

Institute for Renewable Energy



# TypeDLT User Guide

## Contacts

Giuseppe De Michele	<a href="mailto:giuseppe.demichele@eurac.edu">giuseppe.demichele@eurac.edu</a>
Ulrich Filippi Oberegger	<a href="mailto:ulrich.filippi@eurac.edu">ulrich.filippi@eurac.edu</a>
Luca Baglivo	<a href="mailto:luca.baglivo@eurac.edu">luca.baglivo@eurac.edu</a>

---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	TRNSYS . . . . .	2
1.2	Radiance . . . . .	2
1.2.1	Three-Phase Method . . . . .	3
<b>2</b>	<b>TypeDLT</b>	<b>5</b>
2.1	Description and functionality . . . . .	5
2.2	Installation . . . . .	8
2.2.1	Radiance . . . . .	8
2.2.2	Perl environment for OS Windows . . . . .	8
2.2.3	TypeDLT . . . . .	8
2.3	Tool Chain . . . . .	10
2.3.1	BSDF data generation - WINDOW 7.3 . . . . .	10
2.3.2	Google SketchUp . . . . .	12
2.3.3	Manual tweaks . . . . .	13
2.4	TRNSYS . . . . .	17
2.4.1	Radiance parameters . . . . .	19

# CHAPTER 1

---

## Introduction

---

The TypeDLT is a new TRNSYS component dedicated to daylighting analysis of Complex Fenestration Systems (CFS). The tool is able to perform dynamic daylighting simulations using the Bidirectional Scattering Distribution Function (BSDF) that characterize the CFS. TypeDLT implements RADIANCE, a validated software for daylighting simulation, within TRNSYS, that in its standard configuration and use provides thermal simulations. In particular, the calculation algorithm implemented is the Three-Phase Method (3PM). This method allows to perform fast simulation climate-based using the BSDF data in input. The code, which defines the TypeDLT, contains only the functions needed to launch the 3PM. The control algorithm is maintained outside the TypeDLT in order to let the user a great flexibility in the control strategy definition.

### 1.1 TRNSYS

TRNSYS [1] is a complete and extensible simulation environment for the transient simulation of systems, including multi-zone buildings. One of the key factors in TRNSYS is its modular architecture based on DLL (Dynamic-link library) concept. Thanks to this feature, user and third-party developers may easily add custom component models using common programming language (e.g. C, C++, FORTRAN, etc.).

### 1.2 Radiance

Radiance [2] is a software developed at the Environmental Energy Technologies Division of Lawrence Berkeley National Laboratory in Berkeley, California, by Greg Ward and the Lighting Systems Research group. It is an accurate tool for predict the visible radiation in a space using climate-based approach even on an annual basis. Radiance uses the back ray tracing approach that consist in following the light path (specular reflected, transmitted and refracted) from the reception point (eyes or sensors) into the scene to the light source.

In the practice is consider the best and more flexible software for lighting simulation,

in fact is used as calculation engine in the most lighting design software available. It is also the most validate lighting simulation tool.

### 1.2.1 Three-Phase Method

The Three-Phase Method [3, 4] is an algorithm developed in Radiance that perform hourly/annually daylighting simulation for complex fenestration system and light pipes. The name is due to the sub-division of light path in three step (figure 1.1) defined in the calculation by three matrices. The three phases of the transferred flux are:

1. from sky dome to exterior of window, described by Daylighting matrix (D);
2. between the interior and exterior of the window, described by the BSDF matrix called also Transmission matrix (T);
3. from the interior of the window to the sensor points, described by the View matrix (V).

The results is achieved by multiplying the sky vector/matrix for matrix listed above.

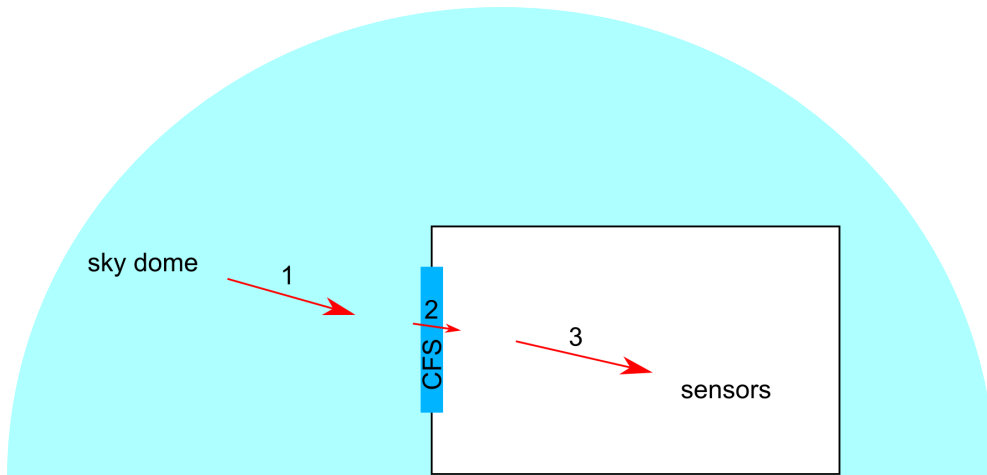


Figure 1.1: Three-Phase method description

The equations that describe the process are:

$$i = VT D_s \quad (1.1)$$

$$I = VTDS \quad (1.2)$$

where, by [5] :

- $i$  = point in time illuminance or luminance result;
- $I$  = matrix containing time series of illuminance or luminance result, annual simulation;
- $V$  = view matrix, relating outgoing directions on window to desired results at interior;

- T = transmission matrix, relating incident window directions to exiting directions (BSDF);
- D = daylight matrix, relating sky patches to incident directions on window;
- s = sky vector, assigning luminance values to patches representing sky directions;
- S = sky matrix, a collection of sky vectors.

The V and D matrices are created with a Radiance simulation. The T matrix can be created using LBNL WINDOW software, by simulation (e.g. TracePro or Radiance genBSDF) or can be measured with a goniophotometer. The s vector and S matrix are generated from a Radiance sky description [5]. Once evaluate the V,D and T matrix, that require a quite long time, these do not must be again calculated. Then the matrix multiplication is very fast, and in few second gives the results.

In particular, TypeDLT uses the equation 1.1 hourly-based. This strategy increases the calculation time but allows to connect the results of the daylighting simulation after each time step with the other Types onto the TRNSYS deck.

## 2.1 Description and functionality

The new TypeDLT (DayLightTing), written in the C++ programming language, performs the Three-Phase Method for dynamic, climate-based daylighting simulations of CFS taking in input the BSDF data [6]. The TypeDLT receives the following inputs from the weather file:

- Latitude and longitude
- Month, day of the month, hour of the day
- Direct normal illuminance
- Diffuse horizontal illuminance

In this way it can identify the building location (and thus calculate the sun path) and the information for the climate-based analysis. The following parameters are set in the Type's control panel:

- Zone ID, thermal zone for which the shading system is controlled
- Control1-10, gives which BSDF is used for each of the 10 control

These parameters are necessary to identify the thermal zone where CFS are controlled, which CFS assigned to that zone are controlled, and which are the initial conditions for the simulation. A BSDF data file has to be provided for each shading state. In order to independently control a series of thermal zones, the TypeDLT can be added as functional block to the simulation model multiple times, with different parameters, inputs, and outputs. The outputs of the Type are:

- Maximum (or average, minimum) illuminance value over the sensors grid
- Control1-10, current shading state for each control

The control algorithm, defined through a series of equations which can involve inputs and outputs of other Types, the TypeDLT and the Type56 are interlinked. Loops between Types can be formed. By default, TRNSYS performs iterations after each time step until all parameters, inputs and outputs of every Type have converged. Therefore, the main difference with other integrated software is the time-step interaction of the TypeDLT with the other Types contained into the TRNSYS deck.

The control algorithm is not incorporated into the TypeDLT, but implemented through the "Equation" functional blocks that TRNSYS offers by default. This has the big advantage that the TypeDLT does not need to be recompiled every time the control strategy has changed and can thus be reused more easily by the user. This approach gives the maximum flexibility in the setting of a shading control strategy, since it is possible to test custom control algorithm taking in account all the main aspects of a sustainable design: energy saving, thermal and visual comfort, but with the detriment of calculation time.

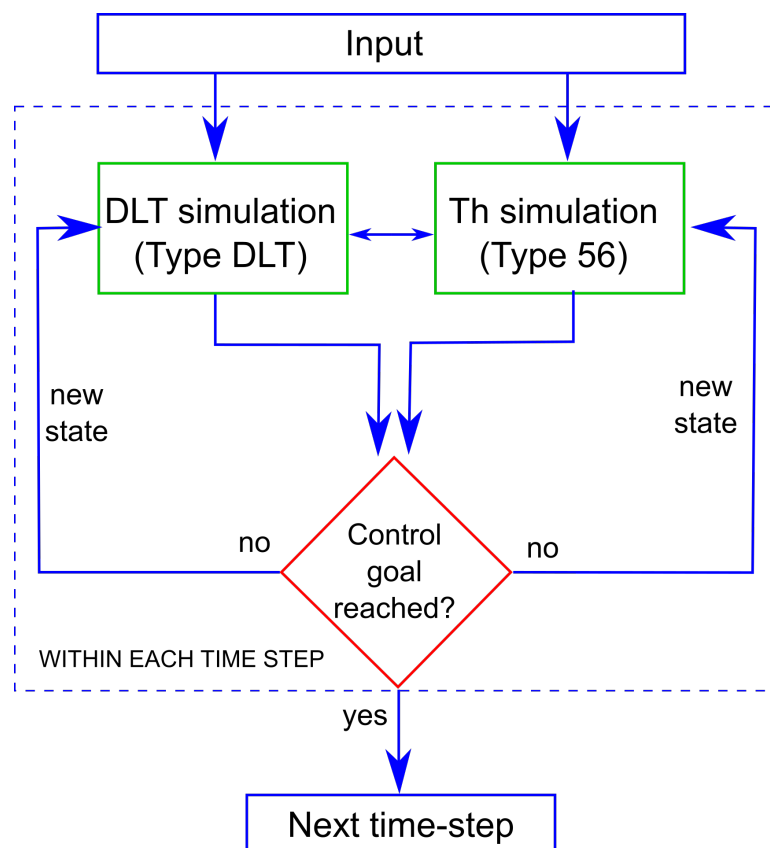


Figure 2.1: Flow chart of the iteration process between daylighting and thermal simulation within each time step

Figure 2.1 shows a typical workflow of the process. Both the daylighting (TypeDLT) and the thermal (Type56) simulations take their inputs. Then, the BSDF data and the Types' outputs are passed to a control "Equation" (a special functional block of TRNSYS in which an equation system can be defined) that modifies the shading states. By default, TRNSYS iterates until all Type outputs have reached convergence. Only then, it advances in time by performing the next time step.

At each time step the TypeDLT outputs can be shared with all the other types onto the TRNSYS deck (Figure 2.2), for example can be used for the artificial light calculations, in the control algorithm and/or passed to the Type56 as shading factor.

The advantages available by the use of TypeDLT are:

- Customized shading control based on: daylighting parameters, thermal parameters, weather file parameters or a combination of that
- Use of the BSDF file as input data
- Multi-zone daylighting analysis by adding more TypeDLT
- Easy calculation of artificial lighting usage
- No limit in the complexity of the model geometry
- Complete overview of the CFS effects on building in terms of energy demand and thermal and visual comfort
- Possibility to control up to 10 windows/group of windows with different strategies.

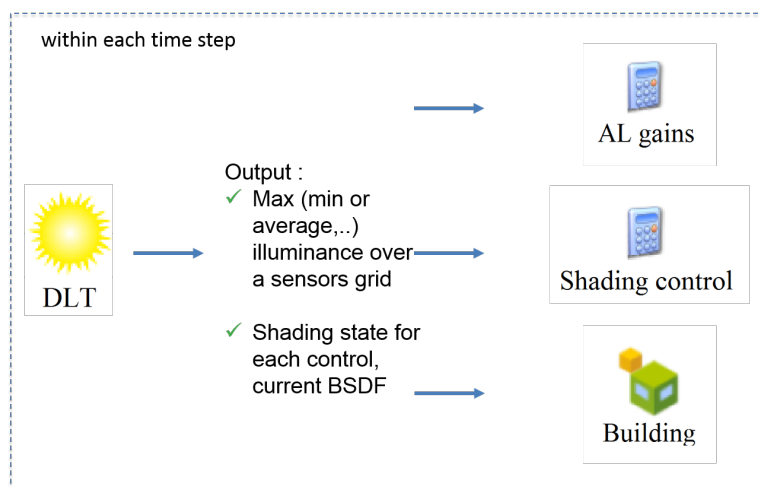


Figure 2.2: TypeDLT outputs and possible connection with other types

The disadvantages are due to:

- Time consuming simulation
- Not user-friendly simulation set-up

The hourly-based approach allows a complete information sharing between the TypeDLT and the rest of the Types over the TRNSYS deck at each time-step, on the other hand this strategy take a lot of time because a daylighting simulation is performed at each time-step slowing down the energy simulation with TRNSYS.

As explained in the following pages, the phase of set-up of the simulation take time if the user is not yet familiar with the procedure and required the use of different software. The feature of a multiple shadings control within the same scene is currently under development, and will be available in next release.



## 2.2 Installation

Below is sorted the list of software required for the TypeDLT running. Keep in mind that those software are only suggested, except for Radiance, and the user is free to use an equivalent software that does the same job.

### 2.2.1 Radiance

Download the binary for Windows from <https://github.com/NREL/Radiance/releases> and run the exe. Install Radiance to (*C:\Radiance*).

### 2.2.2 Perl environment for OS Windows

A Perl interpreter is necessary to run Perl application on Windows. In fact, some of the functions used in Three-Phase Method are written in Perl programming language, therefore an interpreter is required. For this purpose is possible get the proper version of Perl from <http://strawberryperl.com/> install it and make sure to add the Perl executable directory to your PATH (the installers will optionally do this for you).

### 2.2.3 TypeDLT

Recompiling the DLT.dll file became necessary if you use a system different from Windows 8.1 64 bit, for which the TypeDLT has been compiled, or if the C++ code has been modified.

For this step, the batch file *compilation\_into\_dll.bat* is supplied. The batch file contains two lines of code that execute mingw32-g++.exe which creates the DLL starting from the C++ code. Follow the steps below to use the batch file:

1. Install code::block from <http://www.codeblocks.org/downloads>.
2. Install mingw-get-setup.exe from <http://sourceforge.net/projects/mingw/files/latest/download?source=files> and install it in C: without space.
3. Open Environment variables (see <http://www.computerhope.com/issues/ch000549.htm> if you are not familiar with) add

*C:\MinGW\bin*

to the PATH as shown in figure 2.3.

4. Open the Windows terminal following the steps:
  - Type WinKey + R
  - Input "cmd"
  - Type Enter
5. Move into the MinGW folder with the command

*cd C:\MinGW\bin*

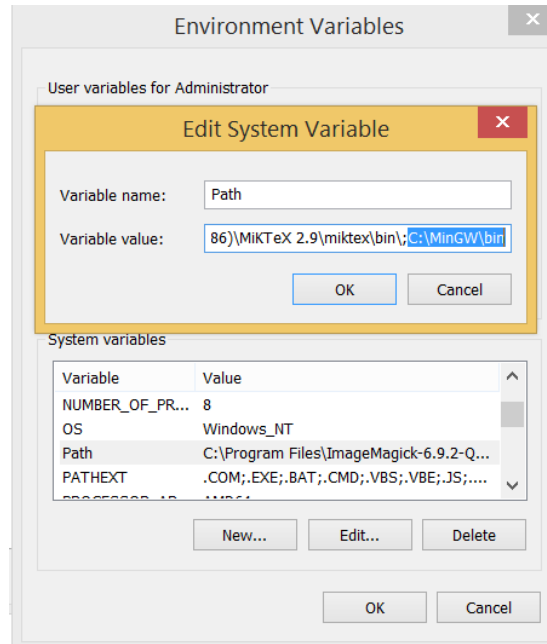


Figure 2.3: Environment variable setting

and run the command

*mingw-get install c++*

as in Figure 2.4.

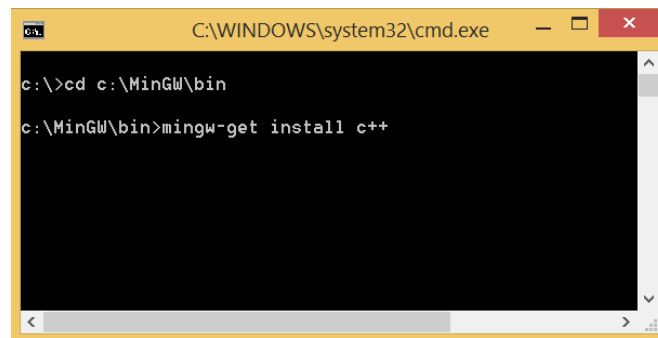


Figure 2.4: Terminal with the commands to launch

6. Launch the batch file *compilation\_into\_dll.bat* within the folder that we provided you. This because the file needs other files contained in this folder.

Once the DLT.dll is created the file has to be located in the TRNSYS folder

*C:\Trnsys17\UserLib\ReleaseDLLs*

The proforma files (.tfs and .btp) have to be placed in the proformas folder within an existing folder or in a new one, such as

*C:\Trnsys17\Studio\Proformas\mycomponent*

## 2.3 Tool Chain

In order to carry on with the daylighting simulation it is necessary to prepare the daylighting model, manipulate some configuration files and retrieve the BSDF data.

### 2.3.1 BSDF data generation - WINDOW 7.3

The BSDF data can be generated through several methods: window modelling software (e.g. WINDOW 7.3), simulation programs (e.g. TrancePro or Radiance's "genBSDF") or measurements with a goniophotometer. In this section will be given an overview of the BSDF file creation with the software WINDOW 7.3.

The software WINDOW 7.3, developed by the LBNL in California, allows to define or calculate the BSDF data for standard system and parameters such as the Solar Heat Gain Coefficient (SHGC or g-value), heat transfer coefficient (U-factor) and visible transmittance (Tvis).

Within the Glazing System Library (Figure 2.5) it is possible to create the fenestration system choosing the glass and the shade typology from the respective database, IGDB and CGDB. Figure 2.5 shows a system composed by a double pane glass with solar control and, as shading device, a system of louvre tilted at 30-degree tilt angle.

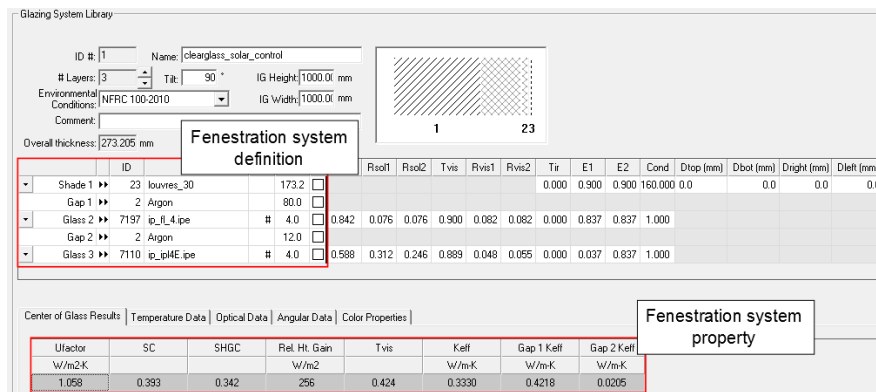


Figure 2.5: Glazing System Library

From the Shading Layer Library (Figure 2.6), it is also possible to assign a custom shading device choosing from the shading type: venetian blinds, homogeneous diffusing shade, woven shade, fritted shade. Then, for each typology, it is possible to choose geometric and optical parameters. WINDOW allows also to use BSDF data pre-calculated in the shading type by selecting "shade with XML data".

Once defined the fenestration system, it is required to set the optical calculation options in the software preferences (Figure 2.7). Checking the relative boxes, WINDOW gives in output the BSDF matrices of the complete fenestration system in a "xml" extension. In particular, it is possible also to have in output the BSDF data for both solar and visible band, remembering that for the daylighting simulations only the visible one is needed.

Particular attention should be paid to the angular basis option; this option defines the sky division in order to have a BSDF matrix 145x145 size, and to avoid error in the

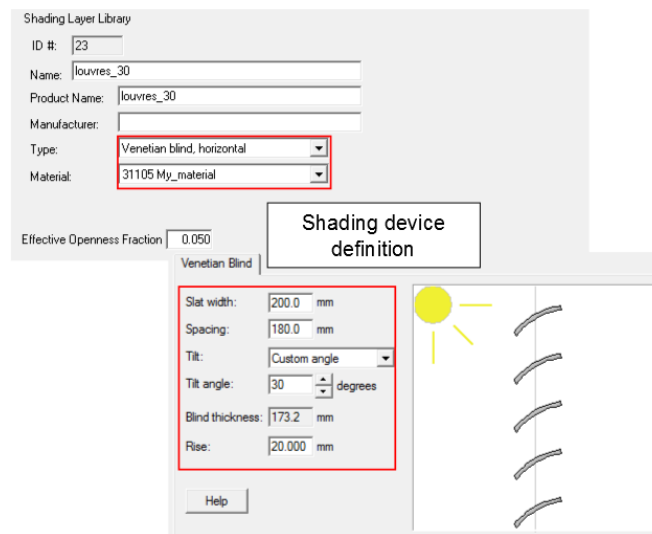


Figure 2.6: Shading Layer Library

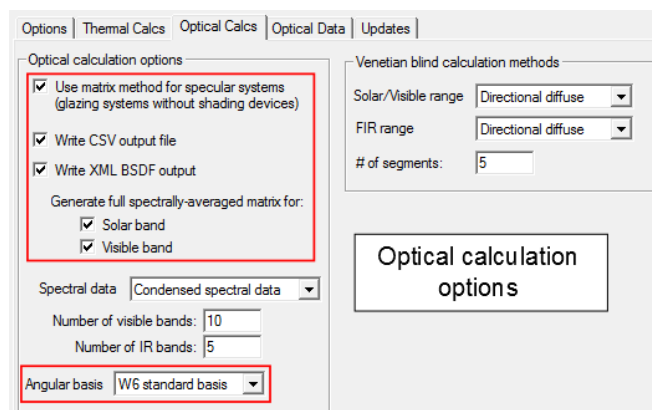


Figure 2.7: Optical preference setting - WINDOW 7.3

simulation. It is necessary to set the "W6 standard basis" (Figure 2.7) that correspond to the Klems division. The BSDF file has to be saved with a name defined by an integer number which corresponds to a specific shading state in the TypeDLT simulation (e.g. 1.xml). Once created the BSDF file it is essential to locate it in a specific folder, where the TypeDLT can find it. If the user will simulate a dynamic venetian system then will exist several BSDF files, one for each state of the blind.

**Importing new window system in TRNSYS** When a new window technology is created became necessary adding this new system to the Type56 window database. In TRNSYS will be add only the glazing part of the whole system, the shading will be considered apart from the glazing system using a shading factor ( $F_c$ ) that represent the percentage of shaded area of the glazing surface. The tutorial [http://sel.me.wisc.edu/trnsys/downloads/tutorials\\_and\\_examples/window5/windowtutorial.pdf](http://sel.me.wisc.edu/trnsys/downloads/tutorials_and_examples/window5/windowtutorial.pdf) will explain you how to add a new window in TRNSYS.

### 2.3.2 Google SketchUp

It has been chosen to use different geometries for the simulations, that are a model for the thermal simulation and another for the daylighting simulation; this choice is due to the large simplification of the thermal model. In fact, the latter model is defined as plan surfaces that represent walls, instead, the daylighting model needs a volumetric geometry in order to consider the presence of shading objects, such as the walls thickness, pillars, but also furniture. This strategy is also useful if a building with a large number of thermal zone is simulated, in this case is not necessary redraw all the zones, but only the zone that we want to analyse with the daylighting simulation. This is possible because the simulations (thermal and daylighting) work on different geometries. Geometry can be drawn with Google SketchUp. The SketchUp's plug-in su2rad allows to export the SketchUp geometry (\*.skp) in to Radiance geometry (\*.rad) with a useful file structure. The possible ways to export in Radiance geometry are:

- by group, for complex geometry;
- by material, small number of files;
- by layer, the more useful for geometry not so complex.

In order to simplify the procedure to define the reflection factor of the surfaces, two ways are suggested:

1. adding the material to the su2rad library, this procedure allows to set the reflection coefficient within SketchUp
2. creating an alternative materials.rad file to be substitute to the one exported by su2rad.

It is a good practice using standard material identifier that remember the name of the structural component (e.g. Internal\_walls). The exportation mode "by layer" is very useful if at each layer a material/color is associated. In this way the user can control the coefficient of reflection of the surfaces without difficulties acting directly on all the surfaces contained in the layer especially working with the Radiance materials out of Sketch-up. With the "by group" mode is crucial to set well the materials before the export process. Otherwise, finding possible mistakes in the \*.rad file becomes a hard task.

The name of the scene has to be defined considering the ID of the zone, that will be also the name of the folder containing the Radiance files (e.g. Zone1). As shown in Figure 2.8, su2rad generates a series of files. The folder "objects" contains all the elements that define the scene, each file takes the name of the layer (if the "by layer" mode is selected) and contains all the elements associate with that layer. Name the layer with the structural component (Figure 2.8) that compose the scene helps in maintain the order with the files and in the material/color association. Several folders contain file not used by the TypeDLT and in the next steps will be deleted.

The Zone1.rad and materials.rad files are the files managed by the user in order to set the inputs for the TypeDLT. The Zone1.rad file collects all the objects of the scene (Figure 2.9) and the materials file in order to have few files during the creation of the

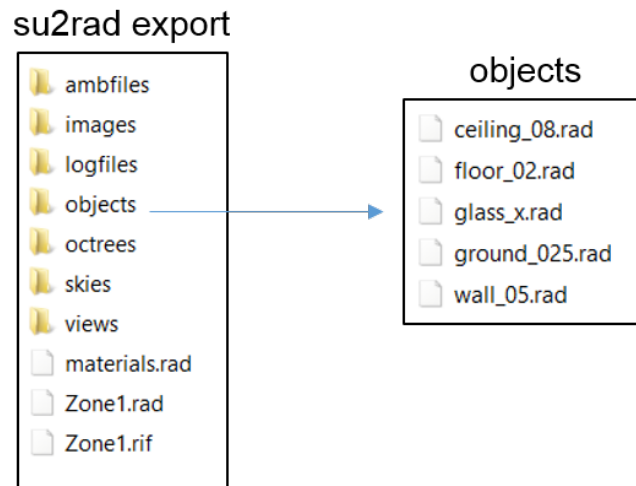


Figure 2.8: File rad exported by su2rad

octree file in Radiance. The materials.rad file contains the materials definition from SketchUp, in this file it is very easy to change the materials reflectance of the surfaces operating on the RGB channels in the last line of each material definition.

Zone1.rad

```
!xform skies/not_used_1108_1330.sky
!xform ./materials.rad

!xform objects/wall_05.rad

!xform objects/ground_025.rad

!xform objects/glass_x.rad

!xform objects/floor_02.rad

!xform objects/ceiling_08.rad
```

Figure 2.9: Content of the Zone1.rad file

**Window and group of windows** An important annotation have to be done about the windows grouping. Windows that have the same orientation and that receive the incident daylight without discrepancies due to external obstruction can be grouped in the same layer. In this way, the windows grouped in the same layer will have the same shading control and will not be possible to control them separately.

If you need more information on the windows grouping please consult the Appendix - Additional Consideration of the Three-Phase Method tutorial [5].

### 2.3.3 Manual tweaks

First is required to bring the folder create by the su2rad in the project folder of the TRNSYS simulation, and name the folder ZoneN where N is a positive integer number

that define the zone analysed.

In figure 2.10 is shown an example of how locate the folder in the right place. The integer number that identifies the zone is the same number that has to be defined in Type's control panel under the input *Zone ID* and the name of the Radiance scene, Zone1 in this case.

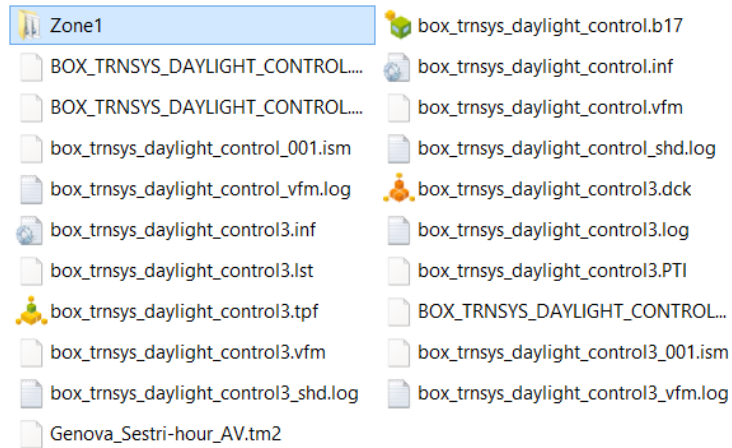


Figure 2.10: Positioning of the Radiance simulation folder in the TRNSYS project folder

Within the folder Zone1 is then necessary creating some folders:

- **data**, it will contain the matrices for the calculation, the file with the sensors grid and another subfolder:
  - **bsdf**, it will contain the bsdf data listed with integer numbers
- **temp**, it will contain temporary files create during the matrices generation
- **window**, it will contain the windows in the scene and a new file called win.vd

And deleting some other unused folder:

- ambfiles
- images
- logfiles
- octrees
- skies

In order to make available the input data for TypeDLT it is necessary to modify some files generated with SketchUp and to create a new one, which will collect all the windows in the scene. For this part, it is recommended to use a simple text editor. 3PM requires that the windows in the scene are a secondary light source, so that the user has to change the windows material in glow material, see Three-Phase Method tutorial for in-depth considerations [5]. To do this the user has to open the rad file

<pre>#objects/window_south.rad void glow window_south 0 0 4 1 1 1 0  window_south polygon window 0 0 12  0.5  -.15  1     0.5  -.15  2     3.5  -.15  2     3.5  -.15  1</pre>	<pre>#objects/window_south.rad void glow window_south 0 0 4 1 1 1 0  window_south polygon window1 0 0 12  0.5  -.15  1     0.5  -.15  2     2  -.15  2     2  -.15  1  window_south polygon window2 0 0 12  2.5  -.15  1     2.5  -.15  2     4  -.15  2     4  -.15  1</pre>
(a)	(b)

Figure 2.11: Window file modified with the glow material for one window 2.11(a) and a group of two windows 2.11(b)

containing the window/windows geometries and add in the heading of this file the glow definition as shown in figure 2.11. Then, move the window.rad file from the "objects" folders to the "window" folder.

A new file called [win.vd](#) has to be created within the "window" folder. In this new file insert the number of windows/group of windows and the list of this windows/group of windows following a specific format: window modifier, view direction (vd) and view up (vp). The view direction is the window normal vector facing toward the outside of the room, the view up is the window up vector, vd and vu are always orthogonal.

Figure 2.12 shows how to choose vd and vu from a sample geometry in which there is a window facing south. It is also shown how to report these information in the win.vd file for a window with modifier window\_south.

Delete some rows from the rad file that contains all the objects into the scene, tutorial.rad in this case (figure 2.13). In particular, the row that calls the sky definition and the window objects. In fact, TypeDLT creates at each time step a new sky file with the information contained in the weather file. And, the window/group of windows are used only for the matrices definition, in the calculation are then used the BSDFs data.

Add the file containing the sensors to be used in the simulations within the "data" folder with the name [grid.pts](#). The format to follow is: x, y, z, dx, dy, dz where x, y and z are the three dimensional coordinates of the point and dx,dy and dz are the coordinates of the vector that define view direction of the sensor. The example file (figure 2.14) is composed by row grid with 9 sensors located in the middle of the



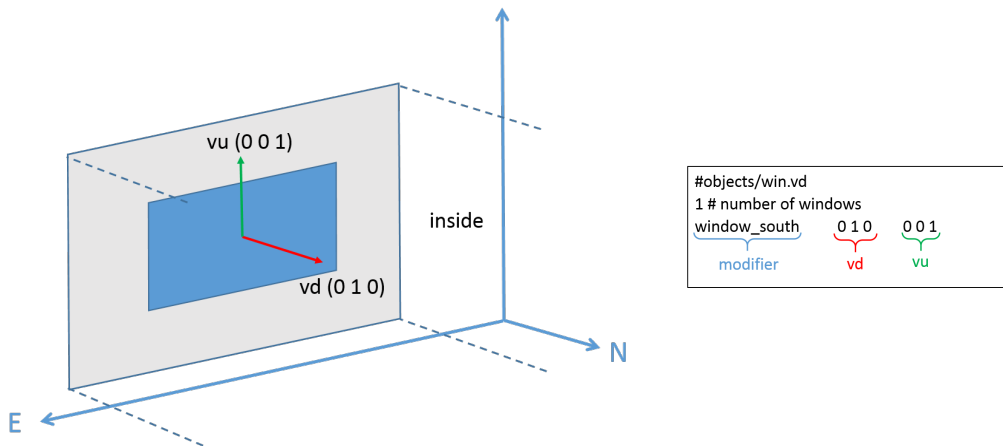


Figure 2.12: New file with the list of windows in the scene and orientation of the vu and vd vectors

```
tutorial.rad
1 !xform skies/not_used_1100_1330.sky
2 !xform ./materials.rad
3
4
5 !xform objects/ground.rad
6
7 !xform objects/window_south.rad
8
9 !xform objects/wall.rad
10
11 !xform objects/floor.rad
12
13 !xform objects/ceiling.rad
14
```

Figure 2.13: Contents of the tutorial.rad file

scene 1 meter from the window with a step of 0.5 m.

```
grid.pts
1 2.5 1 0.8 0 0 1
2 2.5 1.5 0.8 0 0 1
3 2.5 2 0.8 0 0 1
4 2.5 2.5 0.8 0 0 1
5 2.5 3 0.8 0 0 1
6 2.5 3.5 0.8 0 0 1
7 2.5 4 0.8 0 0 1
8 2.5 4.5 0.8 0 0 1
9 2.5 5 0.8 0 0 1
10
```

Figure 2.14: Format of the file containing the sensors grid

TypeDLT will create three files in the Zone folder: dmx.bat, vmx.bat and 3pm.bat. Those three batch files run the Radiance functions that generate respectively the day-lighting matrix and the view matrix, which will be generate in the "data" folder (figure

2.15). With the latter batch file the sky is created and the daylighting simulation performed. Figure 2.15 shows how the contents of the Zone folder look after the manual tweaks and a simulation in TRNSYS.

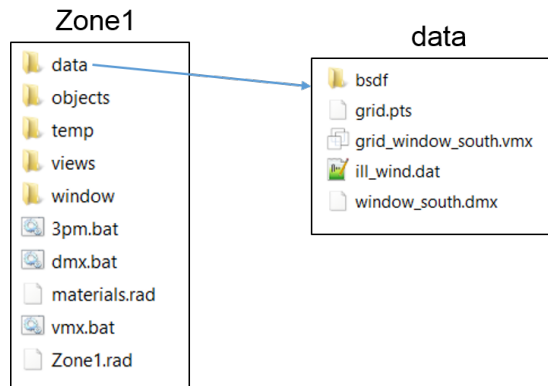


Figure 2.15: Contents of the Zone1 folder

## 2.4 TRNSYS

The creation of the thermal model can be done following the tutorial of TRNSYS3D plug-in for SketchUp in the TRNSYS documentation folder:

....\Trnsys17\Documentation\A4\_3DBuildingTutorial.pdf

Locate the TypeDLT onto the TRNSYS deck. First, connect the following variable from the *Weather data* Type:

- Latitude and Longitude
- Direct normal and diffuse horizontal illuminance
- Month, day of the month and hour of the day

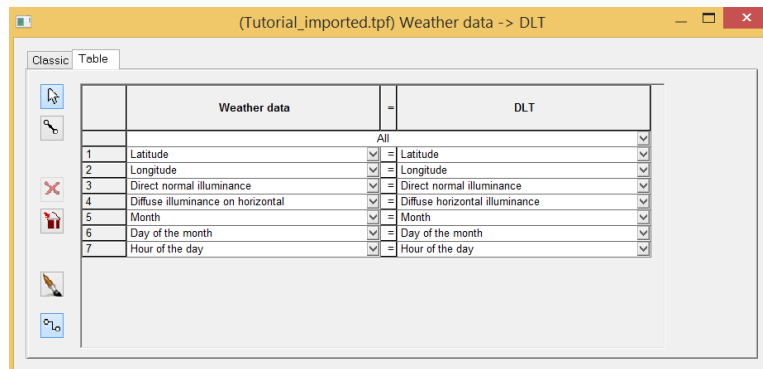


Figure 2.16: Connections between Weather data and TypeDLT

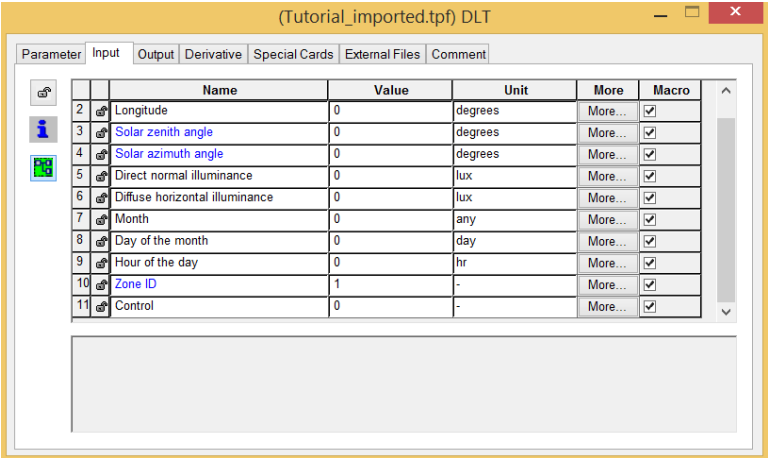


Figure 2.17: Input window TypeDLT

Double click on the TypeDLT icon, set the Zone ID that correspond to the integer number that you have assigned to the folder which contain the Radiance files on the input window. The default value in TRNSYS is 1 (Figure 2.17).

At this point, the TypeDLT is ready to simulate and the user can connect the Type's outputs and inputs following its needs.

Once the TRNSYS deck is ready and the first simulation is run the view and daylight-

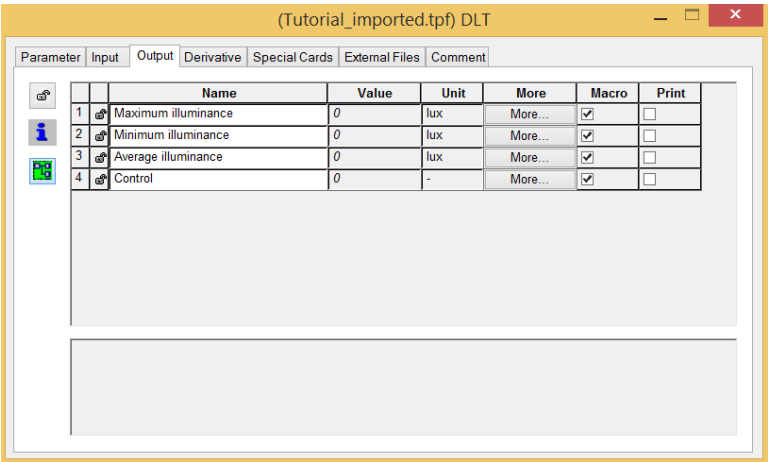
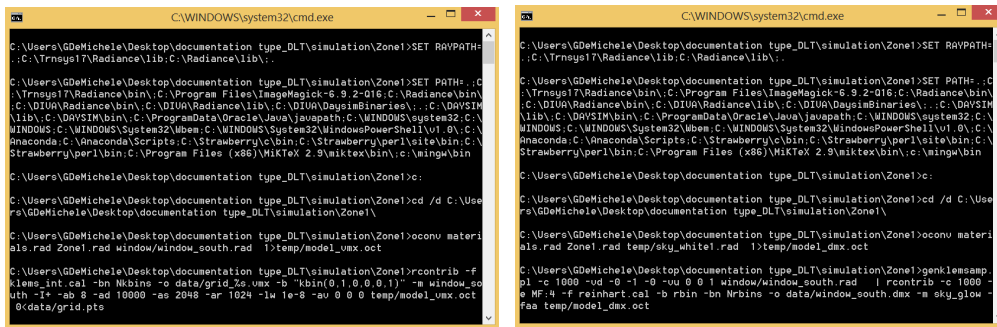


Figure 2.18: Output window TypeDLT

ing matrices (see section 1.2.1 for theory) are generated and two terminal windows will be opened, Figure 2.19.



(a) View matrix generation

(b) Daylighting matrix generation

Figure 2.19: Terminal windows opened for the view and daylighting matrix generation

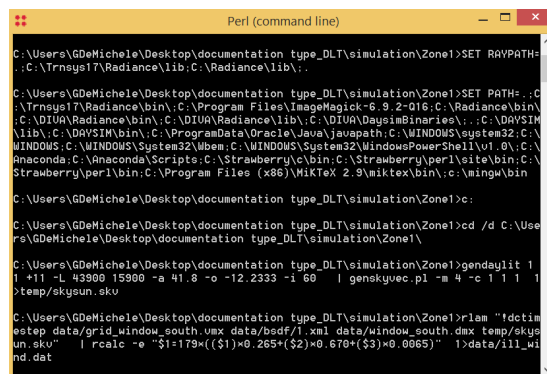


Figure 2.20: Terminal window - Sky vector generation and matrix multiplication

The vmx and dmX matrices generation requires several minutes depending also by the complexity of the geometry and the number of windows in the scene. Once the matrices are crated it is not necessary to regenerate them at each time-step and for the next simulations of the same Radiance scene. At each time-step will be created the sky vector and performed the matrices multiplication (Figure 2.20) that require few seconds. It is very important to know that the \*.vmx and \*.dmx files generated are not overwrite in any case, therefore if you need to regenerate these files delete them manually in the folder "data".

## 2.4.1 Radiance parameters

The Radiance parameters that define the quality of the simulation and the results accuracy can not be modified yet. At the state of art the simulation is performed with fixed parameters, that has been chosen according to the Three-Phase Method tutorial. The default values for the view matrix calculation are:

`-l+ -ab 12 -ad 50000 -lw 2e-5`

where [7]:

- l, boolean switch to compute irradiance rather than radiance

- ab, ambient bounce, is the maximum number of diffuse bounces computed by the indirect calculation
- ad, ambient division, the error in the Monte Carlo calculation of indirect illuminance will be inversely proportional to the square root of this number
- lw, limit the weight of each ray to a minimum

The default values for the daylighting matrix calculation are:

-c 1000 -e MF:4

where [5]:

- c is the number of sample rays produced per Klems division
- -e allows to change the sky division. In this case MF:4 - 2305 divisions

In order to change these values the user has to modify them in the C++ code and repeat the dll compilation procedure.

---

## Bibliography

---

- [1] TRNSYS, *URL: <http://www.trnsys.com/>*.
- [2] G. Ward, J. *The RADIANCE Lighting Simulation and Rendering System*, 21st Annu. Conf. Comput. Graph. Interact. Tech., pp. 459-472, 1989.
- [3] M. Saxena, G. Ward, T. Perry, L. Heschong, and R. Higa, *DYNAMIC RADIANCE - PREDICTING ANNUAL DAYLIGHTING WITH VARIABLE FENESTRATION OPTICS USING BSDF*, Fourth National Conference of IBPSA-USA, pp. 402-409, 2010.
- [4] G. Ward, R. Mistrick, E. S. Lee, A. McNeil, and J. C. Jonsson, *Simulating the Daylight Performance of Complex Fenestration Systems Using Bidirectional Scattering Distribution Functions within Radiance*, Leukos, vol. 7, pp. 241-261, 2011.
- [5] A. McNeil, *The Three-Phase Method for Simulating Complex Fenestration with Radiance*, LBNL, 2014.
- [6] G. De Michele, U. Filippi Oberegger, L. Baglivo, *Coupling dynamic energy and daylighting simulations for complex fenestration systems*, BSA 2015 2nd IBPSA-Italy Conference, Bolzano-Bozen, pp. 289-296, 2015.
- [7] A. McNeil, G. Antonutto, *Radiance premier*, *URL: <http://www.radiance-online.org/learning/tutorials/radiance-primer.pdf>*.