically according the assignment done in the definition of the visualization between the attribute and the name of the according AutoCAD block reference.

Before one can use the command, one has to define a visualization by blocks for that specific attribute. To do so please use the command **→ Define visualization** from the context menu of the attribute definition.

The figure below shows an example for the assignment of attribute values with blocks in the visualization definition.




In the column “Value range“ you enter the attribute values to which you assign AutoCAD block references from the drawing (column “Visualization settings”).

### Remark

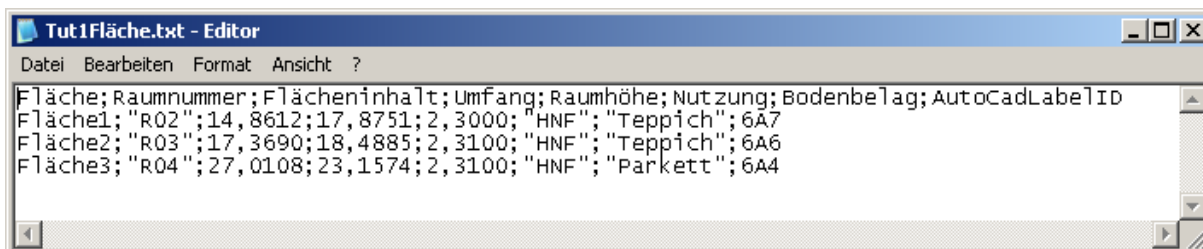
If you run the command **Import → Visualizations** and without wanting it you only see the object labels in your drawing instead of the AutoCAD blocks, then please switch them on by using the command **→ Visualization** from the attributes context menu.

## Import → Table

	Class

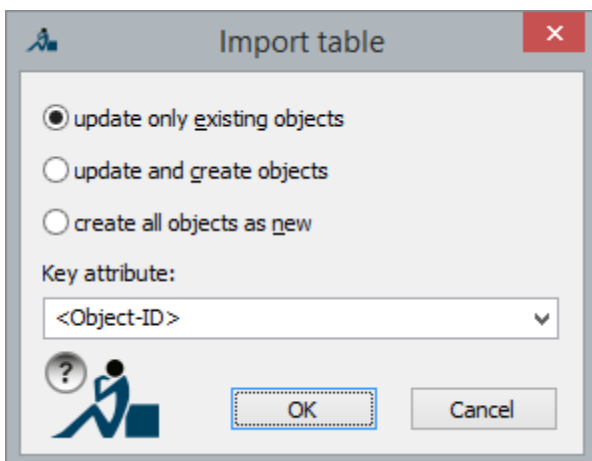
This command permits the import of an generally readable ASCII table. This table may have been exported previously by the feature data management or it may have been created in another application.

The basic format of the import table requires a header with the column name and data rows. Every row contains the data for exactly one object. With the help of the column names the according attributes will be identified within the structure definition. The columns may be in any order within the table. The table may contain additional columns, which will not be imported.



The settings, which have been adjusted under → **Data** → **Configure export...** within the feature data menu are valid for the decimal separator as well as for the separator for the columns.

After selecting the file with the table to be imported, you can select the extend of the import within the following window.



**Update only existing objects:** With the help of the selected key attribute (or the object ID), the corresponding object within the structure view will be determined for every data row. Its attributes will be imported from the table. Data rows without an according object will not be imported.

**Updating and generating new objects:** As extension of the previous option, data rows will be transformed into new hylasFM objects without having an existing object within the list.

**Regenerating all objects:** Without regarding a key attribute all data rows will be imported as new feature data objects.

The key attribute may be the object ID (The column name within the table has to be the same as the class name within the structure definition.) or an attribute with key characteristics. If there is no qualified key attribute available, only the option to generate new objects is available.

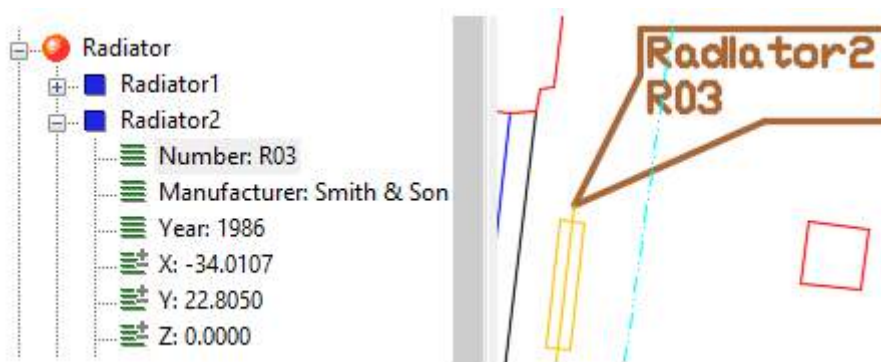
After the import a protocol will be shown, which displays mistakes that might have been appeared. You can find all affected objects within the object collector "Import mistake".

**Tip: Import with coordinates:**


Should it be necessary - as in the previous table example - to import the objects with their X, Y and Z position, then you should proceed as follows:

The class (here "Radiator") must be given the type "Label position" (Configure class → tab „Type", option "Label position"). Thereby the class is automatically given a few extra attributes, amongst others X,Y and Z. These are the coordinates of the label's insertion point.

When importing objects of this class from a table, the X,Y and Z attributes can also be read in and the labels will then be automatically positioned in the drawing with these coordinates.



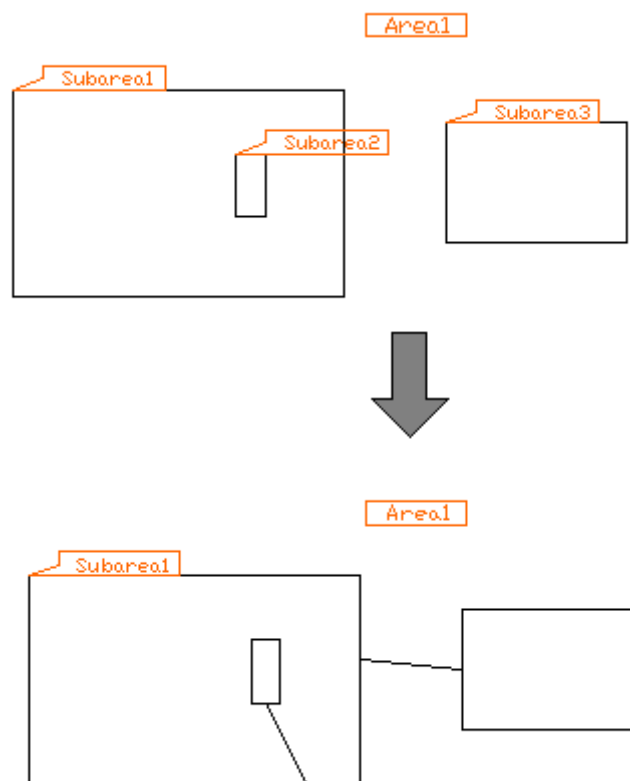
**Miscellaneous → Reduce polygon to one polyline each**

	Class
	only for the class "Area" and classes derived from it

With this command you can reduce all polygons that belong to an object of the selected class to one polygon. The respective polylines are connected by short joint.


This function is very particular and only suitable for export into certain systems. Some databases can only read one polyline per area.

An example: Here, room1 with three polygons has been converted into an object with only one polygon.



**Note:** This step cannot be reversed!

**Miscellaneous → Area proof**


	Class
	only for the class "Area" and derived classes

This command allows you to write an area proof into a text file in EXCEL format. Settings regarding the format have to be carried out under **→Configure export...**

In the area proof all subareas that are linked with the selected class by “Link” as well as their polyline interpolation points and surface areas are listed:

	A	B	C	D	E
1	Area	Subarea	Perimeter	SurfaceArea	
2					
3	Area1		17,875	14,861	
4		Subarea1	15,76	13,519	Factor: 1,000
5			P1	X=-33,8622788130934	Y=21,6217432423685
6			P2	X=-33,4745508837611	Y=24,3799248046818
7			P3	X=-31,5371833742681	Y=24,1470033356754
8			P4	X=-31,5417991463487	Y=24,1930712006793
9			P5	X=-31,4355234454381	Y=24,1854369393061
10			P6	X=-30,0821503035215	Y=24,0054545782347
11			P7	X=-29,9432535150903	Y=23,9879137070055
12		Subarea2	1,062	0,071	Factor: -1,000
13			P1	X=-32,9943651738431	Y=22,7298432713018
14			P2	X=-33,0260632678159	Y=22,4661884968503
15			P3	X=-32,7624084933644	Y=22,4344904028775
16			P4	X=-32,7307103993916	Y=22,698145177329
17			P5	X=-32,9943651738431	Y=22,7298432713018
18			P6	X=-32,9943651738431	Y=22,7298432713018
19		Subarea3	6,624	1,413	Factor: 1,000
20			P1	X=-33,4745508837611	Y=24,3799248046818
21			P2	X=-33,9709508543994	Y=24,4396048652843
22			P3	X=-34,103378741506	Y=23,3154297414823
23			P4	X=-34,1658931707581	Y=23,3127290580627
24			P5	X=-34,1810820226976	Y=23,2154319932404
25			P6	X=-34,3415592976318	Y=21,851716884296
26			P7	X=-34,3599738881994	Y=21,6816317069192
27			P8	X=-33,8622788130934	Y=21,6217432423685
28			P9	X=-33,4745508837611	Y=24,3799248046818
29			P10	X=-33,4745508837611	Y=24,3799248046818
30	Area2		18,488	17,369	
31		Subarea4	16,289	15,759	Factor: 1,000
32			P1	X=-34,2019578956888	Y=18,0241070483041
33			P2	X=-33,8604732164207	Y=21,151418007629
34			P3	X=-32,9232873997148	Y=21,0439561923674
35			P4	X=-31,9794864895524	Y=20,9357358616243
36			P5	X=-28,9010502563326	Y=20,5827489300128
37			P6	X=-30,1069959315415	Y=19,9704306567427

**Miscellaneous → Replace arcs within polygons by chord polygons**


	Class
	only for the class "Area" and derived classes

This command converts arcs into straight line segments within the according polyline.

The number of emerging line segments will be determined by the "Key\_Hylas\_Recognition\_VerticesPerMeter" key within the configuration file `FaroConfigCurrentUserCommon.cfg`.



**Miscellaneous → Delete unnecessary interpolation points within polygons**

	Class
	only for the class "Area" and derived classes

This command deletes all interpolation points of a polygon if they are placed on a straight line between two interpolation points or if they are congruent with other interpolation points.

**Delete all “NAME“ objects**

	Class


This command deletes all objects of a class if you answer the confirmation prompt with YES. This command **cannot be reversed**.

**Place label**

	Object
---	--------

This command allows you to add labels to objects that have been captured without a label. This situation occurs if the class configuration was set to “No label” and has been changed later. You are prompted to enter the insertion point of the label.

**Highlight \ Room, Area, Polygons**

	Object, Polygons
	only for objects of the classes "Room" and "Area" and derived classes

This command is used for the temporary visualization of individual laminar objects. It works like a switch. The current hatching state is indicated by a checkmark in front of the command.

By running the command again, the hatchings are hidden.

By calling up the command →**Highlight** from the context menu of a class, the hatchings of individual objects are overwritten.

**Note**

All hatchings that result from highlighting are deleted when saving and closing the drawing. The hatch patterns are intended for the temporary visualization of objects that have been recognized by the software and will therefore **not** be saved. Should you need permanent hatchings, please use the command → **Define visualization...**


**Show**

	Object

The command "**Show**" changes the display window in such a way that the object is shown in the center of the screen.

At first, the insertion position of the label is shortly indicated by an arrow. Then, the AutoCAD graphic belonging to the object is highlighted. Should the graphic not be visible in the current display window, the window is moved so that the respective position becomes visible.

**Show ID in label**

	Object
	for all objects that have a label

With this command the object ID can be shown or hidden in the respective label. The IDs of all objects of these classes are shown.

**Hint**

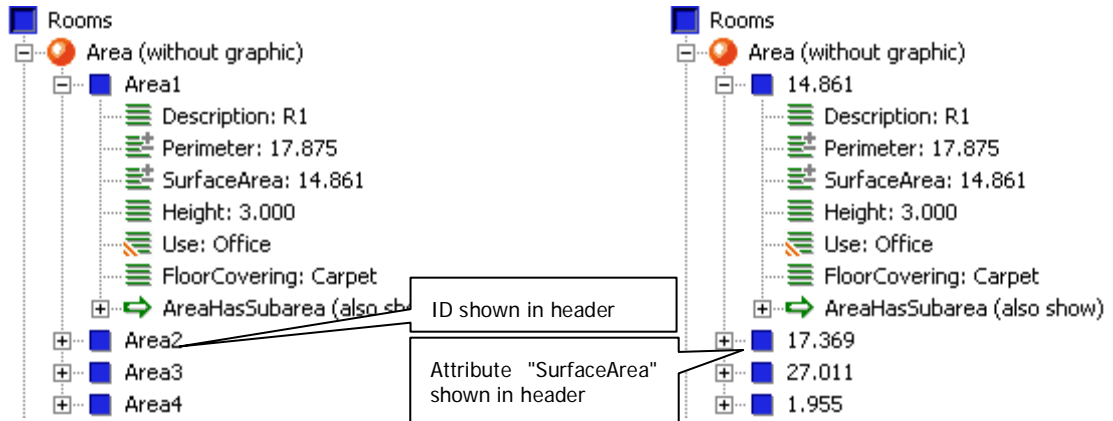
You can hide the labels of a class completely by hiding the IDs with this command and additionally hiding all attributes (→ Show in label).

**Show ID in header**

	Object


With this command the object ID can be shown in the header again.

This command is used in connection with the attribute command → **Show in header**. “Show in header” allows displaying any attribute instead of the ID in the header of an object. With “Show ID in header” you undo this action.



Also see → **Show in header** of attributes and → **Sort "Name" objects** of the class.

## Change ID

	Object

This command is used to rename object IDs.

When adding objects, the software automatically assigns a definite object ID. By default it is composed of the class name and a number. This command allows you to assign an ID of your choice to the objects.

However, we recommend adding an attribute of the type "Key" (e.g. room number) instead of changing the ID. You can also display this attribute in the structure view instead of the ID (→ **Show ID in header**) and hide the actual ID in the label (→ **Show ID in label**). This procedure maintains the coherent strategy of assigning IDs. Every object has the name of its class and carries a chronological number according to its creation.

### Remark

Valid IDs have to comply with the key property. I.e. they have to be unambiguous throughout the complete drawing. There are no two objects of a class with the same ID.

There are some restrictions for the syntax of object IDs: Keys start with a character or “\_” (underscore). Then any number of characters, numbers, “.” (dots), “-” (hyphens) and “\_” (underscores) can follow.




**Delete**

	Object
---	--------

The command “Delete” removes an object from the structure and deletes its object label in the drawing.

**Note:** This command cannot be reversed.

## Assign other graphic

	Object not available for objects of classes without graphic
---	--

With this command you may assign new or additional AutoCAD graphic to a feature data object.


In order to do so click an object within the structure view and call up "Assign other graphic" from the context menu. You will be prompted as follows:

Command: `_KUBT_HY_HYLASCONTEXT`

Current setting for other graphics: Replace

Select objects or [APpend]: *(Option "re" in order to replace existing graphic with new one.)*  
*(Option "ap" in order to assign additional graphic.)*

**Adopt link**

	Object

With this command you may add the links of other objects of the same class to an object.

After calling up this context menu command you will be prompted to select the objects whose links are to be adopted.

Adopt links of the following objects:

Select objects: *(Select labels of the class "NAME" within the drawing or objects within the structure view by clicking on them with the mouse.)*

The objects, selected within the structure view will be displayed in boldface. By clicking the object again you will reverse the selection.

Confirm your selection with **ENTER** or abort with **ESC** or **Ctrl+C**.

All objects, linked to the selected objects will now be linked to the reference object.

**Copy**

	Object

This command may be used for a fast copying of an object with all its attributes. The command line will read:

Current setting: copy linked graphic=ON

Current setting: copy link=ON


Insertion point for label or [Graphic/Links]:

The "Graphic" option determines if the CAD graphic attached to the object is to be copied as well.

If the object to be copied is linked to other objects and the option "Links" has been activated, the copies will also be linked to those objects.



The command will generate copies until it will be aborted with **ENTER** or **ESC**.

**Dissolve link to "NAME"**

	Object
	Only available for objects that are linked

This command releases an object from a link.

## Modify (inner point)

	Polygon
	only for Polygons

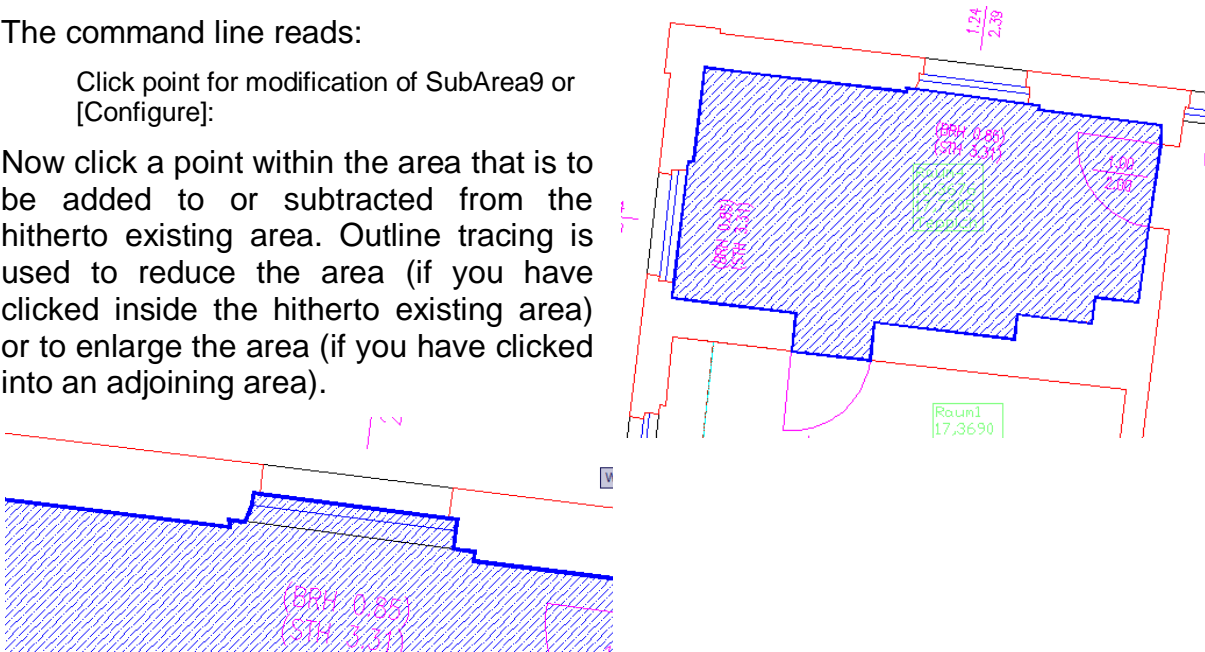
With this command you can extend or reduce an existing polygon.

A blue hatching with a blue outline appears that indicates the current extension or shape of the subarea. The width of the outline depends on the tolerance value, that can be set using the option "Configure", and visualizes the ability of the process to skip gaps.

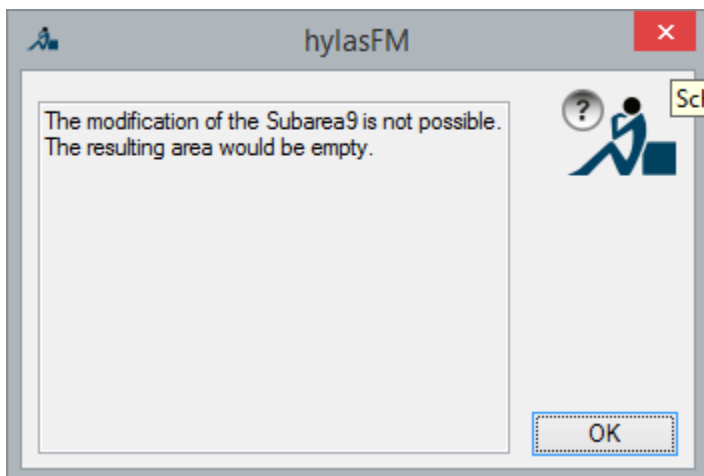
The command line reads:

Click point for modification of SubArea9 or  
[Configure]:

Now click a point within the area that is to be added to or subtracted from the hitherto existing area. Outline tracing is used to reduce the area (if you have clicked inside the hitherto existing area) or to enlarge the area (if you have clicked into an adjoining area).





The new polygon may not be empty. If you click into the area hatched in blue, so that the total existing area is recognized as subtraction area, following message is shown:



The setting option "**Configure**" basically shows the same behavior as within the command → **Add (interior point)** of an area. The difference is, that enclaves can not be recessed.

## Modify (interpolation points)

	Polygon
	only for polygons

With this command you can extend or reduce a polygon without using an already existing boundary line. The boundary of the modified polygon is determined by points you define yourself.

After starting the command you are prompted:

Command: HYLASCONTEXT

Click point on the boundary of the polygon:

*(Click on a point on the polygon boundary)*

Click additional new interpolation point or [Close]:

*(Click on further points for the new polygon boundary, the last point must lie on the polygon boundary again.)*

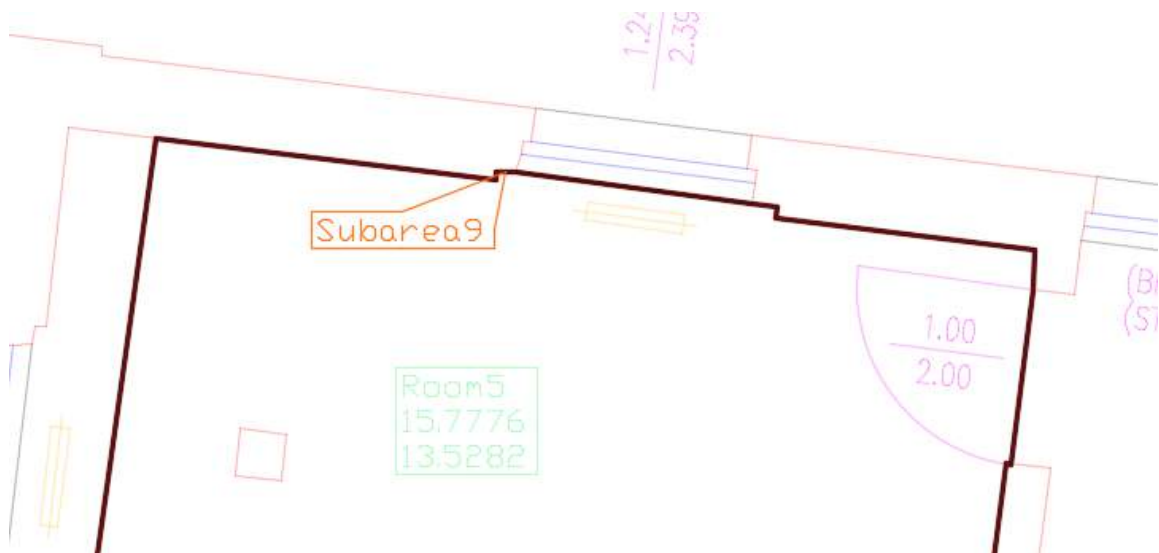
Both sides of the polygon are now hatched in red or blue and the following appears in the command line:

Click into the new polygon:

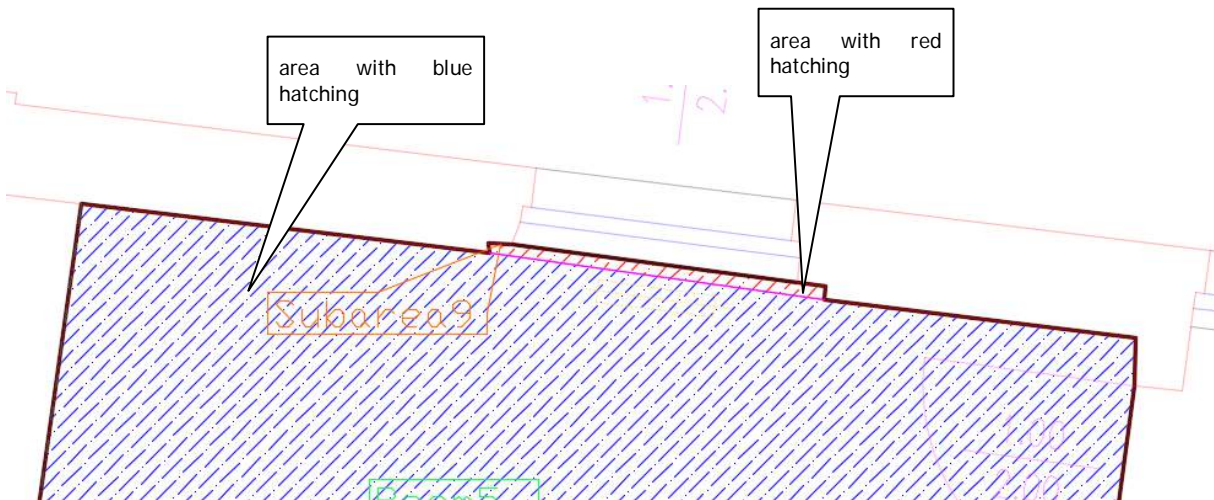
*(Click into the side of the hatched area that is to be kept.)*

Now, the shape of the polyline is being adapted.

This command is very useful if **→Modify (interior point)** cannot be used, as in following example



Here, the window reveals are NOT to be included in the polygon. Thus, something has to be cut from the polyline. To do so use this command and click on the two points at which the reveal starts. Following figure appears. A pink line separates the polygon into two areas – a smaller one with blue hatching and a bigger one with red hatching.



Click into the lower, big hatching area and receive the desired polyline.



Another typical application of the command “Modify (interpolation points)” is enlarging a polygon. If, for example, half of a threshold has to be added to the room area, the outline extension can be indicated by four new interpolation points in the plan above.



If you now click into the large left area, you will get the polygon that has been extended by half of the threshold.



**Assign other polyline**

	Polygon
	


You can assign a new 2D polyline to a polygon with this command.

Within the polyline you will be prompted:

Select new polyline for polygon::

*(select new graphic)*

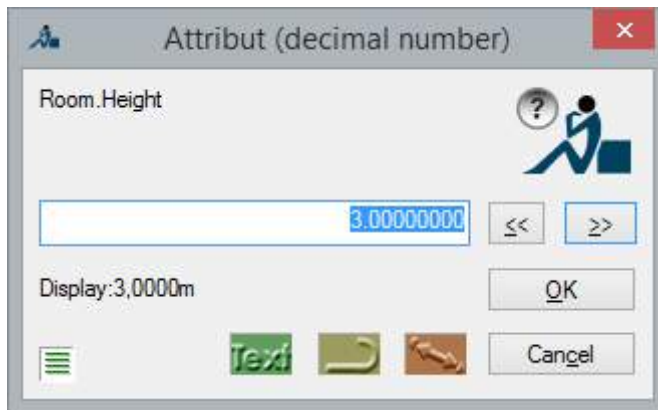
## Change

	Attribute
	only available for attributes that can be changed manually

This command allows changing the value of the attribute "Name".




Depending on the definition of the attribute different input windows appear. For some attributes the values can be taken directly from the drawing.

Here you see the window for changing an attribute of the type "Decimal number":



Enter the new attribute value in the input field and confirm your entry with **OK**. By means of the attribute definition the software then checks whether the entered value is valid or not.


The three coloured buttons below allow you to pick attribute values from the drawing:

-  You can select curved lines, whose length shall be used.
-  You are prompted to pick two points in the drawing, the distance between them shall be used.
-  You are prompted to pick two points in the drawing, the distance between them shall be used.

### Remark

The attribute value can also be changed by double clicking on the respective attribute.

**Update**


	Attribute
	only available for calculation attributes

By means of this command the calculation attribute will be recalculated.

Calculation attributes (→**Add attribute** within the definition) cannot be changed manually since their value results from other attributes (e.g. surface area from width x height). The (sometimes time consuming) calculation of an attribute can be initiated with this command.

If **all** calculation attributes are to be updated, use →**Check/Create consistency...**

**Show in label**

	Attribute
	only available for objects with label


With this command the attribute of an object can be shown or hidden in the corresponding label.

The attributes of **all** objects of a class are shown or hidden.

Similar to the command →**Show ID in label**, this command works as a switch. Any attribute can be shown in the object label.

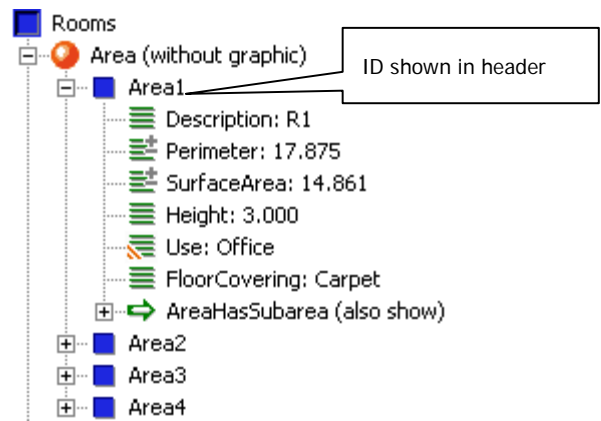
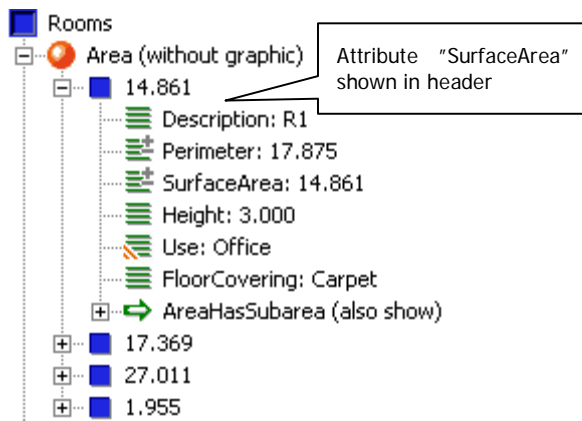
The order of the attributes in the label depends on the order of the attributes in the structure tree. (see →**Sort attributes, show/hide...**)

**Show in header**

	Attribute


With this command an attribute instead of the object ID can be displayed in the structure view. Thus it is possible to sort the objects by a certain attribute (→Sort "NAME" objects (→ascending, →descending)).

This command is used in connection with the command →**Show ID in header**. With "Show ID in header" you switch back to the object ID.





**Filter objects with this value**

	Attribute


This command allows you to collect all objects that have exactly this attribute value within an object collector. A window appears which displays the name of the object collector and the number of collected objects. You find it in the structure tree under object collector .

Object collectors serve to edit several objects at the same time. For more information please refer to chapter →**11.3 Commands of the context menu of the object collector.**


## Visualization...

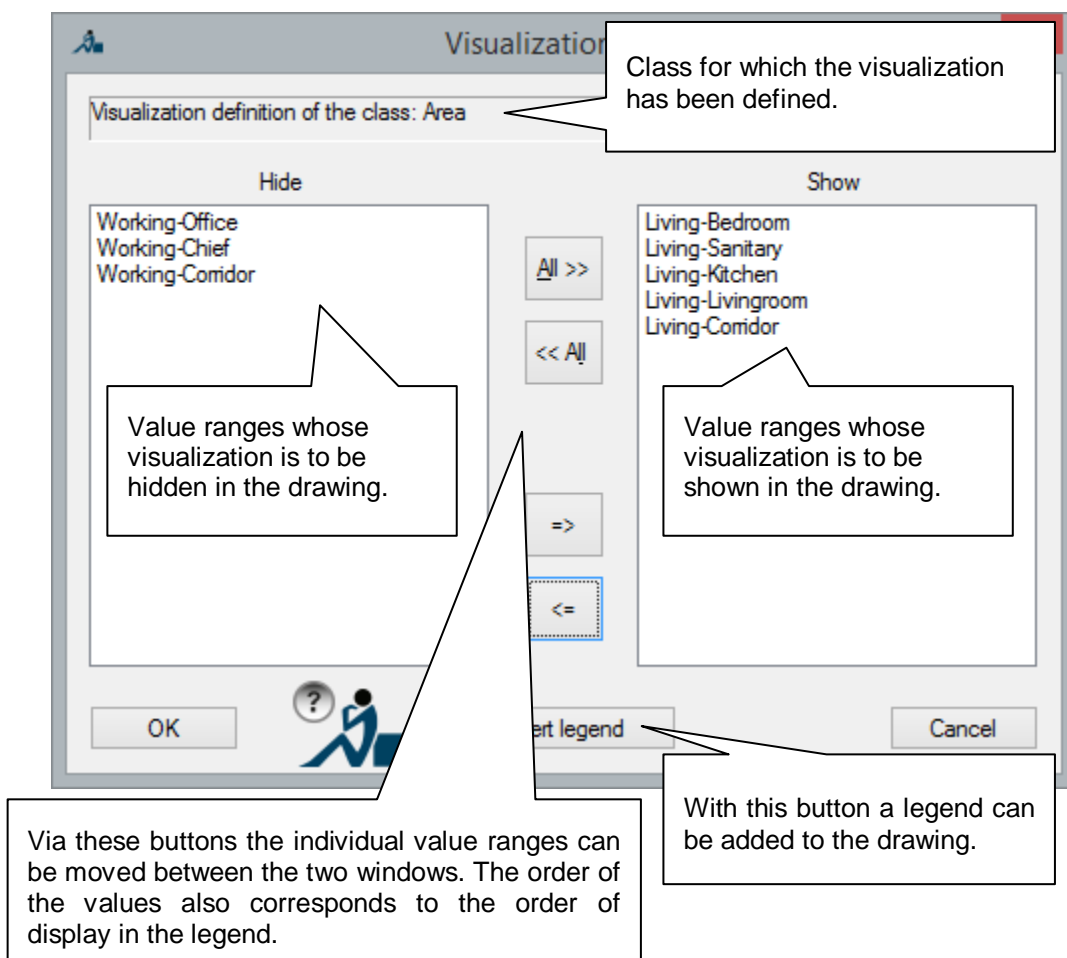
	Attribute
	Attribute of a class with label or an area or curve class

This command allows you to show or hide an existing attribute visualization.

If a visualization definition does not exist for this attribute () , you will be prompted, if you want to generate one. If you confirm that, the definition window will open. You will find further information on that under → **Define visualization...**

An attribute visualization describes which attribute values are to be displayed in which way.

If a visualization definition already exists () , following window opens:



Via this window you can show or hide individual or all value ranges of the visualization.



Via the button “Insert legend“ a legend can be added to the drawing by entering an insertion point. If you change the visualization definition or the data, already inserted legends are **not** updated.

You find the description of how to configure a legend under → **Define visualization.**

With OK the visualization is created or updated. The aim is always to keep the correct order of display of the hatchings (as described under the command ➔Visualization ➔ Update order of display).



**Link with “NAME“**

	Link
	Backward link

This command is used to link one or more objects with a reference object. The reference object is the object that contains the link on which you call up the command.

After running this context menu command, you are asked to select the objects to be linked.

Select “Name” label:

Select objects:



*(Select labels of the class "NAME" in the drawing  
or objects in the structure view by mouse click.)*

The objects selected in the structure view appear in bold print. By clicking an object again, you undo the selection.

Confirm your selection with **RETURN** or cancel with **ESC** or **Ctrl+C**.

The selected objects will now be linked with the reference object provided their type (class) is compatible with the link. This applies for all objects of the class “NAME” and classes derived from it.

**Add and link "NAME" → Add "NAME"**

	Link
	Backward link

If a new object is to be created and linked immediately with the reference object, this command can be used. Let us assume that an object Room1 and a link RoomHasEquipment already exist. Then, equipment can be added and immediately be linked with the room via the context menu entry of RoomHasEquipment of Room1.

When selecting this option, a submenu appears with all available methods of creation (e.g. "Add equipment").


### 11.3 Commands of the context menu of the object collector

Sometimes, an attribute has to be changed for several objects at the same time. For this purpose the concept of object collectors exists. The objects are mirrored to an object collector as *reference* and can be edited there together.

Object collectors are also applied when a feature data command selects several objects. If, for example, the consistency check (**→Check/Create consistency...**) is enabled, it might happen that several objects are detected as inconsistent. These objects are then automatically moved into an object collector as reference and can then be adjusted systematically.

**Note:** Object collectors exist only as long as a drawing is open.


**Add object collector**

	Structure name (Object collector)
	<input type="text"/>

This command creates a new object collector.


Using **→Add objects** you can edit objects together.

**Remove all object collectors**

	Structure name (Object collector)  
---	---

This command deletes all object collectors. The objects themselves are not deleted only their *reference* within the object collector is removed.

**Add objects**


	Object collector
---	------------------

With this command objects can be added to the object collector in order to be edited together. There are two methods of selecting objects that can be combined freely:

- Selection of the objects in the drawing window - Select the label.
- Selection of the objects in the structure view - Open a class and click on objects. By pressing **Shift** or **Ctrl** at the same time parts or individual objects can be marked.


Finish the selection with **RETURN** or by clicking into the drawing window with the right mouse button.

**Remove all objects from object collector “NAME”**

	Object collector

All objects are removed from the object collector. The objects are **not** deleted - only their *references* are removed from the object collector.


**Rename “NAME“**

	Object collector

You may rename the object collector with this command.




**Delete “NAME”**

	Object collector
---	------------------

With this command the current object collector is deleted. The objects are not deleted, only their *references* within the object collector are removed.


**Delete all objects permanently**

	Object collector
	only available if the object collector contains objects

With this command the objects are **deleted permanently**. Not only the references within the object collector, but the actual objects in the classes, too, are deleted. A security prompt appears.

**Caution!** This command cannot be reversed!

**Remove from object collector**

	Object
	Only available for objects within the object collector

**The object will *not* be released.** Only the pointer to the object within the object collector will be deleted.

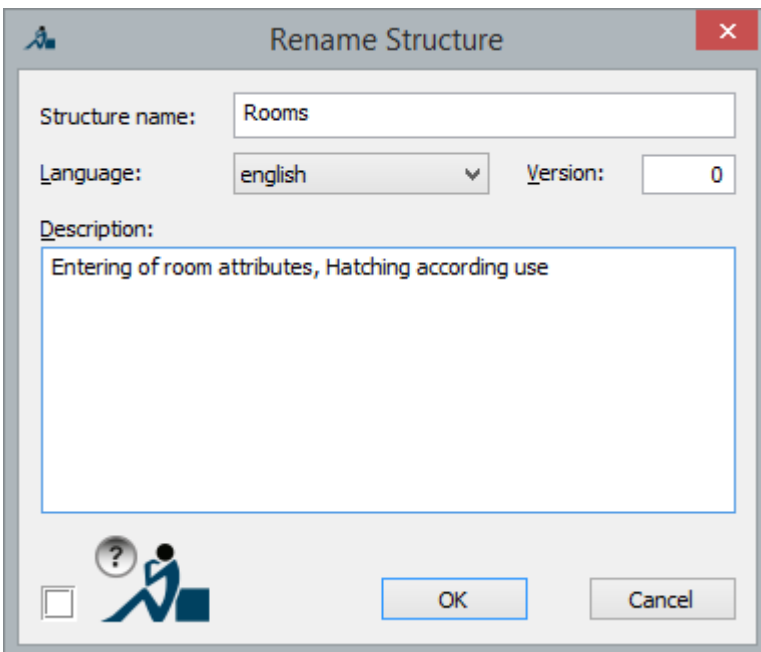
## 11.4 Commands of the context menu of the structure definition

### Rename structure

	Definition

This command allows you to rename the structure definition. Renaming the structure definition is recommended after it has been changed (new class, new attribute or new link). By doing so, the difference to an already existing drawing with the old structure definition can be pointed out explicitly.

Also add an informative description in the corresponding field. This description will be visible when loading the structure definition and, thus, makes the selection of a suitable structure definition easier.




There are two ways of applying the change of a structure definition to other drawings:

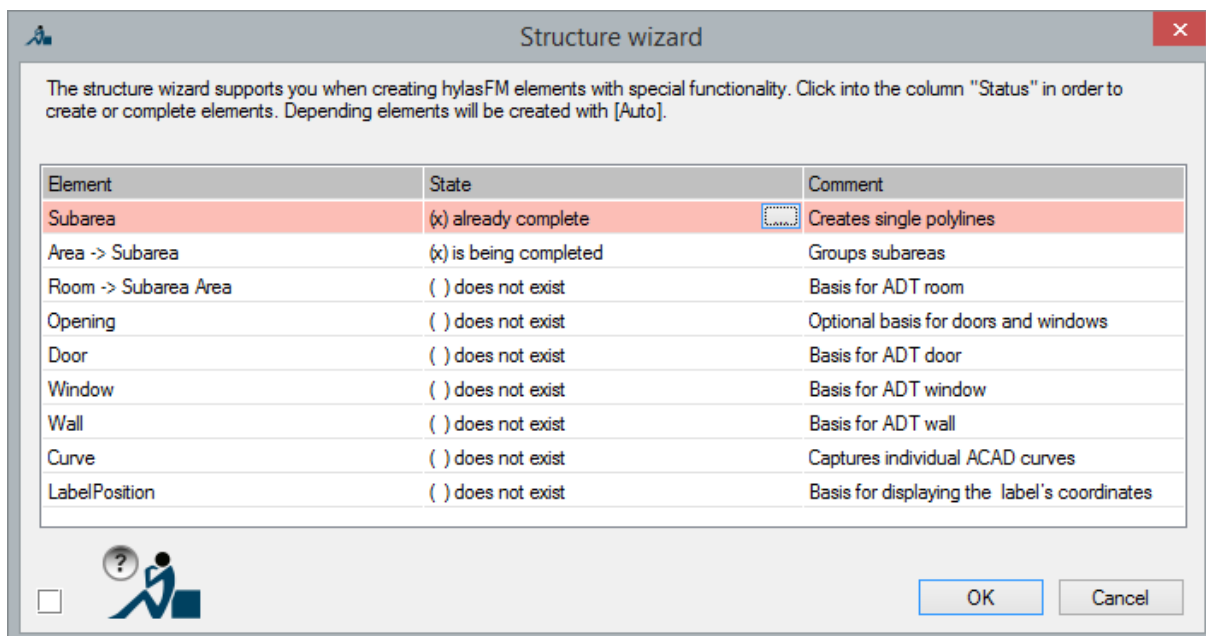
1. The structure definition can be saved in the original drawing (→ **Structure** → **Save structure definition...**) and then be loaded in the other drawings (→ **Structure** → **Load structure definition...**).
2. Any objects of the new/changed classes are created in the original drawing and then are copied into the other drawings using the feature data clipboard (→ **Copy** → **Paste**). The structure definition there will automatically be adapted. The objects can then be deleted immediately.

It is recommended to test a structure definition before starting the actual capturing of data so that a change during the run of a project can be avoided.

## Structure wizard

	Definition
	11.4.1

With this command you can create or complete classes with special functionality. When selecting the entry in the context menu, following window opens.




In the column “Element” the predefined classes are to be found. After an arrow there may stand other classes, whose existence is necessary for the respective element. The class “Area” requires the class “Subarea” in order to be used sensibly.

By clicking into the column “Status”, a button [...] appears in the respective line. If you click this button, the status is changed. Following entries are possible.

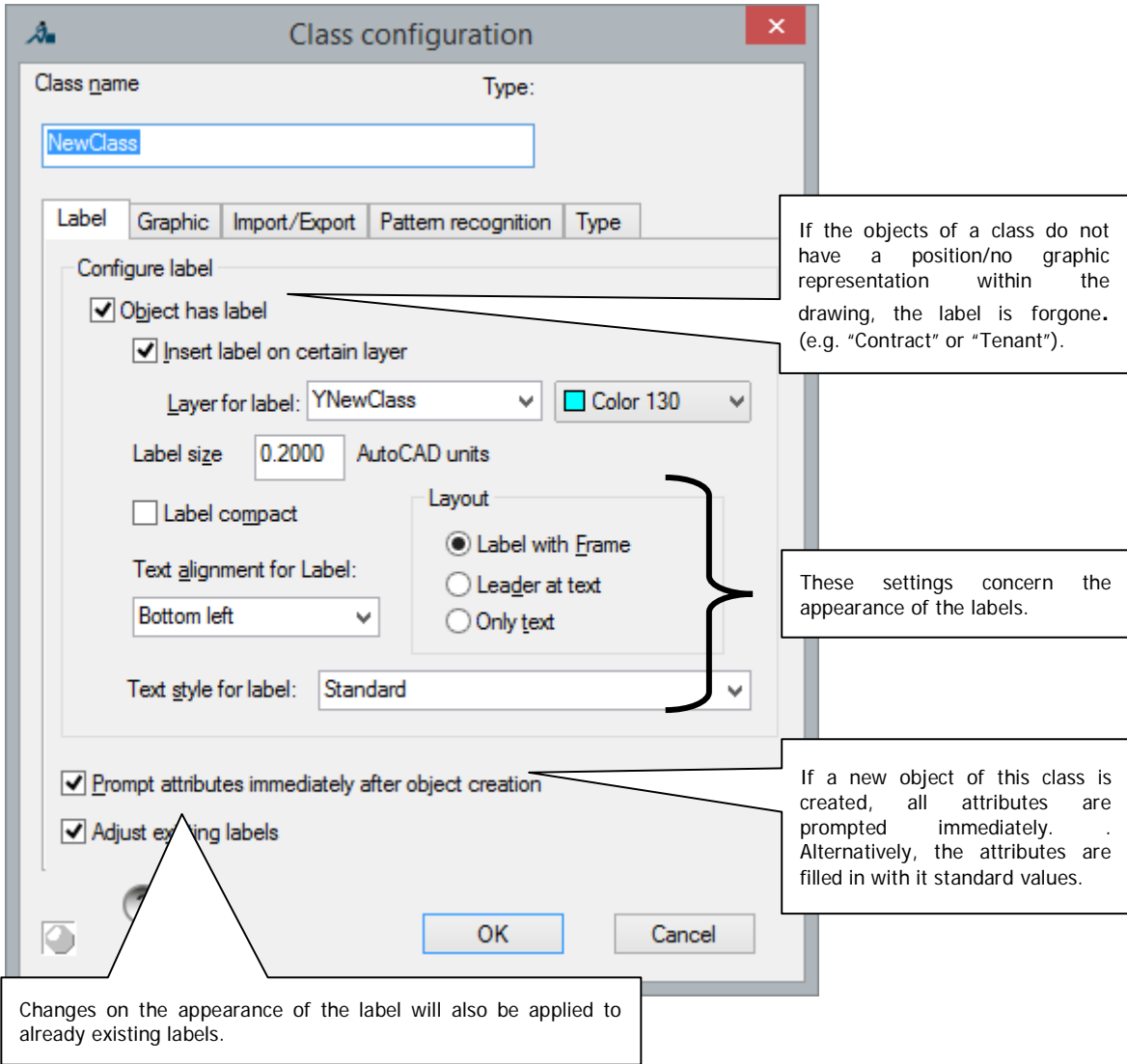
<b>already complete</b>	The class with all its attributes already exists in the structure.
<b>exists incomplete</b>	The class already exists in the structure but at least one attribute is missing.
<b>does not exist</b>	The class does momentarily not exist in the structure.
<b>is being created</b>	The class is momentarily not available in the structure, but will be created with all its attributes upon leaving the dialogue.
<b>is being completed</b>	The class already exists in the structure. The missing attributes will be added.

When creating or completing a class, the classes that are required for the class are created as well. The status of these classes to be created automatically gets the addition “(Auto)”. In the figure above, for example, the class “Area” is to be created. The class “Subarea” is a prerequisite and will thus also be created.

**Add class**

	Class definition
	only available for the structure entry <b>"Classes"</b>

With the command "Add class..." you define a new class.



The screenshot shows the 'Class configuration' dialog box with the following settings and callouts:

- Class name:** NewClass
- Label tab:**
  - Object has label
  - Insert label on certain layer
    - Layer for label: YNewClass
    - Color: Color 130
  - Label size: 0.2000 AutoCAD units
  - Label compact
  - Text alignment for Label: Bottom left
  - Text style for label: Standard
  - Layout:
    - Label with Frame
    - Leader at text
    - Only text
- Prompt attributes immediately after object creation
- Adjust existing labels

Callout 1 (top right): If the objects of a class do not have a position/no graphic representation within the drawing, the label is forgone. (e.g. "Contract" or "Tenant").

Callout 2 (middle right): These settings concern the appearance of the labels.

Callout 3 (bottom right): If a new object of this class is created, all attributes are prompted immediately. Alternatively, the attributes are filled in with it standard values.

Callout 4 (bottom left): Changes on the appearance of the label will also be applied to already existing labels.

Enter the name of the new class in the field "**Class name**". If the desired name is already in use, you will receive an error message when leaving the dialogue with OK.

On the tab "Label" you can configure the properties of the labels of this class.

**Tab Graphic**

A feature data object is often linked to an AutoCAD graphic. For example, "Subareas" reference 2D polylines. In general the already existing AutoCAD graphic is requested when creating an object. However, sometimes you may want to insert a special AutoCAD block when creating an object (e.g. a door symbol).

In any case a certain layer is to be assigned to the CAD graphic that is linked to the object. You can make the respective settings on the tab "Graphic".

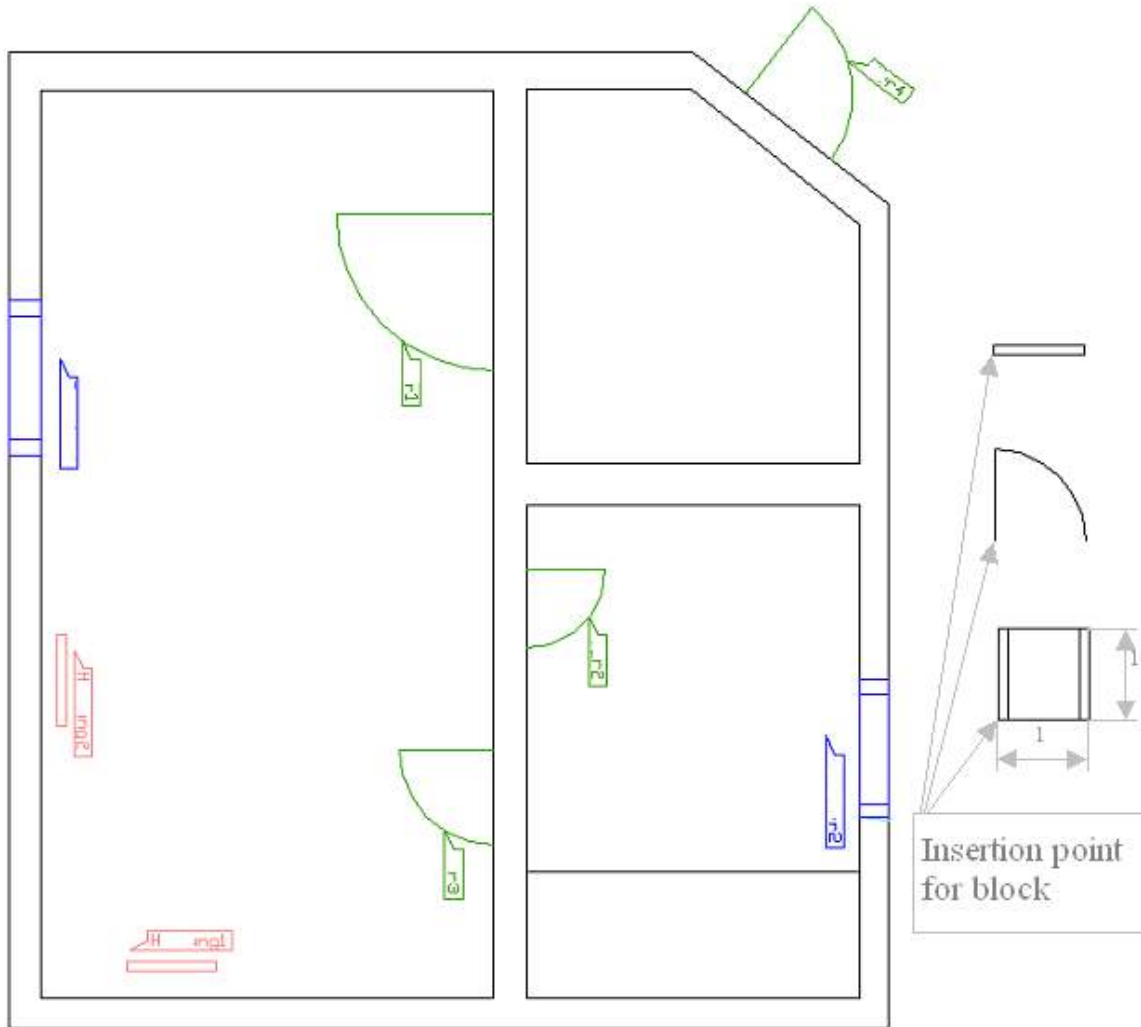
The screenshot shows the 'Class configuration' dialog box with the following settings and callouts:

- Class name:** NewClass
- Type:** (empty)
- Graphic tab:** Selected
- Graphic linked with object:**
  - Link object with graphic (AutoCAD drawing elements)**
  - Type of linked graphic:**
    - any AutoCAD drawing elements
    - Block (Block name: [dropdown])
  - Block insertion method:**
    - Request insertion point
    - Align at object
    - Fit block
    - Place automatically
- Copy graphic
- Move graphic to certain layer (Layer: [dropdown], Color: Black)
- Also delete graphic if object is deleted
- Adjust existing graphic
- Callouts:**
  - Objects of this class are linked with an AutoCAD graphic.
  - If this option is enabled, an AutoCAD block is always inserted when creating an object.
  - Methods for inserting and aligning the block during the object creation (see below)
  - The selected graphic is copied. This copy (not the original) will be linked with the object.
  - The linked graphic remains on its original layer or will be created on the current layer.
  - The linked graphic is being put on this layer.
  - Changed settings, e.g. the change of the layer, shall be taken over for already existing objects that are linked with graphic.
  - If an object is deleted, its linked graphic deleted as well.

If the option “Block” under “Type of linked graphic” is enabled, you can make use of a powerful mechanism for the insertion of blocks.

The insertion method “Request insertion point” always creates the block parallel to the X axis and without scaling. Following figure illustrates the two more interesting insertion methods “Align at object” and “Fit block”.

Door blocks that are scaled according to their inner width are to be inserted. We also want to insert window blocks that are scaled according to their inner width *and* the wall thickness. The insertion method “Fit block” can handle both cases. The method “Align at object” is used to align radiators at walls but without scaling them.



On the right in the figure you can see the block definitions (from top to bottom, blocks of the classes “Radiator”, “Door” and “Window”). On the left there is a plan with inserted feature data objects. The blocks in the plan have been inserted simultaneously with the creation of the feature data objects and their labels.

The insertion method “Place automatically” inserts the block at the same position as the feature label without further scaling and rotating.



## How do the blocks have to be constructed in AutoCAD?

If the insertion methods “Align at object” or “Fit block” are to be employed, the following needs to be considered

- The *principal axis* of the block corresponds to the X axis in the drawing. The block will later be aligned and scaled at this principal axis.
- The lower left corner is selected as block insertion point.

For the method “Fit block” additionally applies the following

- The width of the block is one AutoCAD unit.

The insertion of a door is carried out using the context menu entry → **Add door** of the class “Door”. The following appears in the command line

Insertion point for label:	<i>(Point for door label)</i>
Insertion point for block:	<i>(Point at door hinge)</i>
Point for x-alignment and dimension:	<i>(Point at door handle)</i>
Point for y-dimension: or [ofX/Numerical value] <ofX>: X	<i>(Option [ofX], since extension in y-direction is to be the same as in x-direction)</i>
\\Scaling will be taken over from x-direction.	<i>(only with option [ofX])</i>
Point for side:	<i>(only with option [ofX]: Point in opening direction)</i>

The door label is aligned along the X axis.

Due to the shape of a square of the window block definition (1m x 1m), the window can be precisely fitted into the wall.

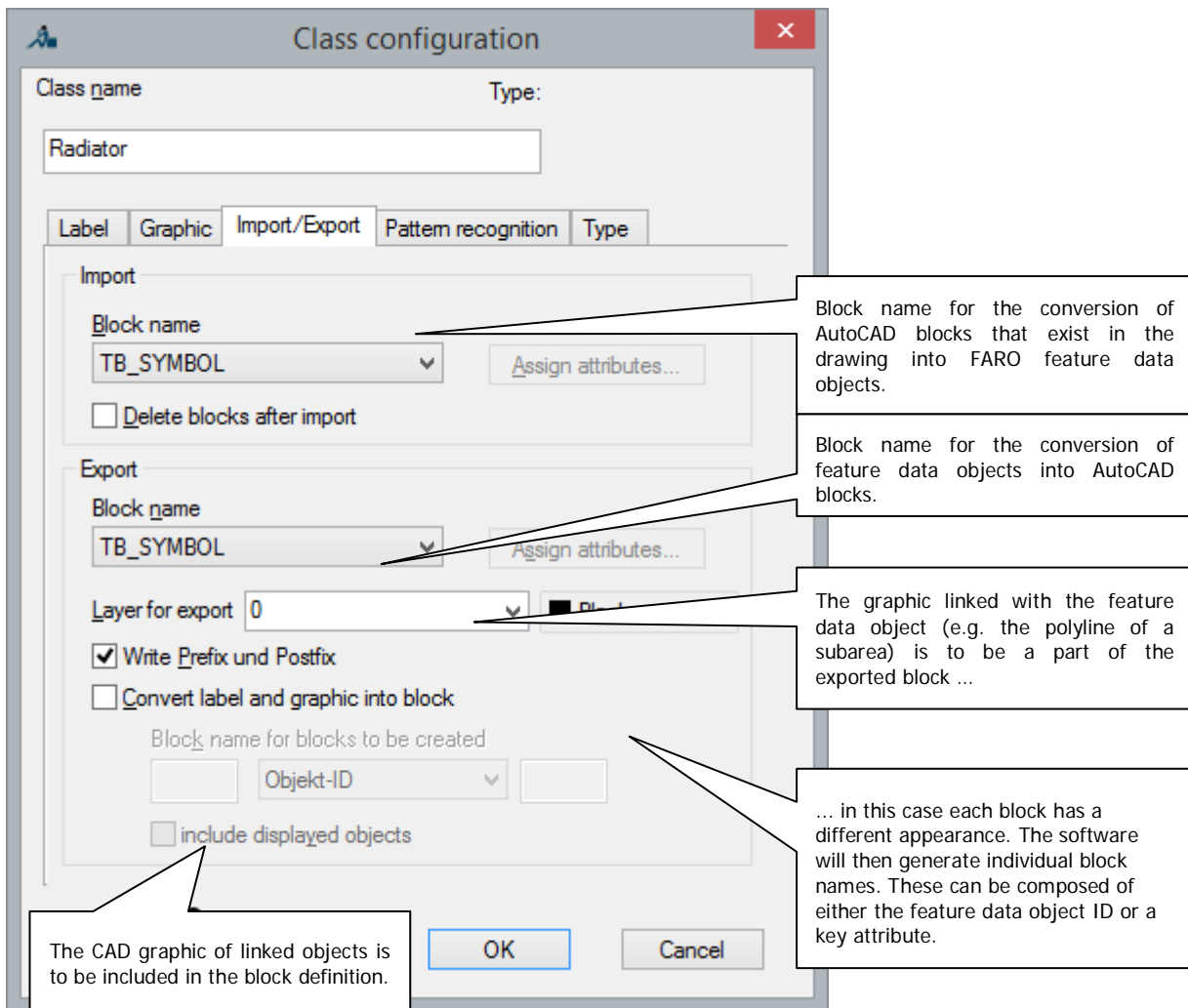
The insertion of a window is carried out with context menu entry → **Add window** of the class “Window”. The following appears in the command line

Insertion point for label:	<i>(Point for window label)</i>
Insertion point for block:	<i>(Point at window corner)</i>
Point for x-alignment and dimension	<i>(Point at window corner and on same wall side)</i>
Point for y-dimension: or [ofX/Numerical value] <ofX>:	<i>(Point at on wall side)</i>

The insertion of radiators is carried out analogous.

## Tab Import/Export

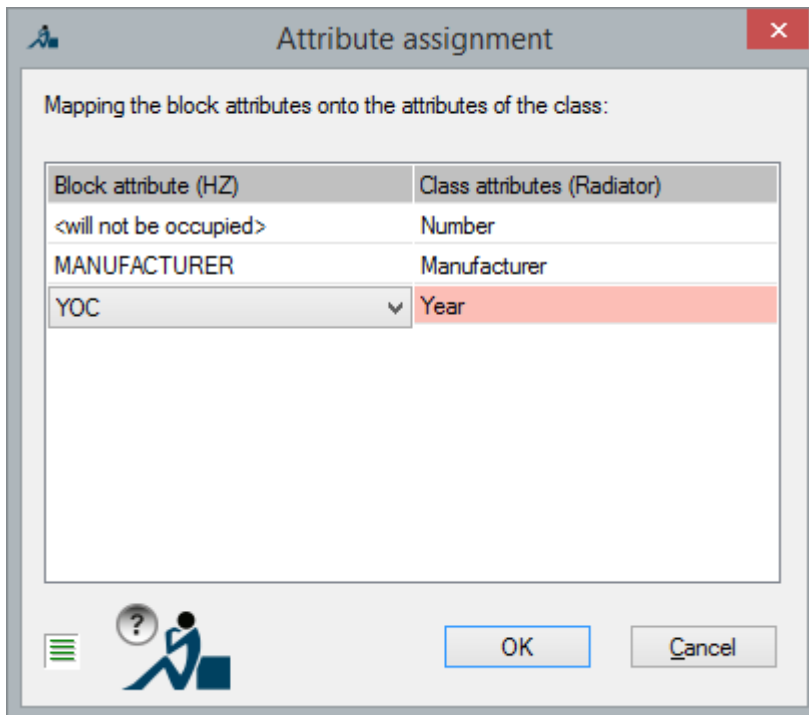
Sometimes it is required to create ordinary AutoCAD blocks from feature labels. This can be defined via the export functionality. On the other hand, there is also the possibility to import AutoCAD objects that already exist in the plan as feature data objects. The block names and the attribute allocation for these import / export functions are defined on the tab "Import/Export". The conversion itself is carried out later using the context menu of the class with **→Export→Blocks** or **→Import→Blocks**.



For the export of blocks you can also set whether the existing graphic (e.g. the room polygon) is to be included in the block. In this case enable "Convert label and graphic into block". Since every block will look different using this option, the software has to create a new block definition – with individual name - for every reference. You can determine the middle part of the block name yourself using a key attribute or object ID.

For "Areas" and derived classes (e.g. "Rooms") polylines of the related "Subareas" are often to be included in the export block. In this case enable the option "Include displayed objects". Objects are displayed if this option has been set within the link definition (see context menu of the link definition **→Configure... (links)**).

If the block has attributes, they can be assigned to the FARO feature attributes using the button “Assign attributes...”:



Just a few words about dynamic parameters:

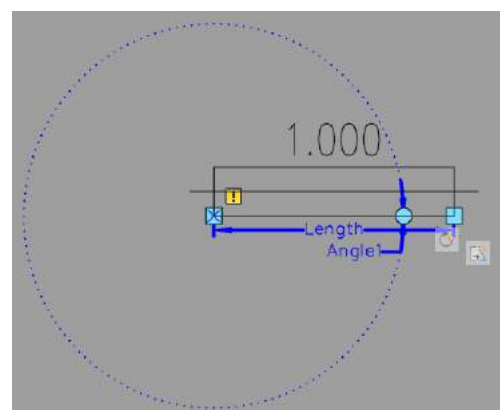
These are to be found in so called **dynamic blocks**. In AutoCAD it is possible, with the help of these, to display blocks dynamically in the graphic window. The content and appearance of a block can be changed directly on the block using its own grips, for example it is later possible to alter the length of a "Radiator" block or the size of a symbol or the hatch type etc.

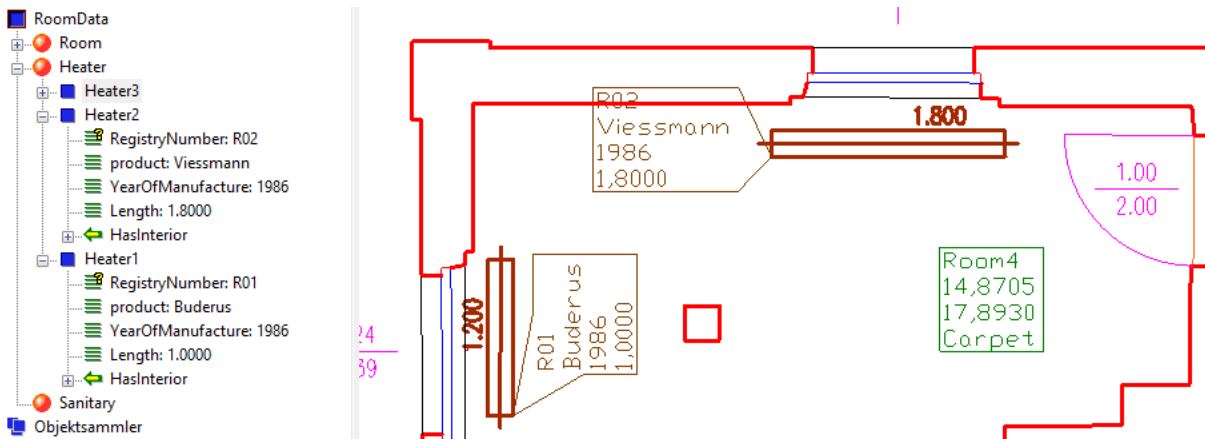
The control of these dynamic properties is achieved using "Parameters".

When using import / export of blocks to / from FARO feature data it is possible to enter these parameters for the attributes.

#### An example:

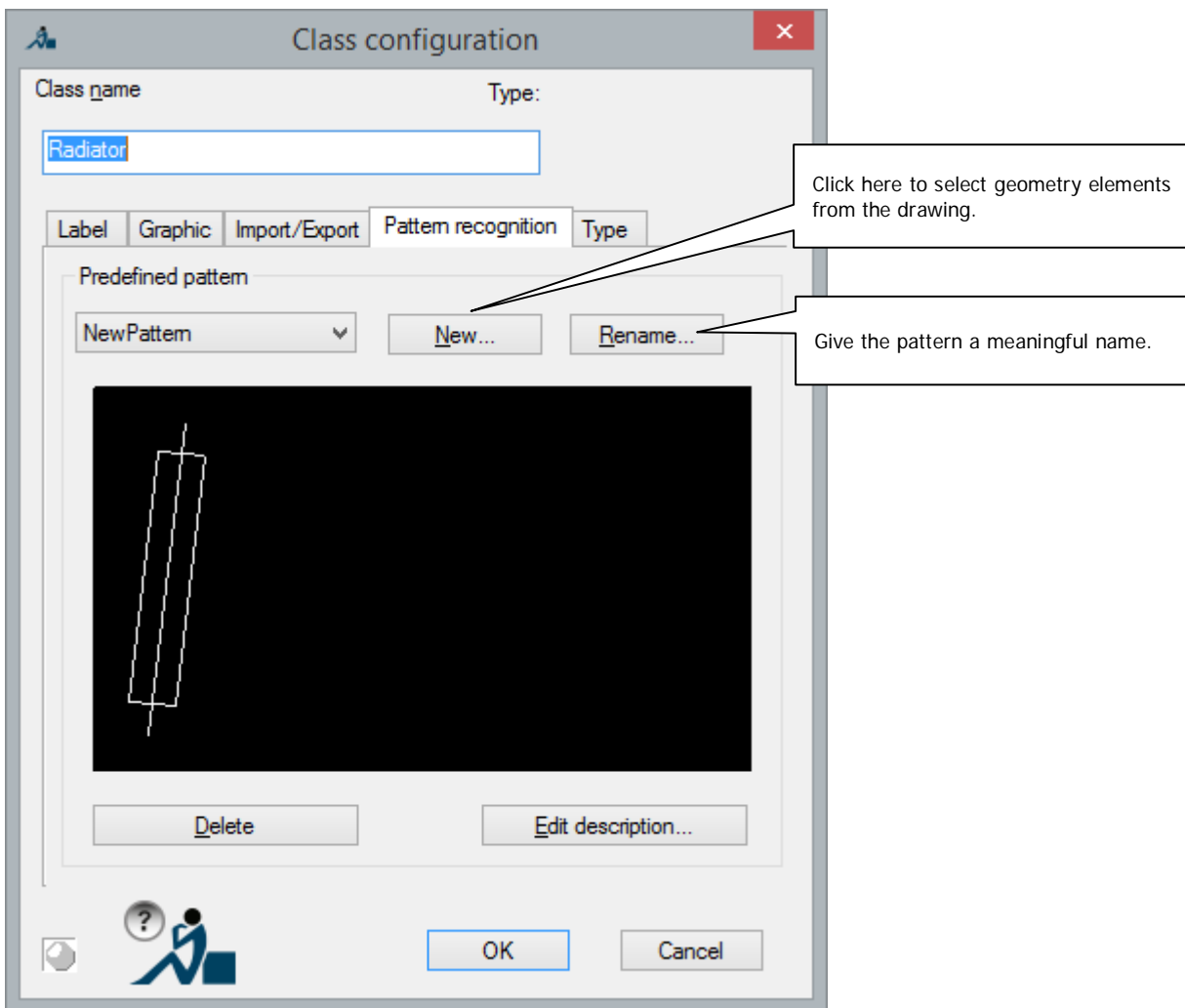
A radiator has the parameter "Length", which controls the length of the radiator. If there are many block references in the drawing with different length values (thereby radiators of different lengths), then it is possible to transfer this information directly into a FARO feature attribute named "Length".





### Tab Pattern recognition

If there are recurring patterns in plans that are, however, not AutoCAD blocks (if so: see above), one feature data object each can be assigned to these patterns with the feature data management, too.



With the button **New...** you select the pattern in the CAD graphic. Before you do that you should have set the tolerance value to a suitable, not too small value (0.02m is recommended) (Feature data menu → **Configuration...**) since this value is saved

during the recording of the pattern and otherwise not all recurrences of the pattern are recognized in the plan later.

If you have made all settings leave the dialogue with **OK**.

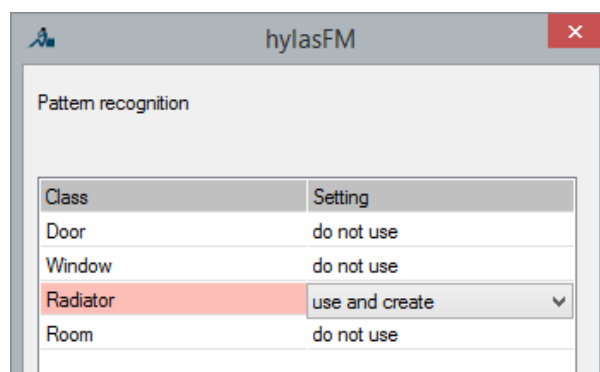
The pattern recognition is put to use in two different ways

7. There is now a new entry in the context menu of the class, e.g.

"...  Radiator → Add radiator (Pattern recognition)"

With this function a single object of this class is created. You are supported by the pattern recognition when assigning the graphic. To learn more refer to section **→Add "NAME" (Pattern recognition) \ Window, door, wall.**

8. With the command **→Pattern recognition→Interpret plan...** bigger parts or the complete plan can be analyzed. The complete section is analyzed for the pattern and a feature data object of the respective class is created for every pattern found.



For more information on this matter, refer to section **→Pattern recognition→Interpret plan**

### Tab Type

Classes may get a special functionality when the type has been set accordingly.

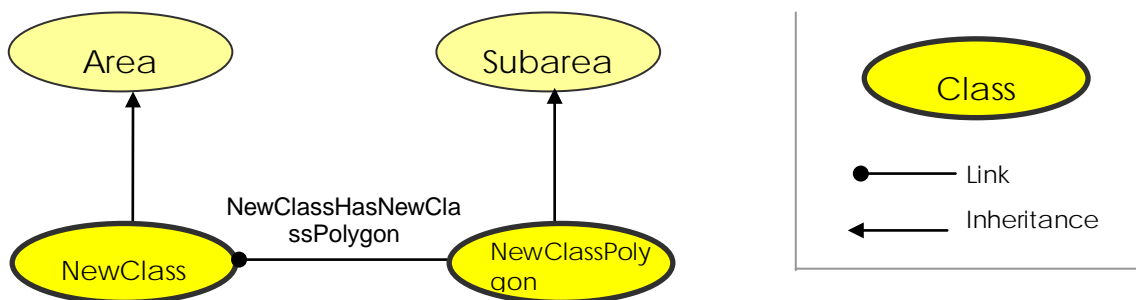
The "Area" type is of essential importance within hylasFM. With it, an efficient generation and management of area objects is possible. An area combines any number of polygons (each represented by a polyline). This is supported by topologic algorithms like enclave recognition and skipping gaps.

Settings for the polygons of an area:  
**Copy polygons:** selected polylines will be copied during the transfer into hylasFM and the originals will stay as they are.  
**Move polygons/enclaves:** optionally you may move polygons to a certain layer while they are being transferred into hylasFM area objects.

The "Curve" type makes the management of line-like objects easier. The selected graphic will be transferred into individual polylines. The class will be extended by an attribute "Length", which states the length of the polyline, belonging to the object.

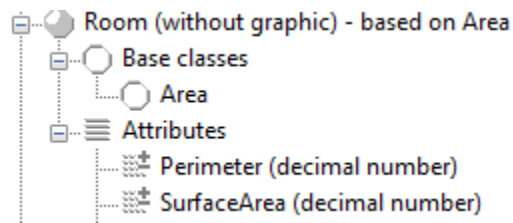
The "Label position" extends a class by X-, Y- and Z-attributes, which state the coordinates of the label insertion point.

When selecting a type of a class, an according base class will be added in the background. If you select "Area" as type, the following class structure will be added automatically:



But only the "NewClass" class is relevant while working. On behalf of the longterm users the structure, seeming so complicated, has been kept. This has been done in order to ensure the compatibility of already running projects (and also for the exchange with partners who use former versions).

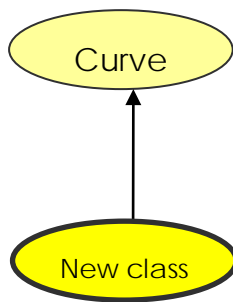
It looks like this in the tree structure:



The attributes “Perimeter” and “SurfaceArea” appear automatically. They are inherited from the class “Area”.

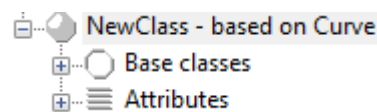
### Type “Curve”:

If the type “Curve” is selected, the following class structure is automatically created:



The class inherits a special property from the base class “Curve”, that precisely one polyline may belong to one object and that automatically created attribute “Length” defines the length of this polyline.

It looks like this in the tree structure:



### Type “Labelposition”:

Type “Labelposition” extends a class with the attributes “X”, “Y” and “Z”, which are the coordinates of the label insertion point.

For special use cases this type also offers further attributes for calculating the minimum perimeter of the appended graphic object.

If these attributes are not required they can also be deleted.

Further information to the special classes “Curve”, “Area” and “Split face” can be found in section **4.2.1 Classes with special functionality**.

### Comments on creating a new class

There are two methods to apply the change of the structure definition – the creation of a new class is such a change – to other drawings that still contain the old structure definition:

1. The structure definition can be saved in the original drawing (→ **Save structure definition...**) and can then be loaded into the other drawings (→ **Load structure definition...**).
2. Any objects of the new/changed classes are created in the original drawing and then are copied into the other drawings using the feature data clipboard (→ **Copy** → **Paste**). The structure definition there will automatically be adapted. The objects can be deleted right away.

It is recommended to test a structure definition before starting the actual capturing of data so that a change during the run of a project can be avoided.


also see → **Rename structure** and → **Configure... (class)**

To add a base class B to a class K, class K has to be created first. After the creation there is an entry base classes underneath class K in the structure tree. The context menu of this entry allows adding the base class B.

see → **Add base class**.



**Insert copy**


	Class definition
	only available for the structure base " <b>Classes</b> "

This command inserts a previously copied class definition from the clipboard.

In connection with the command → "**Copy**" it is possible to:

- create a copy of a class in one and the same structure definition.
- copy classes from one structure definition into another one.

## Sort classes, show/hide...

	Class definition
	only available for the structure entry "Classes"

With this command you may:

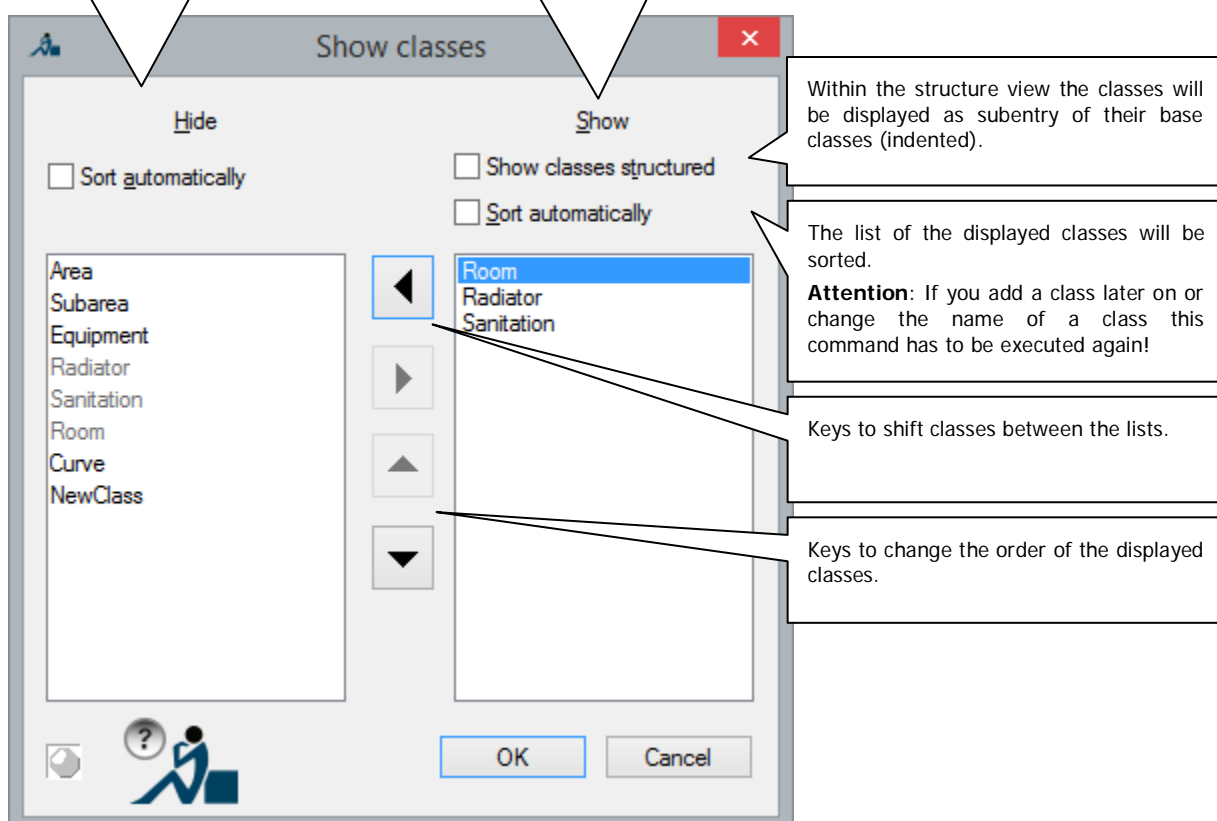
- Sort the classes
  - in the top data division of the structure view
  - in the lower definition part of the structure view
- Show and hide classes within the data division of the structure view.

The following window appears:

All existing classes will be displayed in the column "Hide". All that have already been transferred into the right column "Show" with the arrow key ▶, have been marked in grey.

You may change the **order within the structure view** with the ▲ and ▼ keys.

The classes that are to be displayed within the data division of the structure view will be displayed in the column "Show". By marking the class and confirming with the ◀ arrow key a class will be deleted from that list and therefore hidden. With the help of the ▲ and ▼ keys you may control the **order within the data division of the structure view**.

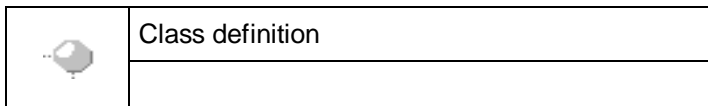


Within the structure view the classes will be displayed as subentry of their base classes (indented).

The list of the displayed classes will be sorted.  
**Attention:** If you add a class later on or change the name of a class this command has to be executed again!

Keys to shift classes between the lists.

Keys to change the order of the displayed classes.


**Configure... (class)**

A window for the determination of the properties of the respective class opens.

The setting options are the same as those of the command for creating a new class → **Add class**

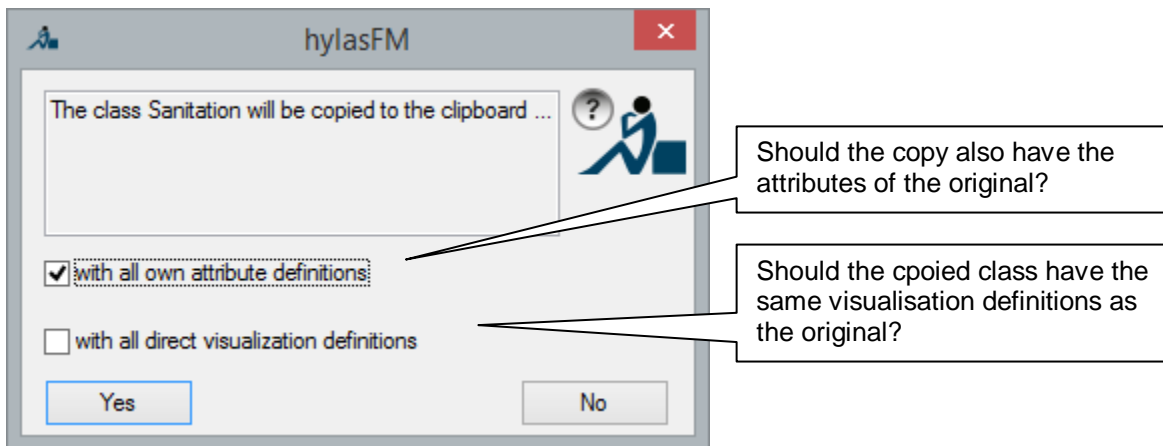
Additionally, you may decide for the tabs "Label", "Graphic" and "Type" whether existing labels, graphic or polygons are to be adapted to the changed settings.

## Copy class

	Class definition

This command copies the class definition with all of its settings, properties and (optionally) attributes into the clipboard.


The following dialogue box opens:



By using the command → **"Insert from clipboard"** under „ "Classes" it is possible to

- either insert the class into the same drawing (a copy is created).
- or copy the class into another drawing.

**Remove class**

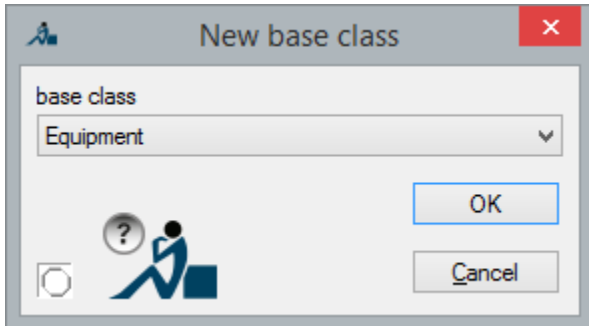
	Class definition

This command deletes a class permanently with all its objects. This step **cannot be reversed!**

## Add base class

○	Base classes

With this command you add a new base class to your class.



Potential base classes are offered within in the selection field.

The base class hands down all its attributes and link types to the derived class. If objects of the derived class already exist, the new attributes are added to those.

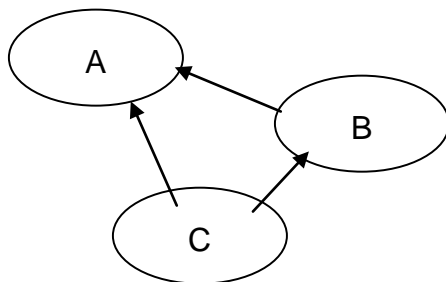
A link that has been defined for a base class can also be used for derived classes.

Example:

Windows and doors have a common base class "Opening". Thus, all windows and doors also have the attributes of the class "Opening". They can, however, have further attributes. Furthermore, windows and doors can be linked with rooms by a link "RoomHasOpening".

A class can have several base classes.

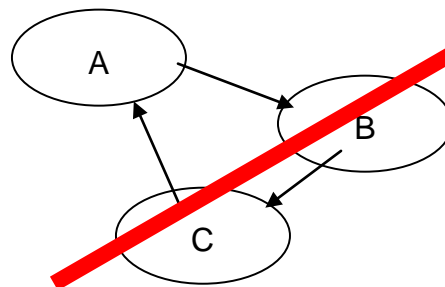
Cyclical inheritance is **not** allowed.



valid inheritance:

C has A and B as base classes


B has A as base class



invalid inheritance:

Classes form a cycle

**Remove base class**


	Based on "NAME"

This command serves to remove the base class "Name" of the current class.

**Caution:** The current class will thereby lose all its attributes which it has inherited from the base class. That means all values that have been collected already will be deleted irrevocably.

Links that become invalid due to this command will be removed as well (→**Add base class**).

## Add attribute

	Attribute definition
	only available for the structure entry "Attributes"

This command is used to add new attributes to a class.

Enter a name for the attribute in the input field "**Attribute name**". The name must be unique within the class – in other classes (also in base classes) attributes with the same name can exist.

Every attribute is of a certain type. There are the following **attribute types**:

Attribute type	Description	Example(s)
Integral number	integer	-∞, -1, 0, 1, 2, 42, +∞
Decimal number	floating-point number	-∞, -0.12, 3.141, +∞
Text	String of characters, any text	This is a text with numbers 1, 2 and special characters @\$!
Selective list	Value from a list; list <b>cannot</b> be extended during data capturing	Floor covering
Extendable selective list	Value from a list; list can be extended during data capturing	Room utilization
Selective tree	Value from a structured list; if the list has many entries, this hierarchical selection faster and provides a better overview	Inventory
File	Reference to external file; when changing an attribute, the file is opened with the application that has been assigned to it by the operating system	Photo, text file
Key	Text string; must be unique with in the drawing	Room number
Date	Date, displayable in any format	Recording date
Numeric calculation attribute	Calculation attribute; results from other attributes and linked objects	Wall surface area (Room.Perimeter x Room.Height – Opening.Area)



Text calculation attribute	Calculation attribute based on text; results from other attributes and linked objects	Text composed of several text attributes
----------------------------	---	--

The setting options for the individual attribute types are described in the following figures. Many settings are the same for all types. The right field (Decimal number) contains specific characteristics.

The image shows a screenshot of the 'Attribute Definition' dialog box in AutoCAD. The dialog is titled 'Attribute Definition' and contains several sections. The 'Attribute name' field is set to 'Height' and the 'Description' field contains 'Define room height'. The 'Attribute type' is set to 'decimal number'. The 'Display attribute on label' section is checked, with 'Prefix' set to 'HT:' and 'Suffix' set to 'm'. The 'Decimal number' section is highlighted in yellow and contains fields for 'Smallest allowed value' (0.000), 'Biggest allowed value' (+∞), 'Standard value' (0.000), 'Type of display' (<like AutoCAD), 'Display accuracy' (0.001), and 'Scale factor for display and export' (1). The 'Offer standard' radio button is selected. The 'Attribute definition code' field contains 'double,(0,+∞),1,1,eAutocadFormat,Last,("HT:","m")'. The 'OK' and 'Cancel' buttons are at the bottom.

**Attribute type** – has influence on the other setting options.

Name of the attribute

Any text for a closer description; is shown when filling in the attribute

Range of the attribute values; ∞ means infinite.

Default value, when a new object is created and the option "Offer standard" is activated.

display type

Decimal places in the label and for export → blocks

For every new object always the same value is offered.

Counts up by 1 with every new object (Room4 → Room5)

The value of the object created last will be offered.

Display within the label and for the conversion of AutoCAD block attributes (see → Export → Blocks)  
Example for the attribute value of 1.3:  
**HT:1.30m**

Attribute definition code (only compatible with hylasFM1.0)

Factor is used for Export → Blocks and the output within the label. (Example: plan in m; window width in cm → factor = 100)

The right part of the dialogue changes according to the set attribute type.

**integral number and text**

integral number

Smallest allowed value

Biggest allowed value

Standard value

Value range of the attribute; oo means infinite.

Default value, when a new object is created and the option "Offer standard" is activated.

text

Standard value

Key property

Default value, when a new object is created and the option "Offer standard" is enabled.

Every attribute of the current drawing has to be unique, i.e. the attribute may serve as "key" within a relational data base.

**Selective lists:**

selective list

Living  
Dining  
Sleeping  
Child

Show selection sorted

List of possible attribute values. The list may **not** be extended during the data recording. The first value is the default value, if a new object will be generated and the option "Offer standard" is active.

extendable selective list

Carpet  
Parquet  
Tiles

Show selection sorted

List of possible attribute values. The list may be extended during the data recording. The first value is the default value, if a new object will be generated and the option "Offer standard" is active.

### Selective tree

If a list has very many entries the hierarchical selection in an **selective tree** is much faster and clearer than the selection in an **selective list**. A common example for this is the selection of the room use according to an industrial standard or norm.

Let us take a look at a simple example: The type of a creature shall to be selectable as attribute value. The hierarchy could look like this within a spreadsheet program:

	A	B	C	D
1	plant			
2		tree		
3			oak	
4			fir	
5			linden	
6		flower		
7			cornflower	
8	animal			
9		mammal		
10			lion	
11			hare	
12		bird		
13			sparrow	
14			dove	
15				
16				

The output of this "table" as CSV file "creatures.csv" in ASCII format looks like this:

```

plant;;
;tree;
;;oak
;;fir
;;linden
;flower;
;;cornflower
animal;;
;mammal;
;;lion
;;hare
;bird;
;;sparrow
;;dove
    
```

This file "creatures.csv " may be imported (button "Import...") during the attribute definition of an selective tree.

enumtree

Separator:

Display depth:

Only last entries selectable

Show un-branched values compactly

Import... Edit... Test...

Separator between two hierarchical levels for the output within the label and during Export→Blocks

Number of outputs divided by separators, counted from the lowest hierarchical level

An entry to be selected has always to be from the lowest hierarchical level (e.g. fir).

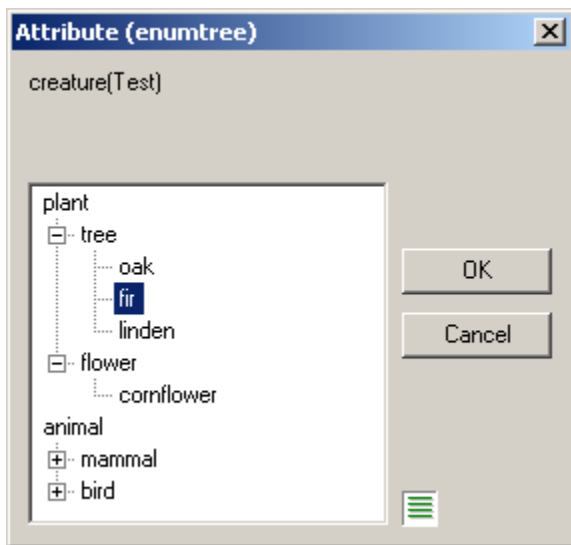
If a hierarchical level has only one entry, there won't be an indent for the output when the option is active. (In this example only relevant flower-cornflower.)

Import of an ASCII CSV file

Direct editing of the hierarchy

Testing to fill in the attributes with the set options

When filling in an attribute the following window will appear with the "fir" entry selected:



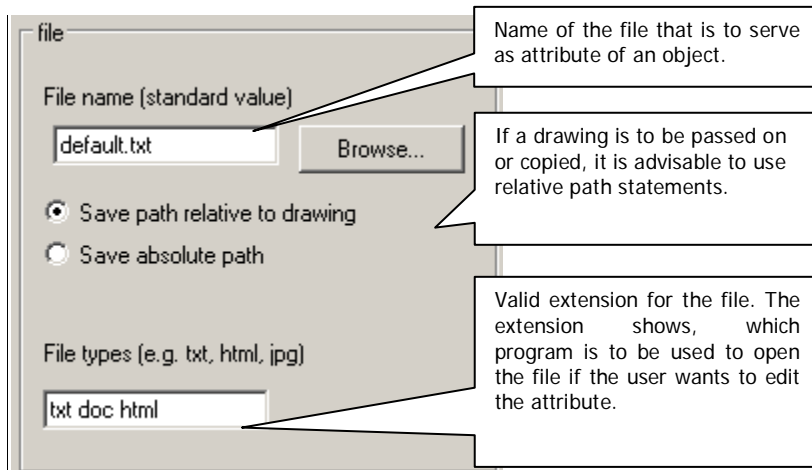
With the options set above ("Separator –" and "Display depth 2" ) "tree - fir" will be displayed within the label. The displayed attribute value would be "plant-tree-fir" if the entry for the display depth was 3. If the option "Only last entries selectable" would be deactivated, you could even select "tree" or "plant".

An interesting special case of a selctive tree is its **use as image of selection values to output values**. If you for examplpe are asked to select a country by its English name, but as attribute value only a token (country code) is to be shown, the following CSV file could be used:

```
Germany;D
Great Britain;GB
Italy;It
```

If you set "Display depth" to 1, "Only last entries selectable" to *off* and "Show unbranched entries compactly) to *on* the appearing attribute value will always be the country code only.

## File



## Key

A key attribute is an attribute with clear value assignment. Every value is allowed only once. That way the object can clearly be identified on the basis of this attribute within a data base and can be used as "Key".

Initial value for the key (for the first object of the class). A key has to be a valid XML key (starting with a letter, no spaces or special characters).  
Please use the attribute type "Text" with key characteristics, if these restrictions are too strong

## Date

Value range of the attribute. When clicking on the arrow, a calendar will open:

November 2010						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Selection of the date format. Next to the suggested values you may define your own format. In order to do so use the syntax "D" or day, "M" for month and "Y" for year.

## Numeric / Text calculation attribute

Attributes of the type numeric calculation attribute or text calculation attribute are calculation attributes and they are not filled in manually.

Calculation attributes are a powerful tool for the determination of *dependent terms*. You may for instance calculate the surface area of a "door" with numeric calculation attribute using the "width" and the "height". But you may also make complicated calculations like the calculation of the net cleaning areas. With text calculation attribute you can string the texts of different object attributes together. Thus identifiers may be composed of several attributes.

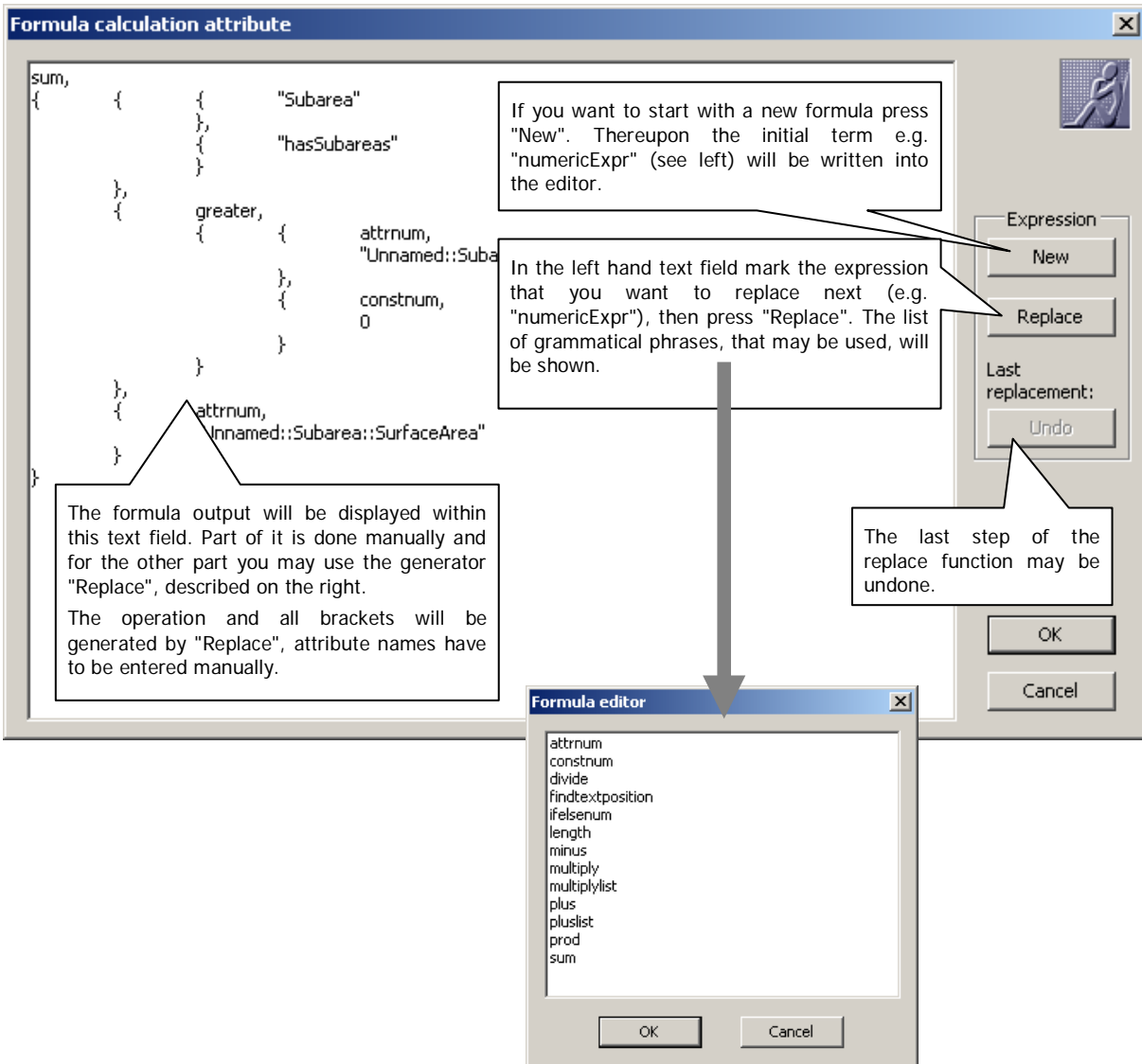
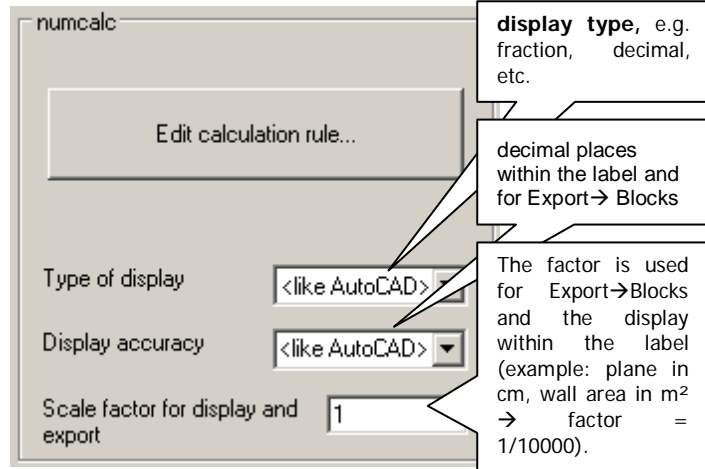
The definition of calculation attributes requires some practice. But the improvement of the data quality and the recording efficiency justify the effort. Following is the calculation rule for an attribute project::door::area as multiplication of height and width:

```
multiply, { {attrnum, "project::door::width"}, {attrnum, "project::door::height"} }
```

The individual logical terms always have the form operation,parameter. In the majority of cases the parameters are phrased as list {...}. The target attribute project::door::area does not have to be stated since this is the definition of exactly that attribute.

The figure on the right shows the specific part of the window for the definition of an attribute, when "Numeric calculation attribute" or "Text calculation attribute" have been selected as attribute types.

The large button **Edit calculation rule** opens a window for the editing of the formula.



The formulas consist of nested phrases in curly brackets. The separator between two phrases is the comma.

Tabs may be created by pressing the **Ctrl + tab key**. Indents and spaces are for an easier readability only and are not required by the software.

You may also always enter the calculation attribute manually. Therefore the use of a separate text editor from which the formula can be copied and pasted to the text field by **Ctrl + C** and **Ctrl + V** is suggested.

Here are the meanings of the primary phrases. A complete (syntactic) description of the definition of calculation attributes will be given in the paragraph of the grammar rules.

### Numeric calculation attribute

<b>attrnum</b>	Value of a numeric attribute. Numeric attributes are <b>integral number</b> , <b>decimal number</b> and also other <b>numeric calculation attributes</b> . Attributes must always be stated with their complete name (StructureName::ClassName::AttributeName). Example: <code>attrnum, "Unnamed::Subarea::Factor"</code>
<b>constnum</b>	If a number is to be used, that does not come from an attribute or another calculation, you will have to write: <code>constnum, 3.141</code>
<b>plus/ minus/ multiply/ divide</b>	Adding/subtraction/multiplication/division of two numerical values. Example: The following phrase will return a 6 <code>multiply, {{constnum, 3}, {constnum, 2}}</code> A surface area, reduces by 1.5% will be returned from <code>multiply, {{constnum, 0.015}, {attrnum, "Unnamed::Area::SurfaceArea"}}</code>
<b>sum/prod</b>	Sum/product accumulation of numeric attributes of linked objects. At first it is stated from which classes and which links the objects are to be taken. Then you may state a condition, which specifies, whether the value shall be used or not. If for example only values higher than 2.5 shall be considered, you will write: <code>{greater, {{attrnum, "Unnamed::Door::SurfaceArea"}, {constnum, 2.5}}}</code>  If there are no exceptions the entry will be: <code>{constbool, true}</code>  At last the values to be added/multiplied will be stated. The following example adds up the window areas of a room: <code>sum, {   {{ "Window" }, { "hasWindow" } },   {constbool, true},   {attrnum, "Unnamed::Window::SurfaceArea"} }</code>  The following example <i>counts</i> the windows of a room, that are larger than 1.5m <sup>2</sup> by adding up constant 1 <code>sum, {</code>

	<pre> {"Window"}, {"hasWindow"}}, {greater, {attrnum, "Unnamed::Window::SurfaceArea"}, {constnum, 1.5}}, {constnum, 1} } </pre>
<b>ifelsenum</b>	<p>Returns a numerical value depending on one or several conditions</p> <pre> ifelsenum, {   {     {greater, {{attrnum, "Unnamed::Room::Height"}, {constnum, 2}}},     {constnum, 1}   },   {     {less, {{attrnum, "Unnamed::Room::Height"}, {constnum, 0}}},     {constnum, 0}   },   {     {constbool, true},     {constnum, 0.5}   } } </pre> <p>The example returns the value 1 for a room height &gt;2, 0 for a negative height. In any other case (0 &lt;= room height &lt;= 2) the value will be 0.5.</p>
<b>length</b>	<p>Returns the number of characters of a text.</p> <p>Examples:</p> <pre> length, { consttext, "Hello" } </pre> <p>returns 5.0</p> <pre> length, { attrtext, "Unnamed::floor::designation" } </pre> <p>Returns the text length of the attribute designation</p>
<b>findtext position</b>	<p>Gives the position of the second text within the first text. If the second text is not found, length+1 will be returned.</p> <pre> findtextposition, {{consttext, "blue Monday"}, { consttext, "Mo"}} </pre> <p>returns 5.0</p> <pre> findtextposition, {{consttext, "blue Monday"}, { consttext, "bl"}} </pre> <p>returns 0.0</p>

**Remark:** For correct calculation results one has to assure that the units of all attributes are concurrent.

### Text calculation attribute

<b>attrtext</b>	<p>Value of a text based attribute of the type Text or <b>Text calculation attribute</b>. Attributes have to be stated with their complete name (StructureName::ClassName::AttributeName) Example:</p> <pre> attrtext, "Unnamed::floor::designation" </pre>
<b>consttext</b>	<p>If a text is to be used that is not from an attribute or another calculation you will have to write:</p> <pre> consttext, "Room" </pre>
<b>append</b>	<p>Ties texts together.</p> <p>Example: The following phrase returns "RoomCovering"</p> <pre> append, {{ consttext, "Room"}, { consttext, "Covering"}} </pre> <p>A room type that has been extended with a constant word "room" will be</p>



	<p>expressed like this:</p> <pre>append,{{ consttext,"room"},{ attrtext,"Unnamed::room::type"}}</pre>
<p><b>concatenate</b></p>	<p>1.) Listing of text attributes of linked objects                  2.) Copying of a text attribute into another object.</p> <p>Example for 2.). In the room the floor description is to be displayed as attribute:</p> <p>At first will be stated from which classes and links the objects are to be taken. Then a condition may be stated, which specifies if the value shall be looked at or not. In our case the attribute is to be copied only (without condition). The condition may be stated with {constbool,true}.</p> <p>11.4.2</p> <pre>concatenate,{     {         { "floor" }, {"hasRoom" },         { constbool, true },         { attrtext,"Unnamed::floor::designation" }     } }</pre>
<p><b>ifelsetext</b></p>	<p>Returns a text depending on one or several conditions.</p> <pre>ifelsetext, {     { {greater,{{attrnum,"Unnamed::room::height"},{constnum,2}}},       {consttext,"high"}     },     { {less,{{attrnum,"Unnamed::room::type" },{constnum,0}}},       {consttext,"wrong room height"}     },     { {constbool,true},       {consttext,"low"}     } }</pre> <p>The example returns the value “high” for a room height &gt;2, “wrong room height” for a negative height. In any other case (0 &lt;= room height &lt;= 2) the value will be “low”.</p>
<p><b>split</b></p>	<p>Fragments the first phrase into a list of strings using the second text as separator (string). Returns subscripted text from the third phrase of the list.</p> <p>For negative numbers the count will start from the end.</p> <pre>split, {     { consttext, "This&lt;I&gt;is&lt;I&gt;a&lt;I&gt;table" },     { consttext, "&lt;I&gt;" },     { constnum, -1 } }</pre> <p>Returns “table“ as for the negative algebraic sign the -1 was counted from the end.</p> <p>If the list is too short, the last text or for negative algebraic signs the first text of the list will be returned. In case of an empty list (separator not included in the text) "" is returned.</p>
<p><b>split enumtree</b></p>	<p>Fragments the Selective tree value into a list of text at first. Returns the subscripted text from the second phrase of the list.</p> <p>For negative numbers the count will start from the end. If the list is too short, the last (the leaf) or for negative algebraic signs the first text of the</p>

	<p>list (the root) will be returned.</p> <p>Example:</p> <p>In a hierarchic attribute "type" of the class "plant" (selective tree) the value Tree-conifer-fir-blue spruce has been selected.</p> <pre>splitenumtree, {   { attrtext, "Unnamed::plant::type" },   { constnum, 2 } }</pre> <p>Returns the text "conifer" since for the positive algebraic sign of the number 2 the count started from the beginning.</p>
sub	<p>Returns the text part from the start position of the second phrase and with the length of the third phrase.</p> <p>The position of the first sign is 0.</p> <pre>sub, {{consttext, "one yellow leaf"}, {constnum, 4}, {constnum, 6}}</pre> <p>Returns "yellow".</p>
subst	<p>Substitutes the appearance of the second text with the third text within the first text.</p> <pre>subst, {{consttext, "coloured shirt"},         {consttext, "coloured"}, {consttext, "colored"}}</pre> <p>Returns "colored shirt".</p>

## Grammar for the calculation attributes

Calculation attributes are universal for they can be nested to any depth. A grammar is used for a complete description of the infinite diversity. The grammar describes a calculation term that can be interpreted by the software. It consists of grammar symbols, which may be substituted by several phrases. The vertical line separates alternative phrases from each other.

There will always be a start symbol

“numeric\_Expr“ for the numeric calculation attribute and

“text\_Expr“ for the text calculation attribute.

Starting with the start symbols all symbols have to be substituted little by little. **Elements that can not be broken down any further** are printed **bold** in the expression.

The \* means, that the phrase within the parentheses may be repeated at will.

In the software all symbols appear without underscore (e.g. textExpr).

Symbol	Expression
numeric_expr	sum_expr   product_expr   plus_expr   minus_expr   multiply_expr   division_expr   numeric_const_expr   numeric_attribute_expr   numeric_ifelse_expr   length_expr   findtextposition_expr
text_expr	append_expr   concatenate_expr   text_const_expr   text_attribute_expr   text_ifelse_expr   split_expr   splitenumtree_expr   sub_expr   subst_expr
boolean_expr	and_expr   or_expr   not_expr   less_expr   greater_expr   numeric_equals_expr   text_equals_expr   boolean_const_expr
sum_expr	<b>sum</b> , { { class_list , relation_list } , { boolean_expr } , { numeric_expr } }
product_expr	<b>prod</b> , { { class_list , relation_list } , { boolean_expr } , { numeric_expr } }
plus_expr	<b>plus</b> , { { numeric_expr } , { numeric_expr } }
pluslist_expr	<b>pluslist</b> , { ( { numeric_expr } , ) * }
minus_expr	<b>minus</b> , { { numeric_expr } , { numeric_expr } }
multiply_expr	<b>multiply</b> , { numeric_expr } , { numeric_expr }
multiplylist_expr	<b>multiplylist</b> , { ( { numeric_expr } , ) * }
division_expr	<b>divide</b> , {

	{ numeric_expr } , { numeric_expr } }
numeric_ifelse_expr	<b>ifelsenum</b> , { ( { if_pair_num }, )* { if_pair_num } }
if_pair_num	{ boolean_expr } , { numeric_expr }
length_expr	<b>length</b> , { text_expr }
findtextposition_expr	<b>findtextposition</b> , { { text_expr } , { text_expr } }
numeric_const_expr	<b>constnum</b> , number
numeric_attribute_expr	<b>attrnum</b> , global_attribute_def_name
append_expr	<b>append</b> , { ( { text_expr } , )* }
concatenate_expr	<b>concatenate</b> , { { class_list , relation_list } , { boolean_expr } , { text_expr } }
split_expr	<b>split</b> , { { text_expr } , { text_expr } , { numeric_expr } }
splitenumtree_expr	<b>splitenumtree</b> , { { text_expr } , { numeric_expr } }
sub_expr	<b>sub</b> , { { text_expr } , { numeric_expr } , { numeric_expr } }
subst_expr	<b>subst</b> , { { text_expr } , { text_expr } , { text_expr } }
text_const_expr	<b>consttext</b> , string
text_attribute_expr	<b>attrtext</b> , global_attribute_def_name
and_expr	<b>and</b> , { { boolean_expr } , { boolean_expr } }
or_expr	<b>or</b> , { { boolean_expr } , { boolean_expr } }
not_expr	<b>not</b> , { boolean_expr }
less_expr	<b>less</b> , { { numeric_expr } , { numeric_expr } }
greater_expr	<b>greater</b> , { { numeric_expr } , { numeric_expr } }
numeric_equals_expr	<b>equals</b> , { { numeric_expr } , { numeric_expr } }
text_equals_expr	<b>equalstext</b> , { { text_expr } , { text_expr } }
boolean_const_expr	<b>constbool</b> , (true   false)
class_list	{ ( class_name , )* class_name }
relation_list	{ ( relation_name , )* relation_name }
relation_name	"ID of a link"
class_name	"ID of a class"
global_attribute_def_name	"global ID of an attribute"

string	"string"
number	number

### Examples for numeric calculation attributes

An area list definition "Unnamed" with the classes "Area", "Subarea", "Window", "Door" and "Opening" is the base. "Door" and "Window" inherit from "Opening" and a link "hasOpening" exists. The necessary attributes may be taken from the examples.

#### Subtraction areas

Add up all linked subareas (within the link "hasSubarea" with negative factor)

```
sum, {
  {{"Subarea"}, {"hasSubarea"}},
  {less, {
    {attrnum, "Unnamed::Subarea::Factor"},
    {constnum, 0}
  }},
  {multiply, {
    {multiply, {
      {constnum, -1},
      {attrnum, "Unnamed::Subarea::Factor"}
    }},
    {attrnum, "Unnamed::Subarea::SurfaceArea"}
  }}
}
```

#### Shell subtraction

Subtract 1.5 percent from the total surface area.

```
minus, {
  {attrnum, "Unnamed::Subarea::SurfaceArea"},
  {multiply, {
    {constnum, 0.015},
    {attrnum, "Unnamed::Subarea::SurfaceArea"}
  }}
}
```

#### Wall surface

Calculate the total wall surface of a room.

```
multiply,
{
  {
    attrnum,
    "Rooms::Area::Perimeter"
  },
  {
    attrnum,
    "Rooms::Area::RoomHeight"
  }
}
```

### Examples for text calculation attributes

An area list definition "Unnamed" with the classes "Area", "Subarea" and "Room" is the base. "Room" inherits from "Area". The necessary attributes may be taken from the examples.

#### LookUp Table

Within a database "cryptic" keys are necessary, but for a better overview it is intended to use real names for the recording of the values. You may achieve that with a text calculation attribute that has been defined as LookUp table. Example: The recording of the floor covering type is to be carried out by the user with terms like laminate, carpet, etc. The database needs keys like "S273", "S541", etc. though.

<b>Floor covering</b>	<b>key</b>
laminate	"S273"
PVC	"S783"
carpet	"S541"
other	"XY"

```

ifelsestext,
{
  {
    {
      equalstext,
      {
        {
          attrtext,
          "Unnamed::Room::FloorCovering"
        },
        {
          consttext,
          "lamine"
        }
      }
    },
    {
      consttext,
      "S273"
    }
  },
  {
    {
      equalstext,
      {
        {
          attrtext,
          "Unnamed::Room::FloorCovering"
        },
        {
          consttext,
          "PVC"
        }
      }
    },
    {
      consttext,
      "S783"
    }
  },
  {
    {
      equalstext,
      {
        {
          attrtext,
          "Unnamed::Room::FloorCovering"
        },
        {
          consttext,
          "carpet"
        }
      }
    },
    {
      consttext,
      "S541"
    }
  }
},
{
  {
    constbool,
    true
  },
  {
    consttext,
    "XY"
  }
}
}

```

The condition "other" is expressed with {constbool,true}.

### Linking of keys

A room has desks, which are only assigned to this room. The desks have an inventory number which consists of the room number and the desk number. They will be divided by a hyphen.

```

append,
{
  {
    concatenate,
    {
      {
        "Room"
      },
      {
        "hasDesk"
      }
    },
    {
      constbool,
      true
    }
  }
}

```

```

        },
        {
            attrtext,
            "Unnamed::Room::Description"
        }
    },
    {
        consttext,
        "-"
    }
    {
        attrtext,
        "Unnamed::Desk::DeskNumber"
    }
}

```

The operation `concatenate` is necessary in order to be able to assemble attributes across links, whereas `append` assembles attributes within a class. Therefore you have to get the room description via `concatenate` into the class desk and then, within the desk class, assemble the desk number completely with `append`.

### Updating the calculation attributes


Since the calculation of calculation attributes may be very time-consuming, the updating has always to be started manually. This is possible by:

- double clicking the attribute,
- opening the context menu of the attribute and selecting **→Update**
- using the command **→Check/Create consistency...** within the head of the structure view (big blue square). The option **Update calculation attributes** is enabled and the window is closed by pressing **OK**.

The method mentioned last makes sure that **all** attributes of the drawing will get an up to date value and it should be performed before exporting data and/or passing on the drawing.



## Insert from Clipboard

	Attribute definition
	only available for structure base " <b>Attributes</b> "

This command inserts an attribute definition that was previously copied onto the clipboard.

In connection with the command "Copy attribute definition" it is possible to:

- In connection with the command "Copy attribute definition" it is possible to:
- copy attributes from one structure definition into another one.

## Sort attributes, show/hide...

☰	Attribute definition
	only available for structure entry " <b>Attributes</b> "

With this command you may:

- Sort the attributes
  - in the top data division of the structure view
  - in the lower definition part of the structure view
- Show and hide attributes within the data division of the structure view.

The following window will open:

The order also has an influence on the display of the attributes within the labels.

All existing attributes will be displayed in the column "Hide". All that have already been transferred into the right column "Show" with the arrow key ▶, have been marked in grey.


You may change the **order within the structure view** with the ▲ and ▼ keys.

The attributes that are to be displayed within the data division of the structure view will be displayed in the column "Show". By marking the class and confirming with the ◀ arrow key a class will be deleted from that list and therefore hidden. With the help of the ▲ and ▼ keys you may control the **order within the data division of the structure view**.

List of the displayed attributes will be sorted in alphabetical order.

Liste der angezeigten Attribute wird alphabetisch sortiert.

## Change attribute definition


	Attribute definition
	only available for structure entry "Attributes"

You can change the definition of an attribute any time. See → **Add attribute** for further explanations on the setting options.

As long as no objects exist, that contain the attribute to be changed, a change of the definition is unproblematic. If objects already exist, the software will try to convert the values as good as possible. E.g. a decimal number attribute 1.9 becomes an integral number attribute 2. When converting a text attribute "12Abd 15" a key attribute \_12Abd\_15 will be created and so on. Should the conversion fail, the default value will be put in.

The conversion into a text attribute is always possible without losses.

## Copy attribute definition

	Attribute definition
	only available for structure base "Attributes"


This command helps to quickly define new attributes, in which you copy attribute definitions.

When you call the command, the attribute definition is copied to the clipboard.

With the command → **Insert copy** under „ Attributes“ one can

- either insert the attribute into the same drawing (a copy is created).
- or copy the attribute into another drawing.


**Delete attribute definition**

	Attribute definition
	available for concrete attribute definitions

This command deletes an attribute from the structure definition.

**Caution!** This command cannot be reversed.

## Define visualization

	Visualization definition
	on attributes of a class with label or an area or curve class

This command is used for the definition of an attribute visualization or the configuration of the corresponding legend.

With the help of the visualization you can make sure, that the objects of a class have a different appearance, depending on the value of a certain attribute.



There are five options:

- **Hatching:** Area objects will be hatched differently depending on the attribute value. You may choose the same selection as hatching pattern as in the AutoCAD hatching command. For area hatchings select the pattern "SOLID". Starting with AutoCAD 2011 you may also select transparent hatchings.
- **Material:** Area objects will get an area filling with the characteristics of an AutoCAD material. This way you can get transparency effects, e.g. map areas within a photo, so the image will still be visible. But the filling with "fancy" patterns is also possible (e.g. wood or stone grains).
- **Block:** Depending on the attribute value, different block symbols will be inserted at the objects.
- **Curve:** Depending on the attribute value, line-like objects will get different line characteristics [line width, style, color].
- **Label:** Depending on the attribute value, the object labels can have a different color, size or font design.

The definition of a visualization is always valid **for a certain attribute**. It consists of a list of all ranges, which are relevant for the visualization and the information how these ranges are to be displayed.

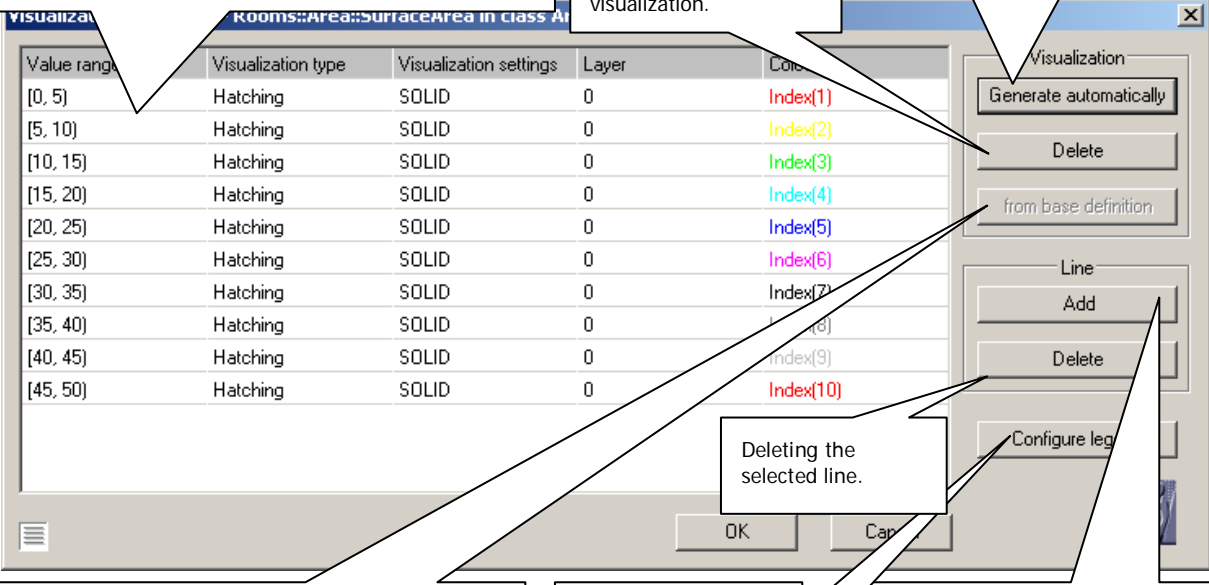
For the five visualization options different requirements have to be fulfilled. The requirements regard the class in which an attribute has been defined. This requirement will always be referred to as "*basic requirement*". If one requirement is fulfilled the according visualization type can be used:

Visualization type	<i>Basic requirement</i>
Block, Label	The class of the attribute has a label.
Hatching, Material	The class of the attribute has been derived from the class "Area" (the class is of the "Area" type) or it is a direct attribute of the class "Area".
Curve	The class of the attribute has been derived from the class "Curve" or "Area" or is a direct attribute of the class "Curve" or "Area".

Visualization definitions may be defined on attributes  defined directly within the class or on inherited attributes .

If visualizations have already been defined on an attribute of a class within the base class these may be copied or overwritten in the derived class.

The definition of the visualization is created via the following window.



Value ranges of the attribute and their visualization settings. At the beginning there are no entries. By clicking into an entry additional control elements appear within the input field, which make the editing of the individual entries possible.

Opens the wizard for the automatic generation of value ranges and corresponding visualization settings.

Deleting the entire visualization.

Deleting the selected line.

If visualizations have already been defined on an attribute of a class within the base class, these may be overwritten in the derived class. With this button you can get a copy of the visualization from the base class.

Leads to the configuration window for the legend

Adds a new line, in order to generate a new entry for a value range and the corresponding visualization settings.  
Deletes a selected line in the table.

Value range	Visualization type	Visualization settings	Layer	Colour
[0, 5]	Hatching	SOLID	0	Index(1)
[5, 10]	Hatching	SOLID	0	Index(2)
[10, 15]	Hatching	SOLID	0	Index(3)
[15, 20]	Hatching	SOLID	0	Index(4)
[20, 25]	Hatching	SOLID	0	Index(5)
[25, 30]	Hatching	SOLID	0	Index(6)
[30, 35]	Hatching	SOLID	0	Index(7)
[35, 40]	Hatching	SOLID	0	Index(8)
[40, 45]	Hatching	SOLID	0	Index(9)
[45, 50]	Hatching	SOLID	0	Index(10)

There are two options to generate a definition.

You either use the button **Generate automatically** and generate a visualization for all value ranges automatically. A wizard leads you step by step through the needed information. Further information regarding the wizard will be given later in this reference.

Or you generate the definition line per line manually with the button **Add**.

### Adding lines manually

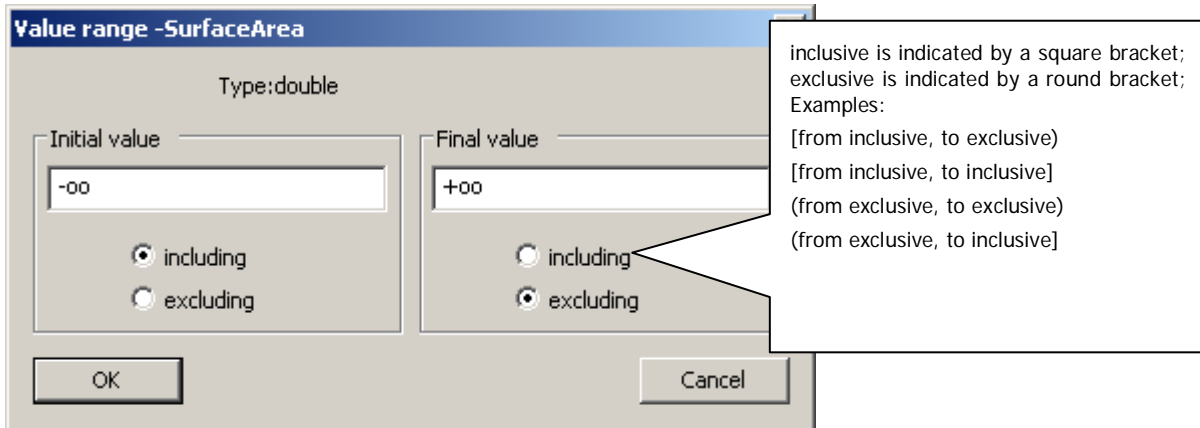
Have you placed the cursor over a line already, this line will be copied and pasted. A default entry is generated as follows.

Visualization definition - Unnamed::Area::SurfaceArea in class Area				
Value range	Visualization type	Visualization settings	Layer	Colour
[from,to ]	Hatching	SOLID	0	ByLayer

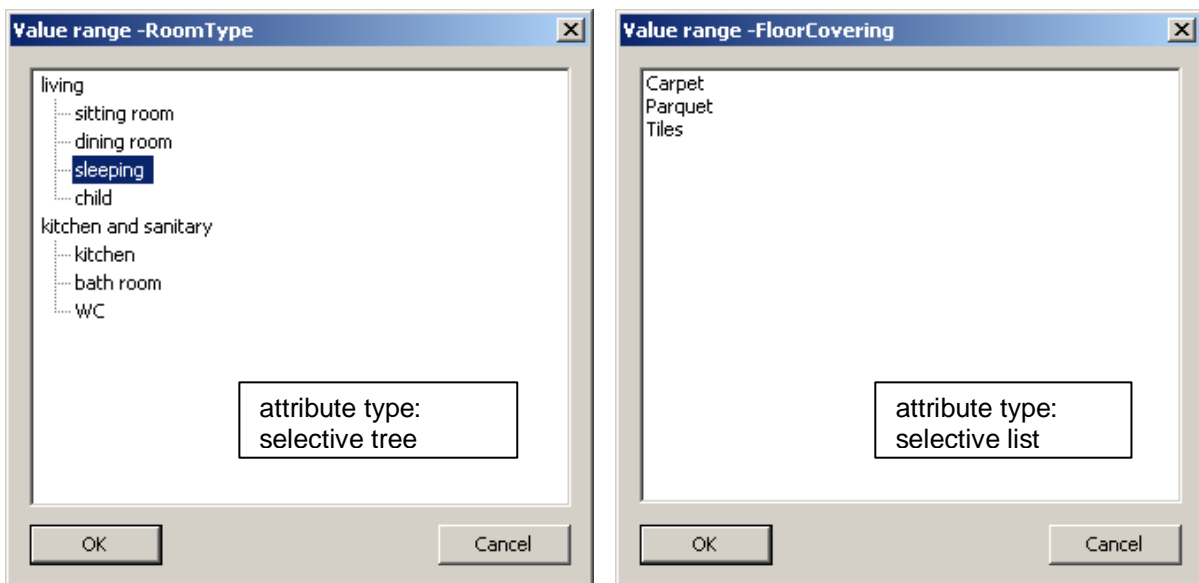
## Column Value range

Click on the **[from, to)** entry. If you click on the appearing button, a window for the determination of the value range opens. This window depends on the attribute type:

- The attribute types "Decimal number", "Integral number", "Numeric calculation attribute", "Text calculation attribute", "Key", "Text" and "Selective list" may be added.



Attribute types "Selective tree" and "Selective list".



If you, for example, want to define a value range for a "Text" attribute that contains all texts starting with an A, the value range would be defined with [A,B). That is: starting value: A (inclusive) to end value: B (exclusive).

## Column Visualization type

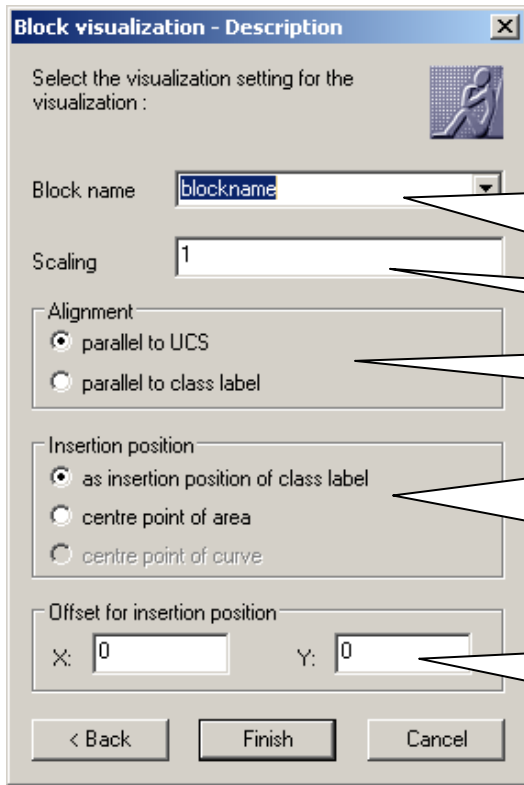
Click on the entry within the column "Visualization type", in this case "Hatching". A select list appears. According to the given *basic requirements* you may now select block, hatching, material, curve or label.



### Column Visualization settings

By clicking on the entry of this column a window for the definition of properties opens. It looks different depending on the visualization type

- For hatching visualizations this will be the pattern of the hatching, e.g. "SOLID".
- For block visualizations the block name will be stated here.
- For curve visualizations the line style will be stated here.
- For material visualizations the material name will be stated here.



Selection of the block name.  
Only blocks that are defined for this AutoCAD drawing will be shown here.

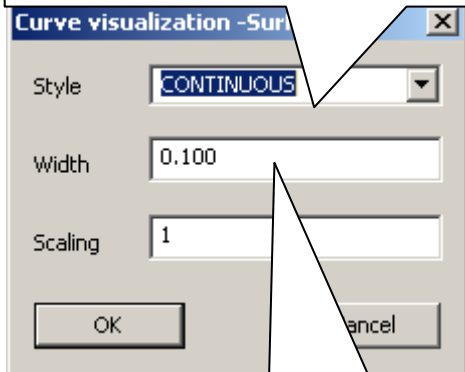
Scaling factor for the inserted block.

The block will either be aligned parallel to the UCS or parallel to the class label.

According to the base requirements the options for the inserting position will be available.  
(Refer to the table above for the definition of the basic requirements.)

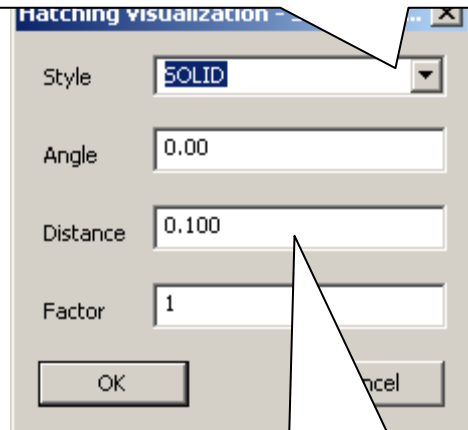
Move inserting position of the block by these values in relation to the label position.

Only line styles that have been included in AutoCAD can be selected.



"Width" is the value for the line width.  
"Scaling" gives the size of the pattern when the line is a dashed line.

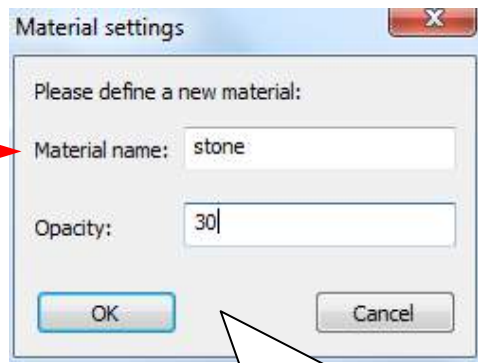
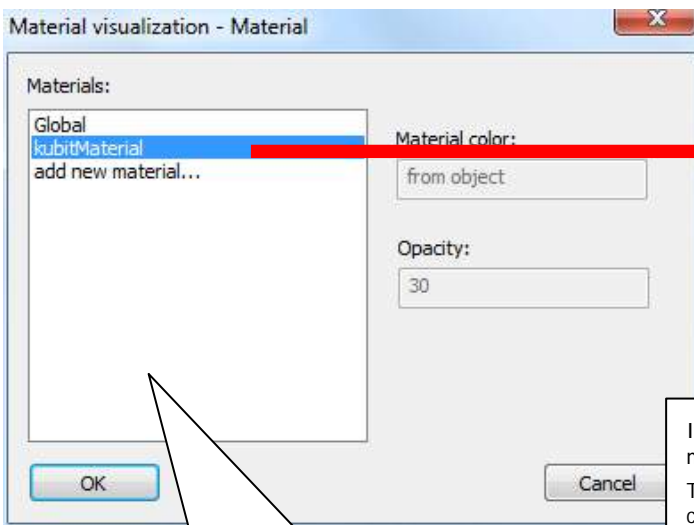
Selection of the hatching pattern. Only styles loaded into AutoCAD are available.



Angle, distance and factor control the appearance of the hatching. They should be selected according to the dimensions of the units used in the drawing.



If "from class" will be selected, the label size and text style of the visualized labels will not be changed.  
 If you select "user defined" you may change the settings for label size and text style.



Selection of a predefined material with the characteristics stated to the right.  
 All materials, defined within this drawing will be displayed.  
**Global:** AutoCAD will automatically generate this material with opacity of 100%.  
**kubitMaterial:** hylasFM will automatically generate this material with opacity of 30%.  
**Add:** Here you may define a new material

If you generate a new material, you may stat the material name and the opacity.  
 The opacity is a value between 0 and 100. "0" means completely transparent and "100" is completely opaque.  
 The material color can not be set here. Usually the color will be assigned within the main dialog of the visualization definition. For special color and filling wishes you may also use the AutoCAD material editor (AutoCAD command "MATERIALS")

### Column Layer / Column Color

By clicking into these columns you can select the layer or the color for the value range. According list fields will appear where you can select the layer name and the color value.

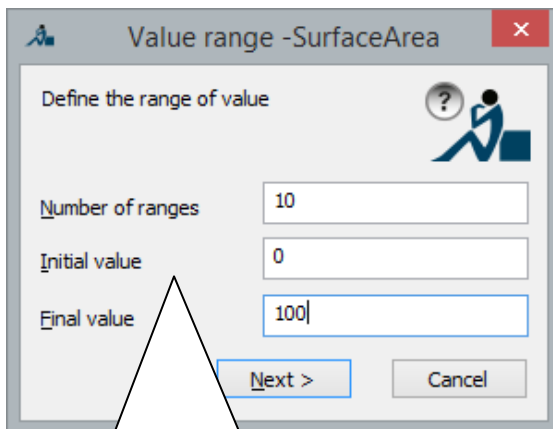
When visualizing materials, there might be a colored entry "of material" within the column "Color". This entry can not be edited any more and will appear, when a color value has been assigned to the material with AutoCAD commands. If you want to change this do the following: With the AutoCAD command "material" you will have to assign the value "ofObject" to the material. If you generate new materials with feature data management only, the value "ofObject" will be assigned by default.

## The wizard “Generate automatically”

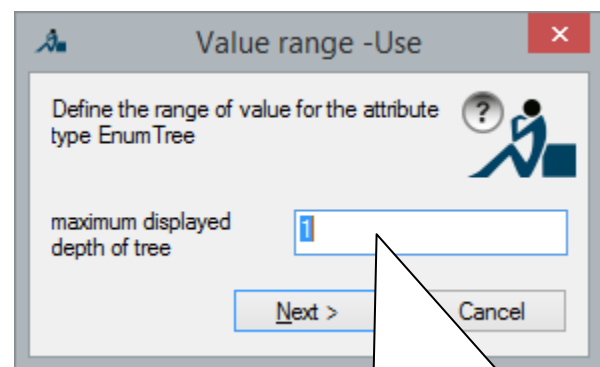
With the help of the visualization wizard a complete visualization is to be defined in a few steps. Two to three steps are necessary:

- 1) Definition of the value range segmentation
- 2) Determination of the visualization type and the general characteristics like layer and color
- 3) Determination of the specific characteristics, depending on the visualization type

**Step1:** Depending on the attribute type, the following window will open:



Generally it is tried to divide the value range of an attribute type evenly. To do so, an inquiry is made for the number and to which area this applies. This is also applicable for the attribute types “integral number”, “numerical calculation”, “string”, “text calculation” and key.



For the attribute type “selective tree” the structure view depth, down to which shall be visualized, is stated. That means all values on a deeper level get the same visualization.

This step does not apply for the attribute types *Selective list* and *Extendable selective list*. In that case, one value range will be generated per value.

**Step2** requires the statement of the visualization type and its general characteristics.

You will have to specify by what the individual value ranges are to be differentiated. That can be the color or the layer.

If you set this option to “vary” a color or a layer will be assigned to every value range automatically.

If you set a constant color and a constant layer here, the visualization will look the same for all value ranges. This might be reasonable as basis of a manual customization of the individual value ranges. In that case a warning appears that reminds you of the fact.

The corresponding window looks like this

**Visualization type -FloorCoveri...**

Please select a visualization type, layer and colour for the attribute visualization

**Visualization type**

- Block
- Curve
- Label
- Hatching
- Material

**Layer**

- Vary layer
- Select layer: 0

**Colour**

- Vary colour
- Select colour: ByBlock

< Back   Next >   Cancel

Here you select the according visualization type. You are asked for the necessary settings in the next step.

Either state a layer on which all visualizations are to be generated or let hylasFM vary the layer. The different layer names are made up of the attribute name and the value range.  
Example for layer names:  
"SurfaceArea [0,20]"  
or  
"FloorCovering carpet"

Either state the color for the visualization or let hylasFM vary the color. Then every value range will get a new color for the visualization.

**Step3** makes an inquiry for the visualization settings. Again, these differ according to their visualization type

**Block visualization -FloorCoveri...**

Select the visualization setting for the visualization type block :

**Block name**: WD\_CHT\_LHT

**Scaling**: 1

**Alignment**

- parallel to UCS
- parallel to class label

**Insertion position**

- as insertion position of class label
- centre point of area
- centre point of curve

**Offset for insertion position**

X: 0.000   Y: 0.000

< Back   Finish   Cancel

Selection of the block name.  
Only blocks, which have been defined within this AutoCAD drawing are available.

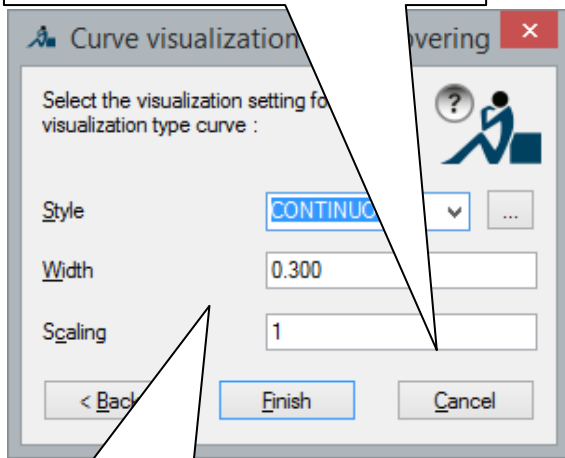
Scaling factor for the inserted blocks.

Align block parallel to UCS or parallel to the class label.

According to the basic requirements the options for the inserting position are available.  
(Refer to the table above for the definition of the basic requirements.)

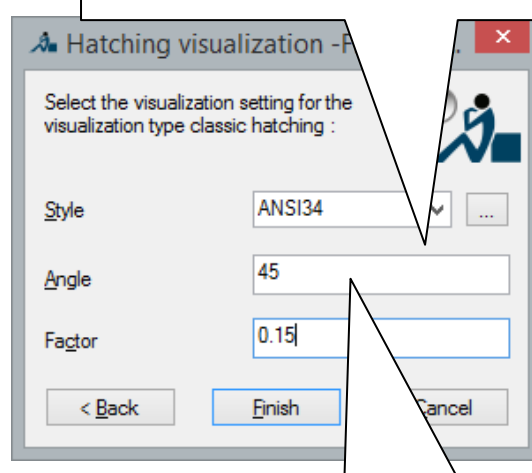
Move inserting position of the block by these values in relation to the label position.

Only line styles that have been loaded into AutoCAD can be selected.

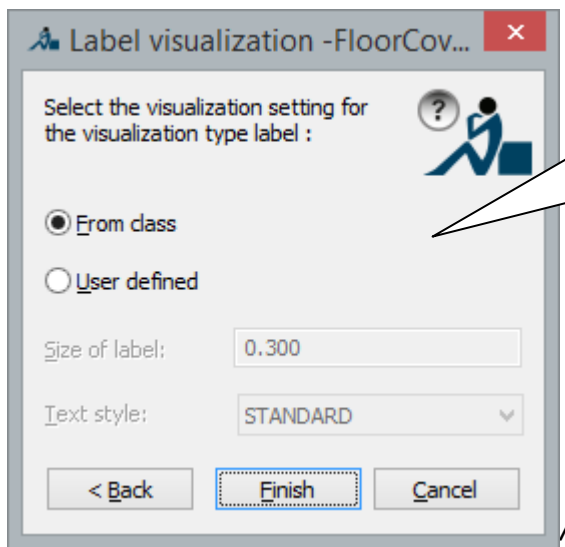


"Width" is the value for the line width.  
 "Scaling" gives the size of the pattern when the line is a dashed line.

Selection of the hatching pattern.  
 Only styles loaded into AutoCAD are available.



Angle, distance and factor control the appearance for the hatching. They should be selected according to the dimensions of the units used in the drawing.



If "of class" will be selected, the label size and text style of the visualized labels will not be changed.  
 If you select "user defined" you may change the settings for label size and text style.

**Material visualization -FloorCovering**

Select the visualization setting for the visualization type transparent filling :

Materials:

- Global
- kubitMaterial
- add new material...

Material color:

Opacity:

**Material settings**

Please define a new material:

Material name: stone

Opacity: 30

OK Cancel

If you generate a new material, you may stat the material name and the opacity.  
The opacity is a value between 0 and 100. "0" means completely transparent and "100" is completely opaque.  
The material color can not be set here. Usually the color will be assigned within the main dialog of the visualization definition. For special color and filling wishes you may also use the AutoCAD material editor (AutoCAD command "MATERIALS")

Selection of a predefined material with the characteristics stated to the right.  
All materials, defined within this drawing will be displayed.  
**Global:** AutoCAD will automatically generate this material with opacity of 100%.  
**kubitMaterial:** hylasFM will automatically generate this material with opacity of 30%.  
**Add:** Here you may define a new material with the desired opacity characteristic.

The visualization definition is generated by pressing the button **Finish**.

### Configure legend

In the main dialog "Visualization definition" you will find an according button which opens the following window:

The screenshot shows the 'Configure legend' dialog box. It features a table with three columns: 'Value range', 'Identifier', and 'Total'. Below the table are settings for 'Calculation of totals' (ON/OFF), 'Header', 'Text size', 'Layer', and a checkbox for 'Hide unused ranges of value'. There are also input fields for 'Prefix' and 'Suffix', and 'OK'/'Cancel' buttons.

Value range	Identifier	Total
Carpet	Carpet	(x)
Parquet	Parquet	(x)
Tiles	Tiles	(x)

Callouts provide the following information:

- Value ranges of the attribute defined within the visualization.
- Identification which appears in the legend.
- Area visualization: all surface areas will be added up
- Curve visualization: all curve lengths will be added up
- Block visualization: number of blocks
- (x) equals ON
- ( ) equals OFF
- Headline of the legend
- Font size of the legend
- Layer on which the legend is to be placed.
- Automatic hiding of value ranges, which are not being used. That means: If the totals formation of a value range is zero, this value range will not be displayed in the legend.
- Prefix/suffix which will appear before/after the sum in the legend.

A legend may be inserted by the command →**Visualization....** That can be found in the top part of the structure tree within the context menu of the corresponding attribute.

**Remark: Changing the visualization does not update already inserted legends!**

## Add link

→	Link definition
	only available for the structure entry "Links"

With this command you can add new links between classes to your data structure.

In the field "**Name of link**" enter a name for the link. Please keep in mind to use a unique and significant name (e.g. "RoomHasRadiator" for a link of the class "Room" with the class "Radiator"). Select the classes that are to be linked with each other in the frame "**classes to be linked**". You can also link a class with itself to express connections between objects of a class (e.g. RoomAdjoinsRoom).

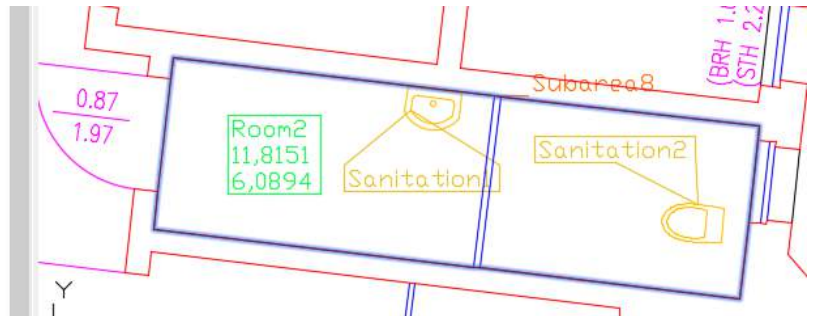
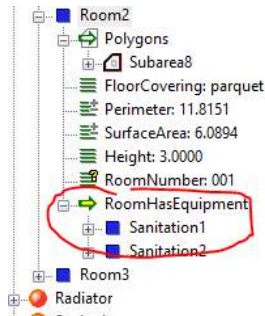
Only the fields "Name of link" and "classes to be linked" have to be filled in. The three facultative options in the frame "Properties of the link" are described in the following three paragraphs.

### Option "Link CLASS2 on the basis of position with CLASS1" → yellow arrow

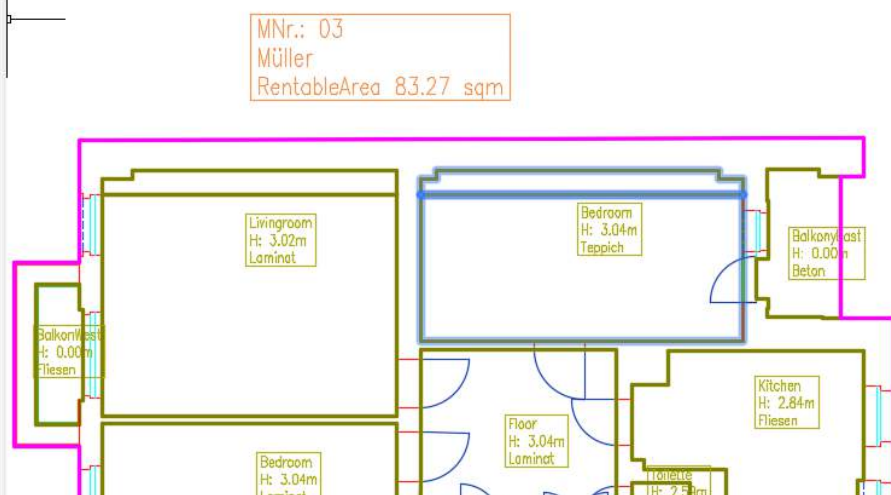
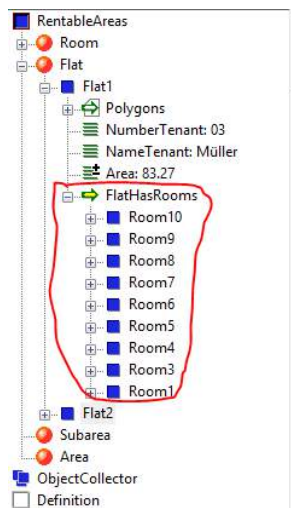
- Link of a class "Area" and another class with label (e.g. equipment). They will automatically be linked with each other if the insertion point of the label is within



the area. Which of the two linked classes will inherit from area is irrelevant.



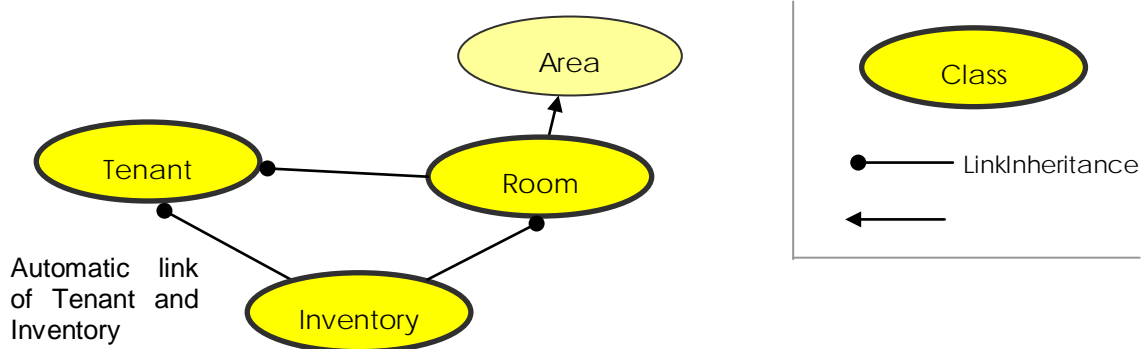
- Link between two "Area" classes: objects will be linked to each other if
  - The areas of the objects overlap each other (lease area and rooms) → setting "At least tangent to".
  - An area lies completely within another area (e.g. level and rooms) → "Completely inside".




- Link between an "Area" class and a "Curve" class: Objects will be linked to each other if the bordering or displaying lines of the objects intersect (e.g. rooms and cables).

For any class you may also generate an automatic link when using other links (whereas the linked class has to be derived from "area"). The usable links may be selected via the button "Edit...".

With this, an automatic link of objects without area characteristics may be executed when having the following structure:



Links with the option “Equipment” are updated by running the **→Check/Create consistency...** from the context menu of the structure view.

**Option “Consider CLASS2 when calculation the area for CLASS1” →white arrow** 

This option is only available if “Class1” inherits from “Area” and “Class2” from “Subarea”. If the option is enabled, the surface areas of objects of “Class1” are calculated using the surface areas of objects of “Class2”.

### **Option ‘Monitor number of objects to be linked (Cardinality)’**

The option “Monitor number of objects to be linked (Cardinality)” helps to ensure the quality of data capturing by formulating further restrictions. The restrictions are expressed in quantities here.

The way the setting options should be read will be explained with help of the example “hasFacility” (see figure above).

*Number of “Room” objects that must be linked with a “Facility” object:*

requires the indication of a quantity interval for the rooms. If facilities can exist without rooms, the interval can start with {0, ... }. Otherwise, it starts with {1, ... }. Thus, the lower limit is defined. The upper limit is determined by the question whether facilities can belong to more than one room. If so, the upper limit needs to be set to { ... ,oo}; should the maximum be one room { ... ,1}.

*Number of “Facility” objects that must be linked with a “Room” object:*

requires the indication of a quantity interval for the facilities. If rooms can exist without facilities, the interval can start with {0, ... }. Otherwise, it starts with {1, ... }. Thus, the lower limit is defined. The upper limit is determined by the question whether a room can have several facilities. If so, the upper limit needs to be set to { ... ,oo}; should the maximum be one facility {...,1}.

*An example “TenantUsesRoom”:*

The default setting allows that every object is linked with any number of other objects. The standard behavior can be useful for links such as “TenantUsesRoom”. Every tenant can use several rooms and every room can generally be assigned to several tenants. However, for most links this standard behavior is undesirable.

*An example “InventoryHasType”:*

With the link “InventoryHasType” every “Inventory” object should only have exactly one “InventoryType”. This is expressed by the interval {1,1}. To an inventory type can be assigned any number of or no inventory at all. In terms of intervals this is expressed by {0,+oo}. +oo stands for *infinite*. This setting ensures that you do not forget to assign a type to every piece of inventory when capturing data.

The cardinality is checked by using in the context menu command of the structure **→Check/Create consistency...**

Another example could be doors that are assigned to rooms. Every room is linked with any number of but at least one door: {1,oo}. However, every door is to be linked with exactly two rooms. Thus, you write {2,2}. But when running **→Check/Create**

**consistency...**, the entrance of the building will be found fault as it is , of course, only linked with one room. The definition of a condition {2,2} is still useful since most doors can be checked this way. The “faulty” entrance door should simply be ignored.

### **Options “Also show / delete linked objects”**


When showing/deleting an object of the first class of the link (**→Show** ), the AutoCAD graphic of objects linked with it are shown as well if the switch “**Also show / delete linked objects**” is enabled.

Example:

A workplace does not have an own graphic but only alphanumerical data (e.g. workplace number). However, it consists of several objects of the class inventory. The class workplace is thus linked with inventory. If a workplace is selected, (by clicking the label or via the command → Show workplace in context menu) all pieces of inventory are to be displayed with it. This can be done with the switch “Also show linked objects”.

When converting feature data objects into AutoCAD blocks, this setting can also be relevant (see context menu of the class **→Export →Blocks**).


**Remove link**

	Link definition
	available for concrete link definitions

A link can be removed any time.

**Caution!** This command cannot be reversed.

**Configure... (links)**

	Link definition
	available for concrete link definitions

The configuration of an existing link provides the same options as the command **→Add link**.