CODE

Optical Spectrum Simulation for Coating Design

by Wolfgang Theiss

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: 11/28/2019, 12:24 AM in Aachen, Germany
# Table of Contents

**Part I** Overview

1. About this document ........................................................................................................... 2
2. Introduction .......................................................................................................................... 2
3. About the author .................................................................................................................. 3

**Part II** User guide

1. Introduction .......................................................................................................................... 5
2. Integral quantities .................................................................................................................. 5
   - The list of integral quantities ............................................................................................... 5
   - Color coordinates ............................................................................................................... 8
   - Virtual office tower ............................................................................................................ 13
   - Other quantities ................................................................................................................ 15
   - Functions of integrals quantities ...................................................................................... 16
3. Integral quantities in main window views .......................................................................... 17
4. Other differences to SCOUT .............................................................................................. 21
5. Presentations ........................................................................................................................ 23
   - Demo presentation ............................................................................................................. 23
   - Generating presentations .................................................................................................. 24
   - Navigation ......................................................................................................................... 25
6. OLE automation .................................................................................................................. 26
   - Introduction ....................................................................................................................... 26
   - Methods ............................................................................................................................ 26
     - acquire ............................................................................................................................. 26
     - add_layer_definition_on_top(stack_index : integer; definition : string) .............. 26
     - add_layer_on_top(stack_index : integer; layer_type : string; thickness : real number; thickness_unit : string) .................................................................................................................. 28
     - acquire_dark ....................................................................................................................... 28
     - acquire_reference .............................................................................................................. 28
     - apply_factor_in_fit_parameter_transfers ........................................................................ 29
     - automatic_spectrum_scaling(i : index) .......................................................................... 29
     - batch_add_sample(name : string) .................................................................................. 29
     - batch_clear_samples ........................................................................................................ 29
     - batch_go .......................................................................................................................... 29
     - batch_restore_fit_results(sample : string) .................................................................... 29
     - clear_fit_parameters ......................................................................................................... 29
     - clear_layer_stack(index : integer) ................................................................................ 30
     - clear_material_list ......................................................................................................... 30
     - clear_report ..................................................................................................................... 30
     - create_fit_parameter(list, object, subobject, subsubobject, parameter : integer) (copy) ................................................................................................................................. 30
     - delete_fit_parameter(index : integer) ............................................................................. 32
     - df_load_from_file ............................................................................................................. 32
     - df_load_from_file_2(n: Integer; const filename, unit_string: String); ....................... 32
     - df_load_from_x_n_k_table(n:integer; const filename : string) .................................... 33
     - df_load_from_x_n_k_table_2 (n:integer; const filename : string) ............................... 33
     - df_to_clipboard (n : integer) .......................................................................................... 33
Part III  Background

1 Background information .................................................. 65
2 Color coordinates XYZ ......................................................... 65
3 Color coordinates x,y,z .......................................................... 66
4 Color coordinates L*a*b* ....................................................... 67
5 Color coordinates Lab .......................................................... 67
6 Ra (color rendering index) ..................................................... 67
7 Light transmittance ............................................................. 68
8 Light reflectance ............................................................... 69
9 Solar direct transmittance .................................................... 70
10 Solar direct reflectance ...................................................... 70

Example ................................................................................. 56
Using CODE in LabView .......................................................... 60

main_window_height................................................................. 51
main_window_left................................................................. 51
main_window_top................................................................. 51
main_window_width............................................................. 51
material_is_in_database...................................................... 51
material_is_in_material_list................................................... 51
measurement(integer) (string).................................................. 51
multidimensional_gridfit (integer)........................................... 52
number_of_dielectric_functions (integer)................................. 52
number_of_fit_parameters..................................................... 52
number_of_fp_sets (integer).................................................... 52
number_of_layers_in_stack (integer)....................................... 52
number_of_layer_stacks (integer).......................................... 52
number_of_optical_functions (integer)..................................... 52
number_of_spectra (integer).................................................... 53
number_of_tec_values (integer)............................................... 53
object_generation (real number)............................................. 53
optical_function(a_string : string) (real number)...................... 53
optical_function_by_index(i : integer) (real number)............... 53
optical_function_text_by_index (index : integer) (string)........ 53
parameter (integer) (real value)............................................ 53
rating_threshold (level_index : integer) (float)....................... 53
rating_description (level_index : integer) (string)................... 54
reference_spectrum(n : integer; formal : integer): string............. 54
refractive_index(n : integer; wavenumber : real number) (real number)................................................................. 54
sample_description: string.................................................... 54
set_spectrum_y_axis(i : integer; min, max, tick_spacing : float).... 54
simulated_spectrum_value.................................................... 54
simulated_spectrum_value_nm................................................. 54
simulation_exportfile(n : integer; formal : integer): string........ 55
spectrum_name(integer) (string)............................................. 55
spectrum_weight(index) (float).............................................. 55
tec_value................................................................. 55
tec_value_name............................................................... 55
tec_valueNickname (tec_index : integer) (string)...................... 55
tec_value_optimize (tec_index : integer) (integer).................... 56
tec_value_target_value (tec_index : integer) (integer)............... 56
tec_value_weight (tec_index : integer) (float).......................... 56
tolerance (real number)..................................................... 56
wight_formula (spectrum_index, subindex : integer) (string)....... 56
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Spectrum product</td>
</tr>
<tr>
<td>12</td>
<td>Photocurrent</td>
</tr>
<tr>
<td>13</td>
<td>( g ) (total solar energy transmittance)</td>
</tr>
<tr>
<td>14</td>
<td>Emissivity</td>
</tr>
<tr>
<td>15</td>
<td>Sheet resistance</td>
</tr>
<tr>
<td>16</td>
<td>( U ) (Thermal transmittance)</td>
</tr>
<tr>
<td>17</td>
<td>Sound insulation</td>
</tr>
<tr>
<td>18</td>
<td>NFRC support</td>
</tr>
<tr>
<td></td>
<td>NFRC VTc</td>
</tr>
<tr>
<td></td>
<td>NFRC 300-2004 solar average</td>
</tr>
<tr>
<td></td>
<td>Tdw-ISO</td>
</tr>
<tr>
<td></td>
<td>Tuv</td>
</tr>
<tr>
<td></td>
<td>NFRC U-value (winter)</td>
</tr>
<tr>
<td></td>
<td>NFRC U-value (summer)</td>
</tr>
<tr>
<td></td>
<td>NFRC SHGC</td>
</tr>
<tr>
<td>19</td>
<td>A spectrum</td>
</tr>
<tr>
<td>20</td>
<td>D65 spectrum</td>
</tr>
<tr>
<td>21</td>
<td>C spectrum</td>
</tr>
<tr>
<td>22</td>
<td>User-defined illuminants for color computation</td>
</tr>
<tr>
<td>23</td>
<td>Color-matching functions</td>
</tr>
<tr>
<td>24</td>
<td>Solar spectral distribution</td>
</tr>
<tr>
<td>25</td>
<td>AM 1.5 spectrum</td>
</tr>
<tr>
<td>Index</td>
<td>96</td>
</tr>
</tbody>
</table>
Part I
1 Overview

1.1 About this document

written by W. Theiss

W. Theiss – Hard- and Software for Optical Spectroscopy
Dr.-Bernhard-Klein-Str. 110, D-52078 Aachen, Germany
Phone: +49 241 5661390  Fax: +49 241 9529100
e-mail: theiss@mtheiss.com  web: www.mtheiss.com

January 2012

1.2 Introduction

The Coating Designer (CODE) is an application to assist you in the optical design of thin film coatings. Originally designed for glass coatings it can be used quite generally whenever optical spectra have to be optimized for a certain application (such as optical filters, solar absorbers, ...). It is based on the SCOUT spectrum simulation software and adds some functions important for thin film design. In fact, Coating Designer contains a complete SCOUT. You can directly import SCOUT configurations by setting the import filter to the SCOUT file format. This documentation is only about the additional features added to SCOUT. It is assumed that the reader is already an experienced user of SCOUT. Help on SCOUT is given in a separate document.

The CODE main window looks, except for the background picture, very much like that of SCOUT:
The introduction of the user guide gives an overview of the additional features of CODE with respect to SCOUT.

### 1.3 About the author

Responsible for all good and bad program features:

Wolfgang Theiss  
Dr.-Bernhard-Klein-Str. 110  
D-52078 Aachen  
Germany

Phone: + 49 241 5661390  
Fax: + 49 241 9529100  
e-mail: theiss@mtheiss.com  
web: www.mtheiss.com
Part II
2 User guide

2.1 Introduction

This guide shows how the additional features of CODE are applied - background information on the computed quantities are described in a later chapter.

There is one additional list of objects, the list of so-called 'integral quantities'. These quantities characterize optical properties of layer stacks and are often used as the only characterizing numbers for products. Typical examples are color coordinates or averaged transmission coefficients in certain spectral ranges.

Integral quantities can be displayed in views (the main window of CODE). Details are described in a separate section.

Several minor differences are summarized in the section 'Other differences to SCOUT' below.

You can use CODE to generate presentations of your work. A CODE presentation is a sequence of CODE configurations with some mechanisms to navigate around. Details are described here.

In order to use OLE automation of both SCOUT and CODE on one machine, the OLE server name of the two programs is different. The programs also differ in the available OLE commands: CODE supports all SCOUT OLE automation features and adds some more. The additional commands and properties of the CODE OLE server are described in a separate section.

2.2 Integral quantities

2.2.1 The list of integral quantities

The list of integral quantities looks like this:

```
<table>
<thead>
<tr>
<th>Value</th>
<th>Type</th>
<th>Spectrum</th>
<th>sigmaep</th>
<th>Optimize</th>
<th>Target value</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ra (Reflection)</td>
<td>R</td>
<td></td>
<td></td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>Ra (Transmission)</td>
<td>T</td>
<td></td>
<td></td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>Front plane diffraction</td>
<td>R</td>
<td></td>
<td></td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>4</td>
<td>Tg (EN 6073)</td>
<td>R</td>
<td></td>
<td></td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>5</td>
<td>g (EN 410)</td>
<td>R</td>
<td></td>
<td></td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>6</td>
<td>g (DIN 67507)</td>
<td>R</td>
<td></td>
<td></td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
```

Please see the technical reference of the SCOUT manual for a description of the general handling of lists like this.

As usual, you create items, i.e. rows, by selecting the desired item type in the drop down box (labeled in the situation above with 'Color coordinate L*') and then press the '+' button.

The columns of the list have the following meanings:

Value
This is the actual, computed value of the integral quantity.

Type
Here the type of the quantity is indicated. There are several color coordinates that can be computed, and various other quantities of technical interest.

**Spectrum**
The name of the spectrum that is refered to when the integral quantity is computed. Certain types of integral quantities require spectral data in well-defined spectral ranges. The spectrum should cover the used spectral range completely. The usual interpolation and extrapolation is performed if the spectral data points in the spectrum do not match the points necessary for the computation of the integral quantity.

To select the spectrum you have to press the F4 function key. This lets you cycle through all entries of the spectrum list. Stop when you have reached the appropriate spectrum. By pressing the F5 function key you can move backwards in the spectrum list.

**Sim/Exp**
After the wanted spectrum is selected you can choose between the simulated spectrum and the experimental one for the computation of the integral quantity. Although usually you may be interested in the values for the simulated spectra only, there might be situations where you want to know the corresponding values for the experimental data also.

You toggle between the simulated and measured spectrum by pressing the F4 or F5 function key.

**Optimize**
This value can be switched between the states 'Off' and 'On'. If the setting is 'Off' nothing happens. If one of the entries is set to 'On' then WCD switches its fitting strategy: Whereas usually the goal is to decrease the deviation between simulated and measured data now the integral quantity is optimized with respect to its target value (see below). The fit deviation now is the sum of the squared differences between the 'value' and the 'target value' of those integral quantities that have their optimize column entry set to 'On'. Each square difference is multiplied by its individual weight (see below).

You can toggle between 'Off' and 'On' by pressing the F4 or F5 function key.

**Target value**
Here you set the target value for the fitting of the integral quantities.

**Weight**
Here you specify the weight that multiplies the squared difference between the value and the target value of the integral quantity.

Special menu commands:

**Edit**
This command activates a sequence of dialogs which can be used to modify the properties of integral quantities. The sequence differs from object to object, but in most cases you will get the following:
Here you can change the text used to identify the integral quantity in views etc., i.e. the name of the object. Instead of the long name 'g_tot (ISO52022-3, summer) (T, Simulation)' you could simply assign the nickname 'g_tot ISO52022' to this quantity. If you leave the nickname blank the original default name will be used.

Some quantities offer several internal quantities which can be displayed as the result of the computation. You can specify the wanted quantity in the edit box 'Output internal quantity'. If you want to display the value of the integral unit together with a unit, you can enter the unit text in this dialog.

You can also set a scaling factor for the value of the integral quantity. If you want to display, for example, an averaged reflectance value (scaled between 0 and 1) in %, you should enter a scaling factor of 100 here. In addition, you should also use a proper unit (like % in this case) to indicate the scaling of the value.

Finally the dialog allows to set the number of decimals used to display the computed value.

The sequence of dialogs shown above will appear for every integral quantity. Some objects require more selections. In this case, additional dialogs will show up and allow the user to make choices. The individual dialogs are discussed below.

**Duplicate**
This command simply duplicates the selected object.

**Export data**
Select an item in the list and use this command to tell the item to export its data. This function is, for example, implemented for objects of type 'Color angle variation'. The angle dependence of \( L^* \), \( a^* \) and \( b^* \) is exported to a comma separated text file (*.csv) which can be read by Excel or similar spreadsheet programs.

**Penalty shape**
Use this command to set the penalty shape function of the selected object. Penalty shape functions replace the squared difference of simulated and target value for integral quantities which is used to measure the quality of a fit. Read more about penalty shapes in the SCOUT technical manual.
Tables
Use this command to create tables of quantities such as the applied color-matching functions or the spectral distributions of the various illuminants. The table values are written to the workbook which opens automatically.

2.2.2 Color coordinates

Color coordinates have several options which are set in the following dialog:

You access this dialog by selecting the color coordinate you want to change and activating the menu item Edit (or by clicking with the right mouse button on the color coordinate).

The following color coordinate types are implemented at present:

X,Y,Z
x,y,z
L*, a*, b*
Lab

All are based on X,Y and Z which are computed using the selected illuminant and observation angle. As observation angle you can select either 2° or 10°. For these angles color matching functions are defined.

The illuminants called A, D65 and C are available for color computations. If you need more, you can import additional illuminants from an external file, including user-defined spectra. The file must be a text file with semicolon separated columns as it can be generated by Excel. An example of the import is given in the section Background/User-defined illuminants for color computation. The additional illuminants will show up in CODE color dialogs and can be selected in order to compute color coordinates. If you store a CODE configuration after you imported additional illuminants, the imported spectra will be stored as part of the configuration.

Color view

Color view objects can be used to roughly visualize the 'color' of a spectrum. With the Edit command in the list of integral quantities the following window opens:
The window shows (after selection of an illuminating spectrum) the computed color coordinates X, Y, and Z and converts these to L*, a*, b* and RGB coordinates (Red, Green, Blue). The latter are used to display the color in the upper right half of the window. Since the conversion from X, Y, Z to RGB values is not very accurate the color view should not taken too seriously. Most PC screens are not calibrated properly anyway. So take the color impression just as a rough indication of the real color. For dark colors with low L* values the visualization is quite useless since the color appears to be black on PC screens. For such cases the bottom rectangle can be used to display a brighter version of the color: It shows a visualization of a color with the same a* and b* values, but a user-defined L* value. Use the slider position to vary L* from 0 to 100. Note, that you must use the Update command in the list of integral quantities or in the main window of CODE in order to re-compute the color coordinates after you have changed the slider.

You can leave the color view window open while you change the optical model, e.g. while changing film thicknesses with sliders: The visualization is updated whenever the model is re-computed.

Color view objects can be displayed in a view in CODE's main window (see below).

**Color angle variation**

This object can be used to inspect and optimize the variation of a coating's color with angle of incidence. For a user-defined angle range, you can set target values for L*, a* and b* For each angle, the object will compute L*, a* and b* using the actual coating. The mean square deviation between actual and target values is computed as the final number. You can set 0 as target for this deviation, and let CODE optimize parameters like film thicknesses in order to minimize the deviation, i.e. to achieve the wanted angular color variation. Be aware that for large angles the polarization of the incoming beam is important - in most cases the choice of unpolarized radiation in the corresponding spectrum object is most useful.

A view object showing a color angle variation object can display the angular dependence of L*, a* and b*. This is described below.

The Edit action for objects of this type first opens dialogs for the nickname, unit, scaling factor and decimals. Then two dialogs for graphics parameters are shown. These parameters are used by view objects that display object of type 'Color angle variation' in the main view. The first dialog sets the parameters for displaying L*, a* and b* vs. angle of incidence (set mode=2 in the view object to show this graph):
The second dialog sets parameters for a plot showing $b^*$ vs. $a^*$ (mode = 3 in the view object):
Then, in the final step, the following dialog lets you define the parameters of the computation:

In the top section the angular range is defined by setting the minimum and the maximum angle and the number of points used to scan the angle range. The example shown above means a scan of angles between 0° and 80° with a 10° resolution. If you check the option 'Count angles from backside normal'.
normal' an angle of 0° means that the radiation is incident from the backside of the sample. In this case the scan from 0° to 80° covers incidence angles in the spectrum simulation object between 180° and 100°.

In the center section you can specify the wanted color variation with angle: For L*, a* and b* you have to enter an expression for the variation of the value with angle. The angle must be referred to as 'x'. The following example shows appropriate settings for a weak increase of L* with angle and constant a* and b* values:

You must enter a formula for the target values in each edit field before you close the dialog. Otherwise you will get a warning message.

The checkboxes labeled 'Use in computation' can be used to control if this quantity (L*, a* or b*) is taken into account for the computation of the deviation. In some cases it might be appropriate to leave L* out of consideration, and optimize only a* and b*.

If some angles are more important for the color appearance of a coating than others, you can express this using an angular dependent weight function. The squared deviation between actual and target values for each angle is multiplied with the weight function at this angle.

The settings of the checkboxes under 'Show in graph' determine whether this quantity is displayed in the view graph.

**Color fluctuation**

This quantity computes the variation of color in the case of fluctuating model parameters, such as thickness values. For each fluctuating quantity the color distance to the center color in L*a*b* space is computed for the upper and lower boundary of the fluctuation. The final value is the average color distance of all variations. This very simple computational scheme is used as a compromise between speed and accuracy.

In the case of large layer stacks with many fluctuating layer thickness values the use of color fluctuation objects may lead to quite large computational times. On the other hand, the design of 'stable' coating properties is quite easy with color fluctuation objects.

**Dominant wavelength**

The so-called dominant wavelength is computed by objects with the same name. Besides the computation of this quantity these objects also display the position of the color in the x-y-plane.
which is used to compute the dominant wavelength. With **Edit** or clicking the right mouse button a new window opens showing the following graph:

![Dominant wavelength graph]

The blue line represents the position of pure spectral colors in the x-y-plane (380 to 700 nm wavelength). The black circle shows the position of 'white' whereas the blue circle shows the position of the current spectrum. If the model changes (e.g. by varying a layer thickness with a slider) the graphics is updated automatically: You can instantly see the blue circle moving through the x-y-plane. This is a very instructive tool for layer design.

**Purity**
This quantity closely related to the dominant wavelength is calculated by objects named 'Purity'. Purity objects feature the same graphics as dominant wavelength objects.

### 2.2.3 Virtual office tower

To inspect the appearance of coatings designed for architectural applications in an intuitive way you can equip a virtual office tower with your newly designed window panes. Create an object of type 'Virtual tower' and set the wanted spectrum. To be as realistic as possible this should be a reflectance spectrum using unpolarized radiation. The layer stack used for the computation of this spectrum should be the one you want to test, i.e. including all panes, air gaps and coatings ordered exactly as they will be mounted in the building. You as observer look at the building from the outside. Do not forget to set the 'sim/exp' option to 'Simulation'. Finally, use the **Edit** command to open another window where the image of the new office tower will be generated. Activate the **Build tower** command and Code starts to compute for each window of the building the angle of incidence and the corresponding reflectance spectrum. In the end the complete building is drawn:
Use the command **File|Save bitmap as ...** to store the image in a bitmap file.

If you have defined thickness fluctuations in your layer stack (see SCOUT technical manual), for every window of the building a different realization of the fluctuating parameters is used. Strong fluctuations may lead to 'designs' like the following:
2.2.4 Other quantities

In addition to color coordinates the following integral quantities may be computed:

Light transmittance
Light reflectance
Solar direct transmittance
Solar direct reflectance
Spectrum product
Photocurrent
g (total solar energy transmittance)
U (Thermal transmittance)

NFRC VTc
NFRC 300 2003 solar average
Tdw-ISO
Tuv
NFRC U-value
NFRC SHGC

Emissivity (normal and effective)
Ra (Color rendering index)
Sheet resistance
Sound insulation

2.2.5 Functions of integrals quantities

Starting with object generation 2.86, the list of integral quantities may contain objects of type 'Function of int. quant.'. These objects are used in cases where you want to compute functions of other integral quantities, such as differences, sums or ratios.

If you edit an object of type 'Function of int. quant.' you are guided through the sequence of dialogs to set the nickname, unit, scaling factor and decimals. Finally a dialog to set the function is displayed:

In this dialog you can refer to other integral quantities by the symbol iq(index). In the example, iq(1) refers to the first item of the list of integral quantities whereas iq(2) denotes the second item. You can refer to optical functions as well, using the symbol of(index). of(4), for example, refers to the 4th item of the list of optical functions.

The assignment of a spectrum to objects of type 'Function of int. quant.' is completely ignored in the computation.
Note that the order of integral quantities is important if you use functions of integral quantities: The computational routine in CODE computes the items of the integral quantity list from the top to the bottom. Functions that refer to items of this list should be placed underneath all the objects that are required for the computation. In the example above the function object may occur as the third item in the list or below.

2.3 Integral quantities in main window views

You can drag integral quantities like color coordinates from the list of integral quantities to a view list. If they carry a numerical value that value is displayed to the right of the name of the quantity. The format of the display is controlled by a mode parameter. If you select the view element and use the Edit command in the list of view elements, you are first asked to select the color for the text. Once the color dialog is closed, you are asked for the mode parameter in a dialog like this:

The name of the integral quantity (or the nickname, if you defined one) is shown, with left justification, inside the rectangle of the view element. If mode is set to 0, the numerical value is written to the right of the view object's rectangle, also left-justified. If mode equals 1, the numerical value is written right-justified with respect to the right side of the rectangle. Mode 1 is useful if you want to create tables of integral values in a view. The following example shows the difference between mode 0 and 1, with the gray rectangle indicating the width of the view objects' rectangles:

<table>
<thead>
<tr>
<th>Mode 0</th>
<th>Mode 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>L* 48.9934</td>
<td>L* 48.9934</td>
</tr>
<tr>
<td>b* -10.3311</td>
<td>b* -10.3311</td>
</tr>
<tr>
<td>a* 1.7144</td>
<td>a* 1.7144</td>
</tr>
<tr>
<td>a* 1.7144</td>
<td>a* 1.7144</td>
</tr>
<tr>
<td>b* -10.3311</td>
<td>b* -10.3311</td>
</tr>
</tbody>
</table>

The following integral quantity objects are displayed in a special way when they are dragged into a view:

**Color view**

These objects are displayed in various ways, depending on your choice. You can display a colored rectangle, the color coordinates as text, a mixture of colored rectangle and coordinates as text, the color impression as text or the position of the color in x-y-plane. The following graph gives an overview of the options on the right side:
The "color impression as text" mechanism uses the coordinates $a^*$ and $b^*$. They are converted to a color name according to the following map:
**Virtual Office Tower**

If you drag objects of type 'Virtual Office Tower' to a view be aware of the large computational work that has to be done to draw the image. The build-up of the view may take quite a long time on slow computers. Tower objects are drawn in views like this:

![Virtual Office Tower Image]

**Color angle variation**

Objects of this type are represented in views the same way as other integral quantities that produce a numerical value as output (see above, mode = 0 or mode = 1). If you set mode = 2 a graph is generated that fills the view element's rectangle:
The graphics parameters used for the plot are set in the dialog that follows the objects Edit command (see above).

The solid lines in the graph represent the variation of $L^*$ (black), $a^*$ (blue) and $b^*$ (red) with angle of incidence. The dashed lines show the corresponding target values. If you choose to display the weight function (check the appropriate checkbox in the object's dialog) it is drawn green.

Setting mode = 3 generates a $b^*$ vs. $a^*$ graph. The graphics parameters for this plot are also set in the Edit routine of the color angle variation object (see above).

You can generate several view objects that refer to the same color angle variation object like in the case below: The two graphs on the left show the same color angle variation object:
2.4 Other differences to SCOUT

In addition to the integral quantity list, there are some minor differences between CODE and SCOUT. These are summarized in this section.

Automatic loading of a start configuration
After program start CODE looks in the program directory file for a configuration named 'Start.wcd'. If it exists it is loaded automatically.

Automatic display of spectra and sliders
In versions of CODE created after 2001 the automatic display of spectra and sliders after loading a new configuration can be set as option of the configuration.

Parameter variation
Like SCOUT, CODE computes the variation of optical spectra with a parameter variation. In addition, the variation of all integral quantities is computed as well. The additional data are stored in the worksheet 'Colors'.

Parameter fluctuation
As in the case of parameter variations, CODE computes in addition to all SCOUT work the fluctuation of all integral quantities and stores the results in the worksheet 'Colors fluctuation'. No automatic graphical display is generated. In some cases it might be useful to create graphs using
objects of type 'Workbook display (rows)', e.g. in order to show color variations in a 'b* vs a*' plot. See the SCOUT manual for details about user-defined views.
Starting with object generation 2.55, we recommend the use of the new object type 'Color box' in the list of special computations (see SCOUT manual) in order to generate graphs of type 'b* vs. a*' .

Additional optical functions
The values of integral quantities can be displayed in the list of 'Optical functions' (see SCOUT technical manual). They will then appear in the workbook when you export results using the Export command. In the batch control window the values of optical functions are listed as well for every sample.
Just drag the wanted items from the list of integral quantities to the list of optical functions.

2.5 Presentations
Presentations are sequences of CODE configurations that you can use to present and organize CODE work. You can easily navigate through the configurations like you are jumping from image to image in a slide show. The difference to a slide show, however, is that each configuration is a fully functional CODE configuration with active sliders or running animations.

Presentations have been introduced to CODE with object generation 3.98 (September 2014).

2.5.1 Demo presentation
On our website you can download a demo presentation. The link is www.mtheiss.com/download/demo_presentation.zip. You will get a zip-file of about 11 MB which contains a folder 'demo presentation'. Extract the folder to your harddisc and open the presentation with the menu command 'Presentation/Load'. You will get a file dialog searching for a presentation file with extension *.cpr. Pick the file the_show.cpr in the folder of the demo presentation.

CODE will load the first page of the presentation and look like this:
The demo presentation has been taken from our interactive optics course VisualOptics. You can use this presentation to follow the instructions and explanations below.

### 2.5.2 Generating presentations

A new generation is generated by the menu command 'Presentation/New'. You are asked to input a new filename which will get the extension *.cpr (for Code PResentation). It is recommended to generate the new file in a new, empty directory - all files of the presentation will be collected in this folder. The presentation file contains a list of the configurations that deliver the content of the pages of the presentation.

Initially the presentation is empty. As soon as the presentation file has been generated you can add configurations to the presentation. With the usual command 'File/open' you can load CODE configurations. In order to add the current configuration to the presentation, use the command 'Presentation/Add configuration to presentation'. You are asked for a name of the page - enter a useful and descriptive name without file extension. Be aware, however, that the configuration is saved in the presentation folder using this name - hence you cannot use characters which are not allowed to occur in filenames.

When a new presentation is created, CODE gets a new item in its treeview (this opens with F7) called 'Presentation'. This treeview item represents the list of configurations. With a right mouse click on the treeview item you can open this list. Using the red arrows you can change the sequence of pages if you need to correct this. After a change of the page sequence you have to execute the menu command 'Presentation/Save' to save your changes.

**Automatic saving of page modifications**
If you open the presentation list in the treeview, you can check or uncheck its local menu command ‘File/Options/Automatically save configurations when presentation pages change’. If checked, configurations are saved when you jump to a new page. This is useful while you are developing the presentation since it can save a huge number of mouse clicks. On the other hand, you can easily damage a carefully made configuration by some unwanted slider movements and save this undesired state accidentally. **To be on the safe side, uncheck the option of automatic saving.**

When you exit CODE and you have generated or modified a presentation, you will be asked if you would like to save the presentation.

2.5.3 **Navigation**

**Keyboard commands**
Within a presentation you can navigate from page to page (i.e. from CODE configuration to CODE configuration) with the Page up (jumps to the new page) and Page down (jumps to the previous page) keys on your keyboard.

**Dropdown box**
In addition, CODE shows a dropdown box on the button panel which represents a list of all pages:

Selecting an item in the dropdown box means to jump to this page immediately.

If the button panel is not visible you can show it by pressing 'b' on your keyboard. Pressing 'b' again hides the button panel again.

**View elements**
There are also some view elements for navigation through presentations. First of all, you can use an item of type 'List view' and assign the presentation to it. In this case you will see a list of the pages - clicking on a line means to jump to this page. Obviously it makes sense to place such a view element on an overview page of your presentation (either at the beginning or the end). If you have picked descriptive page names it can directly be used as a table of content. An example is shown on the left side of the graph below:
If you would like to have a visual navigation help on each page of the presentation you can use a view object of type 'Presentation control' in each configuration. When you generate an object like this, CODE picks a metafile graph named 'presentation_control.emf' from its subdirectory 'Pictures' and uses this picture for the new view element. It looks like this:

There are 3 active regions that respond to mouse clicks. The left part of the graph triggers a jump to the previous page, the right section moves on the next page. A click on the center opens the first page of the presentation - in most cases this page gives an overview of all pages (see above) and can be used to quickly navigate to any page.

'Presentation control' objects are descendants of 'metafile view' elements. You can easily change the graph to a metafile image of your choice.

### 2.5.4 Publish

In order to send you presentation to someone else you have the following options.

If the receiver has CODE as well you can simply send the whole presentation folder. If the folder content is big you could pack it into a zip file and make the file available through download.

If the recipient does not have CODE you can export the presentation to a PDF file, using the command 'Presentation/Generate PDF document of whole presentation'. After you execute this command, CODE will show a file dialog for the PDF document. It will generate a new PDF file, and then load each configuration, following the sequence of the presentation. Each page of the
presentation is entered as a page in the PDF. If a configuration has more than one view it will output each view as a single PDF page. Finally the PDF is stored and you can send it to the wanted destination.

2.6 OLE automation

2.6.1 Introduction

Automatic installation
The installation of the OLE server is done by the CODE setup routine. It registers the application as OLE automation server named 'code.colors' which is after the setup procedure available to all OLE automation clients like LabView, Excel, C++, ... .

Manual registration
You can unregister and register OLE servers manually, using the Windows command line with administrator rights. In order to unregister CODE as OLE server, use the command

c:\program files\code\code.exe /unregserver

where the path to the executable file has to be adjusted, of course.

To manually register CODE please use the command

c:\program files\code\code.exe /regserver

CODE has all OLE commands and properties that our SCOUT software has, and some more in addition. This section lists all methods and properties, repeating all SCOUT items for completeness.

CODE as OLE server must be called as 'code.colors', like in the following line of a VisualBasic example:

Set wcd = CreateObject("code.colors")

The following sections describe the available OLE methods and properties. The application of OLE automation is demonstrated in an Excel VisualBasic example.

2.6.2 Methods

2.6.2.1 acquire
This command executes the global acquire command in CODE: All spectrometers are told to record new spectra.

2.6.2.2 add_layer_definition_on_top(stack_index : integer; definition : string)
A new layer is added on top of the current layer stack with index 'stack_index' in the list of layer stacks (stacks are counted 1, 2, 3, ...).

The new layer is specified by the string 'Definition'. The string may be composed by several terms, separated by tab characters. A tab character has the ASCII code 9 (In VisualBasic, for example, this is obtained by Chr(9)).

Possible layer definitions are
- "Simple layer" + tab + string specifying the material + tab + string specifying the thickness in cm(!) + tab + string specifying the partial wave superposition ("coherent" or "incoherent")
- "Thin film" + tab + string specifying the material + tab + string specifying the thickness in cm(!)
- "Thick layer" + tab + string specifying the material + tab + string specifying the thickness in cm(!)
- "Variable thickness" + tab + string specifying the material + tab + string specifying the thickness in cm(!) + tab + string specifying the thickness variation in cm (!) + tab + string specifying the thickness profile ("1" rectangular, "2" Gaussian)
- "Rough interface" + tab + string specifying the roughness formula + tab + string specifying the value of the c1 parameter + tab + string specifying the value of the c2 parameter
- 'End superlattice' (no parameter)
- 'Begin superlattice'+tab+'10' (the last partial string specifies the number of repetitions of the superlattice)
- Starting with object generation 3.81; "Anisotropic layer" + tab + string specifying the material to be used for the x-y-plane + tab + string specifying the material to be used for the z-plane + tab + string specifying the thickness in cm(!) + tab + string specifying the partial wave superposition ("coherent" or "incoherent")

VisualBasic examples:
The VisualBasic code

```
    wcd.clear_fit_parameters
    wcd.clear_layer_stack (1)
    wcd.clear_material_list
    Call wcd.add_layer_definition_on_top(1, "End superlattice")
    Call wcd.add_layer_definition_on_top(1, "Simple layer" + Chr(9) + "Si3N4" + Chr(9) + "0.0002" + Chr(9) + "coherent")
    Call wcd.add_layer_definition_on_top(1, "Thin film" + Chr(9) + "Ag (JC)" + Chr(9) + "0.000001")
    Call wcd.add_layer_definition_on_top(1, "Begin superlattice" + Chr(9) + "10")
    Call wcd.add_layer_definition_on_top(1, "Thick layer" + Chr(9) + "Si3N4" + Chr(9) + "0.2")
    Call wcd.add_layer_definition_on_top(1, "Rough interface" + Chr(9) + "c1*EXP(-X*X/C2/C2) + Chr(9) + 0.93 + Chr(9) + "45000.0")
    wcd.update_data
    wcd.update_plot
```

creates the following layer stack:
Here is an VB example for the 'Variable thickness' layer type:

```
Call wcd.add_layer_definition_on_top(1, "Variable thickness" + Chr(9) + "Si3N4" + Chr(9) + 
"0.0006" + Chr(9) + "0.00004" + Chr(9) + "2")
```

The following is an example for adding an anisotropic layer to the stack:

```
Call wcd.add_layer_definition_on_top(1, "Anisotropic layer" + Chr(9) + "material_x_y" + Chr(9) + 
"material_z" + Chr(9) + "0.000003" + Chr(9) + "coherent")
```

### 2.6.2.3 `add_layer_on_top(stack_index : integer; layer_type,material:string;thickness:real number; 
thickness_unit : string)`

A new layer is added on top of the current layer stack with index 'stack_index' in the list of layer 
stacks (stacks are counted 1, 2, 3, ...).

The type of the layer is specified by the string 'layer_type'. Possible types are
- 'Thin film' (coherent superposition of partial waves)
- 'Thick layer' (incoherent superposition of partial waves)

The material assignment is given by the 'material' parameter. If the wanted material is found in the list 
of materials, it is taken from there. If not, the database is searched for the requested material. In the 
case of success, the material is transferred from the database to the list of materials and used for 
the layer stack assignment.

The thickness of the new layer is specified by the parameters 'thickness' (a real number) and 
'thickness_unit' (a string). Possible values for the thickness unit are
- 'nm'
- 'micron'
- 'mm'
- 'm'

VisualBasic examples:

```
Call wcd.add_layer_on_top(1, "Thin film", "Si3N4", 0.45, "micron")
Call wcd.add_layer_on_top(1, "Thin film", "Ag (JC)", 19.4, "nm")
```

### 2.6.2.4 `add_to_report(text : string)`

Adds the submitted string to the CODE report window.

### 2.6.2.5 `acquire_dark`

Use this command to tell all spectrometers to record their dark spectrum.

### 2.6.2.6 `acquire_reference`

Use this command to tell all spectrometers to record their reference spectrum.
2.6.2.7 apply_factor_in_fit_parameter_transfers
Execute this command if you want to apply the fit parameter factor if you export or import fit parameter values. Since the default setting is that factors are not used, you have to apply this command before you start to transfer fit parameters and you want the factor to be taken into account.

2.6.2.8 automatic_spectrum_scaling(i : index)
Executes automatic scaling of the graphics of the i-th spectrum.

2.6.2.9 batch_add_sample (name: string)
A new sample column is added to the 'Results' worksheet of the batch control workbook. In subsequent calls you have to specify the input spectra for this sample.

VBA example:
wc.add_sample("S413 #1")

2.6.2.10 batch_clear_samples
Use this command to clear the 'Results' page of the batch control workbook, with the exception of the first column. All input spectra and results are deleted. This is used to prepare a new batch run. Using the batch_add_sample command you can add new samples afterwards.

VBA example:
wcd.batch_clear_samples

2.6.2.11 batch_go
Starts the batch fit based on the current batch control workbook settings. CODE will be busy with the batch fit activity for a while, but still respond to OLE commands. You can inspect the value of the property batch_status to find out if the batch fit operation is still active.

VBA example:
scout.batch_go

2.6.2.12 batch_restore_fit_results (sample : string)
Reloads the obtained batch fit parameters and the spectra for the specified sample.

2.6.2.13 clear_fit_parameters
This command deletes all items from the list of fit parameters.

Visual Basic example:
wcd.clear_fit_parameters

2.6.2.14 clear_layer_stack (index : integer)
Deletes all layers of a layer stack and assigns 'Vacuum' to the top and bottom halfspaces. The position of the layer stack in the list of layer stacks is given by index (stacks are numbered 1, 2, 3, ...).
2.6.2.15 clear_material_list

Clears the list of materials: All items are deleted, with the exception of 'Vacuum'.

VisualBasic example:
    wcd.clear_material_list

2.6.2.16 clear_report

Clears the CODE report window to prepare it for subsequent entries.

2.6.2.17 create_fit_parameter (list, object, subobject, subsubobject, parameter : integer) (copy)

Creates a new fit parameter in the list of fit parameters. The parameters passed to this procedure have the following meaning:

- **list**: the CODE list with the object the fit parameter of which is to be created
  1. Materials
  2. Layer stacks
  3. Spectra list
  4. Master parameter list

- **object**: the index of the object in the list. The objects are counted 1, 2, 3, ...

- **subobject**: if the object has subobjects this parameter specifies the index of the wanted subobject. This can be, for example, a susceptibility in the list of susceptibilities of a dielectric function model.

- **subsubobject**: if the subobject has again subobjects of its own, this is the index of the wanted subobject.

- **parameter**: the index of the parameter to be selected as fit parameter. If you are not sure about the index of a parameter you can drag the object to the fit parameter list and see which parameters are created in which order. The 'create_fit_parameter' OLE command uses exactly the same number and sequence of parameters.

Here is a simple example making use of the create_fit_parameter command:

'Excel VisualBasic demo of a remote fitting procedure using OLE automation

'This object is the SCOUT automation object
Public wcd As Object

Public Sub fit()
    'Create the OLE server
    Set wcd = CreateObject("code.colors")
    'Load a pre-configured configuration with a suitable reflectance spectrum and spectral range
    wcd.configuration_file = "c:\test\ole_test.wcd"
'Show SCOUT on the screen
wcd.Show

'Clear the layer stack - we will build up a new one in a moment
wcd.clear_layer_stack (1)
'Clear the list of materials - materials will be taken from the database
wcd.clear_material_list

'Create a thick layer with 2 mm thickness and fill it with BK7 glass, a material named "Glass Type BK7" must be in the database
Call wcd.add_layer_definition_on_top(1, "Thick layer" + Chr(9) + "Glass Type BK7" + Chr(9) + "0.2")

'Create a thin silver filme on top of the glass substrate, take the material "Ag model" from the database
Call wcd.add_layer_definition_on_top(1, "Thin film" + Chr(9) + "Ag model" + Chr(9) + "0.000001")

'Load a measured spectrum into the first spectrum object, using standard format
Call wcd.load_experiment(1, "C:\test\ag_10.std", 1)

'Clear the list of fitparameters
wcd.clear_fit_parameters

'From the list of layer stacks, take the first object
'and then the second subobject. This is the silver layer.
'Select parameter 1 as fit parameter (this is the thickness)
Call wcd.create_fit_parameter(2, 1, 2, 0, 1)

'From the list of materials, take the 3rd object which is the silver model
'(initially there was vacuum, then glass has been taken from the database,
'and finally, as 3rd object, the Ag model was loaded).
'Take the first susceptibility which is the Drude model.
'Take the 2nd parameter of the Drude model which is the damping constant and select
'it as fit parameter.
Call wcd.create_fit_parameter(1, 3, 1, 0, 2)

'Set boundaries for the damping constant which is fit parameter 2 now (after the layer thickness).
wcd.fit_parameter_value_min(2) = 100
wcd.fit_parameter_value_max(2) = 10000

'The silver model contains several Brendel oscillators in the list of susceptibilities.
'Take the 2nd, 3rd, 4th and 5th susceptibility and select for each one parameter
'2 as fit parameter, i.e. the oscillator strength.
'This is just for demonstration - it is not really reasonable to vary these parameters in the fit.
Call wcd.create_fit_parameter(1, 3, 2, 0, 2)
Call wcd.create_fit_parameter(1, 3, 3, 0, 2)
Call wcd.create_fit_parameter(1, 3, 4, 0, 2)
Call wcd.create_fit_parameter(1, 3, 5, 0, 2)

'Select - just for demonstration - parameter #1 of the first spectrum in the list of spectra.
'This is the angle of incidence of the reflectance spectrum.
Call wcd.create_fit_parameter(3, 1, 0, 0, 1)

'Select - also for demonstration only - the first master parameter as fitparameter.
Call wcd.create_fit_parameter(4, 1, 0, 0, 1)

'Re-compute the model and update the main view
wcd.update_data
wcd.update_plot

' Start the fit
wcd.start

' Start a waiting loop which will finish when SCOUT stops its fit (or you as a user
' press the STOP button in SCOUT.
While wcd.fitting = True
  ' Wait for 1 second
  Application.Wait Now + TimeSerial(0, 0, 1)
Wend

' Write results to a worksheet range
' fit parameter #1 is the thickness
Range("results").Offset(1, 1) = wcd.fit_parameter_value(1)
' export the real part of the refractive index of the Ag model at 1000 nm wavelength
Range("results").Offset(1, 2) = wcd.refractive_index(3, 10000000 / 1000)
' export the imaginary part of the refractive index of the Ag model at 1000 nm wavelength
Range("results").Offset(1, 3) = wcd.kappa(3, 10000000 / 1000)

End Sub

2.6.2.18 delete_fit_parameter (index : integer)

Deletes the fit parameter with the given index. Fit parameters are counted 1, 2, 3, ...

2.6.2.19 df_load_from_file

Using this command the n-th dielectric function is told to import data from a standard file specified by
the 'filename' parameter.

The n-th dielectric function should be of type 'imported dielectric function' (which is not
checked at present).

2.6.2.20 df_load_from_file_2(n: Integer; const filename, unit_string: String);

Using this command the n-th dielectric function is told to import data from a standard file specified by
the 'filename' parameter.

The n-th dielectric function should be of type 'imported dielectric function' (which is not checked at
present).

You can specify the spectral unit of the file data by setting the unit_string parameter to one of the
following:

<table>
<thead>
<tr>
<th>String transmitted</th>
<th>Spectral unit used</th>
</tr>
</thead>
<tbody>
<tr>
<td>'cm^-1' or '1/cm'</td>
<td>wavenumbers</td>
</tr>
<tr>
<td>'nm'</td>
<td>nanometers</td>
</tr>
<tr>
<td>'µm' or 'micron'</td>
<td>microns</td>
</tr>
<tr>
<td>'eV'</td>
<td>electron volts</td>
</tr>
<tr>
<td>'THz'</td>
<td>THz</td>
</tr>
</tbody>
</table>
2.6.2.21 df_load_from_x_n_k_table(n:integer; const filename : string);

Using this command the n-th dielectric function is told to import data from a testfile specified by the 'filename' parameter. The file must contain a table with three columns. The first column has to contain the spectral position, followed by n and k.

The n-th dielectric function should be of type 'imported dielectric function' (which is not checked at present).

2.6.2.22 df_load_from_x_n_k_table_2 (n:integer; const filename : string);

Using this command the n-th dielectric function is told to import data from a testfile specified by the 'filename' parameter. The file must contain a table with three columns. The first column has to contain the spectral position, followed by n and k.

The n-th dielectric function should be of type 'imported dielectric function' (which is not checked at present).

You can specify the spectral unit of the file data by setting the unit_string parameter to one of the following:

<table>
<thead>
<tr>
<th>String transmitted</th>
<th>Spectral unit used</th>
</tr>
</thead>
<tbody>
<tr>
<td>'cm^-1' or '1/cm'</td>
<td>wavenumbers</td>
</tr>
<tr>
<td>'nm'</td>
<td>nanometers</td>
</tr>
<tr>
<td>'</td>
<td>m</td>
</tr>
<tr>
<td>'eV'</td>
<td>electron volts</td>
</tr>
<tr>
<td>'THz'</td>
<td>THz</td>
</tr>
</tbody>
</table>

2.6.2.23 df_to_clipboard (n : integer)

Tells the n-th material in the list of materials to copy its graphics to the clipboard.

2.6.2.24 divide_all_measured_plate_spectra_by_100

This command tells all layer stacks to tell all layers of type 'Measured plate' to divide their spectra by 100. Use this command after you imported spectra that were given in percent.

2.6.2.25 do_no_apply_factor_in_fit_parameter_transfers

Execute this command if you do not want to apply the fit parameter factor if you export or import fit parameter values. This is the default setting.

2.6.2.26 export_measured_spectrum(n:integer; filename:string; format : integer)

Use this command to save the measured data of the n'th spectrum in the spectrum list to the file specified by filename. The format used for the datafile is given by the 'format' parameter.

The following values for the format are implemented up to now:

<table>
<thead>
<tr>
<th>format</th>
<th>File format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard</td>
</tr>
<tr>
<td>2</td>
<td>xy-format</td>
</tr>
</tbody>
</table>
2.6.2.27 export_simulated_spectrum(n:integer; filename:string; format : integer)

Use this command to save the simulated data of the n'th spectrum in the spectrum list to the file specified by filename. The format used for the datafile is given by the 'format' parameter.

The following values for the format are implemented up to now:

<table>
<thead>
<tr>
<th>format</th>
<th>File format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard</td>
</tr>
<tr>
<td>2</td>
<td>xy-format</td>
</tr>
<tr>
<td>4</td>
<td>SpectraCalc</td>
</tr>
</tbody>
</table>

2.6.2.28 grid_test

This procedure does the following: The fit parameter specified by the integer grid_test_para is varied from the value grid_test_min to the value given by grid_test_max. Grid_test_no_points are used to cover this parameter range. For each value the fit deviation is computed. After the procedure the fit parameter is set to the best value found on this grid of tried values. This feature can be used before an automatic fit to find good starting values for some of the fit parameters.

2.6.2.29 hide

Hides CODE from the screen.

2.6.2.30 hide_statusbar

Hides the status bar.

2.6.2.31 import_measured_data_via_variants(index : string; spectral_positions, spectral_data : variant; unit_string : string)

Use this command to pass experimental spectra to a CODE reflectance or transmittance spectrum.

The 'Index' parameter gives the index of the spectrum in the list of spectra (counted like 1, 2, 3, …). The spectral positions (wavelengths, frequencies, wavenumbers, energies) and data (reflectance or transmittance) are passed through variant arrays - they must be 1-dimensional and of the same length. The parameter 'unit_string' sets the spectral unit (eV, 1/cm, THz, nanometer, wavenumber).
2.6.2.32  function import_measured_plate_spectra(stack_index, layer_index : integer; t, rt, rb : string; file_format, option_t, option_rt, option_rb : integer) : integer

This function is used to load measured spectra into a layer of type 'Measured plate'. The layer is specified by the index of the stack in which it exists (counting 1, 2, 3, ...) and the layer index (also counted like 1, 2, 3, ...).

The strings t, rt and rb are used to pass the filenames for the transmittance, top side reflectance and bottom side reflectance.

File_format sets the format of the file.

The integers option_t, option_rt and option_rb define file import options that are used if you are reading from a file containing several spectra. option_t=7 means that the transmittance spectra is the seventh spectrum in the file, for example.

The return value is 0 if the spectra have been successfully loaded. Otherwise a value larger than 0 is returned.

VisualBasic call:

```
dim r as integer
r = wcd.import_measured_plate_spectra(1,2,"c:\temp\t.spc","c:\temp\r_top.spc", "c:\temp\r_bottom.spc", 4, 1, 1, 1)
if r<>0 then
    msgbox "Error importing measured plate spectra!"
endif
```

2.6.2.33  load_df(n : integer, filename : string)

Loads the n-th dielectric function from the specified file. The entries in the list of dielectric functions are numbered starting with 1.

**Warnings:**
- Be sure that the reading object and the file format are compatible.
- The content of the dielectric function object will be replaced by the data in the file.

VisualBasic example:

```
Call wcd.load_df(1, "c:/temp/df2.dfm")
```

2.6.2.34  load_experiment (n:integer, filename: string, format:integer)

Loads the file specified by filename into the n-th spectrum object, using the file format given by 'format'.

2.6.2.35  load_material_from_database (name : string)

This command loads a material from the database into the list of materials. The material is specified by the name parameter which must exactly match an item in the database.

VisualBasic example:

```
wcd.load_material_from_database ("Si3N4")
```
2.6.2.36 pane_wizard_divide_r_by_100

Divides the reflectance spectrum of the pane wizard by 100.

2.6.2.37 pane_wizard_divide_t_by_100

Divides the reflectance spectrum of the pane wizard by 100.

2.6.2.38 pane_wizard_export_results (material : string; version : float; username, comment : string)

The pane wizard analysis is done and the results are exported to the database. The filename is given by material, the file version by version. The database item is stored using the username and comment specified by the username and comment parameters.

2.6.2.39 pane_wizard_load_r (filename:string,file_format:integer,spectral_unit:string)

Loads a spectrum into the reflectance field of the pane wizard. The data are read from the file given by filename, using the file format specified with file_format. The spectral unit of the data must be given in spectral_unit.

2.6.2.40 pane_wizard_load_t (filename:string,file_format:integer,spectral_unit:string)

Loads a spectrum into the transmittance field of the pane wizard. The data are read from the file given by filename, using the file format specified with file_format. The spectral unit of the data must be given in spectral_unit.

2.6.2.41 pane_wizard_set_thickness (thickness : float)

Use this command to pass the thickness of the pane to the pane wizard. The thickness must be specified in micrometers.

2.6.2.42 pane_wizard_spectral_range (min, max : float; points : integer;spectral_unit : string)

Sets the spectral range of the pane wizard data analysis. The spectral unit is given by spectral_unit. The number of data points is specified by points, whereas the minimum and the maximum of the spectral range is set by min and max, respectively.

2.6.2.43 rename_material (old_name, new_name : string)

Renames the material called old_name to new_name.

2.6.2.44 report_to_clipboard

Copies the content of the CODE report window to the clipboard.

2.6.2.45 report_to_file(filename:string)

Saves the content of the CODE report window to a text file the name of which is the string passed to the method.
2.6.2.46 prepare_shutdown
   This method performs internal clean-up actions which are required before you shutdown the OLE server.

   There are 2 important rules applying this command:

   **Do never call this command when you want to continue to use the OLE server!**

   **Do call this command before you shutdown the OLE server!**

2.6.2.47 print_fit(i : integer)
   Tells CODE to print the graphics of spectrum i in the list of spectra. The purpose, of course, is to get information on the fit quality.

   VBA example:

   \texttt{scout.print fit(2)}

2.6.2.48 save_configuration(filename : string)
   Saves the complete CODE configuration to the specified file.

   **Warning: An already existing file it will be overwritten without warning.**

   VisualBasic example:

   \texttt{wcd.save_configuration("c:/temp/fit_sample1.wcd")}

2.6.2.49 save_df(n : integer, filename : string)
   Saves the n-th dielectric function to the specified file. The entries in the list of dielectric functions are numbered starting with 1.

   **Warning: An already existing file it will be overwritten without warning.**

   VisualBasic example:

   Call \texttt{wcd.save_df(2, "c:/temp/df2.dfm")}

2.6.2.50 scout_normal
   Show the program window like a normal window which can be hidden by other windows.

2.6.2.51 scout_on_top
   Show the program window on top of other windows on the screen.
2.6.2.52 searchfiles(dir,pattern : string)

Executes a dialog to select multiple files that can be used afterwards for CODE actions. In 'dir' you can provide the initial path for the search, 'pattern' specifies a filename search pattern.

The selected filenames can be used afterwards using the functions 'foundfiles' and 'found_filename'.

2.6.2.53 send_message(the_text : string)

Use this command to send a message to SCOUT. SCOUT will respond to the message with an action corresponding to the message.

Up to now this feature is not used.

2.6.2.54 set_bottom_halfspace_material (i : integer; name : string)

Assigns the material called name to the bottom halfspace of the i-th layer stack. Stacks are counted 1, 2, 3, ...

A material called name must be in the list of materials. If it is not you can use the command load_material_from_database(name) in order to load the material from the database into the list of materials.

2.6.2.55 set_global_range(min, max : real number, no_points : integer, spectral_unit : string))

Use this command to change the range of all spectral objects such as dielectric functions and computed spectra in one step. CODE reacts as if you had executed the Range command in the main window.

The new spectral range is defined by the minimum value min, the maximum value max, the number of data points no_points and the spectral unit. The spectral unit must be specified as one of the following strings:

<table>
<thead>
<tr>
<th>String transmitted</th>
<th>Spectral unit used by CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'cm^-1' or '1/cm'</td>
<td>wavenumbers</td>
</tr>
<tr>
<td>'nm'</td>
<td>nanometers</td>
</tr>
<tr>
<td>'µm' or 'micron'</td>
<td>microns</td>
</tr>
<tr>
<td>'eV'</td>
<td>electron volts</td>
</tr>
<tr>
<td>'THz'</td>
<td>THz</td>
</tr>
</tbody>
</table>

2.6.2.56 set_spectrum_simulation_range(n : integer; min, mix : real number; no_points : integer; spectral_unit : string)

Use this command to change the range of one of the simulated spectra. CODE reacts as if you had executed the Range command in the window of the n-th spectrum. Spectra are counted starting with 1, i.e. 1, 2, 3, ...

The new spectral range is defined by the minimum value min, the maximum value max, the number of data points no_points and the spectral unit. The spectral unit must be specified as one of the following strings:

<table>
<thead>
<tr>
<th>String transmitted</th>
<th>Spectral unit used by CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'cm^-1' or '1/cm'</td>
<td>wavenumbers</td>
</tr>
<tr>
<td>'nm'</td>
<td>nanometers</td>
</tr>
<tr>
<td>'µm' or 'micron'</td>
<td>microns</td>
</tr>
<tr>
<td>'eV'</td>
<td>electron volts</td>
</tr>
<tr>
<td>'THz'</td>
<td>THz</td>
</tr>
</tbody>
</table>
'cm^-1' or '1/cm' > wavenumbers
'nm' > nanometers
'|m|m' or 'micron' > microns
'eV' > electron volts
'THz' > THz

2.6.2.57 set_top_halfspace_material (i : integer; name : string)
Assigns the material called name to the top halfspace of the i-th layer stack. Stacks are counted 1, 2, 3, ...
A material called name must be in the list of materials. If it is not you can use the command load_material_from_database(name) in order to load the material from the database into the list of materials.

2.6.2.58 show
Displays CODE on the screen.

2.6.2.59 show_maximized
Shows the main window maximized on the screen.

2.6.2.60 show_minimized
Minimizes the main window on the screen.

2.6.2.61 show_normal
Shows the window in its normal position and size, i.e. not maximized and not minimized.

2.6.2.62 show_statusbar
Shows the status bar.

2.6.2.63 smooth_data
Smoothing of the experimental data using 3-point-averaging.

Parameters:
  n_spec: number of the spectrum in the spectrum list
  wn_min: wavenumber minimum of the spectral range
  wn_max: wavenumber maximum of the spectral range
  nt: number of smoothing iterations

2.6.2.64 smooth_data_like_batch_control
Smooth the experimental data using the settings of the batch control window for each spectrum.
2.6.2.65 **spectrum_to_clipboard (n:integer)**  
Tells the n-th spectrum object to copy its graphics to the clipboard.

2.6.2.66 **start**  
Starts the parameter fit using the current configuration.

2.6.2.67 **stop**  
Stops the parameter fit.

2.6.2.68 **update_data**  
Forces the re-computation of the CODE configuration. This command should be used after parameter changes to recalculate the optical model.

2.6.2.69 **update_plot**  
Forces the re-drawing of all visible windows of CODE, including the main window displaying the current view.

2.6.2.70 **view_to_clipboard_wmf**  
Sends the current view as metafile (vector graphics) to the clipboard.

2.6.2.71 **view_to_clipboard_bmp**  
Sends the current view as bitmap to the clipboard.

2.6.3 **Properties**

2.6.3.1 **a_filename (string)**  
Executes the standard dialog to select a filename. Returns the selected name. VBA example:

```
Range("results").Offset(4 + i, 1) = wcd.a_filename;
```

Writing this property has no effect.

2.6.3.2 **absorption_coefficient ...**  
Returns the absorption coefficient of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property. The unit of the absorption coefficient is 1/cm.

Writing this property has no effect.

2.6.3.3 **angle_distribution_no_points (spectrum_index:integer) (integer)**  
Returns or sets the number of angles of the angle of incidence distribution of the spectrum which is specified by the spectrum_index. Spectra are counted 1, 2, 3, ...

If the number of angles is set to a new value, nothing happens if the new number of angles agrees to the current number of angles. If the new number is larger than the current one, angles are added at
the end of the list. If the new number is smaller, angles are deleted from the end of the list until the
desired number of angles is reached.

VBA examples:

wcd.angle_distribution_no_points(1) = 5

MsgBox wcd.angle_distribution_no_points(1)

2.6.3.4 angle_distribution_angle (spectrum_index, angle_index : integer) (float)

Returns or sets an angle of the angle of incidence distribution of the spectrum which is specified by
the spectrum_index. Spectra are counted 1, 2, 3, ...
The angle which is returned or set is specified by the angle_index. Angles in the distribution are
counted 1, 2, 3, ...

VBA code examples:

    wcd.angle_distribution_angle(1, 1) = 15
    wcd.angle_distribution_angle(1, 2) = 25
    wcd.angle_distribution_angle(1, 3) = 35
    wcd.angle_distribution_angle(1, 4) = 45
    wcd.angle_distribution_angle(1, 5) = 55

    MsgBox wcd.angle_distribution_angle(1, 2)

2.6.3.5 angle_distribution_weight (spectrum_index, angle_index : integer) (float)

Returns or sets a weight of the angle of incidence distribution of the spectrum which is specified by
the spectrum_index. Spectra are counted 1, 2, 3, ...
The weight which is returned or set is specified by the angle_index. Angles in the distribution are
counted 1, 2, 3, ...

Please take into account that there is no automatic normalization of the weights of a distribution. It is
your responsibility to properly normalize angular weights.

VBA code examples:

    wcd.angle_distribution_weight(1, 1) = 0.1
    wcd.angle_distribution_weight(1, 2) = 0.3
    wcd.angle_distribution_weight(1, 3) = 0.25
    wcd.angle_distribution_weight(1, 4) = 0.15
    wcd.angle_distribution_weight(1, 5) = 0.2

    MsgBox wcd.angle_distribution_weight(1, 2)
2.6.3.6 **average_time** *(real number)*

Reading this number you obtain the average time (in seconds) used for one iteration of the fitting procedure.

VBA example:

```vba
Range("results").Offset(4 + i, 1) = scout.average_time;
```

Writing this property has no effect.

2.6.3.7 **batch_configuration_file** *(string)*

Setting the `batch_configuration_file` property you can load the content of the batch operation worksheets. You have to pass the name of an existing file (including the extension *.xls*) as shown in the following VBA example:

```vba
wcd.batch_configuration_file = "f:\examples\batch_parameters.xls"
```

This statement causes the loading of the specified batch configuration file.

2.6.3.8 **batch_fit_deviation** *(sample:string) *(float)*

After the performance of a batch fit you can use this command to get the obtained fit deviation for the specified sample.

VBA example:

```vba
quality = wcd.batch_fit_deviation("s413 #3")
```

2.6.3.9 **batch_fitparameter_name** *(index: integer)* *(string)*

Returns as a string the name of the batch fit parameter specified by the index parameter.

VBA example:

```vba
this_name = wcd.batch_fitparameter_name(5)
```

2.6.3.10 **batch_fitparameter_value** *(sample: string, index: integer) *(float)*

Returns as floating point number the value of the batch fit parameter specified by index, obtained in the batch fit for the sample specified by the sample parameter.

VBA example:

```vba
this_value = wcd.batch_fitparameter_value("s413 #3", 5)
```

2.6.3.11 **batch_input_path** *(index : integer) *(string)*

This property is the network path to the input spectra of the spectrum specified by the index parameter. The filenames of the input files are composed from the `batch_input_path`, the `batch_spectrum_filename` and the extension defined in the 'Input options' worksheet of the batch control workbook.

You can read or write the `batch_input_path`.

VBA example:
2.6.3.12 **batch_name_of_spectrum** (i : integer)  (string)

Returns (as string) the name of the i-th spectrum in the batch control workbook. The spectra are labeled 1, 2,..., n if there are n spectra for each sample.

VBA example:
```
my_spectrum_name = wcd.batch_name_of_spectrum(2)
```

Writing this property has no effect.

2.6.3.13 **batch_number_of_spectra**  (integer)

Returns the number of spectra defined in the current batch control workbook. Use this number to iterate through all spectra in a loop.

VBA example:
```
nbs = wcd.batch_number_of_spectra
```

Writing this property has no effect.

2.6.3.14 **batch_prefit_result** (sample : string)  (string)

Returns the pre-fit set found for the specified sample during the batch fit.

VBA example:
```
start_set = wcd.batch_prefit_result("s413 #3")
```

2.6.3.15 **batch_sample_parameter** (sample : string)  (float)

Use this property to set or read the numeric parameter of a sample in batch control workbooks. Specify the sample by its name, and pass the numerical parameter as float.

VBA example:
```
wcd.batch_sample_parameter("s413 #3") = 13.56
```

2.6.3.16 **batch_spectrum_filename** (sample : string, index : integer)  (string)

Specifies the input spectrum for the spectrum specified by the index parameter and the sample specified by the sample parameter. You can read or write this item. The input spectra are labeled 1, 2, ... n if there are n spectra for each sample.

VBA example:
```
wcd.batch_spectrum_filename("s413 #3",1) = "c:\test\s413_3_r"
```

2.6.3.17 **batch_status**  (integer)

Inspect this value to find out if CODE is busy with a batch fit operation.

- If the value of batch_status is 1 the batch fit is still active.
- If the value is 0 the batch fit is finished and you can pick up the results.
VBA example:
This example starts the batch fit and then waits in a loop until the batch fit is done. Every 3 seconds the batch status is checked. Finally a message box indicates the termination of the batch fit.

```vba
wcd.batch_go
While wcd.batch_status <> 0
    Application.Wait Now + TimeSerial(0, 0, 3)
Wend
msgbox "Batch finished"
```

### 2.6.3.18 \texttt{color\_coordinate(spectrum\_index, simulated, color\_type, illuminant, observation\_angle)} (float)

Computes the color coordinate of a spectrum.

The index of the spectrum in the list of spectra is given by the parameter 'spectrum\_index'. The spectra are counted 1, 2, 3, ...

Depending on the parameter 'simulated' the color coordinate of the simulated or the measured spectrum is computed:

- simulated = 0: The color coordinate of the measured spectrum is computed
- simulated = 1: The color coordinate of the simulated spectrum is computed

The 'color\_type' parameter selects the color coordinate to be computed. The following choices are possible:

- 0: L*
- 1: a*
- 2: b*
- 3: X
- 4: Y
- 5: Z
- 6: x
- 7: y
- 8: z
- 9: L
- 10: a
- 11: b

'Illuminant' selects one of the following illumination spectra:

- 0: A
- 1: D65
- 2: C

With 'observation angle' you can set the observation angle for which the color coordinate is computed:

- 0: 2°
- 1: 10°

Example:
Calling \texttt{wcd.color\_coordinate(1, 1, 1, 0, 1)} returns a* of the first simulated spectrum in the list of spectra. It is computed for illuminant A and 10° observation angle.
2.6.3.19 compute_emissivity (spectrum, index : integer) (integer)

Returns or sets the 'emissivity computation flag' of the interface given by index in the spectrum object given by spectrum. A value of 1 indicates that the emissivity is to be computed, a value of 0 means the assigned fixed value is used (no computation is done).

Please note:
Spectra are counted 1, 2, 3, ...

Emissivity positions are counted according to the following scheme:
0 (external surface of the glazing)
1 (surface of the first pane pointing to the interior)
2 (surface of the second pane pointing to the exterior)
3 (surface of the second pane pointing to the interior)

2.6.3.20 configuration_file

Setting the configuration_file property you specify the optical analysis method that is stored in a complete configuration file. You have to pass the name of an existing file as shown in the following VBA example:

```
wcdb.configuration_file = "f:\examples\test.wcd"
```

This statement causes the loading of the specified configuration file. Reading this property returns the name of the configuration file that was loaded by CODE in the last 'load configuration' action.

2.6.3.21 database

Sets or returns the current database path.

2.6.3.22 current_rating (string)

Returns the current rating as string.

2.6.3.23 delete_layer(stack_index,layer_index) (integer)

Deletes the layer with index layer_index in the stack specified by stack_index. Returns 0 if successful and a number <>0 if a problem has occured.

2.6.3.24 df_real_part(n : integer; wavenumber : real number) (real number)

Returns the value of the real part of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

2.6.3.25 df_real_part_nm

Returns the value of the real part of the n-th dielectric function in the dielectric function list at the wavelength specified by the 'wavelength' property (in nm).
2.6.3.26 df_imag_part\(n : \text{integer}; \ wavenumber : \text{real number}\) (real number)

Returns the value of the imaginary part of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

2.6.3.27 df_imag_part_nm

Returns the value of the imaginary part of the n-th dielectric function in the dielectric function list at the wavelength specified by the 'wavelength' property (in nm).

2.6.3.28 dielectric_function_name(i:\text{integer}) \ (\text{string})

Returns the name of the i-th dielectric function. The following VBA command fills a certain cell with the name of the 5th dielectric function:

\[\text{Range("results").Offset(5, 1) = wcd.dielectric_function_name(5)}\]

Setting this property has no effect.

2.6.3.29 emissivity (spectrum, index : integer) \ (float)

Returns or sets the emissivity of the interface given by index in the spectrum object given by spectrum.

Please note:
Spectra are counted 1, 2, 3, ...

Emissivity positions are counted according to the following scheme:
0 (external surface of the glazing)
1 (surface of the first pane pointing to the interior)
2 (surface of the second pane pointing to the exterior)
3 (surface of the second pane pointing to the interior)

2.6.3.30 experimental_spectrum_comment(n:\text{integer}): \text{string}

Returns the comment of the n-th measured spectrum in the spectrum list.

Writing this property has no effect.

2.6.3.31 experimental_spectrum_value

Returns the value of the experimental spectrum that belongs to the n-th spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property.
2.6.3.32 **fft_thickness**

Performs the FFT thickness analysis and returns the obtained thickness.

**Parameters:**

- `n_spec`: number of the spectrum in the spectrum list
- `wn_min`: wavenumber minimum of the used spectral range
- `wn_max`: wavenumber maximum of the used spectral range
- `fft_np`: number of data points used to sample the spectral range
- `d_min`: expected thickness minimum
- `d_max`: expected thickness maximum
- `n`: refractive index of the layer

Visual Basic example: `wcd.fft_thickness(1,650,7000,2048,5,50,2.0)`

2.6.3.33 **fft_strength**

Returns the height of the peak determined in the last `fft_thickness` routine for spectrum `n_spec`. This number can be used to get information on the quality of the analysis.

2.6.3.34 **file_format** (integer)

Sets the file format for subsequent import actions.

The following values are implemented up to now:

<table>
<thead>
<tr>
<th>Format</th>
<th>File format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard</td>
</tr>
<tr>
<td>2</td>
<td>xy-format</td>
</tr>
<tr>
<td>3</td>
<td>JCAMP</td>
</tr>
<tr>
<td>4</td>
<td>SpectraCalc</td>
</tr>
<tr>
<td>5</td>
<td>Perkin-Elmer</td>
</tr>
<tr>
<td>6</td>
<td>Zeiss</td>
</tr>
<tr>
<td>7</td>
<td>Opus</td>
</tr>
</tbody>
</table>

2.6.3.35 **file_object_generation** (real number)

Returns the object generation of the last configuration file that has been loaded into CODE.

2.6.3.36 **fit_deviation** (real number)

Returns the current fit deviation.

```
Dim this_deviation as single
this_deviation = wcd.fit_deviation
```

Setting this property has no effect.

2.6.3.37 **fit_parameter_name**

Returns the name of the n-th fit parameter. The following VBA command fills a certain Excel cell with the name of the selected fit parameter:

```
Range("results").Offset(4 + i, 1) = wcd.fit_parameter_name(i)
```
2.6.3.38 fit_parameter_mode
Sets or reads the variation mode of the n-the fit parameter. The following values are implemented at present:

<table>
<thead>
<tr>
<th>mode</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>Frozen</td>
</tr>
<tr>
<td>0</td>
<td>Downhill simplex</td>
</tr>
<tr>
<td>1</td>
<td>Computed from master parameters by formula</td>
</tr>
<tr>
<td>2</td>
<td>Fit on a grid with 10 points</td>
</tr>
<tr>
<td>3</td>
<td>Fit on a grid with 20 points</td>
</tr>
<tr>
<td>4</td>
<td>Fit on a grid with 50 points</td>
</tr>
<tr>
<td>5</td>
<td>Fit on a grid with 100 points</td>
</tr>
</tbody>
</table>

2.6.3.39 fit_parameter_set
Setting the fit_parameter_set property you specify the file from which a fit parameter set is to be loaded. Reading this property returns the name of the currently loaded fit parameter set.

VBA example:
```
wcdf.fit_parameter_set = "f:\examples\thickness_fit.fps"
```

2.6.3.40 fit_parameter_value
Returns the present value of the fit parameter n. The fit parameters are labeled starting at 1.

```
Dim my_value as single
my_value = wcdf.fit_parameter_value(12)
```

Set this property to pass a value to a fit parameter in CODE:

```
Dim my_value as single
my_value = 3.4
wcdf.fit_parameter_value(12) = my_value
```

2.6.3.41 fit_parameter_value_max
Sets the upper boundary (maximum) of the range of variation of the n-th fit parameter. Reading this quantity returns the current value of the upper boundary.

2.6.3.42 fit_parameter_value_min
Sets the lower boundary (minimum) of the range of variation of the n-th fit parameter. Reading this quantity returns the current value of the lower boundary.

2.6.3.43 fitting
Returns information on the fit activity of CODE.

The value of fitting is 0 if there is no fit running, fitting is equal to 1 during the fit.

VBA example:
```
if wcdf.fitting = 1 then ...
```
2.6.3.44 foundfiles (integer)

Returns the number of files that were selected in the last 'searchfiles' action.

This number can be used to iterate through all the found filenames. The latter can be obtained by the found_filename function.

2.6.3.45 found_filename(i:integer) (string)

This function returns the complete filename of the i-th selected file in the last searchfile action.

2.6.3.46 get_sim_val1(n : integer; wavenumber : real number) (real number)

Returns the 1st simulated value of the n-th simulated spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property. If the spectrum is an ellipsometry spectrum, there are two simulated quantities: Psi and delta, or cos(psi) and tan(delta). In that case the 1st value is the psi-related quantity. The second quantity, related to delta, can be obtained using the function get_sim_val2.

Writing this property has no effect.

2.6.3.47 get_sim_val2(n : integer; wavenumber : real number) (real number)

Returns the 2nd simulated value of the n-th simulated spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property. This is used in cases like ellipsometry where there are two simulated quantities: Psi and delta, or cos(psi) and tan(delta). In that case the 1st value (obtained by the function get_sim_val1) is the psi-related quantity. The second quantity is the one related to delta.

Writing this property has no effect.

2.6.3.48 grid_test_max (single)

This property sets or gets the minimum value of a parameter variation on a fixed grid.

2.6.3.49 grid_test_min (single)

This property sets or gets the minimum value of a parameter variation on a fixed grid.

2.6.3.50 grid_test_no_points (integer)

This property sets the number of points used in the parameter variation on a fixed grid.

2.6.3.51 grid_test_para (integer)

This property sets the fit parameter varied in a parameter variation on a fixed grid.
2.6.3.52 import_measured_plate_spectra

Loads measured spectra into a layer of type 'Plate measured'.

You have to specify the following parameters:

stack_index: Index of the layer stack (integer)
layer_index: Index of the layer in the stack (integer)
T : filename (including path) of the transmittance spectrum
R-Top: filename (including path) of the reflectance spectrum (top side)
R-Bottom: filename (including path) of the reflectance spectrum (bottom side)
file_format: File format as integer number (see file format specification)
option_t : file option for transmittance data
option_r_top : file option for reflectance data (top side)
option_r_bottom : file option for reflectance data (bottom side)

If you read the spectra from individual files which contain one spectrum only, all file options are 1.
If you read all spectra from a single file which contains all required data, you have to use the same
filename for T, R-Top and R-Bottom and specify different values for option_t, option_r_top and
option_r_bottom, depending of the sequence of the data in the file.

2.6.3.53 import_glass_coating_spectra

Loads measured spectra into a layer of type 'Glass coating'.

You have to specify the following parameters:

stack_index: Index of the layer stack (integer)
layer_index: Index of the layer in the stack (integer)
T : filename (including path) of the transmittance spectrum
R-air: filename (including path) of the reflectance spectrum (air side)
R-glass: filename (including path) of the reflectance spectrum (glass side)
file_format: File format as integer number (see file format specification)
option_t : file option for transmittance data
option_r_air : file option for reflectance data (air side)
option_r_glass : file option for reflectance data (glass side)

If you read the spectra from individual files which contain one spectrum only, all file options are 1.
If you read all spectra from a single file which contains all required data, you have to use the same
filename for T, R-Top and R-Bottom and specify different values for option_t, option_r_top and
option_r_bottom, depending of the sequence of the data in the file.

2.6.3.54 incidence_angle

Returns the angle of incidence of the n-th spectrum. The following VBA command fills a certain
Excel cell with the angle of incidence of the 2nd spectrum:

```
Range("results").Offset(2, 1) = wcd.incidence_angle(2)
```

Setting this property sets the value of the incidence angle of the n-th spectrum in the spectrum list.

2.6.3.55 insert_layer_definition (stack_index, layer_index : integer, layer_definition : string) (integer)

Insert a new layer in the stack given by stack_index. The new layer is added below the one specified
by layer_index. The layer is generated using the information given in layer_definition.
Returns 0 if no problem is found. Otherwise a value <> 0 is returned.
2.6.3.56  \textbf{kappa(n : integer; wavenumber : real number) (real number)}

Returns the imaginary part of the refractive index of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

2.6.3.57  \textbf{layer_material}

This function returns the name of the material assigned to layer m in the layer stack n (stacks and layers are counted 1,2,3, ...).

2.6.3.58  \textbf{layer_thickness}

Sets the thickness of the m-th layer in the n-th layer stack to the value passed to the property. Both the layer stacks and the layers in a layer stack are counted starting at 1. Reading this property returns the current value of the layer thickness.

2.6.3.59  \textbf{layer_type(n,m : integer) (string)}

This function returns the type of layer m in the layer stack n (stacks and layers are counted 1,2,3, ...).

2.6.3.60  \textbf{main_window_height}

Height of the main window.

2.6.3.61  \textbf{main_window_left}

Screen coordinate (x) of the main window's upper left corner.

2.6.3.62  \textbf{main_window_top}

Screen coordinate (y) of the main window's upper left corner.

2.6.3.63  \textbf{main_window_width}

Width of the main window.

2.6.3.64  \textbf{material_is_in_database}

Returns 1 if material_name is an object in the current database, 0 if not.

2.6.3.65  \textbf{material_is_in_material_list}

Returns 1 if material_name is an object in the current list of material, 0 if not.

2.6.3.66  \textbf{measurement(i:integer) (string)}

Setting this string to an appropriate filename will cause CODE to import the experimental spectrum of the i-th spectrum in the spectrum list from the specified file. The expected format of the measured data must be set before using the file_format property.
2.6.3.67 multidimensional_gridfit (integer)

Returns or sets the option 'multidimensional gridfit'.

A value of 0 means 'multidimensional gridfit' is switched off, a value of 1 stands for 'multidimensional gridfit' is switched on.

VBA examples:

```vba
wcd.multidimensional_gridfit=1
```

```vba
check = wcd.multidimensional_gridfit
```

2.6.3.68 number_of_dielectric_functions (integer)

Returns the number of dielectric functions in the present CODE configuration. The dielectric functions are counted from 1 to number_of_dielectric_functions.

```vba
Dim np as integer
np = wcd.number_of_dielectric_functions
```

Setting this property has no effect.

2.6.3.69 number_of_fit_parameters

Returns the number of fit parameters used at present. This can be used to write loops with some action executed for all fit parameters. For example, you could create a table with the fit parameter names and current values.

```vba
Dim np as integer
np = wcd.number_of_fit_parameters
```

2.6.3.70 number_of_fp_sets (integer)

Geben Sie hier den Text ein.

2.6.3.71 number_of_layers_in_stack (integer)

Returns the number of layers in the n-th layer stack (stacks are counted 1,2,3, ...).

2.6.3.72 number_of_layer_stacks (integer)

Returns the number of layer stacks.

2.6.3.73 number_of_optical_functions (integer)

Returns the number of items in the list of optical functions.

```vba
Dim nof as integer
nof = wcd.number_of_optical_functions
```
2.6.3.74 number_of_spectra (integer)
Returns the number of spectra in the present configuration. The spectra are counted from 1 to number_of_spectra.

Dim np as integer
np = wcd.number_of_spectra

2.6.3.75 number_of_tec_values (integer)
This integer gives you the number of technical values in the present CODE configuration, i.e. number of rows in the list of integral quantities. Use this number to iterate through all values in a loop.

2.6.3.76 object_generation (real number)
Returns the object generation of the CODE program.

2.6.3.77 optical_function(a_string : string) (real number)
Returns the value of the optical function defined by the text passed in the string a_string. If the optical function cannot be evaluated a value of 0 is returned.

Example:
x1 = wcd.optical_function("SiN(n at 400 nm)"")

Setting this property has no effect.

2.6.3.78 optical_function_by_index(i : integer) (real number)
Returns the value of the i-th optical function in the current list of optical functions in SCOUT.

Example:
x1 = wcd.optical_function_by_index(3)

Setting this property has no effect.

2.6.3.79 optical_function_text_by_index (index : integer) (string)
Returns the text of the optical function with the given index. Optical functions are counted 1, 2, ...

2.6.3.80 parameter (i:integer) (real value)
Sets or reads the value of the i-th parameter.

Defined for future use. Until now no parameters are used in CODE.

2.6.3.81 rating_theshold (level_index:integer) (float)
Returns or sets the rating threshold for the rating level specified by level_index.
2.6.3.82 rating_description (level_index:integer) (string)

Returns or sets the string for the rating level specified by level_index.

2.6.3.83 reference_spectrum(n:integer;formal:integer):string

Setting this string to an appropriate filename will cause CODE to import the reference spectrum of the i-th spectrum in the spectrum list from the specified file. The format of the measured data must be set by the format parameter.

The following values for the format are implemented up to now:

<table>
<thead>
<tr>
<th>format</th>
<th>File format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard</td>
</tr>
<tr>
<td>2</td>
<td>xy-format</td>
</tr>
<tr>
<td>3</td>
<td>JCAMP</td>
</tr>
<tr>
<td>4</td>
<td>SpectraCalc</td>
</tr>
<tr>
<td>5</td>
<td>Perkin-Elmer</td>
</tr>
<tr>
<td>6</td>
<td>Zeiss</td>
</tr>
<tr>
<td>7</td>
<td>Opus</td>
</tr>
</tbody>
</table>

VisualBasic example:

```
wc.d.reference_spectrum(1, 4) = "c:/temp/dummy2.spc"
```

2.6.3.84 refractive_index(n : integer; wavenumber : real number) (real number)

Returns the refractive index of the n-th dielectric function in the dielectric function list at the wavenumber specified by the 'wavenumber' property.

Writing this property has no effect.

2.6.3.85 sample_description: string

Read and write the sample description which is used in some views.

2.6.3.86 set_spectrum_y_axis(i : integer; min, max, tick_spacing : float)

Use this command to set the range of the y-axis of the i-th spectrum (counted 1,2,3,...): min will be the minimum, max the maximum, and tick_spacing will be used as interval between the 'ticks'.

2.6.3.87 simulated_spectrum_value

Returns the value of the n-th simulated spectrum in the spectrum list at the wavenumber specified by the 'wavenumber' property.

2.6.3.88 simulated_spectrum_value_nm

Returns the value of the n-th simulated spectrum in the spectrum list at the wavelength specified by the 'wavelength' parameter (in nm).
2.6.3.89 **simulation_exportfile(n:integer;formal:integer):string**

Assigning a filename to `simulation_exportfile` will cause CODE to save the simulated data of the n'th spectrum in the spectrum list to the specified file. The format used for the datafile is given by the 'format' parameter.

The following values for the format are implemented up to now:

<table>
<thead>
<tr>
<th>format</th>
<th>File format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard</td>
</tr>
<tr>
<td>2</td>
<td>xy-format</td>
</tr>
<tr>
<td>4</td>
<td>SpectraCalc</td>
</tr>
</tbody>
</table>

VisualBasic example:

```plaintext
wcd.simulation_exportfile(1, 4) = "c:/temp/dummy2.spc"
```

2.6.3.90 **spectrum_name(i:integer) (string)**

Returns the name of the i th spectrum.

The following VBA command fills a certain cell with the name of the 2nd spectrum:

```plaintext
Range("results").Offset(2, 1) = wcd.spectrum_name(2)
```

Setting this property has no effect.

2.6.3.91 **spectrum_weight(index) float**

Returns the weight of the spectrum with the given index. Spectra are counted like 1, 2, 3, ...

The following VBA command fills a certain cell with the weight of the 2nd spectrum:

```plaintext
Range("results").Offset(2, 1) = wcd.spectrum_weight(2)
```

Setting this property changes the weight of the indexed spectrum.

2.6.3.92 **tec_value**

Returns the present value of the i'th integral quantity.

2.6.3.93 **tec_value_name**

Returns the name of the i'th integral quantity.

2.6.3.94 **tec_value_nickname (tec_index : integer) (string)**

Returns the nickname of the integral quantity with index tec_index. The index is counted 1, 2, 3, ...

If the integral quantity has no nickname its name is given as result of this function.
2.6.3.95 **tec_value_optimize (tec_index : integer) (integer)**

Returns or sets the status of the optimization_flag of the technical value given by tec_index. A value of 1 means optimization is on, 0 stand for optimization off.

2.6.3.96 **tec_value_target_value (tec_index : integer) (integer)**

Returns or sets the target value of the technical value given by tec_index.

2.6.3.97 **tec_value_weight (tec_index : integer) (float)**

Returns or sets the weight of the technical value given by tec_index.

2.6.3.98 **tolerance  (real number)**

Use this property to read and to set the tolerance which is used for terminating the downhill simplex fitting method.

2.6.3.99 **weight_formula (spectrum_index, subindex : integer) (string)**

Returns or sets the weight formula of the spectrum given by spectrum_index. If the spectrum object is an ellipsometry object you have to set subindex to 1 to refer to the weight_formula for psi. Using a value of 2 for subindex selects the weight formula for delta.

2.6.4 **Example**

**OLE example: Excel worksheet computes technical data for different layer thicknesses**

This example shows a rather common task: You want to investigate how some technical values of a coating depend on the individual layer thicknesses. What you need to do is to compute for each set of thickness values all the wanted quantities and collect them in appropriate tables. You can do this quite easily with Excel which provides the required table structures, a macro language to express your wishes and the ability to control CODE as OLE automation server.

Before you start the Excel part of the work you have to setup a CODE configuration containing a complete model (optical constants, definition of the layer stack, definition of the spectra that are to be computed). Select those model parameters as fit parameters which shall be modified from Excel.

In our example these are three layer thicknesses.

The Excel file containing the prepared macros (see below) looks like this (in the beginning):

<table>
<thead>
<tr>
<th>Window Coating Designer</th>
<th>Configuration: d:/d3theiss/win_coat/dw_test.wcd</th>
</tr>
</thead>
<tbody>
<tr>
<td>WCD names</td>
<td>Comments</td>
</tr>
</tbody>
</table>

The cell named configuration_file will be used by the macro that starts CODE to load the CODE configuration that we are going to work with. To start CODE and load the configuration you can use the CODE menu item **WCD|Start and load configuration** which has been added to the Excel sheet. After the configuration loading the Excel enters the names of the CODE fit parameters and the CODE technical values in the first column called **WCD names**. Now the sheet looks the following way:
Window Coating Designer

<table>
<thead>
<tr>
<th>WCD names</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer no. 1 Oxide: Layer thickness</td>
<td></td>
</tr>
<tr>
<td>Layer no. 2 Ag model: Layer thickness</td>
<td></td>
</tr>
<tr>
<td>Layer no. 3 Oxide: Layer thickness</td>
<td></td>
</tr>
<tr>
<td>Purity (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>Dominant wavelength (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>Light reflectance (D65) (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>L^* / D65 / 2° (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>a^* / D65 / 2° (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>b^* / D65 / 2° (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>Solar direct reflectance (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>Light transmittance (D65) (T, Simulation)</td>
<td></td>
</tr>
<tr>
<td>Solar direct transmittance (T, Simulation)</td>
<td></td>
</tr>
</tbody>
</table>

There are 3 fit parameters (thicknesses) and 9 technical values. In the second column you can enter comments that describe the quantities in your way. For example, you could place here internal terms for the layers or quantities of interest:

<table>
<thead>
<tr>
<th>WCD names</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer no. 1 Oxide: Layer thickness</td>
<td>Top</td>
</tr>
<tr>
<td>Layer no. 2 Ag model: Layer thickness</td>
<td>Metal</td>
</tr>
<tr>
<td>Layer no. 3 Oxide: Layer thickness</td>
<td>Bottom</td>
</tr>
<tr>
<td>Purity (R, Simulation)</td>
<td>Pur</td>
</tr>
<tr>
<td>Dominant wavelength (R, Simulation)</td>
<td>DW R</td>
</tr>
<tr>
<td>Light reflectance (D65) (R, Simulation)</td>
<td>L^* R</td>
</tr>
<tr>
<td>L^* / D65 / 2° (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>a^* / D65 / 2° (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>b^* / D65 / 2° (R, Simulation)</td>
<td></td>
</tr>
<tr>
<td>Solar direct reflectance (R, Simulation)</td>
<td>Solar front</td>
</tr>
<tr>
<td>Light transmittance (D65) (T, Simulation)</td>
<td>LT</td>
</tr>
<tr>
<td>Solar direct transmittance (T, Simulation)</td>
<td>Solar through</td>
</tr>
</tbody>
</table>

Now it's time to enter the values for the thicknesses that you want to include in your computation. Give a name (which can be just a number) for the set in the cells to the right of the term Comments and enter below the parameter values. The macros that performs the computation looks if there is a name in the third row of the worksheet. If so, it expects thickness values below that name. The macro continues to work through the columns from left to right until it cannot find a name anymore, i.e. if it reaches an empty name cell. For large series of values it would be reasonable to write a short VisualBasic macro that creates the input values for you. A prepared worksheet could be this one:
Now execute the computation macro by the menu item \textit{WCD|Compute data} and watch how Excel fills up the table:

The Visual Basic macros used in this example:

\begin{verbatim}
' The wcd object is the global object representing the Windows Coating Designer OLE server
Dim wcd As Object

' This macro creates the wcd object and loads the configuration file that is specified in the
' Excel worksheet "colors" (cell named "configuration_file")
' This configuration must have been created before for the layer stack that is going to be inspected

Sub start_wcd()
    'Create the wcd object
    Set wcd = CreateObject("code.colors")
\end{verbatim}
'Load the configuration file
wcd.configuration_file = Range("colors!configuration_file").Value

'Show the WCD program on the screen
wcd.Show

row_offset = 2

'Get the number of fit parameters (i.e. layer thicknesses) from WCD
no_paras = wcd.number_of_fit_parameters

'Get the number of integral values (e.g. color coordinates) from the WCD list of integral values
no_tec_values = wcd.number_of_tec_values

'Enter the names of the fit parameters and integral quantities to be computed into the Excel worksheet
'in column 1 (A)
For i = 1 To no_paras
    Range("colors!origin").Offset(row_offset + i, 0).Value = wcd.fit_parameter_name(i)
Next i

For i = 1 To no_tec_values
    Range("colors!origin").Offset(row_offset + no_paras + 1 + i, column_count).Value = wcd.tec_value_name(i)
Next i

End Sub

'This macro deletes the OLE server from memory
Sub delete_wcd()
    wcd.prepare_shutdown
    Set wcd = Nothing
End Sub

'This macro displays the WCD main window if it was hidden
Sub show_wcd()
    wcd.Show
End Sub

'This macro hides the WCD main window
Sub hide_wcd()
    wcd.Hide
End Sub

'This macro computes the wanted integral quantities
Sub compute_values()
    Dim fitpara_count As Integer
    Dim tecdata_count As Integer
    Dim no_paras As Integer
    Dim no_tec_values As Integer

    row_offset = 2
' Get the number of fit parameters from WCD
no_paras = wcd.number_of_fit_parameters
' Get the number of integral quantities from WCD
no_tec_values = wcd.number_of_tec_values

column_count = 2
' Check if there is another set of thicknesses to process
While Range("colors!origin").Offset(row_offset, column_count).Value <> ""
    ' If so take the values for the fit parameter from the Excel worksheet
    For i = 1 To no_paras
        If wcd.fit_parameter_mode(i) = 2 Then
            wcd.fit_parameter_value(i) = 0.001 * Range("colors!origin").Offset(row_offset + i, column_count).Value
        Else
            wcd.fit_parameter_value(i) = Range("colors!origin").Offset(row_offset + i, column_count).Value
        End If
    Next i
' Recompute everything in WCD with the new set of parameters
wcd.update_data
' Get from WCD the new values of the integral quantities
For i = 1 To no_tec_values
    Range("colors!origin").Offset(row_offset + no_paras + 1 + i, column_count).Value = wcd.tec_value(i)
Next i
' Proceed to the next column
column_count = column_count + 1
Wend

End Sub

2.6.5 Using CODE in LabView

You must have object generation 3.60 for the mechanism described below!

If you develop automation solutions with LabView you can access CODE in your LabView projects. A typical application would be a scanning spectrometer system, used to obtain one- or two-dimensional thickness profiles. If LabView is used to control the mechanical motion and the data acquisition of the spectrometer hardware, it would be advantageous to control CODE by LabView as well. After a measurement, LabView could pass the recorded spectra to CODE, start the thickness determination, and pick-up the results in order to generate appropriate operator displays.

Here is how you get access to CODE within LabView:

- Generate a control of type 'Automation refnum'
- Right-click on the control and select the command 'Select ActiveX class/Browse'
- A dialog for the selection of a type library opens
Search for the type library 'CODE library' in the dropdown list.
If you do not find it click on the 'Browse' button to the right and select the CODE program file code.exe (in order to see the program file you may have to set the file type to 'All files (*.*)' in the file dialog).
The dialog should now look like this:

Check the checkbox 'Show Creatable Objects Only' and select the only remaining line 'scoutole (scout.scoutole)'.

© 2019 Wolfgang Theiss
Then press OK to leave this dialog.
- In the LabView Block diagram, right-click the new object and select the command 'ActiveX palette' to generate a new object of type 'Automation open'. Connect the 2 objects like shown below:

- From now on you can use all OLE properties and procedures of CODE. In order to check if you can really access CODE commands generate a small test sequence: Show CODE on the screen ('show' command), execute a file selection dialog (by calling the property 'a_filename') and finally call the routine 'prepare_shutdown' to prepare the death of the CODE server.
Before you do any real work in LabView, please test if you can get this sequence to execute properly.
Part III
3 Background

3.1 Background information

This section gives background information (theory, formulas, tables) for some computed quantities.

At present you find information about

- Color coordinates XYZ
- Color coordinates xyz
- Color coordinates L*a*b*
- Color coordinates Lab
- Color-matching functions
- Ra (color rendering index)
- Light transmittance
- Light reflectance
- Solar direct transmittance
- Solar direct reflectance
- Spectrum product
- Photocurrent
- g (total solar energy transmittance)
- Emissivity
- Sheet resistance
- U (Thermal resistance)
- A spectrum
- D65 spectrum
- C spectrum
- Solar spectral distribution
- AM 1.5 spectrum

3.2 Color coordinates XYZ

The color coordinates X,Y,Z are computed according to DIN 5033.

The following steps are taken:

In case of reflectance or transmittance spectra the illuminant D65 with relative spectral distribution $S$ (as given in the table below) is used to compute the intermediate quantity $k$:

$$ k = \frac{100}{\sum_{\lambda=380\,nm}^{780\,nm} S_{\lambda} \, \bar{y}(\lambda) \Delta\lambda} $$

$\bar{y}$: Color - matching function

The tristimulus values $X$, $Y$ and $Z$ are calculated according to the following equations:
where $\tau$ is the transmittance of the layer stack. The color-matching functions are given below.

The given expressions are for 2° observation. For 10° the corresponding values for the color-matching functions are taken. These are also given below.

### 3.3 Color coordinates $x,y,z$

The chromaticity coordinates $x,y,$ and $z$ are computed from $X,Y,$ and $Z$ using the relations

\[
x = \frac{X}{X + Y + Z}
\]

\[
y = \frac{Y}{X + Y + Z}
\]

\[
z = \frac{Z}{X + Y + Z}
\]
3.4 **Color coordinates \(L^*a^*b^*\)**

The color coordinates \(L^*,a^*,b^*\) (CIELAB) are computed according to DIN 5033. They are based on \(X,Y,Z\) (see above).

The following equations are used:

\[
L^* = 116 \frac{Y}{Y_N} - 16
\]

\[
a^* = 500 \left( \frac{X}{X_N} - \frac{Y}{Y_N} \right)
\]

where \(X_N = 95.05\) \(Y_N = 100.00\) for D65 illuminant \(Z_N = 108.90\)

\[
b^* = 200 \left( \frac{Y}{Y_N} - \frac{Z}{Z_N} \right)
\]

If the arguments of the cubic roots are smaller than 0.008856 then the cubic root is replaced by the expression \(7.787 \times \text{argument} + 16/116\).

3.5 **Color coordinates \(Lab\)**

The quantities \(L, a, b\) are computed according to

\[
L = 100 \left( \frac{Y}{Y_N} \right)^{1/2}
\]

\[
a = 175 \left( \frac{0.0102 \times X_N}{Y \times Y_N} \right)^{1/2} \left[ \frac{X}{X_N} - \frac{Y}{Y_N} \right]
\]

where \(X_N = 95.05\) \(Y_N = 100.00\) for D65 illuminant \(Z_N = 108.90\)

\[
b = 70 \left( \frac{0.00874 \times Z_N}{Y \times Y_N} \right)^{1/2} \left[ \frac{Y}{Y_N} - \frac{Z}{Z_N} \right]
\]

The coordinates \(X, Y, Z\) are defined above.

3.6 **Ra (color rendering index)**

This quantity is computed according to standard EN 410. It requires the range of the spectrum to cover 380 ... 780 nm.
3.7 Light transmittance

The light transmittance is computed according to the following equation (DIN 67 504):

\[
\tau = \frac{\sum_{\lambda = 380 \text{ nm}}^{780 \text{ nm}} S_\lambda \, \tau(\lambda) \, V(\lambda) \, \Delta\lambda}{\sum_{\lambda = 380 \text{ nm}}^{780 \text{ nm}} S_\lambda \, V(\lambda) \, \Delta\lambda}
\]

- \( S_\lambda \): relative spectral distribution of illuminant
- \( \tau(\lambda) \): spectral transmittance of layer stack
- \( V(\lambda) \): spectral luminous efficiency

The values of the product \( S^*V \) of the relative spectral distribution of the light source (here: D65) and the spectral luminous efficiency are these:

![Graph showing the product \( S^*V \) vs. wavelength in nanometers.](image)

They are taken from the following table (also from DIN 67504):

<table>
<thead>
<tr>
<th>Wavelength [nm]</th>
<th>( S^*V )</th>
</tr>
</thead>
<tbody>
<tr>
<td>380</td>
<td>0.0000</td>
</tr>
<tr>
<td>390</td>
<td>0.0005</td>
</tr>
<tr>
<td>400</td>
<td>0.0030</td>
</tr>
<tr>
<td>410</td>
<td>0.0103</td>
</tr>
<tr>
<td>420</td>
<td>0.0352</td>
</tr>
<tr>
<td>430</td>
<td>0.0948</td>
</tr>
<tr>
<td>440</td>
<td>0.2274</td>
</tr>
<tr>
<td>450</td>
<td>0.4192</td>
</tr>
<tr>
<td>460</td>
<td>0.6663</td>
</tr>
<tr>
<td>470</td>
<td>0.9850</td>
</tr>
<tr>
<td>480</td>
<td>1.5189</td>
</tr>
</tbody>
</table>
For other illuminants than D65 the product $S^*V$ is divided by the intensity of the D65 light source and multiplied by the intensity of the chosen illuminant.

### 3.8 Light reflectance

The light reflectance is computed according to the following equation (DIN 67 504):

$$
\rho = \frac{\sum_{\lambda=380 \text{ nm}}^{780 \text{ nm}} S_\lambda \rho(\lambda) V(\lambda) \Delta \lambda}{\sum_{\lambda=380 \text{ nm}}^{780 \text{ nm}} S_\lambda V(\lambda) \Delta \lambda}
$$

$S_\lambda$ : relative spectral distribution of D65 illuminant

$\rho(\lambda)$ : spectral reflectance of layer stack

$V(\lambda)$ : spectral luminous efficiency
The values of the product $S^*V$ of the relative spectral distribution of the light source and the spectral luminous efficiency are the same as used for the computation of the light transmittance (see above).

### 3.9 Solar direct transmittance

The solar direct transmittance $t_e$ is computed (according to DIN 67507, EN 410 or ISO9050:2003) as

$$t_e = \frac{\sum_{\lambda=300\text{nm}}^{2500\text{nm}} S_{\lambda} \tau(\lambda) \Delta\lambda}{\sum_{\lambda=300\text{nm}}^{2500\text{nm}} \Delta\lambda}$$

$S_{\lambda}$: relative spectral distribution of solar radiation

$\tau(\lambda)$: spectral transmittance of layer stack

The set of numbers $S_{\lambda}\Delta\lambda$ (Din 67507) used for the integration is given below.

### 3.10 Solar direct reflectance

The solar direct reflectance $r_e$ is computed (according to DIN 67507, EN 410 or ISO9050:2003) as

$$r_e = \frac{\sum_{\lambda=300\text{nm}}^{2500\text{nm}} S_{\lambda} \rho(\lambda) \Delta\lambda}{\sum_{\lambda=300\text{nm}}^{2500\text{nm}} \Delta\lambda}$$

$S_{\lambda}$: relative spectral distribution of solar radiation

$\rho(\lambda)$: spectral reflectance of layer stack

The set of numbers $S_{\lambda}\Delta\lambda$ used for the integration is given below.

### 3.11 Spectrum product

This quantity is the integration over the product of the selected spectrum and a so-called 'Input spectrum'. The latter can be set in a window which is opened by the 'Edit' command of the 'Spectrum product' object. Here you can import the spectrum from a datafile.

The spectral range used for the integration is that of the input spectrum. The required values of the selected spectrum are computed by linear interpolation or extrapolation.
3.12 Photocurrent

Objects of type 'Photocurrent' compute the current density generated by absorption of light in a layer. You must assign a spectrum of type 'Charge carrier generation' to a photocurrent object to make it work correctly. Please see the 'SCOUT technical manual' for documentation of this spectrum type. When you edit a photocurrent object, you should set its unit to 'mA / cm^2'. The Edit procedure will ask you to specify the total illumination power (in W/m^2). Afterwards you have to set the spectral distribution of the illuminating radiation in the following window:

The example shows the normalized AM 1.5 spectrum. You do not have to care about normalization of the input spectrum - photocurrent objects normalize the distribution function before they use it for their computation.

Photocurrent objects compute for each spectral point the number of generated charge carriers and integrate over the whole input spectrum. The final number is the current density in mA / cm^2.

**Application example**

The following example shows the optimization of a simple solar cell layer stack. The illuminating beam is incident from the top through a 4 mm glass pane. Underneath a transparent conductive layer (TCO) of several 100 nm thickness the active layer which is to convert sunlight to electrical current is placed. A aluminum layer at the bottom of the stack provides electrical contact on the backside of the active layer.

The active layer is equipped with a fictitious material which absorbs in the visible spectral range. Its complex refractive index shows two main absorption bands around 500 and 700 nm wavelength:
The light absorption of the active layer with 150 nm thickness (TCO thickness is 500 nm) is this:

Usually not all absorbed photons generate a charge carrier. The conversion efficiency may be much less than 1 and depend on details like the thickness of the active layer. In order to take into account a thickness dependent efficiency, the efficiency of the charge carrier generation spectrum is
computed according to a user-defined formula. In this case the efficiency is simply a constant $C_1$ where $C_1$ is one of the possible fit parameters of the efficiency object.

With the help of a master parameter called 'Absorber_thickness' the coupling of layer thickness and efficiency can be expressed in CODE. The fit parameters for the optimization of the photocurrent are the following:

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Variation</th>
<th>Low limit</th>
<th>High limit</th>
<th>Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stack 1 Layer 4 Absorber Thickness</td>
<td>Absorber thickness</td>
<td>0.0</td>
<td>0.0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Charge carrier generation efficiency C1</td>
<td>0.7-Absorber_thickness/600</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>AbsorberThickness</td>
<td>Downhill simplex</td>
<td>500.0</td>
<td>500.0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Stack 1 Layer 3 TCO Layer thickness</td>
<td>Downhill simplex</td>
<td>200.0</td>
<td>500.0</td>
<td>1</td>
</tr>
</tbody>
</table>

As shown above, the thickness of the layer is simply computed as Absorber_thickness, whereas the efficiency linearly decreases with thickness following the relation $0.7$-Absorber_thickness/600.

Setting a target value of 10 mA/cm²/2 for the current density in the list of integral quantities, the fit optimizes both the TCO and the active layer thickness with the following best result:

![Graph showing the relation of absorber thickness and photocurrent with fixed TCO thickness]
3.13 g (total solar energy transmittance)

The total solar energy transmittance (or solar factor) \( g \) is computed according to DIN 67 507, EN 410, NFRC, ISO9050:2003 or ISO52022-3 (2017). CODE can handle an arbitrary number of glass panes in a glazing. A material object of type 'Gas' has to be used between two panes. In order to compute \( g \) CODE has to use some physical properties of the filling gas. The values have been implemented in the material type 'Gas' which allows to define all kinds of gas mixtures. Please use this type of material for all gas fillings.

\( g \) is the sum of the solar direct transmittance \( t_d \) and the secondary internal heat transfer factor \( q_i \). The quantity \( q_i \) as well as the corresponding quantity \( q_e \) (the secondary external heat transfer factor) are also available as integral quantities.

\( g \) can only be calculated for simulated spectra, not for measured spectra. CODE needs to compute the emissivities from parts of the layer stack. This is not possible in the case of experimental data. CODE needs to compute the emissivity of the coated glass panes in the far infrared (200 ... 1800 \( \text{1/cm} \)). Make sure that your optical model is valid in this spectra range. In most cases it is sufficient to compute the properties of the material silver that is used in the coatings in the far infrared. We recommend to use a model for the optical constants of silver and compute the model in the range 300 ... 50000 \( \text{nm} \) with at least 3000 data points. If there are other materials that might contribute to the emissivity (like ITO or thick metallic NiCr) then you should use the recommended wavelength range for these materials as well.

If you do not have optical constants in this spectral range you can still compute \( g \) if you know the emissivity of the involved surfaces. In this case, Edit the \( g \) quantity in the list of integral quantities. This dialog will open:

![Option for \( U \) computation dialog]

Note that the individual layers are not drawn in the dialog's sketch.

Using the checkboxes at the interfaces you can decide if CODE should compute the emissivity at this interface from the optical model. If the option is unchecked you can type in the emissivity to be used for this interface. If the option is checked CODE will indicate (in gray) the computed value of the emissivity.

The computation of \( g \)-values according to standards NFRC and ISO52022-3 involves an iterative routine that determines the temperature distribution of the glazing. As a consequence one can extract the temperature of the glass panes once the computation is finished. Other quantities like the
fraction of sunlight absorbed in each pane are available as well. These numbers can be extracted using optical functions or defining the wanted quantity as 'internal quantity' that is output as final result instead of the g-value.

**Internal quantities of \( g_{\text{tot}} \) (ISO52022-3 (2017))**:  
The following quantities are available as internal quantities. Make sure you correctly type their names:
- temperature pane 1
- temperature pane 2
- temperature pane 3
- temperature pane 4
- temperature pane 5
- absorbed power pane 1
- absorbed power pane 2
- absorbed power pane 3
- absorbed power pane 4
- absorbed power pane 5
- direct solar transmission
- \( u \)-value
- gain through thermal radiation
- gain through conduction and convection
- \( qi \)

### 3.14 Emissivity

The emissivity (both the normal and effective emissivity can be calculated in CODE) is computed following standard EN 673.

**Warning:** Note that the spectrum must be known in the infrared region 5500 ... 50000 nm or 200 ... 1820 \( 1/\text{cm} \). Make sure that you have reliable optical constants in this range!

### 3.15 Sheet resistance

This function returns the sheet resistance of the layer stack that is assigned to the selected spectrum.

Note the following:
The sheet resistance is computed taking into account all layers which have a non-vanishing conductivity.

You can introduce electrical conductivity to a material in the following ways:
- Use an optical constant model and introduce a Drude model or an extended Drude model. This is the most flexible and recommended way.
- Assign a value for the electrical conductivity using the menu item **Property|Electrical conductivity**.

### 3.16 U (Thermal transmittance)

This quantity is calculated according to standard EN 673. Please read the remarks about the computation rules of \( g \) above.
3.17 Sound insulation

CODE can output sound insulation properties of the layer stack that belongs to the assigned spectrum. The given values are not computed based on a formula or a computational routine. Instead, they are looked up in a list of pre-defined layer stacks for which sound insulation data are available.

Sound insulation data are usually given for glazing compositions which are denoted like this: G4/FA/FA/G4/Ar16/G4
The notation means the following: We have 4 mm glass, then a foil named FA, another foil named FA, 4 mm glass, 16 mm Argon gas and another 4 mm glass pane.

In order to search for eventually existing sound data, CODE has to reduce the layer stack definition to the notation given above. This is achieved by the following sequence of actions:

- Copy the layer stack and remove all thin films or simple layers which are treated coherently. It is assumed that the remaining layers are either glass, foil or gas.
- Foils are identified by the name of the material. All foils which may be used in glazing compositions must be entered in a list which is accessible in the list of integral quantities (menu command: ‘Sound insulation/Edit foil list’). Such a list has 3 columns: The name of the material as it shows up in the optical constant database, the nickname which is used for the foil in the ‘sound insulation notation’, and the foil thickness in microns. If a layer material and the layer thickness agree with a foil definition, then the relevant foil nickname is added to the glazing notation. If the layer thickness is a multiple of the defined foil thickness then the nickname is added several times to the glazing notation.

Now the layer stack is scanned for gas names. The assigned material names in the layer stack are compared to a list of gases. This list must contain the name of the gas used in the optical constant database (first column) and the nickname which is used in the notation of glazing compositions. It opens with the menu command ‘Sound insulation/Edit gas list’.
All layers which are not identified as foils or gas are assumed to be glass.

Once the layer stack definition has been converted to the notation for the glazing component (like G4/FA/G4), CODE checks if for this composition there are sound insulation data available. This is done by going through the list which opens with the menu command 'Sound insulation/Edit stack list':

If a composition match is found, the given RW value (or one of the other available values such as Ra, Ra,tr or C and Ctr) is taken as result. In addition, the text defined in the column 'Result string' is stored in the integral quantity. Both the value and the result text may be displayed in a view (see below).

Please be aware that the 3 lists (foils, gases, stacks) are stored as part of the configuration file. They are not available in other configurations. Please do not change the order of the columns in the stack list.
The list of integral quantities will show the result value of a sound insulation object. If you display the sound insulation object in a view you can edit the view element and select the display of the result text.

3.18 **NFRC support**

Technical data computed according to procedures defined by the US organization NFRC (National Fenestration Rating Council) have been introduced with CODE version 3.79. CODE calculates and optimizes center-of-glass values only - no frame effects are taken into account.

3.18.1 **NFRC VTc**

Averaged transmittance in the visible spectral range (called 'Photopic' in NFRC documents), computed according to the procedure described in NFRC 300-2004. Transmittance data are required in the wavelength range 380 ... 780 nm. The averaging procedure can be used for reflectance data as well.

3.18.2 **NFRC 300-2004 solar average**

Averaged spectral values (used for reflectance as well as for transmittance) computed according to the procedure described in NFRC 300-2004. The calculation is based on the spectral distribution of solar radiation in ISO 9845-1. The required wavelength range for input data is 300 ... 2500 nm.

3.18.3 **Tdw-ISO**

This quantity is a weighted UV-Vis transmission (wavelength range: 300 ... 500 nm) which tries to quantify radiation damage to human skin. Its computation is described in 'Window 4.0 - Documentation of Calculation Procedures', LBL-33943 (1993), page 29.

3.18.4 **Tuv**

This quantity is a weighted UV transmission (wavelength range: 300 ... 390 nm) which tries to quantify radiation damage to human skin. Its computation is described in 'Window 4.0 - Documentation of Calculation Procedures', LBL-33943 (1993), page 29.

3.18.5 **NFRC U-value (winter)**

The NFRC center-of-glass U-value is computed following the procedure described in the document "WINDOW 4.0: Documentation of Calculation Procedures" (July 1993). The method is based on physical principles and experimental results concerning heat transfer through a glazing system. In an iterative process the temperature distribution in the glazing is obtained, taking into account temperature dependent gas properties in between the panes. Gas properties are not taken from the document mentioned above but from the standard ISO 15099.

The expressions for temperature dependent gas properties have been implemented in objects of type 'Gas' in the list of materials (starting with version 3.79). These objects allow the definition of gas mixtures and re-compute all required gas properties at any given temperature. Make sure that you use objects of this type as filling gas in your layer stacks.

NFRC U-values in CODE are computed using the following environmental conditions (NFRC 100-2010 Winter):

Room temperature: 21°
Outside temperature: -18°
Glazing oriented windward, wind speed: 5.5 m/s
Sky temperature: -18°
Sky emissivity: 1.0
Incident solar power: 0 W/m²

Extracting additional information using optical functions
Besides the U-Value you can extract additional values that are obtained during the computation, in particular the obtained pane temperatures. You can do so using defining appropriate optical functions in the list of ‘Optical functions’. Follow the examples below:

The function 'NFRC U-value (R, simulation, temperature pane 1)' returns the temperature of pane 1 where panes are counted 1, 2, 3, ... from the exterior to the interior.
The function 'NFRC U-value (R, simulation, temperature pane 2)' returns the temperature of pane 2 and so forth up to pane 5.

3.18.6 NFRC U-value (summer)
Computed like the quantity NFRC U-value (winter) but applying different environmental conditions (NFRC 100-2010 Summer):

Room temperature: 24°
Outside temperature: 32°
Glazing oriented windward, wind speed: 2.75 m/s
Sky temperature: 32°
Sky emissivity: 1.0
Incident solar power: 0 W/m²

3.18.7 NFRC SHGC
The secondary heat gain coefficient SHGC is computed following the same procedures as the NFRC U-value. With solar power incident on the glazing, the temperature profile is computed and the total transmission of solar energy into the room is calculated. In addition to the directly transmitted radiation sun light is absorbed by the glazing and converted into heat. The fraction of heat entering the room is computed and added to the directly transmitted portion in order to obtain the final SHGC value.
Due to the temperature dependent gas properties, the computational scheme is complex enough. In order to avoid additional difficulties it is assumed that the absorption of solar radiation is evenly distributed across the depth of the panes - this is not a good approximation in the case of absorptive coatings.

Environmental conditions used in the computation of NFRC SHGC values:

Room temperature: 24°
Outside temperature: 32°
Glazing oriented windward, wind speed: 2.75 m/s
Sky temperature: 32°
Sky emissivity: 1.0
Incident solar power: 783 W/m²

Extracting additional information using optical functions
Besides the U-Value you can extract additional values that are obtained during the computation. You can do so using defining appropriate optical functions in the list of ‘Optical functions’. Follow the examples below:
The function 'NFRC SHGC (R, simulation, temperature pane 1)' returns the temperature of pane 1 where panes are counted 1, 2, 3, ... from the exterior to the interior. The function 'NFRC SHGC (R, simulation, temperature pane 2)' returns the temperature of pane 2 and so forth up to pane 5.

The function 'NFRC SHGC (R, simulation, absorbed power pane 1)' returns the absorbed solar power in W/m² of pane 1. Use similar expressions for panes 2 to 5.

### 3.19 A spectrum

The A spectrum used in some calculations is this:

Here is a table of the values:

<table>
<thead>
<tr>
<th>Wavelength[nm]</th>
<th>Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>380</td>
<td>9.8</td>
</tr>
<tr>
<td>385</td>
<td>10.9</td>
</tr>
<tr>
<td>390</td>
<td>12.1</td>
</tr>
<tr>
<td>395</td>
<td>13.4</td>
</tr>
<tr>
<td>400</td>
<td>14.7</td>
</tr>
<tr>
<td>405</td>
<td>16.1</td>
</tr>
<tr>
<td>410</td>
<td>17.7</td>
</tr>
<tr>
<td>415</td>
<td>19.3</td>
</tr>
<tr>
<td>420</td>
<td>21.0</td>
</tr>
<tr>
<td>425</td>
<td>22.8</td>
</tr>
<tr>
<td>430</td>
<td>24.7</td>
</tr>
<tr>
<td>435</td>
<td>26.6</td>
</tr>
<tr>
<td>440</td>
<td>28.7</td>
</tr>
<tr>
<td>445</td>
<td>30.9</td>
</tr>
<tr>
<td>450</td>
<td>33.1</td>
</tr>
<tr>
<td>455</td>
<td>35.4</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>460</td>
<td>37.8</td>
</tr>
<tr>
<td>465</td>
<td>40.3</td>
</tr>
<tr>
<td>470</td>
<td>42.9</td>
</tr>
<tr>
<td>475</td>
<td>45.5</td>
</tr>
<tr>
<td>480</td>
<td>48.2</td>
</tr>
<tr>
<td>485</td>
<td>51.0</td>
</tr>
<tr>
<td>490</td>
<td>53.9</td>
</tr>
<tr>
<td>495</td>
<td>56.8</td>
</tr>
<tr>
<td>500</td>
<td>59.9</td>
</tr>
<tr>
<td>505</td>
<td>62.9</td>
</tr>
<tr>
<td>510</td>
<td>66.1</td>
</tr>
<tr>
<td>515</td>
<td>69.3</td>
</tr>
<tr>
<td>520</td>
<td>72.5</td>
</tr>
<tr>
<td>525</td>
<td>75.8</td>
</tr>
<tr>
<td>530</td>
<td>79.1</td>
</tr>
<tr>
<td>535</td>
<td>82.5</td>
</tr>
<tr>
<td>540</td>
<td>85.9</td>
</tr>
<tr>
<td>545</td>
<td>89.4</td>
</tr>
<tr>
<td>550</td>
<td>92.9</td>
</tr>
<tr>
<td>555</td>
<td>96.4</td>
</tr>
<tr>
<td>560</td>
<td>100.0</td>
</tr>
<tr>
<td>565</td>
<td>103.6</td>
</tr>
<tr>
<td>570</td>
<td>107.2</td>
</tr>
<tr>
<td>575</td>
<td>110.8</td>
</tr>
<tr>
<td>580</td>
<td>114.4</td>
</tr>
<tr>
<td>585</td>
<td>118.1</td>
</tr>
<tr>
<td>590</td>
<td>121.7</td>
</tr>
<tr>
<td>595</td>
<td>125.4</td>
</tr>
<tr>
<td>600</td>
<td>129.0</td>
</tr>
<tr>
<td>605</td>
<td>132.7</td>
</tr>
<tr>
<td>610</td>
<td>136.4</td>
</tr>
<tr>
<td>615</td>
<td>140.0</td>
</tr>
<tr>
<td>620</td>
<td>143.6</td>
</tr>
<tr>
<td>625</td>
<td>147.2</td>
</tr>
<tr>
<td>630</td>
<td>150.8</td>
</tr>
<tr>
<td>635</td>
<td>154.4</td>
</tr>
<tr>
<td>640</td>
<td>158.0</td>
</tr>
<tr>
<td>645</td>
<td>161.5</td>
</tr>
<tr>
<td>650</td>
<td>165.0</td>
</tr>
<tr>
<td>655</td>
<td>168.5</td>
</tr>
<tr>
<td>660</td>
<td>172.0</td>
</tr>
<tr>
<td>665</td>
<td>175.4</td>
</tr>
<tr>
<td>670</td>
<td>178.8</td>
</tr>
<tr>
<td>675</td>
<td>182.1</td>
</tr>
<tr>
<td>680</td>
<td>185.4</td>
</tr>
<tr>
<td>685</td>
<td>188.7</td>
</tr>
<tr>
<td>690</td>
<td>191.9</td>
</tr>
<tr>
<td>695</td>
<td>195.1</td>
</tr>
<tr>
<td>700</td>
<td>198.3</td>
</tr>
<tr>
<td>705</td>
<td>201.4</td>
</tr>
<tr>
<td>710</td>
<td>204.4</td>
</tr>
<tr>
<td>715</td>
<td>207.4</td>
</tr>
<tr>
<td>720</td>
<td>210.4</td>
</tr>
<tr>
<td>725</td>
<td>213.3</td>
</tr>
<tr>
<td>730</td>
<td>216.1</td>
</tr>
<tr>
<td>735</td>
<td>218.9</td>
</tr>
</tbody>
</table>
3.20 D65 spectrum

The D65 spectrum used in some calculations is this:

![D65 spectrum graph]

Here is a table of the values:

<table>
<thead>
<tr>
<th>Wavelength [nm]</th>
<th>Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>380</td>
<td>50.0</td>
</tr>
<tr>
<td>385</td>
<td>52.3</td>
</tr>
<tr>
<td>390</td>
<td>54.6</td>
</tr>
<tr>
<td>395</td>
<td>68.7</td>
</tr>
<tr>
<td>400</td>
<td>82.8</td>
</tr>
<tr>
<td>405</td>
<td>87.1</td>
</tr>
<tr>
<td>410</td>
<td>91.5</td>
</tr>
<tr>
<td>415</td>
<td>92.5</td>
</tr>
<tr>
<td>420</td>
<td>93.4</td>
</tr>
<tr>
<td>425</td>
<td>90.1</td>
</tr>
<tr>
<td>430</td>
<td>86.7</td>
</tr>
<tr>
<td>435</td>
<td>95.8</td>
</tr>
<tr>
<td>440</td>
<td>104.9</td>
</tr>
<tr>
<td>445</td>
<td>110.9</td>
</tr>
<tr>
<td>450</td>
<td>117.0</td>
</tr>
<tr>
<td>455</td>
<td>117.4</td>
</tr>
<tr>
<td>460</td>
<td>117.8</td>
</tr>
<tr>
<td>465</td>
<td>116.3</td>
</tr>
</tbody>
</table>
The C spectrum used in some calculations is this:

Here is a table of the values:

<table>
<thead>
<tr>
<th>Wavelength [nm]</th>
<th>Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>380</td>
<td>31.3</td>
</tr>
<tr>
<td>385</td>
<td>37.9</td>
</tr>
<tr>
<td>390</td>
<td>45.0</td>
</tr>
<tr>
<td>395</td>
<td>52.4</td>
</tr>
<tr>
<td>400</td>
<td>60.1</td>
</tr>
<tr>
<td>405</td>
<td>68.2</td>
</tr>
<tr>
<td>410</td>
<td>76.5</td>
</tr>
<tr>
<td>415</td>
<td>85.0</td>
</tr>
<tr>
<td>420</td>
<td>93.2</td>
</tr>
<tr>
<td>425</td>
<td>100.5</td>
</tr>
<tr>
<td>430</td>
<td>106.7</td>
</tr>
<tr>
<td>435</td>
<td>111.8</td>
</tr>
<tr>
<td>440</td>
<td>115.4</td>
</tr>
<tr>
<td>445</td>
<td>117.2</td>
</tr>
<tr>
<td>450</td>
<td>117.8</td>
</tr>
<tr>
<td>455</td>
<td>117.4</td>
</tr>
<tr>
<td>460</td>
<td>116.9</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>465</td>
<td>117.1</td>
</tr>
<tr>
<td>470</td>
<td>117.6</td>
</tr>
<tr>
<td>475</td>
<td>117.8</td>
</tr>
<tr>
<td>480</td>
<td>117.7</td>
</tr>
<tr>
<td>485</td>
<td>116.7</td>
</tr>
<tr>
<td>490</td>
<td>114.6</td>
</tr>
<tr>
<td>495</td>
<td>111.0</td>
</tr>
<tr>
<td>500</td>
<td>106.5</td>
</tr>
<tr>
<td>505</td>
<td>101.6</td>
</tr>
<tr>
<td>510</td>
<td>97.2</td>
</tr>
<tr>
<td>515</td>
<td>93.8</td>
</tr>
<tr>
<td>520</td>
<td>92.0</td>
</tr>
<tr>
<td>525</td>
<td>91.9</td>
</tr>
<tr>
<td>530</td>
<td>93.1</td>
</tr>
<tr>
<td>535</td>
<td>94.9</td>
</tr>
<tr>
<td>540</td>
<td>97.0</td>
</tr>
<tr>
<td>545</td>
<td>98.7</td>
</tr>
<tr>
<td>550</td>
<td>99.9</td>
</tr>
<tr>
<td>555</td>
<td>100.3</td>
</tr>
<tr>
<td>560</td>
<td>100.0</td>
</tr>
<tr>
<td>565</td>
<td>98.9</td>
</tr>
<tr>
<td>570</td>
<td>97.2</td>
</tr>
<tr>
<td>575</td>
<td>95.1</td>
</tr>
<tr>
<td>580</td>
<td>92.9</td>
</tr>
<tr>
<td>585</td>
<td>90.6</td>
</tr>
<tr>
<td>590</td>
<td>88.5</td>
</tr>
<tr>
<td>595</td>
<td>86.6</td>
</tr>
<tr>
<td>600</td>
<td>85.2</td>
</tr>
<tr>
<td>605</td>
<td>84.4</td>
</tr>
<tr>
<td>610</td>
<td>83.9</td>
</tr>
<tr>
<td>615</td>
<td>83.8</td>
</tr>
<tr>
<td>620</td>
<td>83.7</td>
</tr>
<tr>
<td>625</td>
<td>83.6</td>
</tr>
<tr>
<td>630</td>
<td>83.6</td>
</tr>
<tr>
<td>635</td>
<td>83.4</td>
</tr>
<tr>
<td>640</td>
<td>83.4</td>
</tr>
<tr>
<td>645</td>
<td>83.6</td>
</tr>
<tr>
<td>650</td>
<td>83.8</td>
</tr>
<tr>
<td>655</td>
<td>83.8</td>
</tr>
<tr>
<td>660</td>
<td>82.5</td>
</tr>
<tr>
<td>665</td>
<td>82.8</td>
</tr>
<tr>
<td>670</td>
<td>82.0</td>
</tr>
<tr>
<td>675</td>
<td>81.0</td>
</tr>
<tr>
<td>680</td>
<td>79.8</td>
</tr>
<tr>
<td>685</td>
<td>78.1</td>
</tr>
<tr>
<td>690</td>
<td>76.2</td>
</tr>
<tr>
<td>695</td>
<td>74.3</td>
</tr>
<tr>
<td>700</td>
<td>72.5</td>
</tr>
<tr>
<td>705</td>
<td>70.6</td>
</tr>
<tr>
<td>710</td>
<td>68.8</td>
</tr>
<tr>
<td>715</td>
<td>66.9</td>
</tr>
<tr>
<td>720</td>
<td>64.9</td>
</tr>
<tr>
<td>725</td>
<td>63.0</td>
</tr>
<tr>
<td>730</td>
<td>61.2</td>
</tr>
<tr>
<td>735</td>
<td>59.6</td>
</tr>
<tr>
<td>740</td>
<td>58.4</td>
</tr>
</tbody>
</table>
### 3.22 User-defined illuminants for color computation

The list of integral quantities offers the menu item **File/Options/Import additional illuminants for color computations**. Execute this command if you need to compute color values for illuminants other than the default ones A, D65 and C. You are asked to select a file which must be a text file with columns separated by semicolons. The first column contains 81 wavelengths from 380 to 780 nm in 5 nm steps, beginning in row 2. Starting with the second column, the file contains the name of the illuminant in the top row, following by 81 intensity values in the rows below. The table below shows the demo file illuminants.csv as generated by Excel.

<table>
<thead>
<tr>
<th>Wavelength (nm)</th>
<th>D50</th>
<th>D55</th>
<th>D75</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>380</td>
<td>24.488</td>
<td>32.584</td>
<td>66.703</td>
<td>1.87</td>
<td>1.18</td>
<td>0.82</td>
<td>0.57</td>
<td>1.87</td>
<td>1.05</td>
<td>2.56</td>
<td>1.21</td>
<td>0.9</td>
<td>1.11</td>
<td>0.91</td>
<td>0.96</td>
</tr>
<tr>
<td>385</td>
<td>27.179</td>
<td>35.335</td>
<td>68.333</td>
<td>2.36</td>
<td>1.48</td>
<td>1.02</td>
<td>0.7</td>
<td>2.35</td>
<td>1.31</td>
<td>3.18</td>
<td>1.5</td>
<td>1.2</td>
<td>0.63</td>
<td>0.63</td>
<td>0.64</td>
</tr>
<tr>
<td>390</td>
<td>29.871</td>
<td>38.087</td>
<td>69.963</td>
<td>2.94</td>
<td>1.84</td>
<td>1.26</td>
<td>0.87</td>
<td>2.92</td>
<td>1.63</td>
<td>3.84</td>
<td>1.81</td>
<td>1.36</td>
<td>0.62</td>
<td>0.46</td>
<td>0.45</td>
</tr>
<tr>
<td>395</td>
<td>39.589</td>
<td>49.518</td>
<td>85.946</td>
<td>3.47</td>
<td>2.15</td>
<td>1.44</td>
<td>0.98</td>
<td>3.45</td>
<td>1.9</td>
<td>4.53</td>
<td>2.13</td>
<td>1.6</td>
<td>0.57</td>
<td>0.37</td>
<td>0.33</td>
</tr>
<tr>
<td>400</td>
<td>49.308</td>
<td>60.949</td>
<td>101.929</td>
<td>5.17</td>
<td>3.44</td>
<td>2.57</td>
<td>2.01</td>
<td>5.1</td>
<td>3.11</td>
<td>6.15</td>
<td>3.17</td>
<td>2.59</td>
<td>1.48</td>
<td>1.29</td>
<td>1.19</td>
</tr>
<tr>
<td>405</td>
<td>52.91</td>
<td>64.751</td>
<td>106.911</td>
<td>14.99</td>
<td>15.69</td>
<td>14.36</td>
<td>13.75</td>
<td>18.91</td>
<td>14.8</td>
<td>19.37</td>
<td>13.08</td>
<td>12.8</td>
<td>12.16</td>
<td>12.48</td>
<td></td>
</tr>
<tr>
<td>410</td>
<td>56.513</td>
<td>68.554</td>
<td>111.894</td>
<td>6.13</td>
<td>3.85</td>
<td>2.71</td>
<td>1.95</td>
<td>6.3</td>
<td>4.33</td>
<td>7.37</td>
<td>3.83</td>
<td>3.05</td>
<td>2.12</td>
<td>1.59</td>
<td>1.12</td>
</tr>
<tr>
<td>415</td>
<td>58.273</td>
<td>70.065</td>
<td>112.346</td>
<td>6.24</td>
<td>3.74</td>
<td>2.45</td>
<td>1.59</td>
<td>6.11</td>
<td>3.3</td>
<td>7.05</td>
<td>3.45</td>
<td>2.56</td>
<td>2.7</td>
<td>1.79</td>
<td>0.94</td>
</tr>
<tr>
<td>420</td>
<td>60.034</td>
<td>71.577</td>
<td>112.798</td>
<td>7.01</td>
<td>4.19</td>
<td>2.73</td>
<td>1.76</td>
<td>6.85</td>
<td>3.68</td>
<td>7.71</td>
<td>3.86</td>
<td>2.86</td>
<td>3.74</td>
<td>2.46</td>
<td>1.08</td>
</tr>
<tr>
<td>425</td>
<td>58.926</td>
<td>69.746</td>
<td>107.945</td>
<td>7.79</td>
<td>4.62</td>
<td>3.19</td>
<td>1.93</td>
<td>7.58</td>
<td>4.07</td>
<td>8.41</td>
<td>4.42</td>
<td>3.5</td>
<td>4.33</td>
<td>1.37</td>
<td></td>
</tr>
<tr>
<td>430</td>
<td>57.818</td>
<td>67.914</td>
<td>103.092</td>
<td>8.56</td>
<td>5.06</td>
<td>3.38</td>
<td>2.01</td>
<td>8.31</td>
<td>4.45</td>
<td>9.15</td>
<td>5.21</td>
<td>3.82</td>
<td>0.75</td>
<td>4.49</td>
<td>1.78</td>
</tr>
<tr>
<td>435</td>
<td>66.321</td>
<td>76.76</td>
<td>112.145</td>
<td>43.67</td>
<td>34.98</td>
<td>31.85</td>
<td>30.28</td>
<td>40.07</td>
<td>36.11</td>
<td>44.14</td>
<td>34.1</td>
<td>32.61</td>
<td>34.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>440</td>
<td>74.825</td>
<td>85.605</td>
<td>121.198</td>
<td>16.94</td>
<td>11.81</td>
<td>9.47</td>
<td>8.03</td>
<td>16.06</td>
<td>10.74</td>
<td>17.52</td>
<td>12.42</td>
<td>10.77</td>
<td>14.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>445</td>
<td>81.036</td>
<td>91.799</td>
<td>127.104</td>
<td>10.72</td>
<td>6.27</td>
<td>4.02</td>
<td>2.55</td>
<td>10.32</td>
<td>5.48</td>
<td>11.35</td>
<td>7.68</td>
<td>5.84</td>
<td>10.4</td>
<td>6.95</td>
<td></td>
</tr>
<tr>
<td>450</td>
<td>87.247</td>
<td>97.993</td>
<td>133.01</td>
<td>11.35</td>
<td>6.63</td>
<td>4.25</td>
<td>2.7</td>
<td>10.91</td>
<td>5.78</td>
<td>12.86</td>
<td>6.57</td>
<td>10.76</td>
<td>7.19</td>
<td>2.71</td>
<td></td>
</tr>
<tr>
<td>460</td>
<td>90.612</td>
<td>100.463</td>
<td>132.355</td>
<td>12.37</td>
<td>7.19</td>
<td>4.59</td>
<td>2.9</td>
<td>11.83</td>
<td>6.25</td>
<td>13.08</td>
<td>10.24</td>
<td>7.86</td>
<td>10.11</td>
<td>6.72</td>
<td></td>
</tr>
<tr>
<td>465</td>
<td>90.99</td>
<td>100.188</td>
<td>129.838</td>
<td>12.75</td>
<td>7.44</td>
<td>4.72</td>
<td>2.99</td>
<td>12.17</td>
<td>6.41</td>
<td>13.45</td>
<td>10.84</td>
<td>8.35</td>
<td>9.27</td>
<td>6.13</td>
<td></td>
</tr>
<tr>
<td>470</td>
<td>91.368</td>
<td>99.913</td>
<td>127.322</td>
<td>13.75</td>
<td>7.54</td>
<td>4.83</td>
<td>3.04</td>
<td>12.4</td>
<td>6.52</td>
<td>13.71</td>
<td>11.33</td>
<td>8.75</td>
<td>8.29</td>
<td>5.46</td>
<td></td>
</tr>
<tr>
<td>480</td>
<td>95.109</td>
<td>102.739</td>
<td>126.8</td>
<td>13.23</td>
<td>7.65</td>
<td>4.87</td>
<td>3.09</td>
<td>12.58</td>
<td>6.59</td>
<td>13.95</td>
<td>11.98</td>
<td>9.31</td>
<td>7.91</td>
<td>5.66</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.33</td>
<td>3.01</td>
<td>4.08</td>
<td>3.09</td>
<td>12.52</td>
<td>6.56</td>
<td>13.93</td>
<td>12.17</td>
<td>9.48</td>
<td>16.64</td>
<td>1.49</td>
<td>10.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.01</td>
<td>4.08</td>
<td>3.09</td>
<td>12.52</td>
<td>6.56</td>
<td>13.93</td>
<td>12.17</td>
<td>9.48</td>
<td>16.64</td>
<td>1.49</td>
<td>10.83</td>
<td></td>
</tr>
</tbody>
</table>

© 2019 Wolfgang Theiss
665;100.597;92.133;74.873;2.81;2.96;3.23;3.69;2.49;2.57;8.65;12.58;13.2;2;2.14;2.42
670;103.003;93.95;75.422;2.47;2.55;2.75;3.13;2.19;2.2;7.27;10.99;10.57;1.52;1.54;1.64
675;101.068;91.953;73.499;2.18;2.19;2.33;2.64;1.93;1.87;6.44;9.98;9.18;1.35;1.33;1.36
680;99.133;89.956;71.576;1.93;1.89;1.99;2.24;1.71;1.65;8.53;9.22;8.25;1.47;1.46;1.49
685;93.257;84.817;67.714;1.72;1.64;1.7;1.91;1.52;1.37;5.41;8.62;7.57;1.79;1.94;2.14
690;87.381;79.677;63.852;1.67;1.53;1.55;1.71;1.43;1.29;5.04;8.08;7.07;1.36;1.74;2.34
695;89.492;81.258;64.464;1.43;1.41;1.27;1.39;1.26;1.05;5.47;7.39;6.35;1.02;1.2;1.42
700;91.604;82.842;65.076;1.29;1.1;1.09;1.18;1.13;0.91;1.42;6.71;5.75;1.14;1.35;1.61
705;92.246;83.84;66.573;1.19;0.99;0.96;1.03;1.05;0.81;3.77;6.16;5.25;3.32;4.1;5.04
710;92.889;84.844;68.07;1.08;0.88;0.85;0.88;0.96;0.71;3.46;5.63;4.8;4.49;5.58;6.98
715;84.872;77.539;62.256;0.96;0.76;0.71;0.74;0.85;0.61;3.08;5.03;4.29;2.05;2.51;3.19
720;76.854;70.235;56.443;0.88;0.68;0.62;0.64;0.78;0.54;2.73;4.46;3.8;0.49;0.57;0.71
725;81.683;74.769;60.343;0.81;0.61;0.54;0.54;0.72;0.48;2.47;4.02;3.43;0.24;0.27;0.3
730;86.511;79.301;64.242;0.77;0.56;0.49;0.49;0.68;0.44;2.25;3.66;3.12;0.21;0.23;0.26
735;89.546;82.147;66.697;0.75;0.54;0.46;0.46;0.67;0.43;2.06;3.36;2.86;0.21;0.21;0.21
740;92.58;84.993;69.151;0.73;0.51;0.43;0.42;0.65;0.41;0.9;3.09;2.64;0.24;0.24;0.28
745;85.405;78.437;63.899;0.68;0.47;0.39;0.37;0.61;0.37;1.75;2.85;2.43;0.24;0.24;0.28
750;78.237;71.888;58.629;0.69;0.47;0.39;0.37;0.62;0.38;1.62;2.65;2.26;0.21;0.2;0.21
755;67.961;62.337;50.623;0.64;0.43;0.35;0.33;0.59;0.35;1.54;2.51;2.14;0.17;0.24;0.17
760;57.692;52.793;42.617;0.68;0.46;0.38;0.35;0.62;0.39;1.45;2.37;2.02;0.21;0.32;0.22
765;70.307;64.36;51.985;0.69;0.47;0.39;0.36;0.64;0.41;1.32;2.15;1.83;0.22;0.26;0.19
770;82.923;75.927;61.352;0.61;0.40;0.33;0.31;0.55;0.33;1.71;1.89;1.61;0.17;0.16;0.15
775;80.599;73.872;59.838;0.52;0.33;0.28;0.26;0.47;0.26;0.99;1.61;1.38;0.12;0.12;0.11
780;78.274;71.818;58.324;0.43;0.27;0.21;0.19;0.4;0.21;0.81;1.32;1.12;0.09;0.09;0.05

Opened in Excel this looks like this:

<table>
<thead>
<tr>
<th>CODE</th>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>D50</td>
<td>24.488</td>
</tr>
<tr>
<td>D55</td>
<td>32.584</td>
</tr>
<tr>
<td>D75</td>
<td>66.703</td>
</tr>
<tr>
<td>D70</td>
<td>51.985</td>
</tr>
<tr>
<td>D75</td>
<td>69.963</td>
</tr>
<tr>
<td>D70</td>
<td>2.81</td>
</tr>
<tr>
<td>D75</td>
<td>1.89</td>
</tr>
<tr>
<td>D70</td>
<td>3.69</td>
</tr>
<tr>
<td>D75</td>
<td>2.49</td>
</tr>
<tr>
<td>D70</td>
<td>2.57</td>
</tr>
<tr>
<td>D75</td>
<td>8.65</td>
</tr>
<tr>
<td>D70</td>
<td>12.58</td>
</tr>
<tr>
<td>D75</td>
<td>13.2</td>
</tr>
<tr>
<td>D70</td>
<td>2.14</td>
</tr>
<tr>
<td>D75</td>
<td>2.42</td>
</tr>
<tr>
<td>D70</td>
<td>2.42</td>
</tr>
<tr>
<td>D75</td>
<td>100.597</td>
</tr>
<tr>
<td>D70</td>
<td>92.133</td>
</tr>
<tr>
<td>D75</td>
<td>74.873</td>
</tr>
<tr>
<td>D70</td>
<td>2.81</td>
</tr>
<tr>
<td>D75</td>
<td>2.96</td>
</tr>
<tr>
<td>D70</td>
<td>3.23</td>
</tr>
<tr>
<td>D75</td>
<td>3.69</td>
</tr>
<tr>
<td>D70</td>
<td>2.49</td>
</tr>
<tr>
<td>D75</td>
<td>2.57</td>
</tr>
<tr>
<td>D70</td>
<td>8.65</td>
</tr>
<tr>
<td>D75</td>
<td>12.58</td>
</tr>
<tr>
<td>D70</td>
<td>13.2</td>
</tr>
<tr>
<td>D75</td>
<td>2.14</td>
</tr>
<tr>
<td>D70</td>
<td>2.42</td>
</tr>
</tbody>
</table>

© 2019 Wolfgang Theiss
<table>
<thead>
<tr>
<th></th>
<th>CODE</th>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>475</td>
<td>93.238</td>
<td>101.32</td>
</tr>
<tr>
<td>480</td>
<td>95.109</td>
<td>102.73</td>
</tr>
<tr>
<td>485</td>
<td>93.536</td>
<td>100.40</td>
</tr>
<tr>
<td>490</td>
<td>91.963</td>
<td>98.078</td>
</tr>
<tr>
<td>495</td>
<td>93.843</td>
<td>99.379</td>
</tr>
<tr>
<td>500</td>
<td>95.724</td>
<td>100.68</td>
</tr>
<tr>
<td>505</td>
<td>96.169</td>
<td>100.68</td>
</tr>
<tr>
<td>510</td>
<td>96.613</td>
<td>100.69</td>
</tr>
<tr>
<td>515</td>
<td>96.871</td>
<td>100.34</td>
</tr>
<tr>
<td>520</td>
<td>97.129</td>
<td>99.987</td>
</tr>
<tr>
<td>525</td>
<td>99.614</td>
<td>102.09</td>
</tr>
<tr>
<td>530</td>
<td>102.09</td>
<td>104.21</td>
</tr>
<tr>
<td>535</td>
<td>101.42</td>
<td>103.15</td>
</tr>
<tr>
<td>540</td>
<td>100.75</td>
<td>102.10</td>
</tr>
<tr>
<td>545</td>
<td>101.53</td>
<td>102.53</td>
</tr>
<tr>
<td>550</td>
<td>102.31</td>
<td>102.96</td>
</tr>
<tr>
<td>555</td>
<td>101.15</td>
<td>101.48</td>
</tr>
<tr>
<td>560</td>
<td>100.70</td>
<td>100.10</td>
</tr>
<tr>
<td>565</td>
<td>98.868</td>
<td>98.608</td>
</tr>
<tr>
<td>570</td>
<td>97.735</td>
<td>97.216</td>
</tr>
<tr>
<td>575</td>
<td>98.327</td>
<td>97.482</td>
</tr>
<tr>
<td>580</td>
<td>98.918</td>
<td>97.749</td>
</tr>
<tr>
<td>585</td>
<td>96.208</td>
<td>94.59</td>
</tr>
<tr>
<td>590</td>
<td>93.499</td>
<td>91.432</td>
</tr>
<tr>
<td>595</td>
<td>95.593</td>
<td>92.926</td>
</tr>
<tr>
<td>600</td>
<td>97.688</td>
<td>94.419</td>
</tr>
<tr>
<td>605</td>
<td>98.478</td>
<td>94.78</td>
</tr>
<tr>
<td>610</td>
<td>99.269</td>
<td>95.14</td>
</tr>
<tr>
<td>615</td>
<td>99.155</td>
<td>94.68</td>
</tr>
<tr>
<td>620</td>
<td>99.042</td>
<td>94.22</td>
</tr>
<tr>
<td>625</td>
<td>97.382</td>
<td>92.334</td>
</tr>
<tr>
<td>630</td>
<td>95.722</td>
<td>90.448</td>
</tr>
<tr>
<td>635</td>
<td>97.291</td>
<td>91.389</td>
</tr>
<tr>
<td>640</td>
<td>96.857</td>
<td>92.33</td>
</tr>
<tr>
<td>645</td>
<td>97.262</td>
<td>90.592</td>
</tr>
<tr>
<td>650</td>
<td>95.667</td>
<td>88.854</td>
</tr>
<tr>
<td>655</td>
<td>96.929</td>
<td>89.586</td>
</tr>
<tr>
<td>660</td>
<td>98.190</td>
<td>90.317</td>
</tr>
<tr>
<td>665</td>
<td>100.59</td>
<td>92.133</td>
</tr>
<tr>
<td>670</td>
<td>103.00</td>
<td>93.95</td>
</tr>
<tr>
<td>675</td>
<td>101.06</td>
<td>91.953</td>
</tr>
<tr>
<td>680</td>
<td>99.133</td>
<td>89.956</td>
</tr>
</tbody>
</table>

© 2019 Wolfgang Theiss
Here is a graph of these demo spectra:
3.23 Color-matching functions

The color-matching functions for 2° used for the computation of color coordinates are these:

The values are taken from DIN 5033:

<table>
<thead>
<tr>
<th>Wavelength [nm]</th>
<th>xbar</th>
<th>ybar</th>
<th>zbar</th>
</tr>
</thead>
<tbody>
<tr>
<td>380</td>
<td>0.0014</td>
<td>0.0000</td>
<td>0.0065</td>
</tr>
<tr>
<td>385</td>
<td>0.0022</td>
<td>0.0001</td>
<td>0.0105</td>
</tr>
<tr>
<td>390</td>
<td>0.0042</td>
<td>0.0001</td>
<td>0.0202</td>
</tr>
<tr>
<td>395</td>
<td>0.0076</td>
<td>0.0002</td>
<td>0.0362</td>
</tr>
<tr>
<td>400</td>
<td>0.0143</td>
<td>0.0004</td>
<td>0.0679</td>
</tr>
<tr>
<td>405</td>
<td>0.0232</td>
<td>0.0006</td>
<td>0.1102</td>
</tr>
<tr>
<td>410</td>
<td>0.0435</td>
<td>0.0012</td>
<td>0.2074</td>
</tr>
<tr>
<td>415</td>
<td>0.0776</td>
<td>0.0022</td>
<td>0.3713</td>
</tr>
<tr>
<td>420</td>
<td>0.1344</td>
<td>0.0040</td>
<td>0.6456</td>
</tr>
<tr>
<td>425</td>
<td>0.2148</td>
<td>0.0073</td>
<td>1.0391</td>
</tr>
<tr>
<td>430</td>
<td>0.2839</td>
<td>0.0116</td>
<td>1.3856</td>
</tr>
<tr>
<td>435</td>
<td>0.3285</td>
<td>0.0168</td>
<td>1.6230</td>
</tr>
<tr>
<td>440</td>
<td>0.3483</td>
<td>0.0230</td>
<td>1.7471</td>
</tr>
<tr>
<td>445</td>
<td>0.3481</td>
<td>0.0298</td>
<td>1.7826</td>
</tr>
<tr>
<td>450</td>
<td>0.3362</td>
<td>0.0380</td>
<td>1.7721</td>
</tr>
<tr>
<td>455</td>
<td>0.3187</td>
<td>0.0480</td>
<td>1.7441</td>
</tr>
<tr>
<td>460</td>
<td>0.2908</td>
<td>0.0600</td>
<td>1.6692</td>
</tr>
<tr>
<td>465</td>
<td>0.2511</td>
<td>0.0739</td>
<td>1.5281</td>
</tr>
<tr>
<td>470</td>
<td>0.1954</td>
<td>0.0910</td>
<td>1.2876</td>
</tr>
<tr>
<td>475</td>
<td>0.1421</td>
<td>0.1126</td>
<td>1.0419</td>
</tr>
<tr>
<td>Code</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>------</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>480</td>
<td>0.0956</td>
<td>0.1390</td>
<td>0.8130</td>
</tr>
<tr>
<td>485</td>
<td>0.0580</td>
<td>0.1693</td>
<td>0.6162</td>
</tr>
<tr>
<td>490</td>
<td>0.0320</td>
<td>0.2080</td>
<td>0.4652</td>
</tr>
<tr>
<td>495</td>
<td>0.0147</td>
<td>0.2586</td>
<td>0.3533</td>
</tr>
<tr>
<td>500</td>
<td>0.0049</td>
<td>0.3230</td>
<td>0.2720</td>
</tr>
<tr>
<td>505</td>
<td>0.0024</td>
<td>0.3926</td>
<td>0.2123</td>
</tr>
<tr>
<td>510</td>
<td>0.0093</td>
<td>0.5030</td>
<td>0.1582</td>
</tr>
<tr>
<td>515</td>
<td>0.0291</td>
<td>0.6082</td>
<td>0.1117</td>
</tr>
<tr>
<td>520</td>
<td>0.0633</td>
<td>0.7100</td>
<td>0.0782</td>
</tr>
<tr>
<td>525</td>
<td>0.1096</td>
<td>0.7932</td>
<td>0.0573</td>
</tr>
<tr>
<td>530</td>
<td>0.1655</td>
<td>0.8620</td>
<td>0.0422</td>
</tr>
<tr>
<td>535</td>
<td>0.2257</td>
<td>0.9149</td>
<td>0.0298</td>
</tr>
<tr>
<td>540</td>
<td>0.2904</td>
<td>0.9540</td>
<td>0.0203</td>
</tr>
<tr>
<td>545</td>
<td>0.3597</td>
<td>0.9803</td>
<td>0.0134</td>
</tr>
<tr>
<td>550</td>
<td>0.4334</td>
<td>0.9950</td>
<td>0.0087</td>
</tr>
<tr>
<td>555</td>
<td>0.5121</td>
<td>1.0000</td>
<td>0.0057</td>
</tr>
<tr>
<td>560</td>
<td>0.5945</td>
<td>0.9959</td>
<td>0.0039</td>
</tr>
<tr>
<td>565</td>
<td>0.6784</td>
<td>0.9766</td>
<td>0.0027</td>
</tr>
<tr>
<td>570</td>
<td>0.7621</td>
<td>0.9520</td>
<td>0.0021</td>
</tr>
<tr>
<td>575</td>
<td>0.8425</td>
<td>0.9154</td>
<td>0.0018</td>
</tr>
<tr>
<td>580</td>
<td>0.9163</td>
<td>0.8700</td>
<td>0.0017</td>
</tr>
<tr>
<td>585</td>
<td>0.9786</td>
<td>0.8163</td>
<td>0.0014</td>
</tr>
<tr>
<td>590</td>
<td>1.0263</td>
<td>0.7570</td>
<td>0.0011</td>
</tr>
<tr>
<td>595</td>
<td>1.0567</td>
<td>0.6949</td>
<td>0.0010</td>
</tr>
<tr>
<td>600</td>
<td>1.0622</td>
<td>0.6310</td>
<td>0.0008</td>
</tr>
<tr>
<td>605</td>
<td>1.0456</td>
<td>0.5668</td>
<td>0.0006</td>
</tr>
<tr>
<td>610</td>
<td>1.0026</td>
<td>0.5030</td>
<td>0.0003</td>
</tr>
<tr>
<td>615</td>
<td>0.9384</td>
<td>0.4412</td>
<td>0.0002</td>
</tr>
<tr>
<td>620</td>
<td>0.8544</td>
<td>0.3810</td>
<td>0.0002</td>
</tr>
<tr>
<td>625</td>
<td>0.7514</td>
<td>0.3210</td>
<td>0.0001</td>
</tr>
<tr>
<td>630</td>
<td>0.6424</td>
<td>0.2650</td>
<td>0.0000</td>
</tr>
<tr>
<td>635</td>
<td>0.5419</td>
<td>0.2170</td>
<td>0.0000</td>
</tr>
<tr>
<td>640</td>
<td>0.4479</td>
<td>0.1750</td>
<td>0.0000</td>
</tr>
<tr>
<td>645</td>
<td>0.3608</td>
<td>0.1382</td>
<td>0.0000</td>
</tr>
<tr>
<td>650</td>
<td>0.2835</td>
<td>0.1070</td>
<td>0.0000</td>
</tr>
<tr>
<td>655</td>
<td>0.2187</td>
<td>0.0816</td>
<td>0.0000</td>
</tr>
<tr>
<td>660</td>
<td>0.1649</td>
<td>0.0610</td>
<td>0.0000</td>
</tr>
<tr>
<td>665</td>
<td>0.1212</td>
<td>0.0446</td>
<td>0.0000</td>
</tr>
<tr>
<td>670</td>
<td>0.0874</td>
<td>0.0320</td>
<td>0.0000</td>
</tr>
<tr>
<td>675</td>
<td>0.0636</td>
<td>0.0232</td>
<td>0.0000</td>
</tr>
<tr>
<td>680</td>
<td>0.0468</td>
<td>0.0170</td>
<td>0.0000</td>
</tr>
<tr>
<td>685</td>
<td>0.0329</td>
<td>0.0119</td>
<td>0.0000</td>
</tr>
<tr>
<td>690</td>
<td>0.0227</td>
<td>0.0082</td>
<td>0.0000</td>
</tr>
<tr>
<td>695</td>
<td>0.0158</td>
<td>0.0057</td>
<td>0.0000</td>
</tr>
<tr>
<td>700</td>
<td>0.0114</td>
<td>0.0041</td>
<td>0.0000</td>
</tr>
<tr>
<td>705</td>
<td>0.0081</td>
<td>0.0029</td>
<td>0.0000</td>
</tr>
<tr>
<td>710</td>
<td>0.0058</td>
<td>0.0021</td>
<td>0.0000</td>
</tr>
<tr>
<td>715</td>
<td>0.0041</td>
<td>0.0015</td>
<td>0.0000</td>
</tr>
<tr>
<td>720</td>
<td>0.0029</td>
<td>0.0010</td>
<td>0.0000</td>
</tr>
<tr>
<td>725</td>
<td>0.0020</td>
<td>0.0007</td>
<td>0.0000</td>
</tr>
<tr>
<td>730</td>
<td>0.0014</td>
<td>0.0005</td>
<td>0.0000</td>
</tr>
<tr>
<td>735</td>
<td>0.0010</td>
<td>0.0004</td>
<td>0.0000</td>
</tr>
<tr>
<td>740</td>
<td>0.0007</td>
<td>0.0002</td>
<td>0.0000</td>
</tr>
<tr>
<td>745</td>
<td>0.0005</td>
<td>0.0002</td>
<td>0.0000</td>
</tr>
<tr>
<td>750</td>
<td>0.0003</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>755</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>760</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
For 10° the values are the following:

<table>
<thead>
<tr>
<th>Wavelength [nm]</th>
<th>xbar10</th>
<th>ybar10</th>
<th>zbar10</th>
</tr>
</thead>
<tbody>
<tr>
<td>380</td>
<td>0.0002</td>
<td>0.0000</td>
<td>0.0007</td>
</tr>
<tr>
<td>385</td>
<td>0.0007</td>
<td>0.0001</td>
<td>0.0029</td>
</tr>
<tr>
<td>390</td>
<td>0.0024</td>
<td>0.0003</td>
<td>0.0105</td>
</tr>
<tr>
<td>395</td>
<td>0.0072</td>
<td>0.0008</td>
<td>0.0323</td>
</tr>
<tr>
<td>400</td>
<td>0.0191</td>
<td>0.0020</td>
<td>0.0860</td>
</tr>
<tr>
<td>405</td>
<td>0.0434</td>
<td>0.0045</td>
<td>0.1971</td>
</tr>
<tr>
<td>410</td>
<td>0.0847</td>
<td>0.0088</td>
<td>0.3894</td>
</tr>
<tr>
<td>415</td>
<td>0.1406</td>
<td>0.0145</td>
<td>0.6568</td>
</tr>
<tr>
<td>420</td>
<td>0.2045</td>
<td>0.0214</td>
<td>0.9725</td>
</tr>
<tr>
<td>425</td>
<td>0.2647</td>
<td>0.0295</td>
<td>1.2825</td>
</tr>
<tr>
<td>430</td>
<td>0.3147</td>
<td>0.0387</td>
<td>1.5535</td>
</tr>
<tr>
<td>435</td>
<td>0.3577</td>
<td>0.0496</td>
<td>1.7985</td>
</tr>
<tr>
<td>440</td>
<td>0.3837</td>
<td>0.0621</td>
<td>1.9673</td>
</tr>
<tr>
<td>445</td>
<td>0.3867</td>
<td>0.0747</td>
<td>2.0273</td>
</tr>
<tr>
<td>450</td>
<td>0.3707</td>
<td>0.0895</td>
<td>1.9948</td>
</tr>
<tr>
<td>455</td>
<td>0.3430</td>
<td>0.1063</td>
<td>1.9007</td>
</tr>
<tr>
<td>460</td>
<td>0.3023</td>
<td>0.1282</td>
<td>1.7454</td>
</tr>
<tr>
<td>465</td>
<td>0.2541</td>
<td>0.1528</td>
<td>1.5549</td>
</tr>
<tr>
<td>470</td>
<td>0.1956</td>
<td>0.1852</td>
<td>1.3176</td>
</tr>
<tr>
<td>475</td>
<td>0.1323</td>
<td>0.2199</td>
<td>1.0302</td>
</tr>
<tr>
<td>480</td>
<td>0.0805</td>
<td>0.2536</td>
<td>0.7721</td>
</tr>
<tr>
<td>485</td>
<td>0.0411</td>
<td>0.2977</td>
<td>0.5701</td>
</tr>
<tr>
<td>490</td>
<td>0.0162</td>
<td>0.3391</td>
<td>0.4153</td>
</tr>
<tr>
<td>495</td>
<td>0.0051</td>
<td>0.3954</td>
<td>0.3024</td>
</tr>
<tr>
<td>500</td>
<td>0.0038</td>
<td>0.4608</td>
<td>0.2185</td>
</tr>
<tr>
<td>505</td>
<td>0.0154</td>
<td>0.5314</td>
<td>0.1592</td>
</tr>
<tr>
<td>510</td>
<td>0.0375</td>
<td>0.6067</td>
<td>0.1120</td>
</tr>
<tr>
<td>515</td>
<td>0.0714</td>
<td>0.6857</td>
<td>0.0822</td>
</tr>
<tr>
<td>520</td>
<td>0.1177</td>
<td>0.7618</td>
<td>0.0607</td>
</tr>
<tr>
<td>525</td>
<td>0.1730</td>
<td>0.8233</td>
<td>0.0431</td>
</tr>
<tr>
<td>530</td>
<td>0.2365</td>
<td>0.8752</td>
<td>0.0305</td>
</tr>
<tr>
<td>535</td>
<td>0.3042</td>
<td>0.9238</td>
<td>0.0206</td>
</tr>
<tr>
<td>540</td>
<td>0.3768</td>
<td>0.9620</td>
<td>0.0137</td>
</tr>
<tr>
<td>545</td>
<td>0.4516</td>
<td>0.9822</td>
<td>0.0079</td>
</tr>
<tr>
<td>550</td>
<td>0.5298</td>
<td>0.9918</td>
<td>0.0040</td>
</tr>
<tr>
<td>555</td>
<td>0.6161</td>
<td>0.9991</td>
<td>0.0011</td>
</tr>
<tr>
<td>560</td>
<td>0.7052</td>
<td>0.9973</td>
<td>0.0000</td>
</tr>
<tr>
<td>565</td>
<td>0.7938</td>
<td>0.9824</td>
<td>0.0000</td>
</tr>
<tr>
<td>570</td>
<td>0.8787</td>
<td>0.9556</td>
<td>0.0000</td>
</tr>
<tr>
<td>575</td>
<td>0.9512</td>
<td>0.9152</td>
<td>0.0000</td>
</tr>
<tr>
<td>580</td>
<td>1.0142</td>
<td>0.8689</td>
<td>0.0000</td>
</tr>
<tr>
<td>585</td>
<td>1.0743</td>
<td>0.8256</td>
<td>0.0000</td>
</tr>
<tr>
<td>590</td>
<td>1.1185</td>
<td>0.7774</td>
<td>0.0000</td>
</tr>
<tr>
<td>595</td>
<td>1.1343</td>
<td>0.7204</td>
<td>0.0000</td>
</tr>
<tr>
<td>600</td>
<td>1.1240</td>
<td>0.6583</td>
<td>0.0000</td>
</tr>
<tr>
<td>605</td>
<td>1.0891</td>
<td>0.5939</td>
<td>0.0000</td>
</tr>
<tr>
<td>610</td>
<td>1.0305</td>
<td>0.5280</td>
<td>0.0000</td>
</tr>
<tr>
<td>615</td>
<td>0.9507</td>
<td>0.4618</td>
<td>0.0000</td>
</tr>
<tr>
<td>620</td>
<td>0.8563</td>
<td>0.3981</td>
<td>0.0000</td>
</tr>
<tr>
<td>625</td>
<td>0.7549</td>
<td>0.3396</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
3.24 Solar spectral distribution

The product $S_{\lambda}$ used for the computation of the solar direct transmittance and reflectance is given for the required wavelengths in the following (DIN 67 507):

<table>
<thead>
<tr>
<th>Wavelength [nm]</th>
<th>$S_{\lambda}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>0.0050</td>
</tr>
<tr>
<td>340</td>
<td>0.0240</td>
</tr>
<tr>
<td>380</td>
<td>0.0320</td>
</tr>
<tr>
<td>420</td>
<td>0.0500</td>
</tr>
<tr>
<td>460</td>
<td>0.0650</td>
</tr>
<tr>
<td>500</td>
<td>0.0630</td>
</tr>
<tr>
<td>540</td>
<td>0.0580</td>
</tr>
<tr>
<td>580</td>
<td>0.0540</td>
</tr>
<tr>
<td>620</td>
<td>0.0550</td>
</tr>
<tr>
<td>660</td>
<td>0.0490</td>
</tr>
<tr>
<td>700</td>
<td>0.0460</td>
</tr>
<tr>
<td>740</td>
<td>0.0410</td>
</tr>
<tr>
<td>780</td>
<td>0.0370</td>
</tr>
<tr>
<td>900</td>
<td>0.1390</td>
</tr>
<tr>
<td>1100</td>
<td>0.0970</td>
</tr>
<tr>
<td>1300</td>
<td>0.0580</td>
</tr>
<tr>
<td>1500</td>
<td>0.0390</td>
</tr>
</tbody>
</table>
The relative spectral intensity of solar radiation in Europe is approximately given by the AM 1.5 spectrum:

```
1700  0.0260
1900  0.0180
2500  0.0440
```
Index

- % -
%  5

- 1 -
10°  8

- 2 -
2°  8

- 3 -
3.79  78
3-point-averaging  39

- A -
a  8, 67
A spectrum  80
a*  8, 17, 21, 44, 67
absorption  79
absorption bands  71
absorption coefficient  40
absorption of light  71
acquire  26
ActiveX palette  60
add_layer_definition_on_top  26
add_layer_on_top  28
add_to_report  28
additional illuminants  8
air  75
AM 1.5  71
angle distribution incidence spectrum  40, 41
angle distribution incidence weight spectrum  41
angle of incidence  8, 13, 17
angle range  8
angle variation  8
architectural applications  13
Argon  75
author  3
automatic fit  34
Automatic saving of page modifications  23

automation  60
Automation open  60
Automation refnum  60
average time  42
averaging  78

- B -
b  8, 67
b*  8, 17, 21, 44, 67
batch control window  21
Begin superlattice  26
black  8
Block diagram  60
bottom  38
building  13
button panel  24

- C -
C  8, 67
C spectrum  84
cell  56
center-of-glass  78
center-of-glass value  78
Charge carrier generation  71
checkbox  17, 75
checkboxes  8
CIELAB  67
clear_fit_parameters  29
clear_layer_stack  29
clear_material_list  30
clear_report  30
clipboard  36
coating  8, 56
coatings  79
coatings on glass  2
CODE  21
CODE library  60
coherent  26, 76
coherent superposition  28
color  8, 44
Color angle variation  17
Color box  21
color coordinate  44
color coordinates  8, 17, 65, 67, 91
color dialog  17
color distance  8
Color fluctuation  8
Index

fitting 48
fluctuation 8
fluctuations 13
foil 76
format 17
formula 8, 48
found_filename 49
foundfiles 49
frame effects 78
Frozen 48
function 16
Functions of integrals quantities 16

- G -
g 74
gas 76
gas filling 75
gas properties 78
Generate PDF document 25
get_sim_val1 49
get_sim_val2 49
glass 76
glass pane 71, 75
glazing 79
glazing composition 76
glazing system 78
graphics 8
grid 34
grid_test 34
grid_test_max 49
grid_test_min 49
grid_test_no_points 49
grid_test_para 49

- H -
halfspace 29, 38, 39
hardware 60
heat 79
heat transfer 74, 78
Height 51
hide 34
human 78

- I -
illuminant 8, 44, 80, 82, 84
illumination power 71
image 17
imaginary part 46
import 29, 33, 34
import data 32
import illuminants 8
imported dielectric function 32
incidence_angle 50
incoherent 26
incoherent superposition 28
index 53, 55
infrared 75
integral quantities 2, 5, 17, 21, 53, 76
integral quantity 17, 55
interpolation 5
ISO 15099 78
ISO 9845-1 78
iteration 42

- K -
k 65
kappa 51
Krypton 75

- L -
L 8, 67
L* 8, 17, 44, 67
L*a*b* space 8
LabView 60
layer 26, 28, 51
layer definitions 26
layer stack 13, 26, 28, 29, 38, 39, 51, 56, 76
layer stack definition 76
layer stacks 30, 52
layer thickness 8, 76
layer_thickness 51
layers 29, 52
left 51
light reflectance 15, 69
light transmittance 15, 68
list of configurations 23
list of fit parameters 30
list of integral quantities 5, 76
list of layer stacks 26, 28
list of materials 28, 30, 35
list of optical functions 21
‘List view’ 24
load dielectric function 35

© 2019 Wolfgang Theiss
- M -

macro language 56
macros 56
main window 17, 51
master parameter 30, 48, 71
Master parameter list 30
material 35, 36, 38, 39, 51
Materials 30
maximum 48
message 38
methods 26
micron 28
mm 28
mode parameter 17
model 56, 75
model parameters 56

- N -

National Fenestration Rating Council 78
Navigation 24
NFRC 78, 79
NFRC 300-2004 78
nickname 5, 16, 17
nm 28
normal emissivity 75
notation 76
number of optical functions 52
number_of_fit_parameters 52
number_of_spectra 53
number_of_tec_values 53
numerical value 17

- O -

observation angle 8, 44
observer 13
office tower 13
OLE 26
OLE automation 56
OLE commands 26
OLE example 56
operator display 60
optical constants 35, 37, 56, 75
optical design 2
optical function 53
optical functions 16, 21, 52, 78
Optimize 5
options 8
order 16
outside 74
outside temperature 78, 79

- P -

page 23, 25
Page down 24
Page up 24
parameter fit 40
Parameter fluctuation 21
Parameter variation 21, 71
partial wave 26
PDF file 25
Penalty shape 5
Photocurrent 71
photons 71
Photopic 78
physical properties 75
polarization 8, 13
position 39
prepare_shutdown 60
presentation 22, 23, 24
presentation folder 25
print 37
properties 26
Psi 49
Purity 8

- Q -

quantity 5

- R -

Ra 67
radiation 78
range 38
rating 45
real part 45
re-computation 40
rectangle 17
reflectance 13, 34
refractive index 47, 51, 54
Tables 5
target value 5, 8
TCO 71
tdw-ISO 78
tec_value 55
tec_value_name 55
technical data 56
technical values 53, 56
temperature distribution 78
temperature profile 79
text 53
text file 36
Thermal transmittance 75
Thick layer 26, 28
thickness 28, 51, 56
thickness fluctuations 13
thickness profile 60
thickness unit 28
thickness variation 26
Thin film 26, 28
thin film design 2
top 39
total solar energy transmittance 74
transfer 29, 33
transmittance 34, 65, 78
transparent conductive layer 71
treewview 23
tristimulus values 65
Tuv 78
type 5
type library 60
- U -
U 75
unit 5, 28
update_data 40
update_plot 40
upper boundary 48
US standard 78
user-defined formula 71
UV 78
U-value 78, 79
- V -
Vacuum 29, 30
value 5, 48
Variable thickness 26
variant 34
variant arrays 34
variation mode 48
variations 8
view element 17
view object 71
views 17
Virtual Office Tower 13, 17
visual navigation help 24
VisualBasic 26
visualization 8
VTc 78
- W -
wavelength 45, 46, 54
wavenumber 40, 46, 51, 54
WCD 2
weight 5, 55
weight function 8, 17
weighted 78
weighted average 78
white 8
Width 51
wind speed 78
window 39
WINDOW 4.0 78
window panes 13
Windows Coating Designer 2
Wolfgang Theiss 3
workbook 5, 21
worksheet 56
- X -
X 8, 44, 65, 66, 67
Xenon 75
x-y-plane 8
- Y -
Y 8, 44, 65, 66, 67
- Z -
Z 8, 44, 65, 66, 67
zip file 25