|  |  |  |
| --- | --- | --- |
| **PMEngine API Specification** | |  |
| Version: | 3.3 |  |
| Subject: | *API specifications for PMEngine version 3.2 or higher.* | Research and Development Application Note |
| Contents: | 100 Pages, No Disks | March, 2018 |

Running the Microsoft Mail 3.2 Windows Client from the Network

WA0888

8/25/93

1993

**Application Note History**

Application Note Name: *PMEngine API .NET.doc*

Title: *PMEngine API Specification*

Product: PMEngine.dll, RadiantCommon.dll

Versions: 3.3

Original Author:

Creation Date: November, 2010

Release Date: March, 2018

Updates

|  |  |
| --- | --- |
| Revision Date | Changes by/ Comments |
| March, 2018 | K. Bilty / Updated |
| December, 2015 | M. Jones / Updated, added MeasurementF and PMMeasurementF. |
| March, 2015 | A.Usher / |
| March, 2013 | S. Skelly / Updated to version 3.2 |
| November, 2010 | Updated to version 3.0 |
| October, 2007 | Updated to version 1.1 |
| November, 2008 | Updated address |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

*.*

INFORMATION PROVIDED IN THIS DOCUMENT AND ANY SOFTWARE THAT MAY ACCOMPANY THIS DOCUMENT (collectively referred to as a Application Note) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE. The user assumes the entire risk as to the accuracy and the use of this Application Note.   
  
 Copyright © 1993-2018 Radiant Vision Systems, LLC. All Rights Reserved.  
 Microsoft is a registered trademarks and Windows is a trademark of Microsoft Corporation.

ProMetric is a registered trademark of Radiant Vision Systems, LLC.  
 *This document was created using Microsoft Word for Windows.*

**Table of Contents**

[INTRODUCTION 15](#_Toc507663387)

[Initial Setup: 15](#_Toc507663388)

[Main Object Setup (typical): 15](#_Toc507663389)

[PMEngine Class 16](#_Toc507663390)

[Background 16](#_Toc507663391)

[Constructors 16](#_Toc507663392)

[New 16](#_Toc507663393)

[Initializes a new instance of the PMEngine class.](#_Toc507663394)

[Public Methods 16](#_Toc507663395)

[AutoAdjustExposureSyncWithDoEvents Method 16](#_Toc507663396)

[Auto adjusts the exposure for a single filter, single NDFilter for a particular bin level](#_Toc507663397)

[CompactAndRepairCalibrationDatabase Method 17](#_Toc507663398)

[Compacts and repairs the calibration database.](#_Toc507663399)

[CompactAndRepairMeasurementDatabase Method 18](#_Toc507663400)

[Compacts and repairs the measurement database.](#_Toc507663401)

[CopyMeasurementToDB Method 18](#_Toc507663402)

[Copies a measurement to another database.](#_Toc507663403)

[CreateNewMeasurementDatabase Method 18](#_Toc507663404)

[Creates a new measurement database.](#_Toc507663405)

[CurrentMeasurementByID Method 18](#_Toc507663406)

[Retrieves an asynchronous PMMeasurement by its string ID. Used together with the asynchronous version of TakeMeasurement.](#_Toc507663407)

[CurrentMeasurementFByID Method 19](#_Toc507663408)

[Retrieves an asynchronous PMMeasurementF by its string ID. Used together with the asynchronous version of TakeMeasurementF.](#_Toc507663409)

[DeleteMeasurement Method 19](#_Toc507663410)

[Deletes the measurement from the database.](#_Toc507663411)

[DeleteMeasurementSetup Method 19](#_Toc507663412)

[Deletes a measurement setup from the database.](#_Toc507663413)

[GetMeasurementList Method 20](#_Toc507663414)

[Gets the list of Measurements in the current measurement database.](#_Toc507663415)

[GetMeasurementSetupList Method 20](#_Toc507663416)

[Gets the list of Measurement Setups in the current calibration database.](#_Toc507663417)

[InitializeCamera Method 20](#_Toc507663418)

[Initializes the camera.](#_Toc507663419)

[MeasurementListDataSet Method 21](#_Toc507663420)

[Returns the measurement dataset.](#_Toc507663421)

[ReadMeasurementByDescription Method 21](#_Toc507663422)

[Reads a PMMeasurement from the database.](#_Toc507663423)

[ReadMeasurementFByDescription Method 21](#_Toc507663424)

[Reads a PMMeasurementF from the database.](#_Toc507663425)

[ReadMeasurementfromDatabase Method 22](#_Toc507663426)

[Reads a PMMeasurement from the database.](#_Toc507663427)

[ReadMeasurementFfromDatabase Method 22](#_Toc507663428)

[Reads a PMMeasurementF from the database.](#_Toc507663429)

[ReadMeasurementSetupfromDatabase Method 22](#_Toc507663430)

[Reads a measurement setup from the database.](#_Toc507663431)

[SetCalibrationDatabase Method 23](#_Toc507663432)

[Sets the calibration database. Database connections are initiated in InitializeCamera().](#_Toc507663433)

[ShowAdjustExposureForm Method 23](#_Toc507663434)

[Shows the Adjust Exposure Form.](#_Toc507663435)

[ShowDefinePointsofInterestDialog Method 24](#_Toc507663436)

[Shows the Define Points of Interest dialog window.](#_Toc507663437)

[ShowExportMeasurementDataForm Method 24](#_Toc507663438)

[Shows the Export Measurement Data Form.](#_Toc507663439)

[ShowFocusModeForm Method 24](#_Toc507663440)

[Shows the Focus Mode Form.](#_Toc507663441)

[ShowMeasurementOperationsForm Method 25](#_Toc507663442)

[Shows the Measurement Operations Form](#_Toc507663443)

[ShowSubFrameDialog Method 25](#_Toc507663444)

[Shows the subframe dialog window.](#_Toc507663445)

[Shutdown Method 25](#_Toc507663446)

[Shutsdown the camera and clears ProMetric settings.](#_Toc507663447)

[TakeMeasurement Method - Synchronous 25](#_Toc507663448)

[Takes a calibrated measurement performed synchronously.](#_Toc507663449)

[TakeMeasurement Method - Asynchronous 26](#_Toc507663450)

[Takes a calibrated measurement performed asynchronously.](#_Toc507663451)

[UpdateMeasurementTableAdapter Method 26](#_Toc507663452)

[Updates the TableAdapter that is used for the MeasurementListDataSet.](#_Toc507663453)

[WriteMeasurementToDatabase Method 27](#_Toc507663454)

[Write a PMMeasurement to the database, or updates measurement information in the database.](#_Toc507663455)

[WriteMeasurementSetuptoDatabase Method 27](#_Toc507663456)

[Writes a measurement setup to the database.](#_Toc507663457)

[Public Properties 27](#_Toc507663458)

[Calibration Property 27](#_Toc507663459)

[Provides access to PMEngine’s calibration object.](#_Toc507663460)

[CalibrationDatabaseName Property 28](#_Toc507663461)

[Gets the calibration database name.](#_Toc507663462)

[CurrentMeasurement Property 28](#_Toc507663463)

[Gets the most recently captured measurement. Only used with the asynchronous TakeMeasurement and TakeMeasurementF methods.](#_Toc507663464)

[DBSettings Property 28](#_Toc507663465)

[Provides Access to PMEngine's DBSettings object](#_Toc507663466)

[MeasurementCancelled Property 28](#_Toc507663467)

[Gets whether the last measurement was cancelled.](#_Toc507663468)

[MeasurementDatabaseName Property 28](#_Toc507663469)

[Gets or sets the measurement database.](#_Toc507663470)

[Public Events 29](#_Toc507663471)

[ExposureCompleted Event 29](#_Toc507663472)

[Occurs when the last exposure of a measurement is completed.](#_Toc507663473)

[MeasurementCancelledEvent Event 29](#_Toc507663474)

[Occurs when TakeMeasurement is cancelled.](#_Toc507663475)

[MeasurementCompleted Event 29](#_Toc507663476)

[Occurs when the measurement is completed.](#_Toc507663477)

[MeasurementUpdated Event 29](#_Toc507663478)

[Occurs after a measurement is saved.](#_Toc507663479)

[MeasurementSetup Class 31](#_Toc507663480)

[Background 31](#_Toc507663481)

[Constructors 31](#_Toc507663482)

[New 31](#_Toc507663483)

[Initializes a new instance of the MeasurementSetup class.](#_Toc507663484)

[Public Methods 31](#_Toc507663485)

[Clone Method 31](#_Toc507663486)

[Returns a clone of the current Measurement Setup.](#_Toc507663487)

[SetCaptureFilterArray Method 31](#_Toc507663488)

[Copies an entire array, of size 0 to 6, to the CaptureFilter array in the Measurement Setup.](#_Toc507663489)

[Public Properties 31](#_Toc507663490)

[ApplyRADA Method 31](#_Toc507663491)

[AutoAdjustExposure Property 32](#_Toc507663492)

[Sets whether the exposure will be automatically adjusted when a measurement is taken.](#_Toc507663493)

[BinningX Property 32](#_Toc507663494)

[Gets or sets the binning in the x direction (horizontal binning).](#_Toc507663495)

[BinningY Property 32](#_Toc507663496)

[Gets or sets the binning in the y direction (vertical binning).](#_Toc507663497)

[CaptureFilter Property 32](#_Toc507663498)

[Gets or sets the array of Booleans that specifies which color filters to use when taking a measurement.](#_Toc507663499)

[ColorCalID Property 33](#_Toc507663500)

[Gets or sets the color calibration ID.](#_Toc507663501)

[ColorShiftCorrectionID Property 33](#_Toc507663502)

[Gets or sets the color shift correction ID.](#_Toc507663503)

[ConoscopeCalibrationID Property 33](#_Toc507663504)

[Gets or sets the conoscope calibration ID.](#_Toc507663505)

[CropThreshhold Property 33](#_Toc507663506)

[Gets or sets the Crop Threshold value to use to crop all filters.](#_Toc507663507)

[CropThresholdType Property 34](#_Toc507663508)

[Gets or sets the type of crop thresholding to use when applying crop thresholding.](#_Toc507663509)

[DarkImageOptions Property 34](#_Toc507663510)

[Gets or sets the Dark Image Options.](#_Toc507663511)

[Description Property 34](#_Toc507663512)

[Gets or sets the Measurement Setup's Description.](#_Toc507663513)

[DistanceUnit Property 35](#_Toc507663514)

[Gets or sets the Measurement Setup's Distance Unit.](#_Toc507663515)

[DUTDistance Property 35](#_Toc507663516)

[Gets or sets the distance for the DUT.](#_Toc507663517)

[ExposureDelay Property 35](#_Toc507663518)

[Gets or sets the exposure delay.](#_Toc507663519)

[ExposureLimitMethod Property 35](#_Toc507663520)

[Gets or sets the whether to show the over exposure method or what method to use for evaluating over exposure](#_Toc507663521)

[ExposureTime Property 36](#_Toc507663522)

[Gets or sets the exposure times for each color filter.](#_Toc507663523)

[FixedExposureTime Property 36](#_Toc507663524)

[Gets or sets the fixed exposure times for each color filter.](#_Toc507663525)

[FlatFieldID Property 37](#_Toc507663526)

[Gets or sets the flat field ID.](#_Toc507663527)

[Flip Property 37](#_Toc507663528)

[Gets or sets whether the measurement will be flipped.](#_Toc507663529)

[FluxScalingFactor Property 37](#_Toc507663530)

[Gets or sets the FluxScalingFactor.](#_Toc507663531)

[GoniometerType Property 37](#_Toc507663532)

[Gets or sets the GoniometerType for intensity measurements.](#_Toc507663533)

[HDRImagingEnabled Property 37](#_Toc507663534)

[Gets or sets whether to use HDR (High Dynamic Range) imaging.](#_Toc507663535)

[ImageScalingCalibrationID Property 38](#_Toc507663536)

[Gets or sets the image scaling ID.](#_Toc507663537)

[LastChangedDate Property 38](#_Toc507663538)

[Gets or sets the Last Changed Date.](#_Toc507663539)

[LensDistance Property 38](#_Toc507663540)

[LensDistortionID Property 38](#_Toc507663541)

[Gets or sets the LensDistortion ID.](#_Toc507663542)

[LensfNumber Property 39](#_Toc507663543)

[Gets or sets the LensfNumber value.](#_Toc507663544)

[MakeIntensity Property 39](#_Toc507663545)

[Gets or sets whether an illuminance measurement will be made into an intensity measurement.](#_Toc507663546)

[MeasurementSetupID Property 39](#_Toc507663547)

[Gets or sets the database ID of the Measurement Setup.](#_Toc507663548)

[Mirror Property 39](#_Toc507663549)

[Gets or sets whether the measurement will be mirrored.](#_Toc507663550)

[ModelNumber Property 40](#_Toc507663551)

[Gets or sets the ModelNumber of the measurement.](#_Toc507663552)

[MultiPointID Property 40](#_Toc507663553)

[Gets or sets the multipoint ID.](#_Toc507663554)

[NbrCaptureFilters Property 40](#_Toc507663555)

[Gets the number of filters to be captured.](#_Toc507663556)

[NbrFramesToAverage Property 40](#_Toc507663557)

[Gets or sets the number of frames to average when a measurement is taken.](#_Toc507663558)

[NDFilterPosition Property 40](#_Toc507663559)

[Gets or sets the ND filter position to set for each color filter.](#_Toc507663560)

[Notes Property 41](#_Toc507663561)

[Gets or sets the notes of the Measurement Setup.](#_Toc507663562)

[PhotometricUnit Property 41](#_Toc507663563)

[Gets or sets the Measurement Setup's PhotometricUnit.](#_Toc507663564)

[RemoveMoire Property 41](#_Toc507663565)

[Gets or sets whether to remove the moire.](#_Toc507663566)

[RequestDescription Property 41](#_Toc507663567)

[Gets or sets whether the Request Description dialog will show in ProMetric.](#_Toc507663568)

[SaveinDatabase Property 42](#_Toc507663569)

[Gets or sets whether the measurement will be saved in the database when it is taken.](#_Toc507663570)

[Show3dPlot Property 42](#_Toc507663571)

[Gets or sets whether the 3D IsoPlot will be automatically shown in ProMetric after the measurement is taken.](#_Toc507663572)

[ShowAnalysisForm Property 42](#_Toc507663573)

[Gets or sets whether the Analysis Form will be automatically shown in ProMetric after the measurement is taken.](#_Toc507663574)

[ShowBitmap Property 42](#_Toc507663575)

[Gets or sets whether the Bitmap will be automatically shown in ProMetric after the measurement is taken.](#_Toc507663576)

[ShowCIEColorChart Property 43](#_Toc507663577)

[Gets or sets whether the CIE Color Chart will be automatically shown in ProMetric after the measurement is taken.](#_Toc507663578)

[ShowCrossSectionGraph Property 43](#_Toc507663579)

[Gets or sets whether the Cross-Section Graph will be automatically shown in ProMetric after the measurement is taken.](#_Toc507663580)

[ShowHistogram Property 43](#_Toc507663581)

[Gets or sets whether the Histogram will be automatically shown in ProMetric after the measurement is taken.](#_Toc507663582)

[ShowIsometricPlot Property 43](#_Toc507663583)

[Gets or sets whether the Iso Plot will be automatically shown in ProMetric after the measurement is taken.](#_Toc507663584)

[ShowStatusForm Property 44](#_Toc507663585)

[Gets or sets whether the Status Form will be shown while a measurement is being taken.](#_Toc507663586)

[SourceRatedFlux Property 44](#_Toc507663587)

[Gets or sets the Source Rated Flux](#_Toc507663588)

[SpectralResponse Property 44](#_Toc507663589)

[Gets or sets the SpectralResponse](#_Toc507663590)

[StrayLightID Property 44](#_Toc507663591)

[Gets or sets the StrayLight ID.](#_Toc507663592)

[SubFrameRegion Property 45](#_Toc507663593)

[Gets or sets the size of the subframe to use when a measurement is taken.](#_Toc507663594)

[ThresholdFilter Property 45](#_Toc507663595)

[Gets or sets the threshold level to apply to each filter .](#_Toc507663596)

[Transpose Property 45](#_Toc507663597)

[Gets or sets whether the measurement will be transposed.](#_Toc507663598)

[UseFixedExposureTimes Property 45](#_Toc507663599)

[Gets or sets whether the Fixed Exposure Times will be used when the measurement is taken.](#_Toc507663600)

[Calibrations Class 47](#_Toc507663601)

[Background 47](#_Toc507663602)

[Constructors 47](#_Toc507663603)

[Public Methods 47](#_Toc507663604)

[DeleteColorCalibrationFromDatabase Method 47](#_Toc507663605)

[Deletes a color calibration from the database](#_Toc507663606)

[DeleteFlatFieldCalibrationFromDatabase Method 47](#_Toc507663607)

[Deletes a flat field calibration from the database](#_Toc507663608)

[GetColorCalibrationList Method 47](#_Toc507663609)

[Gets the array of ListItems of Color Calibrations/Brightness Scalings](#_Toc507663610)

[GetColorCalibrationList Method 48](#_Toc507663611)

[Gets the array of ListItems of Color Calibrations/Brightness Scalings](#_Toc507663612)

[GetColorCalibrationName Method 48](#_Toc507663613)

[Gets the Color Calibration Name for a color calibration ID.](#_Toc507663614)

[GetColorShiftCorrectionList Method 48](#_Toc507663615)

[Gets the array of ListItems of Color Calibrations/Brightness Scalings](#_Toc507663616)

[GetColorShiftCorrectionName Method 49](#_Toc507663617)

[Gets the description of the Color Shift Correction Calibration when given the Color Shift Correction ID.](#_Toc507663618)

[GetFlatFieldCalibrationList Method 49](#_Toc507663619)

[Gets the array of ListItems of flat field calibrations](#_Toc507663620)

[GetFlatFieldCalibrationName Method 49](#_Toc507663621)

[Returns the description of the Flat Field Calibration when given the FlatFieldCalibrationID.](#_Toc507663622)

[GetImageScalingCalibrationList Method 49](#_Toc507663623)

[Gets the array of ListItems of Image Scaling Calibrations](#_Toc507663624)

[GetImageScalingCalibrationName Method 50](#_Toc507663625)

[Returns the description of the Image Scaling Calibrations when given the ImageScalingCalibrationID.](#_Toc507663626)

[GetLensDistortionCalibrationList Method 50](#_Toc507663627)

[Gets the array of ListItems of Lens Distortion Calibrations](#_Toc507663628)

[GetLensDistortionCalibrationName Method 50](#_Toc507663629)

[Returns the description of the Lens Distortion Calibration when given the LensDistortionCalibrationID.](#_Toc507663630)

[GetMultiPointCalibrationList Method 50](#_Toc507663631)

[Gets the array of ListItems of Multi-Point Calibrations](#_Toc507663632)

[GetMultiPointCalibrationName Method 51](#_Toc507663633)

[Returns the description of the Multi-Point Calibration when given the MultiPointCalibrationID.](#_Toc507663634)

[GetStrayLightCalibrationList Method 51](#_Toc507663635)

[Gets the array of ListItems of Stray Light Calibrations](#_Toc507663636)

[GetStrayLightCalibrationName Method 51](#_Toc507663637)

[Returns the description of the Stray Light Calibration when given the StrayLightCalibrationID.](#_Toc507663638)

[LoadColorCalibrationComboBox Method 51](#_Toc507663639)

[Loads a ComboBox with the Color Calibration ListItems](#_Toc507663640)

[LoadColorShiftCorrectionComboBox Method 52](#_Toc507663641)

[Loads a ComboBox with the Color Shift Correction ListItems](#_Toc507663642)

[LoadFlatFieldCalibrationComboBox Method 52](#_Toc507663643)

[Loads a ComboBox with the Flat Field Calibration ListItems](#_Toc507663644)

[LoadImageScalingCalibrationComboBox Method 52](#_Toc507663645)

[Loads a ComboBox with the Image Scaling Calibration ListItems](#_Toc507663646)

[LoadLensDistortionCalibrationComboBox Method 52](#_Toc507663647)

[Loads a ComboBox with the Lens Distortion Calibration ListItems](#_Toc507663648)

[LoadMultiPointCalibrationComboBox Method 53](#_Toc507663649)

[Loads a ComboBox with the Multi-Point Calibration ListItems](#_Toc507663650)

[LoadStrayLightCalibrationComboBox Method 53](#_Toc507663651)

[Loads a ComboBox with the Stray Light Calibration ListItems](#_Toc507663652)

[SaveFlatFieldCalibrationToDatabase Method 53](#_Toc507663653)

[Saves the flatfield calibration, designated by MeasurementSetup.FlatFieldID, to the current Measurement database.](#_Toc507663654)

[ShowAdaptiveSelectColorCalibration Method 53](#_Toc507663655)

[Shows the Adaptive Select Color Calibration Form](#_Toc507663656)

[ShowColorShiftCorrectionList Method 54](#_Toc507663657)

[Shows the Color Shift Correction List Form](#_Toc507663658)

[ShowFlatFieldCalibrationList Method 54](#_Toc507663659)

[Shows the Flat Field Calibration List Form](#_Toc507663660)

[ShowFlatFieldCalibrationWizard Method 54](#_Toc507663661)

[Shows the flat field calibration wizard](#_Toc507663662)

[ShowFourColorCalibration Method 54](#_Toc507663663)

[Shows the Four Color Calibration Form](#_Toc507663664)

[ShowFourColorCalibrationScaleLumWithIllum Method 55](#_Toc507663665)

[Shows the Four Color Calibration Form, scales with illuminance](#_Toc507663666)

[ShowImageScalingCalibrationForm Method 55](#_Toc507663667)

[Shows the Image Scaling Calibration Form](#_Toc507663668)

[ShowImageScalingCalibrationList Method 55](#_Toc507663669)

[Shows the Image Scaling Calibration List Form](#_Toc507663670)

[ShowLensDistortionCalibrationList Method 55](#_Toc507663671)

[Shows the Lens Distortion Calibration List Form](#_Toc507663672)

[ShowLuminanceScalingCalibrationList Method 56](#_Toc507663673)

[Shows the Luminance Scaling/Color Calibration Calibration List Form](#_Toc507663674)

[ShowLuminanceScalingDialog Method 56](#_Toc507663675)

[Shows the Luminance rescaling Form](#_Toc507663676)

[ShowLuminanceScalingDialogScaleLumWithIllum Method 56](#_Toc507663677)

[Shows the Luminance Rescaling Form, Scales with Illuminance](#_Toc507663678)

[ShowMultiAreaColorCalibration Method 56](#_Toc507663679)

[Shows the Multi-Area Color Calibration Form](#_Toc507663680)

[ShowMultiColorCalibration Method 57](#_Toc507663681)

[Shows the Multi-Color Calibration Form](#_Toc507663682)

[ShowMultiPointCalibration Method 57](#_Toc507663683)

[Shows the Multi-Point Calibration Form](#_Toc507663684)

[ShowMultiPointCalibrationList Method 57](#_Toc507663685)

[Shows the Multi-Point Calibration List Form](#_Toc507663686)

[ShowOneColorCalibration Method 57](#_Toc507663687)

[Shows the One Color Calibration Form](#_Toc507663688)

[ShowOneColorCalibrationScaleLumWithIllum Method 58](#_Toc507663689)

[Shows the One Color Calibration Form, scales with illuminance](#_Toc507663690)

[ShowSingleFilterBrightnessCalibration Method 58](#_Toc507663691)

[Shows the single filter brightness scaling form](#_Toc507663692)

[ShowSingleFilterBrightnessCalibrationScaleLumWithIllum Method 58](#_Toc507663693)

[Shows the single filter brightness scaling form, scales with illuminance](#_Toc507663694)

[ShowStraylightCalibration Method 59](#_Toc507663695)

[Shows the Stray Light Calibration Form](#_Toc507663696)

[ShowStraylightCalibrationList Method 59](#_Toc507663697)

[Shows the Stray Light Calibration List Form](#_Toc507663698)

[Public Properties 59](#_Toc507663699)

[ScaleFactorCol Property 59](#_Toc507663700)

[Gets the ScaleFactorCol value.](#_Toc507663701)

[ScaleFactorRow Property 60](#_Toc507663702)

[Gets the ScaleFactorRow value.](#_Toc507663703)

[PMMeasurement /PMMeasurementF Classes 61](#_Toc507663704)

[Background 61](#_Toc507663705)

[Constructors 61](#_Toc507663706)

[New 61](#_Toc507663707)

[Initializes a new instance.](#_Toc507663708)

[New with Measurement 61](#_Toc507663709)

[Initializes a new instance using the data of the passed in Measurement/MeasurementF.](#_Toc507663710)

[Public Methods 61](#_Toc507663711)

[Clone Function Method 61](#_Toc507663712)

[Returns a new clone of the measurement object.](#_Toc507663713)

[Clone Subroutine Method 61](#_Toc507663714)

[Clones the existing measurement into the passed in measurement.](#_Toc507663715)

[CropOut Method 62](#_Toc507663716)

[Crops the measurement to the size of the passed in rectangle and returns it as a new measurement.](#_Toc507663717)

[Measurement Method 62](#_Toc507663718)

[Creates a copy of the base Measurement/MeasurementF object.](#_Toc507663719)

[ShowMeasurementInformationForm Method 62](#_Toc507663720)

[Opens the Measurement Information Form to let the user change the editable information the PMMeasurement.](#_Toc507663721)

[ShowMeasurementInformationForm Method 62](#_Toc507663722)

[Shows the Measurement Information Form as a dialog, does not save to database, but will return an OK if Measurement is changed](#_Toc507663723)

[ShowMeasurementInformationForm Method 63](#_Toc507663724)

[Shows the Measurement Information Form in the parent form, and will save the information back to the database when the user presses OK](#_Toc507663725)

[Public Properties 63](#_Toc507663726)

[ColorShiftCorrectionDescription Property 63](#_Toc507663727)

[Gets or sets the Color Shift Correction Calibration Description when taken.](#_Toc507663728)

[Current Property 63](#_Toc507663729)

[Gets or sets the current of the device being tested.](#_Toc507663730)

[FlatFieldCalibrationName Property 63](#_Toc507663731)

[Gets or sets the flat field calibration name of the measurement when taken.](#_Toc507663732)

[FluxScalingFactor Property 64](#_Toc507663733)

[Gets or sets the flux scaling factor](#_Toc507663734)

[ImageScalingCalDescription Property 64](#_Toc507663735)

[Gets or sets the Image Scaling Calibration description](#_Toc507663736)

[LensDistortionDescription Property 64](#_Toc507663737)

[Gets or sets the lens distortion calibration description.](#_Toc507663738)

[MeasurementSetupDescription Property 64](#_Toc507663739)

[Gets or sets the measurement setup description.](#_Toc507663740)

[MultiPointCalibrationDescription Property 64](#_Toc507663741)

[Gets or sets the multi-point calibration description.](#_Toc507663742)

[ModelNumber Property 65](#_Toc507663743)

[Gets or sets the Model Number.](#_Toc507663744)

[SourceRatedFlux Property 65](#_Toc507663745)

[Gets or sets the source rated flux of the device being tested.](#_Toc507663746)

[StrayLightDescription Property 65](#_Toc507663747)

[Gets or sets the Stray Light Calibration Description.](#_Toc507663748)

[Technician Property 65](#_Toc507663749)

[Gets or sets the technician’s name when the measurement is taken.](#_Toc507663750)

[Voltage Property 66](#_Toc507663751)

[Gets or sets the voltage of the device being tested.](#_Toc507663752)

[Operators 66](#_Toc507663753)

[Operator – 66](#_Toc507663754)

[Returns the result of subtracting one measurement from another](#_Toc507663755)

[The result of subtracting the second measurement from the first.](#_Toc507663756)

[Operator \* 66](#_Toc507663757)

[Returns the result of multiplying one measurement and another](#_Toc507663758)

[The result of multiplying the second measurement by the first.](#_Toc507663759)

[Operator / 67](#_Toc507663760)

[Returns the result of dividing one measurement by another](#_Toc507663761)

[The result of dividing the first measurement by the second.](#_Toc507663762)

[Operator + 67](#_Toc507663763)

[Adds one measurement to another](#_Toc507663764)

[The result of adding the two measurements.](#_Toc507663765)

[Measurement/MeasurementF Class 68](#_Toc507663766)

[Background 68](#_Toc507663767)

[Constructors 68](#_Toc507663768)

[New 68](#_Toc507663769)

[Initializes a new instance of the Measurement/MeasurementF class.](#_Toc507663770)

[Public Methods 68](#_Toc507663771)

[Crop Method 68](#_Toc507663772)

[Crops the measurement to the rectangle specified.](#_Toc507663773)

[CropOut Method 68](#_Toc507663774)

[Crops the measurement to the rectangle specified.](#_Toc507663775)

[GetAverageColor Method 68](#_Toc507663776)

[Gets the average color of the measurement.](#_Toc507663777)

[GetColor Method – Single Pixel 69](#_Toc507663778)

[Get a CIEColor object of a single pixel.](#_Toc507663779)

[GetColor Method – Single Pixel (interpolated) 69](#_Toc507663780)

[Get a CIEColor object of a single pixel (interpolated)](#_Toc507663781)

[GetColorArray Method (Measurement only) 69](#_Toc507663782)

[Gets 2D arrays containing the luminance and chromaticity data for all pixels in the measurement. Not available with MeasurementF.](#_Toc507663783)

[GetPixel Method – Single Pixel 70](#_Toc507663784)

[Gets the tristimulus array of a single pixel](#_Toc507663785)

[GetPixel Method – Single Pixel (Interpolated) 70](#_Toc507663786)

[Gets the tristimulus array of a single pixel (interpolated).](#_Toc507663787)

[GetTristimulusArrayF Method 71](#_Toc507663788)

[Gets the tristimulus array.](#_Toc507663789)

[Public Properties 71](#_Toc507663790)

[BinLevelCol Property 71](#_Toc507663791)

[Gets or sets the horizontal binning of the measurement.](#_Toc507663792)

[BinLevelRow Property 71](#_Toc507663793)

[Gets or sets the vertical binning of the measurement.](#_Toc507663794)

[CCDTemperature Property 71](#_Toc507663795)

[Gets or sets the CCDTemperature of the Measurement.](#_Toc507663796)

[CenterCol Property 72](#_Toc507663797)

[Gets or sets the Center Column in Pixels from the Left side of the Measurement.](#_Toc507663798)

[CenterRow Property 72](#_Toc507663799)

[Gets or sets the Center Row in Pixels from the Top of the Measurement.](#_Toc507663800)

[ColorCalibrationName Property 72](#_Toc507663801)

[Gets or sets the name of the color calibration used when the Measurement was created](#_Toc507663802)

[Content Property 72](#_Toc507663803)

[Gets or sets the Content of the Measurement](#_Toc507663804)

[DatabaseID Property 72](#_Toc507663805)

[Gets or sets the database ID of the measurement.](#_Toc507663806)

[Description Property 73](#_Toc507663807)

[Gets or sets the description of the measurement.](#_Toc507663808)

[DistanceUnit Property 73](#_Toc507663809)

[Gets or sets the DistanceUnit of the measurement.](#_Toc507663810)

[DUTDistance Property 73](#_Toc507663811)

[Gets or sets the DUT’s Distance](#_Toc507663812)

[MeasurementDateTime Property 73](#_Toc507663813)

[Gets or sets the date and time of the measurement when measured.](#_Toc507663814)

[NbrCols Property 74](#_Toc507663815)

[Gets the number of columns in the array.](#_Toc507663816)

[NbrRows Property 74](#_Toc507663817)

[Gets the number of rows in the array.](#_Toc507663818)

[Notes Property 74](#_Toc507663819)

[Gets or sets the notes of the measurement.](#_Toc507663820)

[PhotometricTerm Property 74](#_Toc507663821)

[Gets or sets the photometric term of the measurement.](#_Toc507663822)

[PhotometricUnit Property 75](#_Toc507663823)

[Gets or sets the photometric unit of the measurement.](#_Toc507663824)

[SpectralResponse Property 75](#_Toc507663825)

[Gets or sets the spectral response of the measurement.](#_Toc507663826)

[Operators 75](#_Toc507663827)

[Operator – 75](#_Toc507663828)

[Returns the result of subtracting one measurement from another](#_Toc507663829)

[The result of subtracting the second measurement from the first.](#_Toc507663830)

[Operator \* 75](#_Toc507663831)

[Returns the result of multiplying one measurement and another](#_Toc507663832)

[The result of multiplying the second measurement by the first.](#_Toc507663833)

[Operator / 76](#_Toc507663834)

[Returns the result of dividing one measurement by another](#_Toc507663835)

[The result of dividing the first measurement by the second.](#_Toc507663836)

[Operator + 76](#_Toc507663837)

[Returns the result of adding one measurement to another.](#_Toc507663838)

[The result of adding the two measurements.](#_Toc507663839)

[CommonFunctions Class 78](#_Toc507663840)

[Background 78](#_Toc507663841)

[Constructors 78](#_Toc507663842)

[Public Methods 78](#_Toc507663843)

[CascadeMyWindows Method 78](#_Toc507663844)

[Cascades the child windows of a MDIForm.](#_Toc507663845)

[CheckComboBoxForListItemID Method 78](#_Toc507663846)

[Checks the combox to see if it has a listitem object that has the passed in ID.](#_Toc507663847)

[CheckListItemArrayforExisting Method 78](#_Toc507663848)

[Checks the array of listitem objects for the passed in the Description and will return the ID if it exists.](#_Toc507663849)

[CloseMyWindows Method 79](#_Toc507663850)

[Closes the child windows of a MDIForm.](#_Toc507663851)

[GetDescriptionFromListItemArray Method 79](#_Toc507663852)

[Returns a string that corresponds to a description of a ListItem when given an ID.](#_Toc507663853)

[GetDistanceScaleTypesList Method 79](#_Toc507663854)

[Fills an array of ListItems that contains the three DistanceScaleTypes.](#_Toc507663855)

[GetListItemIDfromComboBox Method 80](#_Toc507663856)

[Gets the ID of the listitem of the currently selected object of the combo box.](#_Toc507663857)

[GetListItemIDfromToolStripComboBox Method 80](#_Toc507663858)

[Gets the ID of the listitem of the currently selected object of the toolstrip combo box.](#_Toc507663859)

[LoadComboBox Method 80](#_Toc507663860)

[Loads a combo box with an array of ListItems.](#_Toc507663861)

[LoadComboBoxNoSelect Method 80](#_Toc507663862)

[Loads a combobox with an array of ListItems. Does not select an item.](#_Toc507663863)

[LoadDistanceScaleTypesComboBox Method 81](#_Toc507663864)

[Loads a combo box with an array of DistanceScaleType ListItems.](#_Toc507663865)

[LoadDistanceUnitTermComboBox Method 81](#_Toc507663866)

[Loads a combobox with an array of DistanceUnitType ListItems.](#_Toc507663867)

[LoadDistanceUnitTermComboBoxNoDegrees Method 81](#_Toc507663868)

[Loads a combobox with an array of DistanceUnitType ListItems (the degrees unit is not included)](#_Toc507663869)

[LoadDistanceUnitTermComboBoxOnlyDegrees Method 82](#_Toc507663870)

[Loads a combobox with an array of DistanceUnitType ListItems (only degrees unit is included).](#_Toc507663871)

[LoadPhotometricUnitTypeComboBox Method 82](#_Toc507663872)

[Loads a combobox with an array of PhotometricUnti ListItems and selects the passed in photometric unit.](#_Toc507663873)

[LoadPhotometricUnitTypeComboBoxNoSelect Method 82](#_Toc507663874)

[Loads a combobox with an array of PhotometricUnti ListItems (does not select an item).](#_Toc507663875)

[LoadSpectralResponseTypeComboBox Method 83](#_Toc507663876)

[Loads a combobox with an array of SpectralResponseType ListItems.](#_Toc507663877)

[LoadToolStripComboBox Method 83](#_Toc507663878)

[Loads a toolstripcombobox with an array of ListItems. Selects the item passed in.](#_Toc507663879)

[LoadToolStripComboBoxNoSelect Method 83](#_Toc507663880)

[Loads a toolstripcombobox with an array of ListItems. Does not select an item.](#_Toc507663881)

[MaximizeMyWindows Method 83](#_Toc507663882)

[Maximizes the child windows of a MDIForm.](#_Toc507663883)

[MinimizeMyWindows Method 84](#_Toc507663884)

[Minimizes the child windows of a MDIForm.](#_Toc507663885)

[SetComboBoxtoListItemID Method 84](#_Toc507663886)

[Sets the combox selected item to the listitem object that has the passed in ID.](#_Toc507663887)

[SetToolStripComboBoxtoListItemID Method 84](#_Toc507663888)

[Sets the toolstripcombox selected item to the listitem object that has the passed in ID.](#_Toc507663889)

[TileMyWindows Method 85](#_Toc507663890)

[Tiles the child windows of a MDIForm.](#_Toc507663891)

[Wait Method 85](#_Toc507663892)

[Pauses the program for the designate amount of time](#_Toc507663893)

[RegionOfInterest Class 86](#_Toc507663894)

[Background 86](#_Toc507663895)

[Constructors 86](#_Toc507663896)

[Public Properties 86](#_Toc507663897)

[Center Property 86](#_Toc507663898)

[Gets or sets the center location of the regionofinterest.](#_Toc507663899)

[DistanceScale Property 86](#_Toc507663900)

[Gets or sets the geometry distance scale.](#_Toc507663901)

[DistanceUnit Property 86](#_Toc507663902)

[Gets or sets the geometry distance unit.](#_Toc507663903)

[Location Property 87](#_Toc507663904)

[Gets or sets the location of the regionofinterest. This is usually associated with the top left corner of the regionofinterest.](#_Toc507663905)

[LocationDistanceScale Property 87](#_Toc507663906)

[Gets or sets the location distance scale.](#_Toc507663907)

[LocationDistanceUnit Property 87](#_Toc507663908)

[Gets or sets the location distance unit.](#_Toc507663909)

[Sizemm Property 87](#_Toc507663910)

[Returns a generic size value in millimeters.](#_Toc507663911)

[Public Methods 88](#_Toc507663912)

[ChangeLocationDistanceScaleType Method 88](#_Toc507663913)

[Changes the location distance scale type to the new type. Alters the location.](#_Toc507663914)

[ChangeLocationDistanceUnitType Method 88](#_Toc507663915)

[Changes the location distance unit. Alters the location.](#_Toc507663916)

[GetColor Method 88](#_Toc507663917)

[Gets a CIEColor object.](#_Toc507663918)

[Clone Method 88](#_Toc507663919)

[Makes a copy of the regionofinterest.](#_Toc507663920)

[ROICircle Class 90](#_Toc507663921)

[Background 90](#_Toc507663922)

[Constructors 90](#_Toc507663923)

[New 90](#_Toc507663924)

[Initializes a new instance of the ROICircle class.](#_Toc507663925)

[New with DistanceScale and Diameter 90](#_Toc507663926)

[Initializes a new instance of the ROICircle class. Sets the distancecale property and diameter.](#_Toc507663927)

[New with DistanceUnit and Diameter 90](#_Toc507663928)

[Initializes a new instance of the ROICircle class. Sets the distanceunit property and diameter.](#_Toc507663929)

[Public Properties 90](#_Toc507663930)

[Diameter Property 90](#_Toc507663931)

[Gets or sets the circle’s diameter.](#_Toc507663932)

[Radius Property 91](#_Toc507663933)

[Gets or sets the circle’s radius.](#_Toc507663934)

[ROIRectangle Class 92](#_Toc507663935)

[Background 92](#_Toc507663936)

[Constructors 92](#_Toc507663937)

[New 92](#_Toc507663938)

[Initializes a new instance of the ROIRectangle class.](#_Toc507663939)

[New with DistanceScale, width and height 92](#_Toc507663940)

[Initializes a new instance of the ROIRectangle class. Sets the distancecale property, width and height.](#_Toc507663941)

[New with DistanceUnit, width and height 92](#_Toc507663942)

[Initializes a new instance of the ROIRectangle class. Sets the distanceunit property, width and height.](#_Toc507663943)

[New with System.Drawing.Rectangle 93](#_Toc507663944)

[Initializes a new instance of the ROIRectangle class. Sets the distancescale to pixels. Set the location, width and height to passed in rectangle.](#_Toc507663945)

[Public Properties 93](#_Toc507663946)

[Height Property 93](#_Toc507663947)

[Gets or sets the rectangle’s height.](#_Toc507663948)

[Width Property 93](#_Toc507663949)

[Gets or sets the rectangle’s Width.](#_Toc507663950)

[ROIEllipse Class 94](#_Toc507663951)

[Background 94](#_Toc507663952)

[Constructors 94](#_Toc507663953)

[New 94](#_Toc507663954)

[Initializes a new instance of the ROIEllipse class.](#_Toc507663955)

[New with DistanceScale, width and height 94](#_Toc507663956)

[Initializes a new instance of the ROIEllipse class. Sets the distancecale property, width and height.](#_Toc507663957)

[New with System.Drawing.Rectangle 94](#_Toc507663958)

[Initializes a new instance of the ROIEllipse class. Sets the distancescale to pixels. Set the location, width and height to passed in ellipse.](#_Toc507663959)

[Public Properties 94](#_Toc507663960)

[Height Property 94](#_Toc507663961)

[Gets or sets the ellipse’s height.](#_Toc507663962)

[Width Property 95](#_Toc507663963)

[Gets or sets the ellipse’s Width.](#_Toc507663964)

[ROIEntireImage Class 96](#_Toc507663965)

[Background 96](#_Toc507663966)

[Constructors 96](#_Toc507663967)

[New 96](#_Toc507663968)

[Initializes a new instance of the ROIEntireImage class.](#_Toc507663969)

[CIEColor Class 97](#_Toc507663970)

[Background 97](#_Toc507663971)

[Constructors 97](#_Toc507663972)

[New 97](#_Toc507663973)

[Initializes a new instance of the CIEColor class.](#_Toc507663974)

[New with Color Coordinates 97](#_Toc507663975)

[Initializes a new instance of the CIEColor class. Color coordinates are passed in.](#_Toc507663976)

[New with Luminance and Color Coordinates 97](#_Toc507663977)

[Initializes a new instance of the CIEColor class. Luminance and color coordinates are passed in.](#_Toc507663978)

[New with Tristimulus Values 98](#_Toc507663979)

[Initializes a new instance of the CIEColor class. Tristimulus values are passed in.](#_Toc507663980)

[Public Methods 98](#_Toc507663981)

[astar Method 98](#_Toc507663982)

[Returns a\* of CIELAB color space.](#_Toc507663983)

[bstar Method 98](#_Toc507663984)

[Returns b\* of CIELAB color space.](#_Toc507663985)

[CalcColorError Method 99](#_Toc507663986)

[Returns b\* of CIELAB color space.](#_Toc507663987)

[Clone Method 99](#_Toc507663988)

[Returns a clone of the current CIEColor object.](#_Toc507663989)

[GetLuv Method 99](#_Toc507663990)

[Gets the L u' and v' values.](#_Toc507663991)

[GetLxy Method 100](#_Toc507663992)

[Gets the L Cx and Cy values.](#_Toc507663993)

[Lstar Method 100](#_Toc507663994)

[Returns L\* of CIELAB color space.](#_Toc507663995)

[Scale Method 100](#_Toc507663996)

[Applies a scaling factor to the Tristimulus Values of the CIEColor object.](#_Toc507663997)

[ustar Method 101](#_Toc507663998)

[Returns u\* of CIELUV color space.](#_Toc507663999)

[vstar Method 101](#_Toc507664000)

[Returns v\* of CIELUV color space.](#_Toc507664001)

[Public Properties 101](#_Toc507664002)

[CCTemp Property 101](#_Toc507664003)

[Gets the color coordinate temperature.](#_Toc507664004)

[Cx Property 101](#_Toc507664005)

[Gets the CIE 1931 Cx color coordinate.](#_Toc507664006)

[Cy Property 102](#_Toc507664007)

[Gets the CIE 1931 Cy color coordinate.](#_Toc507664008)

[Cz Property 102](#_Toc507664009)

[Gets the CIE 1931 Cz color coordinate.](#_Toc507664010)

[DeltaUV Property 102](#_Toc507664011)

[Gets the delta uv value from another CIE object.](#_Toc507664012)

[DeltaUVfromCCT Property 102](#_Toc507664013)

[Gets the delta uv from CCT value.](#_Toc507664014)

[DominantWavelength Property 103](#_Toc507664015)

[Gets the dominant wavelength of the color.](#_Toc507664016)

[Lv Property 103](#_Toc507664017)

[Gets the Brightness (luminance, illuminance).](#_Toc507664018)

[u Property 103](#_Toc507664019)

[Gets the CIE 1976 u’ color coordinate.](#_Toc507664020)

[u1960 Property 103](#_Toc507664021)

[Gets the CIE 1960 u color coordinate.](#_Toc507664022)

[v Property 104](#_Toc507664023)

[Gets the CIE 1976 v’ color coordinate.](#_Toc507664024)

[v1960 Property 104](#_Toc507664025)

[Gets the CIE 1960 v color coordinate.](#_Toc507664026)

[X Property 104](#_Toc507664027)

[Gets or sets the Tristimulus X value.](#_Toc507664028)

[Y Property 104](#_Toc507664029)

[Gets or sets the Tristimulus Y value.](#_Toc507664030)

[Z Property 104](#_Toc507664031)

[Gets or sets the Tristimulus Z value.](#_Toc507664032)

[Operators 105](#_Toc507664033)

[Operator – 105](#_Toc507664034)

[Returns the result of subtracting one color point from another](#_Toc507664035)

[The result of subtracting the second color point from the first.](#_Toc507664036)

[Operator \* 105](#_Toc507664037)

[Returns the result of multiplying one color point and another](#_Toc507664038)

[The result of multiplying the second color point by the first.](#_Toc507664039)

[Operator / 105](#_Toc507664040)

[Returns the result of dividing one color point by another](#_Toc507664041)

[The result of dividing the first color point by the second.](#_Toc507664042)

[Operator + 106](#_Toc507664043)

[Adds one color point to another](#_Toc507664044)

[The result of adding the two color points.](#_Toc507664045)

[Operator <> 106](#_Toc507664046)

[Checks if two color points are not equal.](#_Toc507664047)

[The result of comparing the two points, returns True if they are not equal, False if they are.](#_Toc507664048)

[Operator = 106](#_Toc507664049)

[Checks if two color points are equal.](#_Toc507664050)

[The result of comparing the two points, returns True if they are equal, False if they are not.](#_Toc507664051)

[ListItem Class 108](#_Toc507664052)

[Background 108](#_Toc507664053)

[Constructors 108](#_Toc507664054)

[New 108](#_Toc507664055)

[Initializes a new instance of the ListItem class.](#_Toc507664056)

[New with ID and Description 108](#_Toc507664057)

[Initializes a new instance of the ListItem class.](#_Toc507664058)

[Public Properties 108](#_Toc507664059)

[Description Property 108](#_Toc507664060)

[Gets or sets the Description.](#_Toc507664061)

[ID Property 108](#_Toc507664062)

[Gets or sets the integer ID.](#_Toc507664063)

[Common Enumerations 110](#_Toc507664064)

[ColorSpace Enum 110](#_Toc507664065)

[DigitizingSpeed Enum 110](#_Toc507664066)

[DistanceScaleTypeEnum 110](#_Toc507664067)

[DistanceUnitTypeEnum 110](#_Toc507664068)

[GoniometerTypeEnum 110](#_Toc507664069)

[PhotometricTermType Enum 110](#_Toc507664070)

[PhotometricUnitTypeEnum 111](#_Toc507664071)

[SpectralResponseTypeEnum 111](#_Toc507664072)

# INTRODUCTION

This document describes the API for PMEngine and RadiantCommon. PMEngine and RadiantCommon are compatible with .NET Framework 4.0.

The basic steps to incorporate the PMEngine API are:

## Initial Setup:

1. Create your project, or copy and rename an existing example project.
2. Go to your project references and confirm/add a reference to PMEngine.dll, RadiantCommon.dll and Radiant.AssemblyLoader.dll (use the browse method to browse to the copies in your project directory).
3. Select to include the Imported namespaces: RiPMEngine and RadiantCommon.

## Main Object Setup (typical):

1. Create a new PMEngine Object
2. Set the Calibration Database
3. Initialize the camera
4. Set the Measurement Database
5. Get a Measurement Setup from the database, or create a new one.
6. Alter the measurementsetup if desired.
7. Take a measurement, using the PMEngine object’s TakeMeasurement function, returns a PMMeasurment object
8. Create a RegionofInterest (ROI) object
9. Set the parameters for the ROI object
10. Dimension a CIEColor object
11. Get a CIEColor object from the ROI using the GetColor method (passing in the PMMeasurment object)
12. Get a value, such as Lv, Cx, or Cy from the CIEColor object

# PMEngine Class

## Background

*RiPMEngine NameSpace*

The PMEngine Class is the primary interface for functionality and database interaction.

## Constructors

### New

#### Initializes a new instance of the PMEngine class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

The PMEngine constructor should always be called on the main thread of the application – the same thread that runs the user interface.

## Public Methods

### AutoAdjustExposureSyncWithDoEvents Method

#### Auto adjusts the exposure for a single filter, single NDFilter for a particular bin level

##### Syntax

###### Declaration

Public Function

AutoAdjustExposureSyncWithDoEvents (

*ColorFilter* As Integer,

*NDFilter* As Integer,

*BinLevel* As Point,

*SubFrameRegion* As Rectangle,

*ExposureLimit* As RadiantVisionSystems.Camera.ExposureLimitType,

*focusDistance* As Single,

*fStop* As Single,

*digitizingSpeed* As RadiantVisionSystems.Camera.CameraEnumerators.DigitizingSpeed

*ct* As Threading.CancellationToken?) As Single

###### Parameters

*ColorFilter*

The selected color filter to adjust the exposure.

*NDFilter*

The selected ND filter to adjust the exposure.

*BinLevel*

The bin level to adjust the exposure

*SubFrameRegion*

Limit luminance values considered to only those within this area.

*ExposureLimit*

Specifies a smoothing factor that is applied to the image when determining the brightest point.

*focusDistance*

The focus distance to use for the lens.

*fStop*

The f-stop to use for the lens.

*digitizingSpeed*

Only Fast is currently supported.

*ct*

If a CancellationToken is supplied, the process will be aborted if the token gets set.

###### Return Value

The exposure time after being adjusted.

###### Remarks

1. The focus distance and f-stop are not used for F-series and older ProMetric hardware. They can be set to 0.
2. In the method name, Sync conveys that the method is a blocking call, i.e., it does not return until it completes computing the exposure time.
3. In the method name, WithDoEvents conveys that the application’s message queue will be pumped while the method is executing.

##### Syntax

###### Declaration

Public Function AutoAdjustExposure(ByRef *MeasurementSetup* As MeasurementSetup,

ByRef *ExposureResults*() As CameraInformationCls.AdjustExposureResult) As Boolean

###### Parameters

*MeasurementSetup*

A measurementsetup that sets the parameters, and determines which filters, to adjust the exposure. The resulting exposure times are saved in the exposuretime property.

*ExposureResults*()

Specifies whether each filter was successfully adjusted or not.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| Successful | 0 | Filter successfully adjusted. |
| OverExposed | 1 | Filter overexposed. |
| UnderExposed | 2 | Filter underexposed. |
| Unknown | 3 | Results unknown, reason unknown. |

###### Return Value

Returns true if no problem adjusting the exposure time for all filters.

###### Remarks

### CompactAndRepairCalibrationDatabase Method

#### Compacts and repairs the calibration database.

##### Syntax

###### Declaration

Public Sub CompactAndRepairCalibrationDatabase()

###### Remarks

### CompactAndRepairMeasurementDatabase Method

#### Compacts and repairs the measurement database.

##### Syntax

###### Declaration

Public Sub CompactAndRepairMeasurementDatabase()

###### Remarks

### CopyMeasurementToDB Method

#### Copies a measurement to another database.

##### Syntax

###### Declaration

Public Sub CopyMeasurementToDB(*measurement* As PMMeasurement, *destinationDBName* As String)

Public Sub CopyMeasurementToDB(*measurement* As PMMeasurementF, *destinationDBName* As String)

###### Parameters

*measurement*

The data to copy to a different database.

*destinationDBName*

The destination database.

###### Remarks

Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### CreateNewMeasurementDatabase Method

#### Creates a new measurement database.

##### Syntax

###### Declaration

Public Function CreateNewMeasurementDatabase(*DatabaseName* As String) As Boolean

###### Parameters

*DatabaseName*

Fully qualified database name and path.

###### Return Value

Returns a true if successfully created.

###### Remarks

### CurrentMeasurementByID Method

#### Retrieves an asynchronous PMMeasurement by its string ID. Used together with the asynchronous version of TakeMeasurement.

##### Syntax

###### Declaration

Public Function CurrentMeasurementByID(*id* As String) As PMMeasurement

###### Parameters

*id*

GUID string ID used to identify the measurement to retrieve.

###### Return Value

Returns the PMMeasurement that was taken asynchronously.

###### Remarks

Monitor the MeasurementComplete event to know when the measurement is completed and ready to be retrieved with this method. Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### CurrentMeasurementFByID Method

#### Retrieves an asynchronous PMMeasurementF by its string ID. Used together with the asynchronous version of TakeMeasurementF.

##### Syntax

###### Declaration

Public Function CurrentMeasurementFByID(*id* As String) As PMMeasurementF

###### Parameters

*id*

GUID string ID used to identify the measurement to retrieve.

###### Return Value

Returns the PMMeasurementF that was taken asynchronously.

###### Remarks

Monitor the MeasurementComplete event to know when the measurement is completed and ready to be retrieved with this method. Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### DeleteMeasurement Method

#### Deletes the measurement from the database.

##### Syntax

###### Declaration

Public Function DeleteMeasurement(*MeasurementID* As Integer) As Boolean

###### Parameters

*MeasurementID*

ID number of the measurement.

###### Return Value

Returns a true if successfully deleted.

###### Remarks

### DeleteMeasurementSetup Method

#### Deletes a measurement setup from the database.

##### Syntax

###### Declaration

Public Sub DeleteMeasurementSetup(*MeasurementSetupID* As Integer)

###### Parameters

*MeasurementSetupID*

The ID of the Measurement Setup in the calibration database to delete.

###### Remarks

##### Syntax

###### Declaration

Public Sub DeleteMeasurementSetup(*Description* As String)

###### Parameters

*Description*

The description of the Measurement Setup in the calibration database to delete.

###### Remarks

### GetMeasurementList Method

#### Gets the list of Measurements in the current measurement database.

##### Syntax

###### Declaration

Public Shared Sub GetMeasurementList(ByRef *Measurements*() As ListItem, *IncludeMeasSetupinDescription* As Boolean)

###### Parameters

*Measurements*()

An array of ListItem object that contains the name and ID of the measurements in the database

*IncludeMeasSetupinDescription*

If set to true, the measurement’s Measurement Setup description will be appended to the name.

###### Remarks

Please note that has been changed to a shared method.

### GetMeasurementSetupList Method

#### Gets the list of Measurement Setups in the current calibration database.

##### Syntax

###### Declaration

Public Sub GetMeasurementSetupList(ByRef *MeasurementSetups*() As ListItem)

###### Parameters

*MeasurementSetups* ()

An array of ListItem object that contains the name and ID of the Measurement Setups in the database

###### Remarks

### InitializeCamera Method

#### Initializes the camera.

##### Syntax

###### Declaration

Public Function InitializeCamera() As Integer

###### Return Value

Returns a -1 if camera does not initialize property. Returns -2 if appropriate security rights are not established.

###### Remarks

The calibration database must be set before this command is called.

### MeasurementListDataSet Method

#### Returns the measurement dataset.

##### Syntax

###### Declaration

Public Function MeasurementListDataSet() As MeasurementDataSet

###### Return Value

Returns a MeasurementDataSet that is filled only with measurement information, but not the data..

###### Remarks

### ReadMeasurementByDescription Method

#### Reads a PMMeasurement from the database.

##### Syntax

###### Declaration

Public Function ReadMeasurementByDescription(*Description* As String) As PMMeasurement

###### Parameters

*Description*

The description of the measurement to read from the database.

###### Return Value

Returns a PMMeasurement object.

###### Remarks

Returns the first measurement in the database that matches the description passed in.

### ReadMeasurementFByDescription Method

#### Reads a PMMeasurementF from the database.

##### Syntax

###### Declaration

Public Function ReadMeasurementFByDescription(*Description* As String) As PMMeasurementF

###### Parameters

*Description*

The description of the measurement to read from the database.

###### Return Value

Returns a PMMeasurementF object.

###### Remarks

Returns the first measurement in the database that matches the description passed in. Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### ReadMeasurementfromDatabase Method

#### Reads a PMMeasurement from the database.

##### Syntax

###### Declaration

Public Shared Function ReadMeasurementfromDatabase(*MeasurementID* As Integer) As PMMeasurement

###### Parameters

*MeasurementID*

The database ID of the measurement to read from the database.

###### Return Value

Returns a PMMeasurement object.

###### Remarks

Returns the measurement in the database that has the ID passed in. Please note this is now a shared function.

### ReadMeasurementFfromDatabase Method

#### Reads a PMMeasurementF from the database.

##### Syntax

###### Declaration

Public Shared Function ReadMeasurementFfromDatabase(*MeasurementID* As Integer) As PMMeasurementF

###### Parameters

*MeasurementID*

The database ID of the measurement to read from the database.

###### Return Value

Returns a PMMeasurementF object.

###### Remarks

Returns the measurement in the database that has the ID passed in. Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### ReadMeasurementSetupfromDatabase Method

#### Reads a measurement setup from the database.

##### Syntax

###### Declaration

Public Function ReadMeasurementSetupfromDatabase(*MeasurementSetupID* As Integer) As MeasurementSetup

###### Parameters

*MeasurementSetupID*

ID number of the measurementsetup

###### Return Value

Returns a MeasurementSetup object.

###### Remarks

### SetCalibrationDatabase Method

#### Sets the calibration database. Database connections are initiated in InitializeCamera().

##### Syntax

###### Declaration

Public Sub SetCalibrationDatabase(*value* As String)

###### Parameters

*value*

Fully qualified database name and path.

###### Remarks

### ShowAdjustExposureForm Method

#### Shows the Adjust Exposure Form.

##### Syntax

###### Declaration

Public Function ShowAdjustExposureForm(ByRef *ExposureTime()* As Single, *NDFilters()* As Integer) As DialogResult

###### Parameters

*ExposureTime()*

The exposure times that are the initial exposure times and the modified times returned from the form.

*NDFilters()*

The NDFilters to be used during adjusting exposures.

###### Return Value

*DialogResult*

Returns DialogResult.Yes if Save is clicked

Returns DialogResult.No if Close (don’t save) is clicked

###### Remarks

##### Syntax

###### Declaration

Public Function ShowAdjustExposureForm(ByRef *MeasurementSetup* As MeasurementSetup) As DialogResult

###### Parameters

*MeasurementSetup*

The exposure times and NDFilters properties are used to set the form. The exposure times property is modified when returned.

###### Return Value

*DialogResult*

Returns DialogResult.Yes if Save is clicked

Returns DialogResult.No if Close (don’t save) is clicked

###### Remarks

### ShowDefinePointsofInterestDialog Method

#### Shows the Define Points of Interest dialog window.

##### Syntax

###### Declaration

Public Sub ShowDefinePointsofInterestDialog(*PMMeasurement* As PMMeasurement)

Public Sub ShowDefinePointsofInterestDialog(*PMMeasurement* As PMMeasurementF)

###### Parameters

*PMMeasurement*

The measurement that is used when displaying the Define Points of Interest Dialog window.

###### Remarks

Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### ShowExportMeasurementDataForm Method

#### Shows the Export Measurement Data Form.

##### Syntax

###### Declaration

Public Sub ShowExportMeasurementDataForm(*MeasurementIDList* As List(Of Integer))

###### Parameters

*MeasurementIDList*

List of IDs of the measurements to be exported

###### Remarks

### ShowFocusModeForm Method

#### Shows the Focus Mode Form.

##### Syntax

###### Declaration

Public Overloads Sub ShowFocusModeForm(*MeasurementSetup* As MeasurementSetup, *StartWithXSection* As Boolean)

###### Parameters

*MeasurementSetup*

Used to set initial parameters when the form is loaded.

*StartWithXSection*

If set to true the form will open to the cross-section.

###### Remarks

##### Syntax

###### Declaration

Public Sub ShowFocusModeForm(*MeasurementSetup* As MeasurementSetup)

###### Parameters

*MeasurementSetup*

Used to set initial parameters when the form is loaded.

###### Remarks

### ShowMeasurementOperationsForm Method

#### Shows the Measurement Operations Form

##### Syntax

###### Declaration

Public Sub ShowMeasurementOperationsForm(*Measurement* As PMMeasurement)

Public Sub ShowMeasurementOperationsForm(*Measurement* As PMMeasurementF)

###### Parameters

*Measurement*

The measurement to perform the operations on.

###### Remarks

Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### ShowSubFrameDialog Method

#### Shows the subframe dialog window.

##### Syntax

###### Declaration

Public Sub ShowSubFrameDialog(*MeasurementSetup* As MeasurementSetup, ByRef *ClippingRegion* As System.Drawing.Rectangle)

###### Parameters

*MeasurementSetup*

Used to set initial parameters when the form is loaded.

*ClippingRegion*

Clipping region established in the subframe dialog window.

###### Return Value

### Shutdown Method

#### Shutsdown the camera and clears ProMetric settings.

##### Syntax

###### Declaration

Public Sub Shutdown()

###### Remarks

Should only be called when closing the application. To use ProMetric again, it will need to be Initialized.

### TakeMeasurement Method - Synchronous

#### Takes a calibrated measurement performed synchronously.

##### Syntax

###### Declaration

Public Function TakeMeasurement(*MeasurementSetup* As MeasurementSetup, *Description* As String, *Voltage* As Single, *Current* As Single, *Technician* As String, *Notes* As String) As PMMeasurement

Public Function TakeMeasurementF(*MeasurementSetup* As MeasurementSetup, *Description* As String, *Voltage* As Single, *Current* As Single, *Technician* As String, *Notes* As String) As PMMeasurementF

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement.

*Description*

The description to be set for the measurement’s description property.

*Voltage*

Sets the measurement’s voltage property.

*Current*

Sets the measurement’s current property.

*Technician*

Sets the measurement’s technician property.

*Notes*

Sets the measurement’s notes property.

###### Return Value

Returns a PMMeasurement or PMMeasurementF object.

###### Remarks

Does not return until the measurement is finished being taken. Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### TakeMeasurement Method - Asynchronous

#### Takes a calibrated measurement performed asynchronously.

##### Syntax

###### Declaration

Public Function TakeMeasurement(*MeasurementSetup* As MeasurementSetup) As String

Public Function TakeMeasurementF(*MeasurementSetup* As MeasurementSetup) As String

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement.

###### Return Value

Returns a string GUID (globally unique identifier) used to retrieve the measurement once it is complete

###### Remarks

Returns before the measurement is finished being taken. Monitor the MeasurementCompleted (for TakeMeasurement) or MeasurementFCompleted (for TakeMeasurementF) Event to determine when the measurement is finished being taken. The measurement can be accessed with the CurrentMeasurementByID or CurrentMeasurementFByID method. Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### UpdateMeasurementTableAdapter Method

#### Updates the TableAdapter that is used for the MeasurementListDataSet.

##### Syntax

###### Declaration

Public Sub UpdateMeasurementTableAdapter()

###### Remarks

### WriteMeasurementToDatabase Method

#### Write a PMMeasurement to the database, or updates measurement information in the database.

##### Syntax

###### Declaration

Public Sub WriteMeasurementToDatabase(ByRef *PMMeasurement* As PMMeasurement,

*MeasurementInfoOnly* As Boolean)

Public Sub WriteMeasurementToDatabase(ByRef *PMMeasurement* As PMMeasurementF,

*MeasurementInfoOnly* As Boolean)

###### Parameters

*PMMeasurement*

Measurement to be written to the database.

*MeasurementInfoOnly*

When set to true it will only write the measurement’s information and not the image data.

###### Remarks

To write a new measurement to the database, set its DatabaseID property to 0; to overwrite an existing measurement, leave the existing DatabaseID. Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### WriteMeasurementSetuptoDatabase Method

#### Writes a measurement setup to the database.

##### Syntax

###### Declaration

Public Sub WriteMeasurementSetuptoDatabase(ByRef *MeasurementSetup* As MeasurementSetup)

###### Parameters

*MeasurementSetup*

The measurement setup object to write to the database.

###### Remarks

## Public Properties

### Calibration Property

#### Provides access to PMEngine’s calibration object.

##### Syntax

###### Declaration

Public ReadOnly Property Calibration() As Calibrations

###### Property Value

The calibration object.

###### Remarks

See the calibration class details about this object.

### CalibrationDatabaseName Property

#### Gets the calibration database name.

##### Syntax

###### Declaration

Public ReadOnly Property CalibrationDatabaseName() As String

###### Property Value

The calibration database name.

###### Remarks

### CurrentMeasurement Property

#### Gets the most recently captured measurement. Only used with the asynchronous TakeMeasurement and TakeMeasurementF methods.

##### Syntax

###### Declaration

Public ReadOnly Property CurrentMeasurement() As PMMeasurement

Public ReadOnly Property CurrentMeasurementF() As PMMeasurementF

###### Property Value

Measurement associated with the asynchronous TakeMeasurement and TakeMeasurementF methods.

###### Remarks

Returns the latest asynchronous measurement regardless of its ID.

### DBSettings Property

#### Provides Access to PMEngine's DBSettings object

##### Syntax

###### Declaration

Public ReadOnly Property DBSettings() As DBSettings

###### Property Value

PMEngine's DBSettings object.

###### Remarks

See DBSettings details about this object.

### MeasurementCancelled Property

#### Gets whether the last measurement was cancelled.

##### Syntax

###### Declaration

Public ReadOnly Property MeasurementCancelled() As Boolean

###### Property Value

Returns a true if the last measurement was cancelled.

###### Remarks

### MeasurementDatabaseName Property

#### Gets or sets the measurement database.

##### Syntax

###### Declaration

Public Property MeasurementDatabaseName() As String

###### Property Value

The name and path of the measurement database.

###### Remarks

## Public Events

### ExposureCompleted Event

#### Occurs when the last exposure of a measurement is completed.

##### Syntax

###### Declaration

Public Event ExposureCompleted as EventHandler

###### Remarks

Occurs after the sensor receives light per the exposure settings use with the measurement and before the data is transferred to the PC. Indicates it is safe to move or extinguish the DUT after this event is raised.

### MeasurementCancelledEvent Event

#### Occurs when TakeMeasurement is cancelled.

##### Syntax

###### Declaration

Public Event MeasurementCancelledEvent as EventHandler

###### Remarks

Occurs when the cancel button is pressed during a TakeMeasurement.

### MeasurementCompleted Event

#### Occurs when the measurement is completed.

##### Syntax

###### Declaration

Public Event MeasurementCompleted as EventHandler

Public Event MeasurementFCompleted as EventHandler

###### Remarks

For the synchronous TakeMeasurement and TakeMeasurementF it occurs after the measurement is taken, processed, and saved to database. For asynchronous TakeMeasurement and TakeMeasurementF it occurs after the measurement is taken and processed, but before it is saved to database.

### MeasurementUpdated Event

#### Occurs after a measurement is saved.

##### Syntax

###### Declaration

Public Event MeasurementUpdated as EventHandler

###### Remarks

# MeasurementSetup Class

## Background

*RiPMEngine NameSpace*

For many functions in the PMEngine class it is required to send an object of the MeasurementSetup class type. The measurement setup can be thought of as including all the parameters that are needed to take a measurement.

## Constructors

### New

#### Initializes a new instance of the MeasurementSetup class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

## Public Methods

### Clone Method

#### Returns a clone of the current Measurement Setup.

##### Syntax

###### Declaration

Public Function Clone() As MeasurementSetup

###### Return Value

A cloned copy of the original MeasurementSetup object.

###### Remarks

### SetCaptureFilterArray Method

#### Copies an entire array, of size 0 to 6, to the CaptureFilter array in the Measurement Setup.

##### Syntax

###### Declaration

Public Sub SetCaptureFilterArray(*CaptureFilterNew()* As Boolean)

###### Parameters

*CaptureFilterNew()*

An array of size 0 to 6 that specifies which filters to capture images for when the measurement is taken.

###### Remarks

This is an alternate way of setting the CaptureFilter property.

## Public Properties

### ApplyRADA Method

Gets or sets whether RADA will be applied.

##### Syntax

###### Declaration

Public Property ApplyRADA() As Boolean

###### Property Value

Boolean that indicates if RADA will occur.

###### Remarks

### AutoAdjustExposure Property

#### Sets whether the exposure will be automatically adjusted when a measurement is taken.

##### Syntax

###### Declaration

Public Property AutoAdjustExposure () As Integer

###### Property Value

Boolean that indicates if the exposure will automatically be adjusted before a measurement is taken.

###### Remarks

This property is overridden by the UseFixedExposureTimes property.

### BinningX Property

#### Gets or sets the binning in the x direction (horizontal binning).

##### Syntax

###### Declaration

Public Property BinningX() As Integer

###### Property Value

The horizontal binning to be used when a measurement is taken.

###### Remarks

### BinningY Property

#### Gets or sets the binning in the y direction (vertical binning).

##### Syntax

###### Declaration

Public Property BinningY() As Integer

###### Property Value

The vertical binning to be used when a measurement is taken.

###### Remarks

### CaptureFilter Property

#### Gets or sets the array of Booleans that specifies which color filters to use when taking a measurement.

##### Syntax

###### Declaration

Public Property CaptureFilter(*FilterNumber* as Integer) As Boolean

###### Parameters

*FilterNumber*

Value between 1-6 corresponding to the filter’s position. Typically, Y filter = 1, X filter = 2, Z filter = 3. If optional XB filter is installed, XB = 4. Positions 0, 5 and 6 are unused.

###### Property Value

Boolean that says whether the image should be included when taking a measurement.

###### Remarks

This property is indexed, which could require unusual syntax in languages other than VB.NET.

### ColorCalID Property

#### Gets or sets the color calibration ID.

##### Syntax

###### Declaration

Public Property ColorCalID() As Integer

###### Property Value

The database ID number of color calibration to be used.

###### Remarks

### ColorShiftCorrectionID Property

#### Gets or sets the color shift correction ID.

##### Syntax

###### Declaration

Public Property ColorShiftCorrectionID () As Integer

###### Property Value

The database ID number of color shift correction to be used.

###### Remarks

### ConoscopeCalibrationID Property

#### Gets or sets the conoscope calibration ID.

##### Syntax

###### Declaration

Public Property ConoscopeCalibrationID () As Integer

###### Property Value

The database ID number of the conoscope calibration to be used.

###### Remarks

### CropThreshhold Property

#### Gets or sets the Crop Threshold value to use to crop all filters.

##### Syntax

###### Declaration

Public Property CropThreshhold() As Single

###### Property Value

Threshold level between 0 and 100 percent that is used to calculate the cropping rectangle to apply to all filters.

###### Remarks

The type of crop thresholding is based on the CropThresholdType.

### CropThresholdType Property

#### Gets or sets the type of crop thresholding to use when applying crop thresholding.

##### Syntax

###### Declaration

Public Property CropThresholdType() As ThresholdType

###### Property Value

*ThresholdType Enum*

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| PercentOfAbsoluteMaximum | 0 | Determines cropping rectangle based upon max bit level. |
| PercentOfMeasurementMaximum | 1 | Determines cropping rectangle based upon the max value of all images in the measurement |
| PercentOfLuminanceMaximum | 2 | Determines the cropping rectangle based upon the luminance max (after applying color calibration) |
| PercentOfLuminanceMaximumWithMedianFilter | 3 | Determines the cropping rectangle based upon the luminance max (after color cal) and with a Median Filter. |

###### Remarks

### DarkImageOptions Property

#### Gets or sets the Dark Image Options.

##### Syntax

###### Declaration

Public Property DarkImageOptions() As DarkImageOptions

###### Property Value

The Dark Image Options value used when the camera should take a dark frame.

*DarkImageOptions Enum*

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| AutoDark | 0 | The camera object decides when to take the dark image |
| ForceNewDark | 1 | Forces the camera object to take a dark frame |
| DarkFrameOnly | 2 | A dark frame is taken but not a light frame |
| NoDark | 3 | Does not let the camera take a new dark frame |

###### Remarks

This is not used with I and Y series cameras.

### Description Property

#### Gets or sets the Measurement Setup's Description.

##### Syntax

###### Declaration

Public Property Description() As String

###### Property Value

A descriptive string for the measurement setup.

###### Remarks

### DistanceUnit Property

#### Gets or sets the Measurement Setup's Distance Unit.

##### Syntax

###### Declaration

Public Property DistanceUnit() As DistanceUnitType

###### Property Value

The Distance Unit to be used when the measurement is created.

*DistanceUnitType Enum*

See the Common Enumerations section

###### Remarks

Intensity measurements are automatically assigned Degrees.

### DUTDistance Property

#### Gets or sets the distance for the DUT.

##### Syntax

###### Declaration

Public Property DUTDistance() As Single

###### Property Value

The distance for the DUT.

###### Remarks

The DUT distance is used for calculating intensity.

### ExposureDelay Property

#### Gets or sets the exposure delay.

##### Syntax

###### Declaration

Public Property ExposureDelay() As Double

###### Property Value

The exposure delay to be used.

###### Remarks

### ExposureLimitMethod Property

#### Gets or sets the whether to show the over exposure method or what method to use for evaluating over exposure

##### Syntax

###### Declaration

Public Property ExposureLimitMethod() As ExposureLimitType

###### Property Value

Shows the over exposure message for measurements that are overexposed, or uses other method to evaluate overexposure.

*ExposureLimitType Enum*

The spectral response of a measurement.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| None | 0 | Uses Standard method (histogram). Doesn’t show a message box if overexposed |
| Standard | 1 | Uses Standard method (histogram). Shows a message box if overexposed. |
| BrightSpot | 2 | Uses the BrightSpot method when adjusting exposure or taking a measurement (message box shows if overexposed) |
| BrightSpot2x2 | 3 | Uses the BrightSpot2x2 method when adjusting exposure or taking a measurement (message box shows if overexposed) |
| BrightSpot3x3 | 4 | Uses the BrightSpot3x3 method when adjusting exposure or taking a measurement (message box shows if overexposed) |
| BrightSpot4x4 | 5 | Uses the BrightSpot4x4 method when adjusting exposure or taking a measurement (message box shows if overexposed) |

###### Remarks

### ExposureTime Property

#### Gets or sets the exposure times for each color filter.

##### Syntax

###### Declaration

Public Property ExposureTime(*FilterNumber* as Integer) As Single

###### 

###### Parameters

*FilterNumber*

Value between 1-6 corresponding to the color filter’s position. Typically, Y filter = 1, X filter = 2, Z filter = 3. If optional XB filter is installed, XB = 4. Positions 0, 5 and 6 are unused.

###### Property Value

The exposure time of the filter when a measurement is taken.

###### Remarks

This property is indexed, which could require unusual syntax in languages other than VB.NET.

### FixedExposureTime Property

#### Gets or sets the fixed exposure times for each color filter.

##### Syntax

###### Declaration

Public Property FixedExposureTime(*FilterNumber* as Integer) As Single

###### 

###### Parameters

*FilterNumber*

Value between 1-6 corresponding to the color filter’s position. Typically, Y filter = 1, X filter = 2, Z filter = 3. If optional XB filter is installed, XB = 4. Positions 0, 5 and 6 are unused.

###### Property Value

The fixed exposure time of the filter when a measurement is taken. These exposure times are used instead of ExposureTime when the UseFixedExposureTime property is set to true.

###### Remarks

This property is indexed, which could require unusual syntax in languages other than VB.NET.

### FlatFieldID Property

#### Gets or sets the flat field ID.

##### Syntax

###### Declaration

Public Property FlatFieldID() As Integer

###### Property Value

The database ID for the flat field calibration to be used.

###### Remarks

### Flip Property

#### Gets or sets whether the measurement will be flipped.

##### Syntax

###### Declaration

Public Property Flip() As Boolean

###### Property Value

Property indicates whether the measurement will be flipped.

###### Remarks

### FluxScalingFactor Property

#### Gets or sets the FluxScalingFactor.

##### Syntax

###### Declaration

Public Property FluxScalingFactor() As Single

###### Property Value

The luminance value of the measurement is multiplied by the flux scaling factor.

###### Remarks

This value will change the luminance of the measurement from the calibrated value.

### GoniometerType Property

#### Gets or sets the GoniometerType for intensity measurements.

##### Syntax

###### Declaration

Public Property GoniometerType() As GoniometerType

###### Property Value

The goniometer type to be used when an illuminance measurement is made into an intensity measurement.

*GoniometerType Enum*

See the Common Enumerations section.

###### Remarks

This property is used for changing illuminance measurements into intensity measurements if the MakeIntensity property is set.

### HDRImagingEnabled Property

#### Gets or sets whether to use HDR (High Dynamic Range) imaging.

##### Syntax

###### Declaration

Public Property HDRImagingEnabled() As Boolean

###### Property Value

Whether to use HDR imaging or not.

###### Remarks

### ImageScalingCalibrationID Property

#### Gets or sets the image scaling ID.

##### Syntax

###### Declaration

Public Property ImageScalingCalibrationID() As Integer

###### Property Value

The database ID for the image scaling calibration to be used.

###### Remarks

If an illuminance calibration is used the image scaling will be based off the scaling performed during the calibration wizard.

### LastChangedDate Property

#### Gets or sets the Last Changed Date.

##### Syntax

###### Declaration

Public Property LastChangedDate() As Date

###### Property Value

The last changed date when saved to the database.

###### Remarks

The last changed date is modified when the measurement setup is saved to the database.

### LensDistance Property

Gets or sets the Lens Distance

##### Syntax

###### Declaration

Public Property LensDistance() As Single

###### Property Value

The value used for lens focus when taking a measurement with the specified measurement setup

###### Remarks

The lens distance is specified in units of meters.

### LensDistortionID Property

#### Gets or sets the LensDistortion ID.

##### Syntax

###### Declaration

Public Property LensDistortionID() As Integer

###### Property Value

The database ID of the lens distortion calibration to be used.

###### Remarks

### LensfNumber Property

#### Gets or sets the LensfNumber value.

##### Syntax

###### Declaration

Public Property LensfNumber() As String

###### Property Value

The value to set the f-number of the lens.

###### Remarks

If an illegal value is used to set the f-number, the aperture will be positioned wide open.

### MakeIntensity Property

#### Gets or sets whether an illuminance measurement will be made into an intensity measurement.

##### Syntax

###### Declaration

Public Property MakeIntensity() As Boolean

###### Property Value

Property indicates whether an illuminance measurement will be made into an intensity measurement.

###### Remarks

Will only make illuminance measurements into intensity measurements. Uses the GoniometerType and DUTDistance properties.

### MeasurementSetupID Property

#### Gets or sets the database ID of the Measurement Setup.

##### Syntax

###### Declaration

Public Property MeasurementSetupID() As Integer

###### Property Value

The database ID of the Measurement Setup.

###### Remarks

The database ID is assigned when the measurement setup is saved to the database.

### Mirror Property

#### Gets or sets whether the measurement will be mirrored.

##### Syntax

###### Declaration

Public Property Mirror() As Boolean

###### Property Value

Property indicates whether the measurement will be mirrored.

###### Remarks

### ModelNumber Property

#### Gets or sets the ModelNumber of the measurement.

##### Syntax

###### Declaration

Public Property ModelNumber() As String

###### Property Value

The modelnumber of the measurement.

###### Remarks

### MultiPointID Property

#### Gets or sets the multipoint ID.

##### Syntax

###### Declaration

Public Property MultiPointID() As Integer

###### Property Value

The database ID for the multi-point calibration to be used.

###### Remarks

### NbrCaptureFilters Property

#### Gets the number of filters to be captured.

##### Syntax

###### Declaration

Public ReadOnly Property NbrCaptureFilters() As Integer

###### Property Value

The number of filters to be captured, counted from CaptureFilter property.

###### Remarks

### NbrFramesToAverage Property

#### Gets or sets the number of frames to average when a measurement is taken.

##### Syntax

###### Declaration

Public Property NbrFramesToAverage() As Integer

###### Property Value

The number of frames to average when a measurement is taken.

###### Remarks

### NDFilterPosition Property

#### Gets or sets the ND filter position to set for each color filter.

##### Syntax

###### Declaration

Public Property NDFilterPosition(*FilterNumber* As Integer) As Integer

###### 

###### Parameters

*FilterNumber*

Value between 1-6 corresponding to the color filter’s position. Typically, Y filter = 1, X filter = 2, Z filter = 3. If optional XB filter is installed, XB = 4. Positions 0, 5 and 6 are unused.

###### Property Value

The ND filter position to set for each color filter.

###### Remarks

This property is indexed, which could require unusual syntax in languages other than VB.NET.

### Notes Property

#### Gets or sets the notes of the Measurement Setup.

##### Syntax

###### Declaration

Public Property Notes() As String

###### Property Value

The notes of the Measurement Setup.

###### Remarks

### PhotometricUnit Property

#### Gets or sets the Measurement Setup's PhotometricUnit.

##### Syntax

###### Declaration

Public Property PhotometricUnit() As PhotometricUnitType

###### Property Value

The Photometric Unit to be used when the measurement is created.

*PhotometricUnitType Enum*

See the Common Enumerations section

###### Remarks

### RemoveMoire Property

#### Gets or sets whether to remove the moire.

##### Syntax

###### Declaration

Public Property RemoveMoire() As Boolean

###### Property Value

Whether to remove the moire or not.

###### Remarks

### RequestDescription Property

#### Gets or sets whether the Request Description dialog will show in ProMetric.

##### Syntax

###### Declaration

Public Property RequestDescription() As Boolean

###### Property Value

Shows the Request Description dialog in ProMetric.

###### Remarks

This property only has an effect in the ProMetric application. It can be used for storing other information as desired.

### SaveinDatabase Property

#### Gets or sets whether the measurement will be saved in the database when it is taken.

##### Syntax

###### Declaration

Public Property SaveinDatabase() As Boolean

###### Property Value

The measurement is saved in the database when taken.

###### Remarks

### Show3dPlot Property

#### Gets or sets whether the 3D IsoPlot will be automatically shown in ProMetric after the measurement is taken.

##### Syntax

###### Declaration

Public Property Show3dPlot() As Boolean

###### Property Value

Shows the 3D IsoPlot in ProMetric.

###### Remarks

This property only has an effect in the ProMetric application. It can be used for storing other information as desired.

### ShowAnalysisForm Property

#### Gets or sets whether the Analysis Form will be automatically shown in ProMetric after the measurement is taken.

##### Syntax

###### Declaration

Public Property ShowAnalysisForm() As Boolean

###### Property Value

Shows the Analysis Form in ProMetric.

###### Remarks

This property only has an effect in the ProMetric application. It can be used for storing other information as desired.

### ShowBitmap Property

#### Gets or sets whether the Bitmap will be automatically shown in ProMetric after the measurement is taken.

##### Syntax

###### Declaration

Public Property ShowBitmap() As Boolean

###### Property Value

Shows the Bitmap in ProMetric.

###### Remarks

This property only has an effect in the ProMetric application. It can be used for storing other information as desired.

### ShowCIEColorChart Property

#### Gets or sets whether the CIE Color Chart will be automatically shown in ProMetric after the measurement is taken.

##### Syntax

###### Declaration

Public Property ShowCIEColorChart() As Boolean

###### Property Value

Shows the CIE Color Chart in ProMetric.

###### Remarks

This property only has an effect in the ProMetric application. It can be used for storing other information as desired.

### ShowCrossSectionGraph Property

#### Gets or sets whether the Cross-Section Graph will be automatically shown in ProMetric after the measurement is taken.

##### Syntax

###### Declaration

Public Property ShowCrossSectionGraph() As Boolean

###### Property Value

Shows the Cross-Section Graph in ProMetric.

###### Remarks

This property only has an effect in the ProMetric application. It can be used for storing other information as desired.

### ShowHistogram Property

#### Gets or sets whether the Histogram will be automatically shown in ProMetric after the measurement is taken.

##### Syntax

###### Declaration

Public Property ShowHistogram() As Boolean

###### Property Value

Shows the Histogram in ProMetric.

###### Remarks

This property only has an effect in the ProMetric application. It can be used for storing other information as desired.

### ShowIsometricPlot Property

#### Gets or sets whether the Iso Plot will be automatically shown in ProMetric after the measurement is taken.

##### Syntax

###### Declaration

Public Property ShowIsometricPlot() As Boolean

###### Property Value

Shows the Iso Plot in ProMetric.

###### Remarks

This property only has an effect in the ProMetric application. It can be used for storing other information as desired.

### ShowStatusForm Property

#### Gets or sets whether the Status Form will be shown while a measurement is being taken.

##### Syntax

###### Declaration

Public Property ShowStatusForm() As Boolean

###### Property Value

Shows the Status Form while a measurement is being taken.

###### Remarks

### SourceRatedFlux Property

#### Gets or sets the Source Rated Flux

##### Syntax

###### Declaration

Public Property SourceRatedFlux() As Single

###### Property Value

The source rated flux of a measurement..

###### Remarks

Can be used for efficiency calculations.

### SpectralResponse Property

#### Gets or sets the SpectralResponse

##### Syntax

###### Declaration

Public Property SpectralResponse() As SpectralResponseType

###### Property Value

The source rated flux of a measurement..

*SpectralResponseType Enum*

See the Common Enumerations section

###### Remarks

Changes the photometric units displayed in plots as specified by the PhotometricUnit property.

### StrayLightID Property

#### Gets or sets the StrayLight ID.

##### Syntax

###### Declaration

Public Property StrayLightID() As Integer

###### Property Value

The database ID for the Stray Light calibration to be used.

###### Remarks

### SubFrameRegion Property

#### Gets or sets the size of the subframe to use when a measurement is taken.

##### Syntax

###### Declaration

Public Property SubFrameRegion() As System.Drawing.Rectangle

###### Property Value

The size of the subframe as a Rectangle in pixel values.

###### Remarks

### ThresholdFilter Property

#### Gets or sets the threshold level to apply to each filter .

##### Syntax

###### Declaration

Public Property ThresholdFilter(*FilterNumber* As Integer) As Single

###### 

###### Parameters

*FilterNumber*

Value between 1-6 corresponding to the color filter’s position. Typically, Y filter = 1, X filter = 2, Z filter = 3. If optional XB filter is installed, XB = 4. Positions 0, 5 and 6 are unused.

###### Property Value

Threshold level between 0 and 100 percent where 0 is no thresholding and 100 thresholds to the maximum possible gray value.

###### Remarks

This property is indexed, which could require unusual syntax in languages other than VB.NET.

### Transpose Property

#### Gets or sets whether the measurement will be transposed.

##### Syntax

###### Declaration

Public Property Transpose() As Boolean

###### Property Value

Property indicates whether the measurement will be transposed.

###### Remarks

### UseFixedExposureTimes Property

#### Gets or sets whether the Fixed Exposure Times will be used when the measurement is taken.

##### Syntax

###### Declaration

Public Property UseFixedExposureTimes() As Boolean

###### Property Value

Property indicates whether the Fixed Exposure Times will be used when the measurement is taken.

###### Remarks

If set to false the values set in the ExposureTime property will be used instead.

# Calibrations Class

## Background

*RIPMEngine NameSpace*

The Calibration object is the provides access to important calibration lists and routines.

## Constructors

None supported. There is already an existing Calibration Object that can be referenced from the PMEngine object.

## Public Methods

### DeleteColorCalibrationFromDatabase Method

#### Deletes a color calibration from the database

##### Syntax

###### Declaration

Public Sub DeleteColorCalibrationFromDatabase(*ColorCalibrationID* As Integer, Optional *refreshCalList* As Boolean, Optional *FireEvent* As Boolean)

###### Parameters

*ColorCalibrationID*

The database ID of the color calibration.

*refreshCalList*

Boolean to refresh the calibration list after the deletion.

*FireEvent*

Boolean to fire *OnCalibrationUpdated* event.

###### Remarks

### DeleteFlatFieldCalibrationFromDatabase Method

#### Deletes a flat field calibration from the database

##### Syntax

###### Declaration

Public Sub DeleteFlatFieldCalibrationFromDatabase(*FlatFieldCalibrationID* As Integer)

###### Parameters

*FlatFieldCalibrationID*

The database ID of the flat field calibration.

###### Remarks

### GetColorCalibrationList Method

#### Gets the array of ListItems of Color Calibrations/Brightness Scalings

##### Syntax

###### Declaration

Public Sub GetColorCalibrationList(ByRef *ColorCalibrations*() As ListItem)

###### Parameters

*ColorCalibrations*

An array of ListItems of the database’s color calibrations.

###### Remarks

### GetColorCalibrationList Method

#### Gets the array of ListItems of Color Calibrations/Brightness Scalings

##### Syntax

###### Declaration

Public Sub GetColorCalibrationList(ByRef *ColorCalibrations*() As ListItem, *ColorCalType* As ColorCalibrationType, *IncludeNone* As Boolean)

###### Parameters

*ColorCalibrations*

An array of ListItems of the database’s color calibrations.

*ColorCalType*

*ColorCalibrationType Enum*

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| ColorMatrix | 0 | Color Matrix Type. |
| MultiArea | 1 | Multi-Area Type |
| AdaptiveSelect | 2 | Adaptive Select Type |

*IncludeNone*

Boolean that says to include “None” in the list.

###### Remarks

### GetColorCalibrationName Method

#### Gets the Color Calibration Name for a color calibration ID.

##### Syntax

###### Declaration

Public Function GetColorCalibrationName(*ColorCalibrationID* As Integer) As String

###### Parameters

*ColorCalibrationID*

The color calibration’s database ID.

###### Return Value

The name of the color calibration.

###### Remarks

### GetColorShiftCorrectionList Method

#### Gets the array of ListItems of Color Calibrations/Brightness Scalings

##### Syntax

###### Declaration

Public Sub GetColorShiftCorrectionList (ByRef *ColorShiftCorrections* () As ListItem)

###### Parameters

*ColorShiftCorrections*

An array of ListItems of the database’s color shift corrections.

###### Remarks

### GetColorShiftCorrectionName Method

#### Gets the description of the Color Shift Correction Calibration when given the Color Shift Correction ID.

##### Syntax

###### Declaration

Public Function GetColorShiftCorrectionName(*ColorShiftCorrectionID* As Integer) As String

###### Parameters

*ColorShiftCorrectionID*

The color shift correction's database ID.

###### Return Value

The name of the color shift correction.

###### Remarks

### GetFlatFieldCalibrationList Method

#### Gets the array of ListItems of flat field calibrations

##### Syntax

###### Declaration

Public Sub GetFlatFieldCalibrationList(ByRef *FlatFieldCalibrations*() As ListItem)

###### Parameters

*FlatFieldCalibrations*

An array of ListItems of the database’s flat field calibrations.

###### Remarks

### GetFlatFieldCalibrationName Method

#### Returns the description of the Flat Field Calibration when given the FlatFieldCalibrationID.

##### Syntax

###### Declaration

Public Function GetFlatFieldCalibrationName(*FlatFieldCalibrationID* As Integer) As String

###### Parameters

*FlatFieldCalibrationID*

The flat field calibration’s database ID.

###### Return Value

The name of the flat field calibration.

###### Remarks

### GetImageScalingCalibrationList Method

#### Gets the array of ListItems of Image Scaling Calibrations

##### Syntax

###### Declaration

Public Sub GetImageScalingCalibrationList(ByRef *ImageScalingCalibrations*() As ListItem)

###### Parameters

*ImageScalingCalibrations*

An array of ListItems of the database’s image scaling calibrations.

###### Remarks

### GetImageScalingCalibrationName Method

#### Returns the description of the Image Scaling Calibrations when given the ImageScalingCalibrationID.

##### Syntax

###### Declaration

Public Function GetImageScalingCalibrationName(*ImageScalingCalibrationID* As Integer) As String

###### Parameters

*ImageScalingCalibrationID*

The image scaling calibration’s database ID.

###### Return Value

The name of the image scaling calibration.

###### Remarks

### GetLensDistortionCalibrationList Method

#### Gets the array of ListItems of Lens Distortion Calibrations

##### Syntax

###### Declaration

Public Sub GetLensDistortionCalibrationList(ByRef *LensDistortionCalibrations*() As ListItem)

###### Parameters

*LensDistortionCalibrations*

An array of ListItems of the database’s lens distortion calibrations.

###### Remarks

### GetLensDistortionCalibrationName Method

#### Returns the description of the Lens Distortion Calibration when given the LensDistortionCalibrationID.

##### Syntax

###### Declaration

Public Function GetLensDistortionCalibrationName(*LensDistortionCalibrationID* As Integer) As String

###### Parameters

*LensDistortionCalibrationID*

The lens distortion calibration’s database ID.

###### Return Value

The name of the lens distortion calibration.

###### Remarks

### GetMultiPointCalibrationList Method

#### Gets the array of ListItems of Multi-Point Calibrations

##### Syntax

###### Declaration

Public Sub GetMultiPointCalibrationList(ByRef *MultiPointCalibrations*() As ListItem)

###### Parameters

*MultiPointCalibrations*

An array of ListItems of the database’s multi-point calibrations.

###### Remarks

### GetMultiPointCalibrationName Method

#### Returns the description of the Multi-Point Calibration when given the MultiPointCalibrationID.

##### Syntax

###### Declaration

Public Function GetMultiPointCalibrationName(*MultiPointCalibrationID* As Integer) As String

###### Parameters

*MultiPointCalibrationID*

The multi-point calibration’s database ID.

###### Return Value

The name of the multi-point calibration.

###### Remarks

### GetStrayLightCalibrationList Method

#### Gets the array of ListItems of Stray Light Calibrations

##### Syntax

###### Declaration

Public Sub GetStrayLightCalibrationList(ByRef *StrayLightCalibrations*() As ListItem)

###### Parameters

*StrayLightCalibrations*

An array of ListItems of the database’s stray light calibrations.

###### Remarks

### GetStrayLightCalibrationName Method

#### Returns the description of the Stray Light Calibration when given the StrayLightCalibrationID.

##### Syntax

###### Declaration

Public Function GetStrayLightCalibrationName(*StrayLightCalibrationID* As Integer) As String

###### Parameters

*StrayLightCalibrationID*

The stray light calibration’s database ID.

###### Return Value

The name of the stray light calibration.

###### Remarks

### LoadColorCalibrationComboBox Method

#### Loads a ComboBox with the Color Calibration ListItems

##### Syntax

###### Declaration

Public Sub LoadColorCalibrationComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

A combobox to fill with the list of color calibrations..

###### Remarks

### LoadColorShiftCorrectionComboBox Method

#### Loads a ComboBox with the Color Shift Correction ListItems

##### Syntax

###### Declaration

Public Sub LoadColorShiftCorrectionComboBox (ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

A combobox to fill with the list of color calibrations..

###### Remarks

### LoadFlatFieldCalibrationComboBox Method

#### Loads a ComboBox with the Flat Field Calibration ListItems

##### Syntax

###### Declaration

Public Sub LoadFlatFieldCalibrationComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

A combobox to fill with the list of flat field calibrations..

###### Remarks

### LoadImageScalingCalibrationComboBox Method

#### Loads a ComboBox with the Image Scaling Calibration ListItems

##### Syntax

###### Declaration

Public Sub LoadImageScalingCalibrationComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

A combobox to fill with the list of Image Scaling calibrations..

###### Remarks

### LoadLensDistortionCalibrationComboBox Method

#### Loads a ComboBox with the Lens Distortion Calibration ListItems

##### Syntax

###### Declaration

Public Sub LoadLensDistortionCalibrationComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

A combobox to fill with the list of Lens Distortion Calibrations..

###### Remarks

### LoadMultiPointCalibrationComboBox Method

#### Loads a ComboBox with the Multi-Point Calibration ListItems

##### Syntax

###### Declaration

Public Sub LoadMultiPointCalibrationComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

A combobox to fill with the list of Multi-Point Calibrations..

###### Remarks

### LoadStrayLightCalibrationComboBox Method

#### Loads a ComboBox with the Stray Light Calibration ListItems

##### Syntax

###### Declaration

Public Sub LoadStrayLightCalibrationComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

A combobox to fill with the list of Stray Light Calibrations..

###### Remarks

### SaveFlatFieldCalibrationToDatabase Method

#### Saves the flatfield calibration, designated by MeasurementSetup.FlatFieldID, to the current Measurement database.

##### Syntax

###### Declaration

Public Sub SaveFlatFieldCalibrationToDatabase(*MeasurementSetup* As MeasurementSetup)

###### Parameters

*MeasurementSetup*

A MeasurementSetup object..

###### Remarks

### ShowAdaptiveSelectColorCalibration Method

#### Shows the Adaptive Select Color Calibration Form

##### Syntax

###### Declaration

Public Function ShowAdaptiveSelectColorCalibration(*MeasurementSetup* As MeasurementSetup) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowColorShiftCorrectionList Method

#### Shows the Color Shift Correction List Form

##### Syntax

###### Declaration

Public Sub ShowColorShiftCorrectionList()

###### Remarks

### ShowFlatFieldCalibrationList Method

#### Shows the Flat Field Calibration List Form

##### Syntax

###### Declaration

Public Sub ShowFlatFieldCalibrationList()

###### Remarks

### ShowFlatFieldCalibrationWizard Method

#### Shows the flat field calibration wizard

##### Syntax

###### Declaration

Public Function ShowFlatFieldCalibrationWizard(*MeasurementSetup* As MeasurementSetup) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Remarks

The Flat Field Calibration’s new ID (0 if cancelled)

### ShowFourColorCalibration Method

#### Shows the Four Color Calibration Form

##### Syntax

###### Declaration

Public Function ShowFourColorCalibration(*MeasurementSetup* As MeasurementSetup) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowFourColorCalibrationScaleLumWithIllum Method

#### Shows the Four Color Calibration Form, scales with illuminance

##### Syntax

###### Declaration

Public Function ShowFourColorCalibrationScaleLumWithIllum(*MeasurementSetup* As MeasurementSetup, ByRef *PhotometerDistance* As Single) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

*PhotometerDistance*

The default photometer distance.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowImageScalingCalibrationForm Method

#### Shows the Image Scaling Calibration Form

##### Syntax

###### Declaration

Public Function ShowImageScalingCalibrationForm(*MeasurementSetup* As MeasurementSetup, *AllowFocalLength* As Boolean) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

*AllowFocalLength*

Shows if the focal length method if true.

###### Return Value

The Image Scaling Calibration's new ID (0 if cancelled).

###### Remarks

### ShowImageScalingCalibrationList Method

#### Shows the Image Scaling Calibration List Form

##### Syntax

###### Declaration

Public Sub ShowImageScalingCalibrationList()

###### Remarks

### ShowLensDistortionCalibrationList Method

#### Shows the Lens Distortion Calibration List Form

##### Syntax

###### Declaration

Public Sub ShowLensDistortionCalibrationList()

###### Remarks

### ShowLuminanceScalingCalibrationList Method

#### Shows the Luminance Scaling/Color Calibration Calibration List Form

##### Syntax

###### Declaration

Public Sub ShowLuminanceScalingCalibrationList()

###### Remarks

### ShowLuminanceScalingDialog Method

#### Shows the Luminance rescaling Form

##### Syntax

###### Declaration

Public Function ShowLuminanceScalingDialog(*MeasurementSetup* As MeasurementSetup) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Remarks

### ShowLuminanceScalingDialogScaleLumWithIllum Method

#### Shows the Luminance Rescaling Form, Scales with Illuminance

##### Syntax

###### Declaration

Public Function ShowLuminanceScalingDialogScaleLumWithIllum(*MeasurementSetup* As MeasurementSetup, ByRef *PhotometerDistance* As Single) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

*PhotometerDistance*

The default photometer distance.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowMultiAreaColorCalibration Method

#### Shows the Multi-Area Color Calibration Form

##### Syntax

###### Declaration

Public Function ShowMultiAreaColorCalibration(*Measurement* As Measurement) As Integer

###### Parameters

*Measurement*

A measurement object on which to draw the multi-areas.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowMultiColorCalibration Method

#### Shows the Multi-Color Calibration Form

##### Syntax

###### Declaration

Public Function ShowMultiColorCalibration(*MeasurementSetup* As MeasurementSetup) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowMultiPointCalibration Method

#### Shows the Multi-Point Calibration Form

##### Syntax

###### Declaration

Public Sub ShowMultiPointCalibration(*MeasurementSetup* As MeasurementSetup)

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Remarks

### ShowMultiPointCalibrationList Method

#### Shows the Multi-Point Calibration List Form

##### Syntax

###### Declaration

Public Sub ShowMultiPointCalibrationList(*MeasurementSetup* As MeasurementSetup)

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Remarks

### ShowOneColorCalibration Method

#### Shows the One Color Calibration Form

##### Syntax

###### Declaration

Public Function ShowOneColorCalibration(*MeasurementSetup* As MeasurementSetup) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowOneColorCalibrationScaleLumWithIllum Method

#### Shows the One Color Calibration Form, scales with illuminance

##### Syntax

###### Declaration

Public Function ShowOneColorCalibrationScaleLumWithIllum(*MeasurementSetup* As MeasurementSetup, ByRef *PhotometerDistance* As Single) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

*PhotometerDistance*

The default photometer distance.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowSingleFilterBrightnessCalibration Method

#### Shows the single filter brightness scaling form

##### Syntax

###### Declaration

Public Function ShowSingleFilterBrightnessCalibration(*MeasurementSetup* As MeasurementSetup) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowSingleFilterBrightnessCalibrationScaleLumWithIllum Method

#### Shows the single filter brightness scaling form, scales with illuminance

##### Syntax

###### Declaration

Public Function ShowSingleFilterBrightnessCalibrationScaleLumWithIllum(*MeasurementSetup* As MeasurementSetup, ByRef *PhotometerDistance* As Single) As Integer

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

*PhotometerDistance*

The default photometer distance.

###### Return Value

The Color Calibration's New ID (0 if cancelled).

###### Remarks

### ShowStraylightCalibration Method

#### Shows the Stray Light Calibration Form

##### Syntax

###### Declaration

Public Sub ShowStraylightCalibration(*MeasurementSetup* As MeasurementSetup)

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Remarks

### ShowStraylightCalibrationList Method

#### Shows the Stray Light Calibration List Form

##### Syntax

###### Declaration

Public Sub ShowStraylightCalibrationList(*MeasurementSetup* As MeasurementSetup)

###### Parameters

*MeasurementSetup*

The parameters needed to take a measurement, needed for initial settings.

###### Remarks

## Public Properties

### ScaleFactorCol Property

#### Gets the ScaleFactorCol value.

##### Syntax

###### Declaration

Public ReadOnly Property ScaleFactorCol(*MeasurementSetup* As MeasurementSetup) As Single

###### Parameters

*MeasurementSetup*

A MeasurementSetup object from which to determine the scalefactorcol.

###### Property Value

The scalefactorcol value is applied to the measurement when it is created.

###### Remarks

The scalefactorcol property comes from the Image Scaling Calibration for luminance measurements, and from the Flat Field Calibration for Illuminance measurements.

### ScaleFactorRow Property

#### Gets the ScaleFactorRow value.

##### Syntax

###### Declaration

Public ReadOnly Property ScaleFactorRow(*MeasurementSetup* As MeasurementSetup) As Single

###### Parameters

*MeasurementSetup*

A MeasurementSetup object from which to determine the scalefactorrow.

###### Property Value

The scalefactorrow value is applied to the measurement when it is created.

###### Remarks

The scalefactorrow property comes from the Image Scaling Calibration for luminance measurements, and from the Flat Field Calibration for Illuminance measurements.

# PMMeasurement /PMMeasurementF Classes

## Background

*RIPMEngine NameSpace*

The PMMeasurement/PMMeasurementF object is the result of taking a measurement or reading from the database. It inherits the Measurement/MeasurementF Class, and in addition contains extra information about the measurement that is also written to the measurement database. Color and luminance values can be obtained from a PMMeasurement/PMMeasurementF. Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

## Constructors

### New

#### Initializes a new instance.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

### New with Measurement

#### Initializes a new instance using the data of the passed in Measurement/MeasurementF.

##### Syntax

###### Declaration

Public Sub New(m As Measurement/MeasurementF)

###### Remarks

## Public Methods

### Clone Function Method

#### Returns a new clone of the measurement object.

##### Syntax

###### Declaration

Public Shadows Function Clone() As PMMeasurement

Public Shadows Function Clone() As PMMeasurementF

###### Return Value

A cloned copy of the original measurement object.

###### Remarks

### Clone Subroutine Method

#### Clones the existing measurement into the passed in measurement.

##### Syntax

###### Declaration

Public Shadows Sub Clone(ByRef MyClone As PMMeasurement/PMMeasurementF)

###### Return Value

A cloned copy of the original measurement object is returned through the argument.

###### Remarks

### CropOut Method

#### Crops the measurement to the size of the passed in rectangle and returns it as a new measurement.

##### Syntax

###### Declaration

Public Shadows Function CropOut(SubFrameRegion As Drawing.Rectangle) As PMMeasurement

Public Shadows Function CropOut(SubFrameRegion As Drawing.Rectangle) As PMMeasurementF

###### Return Value

A copy of the measurement cropped to a smaller size.

###### Remarks

### Measurement Method

#### Creates a copy of the base Measurement/MeasurementF object.

##### Syntax

###### Declaration

Public Function Measurement() As Measurement

Public Function Measurement() As MeasurementF

###### Return Value

A clone of the base measurement object.

###### Remarks

### ShowMeasurementInformationForm Method

#### Opens the Measurement Information Form to let the user change the editable information the PMMeasurement.

##### Syntax

###### Declaration

Public Sub ShowMeasurementInformationForm(ByRef *PMEngineObj* As PMEngine)

###### Parameters

*PMEngineObj*

The base PMEngine Object (needed to save to database).

###### Return Value

###### Remarks

### ShowMeasurementInformationForm Method

#### Shows the Measurement Information Form as a dialog, does not save to database, but will return an OK if Measurement is changed

##### Syntax

###### Declaration

Public Function ShowMeasurementInformationForm() As Microsoft.VisualBasic.MsgBoxResult

###### Return Value

MsgBoxResult.OK or MsgBoxOK.Cancel

###### Remarks

### ShowMeasurementInformationForm Method

#### Shows the Measurement Information Form in the parent form, and will save the information back to the database when the user presses OK

##### Syntax

###### Declaration

Public Sub ShowMeasurementInformationForm(*ParentForm* As System.Windows.Forms.Form, ByRef *PMEngineObj* As PMEngine.PMEngine)

###### Parameters

*ParentForm*

The parent form that will be used.

*PMEngineObj*

The base PMEngine Object (needed to save to database).

###### Remarks

## Public Properties

### ColorShiftCorrectionDescription Property

#### Gets or sets the Color Shift Correction Calibration Description when taken.

##### Syntax

###### Declaration

Public Property ColorShiftCorrectionDescription () As String

###### Property Value

The Color Shift Correction Calibration Description when taken.

###### Remarks

### Current Property

#### Gets or sets the current of the device being tested.

##### Syntax

###### Declaration

Public Property Current() As Single

###### Property Value

The current of the DUT when measured. Set by the user.

###### Remarks

### FlatFieldCalibrationName Property

#### Gets or sets the flat field calibration name of the measurement when taken.

##### Syntax

###### Declaration

Public Property FlatFieldCalibrationName() As String

###### Property Value

The flat field calibration when the measurement is taken

###### Remarks

### FluxScalingFactor Property

#### Gets or sets the flux scaling factor

##### Syntax

###### Declaration

Public Property FluxScalingFactor() As Single

###### Property Value

The flux scaling factor when the measurement is taken.

###### Remarks

### ImageScalingCalDescription Property

#### Gets or sets the Image Scaling Calibration description

##### Syntax

###### Declaration

Public Property ImageScalingCalDescription() As String

###### Property Value

The image scaling calibration description is set when the measurement is taken.

###### Remarks

### LensDistortionDescription Property

#### Gets or sets the lens distortion calibration description.

##### Syntax

###### Declaration

Public Property LensDistortionDescription() As String

###### Property Value

The lens distortion calibration description is set when the measurement is taken.

###### Remarks

### MeasurementSetupDescription Property

#### Gets or sets the measurement setup description.

##### Syntax

###### Declaration

Public Property MeasurementSetupDescription() As String

###### Property Value

The measurement setup description is set when the measurement is taken.

###### Remarks

### MultiPointCalibrationDescription Property

#### Gets or sets the multi-point calibration description.

##### Syntax

###### Declaration

Public Property MultiPointCalibrationDescription() As String

###### Property Value

The multi-point calibration description is set when a measurement is taken.

###### Remarks

### ModelNumber Property

#### Gets or sets the Model Number.

##### Syntax

###### Declaration

Public Property ModelNumber() As String

###### Property Value

The model numbere is set when the measurement is taken.

###### Remarks

### SourceRatedFlux Property

#### Gets or sets the source rated flux of the device being tested.

##### Syntax

###### Declaration

Public Property SourceRatedFlux() As Single

###### Property Value

The source rated flux is set when a measurement is taken. Can be used to calculate efficiency.

###### Remarks

### StrayLightDescription Property

#### Gets or sets the Stray Light Calibration Description.

##### Syntax

###### Declaration

Public Property StrayLightDescription() As String

###### Property Value

The stray light calibration description is set when a measurement is taken.

###### Remarks

### Technician Property

#### Gets or sets the technician’s name when the measurement is taken.

##### Syntax

###### Declaration

Public Property Technician() As String

###### Property Value

The technician’s name. Set by the user.

###### Remarks

### Voltage Property

#### Gets or sets the voltage of the device being tested.

##### Syntax

###### Declaration

Public Property Voltage() As Single

###### Property Value

The voltage of the DUT when measured. Set by the user.

###### Remarks

## Operators

### Operator –

#### Returns the result of subtracting one measurement from another

##### Syntax

###### Declaration

Public Shared Overloads Operator -(*PMMeasurement1* As PMMeasurement,

*PMMeasurement2* As PMEngine.PMMeasurement) As PMMeasurement

Public Shared Overloads Operator -(*PMMeasurement1* As PMMeasurementF,

*PMMeasurement2* As PMEngine.PMMeasurement) As PMMeasurementF

###### Parameters

*PMMeasurement1*

The measurement from which to subtract

*PMMeasurement2*

The measurement to be subtracted from the first measurement

###### Returns

#### The result of subtracting the second measurement from the first.

###### Remarks

Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### Operator \*

#### Returns the result of multiplying one measurement and another

##### Syntax

###### Declaration

Public Shared Overloads Operator \*(*PMMeasurement1* As PMMeasurement,

*PMMeasurement2* As PMEngine.PMMeasurement) As PMMeasurement

Public Shared Overloads Operator \*(*PMMeasurement1* As PMMeasurementF,

*PMMeasurement2* As PMEngine.PMMeasurement) As PMMeasurementF

###### Parameters

*PMMeasurement1*

One measurement to multiply

*PMMeasurement2*

The other measurement to be multiplied.

###### Returns

#### The result of multiplying the second measurement by the first.

###### Remarks

Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### Operator /

#### Returns the result of dividing one measurement by another

##### Syntax

###### Declaration

Public Shared Overloads Operator /(*PMMeasurement1* As PMMeasurement,

*PMMeasurement2* As PMEngine.PMMeasurement) As PMMeasurement

Public Shared Overloads Operator /(*PMMeasurement1* As PMMeasurementF,

*PMMeasurement2* As PMEngine.PMMeasurement) As PMMeasurementF

###### Parameters

*PMMeasurement1*

The measurement from which to divide.

*PMMeasurement2*

The measurement that the first will be divided by.

###### Returns

#### The result of dividing the first measurement by the second.

###### Remarks

Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

### Operator +

#### Adds one measurement to another

##### Syntax

###### Declaration

Public Shared Overloads Operator +(*PMMeasurement1* As.PMMeasurement,

*PMMeasurement2* As PMEngine.PMMeasurement) As PMMeasurement

Public Shared Overloads Operator +(*PMMeasurement1* As PMMeasurementF,

*PMMeasurement2* As PMEngine.PMMeasurement) As PMMeasurementF

###### Parameters

*PMMeasurement1*

The first measurement to be added

*PMMeasurement2*

The second measurement to be added

###### Returns

#### The result of adding the two measurements.

###### Remarks

Use of PMMeasurementF rather than PMMeasurement is more performant and is recommended.

# Measurement/MeasurementF Class

## Background

*RadiantCommon NameSpace*

PMMeasurement is derived from Measurement and PMMeasurementF is derived from MeasurementF.

## Constructors

### New

#### Initializes a new instance of the Measurement/MeasurementF class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

## Public Methods

### Crop Method

#### Crops the measurement to the rectangle specified.

##### Syntax

###### Declaration

Public Overrides Sub Crop(*Rect* As Rectangle)

###### Parameters

###### *Rect*

The rectangle used to define the area of interest.

###### Remarks

### CropOut Method

#### Crops the measurement to the rectangle specified.

##### Syntax

###### Declaration

Public Function CropOut(*Rect* As Rectangle) As Measurement/MeasurementF

###### Parameters

###### *Rect*

The rectangle used to define the area of interest.

###### Return Value

A copy of the measurement cropped to a smaller size.

###### Remarks

### GetAverageColor Method

#### Gets the average color of the measurement.

##### Syntax

###### Declaration

Public Overrides Function GetAverageColor() As CIEColor

###### Parameters

###### Return Value

A CIEColor object.

###### Remarks

### GetColor Method – Single Pixel

#### Get a CIEColor object of a single pixel.

##### Syntax

###### Declaration

Public Overrides Function GetColor(*Col* As Integer, *Row* As Integer) As CIEColor

###### Parameters

*Col*

Column location of the pixel

*Row*

Row location of the pixel

###### Return Value

A CIEColor object.

###### Remarks

### GetColor Method – Single Pixel (interpolated)

#### Get a CIEColor object of a single pixel (interpolated)

##### Syntax

###### Declaration

Public Overrides Function GetColor(*Col* As Single, *Row* As Single) As CIEColor

###### Parameters

*Col*

Column location of the pixel (decimal location)

*Row*

Row location of the pixel (decimal location)

###### Return Value

A CIEColor object.

###### Remarks

The value will be interpolated from the surrounding pixels.

### GetColorArray Method (Measurement only)

#### Gets 2D arrays containing the luminance and chromaticity data for all pixels in the measurement. Not available with MeasurementF.

##### Syntax

###### Declaration

Public Function GetColorArray(*ValueUnits* As Integer, *ColorValueUnits* As Integer, ByRef Y(,) as Single, ByRef cX(,) As Single, ByRef cY(,) As Single) As Integer

###### Parameters

*ValueUnits*

Really the ‘SpectralResponseType’ enum. 0 = Radiometric; 1 = Photometric

*ColorValueUnits*

0 = Cx, Cy (CIE 1931)

1 = u, v (CIE 1960)

2 = X, Z (Tristimulus, normalized to Y)

3 = CCT (only cX array is used for CCT)

*Y*

The luminance array

*cX*

The first color array (see ColorValueUnits)

*cY*

The second color array (see ColorValueUnits)

###### Return Value

0 if successful; -1 if not successful.

###### Remarks

### GetPixel Method – Single Pixel

#### Gets the tristimulus array of a single pixel

##### Syntax

###### Declaration

Public Function GetPixel(*Col* As Integer, *Row* As Integer, *TristimulusIndex* As TristimulusType) As Single

###### Parameters

*Col*

Column location of the pixel

*Row*

Row location of the pixel

*TristimulusIndex*

Tristimulus type used in the measurement.

###### Remarks

### GetPixel Method – Single Pixel (Interpolated)

#### Gets the tristimulus array of a single pixel (interpolated).

##### Syntax

###### Declaration

Public Function GetPixel(*Col* As Single, *Row* As Single, *TristimulusIndex* As TristimulusType) As Single

###### Parameters

*Col*

Column location of the pixel (decimal location)

*Row*

Row location of the pixel (decimal location)

*TristimulusIndex*

Tristimulus type used in the measurement.

###### Remarks

### GetTristimulusArrayF Method

#### Gets the tristimulus array.

##### Syntax

###### Declaration

Public Overrides Function GetTristimulusArrayF(*TristimulusIndex* As TristimulusType) As Single(,)

###### Parameters

*TristimulusIndex*

Tristimulus type used in the measurement.

###### Remarks

## Public Properties

### BinLevelCol Property

#### Gets or sets the horizontal binning of the measurement.

##### Syntax

###### Declaration

Public Property BinLevelCol() As Short

###### Property Value

The horizontal binning of the measurement when taken.

###### Remarks

### BinLevelRow Property

#### Gets or sets the vertical binning of the measurement.

##### Syntax

###### Declaration

Public Property BinLevelRow() As Short

###### Property Value

The vertical binning of the measurement when taken.

###### Remarks

### CCDTemperature Property

#### Gets or sets the CCDTemperature of the Measurement.

##### Syntax

###### Declaration

Public Property CCDTemperature() As Single

###### Property Value

The CCD Temperature of the measurement when measured

###### Remarks

### CenterCol Property

#### Gets or sets the Center Column in Pixels from the Left side of the Measurement.

##### Syntax

###### Declaration

Public Property CenterCol() As Integer

###### Property Value

The center column of the measurement.

###### Remarks

### CenterRow Property

#### Gets or sets the Center Row in Pixels from the Top of the Measurement.

##### Syntax

###### Declaration

Public Property CenterRow() As Integer

###### Property Value

The center row of the measurement.

###### Remarks

### ColorCalibrationName Property

#### Gets or sets the name of the color calibration used when the Measurement was created

##### Syntax

###### Declaration

Public Property ColorCalibrationName() As String

###### Property Value

The color calibration name of the measurement.

###### Remarks

### Content Property

#### Gets or sets the Content of the Measurement

##### Syntax

###### Declaration

Public Property Content() As *MeasurementContentType*

###### Property Value

*MeasurementContentType*.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| Monochromatic | 1 | Single filter measurement |
| TriStimulus | 3 | Multi-filter measurement |

###### Remarks

### DatabaseID Property

#### Gets or sets the database ID of the measurement.

##### Syntax

###### Declaration

Public Property DatabaseID() As Integer

###### Property Value

The database ID of the measurement when saved.

###### Remarks

### Description Property

#### Gets or sets the description of the measurement.

##### Syntax

###### Declaration

Public Property Description() As String

###### Property Value

The description of the measurement.

###### Remarks

### DistanceUnit Property

#### Gets or sets the DistanceUnit of the measurement.

##### Syntax

###### Declaration

Public Property DistanceUnit() As DistanceUnitType

###### Property Value

The DistanceUnit of the measurement.

*DistanceUnitType*.

See the Common Enumerations section

###### Remarks

### DUTDistance Property

#### Gets or sets the DUT’s Distance

##### Syntax

###### Declaration

Public Property DUTDistance() As Single

###### Property Value

The DUT Distance

###### Remarks

The DUT Distance is used when creating an intensity measurement from illuminance. The Distance must be specified in meters.

### MeasurementDateTime Property

#### Gets or sets the date and time of the measurement when measured.

##### Syntax

###### Declaration

Public Property MeasurementDateTime() As Date

###### Property Value

The date and time of the measurement when measured.

###### Remarks

### NbrCols Property

#### Gets the number of columns in the array.

##### Syntax

###### Declaration

Public Overrides ReadOnly Property NbrCols() As Integer

###### Property Value

The number of columns.

###### Remarks

### NbrRows Property

#### Gets the number of rows in the array.

##### Syntax

###### Declaration

Public Overrides ReadOnly Property NbrRows() As Integer

###### Property Value

The number of rows.

###### Remarks

### Notes Property

#### Gets or sets the notes of the measurement.

##### Syntax

###### Declaration

Public Property Notes() As String

###### Property Value

The Notes of the measurement.

###### Remarks

### PhotometricTerm Property

#### Gets or sets the photometric term of the measurement.

##### Syntax

###### Declaration

Public Property PhotometricTerm() As PhotometricTermType

###### Property Value

The Photometric Term of the measurement.

*PhotometricTermType*

See the Common Enumerations section

###### Remarks

### PhotometricUnit Property

#### Gets or sets the photometric unit of the measurement.

##### Syntax

###### Declaration

Public Property PhotometricUnit() As *PhotometricUnitType*

###### Property Value

The Photometric unit of the measurement.

*PhotometricUnitType*

See the Common Enumerations section

###### Remarks

### SpectralResponse Property

#### Gets or sets the spectral response of the measurement.

##### Syntax

###### Declaration

Public Property SpectralResponse() As SpectralResponseType

###### Property Value

The spectral response of the measurement.

*SpectralResponseType*

See the Common Enumerations section

###### Remarks

## Operators

### Operator –

#### Returns the result of subtracting one measurement from another

##### Syntax

###### Declaration

Public Shared Overloads Operator -(*Measurement1* As Measurement,

*Measurement2* As Measurement) As Measurement

Public Shared Overloads Operator -(*Measurement1* As MeasurementF,

*Measurement2* As MeasurementF) As MeasurementF

###### Parameters

*Measurement1*

The measurement from which to subtract.

*Measurement2*

The measurement to be subtracted from the first measurement.

###### Returns

#### The result of subtracting the second measurement from the first.

###### Remarks

Use of MeasurementF rather than Measurement is more performant and is recommended.

### Operator \*

#### Returns the result of multiplying one measurement and another

##### Syntax

###### Declaration

Public Shared Overloads Operator \*(*Measurement1* As Measurement,

*Measurement2* As Measurement) As Measurement

Public Shared Overloads Operator \*(*Measurement1* As MeasurementF,

*Measurement2* As MeasurementF) As MeasurementF

###### Parameters

*Measurement1*

One measurement to multiply

*Measurement2*

The other measurement to be multiplied.

###### Returns

#### The result of multiplying the second measurement by the first.

###### Remarks

Use of MeasurementF rather than Measurement is more performant and is recommended.

### Operator /

#### Returns the result of dividing one measurement by another

##### Syntax

###### Declaration

Public Shared Overloads Operator /(*Measurement1* As Measurement,

*Measurement2* As Measurement) As Measurement

Public Shared Overloads Operator /(*Measurement1* As MeasurementF,

*Measurement2* As MeasurementF) As MeasurementF

###### Parameters

*Measurement1*

The measurement from which to divide.

*Measurement2*

The measurement that the first will be divided by.

###### Returns

#### The result of dividing the first measurement by the second.

###### Remarks

Use of MeasurementF rather than Measurement is more performant and is recommended.

### Operator +

#### Returns the result of adding one measurement to another.

##### Syntax

###### Declaration

Public Shared Overloads Operator +(*Measurement1* As Measurement,

*Measurement2* As Measurement) As Measurement

Public Shared Overloads Operator +(*Measurement1* As MeasurementF,

*Measurement2* As MeasurementF) As MeasurementF

###### Parameters

*Measurement1*

The first measurement to be added

*Measurement2*

The second measurement to be added

###### Returns

#### The result of adding the two measurements.

###### Remarks

Use of MeasurementF rather than Measurement is more performant and is recommended.

# CommonFunctions Class

## Background

*RadiantCommon NameSpace*

These are general utilitarian functions.

## Constructors

None supported. All functions are Shared.

## Public Methods

### CascadeMyWindows Method

#### Cascades the child windows of a MDIForm.

##### Syntax

###### Declaration

Public Shared Sub CascadeMyWindows(*MDIForm* As System.Windows.Forms.Form)

###### Parameters

*MDIForm*

The MDIForm of the application.

###### Remarks

### CheckComboBoxForListItemID Method

#### Checks the combox to see if it has a listitem object that has the passed in ID.

##### Syntax

###### Declaration

Public Shared Function CheckComboBoxForListItemID(*CurrentComboBox* As ComboBox, *ItemID* As Integer) As Boolean

###### Parameters

*CurrentComboBox*

The combo box to search for the selected item (ID).

*ItemID*

The ID of the ListItem to search for.

###### Remarks

### CheckListItemArrayforExisting Method

#### Checks the array of listitem objects for the passed in the Description and will return the ID if it exists.

##### Syntax

###### Declaration

Public Shared Function CheckListItemArrayforExisting(*ListItemArray* As ListItem(), *Description* As String, ByRef *ItemID* As Integer) As Boolean

###### Parameters

*ListItemArray*

The listitem array to search inside of.

*Description*

The description to search for.

*ItemID*

The ID of the listitem that corresponds to the description.

###### Returns

Returns true if the description was found in the array.

###### Remarks

If there is more than one description that matches it will only return the first description.

### CloseMyWindows Method

#### Closes the child windows of a MDIForm.

##### Syntax

###### Declaration

Public Shared Sub CloseMyWindows(*MDIForm* As System.Windows.Forms.Form)

###### Parameters

*MDIForm*

The MDIForm of the application.

###### Remarks

Does not close child windows that have the ControlBox property set to false.

### GetDescriptionFromListItemArray Method

#### Returns a string that corresponds to a description of a ListItem when given an ID.

##### Syntax

###### Declaration

Public Shared Function GetDescriptionFromListItemArray(*ListItemArray* As ListItem(), *SelectedItem* As Integer) As String

###### Parameters

*ListItemArray*

The ListsItem array to search for the selected item (ID).

*SelectedItem*

The selected item (ID) to search for

###### Return Value

The description of the listitem that corresponds to the ID of the listitem.

###### Remarks

### GetDistanceScaleTypesList Method

#### Fills an array of ListItems that contains the three DistanceScaleTypes.

##### Syntax

###### Declaration

Public Shared Sub GetDistanceScaleTypesList(ByRef *DistanceScaleTypes*() As ListItem)

###### Parameters

*DistanceScaleTypes*

ListsItem array of the DistanceScaleTypes.

###### Remarks

### GetListItemIDfromComboBox Method

#### Gets the ID of the listitem of the currently selected object of the combo box.

##### Syntax

###### Declaration

Public Shared Function GetListItemIDfromComboBox(*CurrentComboBox* As ComboBox) As Integer

###### Parameters

*CurrentComboBox*

The combo box to get the ID from

###### Returns

The ID of the ListItem that is selected in the combobox.

###### Remarks

### GetListItemIDfromToolStripComboBox Method

#### Gets the ID of the listitem of the currently selected object of the toolstrip combo box.

##### Syntax

###### Declaration

Public Shared Function GetListItemIDfromToolStripComboBox(*CurrentToolStripComboBox* As ToolStripComboBox) As Integer

###### Parameters

*CurrentToolStripComboBox*

The toolstripcombobox to get the ID from

###### Returns

The ID of the ListItem that is selected in the toolstripcombobox.

###### Remarks

### LoadComboBox Method

#### Loads a combo box with an array of ListItems.

##### Syntax

###### Declaration

Public Shared Sub LoadComboBox(ByRef *ComboBox* As ComboBox, *ListItemArray* As ListItem(), *SelectedItem* As Integer)

###### Parameters

*ComboBox*

The combo box to fill.

*ListItemArray*

The ListsItem array to fill the combo box with.

*SelectedItem*

The item to be selected.

###### Remarks

### LoadComboBoxNoSelect Method

#### Loads a combobox with an array of ListItems. Does not select an item.

##### Syntax

###### Declaration

Public Shared Sub LoadComboBoxNoSelect(ByRef *ComboBox* As ComboBox, *ListItemArray* As ListItem())

###### Parameters

*ComboBox*

The combo box to fill.

*ListItemArray*

The ListsItem array to fill the combo box with.

###### Remarks

Does not select an item so the combo box will not fire the selecteditem changed event.

### LoadDistanceScaleTypesComboBox Method

#### Loads a combo box with an array of DistanceScaleType ListItems.

##### Syntax

###### Declaration

Public Shared Sub LoadDistanceScaleTypesComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

The combo box to fill.

###### Remarks

### LoadDistanceUnitTermComboBox Method

#### Loads a combobox with an array of DistanceUnitType ListItems.

##### Syntax

###### Declaration

Public Shared Sub LoadDistanceUnitTermComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

The combo box to fill.

###### Remarks

### LoadDistanceUnitTermComboBoxNoDegrees Method

#### Loads a combobox with an array of DistanceUnitType ListItems (the degrees unit is not included)

##### Syntax

###### Declaration

Public Shared Sub LoadDistanceUnitTermComboBoxNoDegrees(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

The combo box to fill.

###### Remarks

### LoadDistanceUnitTermComboBoxOnlyDegrees Method

#### Loads a combobox with an array of DistanceUnitType ListItems (only degrees unit is included).

##### Syntax

###### Declaration

Public Shared Sub LoadDistanceUnitTermComboBoxOnlyDegrees(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

The combo box to fill.

###### Remarks

### LoadPhotometricUnitTypeComboBox Method

#### Loads a combobox with an array of PhotometricUnti ListItems and selects the passed in photometric unit.

##### Syntax

###### Declaration

Public Shared Sub LoadPhotometricUnitTypeComboBox(ByRef *ComboBox* As ComboBox, *SpectralResponse* As SpectralResponseType, *PhotometricTerm* As PhotometricTermType, *PhotometricUnit* As PhotometricUnitType)

###### Parameters

*ComboBox*

The combo box to fill.

*SpectralResponse*

SpectralResponseType Enum: See the Common Enumerations section.

*PhotometricTerm*

PhotometricTermType Enum: See the Common Enumerations section.

*PhotometricUnit*

PhotometricUnitType Enum: See the Common Enumerations section. The Photometric Unit to be selected

###### Remarks

### LoadPhotometricUnitTypeComboBoxNoSelect Method

#### Loads a combobox with an array of PhotometricUnti ListItems (does not select an item).

##### Syntax

###### Declaration

Public Shared Sub LoadPhotometricUnitTypeComboBox(ByRef *ComboBox* As ComboBox, *SpectralResponse* As SpectralResponseType, *PhotometricTerm* As PhotometricTermType)

###### Parameters

*ComboBox*

The combo box to fill.

*SpectralResponse*

SpectralResponseType Enum: See the Common Enumerations section.

*PhotometricTerm*

PhotometricTermType Enum: See the Common Enumerations section.

###### Remarks

### LoadSpectralResponseTypeComboBox Method

#### Loads a combobox with an array of SpectralResponseType ListItems.

##### Syntax

###### Declaration

Public Shared Sub LoadSpectralResponseTypeComboBox(ByRef *ComboBox* As ComboBox)

###### Parameters

*ComboBox*

The combo box to fill.

###### Remarks

### LoadToolStripComboBox Method

#### Loads a toolstripcombobox with an array of ListItems. Selects the item passed in.

##### Syntax

###### Declaration

Public Shared Sub LoadToolStripComboBox(ByRef *ToolStripComboBox* As ToolStripComboBox, *ListItemArray* As ListItem(), *SelectedItem* As Integer)

###### Parameters

*ToolStripComboBox*

The toolstrip combo box to fill.

*ListItemArray*

The ListsItem array to fill the combobox with.

*SelectedItem*

The item to be selected.

###### Remarks

### LoadToolStripComboBoxNoSelect Method

#### Loads a toolstripcombobox with an array of ListItems. Does not select an item.

##### Syntax

###### Declaration

Public Shared Sub LoadToolStripComboBoxNoSelect(ByRef *ToolStripComboBox* As ToolStripComboBox, *ListItemArray* As ListItem())

###### Parameters

*ToolStripComboBox*

The toolstripcombobox to fill.

*ListItemArray*

The ListsItem array to fill the combobox with.

###### Remarks

### MaximizeMyWindows Method

#### Maximizes the child windows of a MDIForm.

##### Syntax

###### Declaration

Public Shared Sub MaximizeMyWindows(*MDIForm* As System.Windows.Forms.Form)

###### Parameters

*MDIForm*

The MDIForm of the application.

###### Remarks

Does not maximize child windows that have the ControlBox or MaximizeBox property set to false.

### MinimizeMyWindows Method

#### Minimizes the child windows of a MDIForm.

##### Syntax

###### Declaration

Public Shared Sub MinimizeMyWindows(*MDIForm* As System.Windows.Forms.Form)

###### Parameters

*MDIForm*

The MDIForm of the application.

###### Remarks

Does not minimize child windows that have the ControlBox or MinimizeBox property set to false.

### SetComboBoxtoListItemID Method

#### Sets the combox selected item to the listitem object that has the passed in ID.

##### Syntax

###### Declaration

Public Shared Sub SetComboBoxtoListItemID(*CurrentComboBox* As ComboBox, *ItemID* As Integer)

###### Parameters

*CurrentComboBox*

The combobox to set the selected item.

*ItemID*

The ID of the ListItem to search for and select

###### Remarks

### SetToolStripComboBoxtoListItemID Method

#### Sets the toolstripcombox selected item to the listitem object that has the passed in ID.

##### Syntax

###### Declaration

Public Shared Sub SetToolStripComboBoxtoListItemID(*CurrentToolStripComboBox* As ToolStripComboBox, *ItemID* As Integer)

###### Parameters

*CurrentToolStripComboBox*

The toolstripcombobox to search for the selected item (ID).

*ItemID*

The ID of the ListItem to search for and select.

###### Remarks

### TileMyWindows Method

#### Tiles the child windows of a MDIForm.

##### Syntax

###### Declaration

Public Shared Sub TileMyWindows(*MDIForm* As System.Windows.Forms.Form)

###### Parameters

*MDIForm*

The MDIForm of the application.

###### Remarks

### Wait Method

#### Pauses the program for the designate amount of time

##### Syntax

###### Declaration

Public Shared Sub Wait(*WaitTimeMilliseconds* As Integer)

###### Parameters

*WaitTimeMilliseconds*

The amount of time to pause in milliseconds.

###### Remarks

DoEvents is repeatedly called.

# RegionOfInterest Class

## Background

The RegionOfInterest Class is the base class for the different types (shapes) of Virtual Detectors. The methods in this class are common to the other types of regionofinterest classes. The following regionofinterest classes are currently supported:

* ROICircle
* ROIEllipse
* ROIEntireImage
* ROIRectangle

## Constructors

None supported. See constructors for inheriting classes.

## Public Properties

### Center Property

#### Gets or sets the center location of the regionofinterest.

##### Syntax

###### Declaration

Public Property Center(*m* As MeasurementBase) As PointF

###### Parameters

*m*

A measurement object.

###### Property Value

A PointF object.

###### Remarks

### DistanceScale Property

#### Gets or sets the geometry distance scale.

##### Syntax

###### Declaration

Public Property DistanceScale() As DistanceScaleType

###### Property Value

*DistanceScaleType Enum*

See the Common Enumerations section.

###### Remarks

### DistanceUnit Property

#### Gets or sets the geometry distance unit.

##### Syntax

###### Declaration

Public Property DistanceUnit() As DistanceUnitType

###### Property Value

DistanceScaleUnit.

###### Remarks

### Location Property

#### Gets or sets the location of the regionofinterest. This is usually associated with the top left corner of the regionofinterest.

##### Syntax

###### Declaration

Public Property Location() As PointF

###### Property Value

The location of the regionofinterest.

###### Remarks

### LocationDistanceScale Property

#### Gets or sets the location distance scale.

##### Syntax

###### Declaration

Public Property LocationDistanceScale() As DistanceScaleType

###### Property Value

*DistanceScaleType Enum*

See the Common Enumerations section.

###### Remarks

This property is linked with the Location property.

### LocationDistanceUnit Property

#### Gets or sets the location distance unit.

##### Syntax

###### Declaration

Public Property LocationDistanceUnit() As *DistanceUnitType*

###### Property Value

*DistanceUnitType Enum*

See the Common Enumerations section.

###### Remarks

This property is only relevant when the LocationDistanceScale is set to Physical.

### Sizemm Property

#### Returns a generic size value in millimeters.

##### Syntax

###### Declaration

Public MustOverride ReadOnly Property Sizemm(*m* As MeasurementBase) As Single

###### Parameters

*m*

A measurement object.

###### Property Value

The size in mm of the regionofinterest.

###### Remarks

## Public Methods

### ChangeLocationDistanceScaleType Method

#### Changes the location distance scale type to the new type. Alters the location.

##### Syntax

###### Declaration

Public Sub ChangeLocationDistanceScaleType(*DistanceScale* As DistanceScaleType, *m* As MeasurementBase)

###### Parameters

*DistanceScale*

*DistanceScaleType Enum*

See the Common Enumerations section.

*m*

A measurement object.

###### Remarks

Alters the location

### ChangeLocationDistanceUnitType Method

#### Changes the location distance unit. Alters the location.

##### Syntax

###### Declaration

Public Sub ChangeLocationDistanceUnitType(*DistanceUnit* As DistanceUnitType)

###### Parameters

*DistanceScale*

*DistanceUnitType Enum*

See the Common Enumerations section.

*m*

A measurement object.

###### Remarks

### GetColor Method

#### Gets a CIEColor object.

##### Syntax

###### Declaration

Public MustOverride Function GetColor(*m* As MeasurementBase) As CIEColor

###### Parameters

*m*

A measurement object.

###### Remarks

### Clone Method

#### Makes a copy of the regionofinterest.

##### Syntax

###### Declaration

Public MustOverride Function Clone() As RegionOfInterest

###### Remarks

# ROICircle Class

## Background

This class is used for created a circular virtual detector. It inherits the RegionOfInterest Class and will have all of those properties. It also includes the following properties and methods.

## Constructors

### New

#### Initializes a new instance of the ROICircle class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

### New with DistanceScale and Diameter

#### Initializes a new instance of the ROICircle class. Sets the distancecale property and diameter.

##### Syntax

###### Declaration

Sub New(*DistanceScale* As DistanceScaleType, *Diameter* As Single)

###### Parameters

*DistanceScale*

*DistanceScaleType Enum*

See the Common Enumerations section.

*Diameter*

The diameter of the circle (ScaleType is specified by the DistanceScaleType)

###### Remarks

### New with DistanceUnit and Diameter

#### Initializes a new instance of the ROICircle class. Sets the distanceunit property and diameter.

##### Syntax

###### Declaration

Sub New(*DistanceUnit* As DistanceUnitType, *Diameter* As Single)

###### Parameters

*DistanceUnit*

*DistanceUnit*T*ype Enum*

See the Common Enumerations section.

*Diameter*

The diameter of the circle (units are specified by the DistanceUnitType)

###### Remarks

## Public Properties

### Diameter Property

#### Gets or sets the circle’s diameter.

##### Syntax

###### Declaration

Public Property Diameter() As Single

###### Property Value

The diameter of the circle.

###### Remarks

The distancescaletype and distanceunittype are specified by the distancescale and distanceunit properties.

### Radius Property

#### Gets or sets the circle’s radius.

##### Syntax

###### Declaration

Public Property Radius() As Single

###### Property Value

The radius of the circle.

###### Remarks

The distancescaletype and distanceunittype are specified by the distancescale and distanceunit properties.

# ROIRectangle Class

## Background

This class is used for creating a rectangular virtual detector. It inherits the RegionOfInterest Class and will have all of those properties. It also includes the following properties and methods.

## Constructors

### New

#### Initializes a new instance of the ROIRectangle class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

### New with DistanceScale, width and height

#### Initializes a new instance of the ROIRectangle class. Sets the distancecale property, width and height.

##### Syntax

###### Declaration

Sub New(*DistanceScale* As DistanceScaleType, Width As Single, Height As Single)

###### Parameters

*DistanceScale*

*DistanceScaleType Enum*

See the Common Enumerations section.

*Width*

The width of the rectangle (ScaleType is specified by the DistanceScaleType)

*Height*

The height of the rectangle (ScaleType is specified by the DistanceScaleType)

###### Remarks

### New with DistanceUnit, width and height

#### Initializes a new instance of the ROIRectangle class. Sets the distanceunit property, width and height.

##### Syntax

###### Declaration

Sub New(*DistanceUnit* As DistanceUnitType, Width As Single, Height As Single)

###### Parameters

*DistanceUnit*

*DistanceUnit*T*ype Enum*

See the Common Enumerations section.

*Width*

The width of the rectangle (DistanceUnit is specified by the DistanceUnitType)

*Height*

The height of the rectangle (DistanceUnit is specified by the DistanceUnitType)

###### Remarks

### New with System.Drawing.Rectangle

#### Initializes a new instance of the ROIRectangle class. Sets the distancescale to pixels. Set the location, width and height to passed in rectangle.

##### Syntax

###### Declaration

Public Sub New(*Rect* As Rectangle)

###### Parameters

*Rect*

System.drawing.rectangle

###### Remarks

The scaletype will be set to pixels.

## Public Properties

### Height Property

#### Gets or sets the rectangle’s height.

##### Syntax

###### Declaration

Public Property Height() As Single

###### Property Value

The height of the rectangle.

###### Remarks

The distancescaletype and distanceunittype are specified by the distancescale and distanceunit properties.

### Width Property

#### Gets or sets the rectangle’s Width.

##### Syntax

###### Declaration

Public Property Width() As Single

###### Property Value

The Width of the rectangle.

###### Remarks

The distancescaletype and distanceunittype are specified by the distancescale and distanceunit properties.

# ROIEllipse Class

## Background

This class is used for creating a elliptical virtual detector. It inherits the RegionOfInterest Class and will have all of those properties. It also includes the following properties and methods.

## Constructors

### New

#### Initializes a new instance of the ROIEllipse class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

### New with DistanceScale, width and height

#### Initializes a new instance of the ROIEllipse class. Sets the distancecale property, width and height.

##### Syntax

###### Declaration

Sub New(*DistanceScale* As DistanceScaleType, Width As Single, Height As Single)

###### Parameters

*DistanceScale*

*DistanceScaleType Enum*

See the Common Enumerations section.

*Width*

The width of the ellipse (ScaleType is specified by the DistanceScaleType)

*Height*

The height of the ellipse (ScaleType is specified by the DistanceScaleType)

###### Remarks

### New with System.Drawing.Rectangle

#### Initializes a new instance of the ROIEllipse class. Sets the distancescale to pixels. Set the location, width and height to passed in ellipse.

##### Syntax

###### Declaration

Public Sub New(*Rect* As Rectangle)

###### Parameters

*Rect*

System.drawing.rectangle

###### Remarks

The scaletype will be set to pixels.

## Public Properties

### Height Property

#### Gets or sets the ellipse’s height.

##### Syntax

###### Declaration

Public Property Height() As Single

###### Property Value

The height of the ellipse.

###### Remarks

The distancescaletype and distanceunittype are specified by the distancescale and distanceunit properties.

### Width Property

#### Gets or sets the ellipse’s Width.

##### Syntax

###### Declaration

Public Property Width() As Single

###### Property Value

The Width of the ellipse.

###### Remarks

The distancescaletype and distanceunittype are specified by the distancescale and distanceunit properties.

# ROIEntireImage Class

## Background

This class is used for getting average values from the entire measurement. It inherits the RegionOfInterest Class, but some properties may not apply. It also includes the following properties and methods.

## Constructors

### New

#### Initializes a new instance of the ROIEntireImage class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

# CIEColor Class

## Background

This class is used getting luminance and CIE data. It is usually calculated from a Measurement and RegionofInterest..

## Constructors

### New

#### Initializes a new instance of the CIEColor class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

### New with Color Coordinates

#### Initializes a new instance of the CIEColor class. Color coordinates are passed in.

##### Syntax

###### Declaration

Public Sub New(*u\_Or\_x* As Single, *v\_OR\_y* As Single, *CS* As ColorSpace)

###### Parameters

*u\_Or\_x*

u’ or Cx value.

*v\_OR\_y*

v’ or Cy value.

*CS*

*ColorSpace Enum*

See the Common Enumerations section.

###### Remarks

### New with Luminance and Color Coordinates

#### Initializes a new instance of the CIEColor class. Luminance and color coordinates are passed in.

##### Syntax

###### Declaration

Public Sub New(*Lv* As Single, *u\_Or\_x* As Single, *v\_OR\_y* As Single, *CS* As ColorSpace)

###### Parameters

*Lv*

Brightness value (luminance, illuminance).

*u\_Or\_x*

u’ or Cx value.

*v\_OR\_y*

v’ or Cy value.

*CS*

*ColorSpace Enum*

See the Common Enumerations section.

###### Remarks

### New with Tristimulus Values

#### Initializes a new instance of the CIEColor class. Tristimulus values are passed in.

##### Syntax

###### Declaration

Public Sub New(*TrisX* As Single, *TrisY* As Single, *TrisZ* As Single)

###### Parameters

*TrisX*

Tristimulus X value.

*TrisY*

Tristimulus Y value.

*TrisZ*

Tristimulus Z value.

###### Remarks

## Public Methods

### astar Method

#### Returns a\* of CIELAB color space.

##### Syntax

###### Declaration

Public Function astar(*CIEColorRef* As CIEColor) As Single

###### Parameters

*CIEColorRef*

Reference achromatic color.

###### Return Value

The calculated a\* value.

###### Remarks

### bstar Method

#### Returns b\* of CIELAB color space.

##### Syntax

###### Declaration

Public Function bstar(*CIEColorRef* As CIEColor) As Single

###### Parameters

*CIEColorRef*

Reference achromatic color.

###### Return Value

The calculated b\* value.

###### Remarks

### CalcColorError Method

#### Returns b\* of CIELAB color space.

##### Syntax

###### Declaration

Public Function CalcColorError(*ReferenceColor* As CIEColor, *A* As Single, *B* As Single, *ThetaRad* As Single) As Single

###### Parameters

*ReferenceColor*

Reference color.

*A*

A parameter needed to calculate color error.

*B*

B parameter needed to calculate color error.

*ThetaRad*

Theta (in radians) parameter needed to calculate color error.

###### Return Value

The calculated b\* value.

###### Remarks

### Clone Method

#### Returns a clone of the current CIEColor object.

##### Syntax

###### Declaration

Public Function Clone() As CIEColor

###### Return Value

A cloned copy of the original CIEColor object.

###### Remarks

### GetLuv Method

#### Gets the L u' and v' values.

##### Syntax

###### Declaration

Public Function GetLuv(ByRef *Lv* As Single, ByRef *uvalue* As Single, ByRef *vvalue* As Single) As Short

###### Parameters

*Lv*

Brightness value.

*uvalue*

u' value.

*vvalue*

v' value.

###### Return Value

1 if successful, 0 if not.

###### Remarks

### GetLxy Method

#### Gets the L Cx and Cy values.

##### Syntax

###### Declaration

Public Function GetLuv(ByRef *Lv* As Single, ByRef *CxVal*As Single, ByRef *CyVal* As As Single) As Short

###### Parameters

*Lv*

Brightness value.

*CxVal*As

Cx value.

*CyVal*As

Cy value.

###### Return Value

1 if successful, 0 if not.

###### Remarks

### Lstar Method

#### Returns L\* of CIELAB color space.

##### Syntax

###### Declaration

Public Function Lstar(*CIEColorRef* As CIEColor) As Single

###### Parameters

*CIEColorRef*

Reference achromatic color.

###### Return Value

The calculated L\* value.

###### Remarks

### Scale Method

#### Applies a scaling factor to the Tristimulus Values of the CIEColor object.

##### Syntax

###### Declaration

Public Sub Scale(*ScaleFactor* As Single)

###### Parameters

*ScaleFactor*

The scale factor multiplies the Tristimulus X, Y and Z values.

###### Remarks

### ustar Method

#### Returns u\* of CIELUV color space.

##### Syntax

###### Declaration

Public Function ustar(*CIEColorRef* As CIEColor) As Single

###### Parameters

*CIEColorRef*

Reference achromatic color.

###### Return Value

The calculated u\* value.

###### Remarks

### vstar Method

#### Returns v\* of CIELUV color space.

##### Syntax

###### Declaration

Public Function ustar(*CIEColorRef* As CIEColor) As Single

###### Parameters

*CIEColorRef*

Reference achromatic color.

###### Return Value

The calculated v\* value.

###### Remarks

## Public Properties

### CCTemp Property

#### Gets the color coordinate temperature.

##### Syntax

###### Declaration

Public Property CCTemp() As Single

###### Property Value

The color coordinate temperature.

###### Remarks

### Cx Property

#### Gets the CIE 1931 Cx color coordinate.

##### Syntax

###### Declaration

Public ReadOnly Property Cx() As Single

###### Property Value

The CIE 1931 Cx.

###### Remarks

### Cy Property

#### Gets the CIE 1931 Cy color coordinate.

##### Syntax

###### Declaration

Public ReadOnly Property Cy() As Single

###### Property Value

The CIE 1931 Cy.

###### Remarks

### Cz Property

#### Gets the CIE 1931 Cz color coordinate.

##### Syntax

###### Declaration

Public ReadOnly Property Cz() As Single

###### Property Value

The CIE 1931 Cz.

###### Remarks

### DeltaUV Property

#### Gets the delta uv value from another CIE object.

##### Syntax

###### Declaration

Public ReadOnly Property DeltaUV(*CIEReference* As CIEColor) As Single

###### Parameters

*CIEReference*

The CIE color object that is the reference value.

###### Property Value

The delta uv value from another CIEColor object.

###### Remarks

### DeltaUVfromCCT Property

#### Gets the delta uv from CCT value.

##### Syntax

###### Declaration

Public ReadOnly Property DeltaUVfromCCT() As Single

###### Property Value

The delta uv from CCT value.

###### Remarks

This value is the distance in uv coordinate system and the CCT.

### DominantWavelength Property

#### Gets the dominant wavelength of the color.

##### Syntax

###### Declaration

Public ReadOnly Property DominantWavelength() As Integer

###### Property Value

The dominant wavelength of the color.

###### Remarks

### Lv Property

#### Gets the Brightness (luminance, illuminance).

##### Syntax

###### Declaration

Public ReadOnly Property Lv() As Single

###### Property Value

The brightness value (luminance, illuminance).

###### Remarks

### u Property

#### Gets the CIE 1976 u’ color coordinate.

##### Syntax

###### Declaration

Public ReadOnly Property u() As Single

###### Property Value

The CIE 1976 u’.

###### Remarks

### u1960 Property

#### Gets the CIE 1960 u color coordinate.

##### Syntax

###### Declaration

Public ReadOnly Property u1960() As Single

###### Property Value

The CIE 1960 u.

###### Remarks

### v Property

#### Gets the CIE 1976 v’ color coordinate.

##### Syntax

###### Declaration

Public ReadOnly Property v() As Single

###### Property Value

The CIE 1976 v’.

###### Remarks

### v1960 Property

#### Gets the CIE 1960 v color coordinate.

##### Syntax

###### Declaration

Public ReadOnly Property v1960() As Single

###### Property Value

The CIE 1960 v.

###### Remarks

### X Property

#### Gets or sets the Tristimulus X value.

##### Syntax

###### Declaration

Public Property X() As Single

###### Property Value

The Tristimulus X value..

###### Remarks

### Y Property

#### Gets or sets the Tristimulus Y value.

##### Syntax

###### Declaration

Public Property Y() As Single

###### Property Value

The Tristimulus Y value..

###### Remarks

### Z Property

#### Gets or sets the Tristimulus Z value.

##### Syntax

###### Declaration

Public Property Z() As Single

###### Property Value

The Tristimulus Z value..

###### Remarks

## Operators

### Operator –

#### Returns the result of subtracting one color point from another

##### Syntax

###### Declaration

Public Shared Operator -(*Color1* As CIEColor, *Color2* As CIEColor) As Single

###### Parameters

*Color1*

The color point from which to subtract.

*Color2*

The color point to be subtracted from the first color point.

###### Returns

#### The result of subtracting the second color point from the first.

###### Remarks

### Operator \*

#### Returns the result of multiplying one color point and another

##### Syntax

###### Declaration

Public Shared Operator \*(*Color1* As CIEColor, *Color2* As CIEColor) As CIEColor

###### Parameters

*Color1*

One color point to multiply

*Color2*

The other color point to be multiplied.

###### Returns

#### The result of multiplying the second color point by the first.

###### Remarks

### Operator /

#### Returns the result of dividing one color point by another

##### Syntax

###### Declaration

Public Shared Operator /(*Color1* As CIEColor,

*Color2* As CIEColor) As CIEColor

###### Parameters

*Color1*

The color point from which to divide.

*Color2*

The color point that the first will be divided by.

###### Returns

#### The result of dividing the first color point by the second.

###### Remarks

### Operator +

#### Adds one color point to another

##### Syntax

###### Declaration

Public Shared Operator +(*Color1* As CIEColor,

*Color2* As CIEColor) As CIEColor

###### Parameters

*Color1*

The first color point to be added

*Color2*

The second color point to be added

###### Returns

#### The result of adding the two color points.

###### Remarks

### Operator <>

#### Checks if two color points are not equal.

##### Syntax

###### Declaration

Public Shared Operator <>(*Color1* As CIEColor, *Color2* As CIEColor) As Boolean

###### Parameters

*Color1*

The first color point.

*Color2*

The second color point to be compared to the first.

###### Returns

#### The result of comparing the two points, returns True if they are not equal, False if they are.

###### Remarks

### Operator =

#### Checks if two color points are equal.

##### Syntax

###### Declaration

Public Shared Operator =(*Color1* As CIEColor, *Color2* As CIEColor) As Boolean

###### Parameters

*Color1*

The first color point.

*Color2*

The second color point to be compared to the first.

###### Returns

#### The result of comparing the two points, returns True if they are equal, False if they are not.

###### Remarks

# ListItem Class

## Background

The ListItem Class is used by many functions of PMEngine to provide a coupled description and ID. It is often used with the functions in CommonFunctions to load and read comboboxes.

## Constructors

### New

#### Initializes a new instance of the ListItem class.

##### Syntax

###### Declaration

Public Sub New()

###### Remarks

### New with ID and Description

#### Initializes a new instance of the ListItem class.

##### Syntax

###### Declaration

Public Sub New(*ID* As Integer, *Description* As String)

###### Parameters

*ID*

An Integer ID.

*Description*

A string description.

###### Remarks

## Public Properties

### Description Property

#### Gets or sets the Description.

##### Syntax

###### Declaration

Public Property Description() As String

###### Property Value

A string description.

###### Remarks

### ID Property

#### Gets or sets the integer ID.

##### Syntax

###### Declaration

Public Property ID() As Integer

###### Property Value

An integer ID.

###### Remarks

# Common Enumerations

## ColorSpace Enum

The color space type.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| CIE1931xy | 1 | The 1931 x, y color coordinate system |
| CIE1976uv | 2 | The 1976 u’, v’ color coordinate system |
| Tristimulus | 3 | Tristimulus X, Y, Z. |

## DigitizingSpeed Enum

The digitizing speed of the camera.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| Slow | 0 | The slowest speed. |
| Medium | 1 | The medium speed, not supported for all cameras. |
| Fast | 2 | The fastest speed (Standard for I and Y series) |

## DistanceScaleTypeEnum

The distance scale type of a region of interest.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| Pixels | 0 | Location in pixels |
| Physical | 1 | Location in physical units. |
| Relative | 2 | Location as a fraction of total. |

## DistanceUnitTypeEnum

The Distance Unit Type for a measurement.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| Centimeters | 2 | Centimeters |
| Degrees | 5 | Only valid for intensity measurements. |
| Feet | 3 | Feet |
| Inches | 1 | Inches |
| Meters | 0 | Meters |
| Millimeters | 4 | Millimeters |

## GoniometerTypeEnum

The Goniometer Type for making an illuminance measurement into a intensity measurement.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| SIG | 0 | Not supported for PM |
| TypeA | 1 | Type A goniometer |
| TypeB | 2 | Type B goniometer |
| TypeC | 3 | Type C goniometer |
| ProjectedAngles | 4 | Projected angles coordinate system |

## PhotometricTermType Enum

The Photometric Term of the measurement.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| CCT | 13 | Not supported for PM |
| ContrastRatio | 42 | Unit after dividing one measurement from another |
| Gray | 43 | Not supported for PM |
| Illuminance | 0 | An illuminance type measurement |
| Intensity | 2 | An intensity type measurement (created from an illuminance measurement) |
| Lumens | 3 | Not supported for PM |
| Luminance | 1 | A luminance type measurement |
| Millilumens | 4 | Not supported for PM |
| Mura | 40 | Not supported for PM |
| Percent | 41 | Not supported for PM |
| Scatter | 30 | Not supported for PM |

## PhotometricUnitTypeEnum

The photometric unit of a measurement.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| Illuminance\_FtCandles | 0 | FootCandles or W/ft2 – Illuminance measurements |
| Illuminance\_KiloLux | 3 | Kilolux or kW/m2 – Illuminance measurements |
| Illuminance\_LumensPerCm | 6 | Lumens/cm2 or W/cm2 – Illuminance measurements |
| Illuminance\_Lux | 1 | Lux or W/m2 – Illuminance measurements |
| Illuminance\_LuxSec | 5 | Lux-Sec or W-s/m2 – Illuminance energy measurements |
| Illuminance\_Megalux | 4 | Megalux or MW/m2 – Illuminance measurements |
| Illuminance\_MilliLux | 2 | Millilux or mW/m2 – Illuminance measurements |
| Intensity\_Candela | 20 | Candela or W/s – Intensity measurements |
| Intensity\_MilliCandela | 21 | Millicandela or mW/s – Intensity measurements |
| Luminance\_CandelaPerCm | 16 | Candela /cm2 or W/sr/cm2 – Luminance measurements |
| Luminance\_CandelaPerMeter | 17 | Candela/m2 or W/sr/m2 – Luminance measurements |
| Luminance\_Footlambert | 10 | Footlambert or W/sr/ft2 – Luminance measurements |
| Luminance\_KiloNit | 13 | Knit or kW/sr/m2 – Luminance measurements |
| Luminance\_MegaNit | 14 | Meganit or MW/sr/m2 – Luminance measurements |
| Luminance\_MilliNit | 12 | Millinit or mW/sr/m2 – Luminance measurements |
| Luminance\_MilliCandelaPerCm | 16 | Millicandela/cm2 or mW/sr/cm2 – Luminance measurements |
| Luminance\_Nit | 11 | Nit or W/sr/m2 – Luminance measurements |

## SpectralResponseTypeEnum

The spectral response of a measurement.

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| Photometric | 1 | Photometric type measurements |
| Radiometric | 0 | Radiometric type measurements |