STF

Simple Textfile Format

DIALux - Minimal Text IO Interface

Version 1.0.5

Supported by DIALux Version 4.7 and above

Lüdenscheid, Germany

1. Contents

1. CONTENTS....................................2

2. ABSTRACT....................................3

2.1. CONTENTS OF A SIMPLE TEXTFILE....................................3

3. THE FILE FORMAT....................................3

3.1. GENERAL....................................3

3.2. SECTIONS AND KEYS....................................4

3.2.1. Section Version....................................4

3.2.2. Section Project....................................5

3.2.3. Section Room....................................5

3.2.4. Luminaire Structure Section....................................7

3.2.5. Luminairetype Section....................................9

3.2.6. ~~Furniture Structure Section~~....................................10

3.2.7. Furniture / Object Material Section....................................11

3.3. ADDITIONAL FILES....................................11

4. EXAMPLES....................................12

4.1. PROJECT WITH NO LUMINAIRE INFO....................................12

4.2. PROJECT WITH LUMINAIRE INFO....................................13

4.3. PROJECT WITH FURNITURE INFO....................................16

5. CHANGES....................................19

2. Abstract

The Simple File Interface for DIALux is designated for quick-and -simple connection to Configuration-, CAD, AVA- and other similar programs.
The main objective is not to enable complex geometry transfer, but to allow for simple and easy transfer of lighting plans and luminaire selections.

2.1. Contents of a Simple Textfile

- 1 project
- With n polygonal rooms (flat floor and ceiling, no room elements)
- Each room with any number of luminaire types and positions.
- Any number of luminaire type descriptions.
- Structures of several luminaire combinations.
- Luminaire description text ~~(optional)~~.
- Furniture / object position, size and name of predefined DIALux™ furniture / objects.
- Windows and doors position and size.
- Reflection factor and color information of furniture / objects , windows and doors.
- ~~Structures of several furniture combinations.~~
- ~~Material description of furniture elements.~~

3. The file format

3.1. General

- The text file is written in ASCII format.
- The overall outline of the file is in the [key] tag=<value> format, familiar from the windows initialization files.
- Keys and tags are not case sensitive.
- For not required tags, the mentioned default values are assumed, if the tag is not written in the file.
- Real numbers are written with a dot as decimal separator, and an e ex exponent operator (although there should be no need to use exponents).
- Reflection factors are to be specified in the range 0.01 - 0.99
- The file suffix .stf is used for simple text files. Hint: Microsoft uses the same suffix for the “Microsoft Setup File” file type, which makes it impossible to register the extension for another application.
- DIALux 4.1 (and above) writes all luminaire info into the file, reads in all luminaire structures but replaces all used luminaires with “dummy” luminaires when importing a STF file (STF Placeholder). In DIALux a STF Placeholder has to be replaced by real luminaires and then the position/mounting type has to be adjusted.
- DIALux 4.7 (and above) writes all furniture / object info into the file, but won't read in any furniture / object information.
- Windows, doors and skylights are supported.

3.2. Sections and Keys

The table column “Required” indicates if a key-value pair is required on input of a new project.
For not required tags, the mentioned default values are assumed, if the tag is not written in the file.

3.2.1. Section Version

required; Describes the STF file version

Section tag [VERSION]

Key	*Required*	*Value*
STFF	y	String, STF file version (e.g. 1.0.5)
Progname	y	String, Program name of data source (e.g. DIALux)
Progvers	y	String, Program version of data source (e.g. 4.7.0.0)
Simple	n	Integer; resolve structures to single objects: 0 : (default) Structures are supported. 1 : Resolves structures, single objects are written on export. (Supported when a STF file with this key set is loaded in DIALux and then is exported.)
Coords	n	Integer, explicit coordinates of each object local in structures 0 : (default) Explicit coordinates are not written local in structures. 1 : Explicit coordinates of luminaries are written local in structures. (Supported when a STF file with this key set is loaded in DIALux and then is exported.) (See 3.2.4. Luminaire Structure Section)
~~LumText~~	~~n~~	~~Integer, additional luminaire description text will be placed into the luminaire type section~~ ~~0 off~~ ~~1 on~~ ~~(default=0 or set in dialux.ini in the section [STFF])~~
~~CharSet~~	~~n~~	~~String; char set identification; Range: ANSI, ASCII, DEC, HP, IBMPC~~ ~~(default=ANSI or set in dialux.ini in the section [STFF])~~
~~Furns~~	~~n~~	~~Integer; add furniture information to STF File on output~~ ~~(default=1)~~

3.2.2. Section Project

required; Describes the project

Section tag [PROJECT]

Key	***Required***	***Value***
Name	n	String, name of project
Date	n	yyyy-mm-dd, date of project

Planer	n	String; Address of planer. Multiline separated by ~~
		Line1 = Company Name 1
		Line2 = Company Name 2
		Line3 = Street
		Line4 = ZIP and city
		Example:
		DIAL GmbH~~~~Gustav-Adolf-Straße 4~~D-58507 Lüdenscheid

Description	n	String; Short project description; max. 1024 characters. There are two supported special escape sequences for text formatting: \t TAB \n New Line
Operator	n	String; Operator of project
NrRooms	y	Integer; Number of rooms in the project
Room<nr>	y	where <nr> is 1 ...NrRooms Room reference; This string is used as reference to a section tag with the same string, which describes the room date

3.2.3. Section Room

required if NrRooms > 0; Describes the room to be used

Section tag: [<room reference string>]; see [Project] Room<nr>

Key	***Required***	***Value***
Name	n	String; Name of room
Description	n	String; Short room description; max. 1024 characters. There are two supported special escape sequences for text formatting: \t TAB \n New Line
NrPoints	y	Integer; Number of polygon points for base polygon
Point<nr>	y	2 real numbers separated by space; <nr> = 1 ... NrPoints; 2D coordinates for the polygon.
Height	y	Real number; Height of room; default = 2.8
WorkingPlane	n	Real number; Height of working plane; default = 0.85
R_Ceiling	n	Real number; Reflection factor for ceiling; default = 0.7
R_Floor	n	Real Number; Reflection of floor; default = 0.2
R_Wall<nr>	n	Real number; Reflection factor of wall between Point <nr> and <nr+1>; <nr> is 1 ... NrPoints; default = 0.5
MF	n	real number; 0.01 ... 1.0; Maintenance factor for plan; default = 0.8
NrLums	y	Integer; number of luminaires; can be 0 for room without any luminaires
Lum<nr>	y	String; <nr> is 1 ... NrLums; Luminaire reference; This string is used as reference to a section tag with the same string, which describes the luminaire

Lum<nr>.Pos	y	3 real numbers; 3D coordinates. **These coordinates represent the point in the room on which the center of the bounding box of the luminaire is placed.** (the size of the luminaires bounding box is given in the luminaire section)
Lum<nr>.Rot	n	3 real numbers; rotation angles (in °) around x,y,z; rotation sequence z,x,y; default = 0 0 0
~~Lum<nr>.Ref~~	~~n~~	~~String; Luminaire reference string for external references.~~
NrStruct	y	Integer; number of luminaire structures; can be 0 for rooms without any luminaire structures.
Struct<nr>	y	String; <nr> is 1 ... NrStruct; Luminaire structure reference; This string is used as reference to a section tag with the same string, which describes the luminaire structure.
Struct<nr>.Pos	y	3 real numbers; 3D coordinates of luminaire structure (reference point)
Struct<nr>.Rot	n	3 real numbers; rotation angles (in °) around x,y,z; rotation sequence z,x,y; default = 0 0 0
NrFurns	n	Integer; number of furniture / objects; default = 0
Furn<nr>	y	String; <nr> is 1 ... NrFurns; Furniture / object reference; This string is used as reference to DIALux™ furniture / object file. The name is stored without directory and file extension information. (Special treatment for windows, doors and skylights: **win** defines a window; **door** defines a door. **skylight** defines a skylight)
Furn<nr>.Pos	y	3 real numbers; 3D coordinates of furniture / object position in the room. The furniture's origin is the center of it's bounding box. (See Hints for furniture / object treatment)
Furn<nr>.Rot	n	3 real numbers; rotation angles (in °) around x,y,z; rotation sequence z,x,y; default = 0 0 0 (DIALux ignores rotation for windows, doors and skylights when reading an STF file because windows, doors and skylights are snapped on walls / ceiling.)
~~Furn<nr>.Scale~~	~~n~~	~~3 real numbers; scaling factor x,y,z; default 1 1 1~~
Furn<nr>.Size	y	3 real numbers; Size x,y,z of the furniture / object in m; (See Hints for furniture / object treatment)
Furn<nr>.Ref	y	String; Furniture / object reference; This string is used as reference to a section tag with the same string, which describes the furniture's / object's material information.
~~NrFStruct~~	~~n~~	~~Integer; number of furniture structures;~~
~~FStruct<nr>~~	~~y~~	~~String; <nr> is 1 ... NrFStruct;~~ ~~Furniture structure reference; This string is used as reference to a section tag with the same string, which describes the furniture structure.~~
~~FStruct<nr>.Pos~~	~~y~~	~~3 real numbers; 3D coordinates of furniture structure (reference point)~~
~~FStruct<nr>.Rot~~	~~n~~	~~3 real numbers; rotation angles (in °) around x,y, z; rotation sequence z,x,y; default 0 0 0~~
~~FStruct<nr>.Scale~~	~~n~~	~~3 real numbers; scaling factor x,y,z; default 1 1 1~~
MeanLuxWorkingPlane	n	Real number; Mean lux value on the working plane in lx. Hint: This key is only available in a room which is actually calculated in DIALux.
SpecificConnectedLoad	n	Real number; The specific connected load in the room in W/m².

Isolines	n	String; Filename of a DXF file which contains the isolines of the working plane in the room. Hint: This key is only available in a room which is actually calculated in DIALux.

Hints for furniture / object treatment

Windows and doors are treated in DIALux not as other furniture / objects. The definition of the coordinate origin is the major difference. Another difference is that windows and doors are two-dimensional objects, the other furniture / objects are tree-dimensional objects. So the z-size of a window / door is ignored, but always written in a STF file (default z-size value is 0).

- Furniture / objects except windows and doors:
  The coordinate origin of a furniture / object is defined as the center of the rotated furniture / objects bounding box. All position entries refer to this point.
  The Furn<nr>.Size written by DIALux is the size of the not rotated furniture / objects bounding box. (length, width, height)
- Windows and doors:
  Windows and doors are defined as a rectangular object in the rooms x-y-plane. The x- and y-size define width and length of the window/door. The z-size is ignored, but always written in a STF file (default z-size value is 0). The coordinate origin of a window / door is defined as the middle point in x direction where y is 0. All position entries refer to this point.
- Skylights:
  Skylights are defined like windows and doors, except that the coordinate origin is defined as the center of the rectangle.

3.2.4. Luminaire Structure Section

required if NrStruct > 0; for each luminaire structure specified in the room there must be a section describing the luminaire structure. The "Required" column indicates what must be set if a luminaire structure is referenced in the room section.

Section tag: [<Luminaire structure-tag>] ... depends on project

Key	***Required***	***Value***
Type	y	String; structure type; Range: FIELD, LINE, CIRC~~, FREE~~
~~Lock~~	~~n~~	~~Integer; Structure locking for:~~ ~~0 normal, structure can be modified any way (default)~~ ~~1 structure can only be moved, but not breaked a part~~ ~~2 movement and break a part is not possible~~

Type=FIELD

Lum	y	String; Luminaire reference; This string is used as reference to a section tag with the same string, which describes the luminaire
Lum.Rot	n	3 real numbers; rotation angles (in °) ar ound x,y,z; rotation sequence z,x,y; default = 0 0 0
Extend	y	2 real numbers; x y extend in meter
NrLums	y	2 integers; number of luminaires in x and y direction
~~Chess~~	~~n~~	~~Integer; Checkerboard like arrangement?~~ ~~0 Rectangular (default)~~ ~~1 Checkerboard~~

Arrange	n	Integer; Luminaire fields are always given in center-center coordinates. ~~Integer; Field arrangement code:~~ ~~0 symmetrical (default)~~ 1 center-center ~~2 edge-edge~~
Lum<nr>.Pos	n (ignored)	3 real numbers; 3D coordinates of luminaire (center coords). **Only written if one or more luminaires of the field are placed outside the room or Coords=1 is set in the imported STF file! Not read in by DIALux, because the whole field is read in.**
Lum<nr>.Rot	n (ignored)	3 real numbers; rotation angles (in °) around x,y,z ; rotation sequence z,x,y; **Only written if one or more luminaires of the field are placed outside the room or Coords=1 is set in the imported STF file! Not read in by DIALux, because the whole field is read in.**

Type=LINE

Lum	y	String; Luminaire reference; This string is used as reference to a section tag with the same string, which describes the luminaire
Lum.Rot	n	3 real numbers; rotation angles (in °) ar ound x,y,z; rotation sequence z,x,y; default = 0 0 0
Extend	y	1 real number; length in meter
NrLums	y	Integer; number of luminaires. Value > 1.
Arrange	n	Integer; Luminaire lines are always given in center-center coordinates. ~~Integer; Line arrangement code:~~ ~~0 symmetrical (default)~~ 1 center-center ~~2 edge-edge~~
Lum<nr>.Pos	n (ignored)	3 real numbers; 3D coordinates of luminaire (center coords). **Only written if one or more luminaires of the line are placed outside the room or Coords=1 is set in the imported STF file! Not read in by DIALux, because the whole line is read in.**
Lum<nr>.Rot	n (ignored)	3 real numbers; rotation angles (in °) around x,y,z ; rotation sequence z,x,y; **Only written if one or more luminaires of the line are placed outside the room or Coords=1 is set in the imported STF file! Not read in by DIALux, because the whole line is read in.**

Type=CIRC

Lum	y	String; Luminaire reference; This string is used as reference to a section tag with the same string, which describes the luminaire
Lum.Rot	n	3 real numbers; rotation angles (in °) ar ound x,y,z; rotation sequence z,x,y; default = 0 0 0
Extend	y	2 real numbers; x y radii in meter
NrLums	y	Integer; number of luminaires
Sector	n	2 real numbers; start and stop angle of circular arrangement (default: 0-360°)
Lum<nr>.Pos	n (ignored)	3 real numbers; 3D coordinates of luminaire (center coords). **Only written if one or more luminaires of the circle are placed outside the room or Coords=1 is set in the imported STF file! Not read in by DIALux, because the whole circle is read in.**

Lum<nr>.Rot	n (ignored)	3 real numbers; rotation angles (in °) around x,y,z ; rotation sequence z,x,y; **Only written if one or more luminaires of the circle are placed outside the room or Coords=1 is set in the imported STF file! Not read in by DIALux, because the whole circle is read in.**

This structure type is not supported any longer. Use single luminaries instead.

~~Type=FREE~~

~~NrLums~~	~~y~~	~~Integer; number of luminaires; can be 0 for new project~~
~~Lum<nr>~~	~~y~~	~~String; <nr> is 1 ... NrLums;~~ ~~Luminaire reference; This string is used as reference to a section tag with the same string, which describes the luminaire~~
~~Lum<nr>.Ref~~	~~n~~	~~String; Luminaire reference string for external references.~~
~~Lum<nr>.Pos~~	~~y~~	~~3 real numbers; 3D coordinates of luminaire (center coords)~~
~~Lum<nr>.Rot~~	~~n~~	~~3 real numbers; rotation angles (in °) around x,y, z; rotation sequence z,x,y; default 0 0 0~~
~~NrStruct~~	~~y~~	~~Integer; number of luminaire structures; can be 0 for single placed luminaires or new project~~
~~Struct<nr>~~	~~y~~	~~String; <nr> is 1 ... NrStruct;~~ ~~Luminaire structure reference; This string is used as reference to a section tag with the same string, which describes the luminaire structure.~~
~~Struct<nr>.Pos~~	~~y~~	~~3 real numbers; 3D coordinates of luminaire structure (reference point)~~
~~Struct<nr>.Rot~~	~~n~~	~~3 real numbers; rotation angles (in °) around x,y, z; rotation sequence z,x,y; default 0 0 0~~

3.2.5. Luminairetype Section

required; for each luminaire type specified in the project there must be a section describing the luminaire. Note: DIALux only writes these information and ignores them when importing an STF file.

Section tag: [<Luminaire-tag>] ... depends on project

Key	***Required***	***Value***
Manufacturer	y	String; Manufacturer Name
OrderNr	y	String; Order number
Name	y	String; Luminaire name
~~Ref~~	~~n~~	~~String; Additional reference to the Luminaire~~
Box	n	3 real numbers; Bounding box of Luminaire (size)
Shape	n	Integer; Shape code 0 Rectangular (default) 1 Vertical Cylinder
~~RefPoint~~	~~n~~	~~3 real numbers; defining the reference point of the Luminaire to the Positions specified in the Room sections. If this key is not specified, the Center (!) of the Box is used.~~
Load	n	Real number; Electrical connected load of the Luminaire
Flux	n	Real number; Flux of Luminaire
CF	n	Real number; 0.01 ... 1; lamp correction factor; default=1
NrLamps	n	Integer; Number of lamps in this luminaire

MountingType	n	Integer; Any combination of valid mounting types SurfaceMounted 0x01 Standing 0x02 WallMounted 0x04 Recessed 0x08 Pendant 0x10 TableStanding 0x20 WallRecessed 0x40 FloorRecessed 0x80
Picture	n	String; Filename of the picture for this luminaire. The file format can be of any type supported by the Window OleLoadPicture() method. Mostly used file formats are JPG, BMP and PNG.
Model	n	String; Filename of the 3D SAT model for this luminaire. Only used if the luminaire uses a 3D model file in DIALux.
Description	n	String; A description text for the luminaire. There are two supported special escape sequences for text formatting: \t TAB \n New Line

~~3.2.6. Furniture Structure Section~~

~~required if NrFStruc > 0; for each furniture structure specified in the room or in other furniture sections there must be a section describing the furniture structure. The Required-column indicates what must be set if a furniture structure is referenced somewhere else in the project.~~
~~Section tag: [<Furniture structure-tag>] ... depends on project~~

~~Key~~	***~~Required~~***	***~~Value~~***
~~Lock~~	~~n~~	~~Integer; structure locking for:~~ ~~0 normal, structure can be modified any way (default)~~ ~~1 structure can only be moved, but not breaked a part~~ ~~2 movement and break a part is not possible~~
~~NrFurns~~	~~n~~	~~Integer; number of furniture~~
~~Furn<nr>~~	~~y~~	~~String; <nr> is 1 ... NrFurns;~~ ~~Furniture reference; This string is used as reference to DIALux™ furniture file. The name is stored without directory and file extension information.~~
~~Furn<nr>.Pos~~	~~y~~	~~3 real numbers; 3D coordinates of furniture (coords of foot)~~
~~Furn<nr>.Rot~~	~~n~~	~~3 real numbers; rotation angles (in °) around x,y, z; rotation sequence z,x,y; default 0 0 0~~
~~Furn<nr>.Scale~~	~~n~~	~~3 real numbers; scaling factor x,y,z; default 1 1 1~~
~~Furn<nr>.Ref~~	~~y~~	~~String; Furniture reference; This string is used as reference to a section tag with the same string, which describes the furniture's material information.~~
~~NrFStruct~~	~~n~~	~~Integer; number of furniture structures;~~
~~FStruct<nr>~~	~~y~~	~~String; <nr> is 1 ... NrFStruct;~~ ~~Furniture structure reference; This string is used as reference to a section tag with the same string, which describes the furniture structure.~~
~~FStruct<nr>.Pos~~	~~y~~	~~3 real numbers; 3D coordinates of furniture structure (reference point)~~
~~FStruct<nr>.Rot~~	~~n~~	~~3 real numbers; rotation angles (in °) around x,y, z; rotation sequence z,x,y; default 0 0 0~~
~~FStruct<nr>.Scale~~	~~n~~	~~3 real numbers; scaling factor x,y,z; default 1 1 1~~

3.2.7. Furniture / Object Material Section

optional; the furniture / object material section specifies the default attributes of a furniture / object. The current definition specifies only material information as refection factor and color.

Section tag: [<Furniture / object reference-tag>] ... depends on project

Key	***Required***	***Value***
Poly.Color	n	String; default material definition for this section Format: MMM RRR GGG BBB ~~NAME~~ MMM ... reflexion factor monochrome (0 ... 100) RRR ... reflexion factor red (0 ... 255) GGG ... reflexion factor green (0 ... 255) BBB ... reflexion factor blue (0 ... 255) ~~NAME ... native name of the color/material~~ (default: Poly.Color=52 215 164 63)
~~NrPolys~~	~~n~~	~~Integer; number of color definitions The number of polygons for the furniture.~~
~~Poly<nr>.Color~~	~~n~~	~~String; as shown in *Poly.Color*, <nr> 1 .. NrPolys~~ ~~If this tag is obmitted the Poly.Color tag is used in stead, otherwise the default from Poly.Color will be overridden.~~

3.3. Additional Files

With introduction of Version 4.1.x of DIALux, some additional files will be written in addition to the normal STF file. DIALux stores these files in an own subfolder with the same name as the STF file (without the extension ".stf").

The filenames in the STF are stored in a short relative path from the original position of the STF file. Absolute filenames must not be used.

Example:
In this example the STF file was saved as **C:\Temp\Test Project 1.stf** the DXF file with isolines will be stored as **C:\Temp\Test Project 1\ROOM.R1.dxf**

```
[ROOM.R1]
Isolines=Test Project 1\ROOM.R1.dxf
```

Following additional files are written:

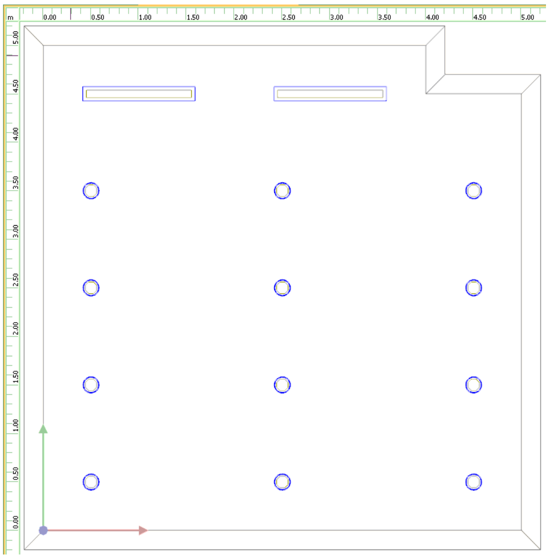
- **Picture** of the luminaire (for example as .JPG or .BMP)
- **Model** of the luminaire (as .SAT) (The stored file may contain volume based SAT model(s) or surface based 3DS model(s).)
- **Isolines** of the room (as .DXF only when project was calculated with DIALux.)

4. Examples

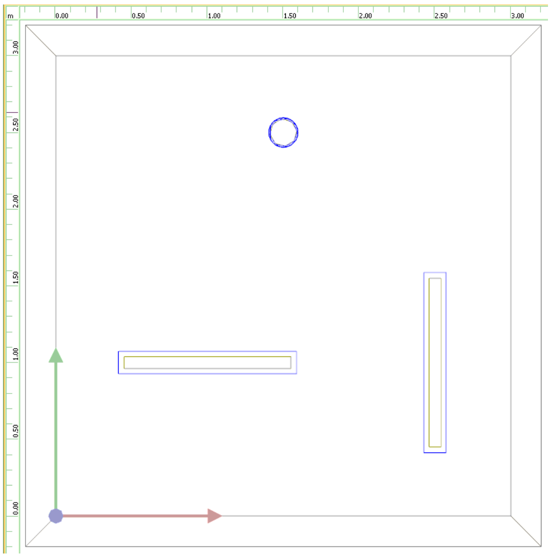
4.1. Project with no Luminaire Info

```
[VERSION]
STFF=1.0.5
Progname=DIALux
Progvers=4.7.0.0

[PROJECT]
Name=Demo Room
Date=2009-03-05
Planer=Company name line 1~~Company name line 2~~Street and number~~ZIP and city name
Description=Demoproject for STF document.\nHere only two rooms are defined without any other information.
Operator=The operators name
NrRooms=2
Room1=Room1
Room2=Room2

[Room1]
Name=Name of first room
Height=2.380
NrPoints=6
Point1=0.000 0.000
Point2=5.000 0.000
Point3=5.000 4.500
Point4=4.000 4.500
Point5=4.000 5.000
Point6=0.000 5.000
NrStruct=0
NrLums=0
NrFurns=0

[Room2]
Name=Name of second room
Height=2.380
NrPoints=4
Point1=0.000 0.000
Point2=3.000 0.000
Point3=3.000 3.000
Point4=0.000 3.000
NrStruct=0
NrLums=0
NrFurns=0
```

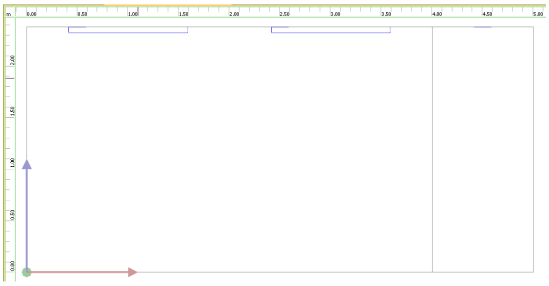
4.2. Project with Luminaire Info

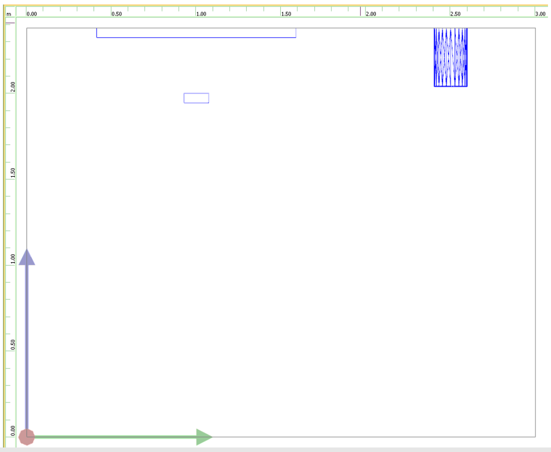


First Room



Second Room



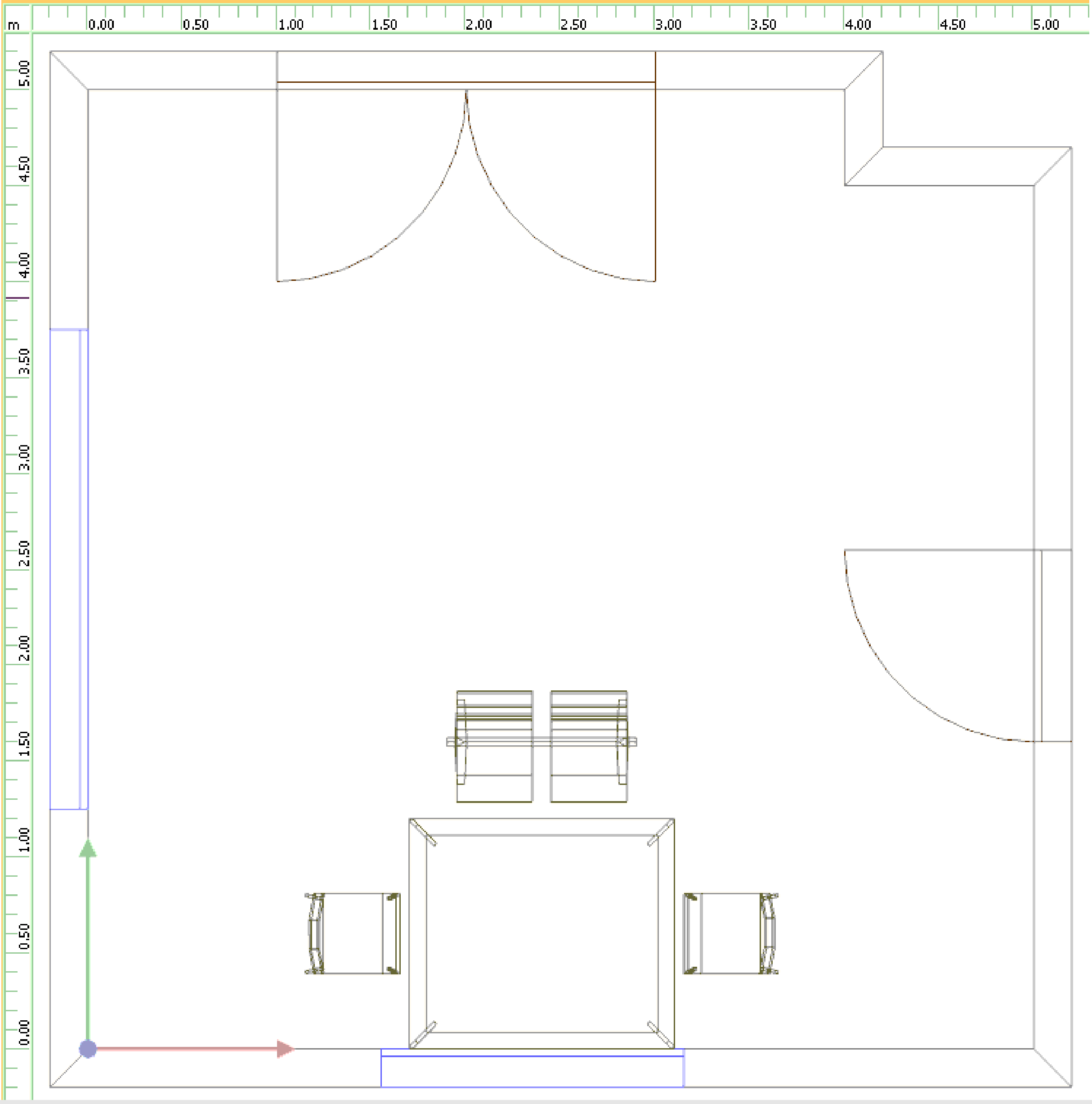


```
[VERSION]
STFF=1.0.5
Progname=DIALux
Progvers=4.7.0.0

[PROJECT]
Name=Demo RoomWithLuminaires
Date=2009-03-05
Planer=Company name line 1~~Company name line 2~~Street and number~~ZIP and
city name
Description=Demoproject for STF document.\nHere only two rooms are defined
without any other information.
Operator=The operators name
NrRooms=2
Room1=Room1
Room2=Room2

[Room1]
Name=Name of first room
Height=2.380
NrPoints=6
Point1=0.000 0.000
```

```
Point2=5.000 0.000
Point3=5.000 4.500
Point4=4.000 4.500
Point5=4.000 5.000
Point6=0.000 5.000
SpecificConnectedLoad=21.878
Lum1=LUMINAIRE.L2
Lum1.Pos=1.000 4.500 2.352
Lum1.Rot=0.000 0.000 90.000
Lum2=LUMINAIRE.L2
Lum2.Pos=3.000 4.500 2.352
Lum2.Rot=0.000 0.000 90.000
NrStruct=1
Struct1=Room1.LUMFIELD.LF1
Struct1.Pos=0.500 0.499 2.379
Struct1.Rot=0.000 0.000 0.000
NrLums=2
NrFurns=0

[Room1.LUMFIELD.LF1]
Type=FIELD
Lum=LUMINAIRE.L1
Lum.Rot=0.000 0.000 0.000
Extend=4.000 3.002
NrLums=3 4
Arrange=1

[Room2]
Name=Name of second room
Height=2.380
NrPoints=4
Point1=0.000 0.000
Point2=3.000 0.000
Point3=3.000 3.000
Point4=0.000 3.000
SpecificConnectedLoad=38.444
Lum1=LUMINAIRE.L3
Lum1.Pos=1.500 2.500 2.209
Lum1.Rot=0.000 0.000 0.000
Lum2=LUMINAIRE.L2
Lum2.Pos=2.500 1.000 2.352
Lum2.Rot=0.000 0.000 0.000
Lum3=LUMINAIRE.L2
Lum3.Pos=1.000 1.000 1.972
Lum3.Rot=0.000 0.000 -90.000
NrStruct=0
NrLums=3
NrFurns=0

[LUMINAIRE.L1]
Manufacturer=ERCO
Name=Compact HIT Downlight 1xHIT-CE 35W
OrderNr=83276000
Box=0.170 0.170 0.003
Shape=1
Load=35.000
Flux=3300.000
NrLamps=1
MountingType=8
Picture=STF105 Demo RoomWithLuminaires\LUMINAIRE.L1.jpg
Model=STF105 Demo RoomWithLuminaires\LUMINAIRE.L1.sat
Description=83276.000\nCompact HIT Downlight\nfür Halogen-
Metalldampflampen\nReflektorfarbe Silber\nGröße 5\nGehäuse und Einbauring:
```

```
Kunststoff, weiß (RAL9002), mit Spreizfederbefestigung bis max. 25mm
Deckendicke.\nAnschlussleitung mit Stecker, L 500mm.\nSpherolitreflektor
wide flood: Aluminium, silber eloxiert, hochglänzend. Abschirmwinkel 30°.
Abblendkappe: Aluminium, silber eloxiert, hochglänzend, am entspiegeltem
Schutzglas befestigt.\nInstallationseinheit separat bestellen.\nGewicht
0,79kg\nENEC10, VDE, Schutzklasse II, GOST

[LUMINAIRE.L2]
Manufacturer=ERCO
Name=TFL Wallwasher 1xT16 54W
OrderNr=65044000
Box=0.146 1.175 0.056
Shape=0
Load=58.000
Flux=4450.000
NrLamps=1
MountingType=1
Picture=STF105 Demo RoomWithLuminaires\LUMINAIRE.L2.jpg
Model=STF105 Demo RoomWithLuminaires\LUMINAIRE.L2.sat
Description=65044.000\nTFL Wallwasher\nfür Leuchtstofflampen\nWeiß
(RAL9002)\nEVG\nGehäuse: Aluminiumprofil, pulverbeschichtet.\n1
Leitungseinführungen. Durchverdrahtung möglich. 3polige Steckklemme.
Elektronisches Vorschaltgerät.\nWandfluterreflektor: Aluminium, silber
eloxiert, seidenmatt. Zum Lampenwechsel abklappbare Blende.\nGewicht
4,00kg\nENEC17, CCC, GOST

[LUMINAIRE.L3]
Manufacturer=ERCO
Name=Zylinder Aufbaudownlight 1xQT32 230W
OrderNr=85201000
Box=0.195 0.195 0.342
Shape=1
Load=230.000
Flux=4210.000
NrLamps=1
MountingType=1
Picture=STF105 Demo RoomWithLuminaires\LUMINAIRE.L3.jpg
Model=STF105 Demo RoomWithLuminaires\LUMINAIRE.L3.sat
Description=85201.000\nZylinder Aufbaudownlight\nfür Halogen-
Glühlampen\nGröße 7 40°\nWeiß (RAL9002)\nZylinder: Aluminium,
pulverbeschichtet.\n1 Leitungseinführung. 3polige
Steckklemme.\nDarklightreflektor: Aluminium, silber eloxiert, glänzend.
Abblendwinkel 40°.\nGewicht 1,40kg\nENEC10, VDE, CCC, GOST
```

4.3. Project with Furniture Info



```
[VERSION]
STFF=1.0.5
Progname=DIALux
Progvers=4.7.0.0

[PROJECT]
Name=Demo RoomFurniture
Date=2009-03-05
Planer=Company name line 1~~Company name line 2~~Street and number~~ZIP and
city name
Description=Demoproject for STF document.\nHere a rooms is defined with
furniture / objects and windows and doors.\n\n(No luminaires are added to
focus only on furniture / objects, windows and doors)
Operator=The operators name
NrRooms=1
Room1=Room1
```

```
[Room1]
Name=Name of first room
Height=2.380
NrPoints=6
Point1=0.000 0.000
Point2=5.000 0.000
Point3=5.000 4.500
Point4=4.000 4.500
Point5=4.000 5.000
Point6=0.000 5.000
NrStruct=0
NrLums=0
NrFurns=8

Furn1=120x140 delta
Furn1.Ref=Room1.F1
Furn1.Rot=0.000 0.000 0.000
Furn1.Pos=2.400 0.600 0.360
Furn1.Size=1.400 1.200 0.720

Furn2=Warten 2fach
Furn2.Ref=Room1.F2
Furn2.Rot=0.000 0.000 0.000
Furn2.Pos=2.399 1.575 0.395
Furn2.Size=0.998 0.579 0.789

Furn3=einfacher Stuhl
Furn3.Ref=Room1.F3
Furn3.Rot=0.000 0.000 -90.000
Furn3.Pos=3.400 0.600 0.443
Furn3.Size=0.420 0.501 0.886

Furn4=einfacher Stuhl
Furn4.Ref=Room1.F4
Furn4.Rot=0.000 0.000 90.000
Furn4.Pos=1.400 0.600 0.443
Furn4.Size=0.420 0.501 0.886

Furn5=win
Furn5.Ref=Room1.F5
Furn5.Rot=90.000 0.000 180.000
Furn5.Pos=2.350 0.000 0.800
Furn5.Size=1.600 1.250 0.000

Furn6=door
Furn6.Ref=Room1.F6
Furn6.Rot=90.000 0.000 -90.000
Furn6.Pos=5.000 2.100 0.000
Furn6.Size=1.000 2.000 0.000

Furn7=door
Furn7.Ref=Room1.F7
Furn7.Rot=90.000 0.000 0.000
Furn7.Pos=2.000 5.000 0.000
Furn7.Size=2.000 2.000 0.000

Furn8=win
Furn8.Ref=Room1.F8
Furn8.Rot=90.000 0.000 90.000
Furn8.Pos=0.000 2.500 0.800
Furn8.Size=2.500 1.500 0.000
```

```
[Room1.F1]
Poly.Color=50 186 186 186

[Room1.F2]
Poly.Color=50 186 186 186

[Room1.F3]
Poly.Color=50 186 186 186

[Room1.F4]
Poly.Color=50 186 186 186

[Room1.F5]
Poly.Color=10 5 5 5

[Room1.F6]
Poly.Color=56 255 128 0

[Room1.F7]
Poly.Color=56 255 128 0

[Room1.F8]
Poly.Color=10 5 5 5
```

5. Changes

From version 1.0.4 to 1.0.5

- Changed: Simple, Coords in 3.2.1. Section Version.
  ("Value" description was extended.)
- Removed: LumText, CharSet and Furns in 3.2.1 Section Version
- Changed: Planer in 3.2.2. Section Project.
  ("Value" description was extended.)
- Changed: Description in 3.2.2. Section Project.
- Changed: Description in 3.2.3 Section Room
- Removed: Lum<nr>.Ref in 3.2.3 Section Room
- Changed: Furn<nr> in 3.2.3 Section Room.
  ("Value" description was extended.)
- Changed: Furn<nr>.Pos in 3.2.3 Section Room.
  (Furniture / object origin)
- Changed: Furn<nr>.Rot in 3.2.3 Section Room.
  ("Value" description was extended.)
- Removed: Furn<nr>.Scale in 3.2.3 Section Room (Furn<nr>.Size is used instead)
- Added: Furn<nr>.Size in 3.2.3 Section Room
- Removed: NrFStruct in 3.2.3 Section Room
- Removed: FStruct<nr> in 3.2.3 Section Room
- Removed: FStruct<nr>.Pos in 3.2.3 Section Room
- Removed: FStruct<nr>.Rot in 3.2.3 Section Room
- Removed: FStruct<nr>.Scale in 3.2.3 Section Room
- Added: Hints for furniture / object treatment
- Removed: Lock in 3.2.4. Luminaire Structure Section
- Removed: Chess in 3.2.4. Luminaire Structure Section
- Changed: Arrange in 3.2.4. Luminaire Structure Section.
  (Is always = 1 center-center coordinates)
- Changed: Lum<nr>.Pos in 3.2.4. Luminaire Structure Section.
  ("Value" description was extended.)
- Changed: Lum<nr>.Rot in 3.2.4. Luminaire Structure Section.
  ("Value" description was extended.)
- Removed: Structure type FREE in 3.2.4. Luminaire Structure Section
- Removed: Ref in 3.2.5. Luminairetype Section.
- Removed: RefPoint in 3.2.5. Luminairetype Section.
- Added: Model in 3.2.5. Luminairetype Section.
- Removed: 3.2.6. Furniture Structure Section.
  (All furniture structures are not supported any longer.
  Furniture / object are only stored as single objects.)
- Removed: The material name is not supported any longer in 3.2.7 Furniture Material Section
- Removed: NrPolys in 3.2.7 Furniture Material Section
- Removed: Poly<nr>.Color in 3.2.7 Furniture Material Section
- Changed: Chapter 4 Examples