ColorCalculationTreasures

Version: 090927.2006
by Artoksi

Don’t think you can not succeed!
…Be independent

Let everybody to create his/her own Color Matching Softwares featured not only quality control but also recipe prediction.

Furthermore, softwares which can communicate with your spectrophotometer.

What else?

USER MANUAL

including an open source code showing all functionality of this DLL.

Engineering Art Center
www.artoksi.com

1) What is ColorCalculationTreasures.DLL?

As the name explains simply, this is a treasure consist of many "instant functions" on color calculation issues used by professional programmers on color matching software including recipe prediction..

Technically, I wrote all essential functions for a color matching software and packed it as DLL for common use.

The user can easily create his/her own Color Matching software (including recipe calculations) by calling the subroutines in this DLL as described in this manual. It is not necessary to know about color matching principals, understanding formulas, techniques, standards, etc. Just pass your input values to subroutine and get the output values in variable you declared in your source code.

That's all!

Note: This manual only describes how to use this DLL in your source code and assumes that you are familiar with programming language issues.

Note: As we said above "everybody can create professional color matching software with this DLL. You don't need to know how it is calculate. ColorCalculationTreasures.DLL will do it for you. But we assume that you are also familiar with Color Management Terminology like color spaces, indices, assessments, recipes, calibration shades, light sources, observer, metamerism and etc.

Note: We have tested and used and still testing and using this DLL in Visual Basic 6 in our lab. This does not mean that this DLL can only work with Visual Basic but informations on this manual may contain specific methods for Visual Basic programming languages such as variable declaration methods, DLL referencing methods, accessing DLL contents, Auto Help issues, etc.

2) What sort of instant functions does this DLL have?

This DLL has 2 types of functions in it. These are;

a) Color calculation functions
b) Spectrophotometers management functions

3) What are these funtions?

There are 40+6=48 instant subroutines as listed here. Each of these are ready to calculate values according to the parameters you supplied to them via your program source code as described later (see Function Description Section) of this document

COLOR CALCULATION FUNCTIONS: (40 pcs)

1- REFLACTANCE40_CAPITALXYZ_CALCULATE
2- CIELab_CAPITALXYZ_CALCULATE
3- CAPITALXYZ_LOWERCASExyz_CALCULATE
4- REFLECTANCE40_LOWERCASExyz_CALCULATE
5- REFLECTANCE40_CIELab_CALCULATE
6- CIELab_DELTA_CALCULATE
7- REFLECTANCE40_HUNTERLab_CALCULATE
8- HUNTER_DELTA_CALCULATE
9- REFLECTANCE40_CMC_CALCULATE
10- CIELab_CMC_CALCULATE
11- CMC_DELTA_CALCULATE
12- REFLECTANCE40_RGB_CALCULATE
13- CIELab_RGB_CALCULATE
14- CAPITALXYZ_RGB_CALCULATE
15- REFLECTANCE40_YELLOWNESS_CALCULATE

16- CAPITALXYZ_YELLOWNESS_CALCULATE
17- REFLECTANCE40_WHITENESS_CALCULATE
18- CAPITALXYZ_WHITENESS_CALCULATE
19- REFLECTANCE40_TINT_CALCULATE
20- CAPITALXYZ_TINT_CALCULATE
21- LOWERCASEXYZ_TINT_CALCULATE
22- REFLECTANCE40_GRAY_SCALE_CALCULATE
23- CIELab_GRAY_SCALE_CALCULATE
24- CIELabDELTAE_GRAY_SCALE_CALCULATE
25- REFLECTANCE40_OPACITY_CALCULATE
26- CAPITALXYZ_OPACITY_CALCULATE
27- REFLECTANCE40_STRENGTH_CALCULATE
28- REFLECTANCE40_BRIGHTNESS_CALCULATE
29- CAPITALXYZ_BRIGHTNESS_CALCULATE
30- REFLECTANCE40_METAMERISM_CALCULATE
31- CIELab_METAMERISM_CALCULATE
32- CIElab_METAMERISM_DIRECTION_CALCULATE
33- REFLECTANCE31_RECETE_CALCULATE
34- REFLECTANCE_KS_CONVERT
35- KS_REFLECTANCE_CONVERT
36- VERSION
37- CIELab_DELTA_ASSESSMENTS
38- RECIPE_REFLEKTANCE31_CALCULATE
39- RGB_CAPITALXYZ_CALCULATE
40- CAPITALXYZ_CIELab_CALCULATE

SPECTROPHOTOMETERS MANAGEMENT FUNCTIONS: (8 pcs)
1- SET_INSTRUMENT
2- INSTRUMENT_STATUS
3- SHOW
4- CALIBRATE_WHITE
5- CALIBRATE_BLACK
6- MEASURE
7- DOWNLOAD
8-UPLOAD

4) How these 48 functions can be called from my program?
or
How to include these functions into my source code?

This is the first in importance benefit which makes this application (DLL) invaluable. Different programming languages have different ways to do this. We will consider Visual Basic 6.0 for all descriptions rest of this manual. If you are using another languages (like Deplhi, DotNet, C, C++, etc) please refer its manuals about “Referencing DLLs”.

For Visual Basic 6.0, follow the easy 3 steps below

Step 1) Adding DLL into your project.
a) Copy DLL into Windows/System32 directory
b) Open Visual Basic 6.0 and select your existing project or create a new empty one
c) Select “Project/References” from Visual Basic 6.0 menu at the top.
d) Click “Browse”
e) Locate “ColorCalculationTreasure.dll” you copied and select it by clicking on it.

Step 2) Assigning a name to call all subroutines in the DLL
Add following two lines in module level of your project.

Public *AnyName1* As ColorCalculateTreasure
Public *AnyName2* As InstrumentCommandTreasure

"AnyName" is any name you will write according to variable name rules in VB6.

For example: Public ColorFunctions As ColorCalculateTreasure
Public InstrumentFunctions as InstrumentCommandTreasure

Step 3) Calling subroutines at anywhere you wish in your project

You can use following single line where you want to call a subroutine from DLL.

Set *AnyName* = New ColorCalculateTreasure

p.s: *AnyName in this statement must be same with step 2*

For example: Set ColorFunctions = New ColorCalculateTreasure

After this last step, when you write *AnyName* and as soon as press dot you will automatically get a scrollable list (in alphabetic order) similar to "Figure 1" showing all subroutines in DLL.

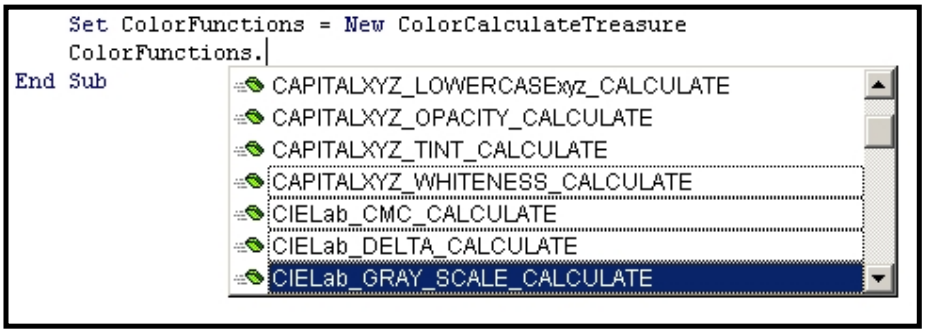
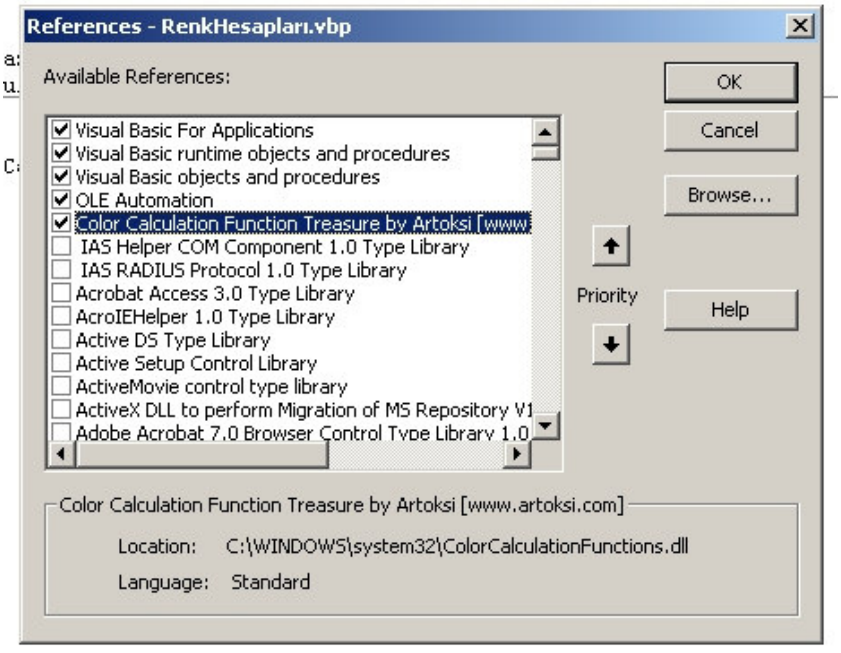


Figure 1: Sample screenshot

You are now ready to create professional color matching softwares according to your imagination border to fit your individual requirements or commercial purposes.

Troubleshooting: If you can not get the function list like above (Figure 1) as soon as pressing dot after the name you assigned please check followings.

Checkpoint1: Check references again from Project/References menu in VB6. "Color Calculation Function Treasure" by Artoksi [www.artoksi.com] should be marked/selected as follows.



Checkpoint2: Check the code you wrote:

At module level:
Public *AnyName* As ColorCalculateTreasure

At anywhere you wish to call a subroutine from the DLL
Set *AnyName* = New ColorCalculateTreasure
Anyname.

Be sure that you are using same variable name for *Anyname* in everywhere in the your source code

For example;

At module level:
Public ColorFunctions As ColorCalculateTreasure

At anywhere you wish to call a subroutine
Set ColorFunctions = New ColorCalculateTreasure
ColorFunctions.

5) How do I know which values/parameters should be transferred to each subroutines as input?
Or
How do I know the format of each subroutines to use them properly?

In VB6, when you select a function it automatically displays which parameters should be input and types of it. (See figure 1)

All the parameters in each function have been named carefully according to maximum understanding principals for users. We believe that you will not need to this manual in fact when you start to work with the DLL (ColorCalculationTreasures). Everything you need is going to be on the screen automatically as soon as you press dot sign following your assigned variable name while writing your code especially in Visual Basic.

If this is not enough clear then you can apply "Function Description Section" of this manual. This section has more information about "How to use them?" of each function of subroutines in the DLL.

6) Is there a sample software to let me understand better how all this above issues work in reality?

Yes, the package you have now has also a fully functional software written in Visual Basic language. This software is given to you with open source code. You can load it via your Visual Basic 6 and run. This helps you to understand better everything.

7) What do you ship when I buy this?

When you buy ColorCalculationTreasure.DLL you will have followings:

1) **ColorCalculationTreasure.DLL file**
2) **ColorCalculationTreasure.DLL user manual** **(downloadable from www.artoksi.com)**
3) **ColorCalculation demo software setup version** **(downloadable from www.artoksi.com)**
4) **ColorCalculation demo software open source code** **(downloadable from www.artoksi.com)**
5) **A dongle for calling DLL (USB)** **(should be shipped with a courier)**

!!! Item no 2 at above list represents this document

7) Why do you use dongle instead of software lock?

Using dongle (hardkey) has mutual benefits for you and us. Our benefit is to have more reliable way to protect our efforts against pricy and your benefit is to protect your softwares also automatically when you use this DLL in your code. You don't need to spend extra money and efforts to protect your product. Already done!

Because, each time you call DLL from your code, it will detect dongle and if it is not exist it will terminate itself. So, your software will be protected against pricy also.

SECTION1
Function Description Section
This chapter describes each subroutine in DLL

Readme first:

1) Each subroutine is named according to same format as follows to ensure better understanding for users during coding. Subroutines names say us what will be converted from and to.

First part of the name shows what will be converted from (inputs of subroutine)
Second part of the name shows what will be converted to (output of subroutine)

Example 1:
REFLECTANCE40_CAPITALXYZ_CALCULATE means that;

At a glance, we can easily understand what does this subroutine do. This subroutine converts reflectance values to X,Y,Z trsitimulus values.

First part is REFLECTANCE40 shows us this subroutine accepts 40 pieces reflectance values as input.

Second part is CAPITALXYZ shows us this subroutine will send back CAPITAL X, Y, Z values (known as tristimulus) as output

Example 2:
CIELab_CMC_CALCULATE means that;

At a glance, we can easily understand what does this subroutine do also. This subroutine converts CIELab to CMC color space.

First part is CIELab shows us this subroutine accepts L, a, b values in CIELab color space as input.

Second part is CMC shows us this subroutine will send back L, a, b, C, H values for CMC color space

2) The numbers suffix at the end of parts has special meaning as follows

40 means 40 pieces value between 360nm-750nm with 10nm interval
31 means 31 pieces value between 400-700 nm with 10nm interval

Example:
REFLECTANCE40 → Means an array with 40 elements containing reflectance values between 360nm-750nm with 10nm intervals.

REFLECTANCE31 → Means an array with 31 elements containing reflectance values between 400nm-700nm with 10nm intervals.

These naming rules are applied to all parameters in subroutines in DLL to help the user understand the function at a glance and use it immediately without reading the documents each time.

Subroutine no : 1

Subroutine name : REFLECTANCE40_CAPITALXYZ_CALCULATE

Description : Calculates X, Y, and Z tristimulus values from reflectance values of a color.

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) and get X, Y, Z tristimulus values.

General form : REFLACTANCE40_CAPITALXYZ_CALCULATE(Reflectance40() As Single, LightSource As String, Observer As Byte, Result_CAPITAL_X As Single, Result_CAPITAL_Y As Single, Result_CAPITAL_Z As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals

Example:
```
Refs(1)=12.345  'Reflectance value for 360nm
Refs(2)=11.395  'Reflectance value for 370nm
      .
      .
      .
Refs(39)=32.123 'Reflectance value for 740nm
Refs(40)=36.523 'Reflectance value for 750nm
```

Parameter 2 -> LightSource As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate X, Y, Z tristimulus values.

Parameter 3 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used to calculate X, Y, Z tristimulus values.

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_CAPITAL_X As Single

Returns X tristimulus value

Parameter 2 -> Result_CAPITAL_Y As Single

Returns Y tristimulus value

Parameter 3 -> Result_CAPITAL_Z As Single

Returns Y tristimulus value

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single

or

Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100

Examples :

*Anyname.*REFLACTANCE40_CAPITALXYZ_CALCULATE(Array40(), ”D65”, 10, ResultX, ResultY, ResultZ)

X, Y, Z tristimulus values under illuminant Daylight 6500K and 10 degree observer of the color in Array40() will be returned via variables ResultX, ResultY and ResultZ consecutively.

*Anyname.*REFLACTANCE40_CAPITALXYZ_CALCULATE(RefArray40(), ”CWF”, 2, TristX, TristY, TristZ)

X, Y, Z tristimulus values under illuminant Coolwhite 4150K and 2 degree observer of the color in Array40() will be returned via variables TristX, TristY and TristZ consecutively.

Subroutine no : 2

Subroutine name : CIELab_CAPITALXYZ_CALCULATE

Description : Calculates X, Y, and Z tristimulus values from CIE L, a and b values of a color.

Purpose : Send CIE L, a, and b values and get X, Y, Z tristimulus values

General form : CIELab_CAPITALXYZ_CALCULATE(CIE_L As Single, CIE_a As Single, CIE_b As Single, LightSource As String, Observer As Byte, Result_CAPITAL_X As Single, Result_CAPITAL_Y As Single, Result_CAPITAL_Z As Single)

Input parameters:

Parameter 1 -> CIE_L As Single
L (Lightness) value at CIELab colorspace of the color.

Parameter 2 -> CIE_a As Single
a (Red-Green) value at CIELab colorspace of the color.

Parameter 3 -> CIE_L As Single
b (Blue-Yellow) value at CIELab colorspace of the color.

Parameter 4 -> LightSource As String
One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate X, Y, Z tristimulus values.

Parameter 3 -> Observer as Byte
One of the options below to tell the subroutine which observer for the specified light source will be used to calculate X, Y, Z tristimulus values.

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_CAPITAL_X
Returns X tristimulus value

Parameter 2 -> Result_CAPITAL_Y
Returns Y tristimulus value

Parameter 3 -> Result_CAPITAL_Z
Returns Z tristimulus value

Notes : As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

*Anyname.*CIELab_CAPITALXYZ_CALCULATE(25.2,14.65,3.77, “D65”,10, TX,TY,TZ)

X,Y and Z tristimulus values for the color specified with L=25.2, a=14.65 and b=3.77 under D65 light source and 10 degree observer will be returned via variables TX,TY and TZ consecutively.

Anyname. CIELab_CAPITALXYZ_CALCULATE (61.17, 75.09, 17.55, “CWF”, 2, Rslt_X, Rslt_Y,Rslt_Z)

X,Y and Z tristimulus values for the color specified with L=61.17, a=75.09 and b=17.55 under CWF (Coolwhite) light source and 2 degree observer will be returned via variables Rslt_X,Rslt_Y and Rslt_Z consecutively.

Subroutine no : 3

Subroutine name : CAPITALXYZ_LOWERCASExyz_CALCULATE

Description : Calculates x, y, and z chromaticity values from X, Y, Z tristimulus values of a color.

Purpose : Send X, Y, Z tristimulus values and get x, y, z chromaticity values.

General form : CAPITALXYZ_LOWERCASExyz_CALCULATE(CAPITAL_X As Single, CAPITAL_Y As Single, CAPITAL_Z As Single, Result_Lowercase_x As Single, Result_Lowercase_y As Single, Result_Lowercase_z As Single)

Input parameters:

Parameter 1 -> CAPITAL_X As Single
Tristimulus X value of the color

Parameter 2 -> CAPITAL_Y As Single
Tristimulus Y value of the color

Parameter 3 -> CAPITAL_Z As Single
Tristimulus Z value of the color

Output parameters:

Parameter 1 -> Result_Lowercase_x
Returns x chromaticity value

Parameter 2 -> Result_Lowercase_y
Returns y chromaticity value

Parameter 3 -> Result_Lowercase_z
Returns z chromaticity value

Notes : As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

*Anyname.*CAPITALXYZ_LOWERCASExyz_CALCULATE (12.234, 5.12, 7.35, Resultx, Resulty, Resultz)

x, y, z chromaticity values for the color specified with X, Y, Z tristimulus values will be returned via variables Resultx, Resulty and Resultz consecutively.

*Anyname.*CAPITALXYZ_LOWERCASExyz_CALCULATE (61.17, 75.09, 17.55, Chrtst_x, Chrtst_y, Chrtst_z)

x, y, z chromaticity values for the color specified with X, Y, Z tristimulus values will be returned via variables Chrtst_x, Chrtst_y and Chrtst_z consecutively.

Subroutine no : 4

Subroutine name : REFLECTANCE40_LOWERCASExyz_CALCULATE

Description : Calculates lowercase x, y, and z chromaticity values from reflectance values of a color.

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) and get x, y, z tristimulus values.

General form : REFLECTANCE40_LOWERCASExyz_CALCULATE(Reflectance40() As Single, LightSource As String, Observer As Byte, Result_Lowercase_x As Single, Result_Lowercase_y As Single, Result_Lowercase_z As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals.

```
Example: Refs(1)=12.345  'Reflectance value for 360nm
         Refs(2)=11.395  'Reflectance value for 370nm
                  .
                  .
                  .
         Refs(39)=32.123 'Reflectance value for 740nm
         Refs(40)=36.523 'Reflectance value for 750nm
```

Parameter 2 -> LightSource As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" At the end of this document to tell the subroutine which light source will be used (simulated) to calculate x, y, z chromaticity values.

Parameter 3 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used to calculate X, Y, Z tristimulus values.

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_Lowercase_x

Returns x chromaticity value

Parameter 2 -> Result_Lowercase_y

Returns y chromaticity value

Parameter 3 -> Result_Lowercase_z

Returns z chromaticity value

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100

Examples :

*Anyname.*REFLACTANCE40_LOWERCASExyz_CALCULATE(Array40(),"D65",10, Resultx, Resulty, Resultz)

x, y, z chromaticity values under illuminant Daylight 6500K and 10 degree observer of the color in Array40() will be returned via variables Resultx, Resulty and Resultz consecutively.

*Anyname.*REFLACTANCE40_CAPITALXYZ_CALCULATE(RefArray40(),"CWF",2, Crmtctx, Crmtcty, Crmtctz)

x, y, z chromaticity values under illuminant Coolwhite 4150K and 2 degree observer of the color in Array40() will be returned via variables Crmtctx, Crmtcty and Crmtctz cosecutively.

Subroutine no : 5

Subroutine name : REFLECTANCE40_CIELab_CALCULATE

Description : Calculates L, a, b values in CIE coordinate system (CIELab color space) from reflectance values of a color.

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) and get L, a and b values for CIE coordinate (CIELab color space) system

General form : REFLECTANCE40_CIELab_CALCULATE(Reflectance40() As Single, LightSource As String, Observer As Byte, Result_CIE_L As Single, Result_CIE_a As Single, Result_CIE_b As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals.

```
Example: Refs(1)=12.345 'Reflectance value for 360nm
         Refs(2)=11.395 'Reflectance value for 370nm
                  .
                  .
                  .
         Refs(39)=32.123 'Reflectance value for 740nm
         Refs(40)=36.523 'Reflectance value for 750nm
```

Parameter 2 -> LightSource As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate CIE L, a and b values.

Parameter 3 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used to calculate CIE L, a and b values.

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_CIE_L As Single

Returns Lightness value "L" in CIE coordinate system L value is between 0 to 100. 100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 2 -> Result_CIE_a As Single

Returns Greenish/Redish value "a" in CIE coordinate system.
Minus values represent green zone
Plus values represent red zone

Parameter 3 -> Result_CIE_b As Single

Returns Yellowish/Blueish value "b" in CIE coordinate system
Minus values represent blue zone
Plus values represent yellow zone

Notes **: 1)** As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100.

3) Output parameters Result_CIE_a and Result_CIE_b may return plus or minus value so be sure that variable has been defined properly in your code.

Examples **:**

*Anyname.*REFLECTANCE40_CIELab_CALCULATE(RedTT10(), "D65", 10, Result_L, Result_a, Result_b)

L, a, b values of CIELab color space under illuminant Daylight 6500K and 10 degree observer of the color in Array RedTT10 will be returned via variables Result_L, Result_a and Result_b consecutively.

*Anyname.*REFLECTANCE40_CIELab_CALCULATE(Sample1(), "U30", 10, CIELab_L, CIELab_a, CIELab_b)

L, a, b values of CIELab color space under illuminant Ultralume 30 (3000K) and 10 degree observer of the color in array Sample1 will be returned via variables CIELab_L, CIELab _a and CIELab _b consecutively.

Subroutine no : 6

Subroutine name : CIELab_DELTA_CALCULATE

Description : Calculates Delta L, DELTA a, DELTA b and DELTA E values in CIE coordinate system from two sets of color (Sample/Standard vs Trial) reflectance values.

Purpose : Send L (CIELab), a (CIELab) and b (CIELab) values for both sample1 and sample2 and get DELTA L, DELTA a, DELTA b and DELTA E values in CIE coordinate system

General form : CIELab_DELTA_CALCULATE(CIELab_L_Std As Single, CIELab_a_Std As Single, CIELab_b_Std As Single, CIELab_L_Trial As Single, CIELab_a_Trial As Single, CIELab_b_Trial As Single, Result_CIELab_DELTAL As Single, Result_CIELAB_DELTAa As Single, Result_CIELab_DELTAb As Single, Result_CIELab_DELTAE As Single)

Input parameters:

Parameter 1 -> CIELab_L_Std As Single
Lightness value "L" [CIELab color space] of the standard color
100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 2 -> CIELab_a_Std As Single
Greenish/Redish value "a" [CIELab color space] of the standard color
Minus values represent green zone
Plus values represent red zone

Parameter 3 -> CIELab_b_Std As Single
Yellowish/Blueish value "b" value [CIELab color space] of the standard color
Minus values represent blue zone
Plus values represent yellow zone

Parameter 4 -> CIELab_L_Trial As Single
Lightness value "L" [CIELab color space] of the trail (sample) color
100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 5 -> CIELab_a_Trial As Single
Greenish/Redish value "a" [CIELab color space] of the trail (sample) color
Minus values represent green zone
Plus values represent red zone

Parameter 6 -> CIELab_b_Trial As Single
Yellowish/Blueish value "b" [CIELab color space] of the trail (sample) color
Minus values represent blue zone
Plus values represent yellow zone

Output parameters:

Parameter 1 -> Result_CIELab_DELTAL As Single
Returns [CIELab color space] DELTA_L (Lightness difference) value between two colors specified.

Parameter 2 -> Result_CIELab_DELTAa As Single
Returns [CIELab color space] DELTA_a (Greenish-Redish difference) value between two colors specified.

Parameter 3 -> Result_CIELab_DELTAb As Single

Returns [CIELab color space] DELTA_b (Yellowish/Blueish difference) value between two colors specified.

Parameter 4 -> Result_CIELab_DELTAE As Single

Returns [CIELab color space] DELTA_E (perceived difference) value between two colors specified.

Notes : 1) As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single

or

Public *VariableName* as Single

2) Output parameters Result_CIELab_DELTAL, CIELab_DELTAa and CIELab_DELTAb may return plus or minus value so be sure that variable has been defined properly in your code.

Examples :

Anyname. CIELab_DELTA_CALCULATE(11.12, -3.06, 2.17, 12.09, -3.21, 1.16, DiffL, Diffa, Diffb, DiffE)

The differences between standard color (L=11.12, a=-3.06, b=2.17) and trial color (L=12.09, a=-3.21, b=1.16) will be calculated and differences between these two colors will be output via variables DiffL, Diffa, Diffb and DiffE

Anyname. CIELab_DELTA_CALCULATE(71.12, 13.46, 21.17, 70.09, 13.30, 21.10, CIELab_DL, CIELab_Da, CIELab_Db, CIELab_Da)

The differences between standard color (L=71.12, a=13.46, b=21.17) and trial color (L=70.09, a=-13.30, b=21.10) will be calculated and differences between these two colors will be output via variables CIELab_DL, CIELab_Da, CIELab_Db and CIELab_DE.

Subroutine no : *7*

Subroutine name : REFLECTANCE40_HUNTERLab_CALCULATE

Description : Calculates L, a, b values in HunterLab color space from reflectance values of a color.

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) and get L, a and b values for HunterLab color space.

General form : REFLECTANCE40_HUNTERLab_CALCULATE(Reflectance40() As Single, LightSource As String, Observer As Byte, Result_Hunter_L As Single, Result_Hunter_a As Single, Result_Hunter_b As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals.

```
Example: Refs(1)=12.345 'Reflectance value for 360nm
         Refs(2)=11.395 'Reflectance value for 370nm
                  .
                  .
                  .
         Refs(39)=32.123 'Reflectance value for 740nm
         Refs(40)=36.523 'Reflectance value for 750nm
```

Parameter 2 -> LightSource As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate Hunter L, Hunter a and Hunter b values.

Parameter 3 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used to calculate Hunter L, a and b values.

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_Hunter_L As Single

Returns Lightness value "L" in Hunter coordinate system
L value is between 0 and 100. 100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 2 -> Result_Hunter_a As Single

Returns Greenish/Redish value "a" in Hunter coordinate system
Minus values represent green zone
Plus values represent red zone

Parameter 3 -> Result_Hunter_b As Single

Returns Yellowish/Blueish value "b" in Hunter coordinate system
Minus values represent blue zone
Plus values represent yellow zone

Notes **: 1)** As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 and 100.

3) Output parameters Result_Hunter_a and Result_Hunter_b may return plus or minus value so be sure that variable has been defined properly in your code.

Examples **:**

Anyname. REFLECTANCE40_HUNTERLab_CALCULATE(RedTT10(), "D65", 10, Result_L, Result_a, Result_b)

L, a, b values of HunterLab color space under illuminant Daylight 6500K and 10 degree observer of the color in Array RedTT10 will be returned via variables Result_L, Result_a and Result_b consecutively.

*Anyname.*REFLECTANCE40_CIELab_CALCULATE(Sample1(), "U30", 10, Hunter_L, Hunter_a, Hunter_b)

L, a, b values of HunterLab color space under illuminant Ultralume 30 (3000K) and 10 degree observer of the color in array Sample1 will be returned via variables HunterLab_L, Hunterlab _a and HunterLab _b consecutively.

Subroutine no : 8

Subroutine name : HUNTER_DELTA_CALCULATE

Description : Calculates Delta L, DELTA a, DELTA b and DELTA E values in HunterLab color Space from two sets of color (Sample/Standard vs Trial) reflectance values.

Purpose : Send L (HunterLab), a (HunterLab) and b (HunterLab) values for both sample1 and sample2 and get DELTA L, DELTA a, DELTA b and DELTA E values in HunterLab color space

General form : HUNTER_DELTA_CALCULATE(Hunter_L_Std As Single, Hunter_a_Std As Single, Hunter_b_Std As Single, Hunter_L_Trial As Single, Hunter_a_Trial As Single, Hunter_b_Trial As Single, Result_HUNTER_DELTAL As Single, Result_HUNTER_DELTAa As Single, Result_HUNTER_DELTAb As Single, Result_HUNTER_DELTAE as Single)

Input parameters:

Parameter 1 -> Hunter_L_Std As Single
L value [HunterLab color space] of the standard (master) color

Parameter 2 -> Hunter _a_Std As Single
a value [HunterLab color space] of the standard (master) color

Parameter 3 -> Hunter _b_Std As Single
b value [HunterLab color space] of the standard (master) color

Parameter 4 -> Hunter _L_Trial As Single
L value [HunterLab color space] of the trial (sample) color

Parameter 5 -> Hunter _a_Trial As Single
a value [HunterLab color space] of the trial (sample) color

Parameter 6 -> Hunter _b_Trial As Single
b value [HunterLab color space] of the trial (sample) color

Output parameters:

Parameter 1 -> Result_Hunter_DELTAL As Single
Returns [HunterLab color space] DELTA_L (Lightness difference) value between two colors specified.

Parameter 2 -> Result_ Hunter_DELTAa As Single
Returns [HunterLab color space] DELTA_a (Greenish-Redish difference) value between two colors specified.

Parameter 3 -> Result_ Hunter_DELTAb As Single
Returns [HunterLab color space] DELTA_b (Yellowish/Blueish difference) value between two colors specified.

Parameter 4 -> Result_ Hunter_DELTAE As Single
Returns [HunterLab color space] DELTA_E (perceived difference) value between two colors specified.

Notes **: 1)** As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Output parameters Result_Hunter_DELTAL, Result_Hunter_DELTAa and Result_Hunter_DELTAb may return plus or minus value so be sure that variable has been defined properly in your code.

Examples **:**

*Anyname.*HUNTER_DELTA_CALCULATE(11.12, -3.06, 2.17, 12.09, -3.21, 1.16, DiffL, Diffa, Diffb, DiffE)

The differences in Hunter color space between standard color (L=11.12, a=-3.06, b=2.17) and trial color (L=12.09, a=-3.21, b=1.16) will be calculated and differences between these two colors will be output via variables DiffL, Diffa, Diffb and DiffE

*Anyname.*HUNTER_DELTA_CALCULATE(71.12, 13.46, 21.17, 70.09, 13.30, 21.10, CIELab_DL, HUnterLab_Da, HunterLab_Db, HunterLab_Da)

The differences in Hunter color space between standard color (L=71.12, a=13.46, b=21.17) and trial color (L=70.09, a=-13.30, b=21.10) will be calculated and differences between these two colors will be output via variables HunterLab_DL, HunterLab_Da, HunterLab_Db and HunterLab_DE.

Subroutine no : 9

Subroutine name : REFLECTANCE40_CMC_CALCULATE

Description : Calculates L, a, b, C (chroma) and H (hue) values in CMC color space from reflectance values of a color.

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) and get L, a, b, C and H values for CMC color space.

General form : REFLECTANCE40_CMC_CALCULATE(Reflectance40() As Single, LightSource As String, Observer As Byte, Result_CMC_L As Single, Result_CMC_a As Single, Result_CMC_b As Single, Result_CMC_C As Single, Result_CMC_H As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals.

```
Example: Refs(1)=12.345 'Reflectance value for 360nm
         Refs(2)=11.395 'Reflectance value for 370nm
                  .
                  .
                  .
         Refs(39)=32.123 'Reflectance value for 740nm
         Refs(40)=36.523 'Reflectance value for 750nm
```

Parameter 2 -> LightSource As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate CMC L, CMC a, CMC b, CMC C and CMC H values.

Parameter 3 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used to calculate Hunter L, a and b values.

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_CMC_L As Single

Returns Lightness value "L" in CMC coordinate system
L value is between 0 and 100. 100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 2 -> Result_CMC_a As Single

Returns Greenish/Redish value "a" in CMC coordinate system
Minus values represent green zone
Plus values represent red zone

Parameter 3 -> Result_CMC_b As Single

Returns Yellowish/Blueish value "b" in CMC coordinate system
Minus values represent blue zone
Plus values represent yellow zone

Parameter 4 -> Result_CMC_C As Single
Returns chroma (Duller-Sharper) value "C" in CMC coordinate system

Parameter 5 -> Result_CMC_H As Single
Returns Hue (Pureness) value "H" in CMC coordinate system

Notes **: 1)** As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 and 100.

3) Output parameters Result_Result_CMC_a, Result_CMC_b and Result_CMC_H may return plus or minus value so be sure that variable has been defined properly in your code.

Examples **:**

*Anyname.*REFLECTANCE40_CMC_CALCULATE(RedTT10(), "D65", 10, Result_L, Result_a, Result_b, Result_C, Result_H)

L, a, b, C and H values of CMC color space under illuminant Daylight 6500K and 10 degree observer of the color in Array RedTT10 will be returned via variables Result_L, Result_a, Result_b, Result_C and Result_H consecutively.

*Anyname.*REFLECTANCE40_CMC_CALCULATE(Sample1(), "U30", 10, CMC_L, CMC_a, CMC_b, CMC_C, CMC_H)

L, a, b, C, H values of CMC color space under illuminant Ultralume 30 (3000K) and 10 degree observer of the color in array Sample1 will be returned via variables CMC_L, CMC_a, CMC_b, CMC_C and CMC_H consecutively.

Subroutine no : 10

Subroutine name : CIELab_CMC_CALCULATE

Description : Calculates L, a, b, C (chroma) and H (hue) values in CMC color space from CIELab L, a, b values of a color.

Purpose : Send L, a, b values for CIELab color space and get L, a, b, C and H values for CMC color space.

General form : CIELab_CMC_CALCULATE(CIELab_L As Single, CIELab_a As Single, CIELab_b As Single, Result_CMC_L As Single, Result_CMC_a As Single, Result_CMC_b As Single, Result_CMC_C As Single, Result_CMC_H As Single)

Input parameters:

Parameter 1 -> CIELab_L As Single
L (Lightness) value in CIELab color space
100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 2 -> CIELab_a As Single
a (Greenish/Redish) value in CIELab color space
Minus values represent green zone
Plus values represent red zone

Parameter 3 -> CIELab_b_Std As Single
b (Yellowish/Blueish) value in CIELab color space
Minus values represent blue zone
Plus values represent yellow zone

Output parameters:

Parameter 1 -> Result_CMC_L As Single
Returns Lightness value "L" in CMC coordinate system
L value is between 0 and 100. 100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 2 -> Result_CMC_a As Single
Returns Greenish/Redish value "a" in CMC coordinate system
Minus values represent green zone
Plus values represent red zone

Parameter 3 -> Result_CMC_b As Single
Returns Yellowish/Blueish value "b" in CMC coordinate system
Minus values represent blue zone
Plus values represent yellow zone

Parameter 4 -> Result_CMC_C As Single
Returns chroma value "C" (Duller-Sharper) in CMC coordinate system

Parameter 5 -> Result_CMC_H As Single
Returns Hue value "H" in (Pureness) CMC coordinate system

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 and 100.

3) Output parameters Result_Result_CMC_a, Result_CMC_b and Result_CMC_H may return plus or minus value so be sure that variable has been defined properly in your code.

Examples :

Anyname. CIELab_CMC_CALCULATE (44,23,0.91,7.56,Result_L, Result_a, Result_b, Result_C, Result_H)

L, a, b, C and H values of CMC color space will be returned via variables Result_L, Result_a, Result_b, Result_C and Result_H consecutively.

Anyname. CIELab_CMC_CALCULATE (71.65, 1.13, 7.01, CMC_L, CMC_a, CMC_b, CMC_C, CMC_H)

L, a, b, C, H values of CMC color space will be returned via variables CMC_L, CMC_a, CMC_b, CMC_C and CMC_H consecutively.

Subroutine no : 11

Subroutine name : CMC_DELTA_CALCULATE

Description : Calculates Delta L, DELTA a, DELTA b, DELTA C, DELTA H and DELTA E values in CMC color space from two sets of color (Sample/Standard vs Trial) specifed with CMC L, CMC a, CMC b, CMC C and CMC H values.

Purpose : Send L (CMC), a (CMC), b (CMC), C (CMC) and H (CMC) values for both sample1 and sample2 and get DELTA L, DELTA a, DELTA b and DELTA E values in CMC color space

General form : CMC_DELTA_CALCULATE(CMC_L_Std As Single, CMC_a_Std As Single, CMC_b_Std As Single, CMC_C_Std As Single, CMC_H_Std As Single, CMC_L_Trial As Single, CMC_a_Trial As Single, CMC_b_Trial As Single, CMC_C_Trial As Single, CMC_H_Trial As Single, LC_Ratio As Byte, Result_CMC_DELTAL As Single, Result_CMC_DELTAa As Single, Result_CMC_DELTAb As Single, Result_CMC_DELTAC As Single, Result_CMC_DELTAh1 As Single, Result_CMC_DELTAH As Single, Result_CMC_DELTAE As Single)

Input parameters:

Parameter 1 -> CMC_L_Std As Single
Lightness value "L" [CMC color space] of the standard color
100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 2 -> CMC_a_Std As Single
Greenish/Redish value "a" [CMC color space] of the standard color
Minus values represent green zone
Plus values represent red zone

Parameter 3 -> CMC_b_Std As Single
Yellowish/Blueish value "b" value [CMC color space] of the standard color
Minus values represent blue zone
Plus values represent yellow zone

Parameter 4 -> CMC_C_Std As Single
Chroma value (Duller-Sharper) "C" [CMC color space] of the standard color

Parameter 5 -> CMC_H_Std As Single
Hue value "H" (Pureness) [CMC color space] of the standard color

Parameter 6 -> CMC_L_Trial As Single
Lightness value "L" [CMC color space] of the trial (sample) color
100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 7 -> CMC_a_Trial As Single
Greenish/Redish value "a" [CMC color space] of the trial (sample) color
Minus values represent green zone
Plus values represent red zone

Parameter 8 -> CMC_b_Trial As Single
Yellowish/Blueish value "b" value [CMC color space] of the trial (sample) color
Minus values represent blue zone and plus values represent yellow zone

Parameter 9 -> CMC_C_Trial As Single
Chroma (Duller-Sharper) value "C" [CMC color space] of the trial (sample) color

Parameter 10-> CMC_H_Trial As Single
Hue (Pureness) value "H" [CMC color space] of the trial (sample) color

Parameter 11-> LC_Ratio As Byte
Acceptability/Perceptability factor (L:c).
"1" for L:c=1:1
"2" for L:c=2:1

Output parameters:

Parameter 1 -> Result_CMC_DELTAL As Single
Returns [CMC color space] DELTAL (Lightness difference) value between two colors specified.

Parameter 2 -> Result_CMC_DELTAa As Single
Returns [CMC color space] DELTAa (Greenish-Redish difference) value between two colors specified.

Parameter 3 -> Result_CMC_DELTAb As Single
Returns [CMC color space] DELTAb (Yellowish/Blueish difference) value between two colors specified.

Parameter 4 -> Result_CMC_DELTAC As Single
Returns [CMC color space] DELTAC (Chroma difference) value between two colors specified.

Parameter 5 -> Result_CMC_DELTAh1 As Single
Returns [CMC color space] DELTAh1 (Mathematical difference of H) value between two colors specified.

Parameter 6 -> Result_CMC_DELTAH As Single
Returns [CMC color space] DELTAH (Calculated difference of H) value between two colors specified.

Parameter 6 -> Result_CMC_DELTAE As Single
Returns [CMC color space] DELTAE (total difference) value between two colors specified.

Notes **: 1)** As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Output parameters Result_CMC_DELTAL, Result_CMC_DELTAa, Result_CMC_DELTAb and Result_CMC_DELTAC may return plus or minus values. Please be sure these parameters type is declared properly in your code.

Examples :

*Anyname.*CMC_DELTA_CALCULATE(11.12, -3.06, 2.17, 12.09, -3.21, 11.16, -3.82, 1.99, 11.99, -3.18, 2, DiffL, Diffa, Diffb, DiffC, Diffh1, DiffH, DiffE)

The differences between standard color (L=11.12, a=-3.06, b=2.17, C=12.09, H=-3.21) and trial color (L=11.16, a=-3.82, b=1.99, C=11.99, H=-3.18) will be calculated according to l:c ratio 2 and differences between these two colors will be output via variables DiffL, Diffa, Diffb, DiffC, Diffh1, DiffH and DiffE

*Anyname.*CMC_DELTA_CALCULATE (71.12, 13.46, 21.17, 70.09, 134.30, 70.92, 14.56, 19.92, 71.19, 138.71, 1, CMC_DL, CMC_Da, CMC_Db, CMC_DC, CMC_Dh1, CMC_DH, CMC_DE)

The differences between standard color (L=71.12, a=13.46, b=21.17, C=70.09, H=134.30) and trial color (L=70.92, a=14.56, b=19.92, C=71.19, H=138.71) will be calculated according to l:c ration 1 and differences between these two colors will be output via variables CMC_DL, CMC_Da, CMC_Db, CMC_DC, CMC_Dh1, CMC_DH and CMC_DE.

Subroutine no : 12

Subroutine name : REFLECTANCE40_RGB_CALCULATE

Description : Calculates R (red) G (Green) B (blue) values from the color given in array REFLECTANCE40.

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) and get R, G, B values.

General form : REFLECTANCE40_RGB_CALCULATE(Reflectance40() As Single, Result_Red As Single, Result_Green As Single, Result_Blue As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals.

Example: Refs(1)=12.345 'Reflectance value for 360nm
Refs(2)=11.395 'Reflectance value for 370nm
.
.
.
Refs(39)=32.123 'Reflectance value for 740nm
Refs(40)=36.523 'Reflectance value for 750nm

Output parameters:

Parameter 1 -> Result_Red As Single

Returns a red value between 0-255 corresponding to the color given in array REFLECTANCE40() as input parameter.

Parameter 2 -> Result_Green As Single

Returns a green value between 0-255 corresponding to the color given in array REFLECTANCE40() as input parameter.

Parameter 3 -> Result_Blue As Single

Returns a blue value between 0-255 corresponding to the color given in array REFLECTANCE40() as input parameter.

Notes : –

Examples :

*Anyname.*REFLECTANCE40_RGB_CALCULATE (ColorrefArray(), Red1, Green1, Blue1)

Three values between 0 and 255 will be returned via output variables Red1, Green1, Blue1 for the color given in array ColorrefArray() as input

*Anyname.*REFLECTANCE40_RGB_CALCULATE (CustomerSample(), cR, cG, cB)

Three values between 0 and 255 will be returned via output variables cR, cG, cB for the color given in array CustomerSample().

Subroutine no : 13

Subroutine name : CIELab_RGB_CALCULATE

Description : Calculates R (red) G (Green) B (Blue) values from the color given with L, a and b values at CIELab color space under a certain light source and observer.

Purpose : Send L, a and b values at CIELab color space and get R, G, B values of it.

General form : CIELab_RGB_CALCULATE(CIE_L As Single, CIE_a As Single, CIE_b As Single, LightSource As String, Observer As Byte, Result_Red As Single, Result_Green As Single, Result_Blue As Single)

Input parameters:

Parameter 1 -> CIE_L As Single
L (Lightness) value at CIELab colorspace of the color.

Parameter 2 -> CIE_a As Single
a (Red-Green) value at CIELab colorspace of the color.

Parameter 3 -> CIE_L As Single
b (Blue-Yellow) value at CIELab colorspace of the color.

Parameter 4 -> LightSource As String
One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate X, Y, Z tristimulus values.

Parameter 3 -> Observer as Byte
One of the options below to tell the subroutine which observer for the specified light source will be used to calculate X, Y, Z tristimulus values.

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_Red As Single
Returns a red value between 0-255 corresponding to the color given with L, a and b CIELab values under a specified light source and observer.

Parameter 2 -> Result_Green As Single
Returns a green value between 0-255 corresponding to the color given with with L, a and b CIELab values under a specified light source and observer.

Parameter 3 -> Result_Blue As Single
Returns a blue value between 0-255 corresponding to the color given with with L, a and b CIELab values under a specified light source and observer.

Notes : --

Examples :

*Anyname.*CIELab_RGB_CALCULATE (11.12, 7.34, 21.89, "D65", 10, Red1, Green1, Blue1)

Three values between 0 and 255 will be returned via output variables Red1, Green1, Blue1 for the

color given with values L=11.12, a=7.34 and b=21.89 at CIELab color space under D65 light source and 10 degree observer.

Anyname.CIELab_RGB_CALCULATE (30.06, 9.17, “CWF”, 2, 13.13, cR, cG, cB)

Three values between 0 and 255 will be returned via output variables cR, cG, cB for the color given with values L=30.06, a=9.17 and b=13.13 at CIELab color space under CWF (Coolwhite) light source and 2 degree observer.

Subroutine no : 14

Subroutine name : CAPITALXYZ_RGB_CALCULATE

Description : Calculates R (red) G (Green) B (blue) values from the color given with X, Y, Z tristimulus values

Purpose : Send X, Y and Z tristimulus values of a color and get R, G, B values of it.

General form : CAPITALXYZ_RGB_CALCULATE (CAPITAL_X As Single, CAPITAL_Y As Single, CAPITAL_Z As Single, Result_Red As Single, Result_Green As Single, Result_Blue As Single)

Input parameters:

Parameter 1 -> CAPITAL_X As Single
X tristimulus value of the color

Parameter 2 -> CAPITAL_Y As Single
Y tristimulus value of the color

Parameter 3 -> CAPITAL_Z As Single
Z tristimulus value of the color

Output parameters:

Parameter 1 -> Result_Red As Single
Returns a red value between 0-255 corresponding to the color given with X, Y, Z tristimulus values

Parameter 2 -> Result_Green As Single
Returns a green value between 0-255 corresponding to the color given with X, Y, Z tristimulus values

Parameter 3 -> Result_Blue As Single
Returns a blue value between 0-255 corresponding to the color given with X, Y, Z tristimulus values

Notes : --

Examples :

Anyname. CAPITALXYZ_RGB_CALCULATE (11.12, 7.34, 21.89, Red1, Green1, Blue1)

Three values between 0 and 255 will be returned via output variables Red1, Green1, Blue1 for the color given with tristimulus values X=11.12, Y=7.34 and Z=21.89

*Anyname.*REFLECTANCE40_RGB_CALCULATE (30.06, 9.17, 13.13, cR, cG, cB)

Three values between 0 and 255 will be returned via output variables cR, cG, cB for the color given with tristimulus values X=30.06, Y=9.17 and Z=13.13.

Subroutine no : 15

Subroutine name : REFLECTANCE40_YELLOWNESS_CALCULATE

Description : Calculates ASTMD1925, ASTME313 and BASFGB yellowness indices for the color given in array REFLECTANCE40.

Purpose : Send reflectance values between 360nm-750nm with 10nm intervals (40 pcs) and get three type yellowness indices (ASTMD1925, ASTME313 and BASFGB)

General form : REFLECTANCE40_YELLOWNESS_CALCULATE(Reflectance40() As Single, Result_ASTMD1925 As Single, Result_ASTME313 As Single, Result_BASFGB As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals.

```
Example: Refs(1)=12.345 'Reflectance value for 360nm
         Refs(2)=11.395 'Reflectance value for 370nm
                  .
                  .
                  .
         Refs(39)=32.123 'Reflectance value for 740nm
         Refs(40)=36.523 'Reflectance value for 750nm
```

Output parameters:

Parameter 1 -> Result_ASTMD1925 As Single

Returns ASTMD1925 yellowness indice for the color given with reflectance values

Parameter 2 -> Result_ASTME313 As Single

Returns ASTME313 yellowness indice for the color given with reflectance values

Parameter 3 -> Result_BASFGB As Single

Returns BASFGB yellowness indice for the color given with reflectance values

Notes : 1) As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

Anyname.REFLECTANCE40_YELLOWNESS_CALCULATE (Sample1(), Yellowness1, Yelowness2, Yellowness3)

Three types of yellowness indice will be returned via output variables Yellowness1, Yellowness2 and Yellowness3 for the color given in REFLECTANCE40 array.

Anyname.REFLECTANCE40_YELLOWNESS_CALCULATE (Sample1(), Yellowness1925, Yelowness313, YellownessBASF)

Three types of yellowness indice will be returned via output variables Yellowness1925, Yellowness313 and YellownessBASF for the color given in REFLECTANCE40 array.

Subroutine no : 16

Subroutine name : CAPITALXYZ_YELLOWNESS_CALCULATE

Description : Calculates ASTMD1925, ASTME313 and BASFGB yellowness indices for the color given with X, Y and Z tristimulus values.

Purpose : Send X, Y and Z trisitimulus values under illuminant C and observer 2 degree and get three type yellowness indices (ASTMD1925, ASTME313 and Result_BASFGB)

General form : CAPITALXYZ_YELLOWNESS_CALCULATE(CAPITAL_X_C2 As Single, CAPITAL_Y_C2 As Single, CAPITAL_Z_C2 As Single, Result_ASTMD1925 As Single, Result_ASTME313 As Single, Result_BASFGB As Single)

Input parameters:

Parameter 1 -> CAPITAL_X_C2 As Single
Tristimulus X value under illuminant C and 2 degree observer

Parameter 2 -> CAPITAL_Y_C2 As Single
Tristimulus Y value under illuminant C and 2 degree observer

Parameter 3 -> CAPITAL_Z_C2 As Single
Tristimulus Z value under illuminant C and 2 degree observer

Output parameters:

Parameter 1 -> Result_ASTMD1925 As Single
Returns ASTMD1925 yellowness indice for the color given with tristimulus X, Y and Z values

Parameter 2 -> Result_ASTME313 As Single
Returns ASTME313 yellowness indice for the color given with with tristimulus X, Y and Z values

Parameter 3 -> Result_BASFGB As Single
Returns BASFGB yellowness indice for the color given with tristimulus X, Y and Z values

Notes : 1) As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

Anyname.CAPITALXYZ_YELLOWNESS_CALCULATE (11.34, 21.33, 42.57, Yellowness1, Yelowness2, Yellowness3)

Three types of yellowness indices will be returned via output variables Yellowness1, Yellowness2 and Yellowness3 for the color given with tristimulus X=11.34, Y=21.33 and Z=42.57 values

Anyname.CAPITALXYZ_YELLOWNESS_CALCULATE (41.32, 61.57, 19.21, Yellowness1925, Yelowness313, YellownessBASF)

Three types of yellowness indices will be returned via output variables Yellowness1925, Yellowness313 and YellownessBASF for the color given with tristimulus X=41.32, Y=61.57 and Z=19.21 values

Subroutine no : 17

Subroutine name : REFLECTANCE40_WHITENESS_CALCULATE

Description : Calculates one of the following whiteness index for the color given with reflectance values

Supported Whiteness index types
WI E313
WI CIE D65 - 2 degree
WI CIE D65 – 10 degree
WI Berger D65 - 2 degree
WI Berger D65 – 10 degree
WI Stensby
WI Taube
WI Ganz

Purpose : Send reflectance values between 360nm-750nm with 10nm intervals (40 pcs) and get whiteness indices

General form : REFLECTANCE40_WHITENESS_CALCULATE(Reflectance40() As Single, WhitenessType As Byte, Result_Whiteness As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals.

```
Example: Refs(1)=12.345 'Reflectance value for 360nm
         Refs(2)=11.395 'Reflectance value for 370nm
                  .
                  .
                  .
         Refs(39)=32.123 'Reflectance value for 740nm
         Refs(40)=36.523 'Reflectance value for 750nm
```

Parameter 2 -> WhitenessType As Byte

Any integer number according to following list to get the desired whiteness index standard.

Selection value	Meaning
1	E313 whiteness index
2	CIE whiteness index under illuminant D65 with 2 degree observer
3	CIE whiteness index under illuminant D65 with 10 degree observer
4	Berger whiteness index under illuminant D65 with 2 degree observer
5	Berger whiteness index under illuminant D65 with 10 degree observer
6	Stensby whiteness index
7	Taube whiteness index
8	Ganz whitenes index

Output parameters:

Parameter 1 -> Result_Whiteness As Single

Calculated Whiteness index according to color given in array REFLACTANCE40 and the selected WhhitenessType parameter.

Notes : **1)** As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

*Anyname.*REFLECTANCE40_WHITENESS_CALCULATE (Sample1(), 3, Result_WI)

CIE whiteness index under illuminant D65 and 10 degree for the color given in array "Sample1" will be output via variable Result_WI.

*Anyname.*REFLECTANCE40_WHITENESS_CALCULATE (SampleRef(), 6, Result_WI)

Stensby index for the color given in array "SampleRef" will be output via variable Result_WI.2

Subroutine no : 18

Subroutine name : CAPITALXYZ_WHITENESS_CALCULATE

Description : Calculates one of the following whiteness index for the color given with X, Y and tristimulus value.

Supported Whiteness index types
- WI E313
- WI CIE D65 - 2 degree
- WI CIE D65 – 10 degree
- WI Berger D65 - 2 degree
- WI Berger D65 – 10 degree
- WI Taube
- WI Ganz

Purpose : Send X, Y and Z tristimulus values and get Whiteness index.

General form : CAPITALXYZ_WHITENESS_CALCULATE(CAPITAL_X As Single, CAPITAL_Y As Single, CAPITAL_Z As Single, WhitenessType As Byte, Result_Whiteness As Single)

Input parameters:

Parameter 1 -> CAPITAL_X As Single
Tristimulus X value of the color (see NOTES/Term 2 of this subroutine)

Parameter 2 -> CAPITAL_Y As Single
Tristimulus Y value of the color (see NOTES/Term 2 of this subroutine)

Parameter 3 -> CAPITAL_Z As Single
Tristimulus Z value of the color (see NOTES/Term 2 of this subroutine)

Parameter 4 -> WhitenessType As Byte
Any integer number according to following list to get the desired whiteness index value.

Selection value	Meaning
1	E313 whiteness index
2	CIE whiteness index under illuminant D65 with 2 degree observer
3	CIE whiteness index under illuminant D65 with 10 degree observer
4	Berger whiteness index under illuminant D65 with 2 degree observer
5	Berger whiteness index under illuminant D65 with 10 degree observer
7	Taube whiteness index
8	Ganz whitenes index

Output parameters:

Parameter 1 -> Result_Whiteness As Single
Calculated Whiteness index according to color given with tristimulus X, Y and Z values and the selected WhitenessType parameter.

Notes : **1)** As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Some of the whiteness indices are calculated under a certain illuminant and observer degree also. Please be sure that supplied tristimulus X, Y and Z values are obtained under correct illuminant and observer for the whiteness type selected.

For example, E313 whiteness index is only meaningfull under illuminant C and 2 degree observer.

Table: Whiteness index and light source relation table

Index name	**Light Source/Observer**
E313 whiteness index	C/2
CIE whiteness index	D65/2 or D65/10
Berger whiteness index	D65/2 or D65/10
Taube whiteness index	D65/10
Ganz whiteness index	Any

Examples :

Anyname. CAPITALXYZ_WHITENESS_CALCULATE (22.98, 41,67, 51.09, 3, Result_WI)

CIE whiteness index according to tristimulus values X=22.98, Y=41.67 and Z=51.09 for D65/10 will be output via variable Result_WI.

*Anyname.*CAPITALXYZ _WHITENESS_CALCULATE (19.23, 44.12, 26.17, 4, Result_WI)

Stensby whiteness index according to tristimulus values X=19.23, Y=44.12 and Z=26.17 for D65/2 will be output via variable Result_WI.

Subroutine no : 19

Subroutine name : REFLECTANCE40_TINT_CALCULATE

Description : Calculates both TINT value (Ganz-Griesser and E313) for the color given in array REFLECTANCE40.

Purpose : Send reflectance values between 360nm-750nm with 10nm intervals (40 pcs) and get TINT values.

General form : REFLECTANCE40_TINT_CALCULATE(Reflectance40() As Single, Result_TintGG As Single, Result_TintE313 As Single)

Input parameters:

Parameter 1 -> Reflectance40() as Single

Any array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals.

Example:
```
Refs(1)=12.345  'Reflectance value for 360nm
Refs(2)=11.395  'Reflectance value for 370nm
      .
      .
      .
Refs(39)=32.123 'Reflectance value for 740nm
Refs(40)=36.523 'Reflectance value for 750nm
```

Output parameters:

Parameter 1 -> Result_TintGG As Single

Ganz-Griesser Tint value for the color given in the array REFLECTANCE40

Parameter 1 -> Result_E313 As Single

E313 Tint value for the color given in array REFLECTANCE40

Notes : 1) As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

*Anyname.*REFLECTANCE40_TINT_CALCULATE (SampleTest(), Result_T1, Result_T2)

TINT values for the color given in array SampleTest will be output via variable Result_T1 (Ganz-Griesser) and Result_T2 (E313)

Anyname. REFLECTANCE40_TINT_CALCULATE (LabBatch(), Result_TINT1, Result_TINT2)

TINT values for the color given in array SampleTest will be output via variable Result_TINT1 (Ganz-Griesser) and Result_TINT2 (E313)

Subroutine no : 20

Subroutine name : CAPITALXYZ_TINT_CALCULATE

Description : Calculates both TINT value (Ganz-Griesser and E313) for the color given with tristimulus X, Y and Z values.

Purpose : Send tristimulus X, Y and Z values and get TINT values.

General form : CAPITALXYZ_TINT_CALCULATE (CAPITAL_X As Single, CAPITAL_Y As Single, CAPITAL_Z As Single, Result_TintGG As Single, Result_TintE313 As Single)

Input parameters:

Parameter 1 -> CAPITAL_X As Single
Tristimulus X value of the color (see NOTES/Term 2 of this subroutine)

Parameter 2 -> CAPITAL_Y As Single
Tristimulus Y value of the color (see NOTES/Term 2 of this subroutine)

Parameter 3 -> CAPITAL_Z As Single
Tristimulus Z value of the color (see NOTES/Term 2 of this subroutine)

Output parameters:

Parameter 1 -> Result_TintGG As Single
Ganz-Griesser Tint value for the color given with tristimulus X, Y and Z values

Parameter 2 -> Result_E313 As Single
E313 Tint value for the color given with tristimulus X, Y and Z values

Notes : 1) As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) TINT values are calculated under D65 and 10 degree observer. Please be sure that X, Y and Z trisitimulus values supplied corresponds this principle.

Examples :

*Anyname.*CAPITALXYZ_TINT_CALCULATE (41.12, 18.91, 72.17, Result_T1, Result_T2)

TINT values for the color given with tristimulus values (X=41.12, Y=18.91 and Z=72.17) will be output via variable Result_T1 (Ganz-Griesser) and Result_T2 (E313)

*Anyname.*CAPITALXYZ_TINT_CALCULATE (12.21, 44.44, 19.89, Result_TINT1, Result_TINT2)

TINT values for the color given with tristimulus values (X=12.21, Y=44.44 and Z=19.89) will be output via variable Result_TINT1 (Ganz-Griesser) and Result_TINT2 (E313)

Subroutine no : 21

Subroutine name : LOWERCASEXYZ_TINT_CALCULATE

Description : Calculates both TINT value (Ganz-Griesser and E313) for the color given with chromaticity coordinates lowercase x, y and z values.

Purpose : Send chromaticity coordinate values lowercase x, y, z and get TINT values.

General form : LOWERCASEXYZ_TINT_CALCULATE(Lowercase_x As Single, Lowercase_y As Single, Lowercase_z As Single, Result_TintGG As Single, Result_TintE313 As Single)

Input parameters:

Parameter 1 -> Lowercase_x As Single
Cromaticity x value of the color (see NOTES/Term 2 of this subroutine)

Parameter 2 -> Lowercase_y As Single
Cromaticity y value of the color (see NOTES/Term 2 of this subroutine)

Parameter 3 -> Lowercase_z As Single
Cromaticity z value of the color (see NOTES/Term 2 of this subroutine)

Output parameters:

Parameter 1 -> Result_TintGG As Single
Ganz-Griesser Tint value for the color given with chromaticity coordinates values lowercase x, y, and z

Parameter 2 -> Result_E313 As Single
E313 Tint value for the color given with chromaticity coordinates values lowercase x, y, and z

Notes : **1)** As all output numeric parameters has decimal point, be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) TINT values are calculated under D65 and 10 degree observer. Please be sure that chromaticity coordinates lowercase x, y and z values supplied corresponds this principle.

Examples :

*Anyname.*LOWERCASEXYZ_TINT_CALCULATE (0.18, 0.32, 0.41, Result_T1, Result_T2)

TINT values for the color given with chromaticity coordinates values (x=0.18, y=0.32 and z=0.41) will be output via variable Result_T1 (Ganz-Griesser) and Result_T2 (E313)

*Anyname.*LOWERCASEXYZ_TINT_CALCULATE (0.12, 0.44, 0.19, Result_TINT1, Result_TINT2)

TINT values for the color given with chromaticity coordinates values (x=0.12, y=0.44 and z=0.19) will be output via variable Result_TINT1 (Ganz-Griesser) and Result_TINT2 (E313)

Subroutine no : 22

Subroutine name : REFLECTANCE40_GRAY_SCALE_CALCULATE

Description : Calculates Gray Scale rating factor for wash fastness and staining of a color.

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) and get wash fastness and staining ratings between 1 and 5.

General form : REFLECTANCE40_GRAY_SCALE_CALCULATE(Reflectance40_Std() As Single, Reflectance40_Trial() As Single, LightSource As String, Observer As Byte, Result_A02WashFastness As String, Result_A03Staining As String)

Input parameters:

Parameter 1 -> Reflectance40_Std() as Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the sample before test process (washing test for wash fastness or crockmeter test for staining).

Example: Refs(1)=12.345 'Reflectance value for 360nm
Refs(2)=11.395 'Reflectance value for 370nm
.
.
.
Refs(39)=32.123 'Reflectance value for 740nm
Refs(40)=36.523 'Reflectance value for 750nm

Parameter 2 -> Reflectance40_Trial() as Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the sample after test process (washing test for wash fastness or crockmeter test for staining).

Example: Refs(1)=11.245 'Reflectance value for 360nm
Refs(2)=12.695 'Reflectance value for 370nm
.
.
.
Refs(39)=33.173 'Reflectance value for 740nm
Refs(40)=32.423 'Reflectance value for 750nm

Parameter 3 -> LightSource As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate Gray Scale Ratings.

Parameter 4 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used for calculation

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_A02WashFastness As String
Returns "Wash Fastness grade" between 1 and 5

Parameter 2 -> Result_A03Staining As String
Returns "Staining grade" between 1 and 5

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100.

3) Output parameters are in string format so be sure that variable has been defined properly in your code.

Examples :

*Anyname.*REFLECTANCE40_GRAY_SCALE_CALCULATE (WhiteFabric(), Dyedfabric(),"D65", 10, F1, F2)

Wash fastness and staining grade will be calculated according to the differences under illuminant Daylight 6500K and 10 degree observer of two colors given in Array WhiteFabric and Dyedfabric and grades will be returned via variables F1 (Wash) and F2 (Staining).

*Anyname.*REFLECTANCE40_GRAY_SCALE_CALCULATE (BeforeTest(), AfterTest(),"D65", 10, WashF, StainingF)

Wash fastness and staining grade will be calculated according to the differences under illuminant Daylight 6500K and 10 degree observer of two colors given in Array BeforeTest and AfterTest and grades will be returned via variables WashF (Wash) and StainingF (Staining).

Subroutine no : 23

Subroutine name : CIELab_GRAY_SCALE_CALCULATE

Description : Calculates Gray Scale rating factors for wash fastness and staining of a color.

Purpose : Send CIELab L, a and b values for both (before test and after test samples) and get wash fastness and staining rating between 1 and 5.

General form : CIELab_GRAY_SCALE_CALCULATE (CIE_L_Std As Single, CIE_a_Std As Single, CIE_b_Std As Single, CIE_L_Trial As Single, CIE_a_Trial As Single, CIE_b_Trial As Single, Result_A02WashFastness As String, Result_A03Staining As String)

Input parameters:

Parameter 1 -> CIE_L_Std As Single
L value in CIELab color space for the sample before test process (washing test for wash fastness or crockmeter test for staining).

Parameter 2 -> CIE_a_Std As Single
a value in CIELab color space for the sample before test process (washing test for wash fastness or crockmeter test for staining).

Parameter 3 -> CIE_b_Std As Single
b value in CIELab color space for the sample before test process (washing test for wash fastness or crockmeter test for staining).

Parameter 4 -> CIE_L_Trial As Single
L value in CIELab color space for the sample after test process (washing test for wash fastness or crockmeter test for staining).

Parameter 5 -> CIE_a_Trial As Single
a value in CIELab color space for the sample after test process (washing test for wash fastness or crockmeter test for staining).

Parameter 6 -> CIE_b_Trial As Single
b value in CIELab color space for the sample after test process (washing test for wash fastness or crockmeter test for staining).

Output parameters:

Parameter 1 -> Result_A02WashFastness As String
Returns “Wash Fastness grade” between 1 and 5

Parameter 2 -> Result_A03Staining As String
Returns “Staining grade” between 1 and 5

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use “single” declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Output parameters are in string format so be sure that variable has been defined properly in your code.

Examples :

*Anyname.*CIELab_GRAY_SCALE_CALCULATE (56.12, -21.34, 14.56, 41.17, -20.17, 12.55, F1, F2)

Wash fastness and staining grade will be calculated according to two samples given with CIELab values (before test -> L=56.12, a=-12.34, b=14.56) and (after test -> L=41.17, a=-20.17, b=12.55) and grades will be returned via variables F1 (Wash) and F2 (Staining).

*Anyname.*CIELab_GRAY_SCALE_CALCULATE (46.12, 29.34, 10.56, 46.17, 29.17, 10.55, WashF, StainingF)

Wash fastness and staining grade will be calculated according to two samples given with CIELab values (before test -> L=46.12, a=29.34, b=10.56) and (after test -> L=45.07, a=29.17, b=10.55) and grades will be returned via variables WashF (Wash) and StainingF (Staining).

Subroutine no : 24

Subroutine name : CIELabDELTAE_GRAY_SCALE_CALCULATE

Description : Calculates Gray Scale rating factors for wash fastness and staining of a color.

Purpose : Send DELTA E [CIELab] value between samples (before and after test) and get wash fastness and staining ratings between 1 and 5.

General form : CIELabDELTAE_GRAY_SCALE_CALCULATE (CIELab_DE As Single, Result_A02WashFastness As String, Result_A03Staining As String)

Input parameters:

Parameter 1 -> CIELab_DE As Single
CIELab DELTA E value between two samples (before test as standard and after test as trial)

Output parameters:

Parameter 1 -> Result_A02WashFastness As String
Returns "Wash Fastness grade" between 1 and 5

Parameter 2 -> Result_A03Staining As String
Returns "Staining grade" between 1 and 5

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

3) Output parameters are in string format so be sure that variable has been defined properly in your code.

Examples :

*Anyname.*CIELabDELTAE_GRAY_SCALE_CALCULATE (2,17, F1, F2)

Wash fastness and staining grade will be calculated according to 2.17 DE difference between two samples and grades will be returned via variables F1 (Wash) and F2 (Staining).

*Anyname.*CIELabDELTAE_GRAY_SCALE_CALCULATE (1,11, WashF, StainingF)

Wash fastness and staining grade will be calculated according to 1.11 DE difference between two samples and grades will be returned via variables WashF (Wash) and WashF (Staining).

Subroutine no : 25

Subroutine name : REFLECTANCE40_OPACITY_CALCULATE

Description : Calculates %opacity of a sample

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) for both samples overblack and overwhite and get %opacity value.

General form : REFLECTANCE40_OPACITY_CALCULATE(Reflectance40_OverBlack() As Single, Reflectance40_OverWhite() As Single, LightSource As String, Observer As Byte, OpacityType As Byte, Result_Opacity As Single)

Input parameters:

Parameter 1 -> Reflectance40_OverBlack() As Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the sample over black

Example: Refs(1)=12.345 'Reflectance value for 360nm
Refs(2)=11.395 'Reflectance value for 370nm
.
.
.
Refs(39)=32.123 'Reflectance value for 740nm
Refs(40)=36.523 'Reflectance value for 750nm

Parameter 2 -> Reflectance40_OverWhite() As Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the sample over white

Example: Refs(1)=19.345 'Reflectance value for 360nm
Refs(2)=21.395 'Reflectance value for 370nm
.
.
.
Refs(39)=42.123 'Reflectance value for 740nm
Refs(40)=56.523 'Reflectance value for 750nm

Parameter 3 -> LightSource As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate %opacity

Parameter 4 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used for calculation

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Parameter 5 -> OpacityType As Byte

One of the options below determining %opacity type

Type	Meaning
1	%Opacity at maximum value
2	%Opacity according to tristimulus Z (Suitbale for blue colors)
3	%Opacity according to tristimulus Y (Suitable for green color)
4	%Opacity according to tristimulus X (Suitable for yellow-red colors)
5	%opacity integrated (Suitable for general purposes)

Output parameters:

Parameter 1 -> Result_Opacity As Single
Returns %opacity value

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100.

Examples :

*Anyname.*REFLECTANCE40_OPACITY_CALCULATE (ColorOverBlack(), ColorOverWhite(), 3, "D65", 2, Opacity)

%Opacity according to tristimulus Y value for D65 and 2 degree of the color specified in arrays ColorOverBlack and ColorOverWhite will be output via variable opacity.

*Anyname.*REFLECTANCE40_OPACITY_CALCULATE (SampleOB(), SampleOW(),2,"D65", 2, result)

%Opacity according to tristimulus X value for D65 and 2 degree of the color specified in arrays SampleOB and SampleOW will be output via variable "result".

Subroutine no : 26

Subroutine name : CAPITALXYZ_OPACITY_CALCULATE

Description : Calculates %opacity of a sample

Purpose : Send tristimulus X, Y and Z values of sample overblack and overwhite and get %opacity value.

General form : CAPITALXYZ_OPACITY_CALCULATE (CAPITAL_X_OverWhite As Single, CAPITAL_Y_OverWhite As Single, CAPITAL_Z_OverWhite As Single, CAPITAL_X_OverBlack As Single, CAPITAL_Y_OverBlack As Single, CAPITAL_Z_OverBlack As Single, OpacityType As Byte, Result_Opacity As Single)

Input parameters:

Parameter 1 -> CAPITAL_X_OverWhite As Single
Tristimulus X value of the color over white.

Parameter 2 -> CAPITAL_Y_OverWhite As Single
Tristimulus Y value of the color over white.

Parameter 3 -> CAPITAL_Z_OverWhite As Single
Tristimulus Z value of the color over white.

Parameter 4 -> CAPITAL_X_OverBlack As Single
Tristimulus X value of the color over black

Parameter 5 -> CAPITAL_Y_OverBlack As Single
Tristimulus Y value of the color over black.

Parameter 6 -> CAPITAL_Z_OverBlack As Single
Tristimulus Z value of the color over black.

Parameter 7 -> OpacityType As Byte
One of the options below determining %opacity type

Type	Meaning
2	%Opacity according to tristimulus Z (Suitbale for blue colors)
3	%Opacity according to tristimulus Y (Suitable for green color)
4	%Opacity according to tristimulus X (Suitable for yellow-red colors)

Output parameters:

Parameter 1 -> Result_Opacity As Single
Returns %opacity value

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use “single” declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

*Anyname.*CAPITALXYZ_OPACITY_CALCULATE (11.23, 45.34, 56.12, 9.12, 44.12, 55.56,2, Opacity)

%Opacity according to tristimulus Y value of the color specified with tristimulus X, Y, Z over white and over black will be output via variable opacity.

*Anyname.*CAPITALXYZ_OPACITY_CALCULATE (4.23, 15.34, 26.12, 5.12, 14.12, 25.16, 1, Opct)

%Opacity according to tristimulus X value of the color specified with tristimulus X, Y, Z over white and over black will be output via variable opacity.

Subroutine no : 27

Subroutine name : REFLECTANCE40_STRENGTH_CALCULATE

Description : Calculates 3 types strengths (Chromatic, Apparent and Integrated) relatively to a color

Purpose : Send two sets of reflectance values between 360nm-750nm with 10nm interval (40 pcs) for two colors and get one's (Trail) strength value relative to the other (standard).

General form : REFLECTANCE40_STRENGTH_CALCULATE (Reflectance40_Std() As Single, Reflectance40_Trial() As Single, Result_ChromaticStrength As Single, Result_ApperentStrength As Single, Result_IntegratedStrength As Single)

Input parameters:

Parameter 1 -> Reflectance40_Std() As Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the color assuming standard.

```
Example: Refs(1)=12.345 'Reflectance value for 360nm
         Refs(2)=11.395 'Reflectance value for 370nm
                  .
                  .
                  .
         Refs(39)=32.123 'Reflectance value for 740nm
         Refs(40)=36.523 'Reflectance value for 750nm
```

Parameter 2 -> Reflectance40_Trial() As Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the color assuming trial.

```
Example: Refs(1)=11.345 'Reflectance value for 360nm
         Refs(2)=14.39 'Reflectance value for 370nm
                  .
                  .
                  .
         Refs(39)=29.16 'Reflectance value for 740nm
         Refs(40)=35.43 'Reflectance value for 750nm
```

Output parameters:

Parameter 1 -> Result_ChromaticStrength As Single

Returns Chromatic strength value of trial color relative to standard color assumed as 100

Parameter 2 -> Result_ApparentStrength As Single

Returns Apparent strength value of trial color relative to standard color assumed as 100

Parameter 3 -> Result_IntegratedStrength As Single

Returns Integrated strength value of trial color relative to standard color assumed as 100.

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100.

Examples :

*Anyname.*REFLECTANCE40_STRENGTH_CALCULATE (Color1Refs(), Coor2Refs(), Cst, ASt, Ist)

Cst (Chromatic strength), Ast (Apparent strength) and Ist (Integrated strength) of the color given in array Color1Refs relative to the color given in array Color2Refs will be output.

*Anyname.*REFLECTANCE40_STRENGTH_CALCULATE (Shade1(), Shade2(), Crmst, AppSt, Intst)

Crmst (Chromatic strength), Appst (Apparent strength) and Intst (Integrated strength) of the color given in array Shade1 relative to the color given in array Shade2 will be output.

Subroutine no : 28

Subroutine name : REFLECTANCE40_BRIGHTNESS_CALCULATE

Description : Calculates 2 types brightness value Z and R457.

Purpose : Send reflectance values between 360nm-750nm with 10nm interval (40 pcs) and get brightness values Z and R457.

General form : REFLECTANCE40_BRIGHTNESS_CALCULATE(Reflectance40() As Single, LightSource As String, Observer As Byte, Result_Brightness_Z As Single, Result_Brightness_R457 As Single)

Input parameters:

Parameter 1 -> Reflectance40 As Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the color

Example: Refs(1)=12.345 'Reflectance value for 360nm
Refs(2)=11.395 'Reflectance value for 370nm
.
.
.
Refs(39)=32.123 'Reflectance value for 740nm
Refs(40)=36.523 'Reflectance value for 750nm

Parameter 2 -> LightSource As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate brightness values

Parameter 3 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used for calculation

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_Brightness_Z As Single

Returns under specified light source and observer Z brightness value of the color given in array Reflectance40

Parameter 2 -> Result_Brightness_R457 As Single

Returns under specified light source and observer R457 brightness value of the color given in array Reflectance40

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100.

Examples :

*Anyname.*REFLECTANCE40_BRIGHTNESS_CALCULATE (Color1(), “D65”, 10, S1, S2)

Under light source D65 and 10 degree observer, Z brightness in variable S1 and R457 brightness in the variable S2 of the color given in Array Color() will be output.

*Anyname.*REFLECTANCE40_BRIGHTNESS_CALCULATE (SampleColor(), “D65”, 10, ZStr, RStr)

Under light source D65 and 10 degree observer, Z brightness in variable ZStr and R457 brightness in Variable Rstr of the color given in Array Color() will be output.

Subroutine no : 29

Subroutine name : CAPITALXYZ_BRIGHTNESS_CALCULATE

Description : Calculates Z type brightness value

Purpose : Send tristimulus X, Y and Z values and get "Z" brightness value.

General form : CAPITALXYZ_BRIGHTNESS_CALCULATE (CAPITAL_X As Single, CAPITAL_Y As Single, CAPITAL_Z As Single)

Input parameters:

Parameter 1 -> CAPITAL_X As Single
Tristimulus X value of the color

Parameter 2 -> CAPITAL_Y As Single
Tristimulus Y value of the color

Parameter 3 -> CAPITAL_Z As Single
Tristimulus Z value of the color

Output parameters:

Parameter 1 -> Result_Brightness_Z As Single
Returns Z brightness value of the color given with X, Y and Z tristimulus value.

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

*Anyname.*CAPITALXYZ_BRIGHTNESS_CALCULATE (41.87, 10.21, 33.56, BrZ)

Z brightness value for the color given with tristimulus values (X=41.87, Y=10.21 and Z=33.56) will be output via variable Brz

*Anyname.*CAPITALXYZ_BRIGHTNESS_CALCULATE (53.87, 23.21, 29.56, Zbrg)

Z brightness value for the color given with tristimulus values (X=53.87, Y=23.21 and Z=29.56) will be output via variable Zbrg.

Subroutine no : 30

Subroutine name : REFLECTANCE40_METAMERISM_CALCULATE

Description : Calculates metamerism (color difference change between two different light source)

Purpose : Send reflectance values and two lightsources and get how the color difference Is changing under different illuminats (metamerism)

General form : REFLECTANCE40_METAMERISM_CALCULATE (Reflectance40_Std() As Single, Reflectance40_Trial() As Single, LightSource1 As String, LightSource2 As String, Observer As Byte, Result_Metamerism As Single)

Input parameters:

Parameter 1 -> Reflectance40_Std As Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the first color

Example: Refs(1)=12.345 'Reflectance value for 360nm
Refs(2)=11.395 'Reflectance value for 370nm
.
.
.
Refs(39)=32.123 'Reflectance value for 740nm
Refs(40)=36.523 'Reflectance value for 750nm

Parameter 2 -> Reflectance40_Trial As Single

An array with 40 elements having reflectance values between 360nm-750nm with 10nm intervals of the second color

Example: Refs(1)=11.345 'Reflectance value for 360nm
Refs(2)=10.395 'Reflectance value for 370nm
.
.
.
Refs(39)=34.123 'Reflectance value for 740nm
Refs(40)=35.523 'Reflectance value for 750nm

Parameter 3 -> LightSource1 As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be accepted as the first illuminant.

Parameter 4 -> LightSource2 As String

One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be accepted as the second illuminant.

Parameter 3 -> Observer as Byte

One of the options below to tell the subroutine which observer for the specified light source will be used for calculation

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters:

Parameter 1 -> Result_Metamerism As Single

Returns metamerism index which shows how the difference between two colors is changing when the light source switched from LightSource1 to LightSource2.

If Result_Metamerism is equal to 1, then metamerism is perfect. In another saying that color difference is not changing when the light source changed.

Be careful! Metamerism=1 does not mean that the colors are not changing from one light source to another. Metamerism=1 only means that, the difference between two colors stay same from one light source to another.

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100.

Examples :

Anyname. REFLECTANCE40_METAMERISM_CALCULATE (Color1(), Color2(),"D65", "TL83",10, Met)

Metamerism index showing difference changing between Color1 and Color2 under illuminant D65 will be output via variable Met.

Anyname. REFLECTANCE40_METAMERISM_CALCULATE (CustomerRed(), LabRed(),"D65", "TL83",10, ResultMet)

Metamerism index showing difference changing between CustomerRed and LabRed under illuminant D65 will be output via variable ResultMet.

Subroutine no : 31

Subroutine name : CIELab_METAMERISM_CALCULATE

Description : Calculates metamerism (color difference change between two different light source)

Purpose : Send L, a, b values in CIELab color space for two colors and get how the color difference is changing under different illuminats (metamerism)

General form : CIELab_METAMERISM_CALCULATE(CIE_L1_Std As Single, CIE_a1_Std As Single, CIE_b1_Std As Single, CIE_L1_Trial As Single, CIE_a1_Trial As Single, CIE_b1_Trial As Single, CIE_L2_Std As Single, CIE_a2_Std As Single, CIE_b2_Std As Single, CIE_L2_Trial As Single, CIE_a2_Trial As Single, CIE_b2_Trial As Single, Result_Metamerism As Single)

Input parameters:

Parameter 1 -> CIE_L1_Std As Single
L value [CIELab] under first light source of the first color

Parameter 2 -> CIE_a1_Std As Single
a value [CIELab] under first light source of the first color

Parameter 3 -> CIE_b1_Std As Single
b value [CIELab] under first light source of the first color

Parameter 4 -> CIE_L1_Trial As Single
L value [CIELab] under first light source of the second color

Parameter 5 -> CIE_a1_Trial As Single
a value [CIELab] under first light source of the second color

Parameter 6 -> CIE_b1_Trial As Single
b value [CIELab] under first light source of the second color

Parameter 7-> CIE_L2_Std As Single
L value [CIELab] under second light source of the first color

Parameter 8 -> CIE_a2_Std As Single
a value [CIELab] under second light source of the first color

Parameter 9 -> CIE_b2_Std As Single
b value [CIELab] under second light source of the first color

Parameter 10-> CIE_L2_Trial As Single
L value [CIELab] under second light source of the second color

Parameter 11-> CIE_a2_Trial As Single
a value [CIELab] under second light source of the second color

Parameter 12-> CIE_b2_Trial As Single
b value [CIELab] under second light source of the second color

Output parameters:

Parameter 1 -> Result_Metamerism As Single

Returns metamerism index which shows how the difference between two colors is changing when the light source switched from LightSource1 to LightSource2.

If Result_Metamerism is equal to 1, then metamerism is perfect. In another saying that color difference is not changing when the light source changed.

Be careful! Metamerism=1 does not mean that the colors are not changing from one light source to another. Metamerism=1 only means that, the difference between two colors stay same from one light source to another.

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

Anyname.CIELab_METAMERISM_CALCULATE (44.12, 3.12, 7.78, 45.01, 3.99, 8.71, 46.17, 4,14, 6.01, 48.03, 5.21, 7.11, MetameriIndex)

Metamerism index showing difference changing between first color and second color will be output via variable MetameriIndex.

In this example:
44.12 means L value under first light source of the first color
3.12 means a value under first light source of the first color
7.78 means b value under first light source of the first color
45.01 means L value under first light source of the second color
3.99 means a value under first light source of the second color
8.71 means b value under first light source of the second color
46.17 means L value under second light source of the first color
4.14 means a value under second light source of the first color
6.01 means b value under second light source of the first color
48.03 means L value under second light source of the second color
5.21 means a value under second light source of the second color
7.11 means b value under second light source of the second color

Anyname.CIELab_METAMERISM_CALCULATE (40.12, 13.12, 27.78, 42.01, 13.99, 28.71, 46.17, 14,14, 26.01, 48.03, 15.21, 27.11, MetIndex)

Metamerism index showing difference changing between first color and second color will be output via variable MetIndex.

In this example:
40.12 means L value under first light source of the first color
13.12 means a value under first light source of the first color
27.78 means b value under first light source of the first color
42.01 means L value under first light source of the second color
13.99 means a value under first light source of the second color
28.71 means b value under first light source of the second color
46.17 means L value under second light source of the first color

14.14 means a value under second light source of the first color
26.01 means b value under second light source of the first color
48.03 means L value under second light source of the second color
15.21 means a value under second light source of the second color
27.11 means b value under second light source of the second color

Subroutine no : 32

Subroutine name : CIELab_METAMERISM_DIRECTION_CALCULATE

Description : Detects metamerism direction according to the position of two colors in ab axes.

Purpose : Send L, a, b values in CIELab color space for two colors and see both colors is at same region or not at ab axes of CIELab color space.

General form : CIELAB_METAMERISM_DIRECTION_CALCULATE(CIE_a1 As Single, CIE_b1 As Single, CIE_a2 As Single, CIE_b2 As Single, Result_Metamerism As String)

Input parameters:

Parameter 1 -> CIE_a1 As Single
a value [CIELab] under first light source of the first color

Parameter 2 -> CIE_b1 As Single
b value [CIELab] under second light source of the first color

Parameter 3 -> CIE_a2 As Single
a value [CIELab] under first light source of the second color

Parameter 4 -> CIE_b1 As Single
b value [CIELab] under second light source of the second color

Output parameters:

Parameter 1 -> Result_Metamerism As String
Returns 2 digit text as described follows.
"00" No metamerism
"10" Metamerisim at horizontal (Red-Green) axe. The color is changing from red to green or green to red between specifed two light sources
"01" Metamerisim at vertical (Blue-Yellow) axe. The color is changing from blue to yellow or yellow to blue between specifed two light sources
"11" Metamerism on both horizontal and vertical axes. Worst case!!!

Notes : -

Examples :

Anyname.CIELAB_METAMERISM_DIRECTION_CALCULATE(0.18,22.12,-0.3,23.09,Mt)

Metamerism direction indicator will be returned via variable Mt. The result will be "10"

Anyname.CIELAB_METAMERISM_DIRECTION_CALCULATE(0.18,22.12,0.3,23.09,Mt)

Metamerism direction indicator will be returned via variable Mt. The result will be "00"

Subroutine no : 33

Subroutine name : REFLECTANCE31_RECIPE_CALCULATE

Description : Calculates dyeing recipe for a color

Purpose : Send reflectance values of target color, used substrate, target substrate, calibration shades, number of dyes in a recipe and get all recipe alternatives in a specified text file

General form : REFLECTANCE31_RECIPE_CALCULATE(DyeShadesReflectance31() As Single, ShadesSubstrateReflectance31() As Single, TargetSubstrateReflectance31() As Single, TargetColorReflectance31() As Single, Prices() As Single, NumberOfDyesInEachRecipe As Integer, OutputTextFileName As String, Screen As Boolean, InformationX As Integer, InformationY As Integer, Result_NumberOfAlternatives As Integer)

Input parameters:

Parameter 1 -> DyeShadesReflectance31() As Single

Three dimensional array with boundaries 6, 15 and 32 elements. Another saying that an array defined as DyeShadesReflectance31(1 to 6, 1 to 15 and 1 to 32). These three dimensional array is for shades' reflectance values (also known as calibration data) with different concentrations (%C) prepared in laboratory. The format of this array is as follows.

First dimension (6 elements) represents dye number.
Second dimension (15 elements) represents calibration data number
Third dimension (32 elements) represent calibration data %C value and its 31 elements reflectance values 400nm-700nm with 10nm intervals.

Filling this 3 dimensional array correctly is very critical to get good recipies.

To understand better, let's assume a scenario as follows.

Scenerio:

Assume, we have 5 dyes named DYE-1, DYE-2, DYE-3, DYE-4 and DYE-5

Each dye has calibration data as follows.

DYE-1 has 7 shades with different concentrations
These are %C=0.005, 0.0075, 0.01, 0.05, 0.1, 0.5 and 1.0

DYE-2 has 5 shades with different concentrations.
These are %C=0.0075, 0.01, 0.15, 0.5 and 1.0

DYE-3 has 4 shades with different concentrations.
These are %C=0.05, 0.075, 0.15 and 0.3

DYE-4 has 9 shades with different concentrations.
These are %C=0.05, 0.075, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 and 0.7

DYE-5 has 7 shades with different concentrations.
These are %C=0.01, 0.05, 0.1, 0.15, 0.3, 0.5 and 1.0

Reflectance values between 400nm-700nm with 10nm intervals of each concentrations are as follows.

For DYE-1 %C=0.005

(49.765, 48.675, 47.51, 48.06, 49.27, 52.415, 60.25, 67.675, 71.24, 72.635, 73.55, 73.755, 73.665, 73.65, 73.805, 74.04, 74.125, 74.12, 74.13, 74.105, 74.015, 73.98, 74.065, 74.19, 74.345, 74.535, 74.705, 74.665, 74.74, 74.81, 75.06)

For DYE-1 %C=0.0075

(45.5, 43.945, 42.485, 42.985, 44.225, 47.63, 56.42, 65.065, 69.44, 71.195, 72.33, 72.73, 72.79, 72.88, 73.105, 73.38, 73.44, 73.295, 73.175, 72.93, 72.59, 72.43, 72.55, 72.885, 73.5, 74.215, 74.845, 75.16, 75.34, 75.415, 75.58)

For DYE-1 %C=0.01

(42.36, 40.485, 38.875, 39.275, 40.49, 43.97, 53.515, 63.465, 68.865, 71.14, 72.54, 73.11, 73.235, 73.35, 73.58, 73.86, 73.84, 73.605, 73.385, 73.05, 72.69, 72.585, 72.765, 73.22, 73.92, 74.68, 75.355, 75.685, 75.895, 76.02, 76.23)

For DYE-1 %C=0.05

(25.435, 21.91, 19.54, 19.585, 20.485, 23.84, 36.96, 55.355, 68.69, 74.34, 76.515, 76.825, 76.36, 75.985, 75.875, 75.895, 75.715, 75.41, 75.225, 74.965, 74.65, 74.465, 74.44, 74.48, 74.655, 74.9, 75.135, 75.19, 75.19, 75.19, 75.255)

For DYE-1 %C=0.1

(15.09, 12.415, 10.725, 10.755, 11.37, 13.735, 24.775, 43.575, 60.1, 68.27, 71.545, 72.355, 72.165, 71.9, 71.84, 71.89, 71.765, 71.505, 71.33, 71.135, 70.9, 70.925, 71.155, 71.49, 71.985, 72.5, 72.93, 72.985, 73.105, 73.425, 73.825)

For DYE-1 %C=0.5

(4.68, 3.5, 2.875, 2.885, 3.08, 3.745, 10.19, 26.855, 47.985, 61.9, 69.215, 72.28, 73.26, 73.54, 73.71, 74.085, 74.455, 74.56, 74.56, 74.485, 74.335, 74.285, 74.295, 74.255, 74.295, 74.395, 74.495, 74.56, 74.615, 74.635, 74.7)

For DYE-1 %C=1

(1.71, 1.475, 1.365, 1.366, 1.415, 1.615, 3.495, 13.965, 32.715, 48.24, 58.59, 64.595, 67.79, 69.32, 70.01, 70.845, 71.935, 72.555, 72.595, 72.395, 72.04, 71.735, 71.66, 71.95, 72.795, 73.875, 74.86, 75.495, 75.87, 76.04, 76.16)

For DYE-2 %C=0.0075

(48.935, 47.425, 45.415, 43.755, 42.505, 41.975, 42.33, 43.515, 45.655, 48.55, 52.565, 57.075, 61.895, 66.245, 69.66, 72.045, 73.265, 73.77, 74.06, 74.2, 74.195, 74.24, 74.35, 74.43, 74.63, 74.84, 75.045, 75.12, 75.2, 75.255, 75.345)

For DYE-2 %C=0.01

(26.76, 23.17, 20.21, 18.2, 16.92, 16.405, 16.705, 17.745, 19.795, 22.69, 27.12, 33.01, 40.75, 49.285, 57.51, 64.465, 69.365, 72.135, 73.385, 73.935, 74.165, 74.345, 74.525, 74.6, 74.815, 75.035, 75.175, 75.285, 75.335, 75.41, 75.48)

For DYE-2 %C=0.15

(11.985, 9.015, 6.975, 5.765, 5.04, 4.725, 4.78, 5.195, 6.14, 7.64, 10.325, 14.555, 21.33, 30.475, 41.635, 53.155, 62.365, 67.995, 70.77, 71.965, 72.37, 72.555, 72.705, 72.82, 73.105, 73.435, 73.695, 73.865, 73.965, 74.05, 74.135)

For DYE-2 %C=0.5

(4.955, 3.49, 2.645, 2.21, 1.99, 1.89, 1.91, 2.035, 2.33, 2.85, 3.985, 6.135, 10.435, 17.445, 27.84, 40.74, 53.245, 62.335, 67.495, 69.975, 70.965, 71.445, 71.725, 71.825, 71.965, 72.095, 72.23, 72.395, 72.51, 72.585, 72.64)

For DYE-2 %C=1

(2.24, 1.725, 1.46, 1.36, 1.335, 1.285, 1.3, 1.31, 1.375, 1.5, 1.835, 2.71, 5.015, 9.62, 17.85, 29.87, 43.58, 55.31, 63.105, 67.29, 69.095, 69.905, 70.44, 70.79, 71.235, 71.645, 71.945, 72.125, 72.25, 72.355, 72.475)

For DYE-3 %C=0.05

(39.68, 37, 33.88, 31.125, 28.57, 26.56, 25.02, 23.985, 23.275, 23.06, 23.225, 23.675, 24.655, 26.1, 28.165, 31.085, 34.88, 39.355, 44.45, 49.91, 55.14, 59.8, 63.77, 66.875, 69.455, 71.515, 73.035, 73.98, 74.585, 74.925, 75.16)

For DYE-3 %C=0.075

(36.75, 33.345, 29.745, 26.62, 23.84, 21.73, 20.14, 19.08, 18.41, 18.15, 18.26, 18.7, 19.61, 20.97, 22.94, 25.735, 29.42, 33.875, 39.015, 44.655, 50.245, 55.44, 60.035, 63.62, 66.41, 68.53, 70.1, 71.17, 71.86, 72.215, 72.455)

For DYE-3 %C=0.15

(27.47, 23.68, 20.095, 17.26, 14.89, 13.16, 11.92, 11.115, 10.6, 10.41, 10.495, 10.805, 11.475, 12.525, 14.065, 16.325, 19.52, 23.64, 28.78, 34.915, 41.58, 48.365, 54.92, 60.485, 65.05, 68.61, 71.115, 72.74, 73.775, 74.33, 74.7)

For DYE-3 %C=0.3

(20.065, 16.115, 12.805, 10.365, 8.485, 7.215, 6.335, 5.78, 5.455, 5.335, 5.385, 5.57, 6.025, 6.735, 7.85, 9.545, 12.04, 15.5, 20.125, 26, 32.85, 40.29, 48, 54.995, 60.945, 65.67, 69.035, 71.245, 72.66, 73.43, 73.92)

For DYE-4 %C=0.05

(51.15, 49.665, 46.43, 42.55, 38.28, 34.22, 30.65, 27.575, 24.395, 21.56, 18.97, 17.12, 16.14, 15.635, 15.71, 17.29, 22.17, 31.415, 43.79, 55.47, 63.175, 67.015, 68.715, 69.56, 70.325, 71.11, 71.825, 72.31, 72.635, 72.79, 72.945)

For DYE-4 %C=0.075

(51.12, 48.85, 44.485, 39.785, 34.895, 30.485, 26.735, 23.615, 20.505, 17.77, 15.325, 13.615, 12.7, 12.24, 12.275, 13.64, 18.075, 27.1, 40.145, 53.365, 62.745, 67.745, 70.07, 71.15, 72.015, 72.835, 73.54, 74.02, 74.33, 74.475, 74.62)

For DYE-4 %C=0.1

(47.435, 44.16, 39.41, 34.525, 29.69, 25.5, 22.02, 19.175, 16.38, 13.98, 11.865, 10.43, 9.685, 9.3, 9.315, 10.45, 14.27, 22.335, 34.785, 48.54, 59.32, 65.795, 69.04, 70.57, 71.49, 72.2, 72.835, 73.27, 73.555, 73.735, 73.875)

For DYE-4 %C=0.2

(40.4, 35.33, 29.54, 24.375, 19.78, 16.07, 13.175, 10.955, 8.895, 7.235, 5.905, 5.03, 4.615, 4.41, 4.425, 5.045, 7.405, 13.295, 23.84, 36.785, 47.105, 52.265, 53.905, 55.055, 57.71, 61.725, 65.9, 69.255, 71.495, 72.705, 73.4)

For DYE-4 %C=0.3

(37.24, 31.345, 25.2, 19.96, 15.555, 12.16, 9.65, 7.785, 6.15, 4.91, 3.935, 3.33, 3.05, 2.925, 2.99, 3.255, 4.865, 9.405, 18.895, 32.68, 46.635, 57.205, 63.825, 67.345, 69.23, 70.415, 71.28, 71.855, 72.28, 72.5, 72.68)

For DYE-4 %C=0.4

(33.31, 27.015, 20.9, 15.99, 12.04, 9.1, 7.01, 5.52, 4.265, 3.365, 2.68, 2.265, 2.095, 2.025, 2.025, 2.25, 3.315, 6.74, 14.78, 27.675, 41.85, 53.39, 61, 65.115, 67.245, 68.56, 69.75, 70.855, 71.615, 71.95, 72.15)

For DYE-4 %C=0.5

(32.15, 25.395, 19.105, 14.29, 10.49, 7.82, 6.03, 4.84, 3.845, 3.09, 2.545, 2.235, 2.105, 2.035, 2.04, 2.19, 3.065, 6.12, 13.79, 26.62, 41.16, 53.295, 61.455, 65.98, 68.495, 70.12, 71.37, 72.35, 73.01, 73.345, 73.565)

For DYE-4 %C=0.6

(26.97, 20.28, 14.49, 10.3, 7.2, 5.15, 3.82, 2.96, 2.32, 1.885, 1.595, 1.455, 1.395, 1.37, 1.38, 1.435, 1.81, 3.655, 9.28, 20.13, 34.105, 47.175, 57, 62.945, 66.29, 68.29, 69.795, 71.065, 71.92, 72.36, 72.635)

For DYE-4 %C=0.7

(29.735, 22.95, 16.84, 12.295, 8.785, 6.4, 4.795, 3.705, 2.865, 2.285, 1.875, 1.665, 1.58, 1.545, 1.545, 1.64, 2.24, 4.63, 11.155, 22.62, 35.785, 45.91, 51.395, 54.26, 57.325, 61.415, 65.645, 68.99, 71.315, 72.65, 73.445)

For DYE-5 %C=0.01

(56.98, 59.625, 61.255, 62.485, 63.39, 64.465, 65.67, 66.855, 67.795, 68.65, 69.265, 69.025, 67.98, 66.29, 63.88, 61.03, 57.99, 54.505, 50.33, 45.865, 42.715, 41.585, 40.445, 38.225, 36.605, 38.115, 46.01, 61.305, 78.485, 90.035, 94.02)

For DYE-5 %C=0.05

(51.47, 53.62, 53.965, 53.94, 53.925, 54.985, 57.07, 59.015, 60.41, 61.71, 62.2, 60.87, 57.79, 53.665, 48.655, 43.38, 38.535, 33.76, 28.575, 23.6, 20.44, 19.465, 18.48, 16.505, 15.135, 16.53, 23.5, 38.465, 60.7, 83.785, 99.195)

For DYE-5 %C=0.1

(48.06, 49.9, 49.35, 48.555, 47.885, 48.895, 51.37, 53.54, 55.105, 56.355, 56.44, 54.25, 50.045, 44.815, 38.91, 33.155, 28.235, 23.65, 18.92, 14.66, 12.135, 11.4, 10.69, 9.25, 8.34, 9.6, 15.155, 27.305, 46.93, 70.625, 90.2)

For DYE-5 %C=0.15

(43.635, 45.3, 44.295, 43.045, 42.175, 43.195, 45.91, 48.27, 49.965, 51.3, 51.25, 48.72, 44.06, 38.485, 32.385, 26.65, 21.99, 17.805, 13.68, 10.13, 8.125, 7.595, 7.11, 6.05, 5.39, 6.38, 10.71, 20.45, 37.205, 59.31, 79.97)

For DYE-5 %C=0.3

(38.2, 39.51, 37.69, 35.79, 34.375, 35.15, 37.945, 40.465, 42.185, 43.59, 43.375, 40.295, 35.015, 29.11, 23.145, 17.95, 14.01, 10.745, 7.75, 5.36, 4.12, 3.83, 3.61, 3.05, 2.73, 3.3, 5.69, 11.475, 22.67, 39.73, 59)

For DYE-5 %C=0.5

(32.015, 33.345, 31.215, 29.1, 27.6, 28.44, 31.405, 34.085, 35.965, 37.4, 36.94, 33.44, 27.84, 21.98, 16.435, 11.94, 8.835, 6.43, 4.38, 2.905, 2.24, 2.125, 2.04, 1.79, 1.645, 1.945, 3, 5.72, 12.17, 24.185, 40.455)

For DYE-5 %C=1

(22.775, 23.81, 21.75, 19.765, 18.395, 19.075, 21.635, 23.88, 25.35, 26.36, 25.57, 22.255, 17.465, 12.775, 8.76, 5.845, 4.065, 2.865, 1.985, 1.46, 1.285, 1.28, 1.275, 1.225, 1.215, 1.29, 1.5, 1.965, 3.74, 8.955, 18.525)

According to this scenario the content of the array DyeShadesReflectance31 will form like this. Please remember that 1st dimension represent dye number, 2nd dimension represent shade number and 3rd dimension represent %C and reflectance values of it between 400nm-700nm with 10nm intervals.

DyeShadesReflectance31(1,1,1)=0.005 -> DYE-1 1st shade concentration
DyeShadesReflectance31(1,1,2)=49.765 -> DYE-1 1st shade 400nm reflectance
DyeShadesReflectance31(1,1,3)=48.675 -> DYE-1 1st shade 410nm reflectance

•
•
•

DyeShadesReflectance31(1,1,31)= 74.81 -> DYE-1 1st shade 690nm reflectance
DyeShadesReflectance31(1,1,32)= 75.06 -> DYE-1 1st shade 700nm reflectance

Later,

DyeShadesReflectance31(1,2,1)=0.0075 -> DYE-1 2nd shade concentration
DyeShadesReflectance31(1,2,2)=45.5 -> DYE-1 2nd shade 400nm reflectance
DyeShadesReflectance31(1,2,3)=43.945 -> DYE-1 2nd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(1,2,31)= 75.415 -> DYE-1 2nd shade 690nm reflectance
DyeShadesReflectance31(1,2,32)= 75.58 -> DYE-1 2nd shade 700nm reflectance

Later

DyeShadesReflectance31(1,3,1)=0.01 -> DYE-1 3rd shade concentration
DyeShadesReflectance31(1,3,2)=42.36 -> DYE-1 3rd shade 400nm reflectance
DyeShadesReflectance31(1,3,3)=40.485 -> DYE-1 3rd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(1,3,31)= 76.02 -> DYE-1 3rd shade 690nm reflectance
DyeShadesReflectance31(1,3,32)= 76.23 -> DYE-1 3rd shade 700nm reflectance

Later

DyeShadesReflectance31(1,4,1)=0.05 -> DYE-1 4th shade concentration
DyeShadesReflectance31(1,4,2)=25.435 -> DYE-1 4th shade 400nm reflectance
DyeShadesReflectance31(1,4,3)=21.91 -> DYE-1 4th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(1,4,31)= 75.19 -> DYE-1 4th shade 690nm reflectance
DyeShadesReflectance31(1,4,32)= 75.255 -> DYE-1 4th shade 700nm reflectance

Later

DyeShadesReflectance31(1,5,1)=0.1 -> DYE-1 5th shade concentration
DyeShadesReflectance31(1,5,2)=15.09 -> DYE-1 5th shade 400nm reflectance
DyeShadesReflectance31(1,5,3)=12.415 -> DYE-1 5th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(1,5,31)= 73.425 -> DYE-1 5th shade 690nm reflectance
DyeShadesReflectance31(1,5,32)= 73.825 -> DYE-1 5th shade 700nm reflectance

Later

DyeShadesReflectance31(1,6,1)=0.5 -> DYE-1 6th shade concentration
DyeShadesReflectance31(1,6,2)=4.68 -> DYE-1 6th shade 400nm reflectance
DyeShadesReflectance31(1,6,3)=3.5 -> DYE-1 6th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(1,6,31)= 74.635 -> DYE-1 6th shade 690nm reflectance
DyeShadesReflectance31(1,6,32)= 74.7 -> DYE-1 6th shade 700nm reflectance

Later

DyeShadesReflectance31(1,7,1)=1 -> DYE-1 7th shade concentration
DyeShadesReflectance31(1,7,2)=1.71 -> DYE-1 7th shade 400nm reflectance
DyeShadesReflectance31(1,7,3)=1.475 -> DYE-1 7th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(1,7,31)= 76.04 -> DYE-1 7th shade 690nm reflectance
DyeShadesReflectance31(1,7,32)= 76.16 -> DYE-1 7th shade 700nm reflectance

Later

DyeShadesReflectance31(1,8,1)=0 -> DYE-1 8th shade is absent
DyeShadesReflectance31(1,8,2)=0 -> DYE-1 8th shade is absent
DyeShadesReflectance31(1,8,3)=0 -> DYE-1 8th shade is absent

•
•
•

DyeShadesReflectance31(1,8,31)= 0 -> DYE-1 8th shade is absent
DyeShadesReflectance31(1,8,32)= 0 -> DYE-1 8th shade is absent

Later

DyeShadesReflectance31(1,9,1)=0 -> DYE-1 9th shade is absent
DyeShadesReflectance31(1,9,2)=0 -> DYE-1 9th shade is absent
DyeShadesReflectance31(1,9,3)=0 -> DYE-1 9th shade is absent

•
•
•

DyeShadesReflectance31(1,9,31)= 0 -> DYE-1 9th shade is absent
DyeShadesReflectance31(1,9,32)= 0 -> DYE-1 9th shade is absent

Later

DyeShadesReflectance31(1,10,1)=0 -> DYE-1 10th shade is absent
DyeShadesReflectance31(1,10,2)=0 -> DYE-1 10th shade is absent
DyeShadesReflectance31(1,10,3)=0 -> DYE-1 10th shade is absent

•
•
•

DyeShadesReflectance31(1,10,31)= 0 -> DYE-1 10th shade is absent
DyeShadesReflectance31(1,10,32)= 0 -> DYE-1 10th shade is absent

Later

DyeShadesReflectance31(1,11,1)=0 -> DYE-1 11th shade is absent
DyeShadesReflectance31(1,11,2)=0 -> DYE-1 11th shade is absent
DyeShadesReflectance31(1,11,3)=0 -> DYE-1 11th shade is absent

•
•
•

DyeShadesReflectance31(1,11,31)= 0 -> DYE-1 11th shade is absent
DyeShadesReflectance31(1,11,32)= 0 -> DYE-1 11th shade is absent

Later

DyeShadesReflectance31(1,12,1)=0 -> DYE-1 12th shade is absent
DyeShadesReflectance31(1,12,2)=0 -> DYE-1 12th shade is absent
DyeShadesReflectance31(1,12,3)=0 -> DYE-1 12th shade is absent

•
•
•

DyeShadesReflectance31(1,12,31)= 0 -> DYE-1 12th shade is absent
DyeShadesReflectance31(1,12,32)= 0 -> DYE-1 12th shade is absent

Later

DyeShadesReflectance31(1,13,1)=0 -> DYE-1 13th shade is absent
DyeShadesReflectance31(1,13,2)=0 -> DYE-1 13th shade is absent
DyeShadesReflectance31(1,13,3)=0 -> DYE-1 13th shade is absent

•
•
•

DyeShadesReflectance31(1,13,31)= 0 -> DYE-1 13th shade is absent
DyeShadesReflectance31(1,13,32)= 0 -> DYE-1 13th shade is absent

Later

DyeShadesReflectance31(1,14,1)=0 -> DYE-1 14th shade is absent
DyeShadesReflectance31(1,14,2)=0 -> DYE-1 14th shade is absent
DyeShadesReflectance31(1,14,3)=0 -> DYE-1 14th shade is absent

•
•
•

DyeShadesReflectance31(1,14,31)= 0 -> DYE-1 14th shade is absent
DyeShadesReflectance31(1,14,32)= 0 -> DYE-1 14th shade is absent

Later

DyeShadesReflectance31(1,15,1)=0 -> DYE-1 15th shade is absent
DyeShadesReflectance31(1,15,2)=0 -> DYE-1 15th shade is absent
DyeShadesReflectance31(1,15,3)=0 -> DYE-1 15th shade is absent

•
•
•

DyeShadesReflectance31(1,15,31)= 0 -> DYE-1 15th shade is absent
DyeShadesReflectance31(1,15,32)= 0 -> DYE-1 15th shade is absent

DYE-1 shades placement into the array has been done. Now let's continue with DYE-2

DyeShadesReflectance31(2,1,1)=0.0075 -> DYE-2 1st shade concentration
DyeShadesReflectance31(2,1,2)=48.935 -> DYE-2 1st shade 400nm reflectance
DyeShadesReflectance31(2,1,3)=47.425 -> DYE-2 1st shade 410nm reflectance

•
•
•

DyeShadesReflectance31(2,1,31)= 75.255 -> DYE-2 1st shade 690nm reflectance
DyeShadesReflectance31(2,1,32)= 75.345 -> DYE-2 1st shade 700nm reflectance

Later,

DyeShadesReflectance31(2,2,1)=0.01 -> DYE-2 2nd shade concentration
DyeShadesReflectance31(2,2,2)=26.76 -> DYE-2 2nd shade 400nm reflectance
DyeShadesReflectance31(2,2,3)=23.17 -> DYE-2 2nd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(2,2,31)= 75.41 -> DYE-2 2nd shade 690nm reflectance
DyeShadesReflectance31(2,2,32)= 75.48 -> DYE-2 2nd shade 700nm reflectance

Later

DyeShadesReflectance31(2,3,1)=0.15 -> DYE-2 3rd shade concentration
DyeShadesReflectance31(2,3,2)=11.985 -> DYE-2 3rd shade 400nm reflectance
DyeShadesReflectance31(2,3,3)=9.015 -> DYE-2 3rd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(2,3,31)= 74.05 -> DYE-2 3rd shade 690nm reflectance
DyeShadesReflectance31(2,3,32)= 74.135 -> DYE-2 3rd shade 700nm reflectance

Later

DyeShadesReflectance31(2,4,1)=0.5 -> DYE-2 4th shade concentration
DyeShadesReflectance31(2,4,2)=4.955 -> DYE-2 4th shade 400nm reflectance
DyeShadesReflectance31(2,4,3)=3.49 -> DYE-2 4th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(2,4,31)= 72.585 -> DYE-2 4th shade 690nm reflectance
DyeShadesReflectance31(2,4,32)= 72.64 -> DYE-2 4th shade 700nm reflectance

Later

DyeShadesReflectance31(2,5,1)=1 -> DYE-2 5th shade concentration
DyeShadesReflectance31(2,5,2)=2.24 -> DYE-2 5th shade 400nm reflectance
DyeShadesReflectance31(2,5,3)=1.725 -> DYE-2 5th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(2,5,31)= 72.355 -> DYE-2 5th shade 690nm reflectance
DyeShadesReflectance31(2,5,32)= 72.475 -> DYE-2 5th shade 700nm reflectance

Later

DyeShadesReflectance31(2,6,1)=0 -> DYE-2 6th shade is absent
DyeShadesReflectance31(2,6,2)=0 -> DYE-2 6th shade is absent
DyeShadesReflectance31(2,6,3)=0 -> DYE-2 6th shade is absent

•
•
•

DyeShadesReflectance31(2,6,31)= 0 -> DYE-2 6th shade is absent
DyeShadesReflectance31(2,6,32)= 0 -> DYE-2 6th shade is absent

Later

DyeShadesReflectance31(2,7,1)=0 -> DYE-2 7th shade is absent
DyeShadesReflectance31(2,7,2)=0 -> DYE-2 7th shade is absent
DyeShadesReflectance31(2,7,3)=0 -> DYE-2 7th shade is absent

•
•
•

DyeShadesReflectance31(2,7,31)= 0 -> DYE-2 7th shade is absent
DyeShadesReflectance31(2,7,32)= 0 -> DYE-2 7th shade is absent

Later

DyeShadesReflectance31(2,8,1)=0 -> DYE-2 8th shade is absent
DyeShadesReflectance31(2,8,2)=0 -> DYE-2 8th shade is absent
DyeShadesReflectance31(2,8,3)=0 -> DYE-2 8th shade is absent

•
•
•

DyeShadesReflectance31(2,8,31)= 0 -> DYE-2 8th shade is absent
DyeShadesReflectance31(2,8,32)= 0 -> DYE-2 8th shade is absent

Later

DyeShadesReflectance31(2,9,1)=0 -> DYE-2 9th shade is absent
DyeShadesReflectance31(2,9,2)=0 -> DYE-2 9th shade is absent
DyeShadesReflectance31(2,9,3)=0 -> DYE-2 9th shade is absent

•
•
•

DyeShadesReflectance31(2,9,31)= 0 -> DYE-2 9th shade is absent
DyeShadesReflectance31(2,9,32)= 0 -> DYE-2 9th shade is absent

Later

DyeShadesReflectance31(2,10,1)=0 -> DYE-2 10th shade is absent
DyeShadesReflectance31(2,10,2)=0 -> DYE-2 10th shade is absent
DyeShadesReflectance31(2,10,3)=0 -> DYE-2 10th shade is absent

•
•
•

DyeShadesReflectance31(2,10,31)= 0 -> DYE-2 10th shade is absent
DyeShadesReflectance31(2,10,32)= 0 -> DYE-2 10th shade is absent

Later

DyeShadesReflectance31(2,11,1)=0 -> DYE-2 11th shade is absent
DyeShadesReflectance31(2,11,2)=0 -> DYE-2 11th shade is absent
DyeShadesReflectance31(2,11,3)=0 -> DYE-2 11th shade is absent

•
•
•

DyeShadesReflectance31(2,11,31)= 0 -> DYE-2 11th shade is absent
DyeShadesReflectance31(2,11,32)= 0 -> DYE-2 11th shade is absent

Later

DyeShadesReflectance31(2,12,1)=0 -> DYE-2 12th shade is absent
DyeShadesReflectance31(2,12,2)=0 -> DYE-2 12th shade is absent
DyeShadesReflectance31(2,12,3)=0 -> DYE-2 12th shade is absent

•
•
•

DyeShadesReflectance31(2,12,31)= 0 -> DYE-2 12th shade is absent
DyeShadesReflectance31(2,12,32)= 0 -> DYE-2 12th shade is absent

Later

DyeShadesReflectance31(2,13,1)=0 -> DYE-2 13th shade is absent
DyeShadesReflectance31(2,13,2)=0 -> DYE-2 13th shade is absent
DyeShadesReflectance31(2,13,3)=0 -> DYE-2 13th shade is absent

•
•
•

DyeShadesReflectance31(2,13,31)= 0 -> DYE-2 13th shade is absent
DyeShadesReflectance31(2,13,32)= 0 -> DYE-2 13th shade is absent

Later

DyeShadesReflectance31(2,14,1)=0 -> DYE-2 14th shade is absent
DyeShadesReflectance31(2,14,2)=0 -> DYE-2 14th shade is absent
DyeShadesReflectance31(2,14,3)=0 -> DYE-2 14th shade is absent

•
•
•

DyeShadesReflectance31(2,14,31)= 0 -> DYE-2 14th shade is absent
DyeShadesReflectance31(2,14,32)= 0 -> DYE-2 14th shade is absent

Later

DyeShadesReflectance31(2,15,1)=0 -> DYE-2 15th shade is absent
DyeShadesReflectance31(2,15,2)=0 -> DYE-2 15th shade is absent
DyeShadesReflectance31(2,15,3)=0 -> DYE-2 15th shade is absent

•
•
•

DyeShadesReflectance31(2,15,31)= 0 -> DYE-2 15th shade is absent
DyeShadesReflectance31(2,15,32)= 0 -> DYE-2 15th shade is absent

DYE-2 shades placement into the array has been done. Now let's continue with DYE-3

DyeShadesReflectance31(3,1,1)=0.05 -> DYE-3 1st shade concentration
DyeShadesReflectance31(3,1,2)=39.68 -> DYE-3 1st shade 400nm reflectance
DyeShadesReflectance31(3,1,3)=37.0 -> DYE-3 1st shade 410nm reflectance

•
•
•

DyeShadesReflectance31(3,1,31)= 74.925 -> DYE-3 1st shade 690nm reflectance
DyeShadesReflectance31(3,1,32)= 75.16 -> DYE-3 1st shade 700nm reflectance

Later,

DyeShadesReflectance31(3,2,1)=0.075 -> DYE-3 2nd shade concentration
DyeShadesReflectance31(3,2,2)=36.75 -> DYE-3 2nd shade 400nm reflectance
DyeShadesReflectance31(3,2,3)=33.345 -> DYE-3 2nd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(3,2,31)= 72.215 -> DYE-3 2nd shade 690nm reflectance
DyeShadesReflectance31(3,2,32)= 72.455 -> DYE-3 2nd shade 700nm reflectance

Later

DyeShadesReflectance31(3,3,1)=0.15 -> DYE-3 3rd shade concentration
DyeShadesReflectance31(3,3,2)=27.47 -> DYE-3 3rd shade 400nm reflectance
DyeShadesReflectance31(3,3,3)=23.68 -> DYE-3 3rd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(3,3,31)= 74.33 -> DYE-3 3rd shade 690nm reflectance
DyeShadesReflectance31(3,3,32)= 74.70 -> DYE-3 3rd shade 700nm reflectance

Later

DyeShadesReflectance31(3,4,1)=0.3 -> DYE-3 4th shade concentration
DyeShadesReflectance31(3,4,2)=20.065 -> DYE-3 4th shade 400nm reflectance
DyeShadesReflectance31(3,4,3)=16.115 -> DYE-3 4th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(3,4,31)= 73.43 -> DYE-3 4th shade 690nm reflectance
DyeShadesReflectance31(3,4,32)= 73.92 -> DYE-3 4th shade 700nm reflectance

Later

DyeShadesReflectance31(3,5,1)= 0 -> DYE-3 5th shade is absent
DyeShadesReflectance31(3,5,2)= 0 -> DYE-3 5th shade is absent
DyeShadesReflectance31(3,5,3)= 0 -> DYE-3 5th shade is absent

•
•
•

DyeShadesReflectance31(3,5,31)= 0 -> DYE-3 5th shade is absent
DyeShadesReflectance31(3,5,32)= 0 -> DYE-3 5th shade is absent

Later

DyeShadesReflectance31(3,6,1)=0 -> DYE-3 6th shade is absent
DyeShadesReflectance31(3,6,2)=0 -> DYE-3 6th shade is absent
DyeShadesReflectance31(3,6,3)=0 -> DYE-3 6th shade is absent

•
•
•

DyeShadesReflectance31(3,6,31)= 0 -> DYE-3 6th shade is absent
DyeShadesReflectance31(3,6,32)= 0 -> DYE-3 6th shade is absent

Later

DyeShadesReflectance31(3,7,1)=0 -> DYE-3 7th shade is absent
DyeShadesReflectance31(3,7,2)=0 -> DYE-3 7th shade is absent
DyeShadesReflectance31(3,7,3)=0 -> DYE-3 7th shade is absent

•
•
•

DyeShadesReflectance31(3,7,31)= 0 -> DYE-3 7th shade is absent
DyeShadesReflectance31(3,7,32)= 0 -> DYE-3 7th shade is absent

Later

DyeShadesReflectance31(3,8,1)=0 -> DYE-3 8th shade is absent
DyeShadesReflectance31(3,8,2)=0 -> DYE-3 8th shade is absent
DyeShadesReflectance31(3,8,3)=0 -> DYE-3 8th shade is absent

•
•
•

DyeShadesReflectance31(3,8,31)= 0 -> DYE-3 8th shade is absent
DyeShadesReflectance31(3,8,32)= 0 -> DYE-3 8th shade is absent

Later

DyeShadesReflectance31(3,9,1)=0 -> DYE-3 9th shade is absent
DyeShadesReflectance31(3,9,2)=0 -> DYE-3 9th shade is absent
DyeShadesReflectance31(3,9,3)=0 -> DYE-3 9th shade is absent

•
•
•

DyeShadesReflectance31(3,9,31)= 0 -> DYE-3 9th shade is absent
DyeShadesReflectance31(3,9,32)= 0 -> DYE-3 9th shade is absent

Later

DyeShadesReflectance31(3,10,1)=0 -> DYE-3 10th shade is absent
DyeShadesReflectance31(3,10,2)=0 -> DYE-3 10th shade is absent
DyeShadesReflectance31(3,10,3)=0 -> DYE-3 10th shade is absent

•
•
•

DyeShadesReflectance31(3,10,31)= 0 -> DYE-3 10th shade is absent
DyeShadesReflectance31(3,10,32)= 0 -> DYE-3 10th shade is absent

Later

DyeShadesReflectance31(3,11,1)=0 -> DYE-3 11th shade is absent
DyeShadesReflectance31(3,11,2)=0 -> DYE-3 11th shade is absent
DyeShadesReflectance31(3,11,3)=0 -> DYE-3 11th shade is absent

•
•
•

DyeShadesReflectance31(3,11,31)= 0 -> DYE-3 11th shade is absent
DyeShadesReflectance31(3,11,32)= 0 -> DYE-3 11th shade is absent

Later

DyeShadesReflectance31(3,12,1)=0 -> DYE-3 12th shade is absent
DyeShadesReflectance31(3,12,2)=0 -> DYE-3 12th shade is absent
DyeShadesReflectance31(3,12,3)=0 -> DYE-3 12th shade is absent

•
•
•

DyeShadesReflectance31(3,12,31)= 0 -> DYE-3 12th shade is absent
DyeShadesReflectance31(3,12,32)= 0 -> DYE-3 12th shade is absent

Later

DyeShadesReflectance31(3,13,1)=0 -> DYE-3 13th shade is absent
DyeShadesReflectance31(3,13,2)=0 -> DYE-3 13th shade is absent
DyeShadesReflectance31(3,13,3)=0 -> DYE-3 13th shade is absent

•
•
•

DyeShadesReflectance31(3,13,31)= 0 -> DYE-3 13th shade is absent
DyeShadesReflectance31(3,13,32)= 0 -> DYE-3 13th shade is absent

Later

DyeShadesReflectance31(3,14,1)=0 -> DYE-3 14th shade is absent
DyeShadesReflectance31(3,14,2)=0 -> DYE-3 14th shade is absent
DyeShadesReflectance31(3,14,3)=0 -> DYE-3 14th shade is absent

•
•
•

DyeShadesReflectance31(3,14,31)= 0 -> DYE-3 14th shade is absent
DyeShadesReflectance31(3,14,32)= 0 -> DYE-3 14th shade is absent

Later

DyeShadesReflectance31(3,15,1)=0 -> DYE-3 15th shade is absent
DyeShadesReflectance31(3,15,2)=0 -> DYE-3 15th shade is absent
DyeShadesReflectance31(3,15,3)=0 -> DYE-3 15th shade is absent

•
•
•

DyeShadesReflectance31(3,15,31)= 0 -> DYE-3 15th shade is absent
DyeShadesReflectance31(3,15,32)= 0 -> DYE-3 15th shade is absent

DYE-3 shades placement into the array has been done. Now let's continue with DYE-4

DyeShadesReflectance31(4,1,1)=0.05 -> DYE-4 1st shade concentration
DyeShadesReflectance31(4,1,2)=51.15 -> DYE-4 1st shade 400nm reflectance
DyeShadesReflectance31(4,1,3)=49.665 -> DYE-4 1st shade 410nm reflectance

•
•
•

DyeShadesReflectance31(4,1,31)= 72.79 -> DYE-4 1st shade 690nm reflectance
DyeShadesReflectance31(4,1,32)= 72.945 -> DYE-4 1st shade 700nm reflectance

Later,

DyeShadesReflectance31(4,2,1)=0.075 -> DYE-4 2nd shade concentration
DyeShadesReflectance31(4,2,2)=51.12 -> DYE-4 2nd shade 400nm reflectance
DyeShadesReflectance31(4,2,3)=48.85 -> DYE-4 2nd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(4,2,31)= 74.475 -> DYE-4 2nd shade 690nm reflectance
DyeShadesReflectance31(4,2,32)= 74.62 -> DYE-4 2nd shade 700nm reflectance

Later

DyeShadesReflectance31(4,3,1)=0.1 -> DYE-4 3rd shade concentration
DyeShadesReflectance31(4,3,2)=47.435 -> DYE-4 3rd shade 400nm reflectance
DyeShadesReflectance31(4,3,3)=44.16 -> DYE-4 3rd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(4,3,31)= 73.735 -> DYE-4 3rd shade 690nm reflectance
DyeShadesReflectance31(4,3,32)= 73.875 -> DYE-4 3rd shade 700nm reflectance

Later

DyeShadesReflectance31(4,4,1)=0.2 -> DYE-4 4th shade concentration
DyeShadesReflectance31(4,4,2)=40.4 -> DYE-4 4th shade 400nm reflectance
DyeShadesReflectance31(4,4,3)=35.33 -> DYE-4 4th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(4,4,31)= 72.705 -> DYE-4 4th shade 690nm reflectance
DyeShadesReflectance31(4,4,32)= 73.40 -> DYE-4 4th shade 700nm reflectance

Later

DyeShadesReflectance31(4,5,1)= 0.3 -> DYE-4 5th shade concentration
DyeShadesReflectance31(4,5,2)= 37.24 -> DYE-4 5th shade concentration
DyeShadesReflectance31(4,5,3)= 31.345 -> DYE-4 5th shade concentration

•
•
•

DyeShadesReflectance31(4,5,31)= 72.5 -> DYE-4 5th shade concentration
DyeShadesReflectance31(4,5,32)= 72.68 -> DYE-4 5th shade concentration

Later

DyeShadesReflectance31(4,6,1)=0 .4 -> DYE-4 6th shade concentration
DyeShadesReflectance31(4,6,2)=33.31 -> DYE-4 6th shade concentration
DyeShadesReflectance31(4,6,3)=27.015 -> DYE-4 6th shade concentration

•
•
•

DyeShadesReflectance31(4,6,31)= 71.95 -> DYE-4 6th shade concentration
DyeShadesReflectance31(4,6,32)= 72.15 -> DYE-4 6th shade concentration

Later

DyeShadesReflectance31(4,7,1)=0.5 -> DYE-4 7th shade concentration
DyeShadesReflectance31(4,7,2)=32.15 -> DYE-4 7th shade concentration
DyeShadesReflectance31(4,7,3)=25.395 -> DYE-4 7th shade concentration

•
•
•

DyeShadesReflectance31(4,7,31)=73.345 -> DYE-4 7th shade concentration
DyeShadesReflectance31(4,7,32)=73.565 -> DYE-4 7th shade concentration

Later

DyeShadesReflectance31(4,8,1)=0.6 -> DYE-4 8th shade concentration
DyeShadesReflectance31(4,8,2)=26.97 -> DYE-4 8th shade concentration
DyeShadesReflectance31(4,8,3)=20.28 -> DYE-4 8th shade concentration

•
•
•

DyeShadesReflectance31(4,8,31)= 72.36 -> DYE-4 8th shade concentration
DyeShadesReflectance31(4,8,32)= 72.635 -> DYE-4 8th shade concentration

Later

DyeShadesReflectance31(4,9,1)=0.7 -> DYE-4 9th shade concentration
DyeShadesReflectance31(4,9,2)=29.735 -> DYE-4 9th shade concentration
DyeShadesReflectance31(4,9,3)=22.95 -> DYE-4 9th shade concentration

•
•
•

DyeShadesReflectance31(4,9,31)= 72.65 -> DYE-4 9th shade concentration
DyeShadesReflectance31(4,9,32)= 73.445 -> DYE-4 9th shade concentration

Later

DyeShadesReflectance31(4,10,1)=0 -> DYE-4 10th shade is absent
DyeShadesReflectance31(4,10,2)=0 -> DYE-4 10th shade is absent
DyeShadesReflectance31(4,10,3)=0 -> DYE-4 10th shade is absent

•
•
•

DyeShadesReflectance31(4,10,31)= 0 -> DYE-4 10th shade is absent
DyeShadesReflectance31(4,10,32)= 0 -> DYE-4 10th shade is absent

Later

DyeShadesReflectance31(4,11,1)=0 -> DYE-4 11th shade is absent
DyeShadesReflectance31(4,11,2)=0 -> DYE-4 11th shade is absent
DyeShadesReflectance31(4,11,3)=0 -> DYE-4 11th shade is absent

•
•
•

DyeShadesReflectance31(4,11,31)= 0 -> DYE-4 11th shade is absent
DyeShadesReflectance31(4,11,32)= 0 -> DYE-4 11th shade is absent

Later

DyeShadesReflectance31(4,12,1)=0 -> DYE-4 12th shade is absent
DyeShadesReflectance31(4,12,2)=0 -> DYE-4 12th shade is absent
DyeShadesReflectance31(4,12,3)=0 -> DYE-4 12th shade is absent

•
•
•

DyeShadesReflectance31(4,12,31)= 0 -> DYE-4 12th shade is absent
DyeShadesReflectance31(4,12,32)= 0 -> DYE-4 12th shade is absent

Later

DyeShadesReflectance31(4,13,1)=0 -> DYE-4 13th shade is absent
DyeShadesReflectance31(4,13,2)=0 -> DYE-4 13th shade is absent
DyeShadesReflectance31(4,13,3)=0 -> DYE-4 13th shade is absent

•
•
•

DyeShadesReflectance31(4,13,31)= 0 -> DYE-4 13th shade is absent
DyeShadesReflectance31(4,13,32)= 0 -> DYE-4 13th shade is absent

Later

DyeShadesReflectance31(4,14,1)=0 -> DYE-4 14th shade is absent
DyeShadesReflectance31(4,14,2)=0 -> DYE-4 14th shade is absent
DyeShadesReflectance31(4,14,3)=0 -> DYE-4 14th shade is absent

•
•
•

DyeShadesReflectance31(4,14,31)= 0 -> DYE-4 14th shade is absent
DyeShadesReflectance31(4,14,32)= 0 -> DYE-4 14th shade is absent

Later

DyeShadesReflectance31(4,15,1)=0 -> DYE-4 15th shade is absent
DyeShadesReflectance31(4,15,2)=0 -> DYE-4 15th shade is absent
DyeShadesReflectance31(4,15,3)=0 -> DYE-4 15th shade is absent

•
•
•

DyeShadesReflectance31(4,15,31)= 0 -> DYE-4 15th shade is absent
DyeShadesReflectance31(4,15,32)= 0 -> DYE-4 15th shade is absent

DYE-4 shades placement into the array has been done. Now let's continue with DYE-5

DyeShadesReflectance31(5,1,1)=0.01 -> DYE-5 1st shade concentration
DyeShadesReflectance31(5,1,2)=56.98 -> DYE-5 1st shade 400nm reflectance
DyeShadesReflectance31(5,1,3)=59.625 -> DYE-5 1st shade 410nm reflectance

•
•
•

DyeShadesReflectance31(5,1,31)=90.035 -> DYE-5 1st shade 690nm reflectance
DyeShadesReflectance31(5,1,32)=94.02 -> DYE-5 1st shade 700nm reflectance

Later,

DyeShadesReflectance31(5,2,1)=0.05 -> DYE-5 2nd shade concentration
DyeShadesReflectance31(5,2,2)=51.47 -> DYE-5 2nd shade 400nm reflectance
DyeShadesReflectance31(5,2,3)=53.62 -> DYE-5 2nd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(5,2,31)= 83.785 -> DYE-5 2nd shade 690nm reflectance
DyeShadesReflectance31(5,2,32)= 99.195 -> DYE-5 2nd shade 700nm reflectance

Later

DyeShadesReflectance31(5,3,1)=0.1 -> DYE-5 3rd shade concentration
DyeShadesReflectance31(5,3,2)=48.06 -> DYE-5 3rd shade 400nm reflectance
DyeShadesReflectance31(5,3,3)=49.90 -> DYE-5 3rd shade 410nm reflectance

•
•
•

DyeShadesReflectance31(5,3,31)= 70.625 -> DYE-5 3rd shade 690nm reflectance
DyeShadesReflectance31(5,3,32)= 90.20 -> DYE-5 3rd shade 700nm reflectance

Later

DyeShadesReflectance31(5,4,1)=0.15 -> DYE-5 4th shade concentration
DyeShadesReflectance31(5,4,2)=43.635 -> DYE-5 4th shade 400nm reflectance
DyeShadesReflectance31(5,4,3)=45.30 -> DYE-5 4th shade 410nm reflectance

•
•
•

DyeShadesReflectance31(5,4,31)= 59.31 -> DYE-5 4th shade 690nm reflectance
DyeShadesReflectance31(5,4,32)= 79.97 -> DYE-5 4th shade 700nm reflectance

Later

DyeShadesReflectance31(5,5,1)= 0.3 -> DYE-5 5th shade concentration
DyeShadesReflectance31(5,5,2)= 38.2 -> DYE-5 5th shade concentration
DyeShadesReflectance31(5,5,3)= 39.51 -> DYE-5 5th shade concentration

•
•
•

DyeShadesReflectance31(5,5,31)= 39.73 -> DYE-5 5th shade concentration
DyeShadesReflectance31(5,5,32)= 59.0 -> DYE-5 5th shade concentration

Later

DyeShadesReflectance31(5,6,1)=0 .5 -> DYE-5 6th shade concentration
DyeShadesReflectance31(5,6,2)=32.015 -> DYE-5 6th shade concentration
DyeShadesReflectance31(5,6,3)=33.345 -> DYE-5 6th shade concentration

•
•
•

DyeShadesReflectance31(5,6,31)= 24.185 -> DYE-5 6th shade concentration
DyeShadesReflectance31(5,6,32)= 40.455 -> DYE-5 6th shade concentration

Later

DyeShadesReflectance31(5,7,1)=1 -> DYE-5 7th shade concentration
DyeShadesReflectance31(5,7,2)=22.775 -> DYE-5 7th shade concentration
DyeShadesReflectance31(5,7,3)=23.81 -> DYE-5 7th shade concentration

•
•
•

DyeShadesReflectance31(5,7,31)=8.955 -> DYE-5 7th shade concentration
DyeShadesReflectance31(5,7,32)=18.525 -> DYE-5 7th shade concentration

Later

DyeShadesReflectance31(5,8,1)=0 -> DYE-5 8th shade is absent
DyeShadesReflectance31(5,8,2)=0 -> DYE-5 8th shade is absent
DyeShadesReflectance31(5,8,3)=0 -> DYE-5 8th shade is absent

•
•
•

DyeShadesReflectance31(5,8,31)= 0 -> DYE-5 8th shade is absent
DyeShadesReflectance31(5,8,32)= 0 -> DYE-5 8th shade is absent

Later

DyeShadesReflectance31(5,9,1)= 0 -> DYE-5 9th shade is absent
DyeShadesReflectance31(5,9,2)= 0 -> DYE-5 9th shade is absent
DyeShadesReflectance31(5,9,3)= 0 -> DYE-5 9th shade is absent

•
•
•

DyeShadesReflectance31(5,9,31)= 0 -> DYE-5 9th shade is absent
DyeShadesReflectance31(5,9,32)= 0 -> DYE-5 9th shade is absent

Later

DyeShadesReflectance31(5,10,1)=0 -> DYE-5 10th shade is absent
DyeShadesReflectance31(5,10,2)=0 -> DYE-5 10th shade is absent
DyeShadesReflectance31(5,10,3)=0 -> DYE-5 10th shade is absent

•
•
•

DyeShadesReflectance31(5,10,31)= 0 -> DYE-5 10th shade is absent
DyeShadesReflectance31(5,10,32)= 0 -> DYE-5 10th shade is absent

Later

DyeShadesReflectance31(5,11,1)=0 -> DYE-5 11th shade is absent
DyeShadesReflectance31(5,11,2)=0 -> DYE-5 11th shade is absent
DyeShadesReflectance31(5,11,3)=0 -> DYE-5 11th shade is absent

•
•
•

DyeShadesReflectance31(5,11,31)= 0 -> DYE-5 11th shade is absent
DyeShadesReflectance31(5,11,32)= 0 -> DYE-5 11th shade is absent

Later

DyeShadesReflectance31(5,12,1)=0 -> DYE-5 12th shade is absent
DyeShadesReflectance31(5,12,2)=0 -> DYE-5 12th shade is absent

DyeShadesReflectance31(5,12,3)=0 -> DYE-5 12th shade is absent

•
•
•

DyeShadesReflectance31(5,12,31)= 0 -> DYE-5 12th shade is absent
DyeShadesReflectance31(5,12,32)= 0 -> DYE-5 12th shade is absent

Later

DyeShadesReflectance31(5,13,1)=0 -> DYE-5 13th shade is absent
DyeShadesReflectance31(5,13,2)=0 -> DYE-5 13th shade is absent
DyeShadesReflectance31(5,13,3)=0 -> DYE-5 13th shade is absent

•
•
•

DyeShadesReflectance31(5,13,31)= 0 -> DYE-5 13th shade is absent
DyeShadesReflectance31(5,13,32)= 0 -> DYE-5 13th shade is absent

Later

DyeShadesReflectance31(5,14,1)=0 -> DYE-5 14th shade is absent
DyeShadesReflectance31(5,14,2)=0 -> DYE-5 14th shade is absent
DyeShadesReflectance31(5,14,3)=0 -> DYE-5 14th shade is absent

•
•
•

DyeShadesReflectance31(5,14,31)= 0 -> DYE-5 14th shade is absent
DyeShadesReflectance31(5,14,32)= 0 -> DYE-5 14th shade is absent

Later

DyeShadesReflectance31(5,15,1)=0 -> DYE-5 15th shade is absent
DyeShadesReflectance31(5,15,2)=0 -> DYE-5 15th shade is absent
DyeShadesReflectance31(5,15,3)=0 -> DYE-5 15th shade is absent

•
•
•

DyeShadesReflectance31(5,15,31)= 0 -> DYE-5 15th shade is absent
DyeShadesReflectance31(5,15,32)= 0 -> DYE-5 15th shade is absent

DYE-5 shades placement into the array has been done. Now let's continue with DYE-6

DyeShadesReflectance31(6,1,1)=0 -> DYE-6 1st shade is absent
DyeShadesReflectance31(6,1,2)=0 -> DYE-6 1st shade is absent
DyeShadesReflectance31(6,1,3)=0 -> DYE-6 1st shade is absent

•
•
•

DyeShadesReflectance31(6,1,31)= 0 -> DYE-6 1st shade is absent
DyeShadesReflectance31(6,1,32)= 75.16 -> DYE-6 1st shade is absent

Later,

DyeShadesReflectance31(6,2,1)=0 -> DYE-6 2nd shade is absent
DyeShadesReflectance31(6,2,2)=0 -> DYE-6 2nd shade is absent
DyeShadesReflectance31(6,2,3)=0 -> DYE-6 2nd shade is absent

•
•
•

DyeShadesReflectance31(6,2,31)= 0 -> DYE-6 2nd shade is absent
DyeShadesReflectance31(6,2,32)= 0 -> DYE-6 2nd shade is absent

Later

DyeShadesReflectance31(6,3,1)=0 -> DYE-6 3rd shade is absent
DyeShadesReflectance31(6,3,2)=0 -> DYE-6 3rd shade is absent
DyeShadesReflectance31(6,3,3)=0 -> DYE-6 3rd shade is absent

•
•
•

DyeShadesReflectance31(6,3,31)= 0 -> DYE-6 3rd shade is absent
DyeShadesReflectance31(6,3,32)= 0 -> DYE-6 3rd shade is absent

Later

DyeShadesReflectance31(6,4,1)=0 -> DYE-6 4th shade is absent
DyeShadesReflectance31(6,4,2)=0 -> DYE-6 4th shade is absent
DyeShadesReflectance31(6,4,3)=0 -> DYE-6 4th shade is absent

•
•
•

DyeShadesReflectance31(6,4,31)= 0 -> DYE-6 4th shade is absent
DyeShadesReflectance31(6,4,32)= 0 -> DYE-6 4th shade is absent

Later

DyeShadesReflectance31(6,5,1)= 0 -> DYE-6 5th shade is absent
DyeShadesReflectance31(6,5,2)= 0 -> DYE-6 5th shade is absent
DyeShadesReflectance31(6,5,3)= 0 -> DYE-6 5th shade is absent

•
•
•

DyeShadesReflectance31(6,5,31)= 0 -> DYE-6 5th shade is absent
DyeShadesReflectance31(6,5,32)= 0 -> DYE-6 5th shade is absent

Later

DyeShadesReflectance31(6,6,1)=0 -> DYE-6 6th shade is absent
DyeShadesReflectance31(6,6,2)=0 -> DYE-6 6th shade is absent

DyeShadesReflectance31(6,6,3)=0 -> DYE-6 6th shade is absent

•
•
•

DyeShadesReflectance31(6,6,31)= 0 -> DYE-6 6th shade is absent
DyeShadesReflectance31(6,6,32)= 0 -> DYE-6 6th shade is absent

Later

DyeShadesReflectance31(6,7,1)=0 -> DYE-6 7th shade is absent
DyeShadesReflectance31(6,7,2)=0 -> DYE-6 7th shade is absent
DyeShadesReflectance31(6,7,3)=0 -> DYE-6 7th shade is absent

•
•
•

DyeShadesReflectance31(6,7,31)= 0 -> DYE-6 7th shade is absent
DyeShadesReflectance31(6,7,32)= 0 -> DYE-6 7th shade is absent

Later

DyeShadesReflectance31(6,8,1)=0 -> DYE-6 8th shade is absent
DyeShadesReflectance31(6,8,2)=0 -> DYE-6 8th shade is absent
DyeShadesReflectance31(6,8,3)=0 -> DYE-6 8th shade is absent

•
•
•

DyeShadesReflectance31(6,8,31)= 0 -> DYE-6 8th shade is absent
DyeShadesReflectance31(6,8,32)= 0 -> DYE-6 8th shade is absent

Later

DyeShadesReflectance31(6,9,1)=0 -> DYE-6 9th shade is absent
DyeShadesReflectance31(6,9,2)=0 -> DYE-6 9th shade is absent
DyeShadesReflectance31(6,9,3)=0 -> DYE-6 9th shade is absent

•
•
•

DyeShadesReflectance31(6,9,31)= 0 -> DYE-6 9th shade is absent
DyeShadesReflectance31(6,9,32)= 0 -> DYE-6 9th shade is absent

Later

DyeShadesReflectance31(6,10,1)=0 -> DYE-6 10th shade is absent
DyeShadesReflectance31(6,10,2)=0 -> DYE-6 10th shade is absent
DyeShadesReflectance31(6,10,3)=0 -> DYE-6 10th shade is absent

•
•
•

DyeShadesReflectance31(6,10,31)= 0 -> DYE-6 10th shade is absent
DyeShadesReflectance31(6,10,32)= 0 -> DYE-6 10th shade is absent

Later

DyeShadesReflectance31(6,11,1)=0 -> DYE-6 11th shade is absent
DyeShadesReflectance31(6,11,2)=0 -> DYE-6 11th shade is absent
DyeShadesReflectance31(6,11,3)=0 -> DYE-6 11th shade is absent

•
•
•

DyeShadesReflectance31(6,11,31)= 0 -> DYE-6 11th shade is absent
DyeShadesReflectance31(6,11,32)= 0 -> DYE-6 11th shade is absent

Later

DyeShadesReflectance31(6,12,1)=0 -> DYE-6 12th shade is absent
DyeShadesReflectance31(6,12,2)=0 -> DYE-6 12th shade is absent
DyeShadesReflectance31(6,12,3)=0 -> DYE-6 12th shade is absent

•
•
•

DyeShadesReflectance31(6,12,31)= 0 -> DYE-6 12th shade is absent
DyeShadesReflectance31(6,12,32)= 0 -> DYE-6 12th shade is absent

Later

DyeShadesReflectance31(6,13,1)=0 -> DYE-6 13th shade is absent
DyeShadesReflectance31(6,13,2)=0 -> DYE-6 13th shade is absent
DyeShadesReflectance31(6,13,3)=0 -> DYE-6 13th shade is absent

•
•
•

DyeShadesReflectance31(6,13,31)= 0 -> DYE-6 13th shade is absent
DyeShadesReflectance31(6,13,32)= 0 -> DYE-6 13th shade is absent

Later

DyeShadesReflectance31(6,14,1)=0 -> DYE-6 14th shade is absent
DyeShadesReflectance31(6,14,2)=0 -> DYE-6 14th shade is absent
DyeShadesReflectance31(6,14,3)=0 -> DYE-6 14th shade is absent

•
•
•

DyeShadesReflectance31(6,14,31)= 0 -> DYE-6 14th shade is absent
DyeShadesReflectance31(6,14,32)= 0 -> DYE-6 14th shade is absent

Later

DyeShadesReflectance31(6,15,1)=0 -> DYE-6 15th shade is absent
DyeShadesReflectance31(6,15,2)=0 -> DYE-6 15th shade is absent

DyeShadesReflectance31(6,15,3)=0 -> DYE-6 15th shade is absent

•
•
•

DyeShadesReflectance31(6,15,31)= 0 -> DYE-6 15th shade is absent
DyeShadesReflectance31(6,15,32)= 0 -> DYE-6 15th shade is absent

Three dimensional array DyeShadesReflectance31 has been filled with necessary values.

Parameter 2 -> ShadesSubstrateReflectance31() As Single

An array with 31 elements having reflectance values between 400nm-700nm with 10nm intervals of the substrate which was used while preparing shades.

Example: ShadesSubstrateReflectance31(1)=81.345 'Reflectance value for 400nm
ShadesSubstrateReflectance31(2)=90.395 'Reflectance value for 410nm

•
•
•

ShadesSubstrateReflectance31(30)=94.123 'Reflectance value for 690nm
ShadesSubstrateReflectance31(31)=95.523 'Reflectance value for 700nm

Important!: In order to get more precious recipe calculation, the substrate should be processed as "Blank Dyeing" and then measured. Blank dyeing means a dyeing process without dyestuff but also with same chemicals (auxiliaries) with calibration shades in same machine of calibration shades. Because white substrate gets a little tint due to chemicals.

Parameter 3 -> TargetSubstrateReflectance31() As Single

An array with 31 elements having reflectance values between 400nm-700nm with 10nm intervals of the substrate which will be used for dyeing of the recipe to be calculated. This substrate may be the same substrate with ShadesSubstrateReflectance31().

Example: TargetSubstrateReflectance31(1)=82.345 'Reflectance value for 400nm
TargetSubstrateReflectance31(2)=91.395 'Reflectance value for 410nm

•
•
•

TargetSubstrateReflectance31(30)=93.123 'Reflectance value for 690nm
TargetSubstrateReflectance31(31)=96.523 'Reflectance value for 700nm

Important!: In order to get more precious recipe calculation, the substrate should be processed as "Blank Dyeing" and then measured. Blank dyeing means a dyeing process without dyestuff but also with same chemicals (auxiliaries) with calibration shades in same machine of calibration shades. Because white substrate gets a little tint due to chemicals.

Parameter 4 -> TargetColorReflectance31() As Single

An array with 31 elements having reflectance values between 400nm-700nm with 10nm intervals of the targeted color whose recipe will be calculated.

Example: TargetColorReflectance31(1)=82.345 'Reflectance value for 400nm
TargetColorReflectance31(2)=91.395 'Reflectance value for 410nm
.
.
.
TargetColorReflectance31(30)=93.123 'Reflectance value for 690nm
TargetColorReflectance31(31)=96.523 'Reflectance value for 700nm

Parameter 5 -> NumberOfDyesInEachRecipe As Integer

Represent how many dyes will be used for the each recipe which will be calculated. It is between 1 and 6 but commonly 3 is used.

Another saying that, you may pass up to 6 dyes and may wish to create many recipe alternatives with 3 dyes in each recipes.

Parameter 6 -> OutputTextFileName As String

All calculated recipies is stored in a text file with a name given with this parameter. For example; let's assume that you assigned "RECIPES" name with this parameter and DLL has calculated more then one alternative recipies with different combinations with different dyes you passed, then recipes will be stored automatically RECIPES1.TXT, RECIPES2.TXT, RECIPESn.TXT etc. You can also specify a name including path.

Each text file has only one recipe alternatives with the same format as follows.

```
DYE  1                          %C=0.05575
DYE  3                          %C=0.00198
DYE  4                          %C=0.21448
Targeted reflectances 400-700nm   -> 14.81 - 11.30 - 9.09 - 8.06 - 8.26 - 9.48 - 10.35 - 10.02 - 8.70 -
7.37 - 6.21 - 5.33 - 4.77 - 4.40 - 4.25 - 4.37 - 4.67 - 4.88 - 4.85 - 4.45 - 3.66 - 2.80 - 2.26 - 2.19 - 2.66 -
4.02 - 6.99 - 12.29 - 20.20 - 30.12 - 41.63 -
Calculated reflectances 400-700nm -> 15.11 - 12.22 - 10.08 - 8.96 - 8.09 - 7.69 - 7.87 - 7.75 - 7.49 - 7.36
- 7.44 - 7.68 - 8.24 - 9.11 - 10.44 - 12.42 - 15.28 - 19.09 - 23.99 - 29.94 - 36.50 - 43.22 - 49.69 - 55.08 -
59.34 - 62.53 - 64.65 - 65.91 - 66.62 - 66.94 - 67.14 -
[CIELab] DELTA E (D65)= 62.63
```

Let's explain the content of these text files:

First three lines represents dye number and corresponding amount of it as %C. The number of dyes will vary according to input parameter *NumberOfDyesInEachRecipe* naturally.

The line *"Targeted reflectances 400-700nm"* represent reflectance values given in input parameter array *TargetColorReflectance31.*

The line *"Calculated reflectances 400-700nm"* represent reflectance values calculated by DLL subroutine.

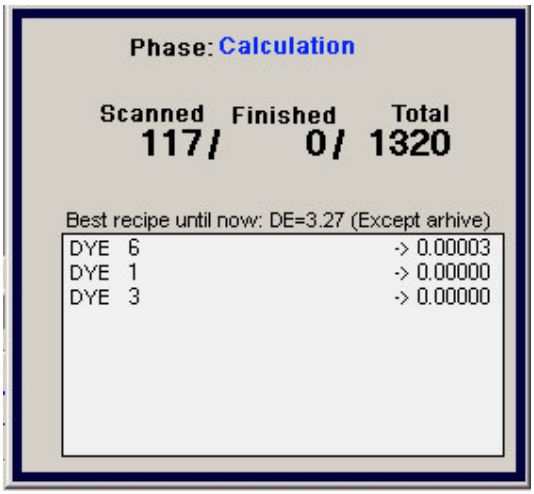
Last line represents DELTA E differences in CIELAB color space [D65/10] between targeted and calculated reflectance values.

Parameter 7 -> Screen as Boolean

Indicates either an information screen which shows some details will be displayed or not. This information screen includes some details (like number of recipe alternatives in process, best recipe Delta E found, number of recipe in que etc). which helps user to view the process in real time.

If this parameter set to “True” the screen is displayed. If it is set to “False” then the screen will not be displayed and the calculation is performed in background. The screen is vanished automatically when the calculation finished.

A sample information screen is as follows.



Parametre 8 -> InformationY as Integer

Indicates Y coordinate of the information screen if it is set to true in parameter 6.

Parametre 9 -> InformationX as Integer

Indicates X coordinate of the information screen if it is set to true in parameter 6.

Output parameters:

Parameter 1 -> Result_NumberOfAlternatives As Integer

This parameter returns how many alternatives has been calculated and so how many text file was created which stores the recipes details.

Notes **: 1)** Reflectance values are % values. This means that all values are between 0 to 100.

2) Recipe calculation is a patent pending method created by Artoksi. This subroutine does not use Kubelka-Munk matrix systems.

Howewer, this subroutine is suitable for exhaust dyeing (such as textile) not pigment dyeing.

Examples **:**

*Anyname.*REFLECTANCE31_RECIPE_CALCULATE (CalibrationData(), Substrate1(), Substrate2(), TobeCalculatedColor31(), 3, “Alternaives”, True, 1300,900,HowMany)

A recipe or recipes will be calculated according to shades given in three dimensional array *CalibrationData* and Substrate1 and Substrate2. Each recipe will contain maximum 3 dyes in it. Each alternative will be stored on disk with name "*Alternatives*" plus number suffix automatically and number of alternatives will be output via variable *HowMany.* An information screen is also displayed according to the coordinates set as Y=1300 and X=900.

*Anyname.*REFLECTANCE31_RECIPE_CALCULATE (LabData(), Substrate1(), Substrate1(), TobeCalculatedColor31(), 3, "Recipes", False,0,0,Pcs)

A recipe or recipes will be calculated according to shades given in three dimensional array *LabData.* Same substrate which was used during shades preparation will be used. Each recipe will contain maximum 3 dyes in it. Each alternative will be stored on disk with filename "*Recipes*" plus number suffix automatically as text files and number of alternatives will be output via variable *Pcs.* No information screen is displayed.

Subroutine no : 34

Subroutine name : REFLECTANCE_KS_CONVERT

Description : Calculates K/S absorbtion value from reflectance at a certain wavelength

Purpose : Send a reflectance value and get K/S value

General form : REFLECTANCE_KS_CONVERT(Reflectance As Single) As Single

Input parameters:

Parameter 1 -> Reflectance As Single
A single reflectance value at a certain wavelength

Output parameters:

Parameter 1 -> REFLECTANCE_KS_CONVERT
Unlike the other subroutines above this one has been designed as a function. This means that output parameter is the function name REFLECTANCE_KS_CONVERT

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

KS=*Anyname.* REFLECTANCE_KS_CONVERT(65.12)

KS value will be calculated for the reflectance %65.12 and result will be output via variable KS

Result=*Anyname.* REFLECTANCE_KS_CONVERT(15.32)

KS value will be calculated for the reflectance %15.32 and result will be output via variable result

Subroutine no : 35

Subroutine name : KS_REFLECTANCE_CONVERT

Description : Calculates reflectance value from a K/S value.

Purpose : Send a K/S value and get reflectance value

General form : KS_REFLECTANCE_CONVERT(KS As Single) As Single

Input parameters:

Parameter 1 -> KS As Single

A single K/S value

Output parameters:

Parameter 1 -> KS_REFLECTANCE_CONVERT

Unlike the other first 30 subroutines above, this one has been designed as a function also. This means that output parameter is the function name KS_REFLECTANCE_CONVERT

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

REF=*Anyname.*KS_ REFLECTANCE_CONVERT(4.16)

Reflectance value will be calculated for the KS=4.16 and result will be output via variable REF.

Result=*Anyname.*KS_ REFLECTANCE_CONVERT(41.65)

Reflectance value will be calculated for the KS=41.65 and result will be output via variable Result.

Subroutine no : 36

Subroutine name : VERSION

Description : Returns version number of DLL

Purpose : To identify you are up-to-date.

General form : VERSION() As String

Input parameters:

No input parameter exist

Output parameters:

Parameter 1 -> VERSION

Like the other subroutines above number 31 and 32, this one has been designed as a function also. This means that output parameter is the function name VERSION

Notes : Output parameter are in string format so be sure that variable has been defined properly in your code.

Examples :

REF=*Anyname.*KS_ REFLECTANCE_CONVERT(4.16)

Reflectance value will be calculated for the KS=4.16 and result will be output via variable REF.

Result=*Anyname.*KS_ REFLECTANCE_CONVERT(41.65)

Reflectance value will be calculated for the KS=41.65 and result will be output via variable Result.

Subroutine no : 37

Subroutine name : CIELab_DELTA_ASSESSMENTS

Description : Returns the reason of difference between two colors

Purpose : Send CIELab values of two color and get the reason of differences.

General form : CIELab_DELTA_ASSESSMENTS (L_std As Single, a_std As Single, b_std As Single, L_Trial As Single, a_Trial As Single, b_Trial As Single, Result_Comment1 As String, Result_Comment2 As String, Result_Comment3 As String)

Input parameters:

Parameter 1 -> L_std As Single
L (Lightness) value at CIELab colorspace of the first color.

Parameter 2 -> a_std As Single
a (Red-Green) value at CIELab colorspace of the first color.

Parameter 3 -> b_std As Single
b (Blue-Yellow) value at CIELab colorspace of the first color.

Parameter 4-> L_std As Single
L (Lightness) value at CIELab colorspace of the second color

Parameter 5-> a_std As Single
a (Red-Green) value at CIELab colorspace of the second color.

Parameter 6-> b_std As Single
b (Blue-Yellow) value at CIELab colorspace of the second color.

Output parameters:

Parameter 1 -> Result_Comment1 As String
Returns one of two different strings. “Lighter” or “Darker”.

Parameter 2 -> Result_Comment2 As String
Returns one of four different strings. “Redish” , “Reder”, “Less red”, “Greenish”, “Greener”, “Less Green”.

Parameter 3 -> Result_Comment3 As String
Returns one of four different strings. “Blueish” , “Blueer”, “Less blue”, “Yellowish”, “Yellower”, “Less Yellow”.

Notes : All assessments result should be considered as “second color to first color”.

Examples :

Anyname. CIELab_DELTA_ASSESSMENTS(21.2, 12.3, 31.6, 23.4, 2.18 ,25.9 ,R1,R2,R3)

The reason of difference of two colors specifed as L1=21.2, a1=12.3, b1=31.6 and L2=23.4, a2=2.18, b2=25.9 will be explained via three out parameters R1, R2,R3

Anyname. CIELab_DELTA_ASSESSMENTS(2.2, -2.3, 31.6, 3.4, 2.18 ,25.9 ,Reason1, Reason2, Reason3)

The reason of difference of two colors specifed as L1=2.2, a1=-2.3, b1=31.6 and L2=3.4, a2=2.18, b2=25.9 will be explained via three out parameters Reason1, Reason2,Reason3

Subroutine no : 38

Subroutine name : RECIPE_REFLECTANCE40_CALCULATE

Description : Returns final color for a given recipe

Purpose : Send amount of dyes in a recipe and get what color will be output.

General form : RECIPE_REFLEKTANCE40_CALCULATE (DyeShadesReflectance31() As Single, ShadesSubstrateReflectance31() As Single, TargetSubstrateReflectance31() As Single, Amounts() As Single, Result_Reflectance40() As Single, Result_Error as String)

Input parameters:

Parameter 1 -> DyeShadesReflectance31() As Single

Three dimensional array with boundaries 6, 15 and 32 elements. Another saying that an array defined as DyeShadesReflectance31(1 to 6, 1 to 15 and 1 to 32). These three dimensional array is for shades' reflectance values (also known as calibration data) with different concentrations (%C) prepared in laboratory.

See REFLECTANCE31_RECIPE_CALCULATE function "Parameter 1" for detailed explanation.

Parameter 2 -> ShadesSubstrateReflectance31() As Single

An array with 31 elements having reflectance values between 400nm-700nm with 10nm intervals of the substrate which was used while preparing shades.

See REFLECTANCE31_RECIPE_CALCULATE function "Parameter 2" for detailed explanation.

Parameter 3 -> TargetSubstrateReflectance31() As Single

An array with 31 elements having reflectance values between 400nm-700nm with 10nm intervals of the substrate which will be used for dyeing of the recipe to be calculated. This substrate may be the same substrate with ShadesSubstrateReflectance31().

See REFLECTANCE31_RECIPE_CALCULATE function "Parameter 3" for detailed explanation.

Parameter 4 -> Amounts() As Single

An 6 element array consist of amount information of each dye in a recipe.

.

For example, assume that DyeShadesReflectance31() array (parameter 1) has data for 3 dyes and the amount of each dye is as follows.

DYE 1'den %C=2.4
DYE 2'den %C=0.55
DYE 3'den %C=1.187

In this case Amounts() array should be like this:

Amounts(1)=2.4
Amounts(2)=0.55
Amounts(3)=1.187
Amounts(4)=0
Amounts(5)=0
Amounts(6)=0

Output parameters:

Parameter 1 -> Result_Reflectance40() As Single

31 element array consist of Calculated reflectance values (360nm-750nm) of the color mixed resulting the recipe given in parameters.

Parameter 1 -> Result_Error As String

This is a string whose length is equal to number of dyes passed to this function. For example, if you have passed three dyes to the function then this parameter will return a string with 3 caharacters. This output parameter shows that, the calculated result is trustable or not. Each character of this parameter can only have one four information. These are “0”, “1”, “2” or “W”.

The place of these informations refers the number of dye. Another saying that, first digit represents the first dye, second digit represents the second dye, third digit represents the third dye and so on.

“0” means that the no problem has been encountered for that dye in order.

“1” indicates a problem. This generally occurs when the minimum %c of the shades of the dye is not as small as required. The solution is to improve dye file by preparing new shades with smaller %c then existing minimum %c.

“2” indicates a problem. This generally occurs when the maximum %c of the shades of the dye is not as big as required. The solution is to improve dye file by preparing new shades with bigger %c then existing maximum %c.

“W” indicates a problem. This generally occurs when two or more reflectance curves has a conflict. The bigger %C, the darker color. The darker color the smaller reflectance values in the range of 400nm-700nm wavelength. In some cases the reflectances values are out of this rule. This is due to measuring error or dyeing error of the shades. The solution is to re-measure the shades and after re-dye the shades if the problem persist

Notes : This function is opposite of REFLECTANCE31_RECIPE_CALCULATE.

Examples :

*Anyname.*RECIPE (Shades(), SubstrateOrg(), SubstrateNew(), Ingredients(), Result31())

Final mixed color reflectance values on SubstrateNew() between 360nm-750nm will be output according to the dye shades given in array Shades() on SubstrateOrg() and amount in array Amounts()

Subroutine no : 39

Subroutine name : RGB_CAPITALXYZ_CALCULATE

Description : Calculates tristimulus X, Y and Z values for a color specified with RGB values

Purpose : Send tristimulus X, Y and Z values and get R [red], G [green], B [blue] values

General form : RGB_CAPITALXYZ_CALCULATE (Red As Single, Green As Single, Blue As Single, Result_CAPITAL_X As Single, Result_CAPITAL_Y As Single, Result_CAPITAL_Z As Single)

Input parameters:

Parameter 1 -> Red As Single
A value between 0-255 indicating R [Red] component of the color.

Parameter 2 -> Yesil As Single
A value between 0-255 indicating G [Green] component of the color.

Parameter 3 -> Mavi As Single
A value between 0-255 indicating B [Blue] component of the color.

Output parameters :

Parameter 1 -> Result_CAPITAL_X As Single
Calculated tristimulus X value of the color

Parameter 2 -> Result_CAPITAL_Y As Single
Calculated tristimulus Y value of the color

Parameter 3 -> Result_CAPITAL_Z As Single
Calculated tristimulus X value of the color

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

Examples :

HerHangiBirIsim1.RGB_CAPITALXYZ_CALCULATE (124, 111, 98, TX, TY, TZ)

Calculates tristimulus X, Y and Z values of the color specified with R [red]=124, G [green]=111 and B [blue]=98 and the results will be in the variables TX, TY and TZ

Subroutine no : 40

Subroutine name : CAPITALXYZ_CIELab_CALCULATE

Description : Calculates L, a and b values for CIELab color space from the values tristimulus X, Y and Z.

Purpose : Send L, a and b values of CIELab color space and gets tristimulus X, Y and Z values.

General form : CAPITALXYZ_CIELab_CALCULATE (CAPITAL_X As Single, CAPITAL_Y As Single, CAPITAL_Z As Single, LightSource As String, Observer As Byte, Result_CIE_L As Single, Result_CIE_a As Single, Result_CIE_b As Single)

Input parameters:

Parameter 1 -> CAPITAL_X As Single
Tristimulus X value of the color

Parameter 2 -> CAPITAL_Y As Single
Tristimulus Y value of the color

Parameter 3 -> CAPITAL_Z As Single
Tristimulus Z value of the color

Parameter 4 -> LightSource As String
One of the options from "LIGHT SOURCES SELECTION REFERENCE TABLE" at the end of this document to tell the subroutine which light source will be used (simulated) to calculate brightness values

Parameter 5 -> Observer as Byte
One of the options below to tell the subroutine which observer for the specified light source will be used for calculation

2 -> Weighing factors for 2 degree observer will be used.
10 -> Weighing factors for 10 degree observer will be used.

Output parameters :

Parameter 1 -> Result_CIE_L As Single
Returns Lightness value "L" in CIE coordinate system L value is between 0 to 100. 100 represents lightest color and 0 (zero) represents darkest colors.

Parameter 2 -> Result_CIE_a As Single
Returns Greenish/Redish value "a" in CIE coordinate system.
Minus values represent green zone
Plus values represent red zone

Parameter 3 -> Result_CIE_b As Single
Returns Yellowish/Blueish value "b" in CIE coordinate system
Minus values represent blue zone
Plus values represent yellow zone

Notes : 1) As all output numeric parameters has decimal point be sure that the type of those parameters are defined properly in your source code. In VB6 we use "single" declaration type to do it. For example;

Dim *VariableName* as Single
or
Public *VariableName* as Single

2) Reflectance values are % values. This means that all values are between 0 to 100.

3) Output parameters Result_CIE_a and Result_CIE_b may return plus or minus value so be sure that variable has been defined properly in your code.

Examples :

HerHangiBirIsim1.CAPITALXYZ_CIELab_CALCULATE (22.3, 16.7, 31.34, "D65", 10, CIE_L, CIE_a, CIE_b)

Calculates L, a and b values for CIELab color space for the color specified with tristimulus values as X=22.3, Y=16.7, 31.34. The results will be presented via variables CIE_L, CIE_a, CIE_b.

SECTION 2
Color Measurement Instrument Control Commands Description

This chapter describes each subroutine in DLL

Önce beni oku:

1) All functions have been named to help to user to understand purpose of command at first view.

Example 1:
CALIBRATE_WHITE means.

This function is used to make the instrument white calibration

or
MEASURE means

This function tells to instrument to take a measurement.

1) This DLL supports following instruments

KoninaMinolta series
- CM2300d
- CM2500c
- CM2500d
- CM2600d
- CM3220d
- CM3600d
- CM3700d

GretagMacbeth series
- CE2020
- CE2020+
- CE2180
- CE2180UV
- CE3000
- CE3100
- CE7000
- CE7000A
- XTH

PenColorArt series
- PenColorArt USB

For other brands and model please write to artoksi@artoksi.com

3) Please keep in mind that each instrument has different features. So, paramaters (SCI/E, UV fitler, Transmission etc) in functions below may be meaningless for he instrument type specified. In such case, the function ignore that parameter if it has no effect on the instrument. We recommend you to check instrument status by calling function “INSTRUMENT_STATUS” after each use of “SET_INSTRUMENT” function.

4) All function in this chapter uses RS232 serial communication channel to communicate with the insturument. Using serial port is different at each programming language. Following method is for Visual Basic 6 as an example.

1) Adding MsComm object into your project as follows.

a) Select Project/Components from menu in VB6 and find "Microsoft Comm Control 6.0" and select it.

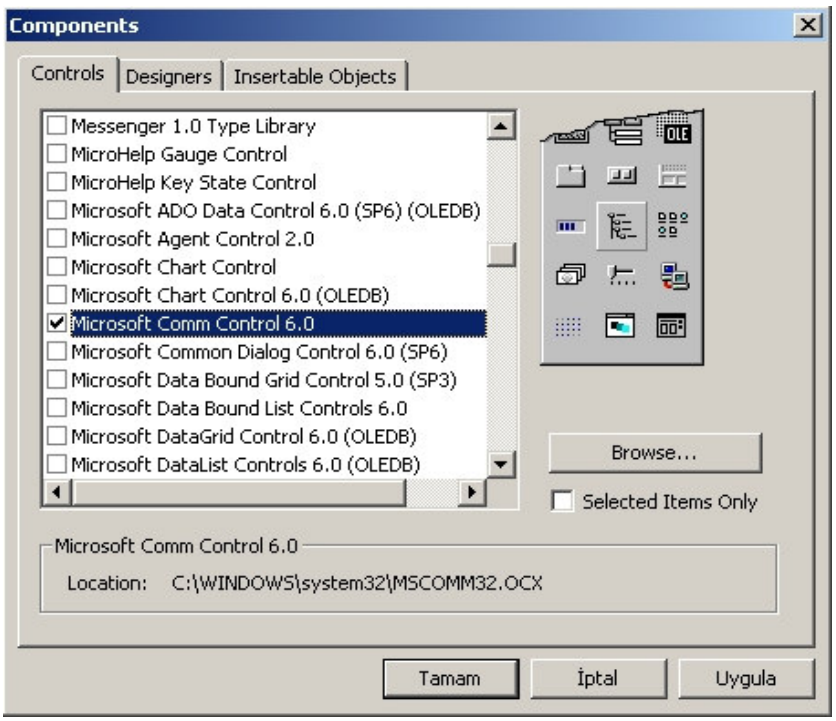


Figure 4: Sample object screen (MScomm object adding)

If you do it correctly, you will see a new object in *"Toolbox"* like in figure 5.

Figure 5: Sample screen (MScomm object was added corecctly)

b) There is one more step after this to use MScomm object. Just simply drag&drop this object into any form you wish to communicate with the instrument.

2) General process on instrument management are as follows.

a) Check instrument status (UV filter, lens, SCI/SCE settings, transmission/reflectance mode etc.) by using INSTRUMENT_STATUS command.

b) If the settings is not as you need, set the instrument as you want by using SET_INSTRUMENT command.

c) Re-check instrument status (UV filter, lens, SCI/SCE settings, transmission/reflectance mode etc.) by using INSTRUMENT_STATUS command.

d) Calibrate the instrument by using CALIBRATE_WHITE and or CALIBRATE_BLACK commands.

e) Measure the color by using MEASURE command

Subroutine no : 1

Subroutine name : SET_INSTRUMENT

Description : Lets you to change the instrument's settings (lens, UV filter, SCI/SCE etc) for a desired measurement type.

Purpose : Set instrument settings for a sample to be measured.

General form : SET_INSTRUMENT (ComObject As Object, ComPortNo As Byte, Instrument As Byte, Target_RT As String, Target_SCIE As String, Target_View As String, Target_UV As String) As String

Input parameters:

Parameter 1 -> ComObject As Object

Obejct full name of serial communication object. Name of the MsComm object in the Form.

For example;
Assume that, a form namely "Form1" and a MsComm object on it namely "MsComm1". In this case, this parameter will be like this:

Form1.MsComm1

Parameter 2 -> ComPortNo As Byte

A number between 1 to 255 indicating port number of serial commnunication channel

Parameter 3 -> Instrument As Byte

A number indicating which instrument you use. See "SUPPORTED INSTRUMENT SELECTION TABLE" at the end of this document.

Parametre 4 -> Target_RT As String

Measurement mode to be applied.
"0" -> Reflectance measurement
"1" -> Transmission measurement

Each instrument has no "Transmission mode". If the specified instrument does not have such a feature then this parameter will be ignored automaticly.

Parametre 5 -> Target_SCIE As String

Defines specular component status.
"0" -> SCI (specular component included)
"1" -> SCE (specular component excluded)

Each instrument has no "SCI mode". If the specified instrument does not have such a feature then this parameter will be ignored automaticly.

Parametre 6 -> Target_View As String

Sample measurement area selection.
"0" Large area view
1" Medium area view
2" Small area view
3" Very small area view

Each instrument can not support all these 4 modes. If the instrument has only one view option then this parameter will be ignored automaticly. If the instrument support more than one viewing mode but the view you specified is not supported by the instrument then function will return a failure information back to you.

Parametre 7 -> Target_UV As String

UV filter position adjustment.

Parameter	**UV filter position**
"0"	UV components included %100, (No UV filter)
"1"	UV components excluded 400nm
"2"	UV components excluded 400nm, low profile
"3"	UV components excluded 420nm
"4"	UV components excluded 420nm low profile
"5"	UV adjusted 400nm normal illumination
"6"	UV adjusted 400nm low illumination
"7"	UV adjusted 420nm normal illumination
"8"	UV adjusted 420nm low illumination
"A"	UV excluded %100

Output parameters:

Parameter 1 -> SET_INSTRUMENT As String

The name of this function is the output parameter saying "succesfully done" or not. If operation has been carried out succesfully then it will return a string as "OKEY". If operation is failed for any reason then it will return a string as "FAIL" with an explanation of reason of this failure.

Notes **: 1)** When any changes occured on instrument status, re-calibrating is a must. See CALIBRATE_WHITE and CALIBRATE_BLACK functions for more information about calibration the instrument.

2) Each instrument does not support all UV modes. If specified UV is unsupported UV mode for he instrument specified then function will return "FAIL" and following by the reason of the failure.

The UV modes comforms to instrument are listed in the table below.

Markası	**Model**	**UV tipi**	**Parametre**
KonicaMinolta	CM2300d	No UV mode	""
	CM2500c	No UV mode	""
	CM2600d	UV included %100	"0"
		UV excluded	"1"
		UV adjusted	"2"
	CM3220d	No UV mode	""
	CM3600d	UV included %100	"0"
		UV excluded 400nm	"1"
		UV excluded 400nm, low profile	"2"
		UV excluded, 420nm	"3"
		UV excluded, 420nm, low profile	"4"
		UV adjusted, 400nm, normal illumination	"5"
		UV adjusted, 400nm, low illumination	"6"
		UV adjusted, 420nm, normal illumination	"7"
		UV adjusted, 420nm, low illumination	"8"
	CM3700d	is being prepared	
GretagMacbeth	CE2180	is being prepared	""
	CE2180UV	is being prepared	
	CE3100	is being prepared	
	CE7000	is being prepared	
	CE7000A	is being prepared	

Examples :

Result=*Anyname.*SET_INSTRUMENT(Form1.MSComm1, 2, 6, “0”, “1”, “2”, “0”)

KonicaMinolta 3600d will be set to “Reflectance”, “SCE”, “small view” and “UV included” mode by using serial channel 2 via MsComm1 object in the form “Form1”

If the command is executed succesfully then variable “Result” will return “OKEY”. If it is not executed successfuly for a reason then it will return FAIL follwing by an explanation.

Example: "FAIL: Target_UV parameter is wrong"

Status=*Anyname.*SET_INSTRUMENT(Settings.MSComm3, 2, 12, “0”, “0”, “1”, “0”)

GretagMacbeth CE7000 will be set up to “reflectance”, “SCE”, “medium view”, and “UV included” mode by using serial channel 2 via MsComm3 object in the form “Settings”

If the command is executed succesfully then variable “Status” will return “OKEY”. If it is not executed successfuly for a reason then it will return FAIL follwing by an explanation.

Example: "FAIL: Target_RT parameter is wrong "

Subroutine no : 2

Subroutine name : SHOW

Description : Displays a message on instrument display if avilable.

Purpose : To show a message to the user

General form : SHOW (ComObject As Object, ComPortNo As Byte, Instrument As Byte, Size As Byte, Statement As String) As String

Input parameters:

Parameter 1 -> ComObject As Object

Obejct full name of serial communication object. Name of the MsComm object in the Form.

For example;
Assume that, a form namely "Form1" and a MsComm object on it namely "MsComm1". In this case, this parameter will be like this:

Form1.MsComm1

Parameter 2 -> ComPortNo As Byte

A number between 1 to 255 indicating port number of serial commnunication channel

Parameter 3 -> Instrument As Byte

A number indicating which instrument you use. See "SUPPORTED INSTRUMENT SELECTION TABLE" at the end of this document.

Parametre 4 -> Size As Byte

The character size of text to be displayed.
1 ……. Normal size
2 ……. Double size

Parametre 5 -> Statement As String

Message to be displayed. If the character size specified is 1 (normal size) then maximum length is 240 caharcter. If the character size specified is 2 (double size) then maximum length is 120 caharcter.

Çıkış parametreleri:

Parametre 1 -> SHOW As String

The name of this function is the output parameter saying "succesfully done" or not. If operation has been carried out succesfully then it will return a string as "OKEY". If operation is failed for any reason then it will return a string as "FAIL" with an explanation of reason of this failure.

Notes : This command is not supported by all instrtuments. When this document printed the model was KonicaMinolta 2300d, 2500d ve 2600d portable models

Example :

Sonuc=*Anyname2*.SHOW (Form1.MSComm1, 2, 3, "1","Place the sample")

A message "Place the sample" with double sized character will be sent to KonicaMinolta 2500d's display by using serial channel 2 and MsComm1 object in the form "Form1"

If the command is executed succesfully then variable “Result” will return “OKEY”. If it is not executed successfuly for a reason then it will return FAIL follwing by an explanation.

Example: "FAIL: Too long text. Max. length is 120 characters"

Status=*Anyname2*.SHOW(Settings.MSComm3, 2, 4, “0”,”Hello”)

A message “Hello” with normal sized character will be sent to KonicaMinolta 2600d’s display by using serial channel 2 and MsComm3 object in the form “Settings””

If the command is executed succesfully then variable “Result” will return “OKEY”. If it is not executed successfuly for a reason then it will return FAIL follwing by an explanation.

Example: "FAIL: Too long text. Max. length is 240 characters"

Subroutine no : 3

Subroutine name : CALIBRATE_WHITE

Desciption : Performs white calibration

Purpose : Calibrating the instrument with white plate

General form : CALIBRATE_WHITE(ComObject As Object, ComPortNo As Byte, Instrument As Byte, WhitePlateNo As String) As String

Input parameters:

Parameter 1 -> ComObject As Object

Obejct full name of serial communication object. Name of the MsComm object in the Form.

For example;
Assume that, a form namely "Form1" and a MsComm object on it namely "MsComm1". In this case, this parameter will be like this:

Form1.MsComm1

Parameter 2 -> ComPortNo As Byte

A number between 1 to 255 indicating port number of serial commnunication channel

Parameter 3 -> Instrument As Byte

A number indicating which instrument you use. See "SUPPORTED INSTRUMENT SELECTION TABLE" at the end of this document.

Parametre 4 -> WhitePlateNo As String

White plate serial no which is given with the instrument

Output parameters:

Parameter 1 -> CALIBRATE_WHITE As String

The name of this function is the output parameter saying "succesfully done" or not. If operation has been carried out succesfully then it will return a string as "OKEY". If operation is failed for any reason then it will return a string as "FAIL" with an explanation of reason of this failure.

Notes : If the instrument you use requires black calibration then you may need to perform black calibration first. If you don't know if the instrument requires black calibration, inpect the accesories given with the instruments. If there is no black calibration parts then the instrument does not requires black calibration.

The sequence of calibration (first white and black or first black and white) is fatal and may be different on each instrument. Refer to the instrument's manual to understand which one is first.

If the instrument has no black calibration accesories this does not mean black calibration is uncessary. It may be optional. It may be optional. In such a case, we recomend you to perform black calibration for better measurement results.

Examples :

Result=*Anyname2.CALIBRATE_WHITE* (Form1.MSComm1, 2, 3, "3001714")

KonicaMinolta 2500d will be calibrated with white plate with serial number 3001714 via MScomm object "MsComm3" in the form "Form1" using serial channel 2

If the command is executed succesfully then variable “Result” will return “OKEY”. If it is not executed successfuly for a reason then it will return FAIL following by an explanation.

Example: "FAIL 11: Black & white calibration is not ok"

Status=*Anyname2*.CALIBRATE_WHITE (Calibration.MSComm3, 2, 12, “XA7142”)

GretagMacbeth 700A will be calibrated with white plate with serial number XA7142 via MScomm object “MsComm3” in the form “Calibration” using serial channel 2.

If the command is executed succesfully then variable “Result” will return “OKEY”. If it is not executed successfuly for a reason then it will return FAIL following by an explanation.

Örnek: "FAIL 05: Lamp did not fire"

Subroutine no : 4

Subroutine no : CALIBRATE_BLACK

Description : Performs black calibrate

Purpose : Calibrating the instrument with black trap.

General form : CALIBRATE_BLACK(ComObject As Object, ComPortNo As Byte, Instrument As Byte, WhitePlateNo As String) As String

Giriş parametreleri:

Parameter 1 -> ComObject As Object

Obejct full name of serial communication object. Name of the MsComm object in the Form.

For example;
Assume that, a form namely "Form1" and a MsComm object on it namely "MsComm1". In this case, this parameter will be like this:

Form1.MsComm1

Parameter 2 -> ComPortNo As Byte

A number between 1 to 255 indicating port number of serial commnunication channel

Parameter 3 -> Instrument As Byte

A number indicating which instrument you use. See "SUPPORTED INSTRUMENT SELECTION TABLE" at the end of this document.

Parametre 4 -> WhitePlateNo As String

White plate serial no which is given with the instrument

Output parameters:

Parameter 1 -> CALIBRATE_BLACK As String

The name of this function is the output parameter saying "succesfully done" or not. If operation has been carried out succesfully then it will return a string as "OKEY". If operation is failed for any reason then it will return a string as "FAIL" with an explanation of reason of this failure.

Notes : If the instrument is a model requires black calibration then you need a accesory called "black trap".
If the black calibration of the instrument is optinal but you don't have black trap and you wish to perform black calibration anyway then hold measurement view towards a darky place which has no obstacle in front of it min. 2.5 meters range.

If black trap is not given with the instrument this does not mean it never requires black calibration. It may be optional.

Examples :

Result=*Anyname2*.CALIBRATE_BLACK (Form1.MSComm1, 2, 3, "3001714")

KonicaMinolta 2500d will be calibrated with white plate with serial number 3001714 via MScomm object "MsComm3" in the form "Form1" using serial channel 2

If the command is executed succesfully then variable "Result" will return "OKEY". If it is not executed successfuly for a reason then it will return FAIL following by an explanation.

Example: "FAIL 11: Black & white calibration is not ok"

Status=*Anyname2*.CALIBRATE_BLACK (Calibration.MSComm3, 2, 12, “XA7142”)

GretagMacbeth 700A will be calibrated with white plate with serial number XA7142 via MScomm object “MsComm3” in the form “Calibration” using serial channel 2.

If the command is executed succesfully then variable “Result” will return “OKEY”. If it is not executed successfuly for a reason then it will return FAIL following by an explanation.

Örnek: "FAIL 05: Lamp did not fire"

Subroutine no : 5

Subroutine name : INSTRUMENT_STATUS

Description : Report all settings available on instruments

Purpose : Having information about the instrument current settings.

General form : INSTRUMENT_STATUS(ComObject As Object, ComPortNo As Byte, Instrument As Byte) As String

Input parameters:

Parameter 1 -> ComObject As Object

Obejct full name of serial communication object. Name of the MsComm object in the Form.

For example;
Assume that, a form namely "Form1" and a MsComm object on it namely "MsComm1". In this case, this parameter will be like this:

Form1.MsComm1

Parameter 2 -> ComPortNo As Byte

A number between 1 to 255 indicating port number of serial commnunication channel

Parameter 3 -> Instrument As Byte

A number indicating which instrument you use. See "SUPPORTED INSTRUMENT SELECTION TABLE" at the end of this document.

Output parameters:

Parametre 1 -> INSTRUMENT_STATUS

The name of this function is the output parameter and returns a string with 38 characters. Each character has a special meaning as follows

Starting character	Length	Meaning	Values
1	4 characters	Indicates that command performed successfuly or not	"FAIL" → Command could not performed successfuly "OKEY" → Komut doğru bir şekilde işlenemedi
5	1 character	Seperator	"-"
6	1 character	Indicates measurements area	"0" → Large area view (LAV) "1" → Medium area view (MAV) "2" → Small area view (SAV) "3" → Very small area view (VSAV)
7	1 character	Seperator	"-"
8	1 character	Indicates lens position	"0" → Large area view (LAV) "1" → Medium area view (MAV) "2" → Small area view (SAV) "3" → Very small area view (VSAV)
9	1 character	Seperator	"-"
10	1 character	Indicates UV filter position	"0" included %100 "1" excluded 400nm "2" excluded 400nm low profile "3" excluded 420nm "4" excluded 420nm low profile "5" adjusted 400nm normal "6" adjusted 400nm düşük "7" adjusted 420nm normal "8" adjusted 420nm low profile "A" excluded %100
11	1 character	Seperator	"-"
12	1 character	İndicates specular component position	"0" SCI "1" SCE
13	1 character	Seperator	"-"
14	1 character	İndicates measurement style	"0" Reflection "1" Transmission
15	1 character	Seperator	"-"
16	9 character	İndicates instrument serial no	"xxxxxxxxx"
25	1 character	Seperator	"-"
26	9 character	Indicates white calibration plate serial no	"xxxxxxxxx"
35	1 character	Seperator	"-"
36	1 character	İndicates white calibration status	"0" white calibration is missing "1" white calibration is ok.
37	1 character	Seperator	"-"
38	1 character	Indicates black calibration status	"0" black calibration is missing "1" black calibration is ok.

Notes : Some digits in this string may return as "?". This does not indicate an error. This indicates that the instrument does not have such a feature.

For example, if you get "?" character at 38th digit, this means that the instrument specified don't have black calibration fetaure.

Examples :

Result=*Anyname2*.INSTRUMENT_STATUS (Form1.MSComm1, 2, 6)

KonicaMinolta 3600d instrument status will return as 38 digit string in variable "Result" using serial channel 2 with MsComm1 object on form "Form1"

If command performed successfully then variable "Result" will return such a string as below.

Example: "OKEY-1-1-0-0-0-180173004-180113004-1-1"

Meaning: Viewing area : Medium
Lens position : Medium
UV filter position : Included %100
Specular component position : SCI
Measurement mode : Reflectance
Instrument serial no : 180173004
White calibration plate no : 180113004
White calibration : Done (ok)
Siyah calibration : Done (ok)

If command can not performed successfuly then varibale “Result will return such a string as below.

Example: "HATA: Instrument could not communicate with PC"

Status=*Anyname2*.INSTRUMENT_STATUS(Settings.MSComm2, 2, 3)

KonicaMinolta 2500d instrument status will return as 38 digit string in variable “Status” using serial channel 2 with MsComm2 object on form “Settings”

If command performed successfully then variable “Result” will return such a string as below.

Example: “OKEY-0-0-0-1-0-180173004-180113004-1-1”

Meaning: Viewing area : Small
Lens position : Small
UV filter position : Included %100
Specular component position : SCE
Measurement mode : Reflectance
Instrument serial no : 180173004
White calibration plate no : 180113004
White calibration : Done (ok)
Siyah calibration : Done (ok)

If command can not performed successfuly then varibale “Result will return such a string as below.

Example: "FAIL: Time out (Receiving data)"

Subroutine no : 6

Subroutine name : MEASURE

Description : To take a measurement

Purpose : To take a measurement with current status parameters

General form : MEASURE (ComObject As Object, ComPortNo As Byte, Instrument As Byte, Result_MeasuredReflectance40() As String) As String

Input parameters:

Parameter 1 -> ComObject As Object

Obejct full name of serial communication object. Name of the MsComm object in the form.

For example;
Assume that, a form namely "Form1" and a MsComm object on it namely "MsComm1". In this case, this parameter will be like this:

Form1.MsComm1

Parameter 2 -> ComPortNo As Byte

A number between 1 to 255 indicating port number of serial commnunication channel

Parameter 3 -> Instrument As Byte

A number indicating which instrument you use. See "SUPPORTED INSTRUMENT SELECTION TABLE" at the end of this document.

Output parameters:

Parametre 1 -> Result_MeasuredReflectance40() As String

An array consist of 40 elements for reflectance values betwwen 360nm-750nm (10nm intervals).

Parametre 2 -> MEASURE As String

The name of this function is also another output parameter and returns a string if the operation has been caried out successfuly or not.

If measurement opearation has been compeleted successfuly then it wil return "OKEY"

If measurement opearation has not been compeleted successfuly then it wil return "FAIL" followed by and explanation about reason of failure.

Example: FAIL: "Instrument is not calibrated"

Notes : 1) The elements of Result_MeasuredReflectance40 array is in "string" format. If you wish to use them in a mathematical calculation, do not forget to convert it to numeric format.

2) If "FAIL" information returns with MEASURE output parameter then all elements of Result_MeasuredReflectance40() array will return "999" also.

Example :

Result=*Anyname2*.MEASURE(Form1.MSComm1, 2, 6, Values())

KonicaMinolta 3600d instrument will take a measurement via serial port 2 of MsComm1 object on form "Form1". The measurement result (reflectance values between 400nm-700nm) will return via variable Values() array with 40 elements. Variable "Result" will also return a string if the measurement was taken successfuly or not.

Example:

If the measurement operation was carried out succesfuly then output parameters will be smilar as below

Result=“OKEY”

Values(1)=”12.34”
Olculen(2)=”23.67”
.
.
.
Olculen(39)=”34.12”
Olculen(40)=”30.22”

If the operation failed then output parameter will be as follows.

Sonuc=“FAIL: "Insufficient light level"
Olculen(1)=”999”
Olculen(2)=”999”
.
.
.
Olculen(39)=”999”
Olculen(40)=”999”

Subroutine no : *7*

Subroutine name : DOWNLOAD

Description : Downloads a color from instruments memory

Purpose : Calling the values of a pre-stored color in instrument's memory

General form : DOWNLOAD(ComObject As Object, ComPortNo As Byte, Instrument As Byte, Adress As Integer, Result_Mode As String, Result_UV As String, Result_DataType As String, Result_Name As String, Result_ColorSpace As String, Result_LightSource1 As String, Result_LightSource2 As String, Result_Observer As String, Result_Reflectance40() As Single) As String

Input parameters:

Parameter 1 -> ComObject As Object

Obejct full name of serial communication object. Name of the MsComm object in the form.

For example;
Assume that, a form namely "Form1" and a MsComm object on it namely "MsComm1". In this case, this parameter will be like this:

Form1.MsComm1

Parameter 2 -> ComPortNo As Byte

A number between 1 to 255 indicating port number of serial commnunication channel

Parameter 3 -> Instrument As Byte

A number indicating which instrument you use. See "SUPPORTED INSTRUMENT SELECTION TABLE" at the end of this document.

Parametre 4 -> Adress as Integer

An integer between 1 to 699 which represent memory adress of the instrument's memory location to be downloaded.

Output parameters:

Parameter 1 -> Result_Mode as String

Returns a number representing the mode of the color in specified memory location.

"0" → MAV-SCI/E
"1" → MAV-SCI
"2" → MAV-SCE
"3" → SAV-SCI/E
"4" → SAV-SCI
"5" → SAV-SCE

Parameter 2 -> Result_UV As String

Returns a string representing the UV setting of the color in specified memory location.

"%100" → %100 included
"<400" → 400nm altı UV bileşenler iptal
"<>" → Ayarlı UV konumu

Parametre 3 -> Result_DataType As String

Returns a string representing the data type of the values in output parameter *Result_Reflectance40()*

"39" → 460-740 arası yansıma değerleri
"6" → Hesaplanmış L, a ve b değerleri

Parameter 4 -> Result_Name As String

Returns a string representing the name of the color assigned while taking the measurement.

Parameter 5 -> Result_ColorSpace As String

Returns a string representing the color space of the color assigned while taking the measurement. If the parameter Result_DataType is "39" then this parameter does not return any value.

"L*a*b*" → CIElab renk uzayı
"XYZ" → Tristimulus değerleri
"Hunter Lab" → HunterLab renk uzayı

Parameter 6 -> Result_LightSource1 As String

Returns the primary light source which is used to calculate the values in output parameter Result_Reflectance40(). If the parameter Result_DataType is "39" then this parameter does not return any value.

For more information about light sources see table "LIGHT SOURCE SELECTION TABLE" at the end of this document.

Parameter 7 -> Result_LightSource2 As String

Returns the secondary light source which is used to calculate the values in output parameter Result_Reflectance40(). If the parameter Result_DataType is "39" then this parameter does not return any value.

For more information about light sources see table "LIGHT SOURCE SELECTION TABLE" at the end of this document.

Parameter 8 -> Result_Observer As String

Returns the observer degree which is used to calculate of the values in parameter 6 and 7. If the parameter Result_DataType is "39" then this parameter does not return any value.

"10" → 10 degree observer
"2" → 2 degree observer

Parametre 9 -> Result_Reflectance40() As Single

Return the values of the color in specified memory location. These values may be XYZ tristimulus values, L*a*b* values, HunterLab values or reflectance values according to the parameter 5.

Notes : This function is valid only for some models which has memory of KonicaMinolta.These instruments are CM2300d, CM2500c, CM2500d ve CM2600d.

Example : Result= DOWNLOAD(Form1.MSComm1, 3, 4, 12, Mode, UV, Type, Name, Space, Light1, Light2, Observer, Values())

The recorded color values in memory location 12 of the instrument 2600d is called via serial port 3 of MsComm1 object on Form1.

If the opertaion is succesfull the output parameters may be like below.

Result=“OKEY”

Mode=”2”
UV=”%100”
Type=”6”
Name=”Color 1A”
Space=”L*a*b*”
Light1=”D65”
Light2=”A”
Observer=”10”
Values(1)=12.23 Values (2)=0.14 Values (3)=4.51 → L*,a*,b* values for the illuminant D65
Values (4)=-2.3 Values (5)=-1.34 Values (6)=8.1 → L*,a*,b* values for the illuminant A

If the operation is not successful then you get an error message in *Result* variable.

Result=“FAIL: No communication”

Subroutine no : 8

Subroutine name : UPLOAD

Description : Uploads a color into the instrument's memory

Purpose : Specify a memory adress and upload the values of a color into that location

General form : UPLOAD (ComObject As Object, ComPortNo As Byte, Instrument As Byte, Adress As Integer, ColorName As String, Reflectance40() As Single) As String

Input parameters:

Parameter 1 -> ComObject As Object

Obejct full name of serial communication object. Name of the MsComm object in the form.

For example;
Assume that, a form namely "Form1" and a MsComm object on it namely "MsComm1". In this case, this parameter will be like this:

Form1.MsComm1

Parameter 2 -> ComPortNo As Byte

A number between 1 to 255 indicating port number of serial commnunication channel

Parameter 3 -> Instrument As Byte

A number indicating which instrument you use. See "SUPPORTED INSTRUMENT SELECTION TABLE" at the end of this document.

Parameter 4 -> Adress as Integer

An integer between 1 to 699 which represent memory adress of the instrument's memory location used for uploading.

Parameter 5-> ColorName as String

Specify the name of the color you wish to see on instruments.

Parameter 6-> Reflectance40() as Single

The corresponding values of the color to be uploaded.

Notes : **1)** This function is valid only for some models which has memory of KonicaMinolta.These instruments are CM2300d, CM2500c, CM2500d ve CM2600d.

2) Because of each memory location can be set for a different condition (settings), the data to be uploaded should match the setting of target memory location. For example if the memory location 12 is set to L*a*b* type of data, you can not upload reflectance values. In such a case you get an error.

Example : Result= Upload(Form1.MSComm1,3, 4,12, "Yellow A1", Values())

The values in *VALUES* array is sent to memory location 12 of the instrument 2600d via serial port 3 of *MScomm1* object on *Form1*. If the process is successfull then Result variable return "OKEY".

Sonuc="OLDU" olarak geri döner.

The values in *VALUES* array must be compatible with the setting of memory location 12. See Notes 2 above for more information.

LIGHT SOURCE SELECTION TABLE	
Parameter value	**Description**
“A”	Incandencent A (Tungsten – 2856K). Home & store illumination. Usefull for metamerism evaluation.
“C”	Daylight (Halogen - 6775K). CIE avarage daylight. Usefull for metamerism evaluation.
“CWF”	Cool white (White flouresance - 4150K). Also known “F2” in USA. Home & store illumination.
“D50”	Daylight in North (5000K). Used for metamery and general purpose of color evaluation.
“D65”	Avarage common daylight in North (6500K). Used for metamery and general purpose of color evaluation. General standart in Europe and Japan.
“D75”	Daylight in North (5000K). Used for metamery and general purpose of color evaluation. Suitable for opaque samples at ASTM D1729 standarts.
“U30”	Ultralume30 (3000K). Also known as F12 in USA or TL83.
“TL83”	Philips TL83 (3000K). Also known as U30 ve F12 in USA. Used in general Europe and pacific.
“TL84”	Philips TL84 (4100K). Also known as F11 in USA. Used in general Europe and pacific.
“F2”	Wide band flouresance (4150K). Similar to “CWF”. Usefull for metamery testing. Used in USA home and store illumination.
“F7”	Wide band flouresance (6500K). Similar to “D65” and usefull for metamery testing. Represents flouresance illumination in USA.
“F11”	Philips TL84 (4000K). Similar to ultralume40. Usefull for metamery testing. Represents USA narrow band flouresance illumination.
“NBF11”	Narrow band flouresance. Usefull for metamery testing. Represents USA narrow band flouresance illumination.

SUPPORTED INSTRUMENT SELECTION TABLE	
Parameter value	**Description**
1	KonicaMinolta CM2300d
2	KonicaMinolta CM2500c
3	KonicaMinolta CM2500d
4	KonicaMinolta CM2600d
5	KonicaMinolta CM3220d
6	KonicaMinolta CM3600d
7	KonicaMinolta CM3700d
8	GretagMacbeth CE2020
9	GretagMacbeth CE2020+
10	GretagMacbeth CE2180
11	GretagMacbeth CE2180UV
12	GretagMacbeth CE3000
13	GretagMacbeth CE3100
14	GretagMacbeth CE7000
15	GretagMacbeth CE7000A
16	GretagMacbeth XTH
17	PenColorArt USB