

# MBS Real Studio LCMS2 Plugin Documentation

Christian Schmitz

May 15, 2012

## 0.1 Introduction

This is the PDF version of the documentation for the Real Studio Plug-in from Monkeybread Software Germany. Plugin part: MBS Real Studio LCMS2 Plugin

## 0.2 Content

- 1 List of all topics 3
- 2 All items in this plugin 29
- 3 List of all classes 241
- 4 List of all modules 243

# Chapter 1

## List of Topics

• 2 LCMS2	29
– 2.16 module LCMS2MBS	108
* 2.16.1 AdaptationMatrix(ConeMatrix as LCMS2Mat3MBS, FromIll as LCMS2CIEXYZMBS, ToIll as LCMS2CIEXYZMBS) as LCMS2Mat3MBS	108
* 2.16.1 AdaptToIlluminant(SourceWhitePt as LCMS2CIEXYZMBS, Illuminant as LCMS2CIEXYZMBS, Value as LCMS2CIEXYZMBS) as LCMS2CIEXYZMBS	109
* 2.16.1 BFDdeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS) as double	109
* 2.16.1 BuildRGB2XYZtransferMatrix(WhitePoint as LCMS2CIExyYMBS, Primaries as LCMS2CIExyYTriple as LCMS2Mat3MBS	109
* 2.16.1 BYTES_ SH(n as UInt32) as UInt32	110
* 2.16.1 ChannelsOf(ColorSpaceSignature as integer) as UInt32	110
* 2.16.1 CHANNELS_ SH(n as UInt32) as UInt32	110
* 2.16.1 CIE2000DeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS, Kl as double, Kc as double, Kh as double) as double	110
* 2.16.1 CIE94DeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS) as double	111
* 2.16.1 CMCdeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS, l as double, c as double) as double	111
* 2.16.1 COLORSPACE_ SH(n as UInt32) as UInt32	111
* 2.16.1 CreateBitmapFromPicture(p as picture, bits as integer = 8) as LCMS2BitmapMBS	111
* 2.16.1 D50_ xyY as LCMS2CIExyYMBS	112
* 2.16.1 D50_ XYZ as LCMS2CIEXYZMBS	112
* 2.16.1 DeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS) as double	112
* 2.16.1 DOSWAP_ SH(n as UInt32) as UInt32	113
* 2.16.1 ENDIAN16_ SH(n as UInt32) as UInt32	113

* 2.16.1 EXTRA_ SH(n as UInt32) as UInt32	113
* 2.16.1 FLAVOR_ SH(n as UInt32) as UInt32	113
* 2.16.1 Float2LabEncoded(c as LCMS2CIELabMBS) as integer()	114
* 2.16.1 Float2LabEncodedV2(c as LCMS2CIELabMBS) as integer()	114
* 2.16.1 Float2XYZEncoded(c as LCMS2CIEXYZMBS) as integer()	114
* 2.16.1 FLOAT_ SH(n as UInt32) as UInt32	114
* 2.16.1 GetAlarmCodes as integer()	114
* 2.16.1 kcmsD50X as double	115
* 2.16.1 kcmsD50Y as double	115
* 2.16.1 kcmsD50Z as double	115
* 2.16.1 kcmsPERCEPTUAL_ BLACK_ X as double	115
* 2.16.1 kcmsPERCEPTUAL_ BLACK_ Y as double	115
* 2.16.1 kcmsPERCEPTUAL_ BLACK_ Z as double	115
* 2.16.1 Lab2LCh(p as LCMS2CIELabMBS) as LCMS2CIELChMBS	116
* 2.16.1 Lab2XYZ(p as LCMS2CIELabMBS, whitepoint as LCMS2CIEXYZMBS = nil) as LCMS2CIEXYZMBS	116
* 2.16.1 LabEncoded2Float(w0 as UInt16, w1 as UInt16, w2 as UInt16) as LCMS2CIELabMBS	116
* 2.16.1 LabEncoded2FloatV2(w0 as UInt16, w1 as UInt16, w2 as UInt16) as LCMS2CIELabMBS	116
* 2.16.1 LCh2Lab(p as LCMS2CIELChMBS) as LCMS2CIELabMBS	116
* 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer) as LCMS2BitmapMBS	117
* 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer, RowBytes as integer) as LCMS2BitmapMBS	117
* 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer, RowBytes as integer, data as memoryblock) as LCMS2BitmapMBS	117
* 2.16.1 OPTIMIZED_ SH(n as UInt32) as UInt32	118
* 2.16.1 PixelFormat(FloatingPoint as boolean, Optimized as boolean, ColorSpace as UInt32, MinIsWhite as boolean, Planar as boolean, EndianSwap as boolean, DoSwap as boolean, ExtraSamples as UInt32, Channels as UInt32, BytesPerSample as UInt32, SwapFirst as boolean) as UInt32	118
* 2.16.1 PLANAR_ SH(n as UInt32) as UInt32	118
* 2.16.1 SetAdaptationState(d as double) as double	119
* 2.16.1 SetAlarmCodes(values() as integer)	119
* 2.16.1 SetLogErrorHandler(handler as LCMS2ErrorHandlerMBS)	119
* 2.16.1 SWAPFIRST_ SH(n as UInt32) as UInt32	120
* 2.16.1 TagInteger(tag as string) as UInt32	120
* 2.16.1 TagString(tag as UInt32) as string	120
* 2.16.1 TempFromWhitePoint(TempK as LCMS2CIExyYMBS) as double	120
* 2.16.1 T_ BYTES(n as UInt32) as UInt32	121
* 2.16.1 T_ CHANNELS(n as UInt32) as UInt32	121
* 2.16.1 T_ COLORSPACE(n as UInt32) as UInt32	121

* 2.16.1 T_ DOSWAP(n as UInt32) as UInt32	121
* 2.16.1 T_ ENDIAN16(n as UInt32) as UInt32	122
* 2.16.1 T_ EXTRA(n as UInt32) as UInt32	122
* 2.16.1 T_ FLAVOR(n as UInt32) as UInt32	122
* 2.16.1 T_ FLOAT(n as UInt32) as UInt32	122
* 2.16.1 T_ OPTIMIZED(n as UInt32) as UInt32	122
* 2.16.1 T_ PLANAR(n as UInt32) as UInt32	123
* 2.16.1 T_ SWAPFIRST(n as UInt32) as UInt32	123
* 2.16.1 Version as string	123
* 2.16.1 WhitePointFromTemp(TempK as double) as LCMS2CIExyYMBS	123
* 2.16.1 xyY2XYZ(p as LCMS2CIExyYMBS) as LCMS2CIEXYZMBS	124
* 2.16.1 XYZ2Lab(p as LCMS2CIEXYZMBS, whitepoint as LCMS2CIEXYZMBS = nil) as LCMS2CIELabMBS	124
* 2.16.1 XYZ2xyY(p as LCMS2CIEXYZMBS) as LCMS2CIExyYMBS	124
* 2.16.1 XYZEncoded2Float(w0 as UInt16, w1 as UInt16, w2 as UInt16) as LCMS2CIEXYZMBS	124
* 2.16.2 kAVG_ SURROUND = 1	125
* 2.16.2 kcmsEmbeddedProfileFalse = 0	125
* 2.16.2 kcmsEmbeddedProfileTrue = 1	125
* 2.16.2 kcmsERROR_ ALREADY_ DEFINED = & h0000000A	125
* 2.16.2 kcmsERROR_ BAD_ SIGNATURE = & h0000000B	125
* 2.16.2 kcmsERROR_ COLORSPACE_ CHECK = 9	125
* 2.16.2 kcmsERROR_ CORRUPTION_ DETECTED = & h0000000C	126
* 2.16.2 kcmsERROR_ FILE = 1	126
* 2.16.2 kcmsERROR_ INTERNAL = 3	126
* 2.16.2 kcmsERROR_ NOT_ SUITABLE = & h0000000D	126
* 2.16.2 kcmsERROR_ NULL = 4	126
* 2.16.2 kcmsERROR_ RANGE = 2	126
* 2.16.2 kcmsERROR_ READ = 5	127
* 2.16.2 kcmsERROR_ SEEK = 6	127
* 2.16.2 kcmsERROR_ UNDEFINED = 0	127
* 2.16.2 kcmsERROR_ UNKNOWN_ EXTENSION = 8	127
* 2.16.2 kcmsERROR_ WRITE = 7	127
* 2.16.2 kcmsFLAGS_ 8BITS_ DEVICELINK = 8	127
* 2.16.2 kcmsFLAGS_ BLACKPOINTCOMPENSATION = & h00002000	128
* 2.16.2 kcmsFLAGS_ CLUT_ POST_ LINEARIZATION = 1	128
* 2.16.2 kcmsFLAGS_ CLUT_ PRE_ LINEARIZATION = & h00000010	128
* 2.16.2 kcmsFLAGS_ FORCE_ CLUT = 2	128
* 2.16.2 kcmsFLAGS_ GAMUTCHECK = & h00001000	128
* 2.16.2 kcmsFLAGS_ GUESSDEVICECLASS = & h00000020	129
* 2.16.2 kcmsFLAGS_ HIGHRESPRECALC = & h00000400	129
* 2.16.2 kcmsFLAGS_ KEEP_ SEQUENCE = & h00000080	129

* 2.16.2 kcmsFLAGS_ LOWRESPRECALC = & h00000800	129
* 2.16.2 kcmsFLAGS_ NOCACHE = & h00000040	129
* 2.16.2 kcmsFLAGS_ NODEFAULTRESOURCEDEF = & h01000000	130
* 2.16.2 kcmsFLAGS_ NOOPTIMIZE = & h00000100	130
* 2.16.2 kcmsFLAGS_ NOWHITEONWHITEFIXUP = 4	130
* 2.16.2 kcmsFLAGS_ NULLTRANSFORM = & h00000200	130
* 2.16.2 kcmsFLAGS_ SOFTPROOFING = & h00004000	130
* 2.16.2 kcmsFREQUENCY_ UNITS_ LINES_ CM = 0	131
* 2.16.2 kcmsFREQUENCY_ UNITS_ LINES_ INCH = 2	131
* 2.16.2 kcmsGlossy = 0	131
* 2.16.2 kcmsILLUMINANT_ TYPE_ A = 6	131
* 2.16.2 kcmsILLUMINANT_ TYPE_ D50 = 1	131
* 2.16.2 kcmsILLUMINANT_ TYPE_ D55 = 5	131
* 2.16.2 kcmsILLUMINANT_ TYPE_ D65 = 2	132
* 2.16.2 kcmsILLUMINANT_ TYPE_ D93 = 3	132
* 2.16.2 kcmsILLUMINANT_ TYPE_ E = 7	132
* 2.16.2 kcmsILLUMINANT_ TYPE_ F2 = 4	132
* 2.16.2 kcmsILLUMINANT_ TYPE_ F8 = 8	132
* 2.16.2 kcmsILLUMINANT_ TYPE_ UNKNOWN = 0	133
* 2.16.2 kcmsMagicNumber = & h61637370	133
* 2.16.2 kcmsMatte = 2	133
* 2.16.2 kcmsMAXCHANNELS = & h00000010	133
* 2.16.2 kcmsPRINTER_ DEFAULT_ SCREENS = 1	133
* 2.16.2 kcmsReflective = 0	133
* 2.16.2 kcmsSig10colorData = & h41434C52	134
* 2.16.2 kcmsSig11colorData = & h42434C52	134
* 2.16.2 kcmsSig12colorData = & h43434C52	134
* 2.16.2 kcmsSig13colorData = & h44434C52	134
* 2.16.2 kcmsSig14colorData = & h45434C52	134
* 2.16.2 kcmsSig15colorData = & h46434C52	134
* 2.16.2 kcmsSig1colorData = & h31434C52	135
* 2.16.2 kcmsSig2colorData = & h32434C52	135
* 2.16.2 kcmsSig3colorData = & h33434C52	135
* 2.16.2 kcmsSig4colorData = & h34434C52	135
* 2.16.2 kcmsSig5colorData = & h35434C52	135
* 2.16.2 kcmsSig6colorData = & h36434C52	135
* 2.16.2 kcmsSig7colorData = & h37434C52	136
* 2.16.2 kcmsSig8colorData = & h38434C52	136
* 2.16.2 kcmsSig9colorData = & h39434C52	136
* 2.16.2 kcmsSigAbstractClass = & h61627374	136
* 2.16.2 kcmsSigAMDisplay = & h414D4420	136
* 2.16.2 kcmsSigAToB0Tag = & h41324230	136

* 2.16.2 kcmsSigAToB1Tag = & h41324231	137
* 2.16.2 kcmsSigAToB2Tag = & h41324232	137
* 2.16.2 kcmsSigBAcsElemType = & h62414353	137
* 2.16.2 kcmsSigBlueColorantTag = & h6258595A	137
* 2.16.2 kcmsSigBlueMatrixColumnTag = & h6258595A	137
* 2.16.2 kcmsSigBlueTRCTag = & h62545243	137
* 2.16.2 kcmsSigBToA0Tag = & h42324130	138
* 2.16.2 kcmsSigBToA1Tag = & h42324131	138
* 2.16.2 kcmsSigBToA2Tag = & h42324132	138
* 2.16.2 kcmsSigBToD0Tag = & h42324430	138
* 2.16.2 kcmsSigBToD1Tag = & h42324431	138
* 2.16.2 kcmsSigBToD2Tag = & h42324432	138
* 2.16.2 kcmsSigBToD3Tag = & h42324433	139
* 2.16.2 kcmsSigCalibrationDateTimeTag = & h63616C74	139
* 2.16.2 kcmsSigCharTargetTag = & h74617267	139
* 2.16.2 kcmsSigChromaticAdaptationTag = & h63686164	139
* 2.16.2 kcmsSigChromaticityTag = & h6368726D	139
* 2.16.2 kcmsSigChromaticityType = & h6368726D	139
* 2.16.2 kcmsSigCLutElemType = & h636C7574	140
* 2.16.2 kcmsSigCmyData = & h434D5920	140
* 2.16.2 kcmsSigCmykData = & h434D594B	140
* 2.16.2 kcmsSigColorantOrderTag = & h636C726F	140
* 2.16.2 kcmsSigColorantOrderType = & h636C726F	140
* 2.16.2 kcmsSigColorantTableOutTag = & h636C6F74	140
* 2.16.2 kcmsSigColorantTableTag = & h636C7274	141
* 2.16.2 kcmsSigColorantTableType = & h636C7274	141
* 2.16.2 kcmsSigColorimetricIntentImageStateTag = & h63696973	141
* 2.16.2 kcmsSigColorSpaceClass = & h73706163	141
* 2.16.2 kcmsSigCopyrightTag = & h63707274	141
* 2.16.2 kcmsSigCrdInfoTag = & h63726469	141
* 2.16.2 kcmsSigCrdInfoType = & h63726469	142
* 2.16.2 kcmsSigCRTDisplay = & h43525420	142
* 2.16.2 kcmsSigCurveSetElemType = & h63767374	143
* 2.16.2 kcmsSigCurveType = & h63757276	143
* 2.16.2 kcmsSigDataTag = & h64617461	143
* 2.16.2 kcmsSigDataType = & h64617461	143
* 2.16.2 kcmsSigDateTimeTag = & h6474696D	143
* 2.16.2 kcmsSigDateTimeType = & h6474696D	143
* 2.16.2 kcmsSigDeviceMfgDescTag = & h646D6E64	144
* 2.16.2 kcmsSigDeviceModelDescTag = & h646D6464	144
* 2.16.2 kcmsSigDeviceSettingsTag = & h64657673	144
* 2.16.2 kcmsSigDeviceSettingsType = & h64657673	144

* 2.16.2 kcmsSigDictType = & h64696374	144
* 2.16.2 kcmsSigDigitalCamera = & h6463616D	144
* 2.16.2 kcmsSigDigitalCinemaProjector = & h64636A70	145
* 2.16.2 kcmsSigDigitalMotionPictureCamera = & h646D7063	145
* 2.16.2 kcmsSigDisplayClass = & h6D6E7472	145
* 2.16.2 kcmsSigDN = & h444E2020	145
* 2.16.2 kcmsSigDNN = & h444E4E20	145
* 2.16.2 kcmsSigDNNP = & h444E4E50	145
* 2.16.2 kcmsSigDNP = & h444E2050	146
* 2.16.2 kcmsSigDToB0Tag = & h44324230	146
* 2.16.2 kcmsSigDToB1Tag = & h44324231	146
* 2.16.2 kcmsSigDToB2Tag = & h44324232	146
* 2.16.2 kcmsSigDToB3Tag = & h44324233	146
* 2.16.2 kcmsSigDyeSublimationPrinter = & h64737562	147
* 2.16.2 kcmsSigEAcElemType = & h65414353	147
* 2.16.2 kcmsSigElectrophotographicPrinter = & h6570686F	147
* 2.16.2 kcmsSigElectrostaticPrinter = & h65737461	147
* 2.16.2 kcmsSigFilmScanner = & h6673636E	147
* 2.16.2 kcmsSigFilmWriter = & h6670726E	147
* 2.16.2 kcmsSigFlexography = & h666C6578	148
* 2.16.2 kcmsSigFocalPlaneColorimetryEstimates = & h66706365	148
* 2.16.2 kcmsSigFormulaCurveSeg = & h70617266	148
* 2.16.2 kcmsSigGamutTag = & h67616D74	148
* 2.16.2 kcmsSigGravure = & h67726176	148
* 2.16.2 kcmsSigGrayData = & h47524159	148
* 2.16.2 kcmsSigGrayTRCTag = & h6B545243	149
* 2.16.2 kcmsSigGreenColorantTag = & h6758595A	149
* 2.16.2 kcmsSigGreenMatrixColumnTag = & h6758595A	149
* 2.16.2 kcmsSigGreenTRCTag = & h67545243	149
* 2.16.2 kcmsSigHlsData = & h484C5320	149
* 2.16.2 kcmsSigHsvData = & h48535620	149
* 2.16.2 kcmsSigIdentityElemType = & h69646E20	150
* 2.16.2 kcmsSigInkJetPrinter = & h696A6574	150
* 2.16.2 kcmsSigInputClass = & h73636E72	150
* 2.16.2 kcmsSigLab2XYZElemType = & h78326C20	150
* 2.16.2 kcmsSigLabData = & h4C616220	150
* 2.16.2 kcmsSigLabV2toV4 = & h32203420	150
* 2.16.2 kcmsSigLabV4toV2 = & h34203220	151
* 2.16.2 kcmsSigLinkClass = & h6C696E6B	151
* 2.16.2 kcmsSigLuminanceTag = & h6C756D69	151
* 2.16.2 kcmsSigLut16Type = & h6D667432	151
* 2.16.2 kcmsSigLut8Type = & h6D667431	151

* 2.16.2 kcmsSigLutAtoBType = & h6D414220	151
* 2.16.2 kcmsSigLutBtoAType = & h6D424120	152
* 2.16.2 kcmsSigLuvData = & h4C757620	152
* 2.16.2 kcmsSigLuvKData = & h4C75764B	152
* 2.16.2 kcmsSigMacintosh = & h4150504C	152
* 2.16.2 kcmsSigMatrixElemType = & h6D617466	152
* 2.16.2 kcmsSigMCH1Data = & h4D434831	152
* 2.16.2 kcmsSigMCH2Data = & h4D434832	153
* 2.16.2 kcmsSigMCH3Data = & h4D434833	153
* 2.16.2 kcmsSigMCH4Data = & h4D434834	153
* 2.16.2 kcmsSigMCH5Data = & h4D434835	153
* 2.16.2 kcmsSigMCH6Data = & h4D434836	153
* 2.16.2 kcmsSigMCH7Data = & h4D434837	153
* 2.16.2 kcmsSigMCH8Data = & h4D434838	154
* 2.16.2 kcmsSigMCH9Data = & h4D434839	154
* 2.16.2 kcmsSigMCHAData = & h4D434841	154
* 2.16.2 kcmsSigMCHBData = & h4D434842	154
* 2.16.2 kcmsSigMCHCData = & h4D434843	154
* 2.16.2 kcmsSigMCHDData = & h4D434844	154
* 2.16.2 kcmsSigMCHEData = & h4D434845	155
* 2.16.2 kcmsSigMCHFData = & h4D434846	155
* 2.16.2 kcmsSigMeasurementTag = & h6D656173	155
* 2.16.2 kcmsSigMeasurementType = & h6D656173	155
* 2.16.2 kcmsSigMediaBlackPointTag = & h626B7074	155
* 2.16.2 kcmsSigMediaWhitePointTag = & h77747074	155
* 2.16.2 kcmsSigMetaTag = & h6D657461	156
* 2.16.2 kcmsSigMicrosoft = & h4D534654	156
* 2.16.2 kcmsSigMotionPictureFilmRecorder = & h6D706672	156
* 2.16.2 kcmsSigMotionPictureFilmScanner = & h6D706673	156
* 2.16.2 kcmsSigMultiLocalizedUnicodeType = & h6D6C7563	156
* 2.16.2 kcmsSigMultiProcessElementType = & h6D706574	156
* 2.16.2 kcmsSigNamedColor2Tag = & h6E636C32	157
* 2.16.2 kcmsSigNamedColor2Type = & h6E636C32	157
* 2.16.2 kcmsSigNamedColorClass = & h6E6D636C	157
* 2.16.2 kcmsSigNamedColorElemType = & h6E636C20	157
* 2.16.2 kcmsSigNamedColorTag = & h6E636F6C	157
* 2.16.2 kcmsSigNamedColorType = & h6E636F6C	157
* 2.16.2 kcmsSigNamedData = & h6E6D636C	158
* 2.16.2 kcmsSigOffsetLithography = & h6F666673	158
* 2.16.2 kcmsSigOutputClass = & h70727472	158
* 2.16.2 kcmsSigOutputResponseTag = & h72657370	158
* 2.16.2 kcmsSigParametricCurveType = & h70617261	158

* 2.16.2 kcmsSigPerceptualReferenceMediumGamut = & h70726D67	158
* 2.16.2 kcmsSigPerceptualRenderingIntentGamutTag = & h72696730	159
* 2.16.2 kcmsSigPhotoCD = & h4B504344	159
* 2.16.2 kcmsSigPhotographicPaperPrinter = & h7270686F	159
* 2.16.2 kcmsSigPhotoImageSetter = & h696D6773	159
* 2.16.2 kcmsSigPMDisplay = & h504D4420	159
* 2.16.2 kcmsSigPreview0Tag = & h70726530	159
* 2.16.2 kcmsSigPreview1Tag = & h70726531	160
* 2.16.2 kcmsSigPreview2Tag = & h70726532	160
* 2.16.2 kcmsSigProfileDescriptionTag = & h64657363	160
* 2.16.2 kcmsSigProfileSequenceDescTag = & h70736571	160
* 2.16.2 kcmsSigProfileSequenceDescType = & h70736571	160
* 2.16.2 kcmsSigProfileSequenceIdTag = & h70736964	160
* 2.16.2 kcmsSigProfileSequenceIdType = & h70736964	161
* 2.16.2 kcmsSigProjectionTelevision = & h706A7476	161
* 2.16.2 kcmsSigPs2CRD0Tag = & h70736430	161
* 2.16.2 kcmsSigPs2CRD1Tag = & h70736431	161
* 2.16.2 kcmsSigPs2CRD2Tag = & h70736432	161
* 2.16.2 kcmsSigPs2CRD3Tag = & h70736433	161
* 2.16.2 kcmsSigPs2CSATag = & h70733273	162
* 2.16.2 kcmsSigPs2RenderingIntentTag = & h70733269	162
* 2.16.2 kcmsSigRedColorantTag = & h7258595A	162
* 2.16.2 kcmsSigRedMatrixColumnTag = & h7258595A	162
* 2.16.2 kcmsSigRedTRCTag = & h72545243	162
* 2.16.2 kcmsSigReflectionHardcopyOriginalColorimetry = & h72686F63	162
* 2.16.2 kcmsSigReflectionPrintOutputColorimetry = & h72706F63	163
* 2.16.2 kcmsSigReflectiveScanner = & h7273636E	163
* 2.16.2 kcmsSigResponseCurveSet16Type = & h72637332	163
* 2.16.2 kcmsSigRgbData = & h52474220	163
* 2.16.2 kcmsSigS15Fixed16ArrayType = & h73663332	163
* 2.16.2 kcmsSigSampledCurveSeg = & h73616D66	163
* 2.16.2 kcmsSigSaturationRenderingIntentGamutTag = & h72696732	164
* 2.16.2 kcmsSigSceneAppearanceEstimates = & h73617065	164
* 2.16.2 kcmsSigSceneColorimetryEstimates = & h73636F65	164
* 2.16.2 kcmsSigScreeningDescTag = & h73637264	164
* 2.16.2 kcmsSigScreeningTag = & h7363726E	164
* 2.16.2 kcmsSigScreeningType = & h7363726E	164
* 2.16.2 kcmsSigSegmentedCurve = & h63757266	165
* 2.16.2 kcmsSigSGI = & h53474920	165
* 2.16.2 kcmsSigSignatureType = & h73696720	165
* 2.16.2 kcmsSigSilkscreen = & h73696C6B	165
* 2.16.2 kcmsSigSolaris = & h53554E57	165

* 2.16.2 kcmsSigStatusA = & h53746141	165
* 2.16.2 kcmsSigStatusE = & h53746145	166
* 2.16.2 kcmsSigStatusI = & h53746149	166
* 2.16.2 kcmsSigStatusM = & h5374614D	166
* 2.16.2 kcmsSigStatusT = & h53746154	166
* 2.16.2 kcmsSigTaligent = & h54474E54	166
* 2.16.2 kcmsSigTechnologyTag = & h74656368	167
* 2.16.2 kcmsSigTextDescriptionType = & h64657363	167
* 2.16.2 kcmsSigTextType = & h74657874	167
* 2.16.2 kcmsSigThermalWaxPrinter = & h74776178	167
* 2.16.2 kcmsSigU16Fixed16ArrayType = & h75663332	167
* 2.16.2 kcmsSigUcrBgTag = & h62666420	168
* 2.16.2 kcmsSigUcrBgType = & h62666420	168
* 2.16.2 kcmsSigUInt16ArrayType = & h75693136	168
* 2.16.2 kcmsSigUInt32ArrayType = & h75693332	168
* 2.16.2 kcmsSigUInt64ArrayType = & h75693634	168
* 2.16.2 kcmsSigUInt8ArrayType = & h75693038	168
* 2.16.2 kcmsSigUnices = & h2A6E6978	169
* 2.16.2 kcmsSigVcgtTag = & h76636774	169
* 2.16.2 kcmsSigVcgtType = & h76636774	169
* 2.16.2 kcmsSigVideoCamera = & h76696463	169
* 2.16.2 kcmsSigVideoMonitor = & h7669646D	169
* 2.16.2 kcmsSigViewingCondDescTag = & h76756564	169
* 2.16.2 kcmsSigViewingConditionsTag = & h76696577	170
* 2.16.2 kcmsSigViewingConditionsType = & h76696577	170
* 2.16.2 kcmsSigXYZ2LabElemType = & h6C327820	170
* 2.16.2 kcmsSigXYZData = & h58595A20	170
* 2.16.2 kcmsSigXYZType = & h58595A20	170
* 2.16.2 kcmsSigYCbCrData = & h59436272	170
* 2.16.2 kcmsSigYxyData = & h59787920	171
* 2.16.2 kcmsSPOT_ CROSS = 7	171
* 2.16.2 kcmsSPOT_ DIAMOND = 3	171
* 2.16.2 kcmsSPOT_ ELLIPSE = 4	171
* 2.16.2 kcmsSPOT_ LINE = 5	171
* 2.16.2 kcmsSPOT_ PRINTER_ DEFAULT = 1	171
* 2.16.2 kcmsSPOT_ ROUND = 2	172
* 2.16.2 kcmsSPOT_ SQUARE = 6	172
* 2.16.2 kcmsSPOT_ UNKNOWN = 0	172
* 2.16.2 kcmsTransparency = 1	172
* 2.16.2 kcmsUseAnywhere = 0	172
* 2.16.2 kcmsUseWithEmbeddedDataOnly = 2	172
* 2.16.2 kCUTSHEET_ SURROUND = 4	173

* 2.16.2 kDARK_ SURROUND = 3	173
* 2.16.2 kDIM_ SURROUND = 2	173
* 2.16.2 kD_ CALCULATE = -1	173
* 2.16.2 kINTENT_ ABSOLUTE_ COLORIMETRIC = 3	173
* 2.16.2 kINTENT_ PERCEPTUAL = 0	173
* 2.16.2 kINTENT_ PRESERVE_ K_ ONLY_ PERCEPTUAL = & h0000000A	174
* 2.16.2 kINTENT_ PRESERVE_ K_ ONLY_ RELATIVE_ COLORIMETRIC = & h0000000B	174
* 2.16.2 kINTENT_ PRESERVE_ K_ ONLY_ SATURATION = & h0000000C	174
* 2.16.2 kINTENT_ PRESERVE_ K_ PLANE_ PERCEPTUAL = & h0000000D	174
* 2.16.2 kINTENT_ PRESERVE_ K_ PLANE_ RELATIVE_ COLORIMETRIC = & h0000000E	174
* 2.16.2 kINTENT_ PRESERVE_ K_ PLANE_ SATURATION = & h0000000F	175
* 2.16.2 kINTENT_ RELATIVE_ COLORIMETRIC = 1	175
* 2.16.2 kINTENT_ SATURATION = 2	175
* 2.16.2 klcmsSignature = & h6C636D73	175
* 2.16.2 kLCMS_ USED_ AS_ INPUT = 0	175
* 2.16.2 kLCMS_ USED_ AS_ OUTPUT = 1	175
* 2.16.2 kLCMS_ USED_ AS_ PROOF = 2	176
* 2.16.2 kPT_ ANY = 0	176
* 2.16.2 kPT_ CMY = 5	176
* 2.16.2 kPT_ CMYK = 6	176
* 2.16.2 kPT_ GRAY = 3	176
* 2.16.2 kPT_ HLS = & h0000000D	177
* 2.16.2 kPT_ HSV = & h0000000C	177
* 2.16.2 kPT_ Lab = & h0000000A	177
* 2.16.2 kPT_ LabV2 = & h0000001E	177
* 2.16.2 kPT_ MCH1 = & h0000000F	177
* 2.16.2 kPT_ MCH10 = & h00000018	178
* 2.16.2 kPT_ MCH11 = & h00000019	178
* 2.16.2 kPT_ MCH12 = & h0000001A	178
* 2.16.2 kPT_ MCH13 = & h0000001B	178
* 2.16.2 kPT_ MCH14 = & h0000001C	178
* 2.16.2 kPT_ MCH15 = & h0000001D	179
* 2.16.2 kPT_ MCH2 = & h00000010	179
* 2.16.2 kPT_ MCH3 = & h00000011	179
* 2.16.2 kPT_ MCH4 = & h00000012	179
* 2.16.2 kPT_ MCH5 = & h00000013	179
* 2.16.2 kPT_ MCH6 = & h00000014	180
* 2.16.2 kPT_ MCH7 = & h00000015	180
* 2.16.2 kPT_ MCH8 = & h00000016	180
* 2.16.2 kPT_ MCH9 = & h00000017	180

* 2.16.2 kPT_ RGB = 4	180
* 2.16.2 kPT_ XYZ = 9	181
* 2.16.2 kPT_ YCbCr = 7	181
* 2.16.2 kPT_ YUV = 8	181
* 2.16.2 kPT_ YUVK = & h0000000B	181
* 2.16.2 kPT_ Yxy = & h0000000E	181
* 2.16.2 kTYPE_ ABGR_ 16 = & h0004049A	182
* 2.16.2 kTYPE_ ABGR_ 16_ PLANAR = & h0004149A	182
* 2.16.2 kTYPE_ ABGR_ 16_ SE = & h00040C9A	182
* 2.16.2 kTYPE_ ABGR_ 8 = & h00040499	182
* 2.16.2 kTYPE_ ALabV2_ 8 = & h001E0499	182
* 2.16.2 kTYPE_ ALab_ 8 = & h000A0499	182
* 2.16.2 kTYPE_ ARGB_ 16 = & h0004409A	183
* 2.16.2 kTYPE_ ARGB_ 8 = & h00044099	183
* 2.16.2 kTYPE_ BGRA_ 16 = & h0004449A	183
* 2.16.2 kTYPE_ BGRA_ 16_ SE = & h0004489A	183
* 2.16.2 kTYPE_ BGRA_ 8 = & h00044499	183
* 2.16.2 kTYPE_ BGR_ 16 = & h0004041A	183
* 2.16.2 kTYPE_ BGR_ 16_ PLANAR = & h0004141A	184
* 2.16.2 kTYPE_ BGR_ 16_ SE = & h00040C1A	184
* 2.16.2 kTYPE_ BGR_ 8 = & h00040419	184
* 2.16.2 kTYPE_ BGR_ 8_ PLANAR = & h00041419	184
* 2.16.2 kTYPE_ CMYK10_ 16 = & h00180052	184
* 2.16.2 kTYPE_ CMYK10_ 16_ SE = & h00180852	184
* 2.16.2 kTYPE_ CMYK10_ 8 = & h00180051	185
* 2.16.2 kTYPE_ CMYK11_ 16 = & h0019005A	185
* 2.16.2 kTYPE_ CMYK11_ 16_ SE = & h0019085A	185
* 2.16.2 kTYPE_ CMYK11_ 8 = & h00190059	185
* 2.16.2 kTYPE_ CMYK12_ 16 = & h001A0062	185
* 2.16.2 kTYPE_ CMYK12_ 16_ SE = & h001A0862	185
* 2.16.2 kTYPE_ CMYK12_ 8 = & h001A0061	186
* 2.16.2 kTYPE_ CMYK5_ 16 = & h0013002A	186
* 2.16.2 kTYPE_ CMYK5_ 16_ SE = & h0013082A	186
* 2.16.2 kTYPE_ CMYK5_ 8 = & h00130029	186
* 2.16.2 kTYPE_ CMYK6_ 16 = & h00140032	186
* 2.16.2 kTYPE_ CMYK6_ 16_ PLANAR = & h00141032	186
* 2.16.2 kTYPE_ CMYK6_ 16_ SE = & h00140832	187
* 2.16.2 kTYPE_ CMYK6_ 8 = & h00140031	187
* 2.16.2 kTYPE_ CMYK6_ 8_ PLANAR = & h00141031	187
* 2.16.2 kTYPE_ CMYK7_ 16 = & h0015003A	187
* 2.16.2 kTYPE_ CMYK7_ 16_ SE = & h0015083A	187
* 2.16.2 kTYPE_ CMYK7_ 8 = & h00150039	187

* 2.16.2 kTYPE_ CMYK8_ 16 = & h00160042	188
* 2.16.2 kTYPE_ CMYK8_ 16_ SE = & h00160842	188
* 2.16.2 kTYPE_ CMYK8_ 8 = & h00160041	188
* 2.16.2 kTYPE_ CMYK9_ 16 = & h0017004A	188
* 2.16.2 kTYPE_ CMYK9_ 16_ SE = & h0017084A	188
* 2.16.2 kTYPE_ CMYK9_ 8 = & h00170049	188
* 2.16.2 kTYPE_ CMYKA_ 8 = & h000600A1	189
* 2.16.2 kTYPE_ CMYK_ 16 = & h00060022	189
* 2.16.2 kTYPE_ CMYK_ 16_ PLANAR = & h00061022	189
* 2.16.2 kTYPE_ CMYK_ 16_ REV = & h00062022	189
* 2.16.2 kTYPE_ CMYK_ 16_ SE = & h00060822	189
* 2.16.2 kTYPE_ CMYK_ 8 = & h00060021	189
* 2.16.2 kTYPE_ CMYK_ 8_ PLANAR = & h00061021	190
* 2.16.2 kTYPE_ CMYK_ 8_ REV = & h00062021	190
* 2.16.2 kTYPE_ CMYK_ DBL = & h00460020	190
* 2.16.2 kTYPE_ CMYK_ FLT = & h00460024	190
* 2.16.2 kTYPE_ CMY_ 16 = & h0005001A	190
* 2.16.2 kTYPE_ CMY_ 16_ PLANAR = & h0005101A	190
* 2.16.2 kTYPE_ CMY_ 16_ SE = & h0005081A	191
* 2.16.2 kTYPE_ CMY_ 8 = & h00050019	191
* 2.16.2 kTYPE_ CMY_ 8_ PLANAR = & h00051019	191
* 2.16.2 kTYPE_ GRAYA_ 16 = & h0003008A	191
* 2.16.2 kTYPE_ GRAYA_ 16_ PLANAR = & h0003108A	191
* 2.16.2 kTYPE_ GRAYA_ 16_ SE = & h0003088A	191
* 2.16.2 kTYPE_ GRAYA_ 8 = & h00030089	192
* 2.16.2 kTYPE_ GRAYA_ 8_ PLANAR = & h00031089	192
* 2.16.2 kTYPE_ GRAY_ 16 = & h0003000A	192
* 2.16.2 kTYPE_ GRAY_ 16_ REV = & h0003200A	192
* 2.16.2 kTYPE_ GRAY_ 16_ SE = & h0003080A	192
* 2.16.2 kTYPE_ GRAY_ 8 = & h00030009	192
* 2.16.2 kTYPE_ GRAY_ 8_ REV = & h00032009	193
* 2.16.2 kTYPE_ GRAY_ DBL = & h00430008	193
* 2.16.2 kTYPE_ GRAY_ FLT = & h0043000C	193
* 2.16.2 kTYPE_ HLS_ 16 = & h000D001A	193
* 2.16.2 kTYPE_ HLS_ 16_ PLANAR = & h000D101A	193
* 2.16.2 kTYPE_ HLS_ 16_ SE = & h000D081A	193
* 2.16.2 kTYPE_ HLS_ 8 = & h000D0019	194
* 2.16.2 kTYPE_ HLS_ 8_ PLANAR = & h000D1019	194
* 2.16.2 kTYPE_ HSV_ 16 = & h000C001A	194
* 2.16.2 kTYPE_ HSV_ 16_ PLANAR = & h000C101A	194
* 2.16.2 kTYPE_ HSV_ 16_ SE = & h000C081A	194
* 2.16.2 kTYPE_ HSV_ 8 = & h000C0019	194

* 2.16.2 kTYPE_ HSV_ 8_ PLANAR = & h000C1019	195
* 2.16.2 kTYPE_ KCMY_ 16 = & h00064022	195
* 2.16.2 kTYPE_ KCMY_ 16_ REV = & h00066022	195
* 2.16.2 kTYPE_ KCMY_ 16_ SE = & h00064822	195
* 2.16.2 kTYPE_ KCMY_ 8 = & h00064021	195
* 2.16.2 kTYPE_ KCMY_ 8_ REV = & h00066021	195
* 2.16.2 kTYPE_ KYMC10_ 16 = & h00180452	196
* 2.16.2 kTYPE_ KYMC10_ 16_ SE = & h00180C52	196
* 2.16.2 kTYPE_ KYMC10_ 8 = & h00180451	196
* 2.16.2 kTYPE_ KYMC11_ 16 = & h0019045A	196
* 2.16.2 kTYPE_ KYMC11_ 16_ SE = & h00190C5A	196
* 2.16.2 kTYPE_ KYMC11_ 8 = & h00190459	196
* 2.16.2 kTYPE_ KYMC12_ 16 = & h001A0462	197
* 2.16.2 kTYPE_ KYMC12_ 16_ SE = & h001A0C62	197
* 2.16.2 kTYPE_ KYMC12_ 8 = & h001A0461	197
* 2.16.2 kTYPE_ KYMC5_ 16 = & h0013042A	197
* 2.16.2 kTYPE_ KYMC5_ 16_ SE = & h00130C2A	197
* 2.16.2 kTYPE_ KYMC5_ 8 = & h00130429	197
* 2.16.2 kTYPE_ KYMC7_ 16 = & h0015043A	198
* 2.16.2 kTYPE_ KYMC7_ 16_ SE = & h00150C3A	198
* 2.16.2 kTYPE_ KYMC7_ 8 = & h00150439	198
* 2.16.2 kTYPE_ KYMC8_ 16 = & h00160442	198
* 2.16.2 kTYPE_ KYMC8_ 16_ SE = & h00160C42	198
* 2.16.2 kTYPE_ KYMC8_ 8 = & h00160441	198
* 2.16.2 kTYPE_ KYMC9_ 16 = & h0017044A	199
* 2.16.2 kTYPE_ KYMC9_ 16_ SE = & h00170C4A	199
* 2.16.2 kTYPE_ KYMC9_ 8 = & h00170449	199
* 2.16.2 kTYPE_ KYMC_ 16 = & h00060422	199
* 2.16.2 kTYPE_ KYMC_ 16_ SE = & h00060C22	199
* 2.16.2 kTYPE_ KYMC_ 8 = & h00060421	199
* 2.16.2 kTYPE_ LabA_ FLT = & h004A009C	200
* 2.16.2 kTYPE_ LabV2_ 16 = & h001E001A	200
* 2.16.2 kTYPE_ LabV2_ 8 = & h001E0019	200
* 2.16.2 kTYPE_ Lab_ 16 = & h000A001A	200
* 2.16.2 kTYPE_ Lab_ 8 = & h000A0019	200
* 2.16.2 kTYPE_ Lab_ DBL = & h004A0018	200
* 2.16.2 kTYPE_ Lab_ FLT = & h004A001C	201
* 2.16.2 kTYPE_ NAMED_ COLOR_ INDEX = & h0000000A	201
* 2.16.2 kTYPE_ RGBA_ 16 = & h0004009A	201
* 2.16.2 kTYPE_ RGBA_ 16_ PLANAR = & h0004109A	201
* 2.16.2 kTYPE_ RGBA_ 16_ SE = & h0004089A	201
* 2.16.2 kTYPE_ RGBA_ 8 = & h00040099	201

* 2.16.2 kTYPE_ RGBA_ 8_ PLANAR = & h00041099	202
* 2.16.2 kTYPE_ RGBA_ FLT = & h0044009C	202
* 2.16.2 kTYPE_ RGB_ 16 = & h0004001A	202
* 2.16.2 kTYPE_ RGB_ 16_ PLANAR = & h0004101A	202
* 2.16.2 kTYPE_ RGB_ 16_ SE = & h0004081A	202
* 2.16.2 kTYPE_ RGB_ 8 = & h00040019	202
* 2.16.2 kTYPE_ RGB_ 8_ PLANAR = & h00041019	203
* 2.16.2 kTYPE_ RGB_ DBL = & h00440018	203
* 2.16.2 kTYPE_ RGB_ FLT = & h0044001C	203
* 2.16.2 kTYPE_ XYZA_ FLT = & h0049009C	203
* 2.16.2 kTYPE_ XYZ_ 16 = & h0009001A	203
* 2.16.2 kTYPE_ XYZ_ DBL = & h00490018	203
* 2.16.2 kTYPE_ XYZ_ FLT = & h0049001C	204
* 2.16.2 kTYPE_ YCbCr_ 16 = & h0007001A	204
* 2.16.2 kTYPE_ YCbCr_ 16_ PLANAR = & h0007101A	204
* 2.16.2 kTYPE_ YCbCr_ 16_ SE = & h0007081A	204
* 2.16.2 kTYPE_ YCbCr_ 8 = & h00070019	204
* 2.16.2 kTYPE_ YCbCr_ 8_ PLANAR = & h00071019	205
* 2.16.2 kTYPE_ YUVK_ 16 = & h00062022	205
* 2.16.2 kTYPE_ YUVK_ 8 = & h00062021	205
* 2.16.2 kTYPE_ YUV_ 16 = & h0008001A	205
* 2.16.2 kTYPE_ YUV_ 16_ PLANAR = & h0008101A	205
* 2.16.2 kTYPE_ YUV_ 16_ SE = & h0008081A	205
* 2.16.2 kTYPE_ YUV_ 8 = & h00080019	206
* 2.16.2 kTYPE_ YUV_ 8_ PLANAR = & h00081019	206
* 2.16.2 kTYPE_ Yxy_ 16 = & h000E001A	206
– 2.1 class LCMS2NamedColorListMBS	29
* 2.1.1 Append(name as string, PCS() as integer, Colorant() as integer) as Boolean	29
* 2.1.1 Colorant(nColor as UInt32) as integer()	30
* 2.1.1 ColorIndex(name as string) as integer	30
* 2.1.1 Constructor(context as variant, n as UInt32, ColorantCount as UInt32, Prefix as string = "", Suffix as string = "")	30
* 2.1.1 Count as integer	31
* 2.1.1 Name(nColor as UInt32) as string	31
* 2.1.1 PCS(nColor as UInt32) as integer()	31
* 2.1.1 Prefix(nColor as UInt32) as string	31
* 2.1.1 Suffix(nColor as UInt32) as string	31
* 2.1.2 Handle as Ptr	32
– 2.3 class LCMS2MLUMBS	37
* 2.3.1 Constructor(context as variant, items as UInt32)	38
* 2.3.1 getASCII(LanguageCode as string, CountryCode as string) as string	38

* 2.3.1	getTranslation(LanguageCode as string, CountryCode as string, byref ObtainedLanguageCode as string, byref ObtainedCountryCode as string) as string	39
* 2.3.1	getUnicode(LanguageCode as string, CountryCode as string) as string	39
* 2.3.1	setASCII(LanguageCode as string, CountryCode as string, ASCIIString as string) as Boolean	40
* 2.3.1	setUnicode(LanguageCode as string, CountryCode as string, UnicodeString as string) as Boolean	40
* 2.3.2	Handle as Ptr	41
* 2.3.3	kNoCountry = ""	41
* 2.3.3	kNoLanguage = ""	41
– 2.2	class LCMS2PipelineMBS	32
* 2.2.1	Append(p as LCMS2PipelineMBS) as Boolean	32
* 2.2.1	Constructor(context as variant, InputChannels as UInt32, OutputChannels as UInt32)	32
* 2.2.1	Eval16(In as Ptr, Out as Ptr)	33
* 2.2.1	EvalFloat(In as Ptr, Out as Ptr)	33
* 2.2.1	EvalReverseFloat(Target as Ptr, Result as Ptr, Hint as Ptr)	33
* 2.2.1	FirstStage as LCMS2StageMBS	34
* 2.2.1	InputChannels as UInt32	34
* 2.2.1	InsertStage(when as integer, stage as LCMS2StageMBS)	34
* 2.2.1	LastStage as LCMS2StageMBS	35
* 2.2.1	OutputChannels as UInt32	35
* 2.2.1	SetSaveAs8bitsFlag(save8bit as boolean) as Boolean	35
* 2.2.1	StageCount as UInt32	35
* 2.2.1	Stages as LCMS2StageMBS()	36
* 2.2.1	UnlinkStage(when as integer) as LCMS2StageMBS	36
* 2.2.2	Context as Variant	36
* 2.2.2	Handle as Ptr	36
* 2.2.3	kAtBegin = 0	36
* 2.2.3	kAtEnd = 1	37
– 2.8	class LCMS2StageMBS	52
* 2.8.1	CLutEntries as Integer	52
* 2.8.1	CLutFloatValues as double()	53
* 2.8.1	CLutHasFloatValues as Boolean	53
* 2.8.1	CLutUInt16Values as UInt16()	53
* 2.8.1	CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32) as LCMS2StageMBS	53
* 2.8.1	CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS	54
* 2.8.1	CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, values() as UInt16) as LCMS2StageMBS	55

* 2.8.1 CreateStageWithCLut16bitGranular(Context as variant, clutPoints as Memoryblock, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS	55
* 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32) as LCMS2StageMBS	56
* 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS	56
* 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, values() as double) as LCMS2StageMBS	57
* 2.8.1 CreateStageWithCLutFloatGranular(Context as variant, clutPoints as Memoryblock, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS	57
* 2.8.1 CreateStageWithIdentity(Context as variant, Channels as UInt32) as LCMS2StageMBS	58
* 2.8.1 CreateStageWithMatrix(Context as variant, Rows as UInt32, Cols as UInt32, Matrix as Memoryblock, Offset as Memoryblock = nil) as LCMS2StageMBS	58
* 2.8.1 CreateStageWithToneCurves(Context as variant, Channels() as LCMS2ToneCurveMBS) as LCMS2StageMBS	59
* 2.8.1 Data as Ptr	59
* 2.8.1 InputChannels as UInt32	59
* 2.8.1 MatrixOffsets as double()	60
* 2.8.1 MatrixValues as double()	60
* 2.8.1 NextItem as LCMS2StageMBS	60
* 2.8.1 OutputChannels as UInt32	60
* 2.8.1 SampleCLut16bit(sampler as LCMS2StageSamplerMBS, Flags as integer = 0)	60
* 2.8.1 SampleCLutFloat(sampler as LCMS2StageSamplerMBS, Flags as integer = 0)	61
* 2.8.1 ToneCurves as LCMS2ToneCurveMBS()	61
* 2.8.1 Type as UInt32	62
* 2.8.2 Handle as Ptr	62
* 2.8.3 kSamplerInspect = & h01000000	62
– 2.4 class LCMS2ToneCurveMBS	42
* 2.4.1 BuildGamma(context as variant, gamma as double) as LCMS2ToneCurveMBS	42
* 2.4.1 BuildParametricToneCurve(context as variant, Type as Int32, params() as double) as LCMS2ToneCurveMBS	42
* 2.4.1 BuildSegmentedToneCurve(context as variant, Segments() as LCMS2CurveSegmentMBS) as LCMS2ToneCurveMBS	43
* 2.4.1 BuildTabulatedToneCurve(context as variant, values() as Single) as LCMS2ToneCurveMBS	43
* 2.4.1 BuildTabulatedToneCurve(context as variant, values() as UInt16) as LCMS2ToneCurveMBS	44
* 2.4.1 EstimateGamma(Precision as double = 0.01) as double	44
* 2.4.1 EvalToneCurve16(value as UInt16) as UInt16	45
* 2.4.1 EvalToneCurveFloat(value as Single) as Single	45

* 2.4.1 IsDescending as Boolean	46
* 2.4.1 IsLinear as Boolean	46
* 2.4.1 IsMonotonic as Boolean	46
* 2.4.1 IsMultisegment as Boolean	46
* 2.4.1 JoinToneCurve(context as variant, X as LCMS2ToneCurveMBS, Y as LCMS2ToneCurveMBS, nPoints as UInt32) as LCMS2ToneCurveMBS	47
* 2.4.1 ParametricType as integer	47
* 2.4.1 Reverse as LCMS2ToneCurveMBS	47
* 2.4.1 Reverse(nResultSamples as integer) as LCMS2ToneCurveMBS	47
* 2.4.1 Smooth(lambda as double) as Boolean	48
* 2.4.2 Handle as Ptr	48
– 2.5 class LCMS2StageSamplerMBS	48
* 2.5.1 SamplerFloat(InValues as Ptr, OutValues as Ptr, InputChannels as integer, OutputChannels as integer) as boolean	48
* 2.5.1 SamplerInteger(InValues as Ptr, OutValues as Ptr, InputChannels as integer, OutputChannels as integer) as boolean	49
– 2.6 class LCMS2SequenceDescriptionMBS	49
* 2.6.1 AttributeFlags as UInt64	49
* 2.6.1 Description as LCMS2MLUMBS	50
* 2.6.1 DeviceMfg as UInt32	50
* 2.6.1 DeviceModel as UInt32	50
* 2.6.1 Manufacturer as LCMS2MLUMBS	50
* 2.6.1 Model as LCMS2MLUMBS	50
* 2.6.1 ProfileID as Memoryblock	50
* 2.6.1 Technology as UInt32	51
– 2.7 class LCMS2ScreeningMBS	51
* 2.7.1 Channel(index as integer) as LCMS2ScreeningChannelMBS	51
* 2.7.2 Channels as UInt32	52
* 2.7.2 Flag as UInt32	52
– 2.15 class LCMS2ScreeningChannelMBS	106
* 2.15.1 Clone as LCMS2ScreeningChannelMBS	107
* 2.15.1 Constructor(Frequency as double = 0.0, ScreenAngle as double = 0.0, SpotShape as UInt32 = 0)	107
* 2.15.1 Constructor(other as LCMS2ScreeningChannelMBS)	107
* 2.15.2 Frequency as Double	107
* 2.15.2 ScreenAngle as Double	108
* 2.15.2 SpotShape as UInt32	108
– 2.9 class LCMS2SequenceMBS	62
* 2.9.1 Constructor(context as variant, Count as UInt32)	63
* 2.9.1 Description(index as integer) as LCMS2SequenceDescriptionMBS	63
* 2.9.2 Count as UInt32	63

* 2.9.2 Handle as Ptr	64
– 2.11 class LCMS2UcrBgMBS	66
* 2.11.1 Constructor(Ucr as LCMS2ToneCurveMBS = nil, Bg as LCMS2ToneCurveMBS = nil, Desc as LCMS2MLUMBS = nil)	66
* 2.11.2 Bg as LCMS2ToneCurveMBS	67
* 2.11.2 Desc as LCMS2MLUMBS	67
* 2.11.2 Ucr as LCMS2ToneCurveMBS	67
– 2.10 class LCMS2Vec3MBS	64
* 2.10.1 Clone as LCMS2Vec3MBS	64
* 2.10.1 Constructor(other as LCMS2Vec3MBS)	64
* 2.10.1 Constructor(v1 as double = 0.0, v2 as double = 0.0, v3 as double = 0.0)	65
* 2.10.1 value(index as UInt32) as double	65
* 2.10.2 X as Double	66
* 2.10.2 Y as Double	66
* 2.10.2 Z as Double	66
– 2.13 class LCMS2TransformMBS	70
* 2.13.1 ChangeBuffersFormat(InputFormat as UInt32, OutputFormat as UInt32) as boolean	70
* 2.13.1 CreateMultiprofileTransform(context as variant, Profiles() as LCMS2ProfileMBS, InputFormat as UInt32, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS	70
* 2.13.1 CreateMultiprofileTransform(Profiles() as LCMS2ProfileMBS, InputFormat as UInt32, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS	71
* 2.13.1 CreateProofingTransform(context as variant, InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Proofing as LCMS2ProfileMBS, Intent as UInt32, ProofingIntent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS	72
* 2.13.1 CreateProofingTransform(InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Proofing as LCMS2ProfileMBS, Intent as UInt32, ProofingIntent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS	73
* 2.13.1 CreateTransform(context as variant, InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS	74
* 2.13.1 CreateTransform(InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS	74
* 2.13.1 InputFormat as integer	75
* 2.13.1 NamedColorList as LCMS2NamedColorListMBS	75
* 2.13.1 OutputFormat as integer	75
* 2.13.1 ToDeviceLink(Version as double, Flags as UInt32) as LCMS2ProfileMBS	75
* 2.13.1 Transform(bitmap as LCMS2BitmapMBS) as boolean	76

* 2.13.1 Transform(inBitmap as LCMS2BitmapMBS,outBitmap as LCMS2BitmapMBS) as boolean	
76	
* 2.13.1 Transform(InputBuffer as Ptr, OutputBuffer as Ptr, Size as UInt32) as boolean	76
* 2.13.1 TransformRGB(c as color) as color	77
* 2.13.2 Context as Variant	77
* 2.13.2 Handle as Ptr	77
– 2.12 class LCMS2ViewingConditionsMBS	67
* 2.12.1 Clone as LCMS2ViewingConditionsMBS	68
* 2.12.1 Constructor(other as LCMS2ViewingConditionsMBS)	68
* 2.12.1 Constructor(whitePoint as LCMS2CIEXYZMBS = nil, Yb as double = 0.0, La as double = 0.0, surround as integer = 0, D_ value as double = 0.0)	68
* 2.12.2 D_ value as Double	68
* 2.12.2 La as Double	69
* 2.12.2 Surround as Int32	69
* 2.12.2 whitePoint as LCMS2CIEXYZMBS	69
* 2.12.2 Yb as Double	69
– 2.14 class LCMS2ProfileMBS	78
* 2.14.1 ChannelCount as UInt32	78
* 2.14.1 ColorSpaceType as Integer	78
* 2.14.1 Constructor(context as variant = nil)	78
* 2.14.1 Constructor(file as folderitem, write as boolean = false)	79
* 2.14.1 CreateBCHSWabstractProfile(context as variant, nLUTPoints as integer, Bright as double, Contrast as double, Hue as double, Saturation as double, TempSrc as integer, TempDest as integer) as LCMS2ProfileMBS	79
* 2.14.1 CreateGrayProfile(context as variant, WhitePoint as LCMS2CIExyYMBS, TransferFunction as LCMS2ToneCurveMBS) as LCMS2ProfileMBS	80
* 2.14.1 CreateInkLimitingDeviceLink(context as variant, ColorSpaceSignature as UInt32, Limit as double) as LCMS2ProfileMBS	80
* 2.14.1 CreateLab2Profile(context as variant = nil, point as LCMS2CIExyYMBS = nil) as LCMS2ProfileMBS	80
* 2.14.1 CreateLab4Profile(context as variant = nil, point as LCMS2CIExyYMBS = nil) as LCMS2ProfileMBS	81
* 2.14.1 CreateLinearizationDeviceLink(context as variant, ColorSpaceSignature as UInt32, TransferFunction() as LCMS2ToneCurveMBS) as LCMS2ProfileMBS	81
* 2.14.1 CreateNULLProfile(context as variant = nil) as LCMS2ProfileMBS	82
* 2.14.1 CreateProfilePlaceholder(context as variant = nil) as LCMS2ProfileMBS	82
* 2.14.1 CreateRGBProfile(context as variant, WhitePoint as LCMS2CIExyYMBS, Primaries as LCMS2CIExyYTripleMBS, TransferFunction() as LCMS2ToneCurveMBS) as LCMS2ProfileMBS	82
* 2.14.1 CreateSRGBProfile(context as variant = nil) as LCMS2ProfileMBS	83
* 2.14.1 CreateXYZProfile(context as variant = nil) as LCMS2ProfileMBS	84
* 2.14.1 DetectTAC as double	84
* 2.14.1 DeviceClass as Integer	84

* 2.14.1 FormatterForBitmap(BitCount as integer = 8) as UInt32	85
* 2.14.1 FormatterForColorspace(nBytes as UInt32, IsFloat as boolean = false) as UInt32	85
* 2.14.1 FormatterForPCS(nBytes as UInt32, IsFloat as boolean = false) as UInt32	85
* 2.14.1 GetProfileInfo(Info as integer, LanguageCode as string, CountryCode as string) as string	86
* 2.14.1 HeaderAttributes as UInt64	86
* 2.14.1 HeaderCreationDateTime as LCMS2DateMBS	86
* 2.14.1 HeaderFlags as UInt32	86
* 2.14.1 HeaderManufacturer as UInt32	87
* 2.14.1 HeaderModel as UInt32	87
* 2.14.1 HeaderProfileID as string	87
* 2.14.1 IsCLUT(Intent as UInt32, UsedDirection as UInt32) as boolean	88
* 2.14.1 IsIntentSupported(Intent as UInt32, UsedDirection as UInt32) as boolean	88
* 2.14.1 IsMatrixShaper as Boolean	88
* 2.14.1 IsTag(TagSignature as integer) as Boolean	89
* 2.14.1 LinkTag(sig as integer, dest as integer) as boolean	89
* 2.14.1 MD5computeID as boolean	89
* 2.14.1 Name as string	89
* 2.14.1 OpenProfileFromFile(context as variant, file as folderitem, write as boolean = false) as LCMS2ProfileMBS	90
* 2.14.1 OpenProfileFromFile(file as folderitem, write as boolean = false) as LCMS2ProfileMBS	90
* 2.14.1 OpenProfileFromMemory(context as variant, data as Memoryblock) as LCMS2ProfileMBS	90
* 2.14.1 OpenProfileFromMemory(data as Memoryblock) as LCMS2ProfileMBS	91
* 2.14.1 OpenProfileFromString(context as variant, data as string) as LCMS2ProfileMBS	91
* 2.14.1 OpenProfileFromString(data as string) as LCMS2ProfileMBS	92
* 2.14.1 PCS as Integer	92
* 2.14.1 PostScriptCRD(context as variant, intent as UInt32, flags as UInt32 = 0) as string	92
* 2.14.1 PostScriptCSA(context as variant, intent as UInt32, flags as UInt32 = 0) as string	93
* 2.14.1 ProfileICCversion as Integer	93
* 2.14.1 ProfileVersion as double	93
* 2.14.1 ReadChromaticAdaptation as LCMS2CIEXYZMBS()	93
* 2.14.1 ReadChromaticity as LCMS2CIExyYTripleMBS	94
* 2.14.1 ReadCIEXYZ(tag as integer) as LCMS2CIEXYZMBS	94
* 2.14.1 ReadColorantOrder as Memoryblock	94
* 2.14.1 ReadDate(tag as integer) as LCMS2DateMBS	94
* 2.14.1 ReadICCDData(tag as integer) as LCMS2ICCDDataMBS	95
* 2.14.1 ReadICCMeasurementConditions as LCMS2ICCMeasurementConditionsMBS	95
* 2.14.1 ReadICCViewingConditions as LCMS2ICCViewingConditionsMBS	95
* 2.14.1 ReadMLU(tag as integer) as LCMS2MLUMBS	95
* 2.14.1 ReadNamedColorList(tag as integer) as LCMS2NamedColorListMBS	96

* 2.14.1 ReadPipeline(tag as integer) as LCMS2PipelineMBS	96
* 2.14.1 ReadRawTag(sig as integer) as Memoryblock	96
* 2.14.1 ReadScreening as LCMS2ScreeningMBS	97
* 2.14.1 ReadSequence(tag as integer) as LCMS2SequenceMBS	97
* 2.14.1 ReadSignature(tag as integer) as UInt32	97
* 2.14.1 ReadTag(tag as integer) as variant	97
* 2.14.1 ReadToneCurve(tag as integer) as LCMS2ToneCurveMBS	98
* 2.14.1 ReadUcrBg as LCMS2UcrBgMBS	98
* 2.14.1 RenderingIntent as Integer	98
* 2.14.1 SaveProfileToFile(file as folderitem) as boolean	98
* 2.14.1 SaveProfileToString as Memoryblock	99
* 2.14.1 SaveProfileToString as string	99
* 2.14.1 TagCount as Integer	99
* 2.14.1 TagLinkedTo(sig as integer) as integer	99
* 2.14.1 TagSignature(index as integer) as Integer	100
* 2.14.1 WriteChromaticAdaptation(value as LCMS2Mat3MBS) as boolean	100
* 2.14.1 WriteChromaticAdaptation(values() as LCMS2CIEXYZMBS) as boolean	100
* 2.14.1 WriteChromaticity(o as LCMS2CIExyYTripleMBS) as boolean	100
* 2.14.1 WriteCIEXYZ(tag as integer, o as LCMS2CIEXYZMBS) as boolean	101
* 2.14.1 WriteColorantOrder(data as Memoryblock) as boolean	101
* 2.14.1 WriteDate(tag as integer, o as LCMS2DateMBS) as boolean	101
* 2.14.1 WriteICCData(tag as integer, o as LCMS2ICCDataMBS) as boolean	101
* 2.14.1 WriteICCMeasurementConditions(value as LCMS2ICCMeasurementConditionsMBS) as boolean	102
* 2.14.1 WriteICCViewingConditions(o as LCMS2ICCViewingConditionsMBS) as boolean	102
* 2.14.1 WriteMLU(tag as integer, o as LCMS2MLUMBS) as boolean	102
* 2.14.1 WriteNamedColorList(tag as integer, o as LCMS2NamedColorListMBS) as boolean	102
* 2.14.1 WritePipeline(tag as integer, o as LCMS2PipelineMBS) as boolean	102
* 2.14.1 WriteRawTag(sig as integer, data as Memoryblock) as boolean	103
* 2.14.1 WriteScreening(o as LCMS2ScreeningMBS) as boolean	103
* 2.14.1 WriteSequence(tag as integer, o as LCMS2SequenceMBS) as boolean	103
* 2.14.1 WriteSignature(tag as integer, o as UInt32) as boolean	104
* 2.14.1 WriteToneCurve(tag as integer, o as LCMS2ToneCurveMBS) as boolean	104
* 2.14.1 WriteUcrBg(o as LCMS2UcrBgMBS) as boolean	105
* 2.14.2 Context as variant	105
* 2.14.2 File as Folderitem	105
* 2.14.2 Handle as Ptr	105
* 2.14.3 kInfoCopyright = 3	106
* 2.14.3 kInfoDescription = 0	106
* 2.14.3 kInfoManufacturer = 1	106
* 2.14.3 kInfoModel = 2	106

– 2.17 class LCMS2Mat3MBS	206
* 2.17.1 Clone as LCMS2Mat3MBS	206
* 2.17.1 Constructor	207
* 2.17.1 Constructor(other as LCMS2Mat3MBS)	207
* 2.17.1 Constructor(v0 as LCMS2Vec3MBS, v1 as LCMS2Vec3MBS, v2 as LCMS2Vec3MBS)	207
* 2.17.1 value(index as UInt32) as LCMS2Vec3MBS	207
* 2.17.2 V0 as LCMS2Vec3MBS	208
* 2.17.2 V1 as LCMS2Vec3MBS	208
* 2.17.2 V2 as LCMS2Vec3MBS	208
– 2.28 class LCMS2JChMBS	231
* 2.28.1 Clone as LCMS2JChMBS	232
* 2.28.1 Constructor(J as double=0.0, C as double=0.0, h as double=0.0)	232
* 2.28.1 Constructor(other as LCMS2JChMBS)	232
* 2.28.2 C as Double	232
* 2.28.2 h as Double	232
* 2.28.2 J as Double	233
– 2.19 class LCMS2CIELChMBS	211
* 2.19.1 Clone as LCMS2CIELChMBS	211
* 2.19.1 Constructor(L as double=0.0, C as double=0.0, h as double=0.0)	211
* 2.19.1 Constructor(other as LCMS2CIELChMBS)	211
* 2.19.1 Lab as LCMS2CIELabMBS	212
* 2.19.2 C as Double	212
* 2.19.2 h as Double	212
* 2.19.2 L as Double	212
– 2.18 class LCMS2CIExyYMBS	208
* 2.18.1 Clone as LCMS2CIExyYMBS	209
* 2.18.1 Constructor(other as LCMS2CIExyYMBS)	209
* 2.18.1 Constructor(X as double=0.0, Y as double=0.0, YY as double=0.0)	209
* 2.18.1 TempFromWhitePoint as double	209
* 2.18.1 XYZ as LCMS2CIEXYZMBS	210
* 2.18.2 x as Double	210
* 2.18.2 y as Double	210
* 2.18.2 YY as Double	210
– 2.24 class LCMS2CIELabMBS	222
* 2.24.1 BFDdeltaE(Other as LCMS2CIELabMBS) as double	222
* 2.24.1 CIE2000DeltaE(Other as LCMS2CIELabMBS, Kl as double, Kc as double, Kh as double) as double	222
* 2.24.1 CIE94DeltaE(Other as LCMS2CIELabMBS) as double	223
* 2.24.1 Clone as LCMS2CIELabMBS	223
* 2.24.1 CMCdeltaE(Other as LCMS2CIELabMBS, l as double, c as double) as double	223

* 2.24.1 Constructor(L as double=0.0, a as double=0.0, b as double=0.0)	224
* 2.24.1 Constructor(other as LCMS2CIELabMBS)	224
* 2.24.1 DeltaE(Other as LCMS2CIELabMBS) as double	224
* 2.24.1 LCh as LCMS2CIELChMBS	225
* 2.24.1 XYZ(whitePoint as LCMS2CIEXYZMBS=nil) as LCMS2CIEXYZMBS	225
* 2.24.2 A as Double	225
* 2.24.2 B as Double	225
* 2.24.2 L as Double	226
– 2.21 class LCMS2CIExyYTripleMBS	214
* 2.21.1 Clone as LCMS2CIExyYTripleMBS	214
* 2.21.1 Constructor	215
* 2.21.1 Constructor(other as LCMS2CIExyYTripleMBS)	215
* 2.21.1 Constructor(Red as LCMS2CIExyYMBS, Green as LCMS2CIExyYMBS, Blue as LCMS2CIExyYMBS)	215
* 2.21.2 Blue as LCMS2CIExyYMBS	215
* 2.21.2 Green as LCMS2CIExyYMBS	216
* 2.21.2 Red as LCMS2CIExyYMBS	216
– 2.20 class LCMS2CIEXYZMBS	213
* 2.20.1 Constructor(x as double=0.0, y as double=0.0, z as double=0.0)	213
* 2.20.1 Lab(whitePoint as LCMS2CIEXYZMBS=nil) as LCMS2CIELabMBS	213
* 2.20.1 xyY as LCMS2CIExyYMBS	213
* 2.20.2 x as Double	214
* 2.20.2 y as Double	214
* 2.20.2 z as Double	214
– 2.22 class LCMS2BitmapMBS	216
* 2.22.1 Constructor	216
* 2.22.1 Constructor(p as picture, bits as integer = 8)	217
* 2.22.1 Constructor(width as integer, height as integer, colorspace as integer)	217
* 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer)	217
* 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer, data as memoryblock)	218
* 2.22.1 Invert	218
* 2.22.1 Picture as picture	218
* 2.22.2 Bits as Integer	219
* 2.22.2 ColorSpaceType as Integer	219
* 2.22.2 Data as MemoryBlock	219
* 2.22.2 Height as Integer	220
* 2.22.2 RowBytes as Integer	220
* 2.22.2 Width as Integer	220
– 2.23 class LCMS2CIECAM02MBS	220

* 2.23.1 Constructor(context as variant, VC as LCMS2ViewingConditionsMBS)	221
* 2.23.1 Forward(value as LCMS2CIEXYZMBS) as LCMS2JChMBS	221
* 2.23.1 Reverse(value as LCMS2JChMBS) as LCMS2CIEXYZMBS	221
* 2.23.2 Handle as Ptr	222
– 2.25 class LCMS2ICCDDataMBS	226
* 2.25.1 Data as Memoryblock	226
* 2.25.1 Flags as UInt32	226
* 2.25.1 Size as UInt32	226
– 2.27 class LCMS2GamutBoundaryDescriptionMBS	228
* 2.27.1 AddPoint(Lab as LCMS2CIELabMBS) as Boolean	230
* 2.27.1 CheckPoint(Lab as LCMS2CIELabMBS) as Boolean	230
* 2.27.1 Compute(options as UInt32 = 0) as Boolean	230
* 2.27.1 Constructor(context as variant = nil)	231
* 2.27.2 Context as Variant	231
* 2.27.2 Handle as Ptr	231
– 2.26 class LCMS2ICCMeasurementConditionsMBS	227
* 2.26.1 Constructor(Observer as UInt32 = 0, Backing as LCMS2CIEXYZMBS = nil, Geometry as UInt32 = 0, Flare as double = 0.0, IlluminantType as UInt32 = 0)	227
* 2.26.2 Backing as LCMS2CIEXYZMBS	227
* 2.26.2 Flare as Double	227
* 2.26.2 Geometry as UInt32	228
* 2.26.2 IlluminantType as UInt32	228
* 2.26.2 Observer as UInt32	228
– 2.32 class LCMS2DateMBS	238
* 2.32.1 date as date	238
* 2.32.2 Day as Integer	238
* 2.32.2 Daylight as Integer	238
* 2.32.2 DayOfWeek as Integer	239
* 2.32.2 DayOfYear as Integer	239
* 2.32.2 Hour as Integer	239
* 2.32.2 Minute as Integer	239
* 2.32.2 Month as Integer	239
* 2.32.2 Second as Integer	239
* 2.32.2 Year as Integer	240
– 2.29 class LCMS2ICCViewingConditionsMBS	233
* 2.29.1 Constructor(IlluminantXYZ as LCMS2CIEXYZMBS = nil, Backing as LCMS2CIEXYZMBS = nil, IlluminantType as UInt32 = 0)	233
* 2.29.2 IlluminantType as UInt32	233
* 2.29.2 IlluminantXYZ as LCMS2CIEXYZMBS	234
* 2.29.2 SurroundXYZ as LCMS2CIEXYZMBS	234

– 2.31 class LCMS2CIEXYZTripleMBS	236
* 2.31.1 Clone as LCMS2CIEXYZTripleMBS	236
* 2.31.1 Constructor	236
* 2.31.1 Constructor(other as LCMS2CIEXYZTripleMBS)	237
* 2.31.1 Constructor(Red as LCMS2CIEXYZMBS, Green as LCMS2CIEXYZMBS, Blue as LCMS2CIEXYZMBS)	237
* 2.31.2 Blue as LCMS2CIEXYZMBS	237
* 2.31.2 Green as LCMS2CIEXYZMBS	237
* 2.31.2 Red as LCMS2CIEXYZMBS	238
– 2.30 class LCMS2CurveSegmentMBS	234
* 2.30.1 Constructor(nGridPoints as integer = 0)	234
* 2.30.1 Params(index as integer) as double	234
* 2.30.1 SampledPoints(index as integer) as Single	235
* 2.30.2 nGridPoints as UInt32	235
* 2.30.2 Type as Integer	235
* 2.30.2 x0 as Single	235
* 2.30.2 x1 as Single	236



# Chapter 2

## LCMS2

### 2.1 class LCMS2NamedColorListMBS

class LCMS2NamedColorListMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Specialized dictionaries for dealing with named color profiles.

#### 2.1.1 Methods

**Append(name as string, PCS() as integer, Colorant() as integer) as Boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a new spot color to the list.

**Notes:**

If the number of elements in the list exceeds the initial storage, the list is reallocated to accommodate things.

Name: The spot color name without any prefix or suffix specified in Constructor.

PCS: Encoded PCS coordinates as three integers.

Colorant: Encoded values for device colorant. (up to 16 entries)

Returns true on success and false on failure.

**Colorant(nColor as UInt32) as integer()**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns encoded colorants of nth color.

**Notes:** Array has 16 entries, but not all may be in use.

**ColorIndex(name as string) as integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a look-up in the dictionary and returns an index on the given color name.

**Notes:** Returns index on name, or -1 if the spot color is not found.

**Constructor(context as variant, n as UInt32, ColorantCount as UInt32, Prefix as string = "", Suffix as string = "")**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new named color list.

**Notes:**

Allocates an empty named color dictionary.

Context: The user-defined context cargo.

N: Initial number of spot colors in the list

Colorant count: Number of channels of device space (i.e, 3 for RGB, 4 for CMYK, etc.)

Prefix, Suffix: fixed strings for all spot color names, e.g., coated, system, ...

On success handle is not zero.

**Count as integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the number of spot colors in a named color list.

**Notes:** Returns the number of spot colors on success, 0 on error.

**Name(nColor as UInt32) as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns name of nth color.

**PCS(nColor as UInt32) as integer()**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns encoded PCS of nth color.

**Notes:** Array has 3 entries.

**Prefix(nColor as UInt32) as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns prefix of nth color.

**Suffix(nColor as UInt32) as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns suffix of nth color.

## 2.1.2 Properties

### Handle as Ptr

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal object reference.

**Notes:** (Read and Write property)

## 2.2 class LCMS2PipelineMBS

### class LCMS2PipelineMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a pipeline.

**Notes:** Pipelines are a convenient way to model complex operations on image data. Each pipeline may contain an arbitrary number of stages. Each stage performs a single operation. Pipelines may be optimized to be executed on a certain format (8 bits, for example) and can be saved as LUTs in ICC profiles.

### 2.2.1 Methods

#### Append(p as LCMS2PipelineMBS) as Boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Appends pipeline l2 at the end of pipeline l1.

**Notes:**

Channel count must match.

Returns true on success and false on failure.

#### Constructor(context as variant, InputChannels as UInt32, OutputChannels as UInt32)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Allocates an empty pipeline.

**Notes:**

Final Input and output channels must be specified at creation time.

context: A user-defined context cargo.

InputChannels, OutputChannels: Number of channels on input and output.

### **Eval16(In as Ptr, Out as Ptr)**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Evaluates a pipeline using 16-bit numbers, optionally using the optimized path.

**Notes:**

In: Input values.

Out: Output values.

For in and out you can use memoryblocks with UInt16 values.

We use Ptr for maximum performance. Please make sure the memoryblocks have right size. An UInt16 value has 2 bytes.

### **EvalFloat(In as Ptr, Out as Ptr)**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Evaluates a pipeline using floating point numbers.

**Notes:**

In: Input values.

Out: Output values.

For in and out you can use memoryblocks with single values.

We use Ptr for maximum performance. Please make sure the memoryblocks have right size. A single value has 4 bytes.

### **EvalReverseFloat(Target as Ptr, Result as Ptr, Hint as Ptr)**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Evaluates a pipeline in the reverse direction, using Newtons method.

**Notes:**

Target: Input values.

Result: Output values.

Hint: Where begin the search.

For target, result and hint you can use memoryblocks with single values.

We use Ptr for maximum performance. Please make sure the memoryblocks have right size. A single value has 4 bytes.

Returns true on success, false on error.

**FirstStage as LCMS2StageMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get a the first stage in the pipeline, or nil if pipeline is empty.

**Notes:** Intended for iterators.

**InputChannels as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the number of input channels of a given pipeline.

**Notes:** Number of channels on success, 0 on error.

**InsertStage(where as integer, stage as LCMS2StageMBS)**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Inserts a stage on either the head or the tail of a given pipeline.

**Notes:**

where: enumerated constant, either kAtBegin or kAtEnd.

stage: Pointer to a stage object

**LastStage as LCMS2StageMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get a the last stage in the pipeline, or nil if pipeline is empty.

**Notes:** Intended for iterators.

**OutputChannels as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns number of output channels of a given pipeline.

**Notes:** Number of channels on success, 0 on error.

**SetSaveAs8bitsFlag(save8bit as boolean) as Boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets an internal flag that marks the pipeline to be saved in 8 bit precision.

**Notes:**

By default all pipelines are saved on 16 bits precision on AtoB/BToA tags and in floating point precision on DToB/BToD tags.

save8bit: State of the flag, true=Save as 8 bits, false=Save as 16 bits

Returns true on success, false on error

**StageCount as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns number of stages of a given pipeline.

**Stages as LCMS2StageMBS()**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an array with all stage objects.

**UnlinkStage(where as integer) as LCMS2StageMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the stage from the pipeline.

**Notes:**

Returns the removed stage object.  
where can be kAtBegin or kAtEnd values.

**2.2.2 Properties****Context as Variant**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The context object.

**Notes:** (Read and Write property)

**Handle as Ptr**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal object reference.

**Notes:** (Read and Write property)

**2.2.3 Constants****kAtBegin = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the constants for UnlinkStage.

**Notes:** At the beginning.

**kAtEnd = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the constants for UnlinkStage.

**Notes:** At the end.

## 2.3 class LCMS2MLUMBS

class LCMS2MLUMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a multi localized unicode string.

**Notes:**

MLU funtions are the low-level interface to access the localization features of V4 ICC profiles. Little CMS does offer a high-level interface for easy operation. You may want, however, handle those objects by yourself.

Obtaining localized info from profiles

In versions prior to 4.0, the ICC format defined a required tag 'desc' which stored ASCII, Unicode, and Script Code versions of the profile description for display purposes. However, this structure allowed the profile to be localized for one language only through Unicode or Script Code. Profile vendors had to ship many localized versions to different countries. It also created problems when a document with localized profiles embedded in it was shipped to a system using a different language. With the adoption of V4 spec as basis, Little CMS solves all those issues honoring a new tag type: mluc and multi localized Unicode. There is a full part of the API to deal with this stuff, but if you dont care about the details and all you want is to display the right string, Little CMS provides a simplified interface for that purpose.

Note that ASCII is strictly 7 bits, so you need to use wide chars if you want to preserve the information in the profile. The localization trick is done by using the language and country codes, which you are supposed to supply. Those are two or three ASCII letters. A list of codes may be found here:

Language Code:

<http://lcweb.loc.gov/standards/iso639-2/iso639jac.html>

Country Codes:

<http://www.iso.ch/iso/en/prods-services/iso3166ma/index.html>

In practice, en for english and US for united states are implemented in most profiles. It is Ok to set a language and a country even if the profile does not implement such specific language and country. Little CMS will search for a proper match.

If you dont care and want just to take the first string in the profile, you can use:

For the language:  
kcmsNoLanguage

For the country:  
kcmsNoCountry

This will force to get the very first string, without any searching. A note of warning on that: you will get an string, but the language would be any, and probably that is not what you want. It is better to specify a default for language, and let LittleCMS to choose any other country (or language!) if what you ask for is not available.

### 2.3.1 Methods

#### **Constructor(context as variant, items as UInt32)**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Allocates an empty multilocalized unicode object.

#### **getASCII(LanguageCode as string, CountryCode as string) as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets an ASCII (7 bit) entry for the given Language and country.

#### **Example:**

```
dim c as new LCMS2MLUMBS(nil, 3)
```

```
call c.setASCII("en", "US", "Hello" )
```

```
call c.setASCII("de", "DE", "Hallo" )
```

```
MsgBox "en: "+c.getASCII("en", "US") + EndOfLine + "de: "+c.getASCII("de", "DE") + EndOfLine +
"any: "+c.getASCII("", "")
```

**Notes:**

Language Code: 3 chars describing the language.

CountryCode: 3 chars describing the country.

Returns the string.

**getTranslation(LanguageCode as string, CountryCode as string, byref ObtainedLanguageCode as string, byref ObtainedCountryCode as string) as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Obtains the translation rule for given multilocalized unicode object.

**Notes:**

Language Code: 3 chars describing the language.

CountryCode: 3 chars describing the country

ObtainedLanguage: 3 chars to get the language translation.

ObtainedCode: 3 chars to get the country translation.

Returns true on success, false on error

**getUnicode(LanguageCode as string, CountryCode as string) as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets an unicode (16 bit) entry for the given Language and country.

**Notes:**

Language Code: 3 chars describing the language

CountryCode: 3 chars describing the country

Returns the string value.

**setASCII(LanguageCode as string, CountryCode as string, ASCIIString as string) as Boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Fills an ASCII (7 bit) entry for the given Language and country.

**Example:**

```
dim c as new LCMS2MLUMBS(nil, 3)
```

```
call c.setASCII("en", "US", "Hello" )
```

```
call c.setASCII("de", "DE", "Hallo" )
```

```
MsgBox "en: "+c.getASCII("en", "US") + EndOfLine + "de: "+c.getASCII("de", "DE") + EndOfLine +
"any: "+c.getASCII("", "")
```

**Notes:**

Language Code: 3 chars describing the language

CountryCode: 3 chars describing the country

ASCIIString: String to add.

Returns true on success, false on error.

**setUnicode(LanguageCode as string, CountryCode as string, UnicodeString as string) as Boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Fills a UNICODE wide char (16 bit) entry for the given Language and country.

**Example:**

```
dim c as new LCMS2MLUMBS(nil, 3)
```

```
call c.setUnicode("de", "DE", "Ktzenchen" )
```

```
dim u as string = c.getUnicode("de", "DE")
```

```
MsgBox u
```

**Notes:**

Language Code: 3 chars describing the language

CountryCode: 3 chars describing the country

WideString: String to add.

Returns true on success, false on error.

### 2.3.2 Properties

#### Handle as Ptr

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal object reference.

**Notes:** (Read and Write property)

### 2.3.3 Constants

**kNoCountry = ""**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the possible country constants.

**Notes:** Any country.

**kNoLanguage = ""**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the possible language constants.

**Notes:** Any language.

## 2.4 class LCMS2ToneCurveMBS

### class LCMS2ToneCurveMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a tone curve or gamma.

### 2.4.1 Methods

#### BuildGamma(context as variant, gamma as double) as LCMS2ToneCurveMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Simplified wrapper to BuildParametricToneCurve.

**Example:**

```
dim t as LCMS2ToneCurveMBS = LCMS2ToneCurveMBS.BuildGamma(nil, 2.2)
MsgBox str(t.EstimateGamma)
```

**Notes:**

Builds a parametric curve of type 1.

Context: user-defined context object.

Gamma: Value of gamma exponent

Returns a newly created tone curve object on success, nil on error.

#### BuildParametricToneCurve(context as variant, Type as Int32, params() as double) as LCMS2ToneCurveMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Builds a parametric tone curve.

**Notes:**

Parameters:

context: user-defined context object.

Type: Number of parametric tone curve. (see LCMS2 manual)

Params: Array of tone curve parameters.

Returns a newly created tone curve object on success, nil on error.

### **BuildSegmentedToneCurve(context as variant, Segments() as LCMS2CurveSegmentMBS) as LCMS2ToneCurveMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Builds a tone curve from given segment information.

**Notes:**

context: Puser-defined context object

Segments: Array of segments

Returns a newly created tone curve object on success, nil on error.

### **BuildTabulatedToneCurve(context as variant, values() as Single) as LCMS2ToneCurveMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Builds a tone curve based on a table of floating point values.

**Example:**

```
dim values(-1) as Single
```

```
for i as integer = 0 to 9
```

```
values.Append 0.5 + i*0.02
```

```
next
```

```
dim t as LCMS2ToneCurveMBS = LCMS2ToneCurveMBS.BuildTabulatedToneCurve(nil, values)
```

```
dim items(-1) as string
```

```
for i as integer = 0 to 10
```

```
dim v as Single = t.EvalToneCurveFloat(i/10.0)
```

```
items.Append str(i/10.0)+" ->" +str(v)
```

```
next
```

```
MsgBox Join(items,EndOfLine)
```

**Notes:**

Tone curves built with this function are not restricted to 0...1.0 domain.

context: user-defined context object.

values: Array of samples. Domain of samples is 0...1.0

Returns a newly created tone curve object on success, nil on error.

See also:

- 2.4.1 BuildTabulatedToneCurve(context as variant, values() as UInt16) as LCMS2ToneCurveMBS 44

**BuildTabulatedToneCurve(context as variant, values() as UInt16) as LCMS2ToneCurveMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Builds a tone curve based on a table of 16-bit values. Tone curves built with this function are restricted to 0...1.0 domain.

**Notes:**

context: user-defined context object.

values: Array of samples. Domain is 0...65535 (UInt32).

Returns a newly created tone curve object on success, nil on error.

See also:

- 2.4.1 BuildTabulatedToneCurve(context as variant, values() as Single) as LCMS2ToneCurveMBS 43

**EstimateGamma(Precision as double = 0.01) as double**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Estimates the apparent gamma of the tone curve by using least squares fitting to a pure exponential expression in the f()

= .

**Example:**

```
dim t as LCMS2ToneCurveMBS = LCMS2ToneCurveMBS.BuildGamma(nil, 2.2)
MsgBox str(t.EstimateGamma)
```

**Notes:**

Precision: The maximum standard deviation allowed on the residuals, 0.01 is a fair value, set it to a big number to fit any curve, no matter how good is the fit.

Returns the estimated gamma at given precision, or -1.0 if the fitting has less precision.

**EvalToneCurve16(value as UInt16) as UInt16**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Evaluates the given 16-bit number across the given tone curve.

**Notes:**

This function is significantly faster than EvalToneCurveFloat, since it uses a pre-computed 16-bit lookup table.

Value: 16 bit Number to evaluate

Returns operation result

**EvalToneCurveFloat(value as Single) as Single**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Evaluates the given floating-point number across the given tone curve.

**Example:**

```
dim t1 as LCMS2ToneCurveMBS = LCMS2ToneCurveMBS.BuildGamma(nil, 1.0)
```

```
dim t2 as LCMS2ToneCurveMBS = LCMS2ToneCurveMBS.BuildGamma(nil, 2.0)
```

```
dim t3 as LCMS2ToneCurveMBS = LCMS2ToneCurveMBS.BuildGamma(nil, 3.0)
```

```
dim v1 as Double = t1.EvalToneCurveFloat(0.5)
```

```
dim v2 as Double = t2.EvalToneCurveFloat(0.5)
```

```
dim v3 as Double = t3.EvalToneCurveFloat(0.5)
```

```
MsgBox str(v1)+" "+str(v2)+" "+str(v3)
```

**Notes:**

Value: floating point number to evaluate  
Returns the result.

### IsDescending as Boolean

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if  $f(0) > f(1)$ , false otherwise.

**Notes:** Does not take unbounded parts into account.

### IsLinear as Boolean

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an estimation of cube being an identity (1:1) in the [ 0..1 ] domain.

**Notes:** Does not take unbounded parts into account. This is just a coarse approximation, with no mathematical validity.

### IsMonotonic as Boolean

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an estimation of monotonicity of curve in the [ 0..1 ] domain.

**Example:**

```
dim t as LCMS2ToneCurveMBS = LCMS2ToneCurveMBS.BuildGamma(nil, 2.2)
MsgBox "IsMonotonic: " + str(t.IsMonotonic)
```

**Notes:** Does not take unbounded parts into account. This is just a coarse approximation, with no mathematical validity.

### IsMultisegment as Boolean

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the tone curve contains more than one segment, false if it has only one segment.

**JoinToneCurve**(context as variant, X as LCMS2ToneCurveMBS, Y as LCMS2ToneCurveMBS, nPoints as UInt32) as LCMS2ToneCurveMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Composites two tone curves in the form 1(()).

**Notes:**

context: user-defined context object.

X, Y : Tone curve objects.

nPoints: Sample rate for resulting tone curve.

Returns a newly created tone curve object on success, nil on error.

**ParametricType** as integer

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the parametric type.

**Reverse** as LCMS2ToneCurveMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a tone curve that is the inverse f-1 of given tone curve.

**Notes:** Returns a newly created tone curve object on success, nil on error.

See also:

- 2.4.1 Reverse(nResultSamples as integer) as LCMS2ToneCurveMBS

47

**Reverse(nResultSamples as integer)** as LCMS2ToneCurveMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a tone curve that is the inverse f-1 of given tone curve.

**Notes:**

In the case it couldnt be analytically reversed, a tabulated curve of nResultSamples is created.

nResultSamples: Number of samples to use in the case origin tone curve couldnt be analytically reversed

Returns a newly created tone curve object on success, nil on error.

See also:

- 2.4.1 Reverse as LCMS2ToneCurveMBS

47

### Smooth(lambda as double) as Boolean

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Smooths tone curve according to the lambda parameter.

**Notes:**

Lambda: degree of smoothing.

Returns true on success, false on error.

## 2.4.2 Properties

### Handle as Ptr

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal object reference.

**Notes:** (Read and Write property)

## 2.5 class LCMS2StageSamplerMBS

### class LCMS2StageSamplerMBS

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for getting sampler callback.

### 2.5.1 Events

#### SamplerFloat(InValues as Ptr, OutValues as Ptr, InputChannels as integer, OutputChannels as integer) as boolean

Plugin Version: 12.1 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sampler event called by SampleCLutFloat.

**Notes:**

In and Out point to 32 bit float values, so please use Single property to access.  
InputChannels and OutputChannels are the number of channels.

**SamplerInteger(InValues as Ptr, OutValues as Ptr, InputChannels as integer, OutputChannels as integer) as boolean**

Plugin Version: 12.1 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sampler event called by SampleCLut16bit.

**Notes:**

In and Out point to 16 bit unsigned integer values, so please use UInt16 property to access.  
InputChannels and OutputChannels are the number of channels.

## 2.6 class LCMS2SequenceDescriptionMBS

**class LCMS2SequenceDescriptionMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a sequence description.

### 2.6.1 Properties

**AttributeFlags as UInt64**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The attribute flags.

**Notes:** (Read and Write property)

**Description as LCMS2MLUMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The description.  
**Notes:** (Read only property)

**DeviceMfg as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The device Mfg.  
**Notes:** (Read and Write property)

**DeviceModel as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The device model.  
**Notes:** (Read and Write property)

**Manufacturer as LCMS2MLUMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The manufacturer name.  
**Notes:** (Read only property)

**Model as LCMS2MLUMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The model string.  
**Notes:** (Read only property)

**ProfileID as Memoryblock**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The profile ID.  
**Notes:**

16 bytes and typically the result of a MD5.  
(Read only property)

### Technology as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The technology value.

#### Notes:

Use one of the following constants: `kcmsSigDigitalCamera`, `kcmsSigFilmScanner`, `kcmsSigReflectiveScanner`, `kcmsSigInkJetPrinter`, `kcmsSigThermalWaxPrinter`, `kcmsSigElectrophotographicPrinter`, `kcmsSigElectrostaticPrinter`, `kcmsSigDyeSublimationPrinter`, `kcmsSigPhotographicPaperPrinter`, `kcmsSigFilmWriter`, `kcmsSigVideoMonitor`, `kcmsSigVideoCamera`, `kcmsSigProjectionTelevision`, `kcmsSigCRTDisplay`, `kcmsSigPMDisplay`, `kcmsSigAMDisplay`, `kcmsSigPhotoCD`, `kcmsSigPhotoImageSetter`, `kcmsSigGravure`, `kcmsSigOffsetLithography`, `kcmsSigSilkscreen`, `kcmsSigFlexography`, `kcmsSigMotionPictureFilmScanner`, `kcmsSigMotionPictureFilmRecorder`, `kcmsSigDigitalMotionPictureCamera` or `kcmsSigDigitalCinemaProjector`.  
(Read and Write property)

## 2.7 class LCMS2ScreeningMBS

### class LCMS2ScreeningMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for screening parameters.

#### 2.7.1 Methods

##### Channel(index as integer) as LCMS2ScreeningChannelMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The screening information for each channel.

#### Notes:

Index from 0 to 15.  
(Read and Write computed property)

## 2.7.2 Properties

### Channels as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of channels.

**Notes:** (Read and Write property)

### Flag as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The flags for screening.

**Notes:**

See flags kcmsPRINTER\_DEFAULT\_SCREENINGS, kcmsFREQUENCY\_UNITS\_LINES\_CM and kcmsFREQUENCY\_UNITS\_LINES\_INCH.

(Read and Write property)

## 2.8 class LCMS2StageMBS

### class LCMS2StageMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A pipeline stage. **Notes:** Stages are single-step operations that can be chained to create pipelines. Actual stage types does include matrices, tone curves, Look-up interpolation and user-defined. There are functions to create new stage types and a plug-in type to allow stages to be saved in multi profile elements tag types. See the plug-in API for further details.

### 2.8.1 Methods

#### CLutEntries as Integer

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of entries in the CLut.

**Notes:** Only if stage is from type kcmsSigCLutElemType.

#### CLutFloatValues as double()

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns CLut floating point values.

**Notes:** Only if stage is from type kcmsSigCLutElemType and CLutHasFloatValues = true.

#### CLutHasFloatValues as Boolean

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether this CLut has floating point values.

**Notes:** Only if stage is from type kcmsSigCLutElemType.

#### CLutUInt16Values as UInt16()

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns CLut integer values.

**Notes:** Only if stage is from type kcmsSigCLutElemType and CLutHasFloatValues = false.

#### CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32) as LCMS2StageMBS

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a stage that contains a 16 bits multidimensional lookup table (CLUT).

**Notes:**

Each dimension has same resolution. The CLUT can be initialized by specifying values in Table parameter. In this variant, we do not pass a table, but you call SampleCLut16bit later.

Context: Pointer to a user-defined context cargo.

GridPoints: the number of nodes (same for each component).

inputChan: Number of input channels.

outputChan: Number of output channels.

Returns a pipeline stage on success, nil on error.

See also:

- 2.8.1 CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS 54
- 2.8.1 CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, values() as UInt16) as LCMS2StageMBS 55

**CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a stage that contains a 16 bits multidimensional lookup table (CLUT).

**Notes:**

Each dimension has same resolution. The CLUT can be initialized by specifying values in Table parameter. The recommended way is to set Table to nil and use StageSampleCLut16bit with a event, because this way the implementation is independent of the selected number of grid points.

Context: Pointer to a user-defined context cargo.

GridPoints: the number of nodes (same for each component).

inputChan: Number of input channels.

outputChan: Number of output channels.

Table: Memoryblock with a table of UInt16, holding initial values for nodes. If nil the CLUT is initialized to zero.

Returns a pipeline stage on success, nil on error.

See also:

- 2.8.1 CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32) as LCMS2StageMBS 53
- 2.8.1 CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, values() as UInt16) as LCMS2StageMBS 55

**CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, values() as UInt16) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a stage that contains a 16 bits multidimensional lookup table (CLUT).

**Notes:**

Each dimension has same resolution. The CLUT can be initialized by specifying values in Table parameter.

Context: Pointer to a user-defined context cargo.

GridPoints: the number of nodes (same for each component).

inputChan: Number of input channels.

outputChan: Number of output channels.

values: array of UInt16, holding initial values for nodes.

Returns a pipeline stage on success, nil on error.

See also:

- 2.8.1 CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32) as LCMS2StageMBS 53
- 2.8.1 CreateStageWithCLut16bit(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS 54

**CreateStageWithCLut16bitGranular(Context as variant, clutPoints as Memoryblock, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Similar to CreateStageWithCLut16bit, but it allows different granularity on each CLUT dimension.

**Notes:**

Context: user-defined context cargo.

clutPoints: Memoryblock with array [ inputChan ] of UInt32 holding the number of nodes for each component.

inputChan: Number of input channels.

outputChan: Number of output channels.

Table: Memoryblock with table of UInt16, holding initial values for nodes. If nil the CLUT is initialized to zero.

Returns a pipeline stage on success, nil on error.

**CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32) as LCMS2StageMBS**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a stage that contains a float multidimensional lookup table (CLUT).

**Notes:**

Each dimension has same resolution. The CLUT can be initialized by specifying values in Table parameter. In this variant, we do not pass a table, but you call SampleCLutFloat later.

Context: user-defined context cargo.

GridPoints: the number of nodes (same for each component).

inputChan: Number of input channels.

outputChan: Number of output channels.

Returns a pipeline stage on success, nil on error.

See also:

- 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS 56
- 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, values() as double) as LCMS2StageMBS 57

**CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a stage that contains a float multidimensional lookup table (CLUT).

**Notes:**

Each dimension has same resolution. The CLUT can be initialized by specifying values in Table parameter. The recommended way is to set Table to nil and use StageSampleCLutFloat with an event, because this way the implementation is independent of the selected number of grid points.

Context: user-defined context cargo.

GridPoints: the number of nodes (same for each component).

inputChan: Number of input channels.

outputChan: Number of output channels.

Table: Memoryblock with a table of Single (Float32) values, holding initial values for nodes. If nil the CLUT

is initialized to zero.

Returns a pipeline stage on success, nil on error.

See also:

- 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32) as LCMS2StageMBS 56
- 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, values() as double) as LCMS2StageMBS 57

**CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, values() as double) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a stage that contains a float multidimensional lookup table (CLUT).

**Notes:**

Each dimension has same resolution. The CLUT can be initialized by specifying values in Table parameter.

Context: user-defined context cargo.

GridPoints: the number of nodes (same for each component).

inputChan: Number of input channels.

outputChan: Number of output channels.

values: Array of single (Float32) values, holding initial values for nodes.

Returns a pipeline stage on success, nil on error.

See also:

- 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32) as LCMS2StageMBS 56
- 2.8.1 CreateStageWithCLutFloat(Context as variant, GridPoints as UInt32, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS 56

**CreateStageWithCLutFloatGranular(Context as variant, clutPoints as Memoryblock, inputChan as UInt32, outputChan as UInt32, Table as Memoryblock) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Similar to CreateStageWithCLutFloat, but it allows different granularity on each CLUT dimension.

**Notes:**

Context: user-defined context cargo.

clutPoints: Memoryblock with Array of UInt32 [ inputChan ] holding the number of nodes for each component.

inputChan: Number of input channels.

outputChan: Number of output channels.

Table: a pointer to a table of Singles (Float32), holding initial values for nodes.

Returns a pipeline stage on success, nil on error.

**CreateStageWithIdentity(Context as variant, Channels as UInt32) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates an empty (identity) stage that does no operation.

**Notes:**

May be needed in order to save the pipeline as AToB/BToA tags in ICC profiles.

Context: user-defined context cargo.

Channels: Number of channels

Returns a pipeline stage on success, nil on error.

**CreateStageWithMatrix(Context as variant, Rows as UInt32, Cols as UInt32, Matrix as Memoryblock, Offset as Memoryblock = nil) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a stage that contains a matrix plus an optional offset.

**Notes:**

Note that Matrix is specified in double precision, whilst CLUT has only float precision. That is because an ICC profile can encode matrices with far more precision than CLUTS.

Context: user-defined context cargo.

Rows, Cols: Dimensions of matrix

Matrix: Memoryblock with a matrix of [ Rows, Cols ] (double values, 8 byte per value)  
Offset: Memoryblock with a vector of [ Cols ] , nil if no offset is to be applied.

Returns a pipeline stage on success, nil on error.

### **CreateStageWithToneCurves(Context as variant, Channels() as LCMS2ToneCurveMBS) as LCMS2StageMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a stage that contains n channels tone curves, one per channel.

#### **Notes:**

Setting Curves to nil forces identity (1:1) curves to be used. The stage keeps and owns a private copy of the tone curve objects.

Context: user-defined context cargo.

Curvess: Array of tone curves objects, one per channel.

Returns a pipeline stage on success, nil on error.

### **Data as Ptr**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The raw stage data.

### **InputChannels as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the number of input channels of a given stage object.

**MatrixOffsets as double()**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns array with matrix offsets.

**Notes:** Only if stage is from type `kcmsSigMatrixElemType` and.

**MatrixValues as double()**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns array with matrix values.

**Notes:** Only if stage is from type `kcmsSigMatrixElemType` and.

**NextItem as LCMS2StageMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns next stage in pipeline list, or nil if end of list.

**Notes:** Intended for iterators.

**OutputChannels as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the number of output channels of a this stage object.

**SampleCLut16bit(sampler as LCMS2StageSamplerMBS, Flags as integer = 0)**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Iterate on all nodes of a given CLUT stage, calling a 16-bit sampler on each node.

**Notes:**

Those functions (`SampleCLut16bit` and `SampleCLutFloat`) are provided to populate CLUT stages in a way that is independent of the number of nodes. The programmer has to provide an object with event that will be invoked on each CLUT node. LittleCMS does fill the `In` parameter with the coordinates that addresses the node. It also fills the `Out` parameter with CLUT contents on the node, so this can be used also to get CLUT contents after reading it from an ICC profile. In this case, a special flag can be specified to make sure

the CLUT is being accessed as read-only and not modified (kSamplerInspect).

Works only with CLut stage objects and returns false if the object is not a Clut.

Sampler: The object to receive events.

Flags: Bit-field flags for different options. Only kSamplerInspect is currently supported.

Returns true on success, false on error.

### **SampleCLutFloat(sampler as LCMS2StageSamplerMBS, Flags as integer = 0)**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Iterate on all nodes of a given CLUT stage, calling a float sampler on each node.

**Notes:**

Those functions (SampleCLut16bit and SampleCLutFloat) are provided to populate CLUT stages in a way that is independent of the number of nodes. The programmer has to provide an object with event that will be invoked on each CLUT node. LittleCMS does fill the In parameter with the coordinates that addresses the node. It also fills the Out parameter with CLUT contents on the node, so this can be used also to get CLUT contents after reading it from an ICC profile. In this case, a special flag can be specified to make sure the CLUT is being accessed as read-only and not modified (kSamplerInspect).

Works only with CLut stage objects and returns false if the object is not a Clut.

Sampler: The object to receive events.

Flags: Bit-field flags for different options. Only kSamplerInspect is currently supported.

Returns true on success, false on error.

### **ToneCurves as LCMS2ToneCurveMBS()**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns array with tone curves.

**Notes:** Only if stage is from type kcmsSigCurveSetElemType and.

### Type as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the type of a given stage object.

**Notes:** Use this type constants: `kcmsSigCurveSetElemType`, `kcmsSigMatrixElemType`, `kcmsSigCLutElemType`, `kcmsSigBAcsElemType`, `kcmsSigEAcsElemType`, `kcmsSigXYZ2LabElemType`, `kcmsSigLab2XYZElemType`, `kcmsSigNamedColorElemType`, `kcmsSigLabV2toV4`, `kcmsSigLabV4toV2`, `kcmsSigIdentityElemType`.

## 2.8.2 Properties

### Handle as Ptr

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal object reference.

**Notes:** (Read and Write property)

## 2.8.3 Constants

`kSamplerInspect = & h01000000`

Plugin Version: 12.1 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flags for Sampling.

**Notes:** Use this flag to prevent changes being written to destination when using `SampleCLutFloat` or `SampleCLut16bit`.

## 2.9 class LCMS2SequenceMBS

`class LCMS2SequenceMBS`

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Profile sequence descriptors.

**Notes:** Profile sequence can be read/written by using `cmsReadTag` and `cmsWriteTag` functions.

### 2.9.1 Methods

#### Constructor(context as variant, Count as UInt32)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new sequence with the given number of entries.

**Example:**

```
dim d as new LCMS2SequenceMBS(nil, 5)
MsgBox str(d.Count)
```

**Notes:** On success the handle property is not zero.

#### Description(index as integer) as LCMS2SequenceDescriptionMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The sequence descriptions.

**Notes:**

Index from 0 to count-1.  
(Read and Write computed property)

### 2.9.2 Properties

#### Count as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of sequence descriptions used in this class.

**Notes:** (Read only property)

**Handle as Ptr**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal object reference.

**Notes:** (Read and Write property)

**2.10 class LCMS2Vec3MBS****class LCMS2Vec3MBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a three dimension vector.

**Example:**

```
dim v as new LCMS2Vec3MBS(1,2,3)
MsgBox str(v.X)+" "+str(v.y)+" "+str(v.z)
```

**2.10.1 Methods****Clone as LCMS2Vec3MBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a copy of the vector.

**Constructor(other as LCMS2Vec3MBS)**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The copy constructor.

**Example:**

```
// make a vector
dim v as new LCMS2Vec3MBS(1,2,3)

// create a copy
```

```
dim w as new LCMS2Vec3MBS(v)

// show values
MsgBox str(w.X)+" "+str(w.y)+" "+str(w.z)
```

See also:

- 2.10.1 Constructor(v1 as double = 0.0, v2 as double = 0.0, v3 as double = 0.0)

65

**Constructor(v1 as double = 0.0, v2 as double = 0.0, v3 as double = 0.0)**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The constructor to initialize the object with the given values.

**Example:**

```
dim v as new LCMS2Vec3MBS(1,2,3)
MsgBox str(v.X)+" "+str(v.y)+" "+str(v.z)
```

See also:

- 2.10.1 Constructor(other as LCMS2Vec3MBS)

64

**value(index as UInt32) as double**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The values by index.

**Notes:**

Index from 0 to 2.

(Read and Write computed property)

## 2.10.2 Properties

### X as Double

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The first value.  
**Notes:** (Read and Write property)

### Y as Double

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The second value.  
**Notes:** (Read and Write property)

### Z as Double

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The third value.  
**Notes:** (Read and Write property)

## 2.11 class LCMS2UcrBgMBS

### class LCMS2UcrBgMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for the cmsSigUcrBgType tag.  
**Notes:** This is for Undercolorremoval and black generation.

### 2.11.1 Methods

**Constructor**(Ucr as LCMS2ToneCurveMBS = nil, Bg as LCMS2ToneCurveMBS = nil, Desc as LCMS2MLUMBS = nil)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new object with given values.

### 2.11.2 Properties

#### Bg as LCMS2ToneCurveMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The black generation value.

**Notes:** (Read and Write property)

#### Desc as LCMS2MLUMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The description.

**Notes:** (Read and Write property)

#### Ucr as LCMS2ToneCurveMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The undercolor removal setting.

**Notes:** (Read and Write property)

## 2.12 class LCMS2ViewingConditionsMBS

### class LCMS2ViewingConditionsMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Viewing conditions. **Notes:** From LCMS documentation: Please note those are CAM model viewing conditions, and not the ICC tag viewing conditions, which I'm naming cmsICCViewingConditions to make differences evident. Unfortunately, the tag cannot deal with surround La, Yb and D value so is basically useless to store CAM02 viewing conditions.

### 2.12.1 Methods

#### Clone as LCMS2ViewingConditionsMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the object.

#### Constructor(other as LCMS2ViewingConditionsMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes object with values from other object.

See also:

- 2.12.1 Constructor(whitePoint as LCMS2CIEXYZMBS = nil, Yb as double = 0.0, La as double = 0.0, surround as integer = 0, D\_ value as double = 0.0) 68

#### Constructor(whitePoint as LCMS2CIEXYZMBS = nil, Yb as double = 0.0, La as double = 0.0, surround as integer = 0, D\_ value as double = 0.0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new viewing conditions object.

See also:

- 2.12.1 Constructor(other as LCMS2ViewingConditionsMBS) 68

### 2.12.2 Properties

#### D\_ value as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The D value value.  
**Notes:** (Read and Write property)

**La as Double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The La value.  
**Notes:** (Read and Write property)

**Surround as Int32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The surround value.

**Notes:**

Use this constants:

kAVG\_SURROUND = 1

kDIM\_SURROUND = 2

kDARK\_SURROUND = 3

kCUTSHEET\_SURROUND = 4

(Read and Write property)

**whitePoint as LCMS2CIEXYZMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The white point.

**Notes:** (Read and Write property)

**Yb as Double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Yb value.

**Notes:** (Read and Write property)

## 2.13 class LCMS2TransformMBS

class LCMS2TransformMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a LCMS 2.x transformation.

### 2.13.1 Methods

**ChangeBuffersFormat(InputFormat as UInt32, OutputFormat as UInt32) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This function does change the encoding of buffers in a yet-existing transform.

**Notes:**

Not all transforms can be changed, cmsChangeBuffersFormat only works on transforms created originally with at least 16 bits of precision. This function is provided for backwards compatibility and should be avoided whenever possible, as it prevents transform optimization.

InputFormat: A bit-field format specifier as described in Formatters section.

OutputFormat: A bit-field format specifier as described in Formatters section.

Returns true on success and false on failure.

**CreateMultiprofileTransform(context as variant, Profiles() as LCMS2ProfileMBS, InputFormat as UInt32, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a multi profile transformation.

**Notes:**

User passes in an array of handles to open profiles. The returned color transform do "smelt" all profiles in a single devicelink. Color spaces must be paired with the exception of Lab/XYZ, which can be interchanged.

context: Optional context object.

Profiles: Array of open profile objects.

InputFormat: A bit-field format specifier as described in Formatters section.

OutputFormat: A bit-field format specifier as described in Formatters section.

Intent: The intent code, as described in Intents section.

Flags: A combination of bit-field of kcmsFLAGS\_ \* constants.

Returns a transform object on success, nil on error.

See also:

- 2.13.1 CreateMultiprofileTransform(Profiles() as LCMS2ProfileMBS, InputFormat as UInt32, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS 71

**CreateMultiprofileTransform(Profiles() as LCMS2ProfileMBS, InputFormat as UInt32, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a multi profile transformation.

**Notes:**

User passes in an array of handles to open profiles. The returned color transform do "smelt" all profiles in a single devicelink. Color spaces must be paired with the exception of Lab/XYZ, which can be interchanged.

context: Optional context object.

Profiles: Array of open profile objects.

InputFormat: A bit-field format specifier as described in Formatters section.

OutputFormat: A bit-field format specifier as described in Formatters section.

Intent: The intent code, as described in Intents section.

Flags: A combination of bit-field of kcmsFLAGS\_ \* constants.

Returns a transform object on success, nil on error.

See also:

- 2.13.1 CreateMultiprofileTransform(context as variant, Profiles() as LCMS2ProfileMBS, InputFormat as UInt32, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS 70

**CreateProofingTransform(context as variant, InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Proofing as LCMS2ProfileMBS, Intent as UInt32, ProofingIntent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Same as CreateTransform(), but including soft-proofing.

**Notes:**

A proofing transform does emulate the colors that would appear as the image were rendered on a specific device. That is, for example, with a proofing transform I can see how will look a photo of my little daughter if rendered on my HP printer. Since most printer profiles does include some sort of gamut-remapping, it is likely colors will not look as the original. Using a proofing transform, it can be done by using the appropriate function. Note that this is an important feature for final users, it is worth of all color-management stuff if the final media is not cheap.

The obtained transform emulates the device described by the "Proofing" profile. Useful to preview final result without rendering to the physical medium. To enable proofing and gamut check you need to include following flags:

cmsFLAGS\_ GAMUTCHECK: Color out of gamut are flagged to a fixed color defined by the function `kcmsSetAlarmCodes`  
 cmsFLAGS\_ SOFTPROOFING: does emulate the Proofing device.

context: Optional context object.

InputProfile: A profile object capable to work in input direction

InputFormat: A bit-field format specifier as described in Formatters section.

OutputProfile: A profile object capable to work in output direction

OutputFormat: A bit-field format specifier as described in Formatters section.

Intent: The intent code.

ProofingIntent: The intent code.

Flags: A combination of bit-field constants described in Table 42.

Returns transform object on success, nil on error.

See also:

- 2.13.1 CreateProofingTransform(InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Proofing as LCMS2ProfileMBS, Intent as UInt32, ProofingIntent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS 73

**CreateProofingTransform**(InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Proofing as LCMS2ProfileMBS, Intent as UInt32, ProofingIntent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Same as CreateTransform(), but including soft-proofing.

**Notes:**

A proofing transform does emulate the colors that would appear as the image were rendered on a specific device. That is, for example, with a proofing transform I can see how will look a photo of my little daughter if rendered on my HP printer. Since most printer profiles does include some sort of gamut-remapping, it is likely colors will not look as the original. Using a proofing transform, it can be done by using the appropriate function. Note that this is an important feature for final users, it is worth of all color-management stuff if the final media is not cheap.

The obtained transform emulates the device described by the "Proofing" profile. Useful to preview final result without rendering to the physical medium. To enable proofing and gamut check you need to include following flags:

cmsFLAGS\_ GAMUTCHECK: Color out of gamut are flagged to a fixed color defined by the function kms-SetAlarmCodes

cmsFLAGS\_ SOFTPROOFING: does emulate the Proofing device.

context: Optional context object.

InputProfile: A profile object capable to work in input direction

InputFormat: A bit-field format specifier as described in Formatters section.

OutputProfile: A profile object capable to work in output direction

OutputFormat: A bit-field format specifier as described in Formatters section.

Intent: The intent code.

ProofingIntent: The intent code.

Flags: A combination of bit-field constants described in Table 42.

Returns transform object on success, nil on error.

See also:

- 2.13.1 CreateProofingTransform(context as variant, InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Proofing as LCMS2ProfileMBS, Intent as UInt32, ProofingIntent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS

**CreateTransform(context as variant, InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a color transform for translating bitmaps.

**Notes:**

context: Optional, the context object.

InputProfile: A profile object capable to work in input direction

InputFormat: A bit-field format specifier as described in Formatters section.

OutputProfile: A profile object capable to work in output direction

OutputFormat: A bit-field format specifier as described in Formatters section.

Intent: The intent code, as described in Intents section.

Flags: A combination of bit-field kcmsFLAGS\_ \* constants.

Returns a transform object on success, NULL on error.

See also:

- 2.13.1 CreateTransform(InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS 74

**CreateTransform(InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a color transform for translating bitmaps.

**Notes:**

context: Optional, the context object.

InputProfile: A profile object capable to work in input direction

InputFormat: A bit-field format specifier as described in Formatters section.

OutputProfile: A profile object capable to work in output direction

OutputFormat: A bit-field format specifier as described in Formatters section.

Intent: The intent code, as described in Intents section.

Flags: A combination of bit-field kcmsFLAGS\_ \* constants.

Returns a transform object on success, NULL on error.

See also:

- 2.13.1 CreateTransform(context as variant, InputProfile as LCMS2ProfileMBS, InputFormat as UInt32, OutputProfile as LCMS2ProfileMBS, OutputFormat as UInt32, Intent as UInt32, Flags as UInt32 = 0) as LCMS2TransformMBS 74

### InputFormat as integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the input format associated with a given transform.

### NamedColorList as LCMS2NamedColorListMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Retrieve a named color list from a given color transform.

**Notes:** Returns named color list dictionary on success, nil on error.

### OutputFormat as integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the output format associated with a given transform.

### ToDeviceLink(Version as double, Flags as UInt32) as LCMS2ProfileMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Generates a device-link profile from a given color transform.

**Notes:**

This profile can then be used by any other function accepting profile handle. Depending on the specified version number, the implementation of the devicelink may vary. Accepted versions are in range 1.0...4.3

Version: The target devicelink version number.

Flags: A combination of bit-field constants kcmsFLAGS\_ \*.

Returns an ICC profile object on success, nil on error.

**Transform(bitmap as LCMS2BitmapMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This function translates bitmaps according of parameters setup when creating the color transform.

**Notes:**

bitmap: the input and output bitmap.

Returns true on success.

See also:

- 2.13.1 Transform(inBitmap as LCMS2BitmapMBS,outBitmap as LCMS2BitmapMBS) as boolean 76
- 2.13.1 Transform(InputBuffer as Ptr, OutputBuffer as Ptr, Size as UInt32) as boolean 76

**Transform(inBitmap as LCMS2BitmapMBS,outBitmap as LCMS2BitmapMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This function translates bitmaps according of parameters setup when creating the color transform.

**Notes:**

inBitmap: the input bitmap.

outBitmap: the output bitmap.

Returns true on success.

See also:

- 2.13.1 Transform(bitmap as LCMS2BitmapMBS) as boolean 76
- 2.13.1 Transform(InputBuffer as Ptr, OutputBuffer as Ptr, Size as UInt32) as boolean 76

**Transform(InputBuffer as Ptr, OutputBuffer as Ptr, Size as UInt32) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This function translates bitmaps according of parameters setup when creating the color transform.

**Notes:**

InputBuffer: A pointer to the input bitmap.

OutputBuffer: A pointer to the output bitmap.

Size: the number of PIXELS to be transformed.

Returns true on success.

See also:

- 2.13.1 Transform(bitmap as LCMS2BitmapMBS) as boolean 76
- 2.13.1 Transform(inBitmap as LCMS2BitmapMBS,outBitmap as LCMS2BitmapMBS) as boolean 76

### TransformRGB(c as color) as color

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Convenience function to convert one RGB color.

**Notes:** Works for input/output data with Float (4 or 8 byte) or Integer (1, 2 or 4 bytes).

## 2.13.2 Properties

### Context as Variant

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The context object.

**Notes:** (Read and Write property)

### Handle as Ptr

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal object reference.

**Notes:** (Read and Write property)

## 2.14 class LCMS2ProfileMBS

### class LCMS2ProfileMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a LCMS2 profile.

### 2.14.1 Methods

#### ChannelCount as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the number of channels used for the colorspace of this profile.

**Example:**

```
dim p as LCMS2ProfileMBS = LCMS2ProfileMBS.CreateSRGBProfile
MsgBox str(p.ChannelCount)
```

#### ColorSpaceType as Integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets/Sets the color space used by the given profile, using the ICC convention.

**Notes:** (Read and Write computed property)

#### Constructor(context as variant = nil)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new empty profile.

**Notes:** On success the handle property is not zero.

See also:

- 2.14.1 Constructor(file as folderitem, write as boolean = false)

**Constructor(file as folderitem, write as boolean = false)**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates profile by reading in an existing profile or creating a new profile at the given location.

**Notes:**

file: file location.

write: Whether to create new profile.

On success the handle property is not zero.

See also:

- 2.14.1 Constructor(context as variant = nil)

78

**CreateBCHSWabstractProfile(context as variant, nLUTPoints as integer, Bright as double, Contrast as double, Hue as double, Saturation as double, TempSrc as integer, TempDest as integer) as LCMS2ProfileMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates an abstract devicelink operating in Lab for Bright/Contrast/Hue/Saturation and white point translation.

**Notes:**

White points are specified as temperatures degree of Kelvin.

context: optional context object.

nLUTPoints: Resulting colormap resolution

Bright: Bright increment. May be negative

Contrast: Contrast increment. May be negative.

Hue: Hue displacement in degree.

Saturation: Saturation increment. May be negative

TempSrc: Source white point temperature

TempDest: Destination white point temperature.

Returns an ICC profile object on success, nil on error.

**CreateGrayProfile(context as variant, WhitePoint as LCMS2CIExyYMBS, TransferFunction as LCMS2ToneCurveMBS) as LCMS2ProfileMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This function creates a gray profile based on White point and transfer function.

**Notes:**

It populates followingtags; this conform a standard gray display profile:

- 1 cmsSigProfileDescriptionTag
- 2 cmsSigMediaWhitePointTag
- 3 cmsSigGrayTRCTag

Context: Optional context object.

WhitePoint: The white point of the gray device or space.

TransferFunction: tone curve describing the device or space gamma.

Returns an ICC profile object on success, NULL on error.

**CreateInkLimitingDeviceLink(context as variant, ColorSpaceSignature as UInt32, Limit as double) as LCMS2ProfileMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This is a devicelink operating in CMYK for ink-limiting.

**Notes:**

Space: any color space signature. Currently only kcmsSigCmykData is supported.

Limit: Amount of ink limiting in % (0..400% )

Returns new profile or nil on error.

**CreateLab2Profile(context as variant = nil, point as LCMS2CIExyYMBS = nil) as LCMS2ProfileMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a Lab to Lab identity, marking it as v2 ICC profile.

**Notes:**

Adjustments for accomodating PCS endoing shall be done by Little CMS when using this profile.

Context: The optional context object.

WhitePoint: Lab reference white. nil for D50.

Returns a handle to an ICC profile object on success, nil on error.

**CreateLab4Profile(context as variant = nil, point as LCMS2CIExyYMBS = nil) as LCMS2ProfileMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a Lab to Lab identity, marking it as v4 ICC profile.

**Notes:**

Context: The optional context object.

WhitePoint: Lab reference white. nil for D50.

Returns a handle to an ICC profile object on success, nil on error.

**CreateLinearizationDeviceLink(context as variant, ColorSpaceSignature as UInt32, TransferFunction() as LCMS2ToneCurveMBS) as LCMS2ProfileMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This is a devicelink operating in the target colorspace with as many transfer functions ascomponents.

**Notes:**

Space: The desired color space signature. Like & h52474220 for RGB.

TransferFunction: tone curves describing the device or space linearization.

Please make sure you pass right number of transfer functions matching number of channels of color space.

A handle to an ICC profile object on success, NULL on error.

**CreateNULLProfile(context as variant = nil) as LCMS2ProfileMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a fake NULL profile.

**Notes:**

This profile return 1 channel as always 0. Is useful only for gamut checking tricks.  
Returns an ICC profile object on success, nil on error.

**CreateProfilePlaceholder(context as variant = nil) as LCMS2ProfileMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates an empty profile object, to be populated by the programmer.

**Notes:**

WARNING: The profile without adding any information is not directly useable.

Context: The context object.

Returns an ICC profile object on success, nil on error.

**CreateRGBProfile(context as variant, WhitePoint as LCMS2CIExyYMBS, Primaries as LCMS2CIExyYTripl TransferFunction() as LCMS2ToneCurveMBS) as LCMS2ProfileMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This function creates a RGB profile based on White point, primaries and transfer functions.

**Notes:**

It populates following tags; this conform a standard RGB Display Profile, and then I add (As per addendum II) chromaticity tag.

Context: Optional context object.

WhitePoint: The white point of the RGB device or space.

Primaries: The primaries in xyY of the device or space.

TransferFunction: 3 tone curves describing the device or space gamma. (if you pass just one, the plugin uses it for all three channels)

Returns the new ICC profile object or nil on any error.

- 1 cmsSigProfileDescriptionTag
- 2 cmsSigMediaWhitePointTag
- 3 cmsSigRedColorantTag
- 4 cmsSigGreenColorantTag
- 5 cmsSigBlueColorantTag
- 6 cmsSigRedTRCTag
- 7 cmsSigGreenTRCTag
- 8 cmsSigBlueTRCTag
- 9 Chromatic adaptation Tag
- 10 cmsSigChromaticityTag

#### CreateSRGBProfile(context as variant = nil) as LCMS2ProfileMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Create an ICC virtual profile for sRGB space.

#### Notes:

sRGB is a standard RGB color space created cooperatively by HP and Microsoft in 1996 for use on monitors, printers, and the Internet.

sRGB white point is D65.

xyY 0.3127, 0.3291, 1.0

Primaries are ITU-R BT.709-5 (xYY)

R 0.6400, 0.3300, 1.0

G 0.3000, 0.6000, 1.0

B 0.1500, 0.0600, 1.0

31

Predefined virtual profiles sRGB transfer functions are defined by:

If RsRGB, GsRGB, BsRGB < 0.04045

R = RsRGB / 12.92

G = GsRGB / 12.92

B = BsRGB / 12.92

elseif RsRGB, GsRGB, BsRGB >= 0.04045

```

R = ((RsRGB + 0.055) / 1.055)2.4
G = ((GsRGB + 0.055) / 1.055)2.4
B = ((BsRGB + 0.055) / 1.055)2.4
end if

```

Context: Optional context object.

Returns an ICC profile object on success, nil on error.

### CreateXYZProfile(context as variant = nil) as LCMS2ProfileMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a XYZ to XYZ identity, marking it as v4 ICC profile.

**Notes:**

WhitePoint used in Absolute colorimetric intent is D50.

Returns the new profile on success or nil on failure.

### DetectTAC as double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Detects total area coverage.

**Notes:**

When several colors are printed on top of each other, there is a limit to the amount of ink that can be put on paper. This maximum total dot percentage is referred to as either TIC (Total Ink Coverage) or TAC (Total Area Coverage). This function does estimate total area coverage for a given profile in % . Only works on output profiles. On RGB profiles, 400% is returned. TAC is detected by subsampling Lab color space on 6x74x74 points.

Returns estimated area coverage in % on success, 0 on error.

### DeviceClass as Integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets/sets the device class signature from profile header.

**Notes:**

Device Classes:

kcmsSigInputClass	& h73636E72	scnr
kcmsSigDisplayClass	& h6D6E7472	mmtr
kcmsSigOutputClass	& h70727472	prtr
kcmsSigLinkClass	& h6C696E6B	link
kcmsSigAbstractClass	& h61627374	abst
kcmsSigColorSpaceClass	& h73706163	spac
kcmsSigNamedColorClass	& h6e6d636c	nmcl

(Read and Write computed property)

### **FormatterForBitmap(BitCount as integer = 8) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Build a suitable formatter for the colorspace of this profile.

**Notes:**

This is a convenience function which prepares you a pixel format for use with LCMS2BitmapMBS class. Formatters are used to describe how bitmap buffers are organized.

### **FormatterForColorspace(nBytes as UInt32, IsFloat as boolean = false) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Build a suitable formatter for the colorspace of this profile.

**Notes:** Formatters are used to describe how bitmap buffers are organized.

### **FormatterForPCS(nBytes as UInt32, IsFloat as boolean = false) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Build a suitable formatter for the colorspace of this profile.

**Notes:** Formatters are used to describe how bitmap buffers are organized.

**GetProfileInfo(Info as integer, LanguageCode as string, CountryCode as string) as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets several information strings from the profile, dealing with localization.

**Notes:**

Info: A selector of which info to return. (kInfoCopyright, kInfoDescription, kInfoManufacturer or kInfo-Model)

Language Code: first name language code from ISO-639/2.

Country Code: first name region code from ISO-3166.

Returns the string. (empty string on error)

**HeaderAttributes as UInt64**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get/set header attribute flags.

**Notes:**

Flags can be a combination of kcmsReflective, kcmsTransparency, kcmsGlossy or kcmsMatte.  
(Read and Write computed property)

**HeaderCreationDateTime as LCMS2DateMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the date and time when profile was created.

**Notes:**

This is a field stored in profile header.

Returns nil on any error.

**HeaderFlags as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get or set header flags of given ICC profile object.

**Notes:**

The profile flags field does contain flags to indicate various hints for the CMM such as distributed processing and caching options. The least-significant 16 bits are reserved for the ICC. Flags in bit positions 0 and 1 shall be used as indicated below.

Position	Field Length (bits)	Field Contents
0	1	Embedded Profile (kcmsEmbeddedProfileFalse if not embedded, kcmsEmbeddedProfileTrue if embedded in file)
1	1	Profile cannot be used independently from the embedded color data (set to kcmsUseWithEmbeddedDataOnly if true, kcmsUseAnywhere if false)

(Read and Write computed property)

### HeaderManufacturer as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get/set the manufacturer signature as described in the header.

**Notes:**

This functionality is widely superseded by the manufacturer tag. Of use only in elder profiles.

(Read and Write computed property)

### HeaderModel as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get/set the model signature as described in the header.

**Notes:**

This functionality is widely superseded by the model tag. Of use only in elder profiles.

(Read and Write computed property)

### HeaderProfileID as string

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get/set header profile ID.

**Notes:**

Profile ID must be a 16 byte long string.  
(Read and Write computed property)

### **IsCLUT(Intent as UInt32, UsedDirection as UInt32) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns whatever a CLUT is present in the profile for the given intent and direction.

**Notes:**

Intent: The intent code.

UsedDirection: UsedAsInput = 0, UsedAsOutput = 1, UsedAsProof = 2.

Returns true CLUT is present for given intent and direction, false otherwise.

### **IsIntentSupported(Intent as UInt32, UsedDirection as UInt32) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the requested intent is implemented in the given direction.

**Notes:**

Little CMS has a fallback strategy that allows to specify any rendering intent when creating the transform, but the intent really being used may be another if the requested intent is not implemented.

UsedDirection: UsedAsInput = 0, UsedAsOutput = 1, UsedAsProof = 2.

Returns true if the intent is implemented, false otherwise.

### **IsMatrixShaper as Boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns whatever a matrix-shaper is present in the profile.

**Notes:**

Note that a profile may hold matrix-shaper and CLUT as well.

Returns true if the profile holds a matrix-shaper, false otherwise.

**IsTag(TagSignature as integer) as Boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if a tag with signature sig is found on the profile.

**Notes:**

Useful to check if a profile contains a given tag.  
Returns true if the tag is found or false otherwise.

**LinkTag(sig as integer, dest as integer) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a directory entry on tag sig that points to same location as tag dest.

**Notes:**

Using this function you can collapse several tag entries to the same block in the profile.

sig: Signature of linking tag.  
dest: Signature of linked tag.

Returns true on success, false on error

**MD5computeID as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Computes a MD5 checksum and stores it as Profile ID in the profile header.

**Notes:** Returns true on success or false on failure.

**Name as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the profile name.

**Notes:** This is a convenience function. The plugin builds this name from manufacturer, model and description strings.

**OpenProfileFromFile(context as variant, file as folderitem, write as boolean = false) as LCMS2ProfileMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens a profile from a file or creates a new profile file there.

**Notes:**

context: Optional context object.

file: The folderitem for the file location.

write: If true, a new profile is created. If false an existing profile is opened.

Returns a new ICC Profile object on success or nil on failure.

See also:

- 2.14.1 OpenProfileFromFile(file as folderitem, write as boolean = false) as LCMS2ProfileMBS 90

**OpenProfileFromFile(file as folderitem, write as boolean = false) as LCMS2ProfileMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens a profile from a file or creates a new profile file there.

**Notes:**

context: Optional context object.

file: The folderitem for the file location.

write: If true, a new profile is created. If false an existing profile is opened.

Returns a new ICC Profile object on success or nil on failure.

See also:

- 2.14.1 OpenProfileFromFile(context as variant, file as folderitem, write as boolean = false) as LCMS2ProfileMBS 90

**OpenProfileFromMemory(context as variant, data as Memoryblock) as LCMS2ProfileMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens an ICC profile which is entirely contained in a memory block.

**Notes:**

Context: Optional, the context object.

Data: The profile data.

Useful for accessing embedded profiles. This buffer must hold a full profile image. Memory must be contiguous.

Returns an ICC profile object on success, nil on error.

See also:

- 2.14.1 OpenProfileFromMemory(data as Memoryblock) as LCMS2ProfileMBS 91

### OpenProfileFromMemory(data as Memoryblock) as LCMS2ProfileMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens an ICC profile which is entirely contained in a memory block.

**Notes:**

Context: Optional, the context object.

Data: The profile data.

Useful for accessing embedded profiles. This buffer must hold a full profile image. Memory must be contiguous.

Returns an ICC profile object on success, nil on error.

See also:

- 2.14.1 OpenProfileFromMemory(context as variant, data as Memoryblock) as LCMS2ProfileMBS 90

### OpenProfileFromString(context as variant, data as string) as LCMS2ProfileMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens an ICC profile which is entirely contained in a string.

**Notes:**

Context: Optional, the context object.

Data: The profile data.

Useful for accessing embedded profiles. This buffer must hold a full profile image. Memory must be contiguous.

Returns an ICC profile object on success, nil on error.

See also:

- 2.14.1 `OpenProfileFromString(data as string)` as `LCMS2ProfileMBS` 92

### **OpenProfileFromString(data as string) as LCMS2ProfileMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens an ICC profile which is entirely contained in a string.

**Notes:**

Context: Optional, the context object.

Data: The profile data.

Useful for accessing embedded profiles. This buffer must hold a full profile image. Memory must be contiguous.

Returns an ICC profile object on success, nil on error.

See also:

- 2.14.1 `OpenProfileFromString(context as variant, data as string)` as `LCMS2ProfileMBS` 91

### **PCS as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets/Sets the profile connection space used by the given profile, using the ICC convention.

**Notes:** (Read and Write computed property)

### **PostScriptCRD(context as variant, intent as UInt32, flags as UInt32 = 0) as string**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A wrapper on `cmsGetPostScriptColorResource` to simplify CRD generation.

**Notes:**

context: Optional a user-defined context cargo.

Intent: The intent code, as described in `Intents` constants.

Flags: A combination of bit-field `kcmsFLAGS*` constants.

Returns: The resource as string or an empty string on error.

**PostScriptCSA(context as variant, intent as UInt32, flags as UInt32 = 0) as string**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A wrapper on cmsGetPostScriptColorResource to simplify CSA generation.

**Notes:**

context: Optional a user-defined context cargo.

Intent: The intent code, as described in Intents constants.

Flags: A combination of bit-field kcmsFLAGS\* constants.

Returns: The resource as string or an empty string on error.

**ProfileICCversion as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get/set the profile ICC version in the same format as it is stored in the header.

**Notes:** (Read and Write computed property)

**ProfileVersion as double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets/Sets the ICC version in profile header.

**Notes:**

The version given as to this function as a float n.m is properly encoded.

(Read and Write computed property)

**ReadChromaticAdaptation as LCMS2CIEXYZMBS()**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads a chromatic adaptation.

**Notes:** On success returns an array of 3 XYZ values.

**ReadChromaticity as LCMS2CIExyYTripleMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads chromaticity tag.

**Notes:**

For `kcmsSigChromaticityTag`.  
Returns nil on error.

**ReadCIEXYZ(tag as integer) as LCMS2CIEXYZMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as CIE XYZ value.

**Notes:**

Works with `kcmsSigBlueColorantTag`, `kcmsSigBlueMatrixColumnTag`, `kcmsSigGreenColorantTag`, `kcmsSigGreenMatrixColumnTag`, `kcmsSigLuminanceTag`, `kcmsSigMediaBlackPointTag`, `kcmsSigMediaWhitePointTag`, `kcmsSigRedColorantTag` and `kcmsSigRedMatrixColumnTag`.  
Returns nil on any error.

**ReadColorantOrder as Memoryblock**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads colorant order.

**Notes:**

For `kcmsSigColorantOrderTag`.  
Returns nil on any error.

**ReadDate(tag as integer) as LCMS2DateMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as date.

**Notes:**

Works with `kcmsSigCalibrationDateTimeTag` and `kcmsSigDateTimeTag`.  
Returns nil on any error.

**ReadICCData(tag as integer) as LCMS2ICCDataMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as ICC Data.

**Notes:**

Works with kcmsSigDataTag, kcmsSigPs2CRD0Tag, kcmsSigPs2CRD1Tag, kcmsSigPs2CRD2Tag, kcmsSigPs2CRD3Tag, kcmsSigPs2CSATag and kcmsSigPs2RenderingIntentTag.

Returns nil on any error.

**ReadICCMeasurementConditions as LCMS2ICCMeasurementConditionsMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads ICC measurement conditions.

**Notes:**

for kcmsSigMeasurementTag.

Returns nil on any error.

**ReadICCViewingConditions as LCMS2ICCViewingConditionsMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as ICCViewingConditions value.

**Notes:** Works with kcmsSigViewingConditionsTag.

**ReadMLU(tag as integer) as LCMS2MLUMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as MLU.

**Notes:**

Works with kcmsSigCharTargetTag, kcmsSigCopyrightTag, kcmsSigDeviceMfgDescTag, kcmsSigDeviceModelDescTag, kcmsSigProfileDescriptionTag, kcmsSigScreeningDescTag and kcmsSigViewingCondDescTag.

Returns nil on any error.

**ReadNamedColorList(tag as integer) as LCMS2NamedColorListMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as named color list.

**Notes:**

Works with `kcmsSigColorantTableTag`, `kcmsSigColorantTableOutTag`, `kcmsSigCrdInfoTag` and `kcmsSigNamedColor2Tag`.

Returns nil on any error.

**ReadPipeline(tag as integer) as LCMS2PipelineMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as pipeline.

**Notes:**

Works with `kcmsSigAtoB0Tag`, `kcmsSigAtoB1Tag`, `kcmsSigAtoB2Tag`, `kcmsSigBtoA0Tag`, `kcmsSigBtoA1Tag`, `kcmsSigBtoA2Tag`, `kcmsSigDtoB0Tag`, `kcmsSigDtoB1Tag`, `kcmsSigDtoB2Tag`, `kcmsSigDtoB3Tag`, `kcmsSigBtoD0Tag`, `kcmsSigBtoD1Tag`, `kcmsSigBtoD2Tag`, `kcmsSigBtoD3Tag`, `kcmsSigGamutTag`, `kcmsSigPreview0Tag`, `kcmsSigPreview1Tag` and `kcmsSigPreview2Tag`.

Returns nil on any error.

**ReadRawTag(sig as integer) as Memoryblock**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads raw tag.

**Notes:**

Similar to `ReadTag*`, but different in two important aspects. The important point is, this is raw data. No processing is performed, so you can effectively read wrong or broken profiles with this function. Obviously, then you have to interpret all those bytes!

sig: Signature of tag to be read

Returns memoryblock with data or nil on any error.

Those functions allows to read/write directly to the ICC profile any data, without checking anything. As a rule, mixing Raw with cooked doesn't work, so writting a tag as raw and then reading it as cooked without

serializing does result into an error. If that is what you want, you will need to dump the profile to memory or disk and then reopen it.

Returns nil on any error.

### **ReadScreening as LCMS2ScreeningMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as screening.

**Notes:**

Works with `kcmsSigScreeningTag`.

Returns nil on any error.

### **ReadSequence(tag as integer) as LCMS2SequenceMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as sequence.

**Notes:**

Only for `kcmsSigProfileSequenceDescTag` and `kcmsSigProfileSequenceIdTag`.

Returns nil on any error.

### **ReadSignature(tag as integer) as UInt32**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads a signature tag.

**Notes:** Works with `kcmsSigColorimetricIntentImageStateTag`, `kcmsSigPerceptualRenderingIntentGamutTag`, `kcmsSigSaturationRenderingIntentGamutTag` or `kcmsSigTechnologyTag`.

### **ReadTag(tag as integer) as variant**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads a tag.

**Notes:**

This is a convenience function which gives you the tag in whatever class the plugin thing is suitable. Check with `isa` what class you get. Returns nil on any error.

### **ReadToneCurve(tag as integer) as LCMS2ToneCurveMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads tag as ToneCurve.

**Notes:** Works with `kcmsSigBlueTRCTag`, `kcmsSigGrayTRCTag`, `kcmsSigGreenTRCTag` and `kcmsSigRedTRCTag`.

### **ReadUcrBg as LCMS2UcrBgMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads UcrBg tag.

### **RenderingIntent as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets/Sets the profile header rendering intent.

**Notes:**

From the ICC spec: The rendering intent field shall specify the rendering intent which should be used (or, in the case of a DeviceLink profile, was used) when this profile is (was) combined with another profile. In a sequence of more than two profiles, it applies to the combination of this profile and the next profile in the sequence and not to the entire sequence. Typically, the user or application will set the rendering intent dynamically at runtime or embedding time. Therefore, this flag may not have any meaning until the profile is used in some context, e.g. in a DeviceLink or an embedded source profile.  
(Read and Write computed property)

### **SaveProfileToFile(file as folderitem) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Saves the contents of a profile to a given file.

**Notes:** Returns true on success and false on failure.

### SaveProfileToString as Memoryblock

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Saves the contents of a profile to a memoryblock.

**Notes:** Returns memoryblock on success and nil on failure.

See also:

- 2.14.1 SaveProfileToString as string 99

### SaveProfileToString as string

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Saves the contents of a profile to a string.

**Notes:** Returns string with profile data on success and "" on failure.

See also:

- 2.14.1 SaveProfileToString as Memoryblock 99

### TagCount as Integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns number of tags of a given profile.

**Notes:** Returns number of tags on success, -1 on error.

### TagLinkedTo(sig as integer) as integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the tag linked to sig, in the case two tags are sharing same resource, or nil if the tag is not linked to any other tag.

**Notes:**

sig: Signature of linking tag.

Returns signature of linked tag, or 0 if no tag is linked.

**TagSignature(index as integer) as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the signature of a tag located in n position being n a 0-based index: i.e., first tag is indexed with n=0.

**Notes:**

index: index to a tag position (0-based)

Returns the tag signature on success, 0 on error.

**WriteChromaticAdaptation(value as LCMS2Mat3MBS) as boolean**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes chromatic adaptation.

**Notes:** Variant of the function which takes matrix of values.

See also:

- 2.14.1 WriteChromaticAdaptation(values() as LCMS2CIEXYZMBS) as boolean 100

**WriteChromaticAdaptation(values() as LCMS2CIEXYZMBS) as boolean**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes chromatic adaptation.

**Notes:**

Pass array with 3 XYZ colors.

Returns true on success.

See also:

- 2.14.1 WriteChromaticAdaptation(value as LCMS2Mat3MBS) as boolean 100

**WriteChromaticity(o as LCMS2CIExyYTripleMBS) as boolean**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes chromaticity tag.

**Notes:**

For kcmsSigChromaticityTag.  
Returns true on success and false on error.

**WriteCIEXYZ(tag as integer, o as LCMS2CIEXYZMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes CIE XYZ tag.

**Notes:** Works with kcmsSigBlueColorantTag, kcmsSigBlueMatrixColumnTag, kcmsSigGreenColorantTag, kcmsSigGreenMatrixColumnTag, kcmsSigLuminanceTag, kcmsSigMediaBlackPointTag, kcmsSigMediaWhitePointTag, kcmsSigRedColorantTag and kcmsSigRedMatrixColumnTag.

**WriteColorantOrder(data as Memoryblock) as boolean**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes colorant order.

**Notes:**

For kcmsSigColorantOrderTag.  
Data should be 16 byte long.  
Returns true on success or false on failure.

**WriteDate(tag as integer, o as LCMS2DateMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes date tag.

**Notes:** Works with kcmsSigCalibrationDateTimeTag and kcmsSigDateTimeTag.

**WriteICCDData(tag as integer, o as LCMS2ICCDDataMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes ICC Data tag.

**Notes:** Works with kcmsSigDataTag, kcmsSigPs2CRD0Tag, kcmsSigPs2CRD1Tag, kcmsSigPs2CRD2Tag, kcmsSigPs2CRD3Tag, kcmsSigPs2CSATag and kcmsSigPs2RenderingIntentTag.

**WriteICCMeasurementConditions(value as LCMS2ICCMeasurementConditionsMBS) as boolean**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes IIC measurement conditions.

**Notes:**

for kcmsSigMeasurementTag.

Writes data and returns true on success or false on failure.

**WriteICCViewingConditions(o as LCMS2ICCViewingConditionsMBS) as boolean**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes ICC Viewing conditions tag.

**Notes:** Works with kcmsSigViewingConditionsTag.

**WriteMLU(tag as integer, o as LCMS2MLUMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes MLU tag.

**Notes:** Works with kcmsSigCharTargetTag, kcmsSigCopyrightTag, kcmsSigDeviceMfgDescTag, kcmsSigDeviceModelDescTag, kcmsSigProfileDescriptionTag, kcmsSigScreeningDescTag and kcmsSigViewingCondDescTag.

**WriteNamedColorList(tag as integer, o as LCMS2NamedColorListMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes named color list tag.

**Notes:** Works with kcmsSigColorantTableTag, kcmsSigColorantTableOutTag, kcmsSigCrdInfoTag and kcmsSigNamedColor2Tag.

**WritePipeline(tag as integer, o as LCMS2PipelineMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes pipeline tag.

**Notes:** Works with kcmsSigAtoB0Tag, kcmsSigAtoB1Tag, kcmsSigAtoB2Tag, kcmsSigBtoA0Tag, kcmsSigBtoA1Tag, kcmsSigBtoA2Tag, kcmsSigDtoB0Tag, kcmsSigDtoB1Tag, kcmsSigDtoB2Tag, kcmsSigDtoB3Tag,

kcmsSigBToD0Tag, kcmsSigBToD1Tag, kcmsSigBToD2Tag, kcmsSigBToD3Tag, kcmsSigGamutTag, kcmsSigPreview0Tag, kcmsSigPreview1Tag and kcmsSigPreview2Tag.

### **WriteRawTag(sig as integer, data as Memoryblock) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes raw tag.

#### **Notes:**

The RAW version does the same as WriteTag\* but without any interpretation of the data. Please note it is fair easy to deal with cooked structures, since there are primitives for allocating, deleting and modifying data. For RAW data you are responsible of everything. If you want to deal with a private tag, you may want to write a plug-in instead of messing up with raw data.

sig: Signature of tag to be written

data: memory block holding the data.

Returns true on success, false on error

Those functions allows to read/write directly to the ICC profile any data, without checking anything. As a rule, mixing Raw with cooked doesn't work, so writting a tag as raw and then reading it as cooked without serializing does result into an error. If that is wha you want, you will need to dump the profile to memory or disk and then reopen it.

### **WriteScreening(o as LCMS2ScreeningMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes screening tag.

**Notes:** Works with kcmsSigScreeningTag.

### **WriteSequence(tag as integer, o as LCMS2SequenceMBS) as boolean**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes sequence tag.

**Notes:** Only for kcmsSigProfileSequenceDescTag and kcmsSigProfileSequenceIdTag.

**WriteSignature(tag as integer, o as UInt32) as boolean**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes a signature tag with the given values.

**Notes:** Works with kcmsSigColorimetricIntentImageStateTag, kcmsSigPerceptualRenderingIntentGamutTag, kcmsSigSaturationRenderingIntentGamutTag or kcmsSigTechnologyTag.

**WriteToneCurve(tag as integer, o as LCMS2ToneCurveMBS) as boolean**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes tone curve tag.

**Example:**

```
// read a profile
dim f as FolderItem = SpecialFolder.Desktop.Child("myprofile.icc")
dim p as LCMS2ProfileMBS = LCMS2ProfileMBS.OpenProfileFromFile(f)

// find gray level tone curve
dim t as LCMS2ToneCurveMBS = p.ReadToneCurve(LCMS2MBS.kcmsSigGrayTRCTag)
if t<>nil then

// let's make a new one with half of old values
dim values(1000) as single

for i as integer = 0 to 1000
values(i) = t.EvalToneCurveFloat(i/1000.0) * 0.5
next

// build new curve with that values
dim n as LCMS2ToneCurveMBS = LCMS2ToneCurveMBS.BuildTabulatedToneCurve(nil, values)

// write back
if not p.WriteToneCurve(LCMS2MBS.kcmsSigGrayTRCTag, n) then
MsgBox "failed to write tone curve"
end if
end if

// write profile
f = SpecialFolder.Desktop.Child("test.icc")
call p.SaveProfileToFile(f)
```

**Notes:** Works with kcmsSigBlueTRCTag, kcmsSigGrayTRCTag, kcmsSigGreenTRCTag and kcmsSigRedTRC-

Tag.

### WriteUcrBg(o as LCMS2UcrBgMBS) as boolean

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes UcrBg tag.

## 2.14.2 Properties

### Context as variant

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The context for this profile.

**Notes:**

Error handling uses it, so you can see which part of your application failed.  
(Read only property)

### File as Folderitem

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file reference.

**Notes:**

Only set for file based profiles, so you can later know what file you used to create the profile object.  
(Read and Write property)

### Handle as Ptr

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Internal object reference.

**Notes:** (Read and Write property)

### 2.14.3 Constants

#### **kInfoCopyright = 3**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the info selectors for GetProfileInfo.

**Notes:** The copyright string.

#### **kInfoDescription = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the info selectors for GetProfileInfo.

**Notes:** The description string.

#### **kInfoManufacturer = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the info selectors for GetProfileInfo.

**Notes:** The manufacturer string.

#### **kInfoModel = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the info selectors for GetProfileInfo.

**Notes:** The model string.

## 2.15 class LCMS2ScreeningChannelMBS

### **class LCMS2ScreeningChannelMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The screening information for one channel.

### 2.15.1 Methods

#### Clone as LCMS2ScreeningChannelMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the screening channel object.

#### Constructor(Frequency as double = 0.0, ScreenAngle as double = 0.0, SpotShape as UInt32 = 0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The constructor.  
See also:

- 2.15.1 Constructor(other as LCMS2ScreeningChannelMBS) 107

#### Constructor(other as LCMS2ScreeningChannelMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes object with values from other object.

See also:

- 2.15.1 Constructor(Frequency as double = 0.0, ScreenAngle as double = 0.0, SpotShape as UInt32 = 0) 107

### 2.15.2 Properties

#### Frequency as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The frequency.  
**Notes:** (Read and Write property)

**ScreenAngle as Double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The screen angle.  
**Notes:** (Read and Write property)

**SpotShape as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The spot shape.  
**Notes:**

See kcmsSpot\* constants.  
 (Read and Write property)

**2.16 module LCMS2MBS****module LCMS2MBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The central LCMS module with all the global methods and constants.

**2.16.1 Methods**

**AdaptationMatrix(ConeMatrix as LCMS2Mat3MBS, FromIll as LCMS2CIEXYZMBS, ToIll as LCMS2CIEXYZMBS) as LCMS2Mat3MBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates adaptation matrix.  
**Notes:**

Returns the final chromatic adaptation from illuminant FromIll to Illuminant ToIll. The cone matrix can be specified in ConeMatrix. If nil, Bradford is assumed.

ConeMatrix: the cone matrix.  
 FromIll: Source illuminant.

ToIll: Destination illuminant.

Returns matrix on success or nil on failure.

**AdaptToIlluminant(SourceWhitePt as LCMS2CIEXYZMBS, Illuminant as LCMS2CIEXYZMBS, Value as LCMS2CIEXYZMBS) as LCMS2CIEXYZMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Chromatic adaptation.

**Notes:** Adapts a color to a given illuminant. Original color is expected to have a SourceWhitePt white point.

**BFDdeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS) as double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Computes the dE between two Lab values.

**BuildRGB2XYZtransferMatrix(WhitePoint as LCMS2CIExyYMBS, Primaries as LCMS2CIExyYTripleMBS as LCMS2Mat3MBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Builds RGB to XYZ transfer matrix.

**Notes:**

Build a White point, primary chromas transfer matrix from RGB to CIE XYZ.

This is just an approximation, I am not handling all the non-linear aspects of the RGB to XYZ process, and assuming that the gamma correction has transitive property in the tranformation chain.

The alghoritm:

\* First I build the absolute conversion matrix using primaries in XYZ. This matrix is next inverted  
Then I eval the source white point across this matrix obtaining the coeficients of the transformation  
Then, I apply these coeficients to the original matrix

WhitePoint: The white point.

Primaries: The primaries.  
Returns matrix on success or nil on failure.

### **BYTES\_SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.  
**Notes:** bytes per sample

### **ChannelsOf(ColorSpaceSignature as integer) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the number of channels in a color space signature.  
**Example:**

```
MsgBox "RGB has " + str(LCMS2MBS.ChannelsOf(LCMS2MBS.kcmsSigRgbData)) + " channels"
```

### **CHANNELS\_SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.  
**Notes:** Channels (Samples per pixel)

### **CIE2000DeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS, K1 as double, Kc as double, Kh as double) as double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the Delta-E 2000.  
**Notes:** Delta-E 2000 is the first major revision of the dE94 equation. Unlike dE94, which assumes that L\* correctly reflects the perceived differences in lightness, dE2000 varies the weighting of L\* depending on where in the lightness range the color falls. dE2000 is still under consideration and does not seem to be widely supported in graphics arts applications.

**CIE94DeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS) as double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the CIE 94 delta e value.

**Notes:**

A technical committee of the CIE (TC1-29) published an equation in 1995 called CIE94. The equation is similar to CMC but the weighting functions are largely based on RIT/DuPont tolerance data derived from automotive paint experiments where sample surfaces are smooth.

It also has ratios, labeled kL (lightness) and Kc (chroma) and the commercial factor (cf) but these tend to be preset in software and are not often exposed for the user (as it is the case in Little CMS).

**CMCdeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS, l as double, c as double) as double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the CMC delta E.

**Notes:**

In 1984 the CMC (Colour Measurement Committee of the Society of Dyes and Colourists of Great Britain) developed and adopted an equation based on LCH numbers. Intended for the textiles industry, CMC l:c allows the setting of lightness (l) and chroma (c) factors. As the eye is more sensitive to chroma, the default ratio for l:c is 2:1 allowing for 2x the difference in lightness than chroma (numbers). There is also a 'commercial factor' (cf) which allows an overall varying of the size of the tolerance region according to accuracy requirements. A cf=1.0 means that a delta-E CMC value <1.0 is acceptable.

CMC l:c is designed to be used with D65 and the CIE Supplementary Observer. Commonly-used values for l:c are 2:1 for acceptability and 1:1 for the threshold of imperceptibility.

**COLORSPACE\_ SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** Pixeltype.

**CreateBitmapFromPicture(p as picture, bits as integer = 8) as LCMS2BitmapMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new bitmap based on the pictures.

**Notes:** Bits can be 8, 16 or 32 bit integers.

### D50\_ xyY as LCMS2CIExyYMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns D50 white point as xyY color.

### D50\_ XYZ as LCMS2CIEXYZMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns D50 white point as XYZ color.

### DeltaE(Lab1 as LCMS2CIELabMBS, Lab2 as LCMS2CIELabMBS) as double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates delta E.

**Notes:**

You don't have to spend too long in the color management world before you come across the term Delta-E. As with many things color, it seems simple to understand at first, yet the closer you look, the more elusive it gets. Delta-E (dE) is a single number that represents the 'distance' between two colors. The idea is that a dE of 1.0 is the smallest color difference the human eye can see. So any dE less than 1.0 is imperceptible and it stands to reason that any dE greater than 1.0 is noticeable. Unfortunately it's not that simple. Some color differences greater than 1 are perfectly acceptable, maybe even unnoticeable. Also, the same dE color difference between two yellows and two blues may not look like the same difference to the eye and there are other places where it can fall down. It's perfectly understandable that we would want to have a system to show errors. After all, we've spent the money on the instruments; shouldn't we get numbers from them? Delta-E numbers can be used for:

- how far off is a print or proof from the original

- how much has a device drifted

- how effective is a particular profile for printing or proofing

- removes subjectivity (as much as possible)

These functions does compute the difference between two Lab colors, using several difference spaces.

The L\*a\*b\* color space was devised in 1976 and, at the same time delta-E 1976 (dE76) came into being. If you can imagine attaching a string to a color point in 3D Lab space, dE76 describes the sphere that is

described by all the possible directions you could pull the string. If you hear people speak of just plain 'delta-E' they are probably referring to dE76. It is also known as dE-Lab and dE- ab. One problem with dE76 is that Lab itself is not 'perceptually uniform' as its creators had intended. So different amounts of visual color shift in different color areas of Lab might have the same dE76 number. Conversely, the same amount of color shift might result in different dE76 values. Another issue is that the eye is most sensitive to hue differences, then chroma and finally lightness and dE76 does not take this into account.

### **DOSWAP\_ SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** Do swap? ie, BGR, KYMC

### **ENDIAN16\_ SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** swap 16 bps endianness?

### **EXTRA\_ SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** Extra samples

### **FLAVOR\_ SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** Flavor 0=MinIsBlack(Chocolate) 1=MinIsWhite(Vanilla).

**Float2LabEncoded(c as LCMS2CIELabMBS) as integer()**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Encodes a Lab value.

**Notes:** Returns array of 3 encoded UInt16 values.

**Float2LabEncodedV2(c as LCMS2CIELabMBS) as integer()**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Encodes a Lab value to ICC v2 convention.

**Notes:** Returns array of 3 encoded UInt16 values.

**Float2XYZEncoded(c as LCMS2CIEXYZMBS) as integer()**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Encodes a XYZ value to ICC convention.

**Notes:** Returns three UInt16 values as integer array.

**FLOAT\_SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** Floating point – With this flag we can differentiate 16 bits as float and as int.

**GetAlarmCodes as integer()**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets the current global codes used to mark out-out-gamut on Proofing transforms.

**Notes:**

Values are meant to be encoded in 16 bits.

Returns array with 16 integer values.

**kcmsD50X as double**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** X value of D50 XYZ normalized to Y=1.0.

**kcmsD50Y as double**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Y value of D50 XYZ normalized to Y=1.0.

**kcmsD50Z as double**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Z value of D50 XYZ normalized to Y=1.0.

**kcmsPERCEPTUAL\_ BLACK\_ X as double**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** X of V4 perceptual black.

**kcmsPERCEPTUAL\_ BLACK\_ Y as double**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Y of V4 perceptual black.

**kcmsPERCEPTUAL\_ BLACK\_ Z as double**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Z of V4 perceptual black.

**Lab2LCh(p as LCMS2CIELabMBS) as LCMS2CIELChMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts lab color to LCh value.

**Lab2XYZ(p as LCMS2CIELabMBS, whitepoint as LCMS2CIEXYZMBS = nil) as LCMS2CIEXYZMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts color to XYZ value.

**Notes:** Setting WhitePoint to NULL forces D50 as white point.

**LabEncoded2Float(w0 as UInt16, w1 as UInt16, w2 as UInt16) as LCMS2CIELabMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Decodes a Lab value, encoded on ICC v4 convention to a lab value.

**Notes:**

w0, w1 and w2: Array of 3 UInt16 holding the encoded values.

Returns lab color.

**LabEncoded2FloatV2(w0 as UInt16, w1 as UInt16, w2 as UInt16) as LCMS2CIELabMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Decodes a Lab value, encoded on ICC v2 convention to a Lab value.

**Notes:** w0, w1 and w2: 3 UInt16 numbers holding the encoded values.

**LCh2Lab(p as LCMS2CIELChMBS) as LCMS2CIELabMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colorimetric space conversion from LCh to Lab.

**NewBitmap(width as integer,height as integer, colorspace as integer) as LCMS2BitmapMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a bitmap object with given size and color space.

See also:

- 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer, RowBytes as integer) as LCMS2BitmapMBS 117
- 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer, RowBytes as integer, data as memoryblock) as LCMS2BitmapMBS 117

**NewBitmap(width as integer,height as integer, colorspace as integer, RowBytes as integer) as LCMS2BitmapMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a bitmap object with given size and color space.

See also:

- 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer) as LCMS2BitmapMBS 117
- 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer, RowBytes as integer, data as memoryblock) as LCMS2BitmapMBS 117

**NewBitmap(width as integer,height as integer, colorspace as integer, RowBytes as integer, data as memoryblock) as LCMS2BitmapMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a bitmap object with given size and color space.

**Notes:**

In this variant, you pass the memoryblock with right size.

Passing memoryblock of wrong size can lead into crashes.

See also:

- 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer) as LCMS2BitmapMBS 117
- 2.16.1 NewBitmap(width as integer,height as integer, colorspace as integer, RowBytes as integer) as LCMS2BitmapMBS 117

**OPTIMIZED\_SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** Optimized – previous optimization already returns the final 8-bit value.

**PixelFormat(FloatingPoint as boolean, Optimized as boolean, ColorSpace as UInt32, MinIsWhite as boolean, Planar as boolean, EndianSwap as boolean, DoSwap as boolean, ExtraSamples as UInt32, Channels as UInt32, BytesPerSample as UInt32, SwapFirst as boolean) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Utility function to create a pixel format.

**Notes:**

Format of pixel is defined by one UInt32, using bit fields as follows

A O TTTTT U Y F P X S EEE CCCC BBB

FloatingPoint: With this flag we can differentiate 16 bits as float and as int

Optimized: Previous optimization already returns the final 8-bit value

ColorSpace: Pixeltype

MinIsWhite: Flavor 0=MinIsBlack(Chocolate) 1=MinIsWhite(Vanilla)

Planar: Planar? 0=Chunky, 1=Planar

EndianSwap: swap 16 bps endianness?

DoSwap: Do swap? ie, BGR, KYMC

ExtraSamples: Extra samples

Channels: Channels (Samples per pixel)

BytesPerSample: bytes per sample

SwapFirst: Swap first - changes ABGR to BGRA and KCMY to CMYK

**PLANAR\_SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** Planar? 0=Chunky, 1=Planar

**SetAdaptationState(d as double) as double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets adaptation state for absolute colorimetric intent, on all but CreateExtendedTransform.

**Notes:**

Little CMS can handle incomplete adaptation states.

d: Degree on adaptation 0=Not adapted, 1=Complete adaptation, in-between=Partial adaptation. Use negative values to return the global state without changing it.

Returns previous global adaptation state.

**SetAlarmCodes(values() as integer)**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the global codes used to mark out-of-gamut on Proofing transforms.

**Notes:**

Values are meant to be encoded in 16 bits.

AlarmCodes: Array of 16 codes. All 16 values must be specified, set to zero for unused channels.

**SetLogErrorHandler(handler as LCMS2ErrorHandlerMBS)**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Allows user to set any specific logger.

**Notes:**

Each time this function is called, the previous logger is replaced. Calling this function with NULL as parameter, does reset the logger to the default Little CMS logger. The default Little CMS logger does nothing.

LCMS2ErrorHandlerMBS is an Interface. You add it to the interface of your window/thread/class. Then you get a method "Error(context as variant, ErrorCode as UInt32, Text as string)" which is called to log error messages. The context parameter is the object you pass for reference in the various context parameters to LCMS2 functions.

**SWAPFIRST\_SH(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shifts the value so you can bitwise or it with other values to get a pixel format.

**Notes:** Swap first - changes ABGR to BGRA and KCMY to CMYK

**TagInteger(tag as string) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts tag integer to string.

**Example:**

```
MsgBox hex(LCMS2MBS.TagInteger("devs"))+" "+LCMS2MBS.TagString(& h64657673)
```

**TagString(tag as UInt32) as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts tag string to integer.

**Example:**

```
MsgBox hex(LCMS2MBS.TagInteger("devs"))+" "+LCMS2MBS.TagString(& h64657673)
```

**TempFromWhitePoint(TempK as LCMS2CIExyYMBS) as double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Correlates a black body temperature in K from given chromaticity.

**Example:**

```
dim c as new LCMS2CIExyYMBS
```

```
c.x = 0.32
```

```
c.y = 0.32
```

```
c.YY = 1.0
```

```
MsgBox str(LCMS2MBS.TempFromWhitePoint(c))
```

**Notes:** Returns temperature. Or zero on any error.

### **T\_ BYTES(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** bytes per sample

### **T\_ CHANNELS(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Channels (Samples per pixel)

### **T\_ COLORSPACE(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Pixeltype

### **T\_ DOSWAP(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Do swap? ie, BGR, KYMC

**T\_ ENDIAN16(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** swap 16 bps endianness?

**T\_ EXTRA(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Extra samples

**T\_ FLAVOR(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Flavor 0=MinIsBlack(Chocolate) 1=MinIsWhite(Vanilla)

**T\_ FLOAT(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Floating point – With this flag we can differentiate 16 bits as float and as int

**T\_ OPTIMIZED(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Optimized – previous optimization already returns the final 8-bit value

**T\_ PLANAR(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Planar? 0=Chunky, 1=Planar

**T\_ SWAPFIRST(n as UInt32) as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unpacks a value from a bitwise pixel format.

**Notes:** Swap first - changes ABGR to BGRA and KCMY to CMYK.

**Version as string**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns version number of LCMS2.

**Example:**

```
MsgBox LCMS2MBS.Version
```

**WhitePointFromTemp(TempK as double) as LCMS2CIExyYMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Correlates a black body chromaticity from given temperature in K.

**Example:**

```
dim c as LCMS2CIExyYMBS = LCMS2MBS.WhitePointFromTemp(6500)
```

```
MsgBox str(c.x)+" "+str(c.y)+" "+str(c.yy)
```

**Notes:**

Valid range is 4000K-25000K.

TempK: Temperature in oK

Returns white point or nil in case of error.

**xyY2XYZ(p as LCMS2CIExyYMBS) as LCMS2CIEXYZMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colorimetric space conversion from xyY to XYZ.

**XYZ2Lab(p as LCMS2CIEXYZMBS, whitepoint as LCMS2CIEXYZMBS = nil) as LCMS2CIELabMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts XYZ color to Lab.

**Notes:** Setting WhitePoint to nil forces D50 as white point.

**XYZ2xyY(p as LCMS2CIEXYZMBS) as LCMS2CIExyYMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colorimetric space conversion from XYZ to xyY.

**XYZEncoded2Float(w0 as UInt16, w1 as UInt16, w2 as UInt16) as LCMS2CIEXYZMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Decodes a XYZ value, encoded on ICC convention to a LCMS2CIEXYZMBS value.

**Notes:**

w0,w1 and w2: Array of 3 UInt16 values holding the encoded values.

Returns XYZ color on success or nil on failure.

## 2.16.2 Constants

**kAVG\_SURROUND = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the surround constants for viewing conditions.

**kcmsEmbeddedProfileFalse = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the header flag constants.

**kcmsEmbeddedProfileTrue = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the header flag constants.

**kcmsERROR\_ALREADY\_DEFINED = & h0000000A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_BAD\_SIGNATURE = & h0000000B**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_COLORSPACE\_CHECK = 9**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ CORRUPTION\_ DETECTED = & h0000000C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ FILE = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ INTERNAL = 3**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ NOT\_ SUITABLE = & h0000000D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ NULL = 4**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ RANGE = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ READ = 5**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ SEEK = 6**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ UNDEFINED = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ UNKNOWN\_ EXTENSION = 8**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsERROR\_ WRITE = 7**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the error constants.

**kcmsFLAGS\_ 8BITS\_ DEVICELINK = 8**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Create 8 bits devicelinks

**kcmsFLAGS\_ BLACKPOINTCOMPENSATION = & h00002000**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Black point compensation.

**kcmsFLAGS\_ CLUT\_ POST\_ LINEARIZATION = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Create postlinearization tables if possible

**kcmsFLAGS\_ CLUT\_ PRE\_ LINEARIZATION = & h00000010**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Create prelinearization tables if possible

**kcmsFLAGS\_ FORCE\_ CLUT = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Force CLUT optimization

**kcmsFLAGS\_ GAMUTCHECK = & h00001000**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Out of Gamut alarm

**kcmsFLAGS\_ GUESSDEVICECLASS = & h00000020**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Guess device class (for transform2devicelink)

**kcmsFLAGS\_ HIGHRESPRECALC = & h00000400**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Use more memory to give better accuracy

**kcmsFLAGS\_ KEEP\_ SEQUENCE = & h00000080**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Keep profile sequence for devicelink creation

**kcmsFLAGS\_ LOWRESPRECALC = & h00000800**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Use less memory to minimize resources

**kcmsFLAGS\_ NOCACHE = & h00000040**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Inhibit 1-pixel cache

**kcmsFLAGS\_ NODEFAULTRESOURCEDEF = & h01000000**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**kcmsFLAGS\_ NOOPTIMIZE = & h00000100**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Inhibit optimizations

**kcmsFLAGS\_ NOWHITEONWHITEFIXUP = 4**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Don't fix scum dot

**kcmsFLAGS\_ NULLTRANSFORM = & h00000200**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Don't transform anyway

**kcmsFLAGS\_ SOFTPROOFING = & h00004000**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the flag constants.

**Notes:** Do softproofing

**kcmsFREQUENCY\_UNITS\_LINES\_CM = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the screening flag constants.

**kcmsFREQUENCY\_UNITS\_LINES\_INCH = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the screening flag constants.

**kcmsGlossy = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the device attribute constants.

**kcmsILLUMINANT\_TYPE\_A = 6**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** A

**kcmsILLUMINANT\_TYPE\_D50 = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** D50

**kcmsILLUMINANT\_TYPE\_D55 = 5**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** D55

**kcmsILLUMINANT\_ TYPE\_ D65 = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** D65

**kcmsILLUMINANT\_ TYPE\_ D93 = 3**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** D93

**kcmsILLUMINANT\_ TYPE\_ E = 7**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** E

**kcmsILLUMINANT\_ TYPE\_ F2 = 4**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** F2

**kcmsILLUMINANT\_ TYPE\_ F8 = 8**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** F8

**kcmsILLUMINANT\_ TYPE\_ UNKNOWN = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the illuminant constants.

**Notes:** Unknown

**kcmsMagicNumber = & h61637370**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Magic number to identify an ICC profile.

**kcmsMatte = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the device attribute constants.

**kcmsMAXCHANNELS = & h00000010**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Maximum number of channels in ICC profiles.

**kcmsPRINTER\_ DEFAULT\_ SCREENS = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the screening flag constants.

**kcmsReflective = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the device attribute constants.

**kcmsSig10colorData = & h41434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig11colorData = & h42434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig12colorData = & h43434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig13colorData = & h44434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig14colorData = & h45434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig15colorData = & h46434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig1colorData = & h31434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig2colorData = & h32434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig3colorData = & h33434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig4colorData = & h34434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig5colorData = & h35434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig6colorData = & h36434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSig7colorData = & h37434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSig8colorData = & h38434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSig9colorData = & h39434C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigAbstractClass = & h61627374**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the profile class signature constants.

**Notes:** Abstract

**kcmsSigAMDisplay = & h414D4420**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigAToB0Tag = & h41324230**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigAToB1Tag = & h41324231**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigAToB2Tag = & h41324232**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBAcsElemType = & h62414353**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigBlueColorantTag = & h6258595A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBlueMatrixColumnTag = & h6258595A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBlueTRCTag = & h62545243**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBToA0Tag = & h42324130**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBToA1Tag = & h42324131**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBToA2Tag = & h42324132**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBToD0Tag = & h42324430**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBToD1Tag = & h42324431**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBToD2Tag = & h42324432**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigBToD3Tag = & h42324433**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigCalibrationDateTimeTag = & h63616C74**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigCharTargetTag = & h74617267**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigChromaticAdaptationTag = & h63686164**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigChromaticityTag = & h6368726D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigChromaticityType = & h6368726D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigCLutElemType = & h636C7574**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigCmyData = & h434D5920**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space signature constants.

**kcmsSigCmykData = & h434D594B**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space signature constants.

**kcmsSigColorantOrderTag = & h636C726F**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigColorantOrderType = & h636C726F**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigColorantTableOutTag = & h636C6F74**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigColorantTableTag = & h636C7274**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigColorantTableType = & h636C7274**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigColorimetricIntentImageStateTag = & h63696973**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigColorSpaceClass = & h73706163**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the profile class signature constants.

**Notes:** Colorspace

**kcmsSigCopyrightTag = & h63707274**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigCrdInfoTag = & h63726469**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigCrdInfoType = & h63726469**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigCRTDisplay = & h43525420**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigCurveSetElemType = & h63767374**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigCurveType = & h63757276**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigDataTag = & h64617461**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDataType = & h64617461**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigDateTimeTag = & h6474696D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDateTimeType = & h6474696D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigDeviceMfgDescTag = & h646D6E64**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDeviceModelDescTag = & h646D6464**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDeviceSettingsTag = & h64657673**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDeviceSettingsType = & h64657673**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigDictType = & h64696374**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigDigitalCamera = & h6463616D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigDigitalCinemaProjector = & h64636A70**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigDigitalMotionPictureCamera = & h646D7063**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigDisplayClass = & h6D6E7472**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the profile class signature constants.

**Notes:** Display

**kcmsSigDN = & h444E2020**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** DIN E: DIN 16536-2 densitometer response, with no polarising filter.

**kcmsSigDNN = & h444E4E20**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** DIN I: DIN 16536-2 narrow band densitometer response, with no polarising filter.

**kcmsSigDNNP = & h444E4E50**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** DIN I: DIN 16536-2 narrow band densitometer response, with polarising filter.

**kcmsSigDNP = & h444E2050**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** DIN E: DIN 16536-2 densitometer response, with polarising filter.

**kcmsSigDToB0Tag = & h44324230**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDToB1Tag = & h44324231**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDToB2Tag = & h44324232**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDToB3Tag = & h44324233**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigDyeSublimationPrinter = & h64737562**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigEAcsElemType = & h65414353**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigElectrophotographicPrinter = & h6570686F**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigElectrostaticPrinter = & h65737461**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigFilmScanner = & h6673636E**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigFilmWriter = & h6670726E**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigFlexography = & h666C6578**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigFocalPlaneColorimetryEstimates = & h66706365**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Colorimetric Intent Image State Tag signature constants.

**kcmsSigFormulaCurveSeg = & h70617266**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the curve element type signature constants.

**kcmsSigGamutTag = & h67616D74**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigGravure = & h67726176**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigGrayData = & h47524159**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigGrayTRCTag = & h6B545243**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigGreenColorantTag = & h6758595A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigGreenMatrixColumnTag = & h6758595A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigGreenTRCTag = & h67545243**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigHlsData = & h484C5320**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSigHsvData = & h48535620**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSigIdentityElemType = & h69646E20**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigInkJetPrinter = & h696A6574**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigInputClass = & h73636E72**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the profile class signature constants.

**Notes:** Input

**kcmsSigLab2XYZElemType = & h78326C20**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigLabData = & h4C616220**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigLabV2toV4 = & h32203420**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigLabV4toV2 = & h34203220**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigLinkClass = & h6C696E6B**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the profile class signature constants.

**Notes:** Link

**kcmsSigLuminanceTag = & h6C756D69**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigLut16Type = & h6D667432**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigLut8Type = & h6D667431**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigLutAtoBType = & h6D414220**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigLutBtoAType = & h6D424120**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigLuvData = & h4C757620**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigLuvKData = & h4C75764B**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigMacintosh = & h4150504C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the platform signature constants.

**Notes:** Mac

**kcmsSigMatrixElemType = & h6D617466**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigMCH1Data = & h4D434831**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigMCH2Data = & h4D434832**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCH3Data = & h4D434833**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCH4Data = & h4D434834**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCH5Data = & h4D434835**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCH6Data = & h4D434836**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCH7Data = & h4D434837**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCH8Data = & h4D434838**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCH9Data = & h4D434839**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCHADData = & h4D434841**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCHBData = & h4D434842**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCHCData = & h4D434843**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCHDDData = & h4D434844**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colospace signature constants.

**kcmsSigMCHEData = & h4D434845**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSigMCHFData = & h4D434846**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSigMeasurementTag = & h6D656173**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigMeasurementType = & h6D656173**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigMediaBlackPointTag = & h626B7074**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigMediaWhitePointTag = & h77747074**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigMetaTag = & h6D657461**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigMicrosoft = & h4D534654**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the platform signature constants.

**Notes:** Windows

**kcmsSigMotionPictureFilmRecorder = & h6D706672**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigMotionPictureFilmScanner = & h6D706673**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigMultiLocalizedUnicodeType = & h6D6C7563**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigMultiProcessElementType = & h6D706574**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigNamedColor2Tag = & h6E636C32**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigNamedColor2Type = & h6E636C32**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigNamedColorClass = & h6E6D636C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the profile class signature constants.

**Notes:** Named Colors

**kcmsSigNamedColorElemType = & h6E636C20**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigNamedColorTag = & h6E636F6C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigNamedColorType = & h6E636F6C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigNamedData = & h6E6D636C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigOffsetLithography = & h6F666673**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigOutputClass = & h70727472**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the profile class signature constants.

**Notes:** Output

**kcmsSigOutputResponseTag = & h72657370**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigParametricCurveType = & h70617261**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigPerceptualReferenceMediumGamut = & h70726D67**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the reference gammut signature constants.

**kcmsSigPerceptualRenderingIntentGamutTag = & h72696730**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigPhotoCD = & h4B504344**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigPhotographicPaperPrinter = & h7270686F**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigPhotoImageSetter = & h696D6773**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigPMDisplay = & h504D4420**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigPreview0Tag = & h70726530**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigPreview1Tag = & h70726531**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigPreview2Tag = & h70726532**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigProfileDescriptionTag = & h64657363**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigProfileSequenceDescTag = & h70736571**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigProfileSequenceDescType = & h70736571**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigProfileSequenceIdTag = & h70736964**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigProfileSequenceIdType = & h70736964**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigProjectionTelevision = & h706A7476**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigPs2CRD0Tag = & h70736430**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigPs2CRD1Tag = & h70736431**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigPs2CRD2Tag = & h70736432**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigPs2CRD3Tag = & h70736433**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigPs2CSATag = & h70733273**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigPs2RenderingIntentTag = & h70733269**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigRedColorantTag = & h7258595A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigRedMatrixColumnTag = & h7258595A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigRedTRCTag = & h72545243**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigReflectionHardcopyOriginalColorimetry = & h72686F63**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Colorimetric Intent Image State Tag signature constants.

**kcmsSigReflectionPrintOutputColorimetry = & h72706F63**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Colorimetric Intent Image State Tag signature constants.

**kcmsSigReflectiveScanner = & h7273636E**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigResponseCurveSet16Type = & h72637332**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigRgbData = & h52474220**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the colorspace signature constants.

**kcmsSigS15Fixed16ArrayType = & h73663332**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigSampledCurveSeg = & h73616D66**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the curve element type signature constants.

**kcmsSigSaturationRenderingIntentGamutTag = & h72696732**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigSceneAppearanceEstimates = & h73617065**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Colorimetric Intent Image State Tag signature constants.

**kcmsSigSceneColorimetryEstimates = & h73636F65**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Colorimetric Intent Image State Tag signature constants.

**kcmsSigScreeningDescTag = & h73637264**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigScreeningTag = & h7363726E**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigScreeningType = & h7363726E**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigSegmentedCurve = & h63757266**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the curve element type signature constants.

**kcmsSigSGI = & h53474920**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the platform signature constants.

**Notes:** SGI

**kcmsSigSignatureType = & h73696720**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigSilkscreen = & h73696C6B**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigSolaris = & h53554E57**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the platform signature constants.

**Notes:** Solaris

**kcmsSigStatusA = & h53746141**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** Status A: ISO 5-3 densitometer response. This is the accepted standard for reflection densitometers

for measuring photographic colour prints.

#### **kcmsSigStatusE = & h53746145**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** Status E: ISO 5-3 densitometer response which is the accepted standard in Europe for colour reflection densitometers.

#### **kcmsSigStatusI = & h53746149**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** Status I: ISO 5-3 densitometer response commonly referred to as narrow band or interference-type response.

#### **kcmsSigStatusM = & h5374614D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** Status M: ISO 5-3 densitometer response for measuring colour negatives.

#### **kcmsSigStatusT = & h53746154**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the response curve type signature constants.

**Notes:** Status T: ISO 5-3 wide band colour reflection densitometer response which is the accepted standard in the United States for colour reflection densitometers.

#### **kcmsSigTaligent = & h54474E54**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the platform signature constants.

**Notes:** Taligent

**kcmsSigTechnologyTag = & h74656368**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigTextDescriptionType = & h64657363**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigTextType = & h74657874**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigThermalWaxPrinter = & h74776178**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigU16Fixed16ArrayType = & h75663332**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigUcrBgTag = & h62666420**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigUcrBgType = & h62666420**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigUInt16ArrayType = & h75693136**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigUInt32ArrayType = & h75693332**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigUInt64ArrayType = & h75693634**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigUInt8ArrayType = & h75693038**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigUnices = & h2A6E6978**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the platform signature constants.

**Notes:** Unix systems

**kcmsSigVcgtTag = & h76636774**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigVcgtType = & h76636774**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigVideoCamera = & h76696463**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigVideoMonitor = & h7669646D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the technology signature constants.

**kcmsSigViewingCondDescTag = & h76756564**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigViewingConditionsTag = & h76696577**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the tag signature constants.

**kcmsSigViewingConditionsType = & h76696577**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigXYZ2LabElemType = & h6C327820**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigXYZData = & h58595A20**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSigXYZType = & h58595A20**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the data type signature constants.

**kcmsSigYCbCrData = & h59436272**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSigYxyData = & h59787920**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color-space signature constants.

**kcmsSPOT\_ CROSS = 7**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the spot shape constants.

**kcmsSPOT\_ DIAMOND = 3**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the spot shape constants.

**kcmsSPOT\_ ELLIPSE = 4**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the spot shape constants.

**kcmsSPOT\_ LINE = 5**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the spot shape constants.

**kcmsSPOT\_ PRINTER\_ DEFAULT = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the spot shape constants.

**kcmsSPOT\_ROUND = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the spot shape constants.

**kcmsSPOT\_SQUARE = 6**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the spot shape constants.

**kcmsSPOT\_UNKNOWN = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the spot shape constants.

**kcmsTransparency = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the device attribute constants.

**kcmsUseAnywhere = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the header flag constants.

**kcmsUseWithEmbeddedDataOnly = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the header flag constants.

**kCUTSHEET\_SURROUND = 4**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the surround constants for viewing conditions.

**kDARK\_SURROUND = 3**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the surround constants for viewing conditions.

**kDIM\_SURROUND = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the surround constants for viewing conditions.

**kD\_CALCULATE = -1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Special value for D Value of ViewingConditions.

**kINTENT\_ABSOLUTE\_COLORIMETRIC = 3**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Absolute Colorimetric ICC Intent.

**kINTENT\_PERCEPTUAL = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Perceptual Colorimetric ICC Intent.

**kINTENT\_PRESERVE\_K\_ONLY\_PERCEPTUAL = & h0000000A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Special LCMS intent.

**kINTENT\_PRESERVE\_K\_ONLY\_RELATIVE\_COLORIMETRIC = & h0000000B**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Special LCMS intent.

**kINTENT\_PRESERVE\_K\_ONLY\_SATURATION = & h0000000C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Special LCMS intent.

**kINTENT\_PRESERVE\_K\_PLANE\_PERCEPTUAL = & h0000000D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Special LCMS intent.

**kINTENT\_PRESERVE\_K\_PLANE\_RELATIVE\_COLORIMETRIC = & h0000000E**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Special LCMS intent.

**kINTENT\_PRESERVE\_K\_PLANE\_SATURATION = & h0000000F**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Special LCMS intent.

**kINTENT\_RELATIVE\_COLORIMETRIC = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Relative Colorimetric ICC Intent.

**kINTENT\_SATURATION = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the intent constants.

**Notes:** Saturation ICC Intent.

**klcmsSignature = & h6C636D73**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Little CMS signature.

**kLCMS\_USED\_AS\_INPUT = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the used direction constants.

**kLCMS\_USED\_AS\_OUTPUT = 1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the used direction constants.

**kLCMS\_USED\_AS\_PROOF = 2**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the used direction constants.

**kPT\_ANY = 0**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Don't check colorspace

**kPT\_CMY = 5**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** CMY

**kPT\_CMYK = 6**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** CMYK

**kPT\_GRAY = 3**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Gray

**kPT\_HLS = & h0000000D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** HLS

**kPT\_HSV = & h0000000C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** HSV

**kPT\_Lab = & h0000000A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Lab

**kPT\_LabV2 = & h0000001E**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Identical to kPT\_Lab, but using the V2 old encoding

**kPT\_MCH1 = & h0000000F**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 1 channels.

**kPT\_MCH10 = & h00000018**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 10 channels.

**kPT\_MCH11 = & h00000019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 11 channels.

**kPT\_MCH12 = & h0000001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 12 channels.

**kPT\_MCH13 = & h0000001B**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 13 channels.

**kPT\_MCH14 = & h0000001C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 14 channels.

**kPT\_MCH15 = & h0000001D**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 15 channels.

**kPT\_MCH2 = & h00000010**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 2 channels.

**kPT\_MCH3 = & h00000011**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 3 channels.

**kPT\_MCH4 = & h00000012**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 4 channels.

**kPT\_MCH5 = & h00000013**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 5 channels.

**kPT\_MCH6 = & h00000014**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 6 channels.

**kPT\_MCH7 = & h00000015**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 7 channels.

**kPT\_MCH8 = & h00000016**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Multichannel with 8 channels.

**kPT\_MCH9 = & h00000017**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** noMultichannel with 9 channels.ne

**kPT\_RGB = 4**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** RGB

**kPT\_XYZ = 9**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** XYZ

**kPT\_YCbCr = 7**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** YCbCr

**kPT\_YUV = 8**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Lu'v'

**kPT\_YUVK = & h0000000B**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Lu'v'K

**kPT\_Yxy = & h0000000E**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the pixel type constants.

**Notes:** Yxy

**kTYPE\_ABGR\_16 = & h0004049A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ABGR\_16\_PLANAR = & h0004149A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ABGR\_16\_SE = & h00040C9A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ABGR\_8 = & h00040499**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ALabV2\_8 = & h001E0499**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ALab\_8 = & h000A0499**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ ARGB\_ 16 = & h0004409A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ ARGB\_ 8 = & h00044099**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ BGRA\_ 16 = & h0004449A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ BGRA\_ 16\_ SE = & h0004489A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ BGRA\_ 8 = & h00044499**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ BGR\_ 16 = & h0004041A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ BGR\_ 16\_ PLANAR = & h0004141A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ BGR\_ 16\_ SE = & h00040C1A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ BGR\_ 8 = & h00040419**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ BGR\_ 8\_ PLANAR = & h00041419**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK10\_ 16 = & h00180052**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK10\_ 16\_ SE = & h00180852**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK10\_ 8 = & h00180051**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK11\_ 16 = & h0019005A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK11\_ 16\_ SE = & h0019085A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK11\_ 8 = & h00190059**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK12\_ 16 = & h001A0062**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK12\_ 16\_ SE = & h001A0862**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK12\_ 8 = & h001A0061**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK5\_ 16 = & h0013002A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK5\_ 16\_ SE = & h0013082A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK5\_ 8 = & h00130029**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK6\_ 16 = & h00140032**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK6\_ 16\_ PLANAR = & h00141032**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK6\_ 16\_ SE = & h00140832**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK6\_ 8 = & h00140031**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK6\_ 8\_ PLANAR = & h00141031**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK7\_ 16 = & h0015003A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK7\_ 16\_ SE = & h0015083A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK7\_ 8 = & h00150039**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK8\_ 16 = & h00160042**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK8\_ 16\_ SE = & h00160842**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK8\_ 8 = & h00160041**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK9\_ 16 = & h0017004A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK9\_ 16\_ SE = & h0017084A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK9\_ 8 = & h00170049**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYKA\_ 8 = & h000600A1**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK\_ 16 = & h00060022**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK\_ 16\_ PLANAR = & h00061022**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK\_ 16\_ REV = & h00062022**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK\_ 16\_ SE = & h00060822**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK\_ 8 = & h00060021**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK\_ 8\_ PLANAR = & h00061021**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK\_ 8\_ REV = & h00062021**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMYK\_ DBL = & h00460020**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point with doubles.

**kTYPE\_ CMYK\_ FLT = & h00460024**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point.

**kTYPE\_ CMY\_ 16 = & h0005001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMY\_ 16\_ PLANAR = & h0005101A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMY\_ 16\_ SE = & h0005081A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMY\_ 8 = & h00050019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ CMY\_ 8\_ PLANAR = & h00051019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAYA\_ 16 = & h0003008A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAYA\_ 16\_ PLANAR = & h0003108A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAYA\_ 16\_ SE = & h0003088A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAYA\_ 8 = & h00030089**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAYA\_ 8\_ PLANAR = & h00031089**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAY\_ 16 = & h0003000A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAY\_ 16\_ REV = & h0003200A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAY\_ 16\_ SE = & h0003080A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ GRAY\_ 8 = & h00030009**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_GRAY\_8\_REV = & h00032009**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_GRAY\_DBL = & h00430008**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point with doubles.

**kTYPE\_GRAY\_FLT = & h0043000C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point.

**kTYPE\_HLS\_16 = & h000D001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_HLS\_16\_PLANAR = & h000D101A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_HLS\_16\_SE = & h000D081A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_HLS\_8 = & h000D0019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_HLS\_8\_PLANAR = & h000D1019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_HSV\_16 = & h000C001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_HSV\_16\_PLANAR = & h000C101A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_HSV\_16\_SE = & h000C081A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_HSV\_8 = & h000C0019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ HSV\_ 8\_ PLANAR = & h000C1019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KCMY\_ 16 = & h00064022**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KCMY\_ 16\_ REV = & h00066022**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KCMY\_ 16\_ SE = & h00064822**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KCMY\_ 8 = & h00064021**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KCMY\_ 8\_ REV = & h00066021**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC10\_ 16 = & h00180452**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC10\_ 16\_ SE = & h00180C52**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC10\_ 8 = & h00180451**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC11\_ 16 = & h0019045A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC11\_ 16\_ SE = & h00190C5A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC11\_ 8 = & h00190459**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC12\_ 16 = & h001A0462**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC12\_ 16\_ SE = & h001A0C62**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC12\_ 8 = & h001A0461**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC5\_ 16 = & h0013042A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC5\_ 16\_ SE = & h00130C2A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC5\_ 8 = & h00130429**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC7\_ 16 = & h0015043A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC7\_ 16\_ SE = & h00150C3A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC7\_ 8 = & h00150439**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC8\_ 16 = & h00160442**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC8\_ 16\_ SE = & h00160C42**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC8\_ 8 = & h00160441**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC9\_ 16 = & h0017044A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC9\_ 16\_ SE = & h00170C4A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC9\_ 8 = & h00170449**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC\_ 16 = & h00060422**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC\_ 16\_ SE = & h00060C22**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_ KYMC\_ 8 = & h00060421**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_LabA\_FLT = & h004A009C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point.

**kTYPE\_LabV2\_16 = & h001E001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_LabV2\_8 = & h001E0019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_Lab\_16 = & h000A001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_Lab\_8 = & h000A0019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_Lab\_DBL = & h004A0018**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point with doubles.

**kTYPE\_Lab\_FLT = & h004A001C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point.

**kTYPE\_NAMED\_COLOR\_INDEX = & h0000000A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGBA\_16 = & h0004009A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGBA\_16\_PLANAR = & h0004109A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGBA\_16\_SE = & h0004089A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGBA\_8 = & h00040099**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGBA\_8\_PLANAR = & h00041099**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGBA\_FLT = & h0044009C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point.

**kTYPE\_RGB\_16 = & h0004001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGB\_16\_PLANAR = & h0004101A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGB\_16\_SE = & h0004081A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGB\_8 = & h00040019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGB\_8\_PLANAR = & h00041019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_RGB\_DBL = & h00440018**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point with doubles.

**kTYPE\_RGB\_FLT = & h0044001C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point.

**kTYPE\_XYZA\_FLT = & h0049009C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point.

**kTYPE\_XYZ\_16 = & h0009001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_XYZ\_DBL = & h00490018**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point with doubles.

**kTYPE\_XYZ\_FLT = & h0049001C**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**Notes:** Floating point.

**kTYPE\_YCbCr\_16 = & h0007001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YCbCr\_16\_PLANAR = & h0007101A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YCbCr\_16\_SE = & h0007081A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YCbCr\_8 = & h00070019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YCbCr\_8\_PLANAR = & h00071019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YUVK\_16 = & h00062022**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YUVK\_8 = & h00062021**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YUV\_16 = & h0008001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YUV\_16\_PLANAR = & h0008101A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YUV\_16\_SE = & h0008081A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YUV\_8 = & h00080019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_YUV\_8\_PLANAR = & h00081019**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

**kTYPE\_Yxy\_16 = & h000E001A**

Plugin Version: 11.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the color space type constants.

## 2.17 class LCMS2Mat3MBS

**class LCMS2Mat3MBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a 3 by 3 matrix.

### 2.17.1 Methods

**Clone as LCMS2Mat3MBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the matrix.

**Constructor**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new matrix.

See also:

- 2.17.1 Constructor(other as LCMS2Mat3MBS) 207
- 2.17.1 Constructor(v0 as LCMS2Vec3MBS, v1 as LCMS2Vec3MBS, v2 as LCMS2Vec3MBS) 207

**Constructor(other as LCMS2Mat3MBS)**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a copy of the given matrix.

See also:

- 2.17.1 Constructor 207
- 2.17.1 Constructor(v0 as LCMS2Vec3MBS, v1 as LCMS2Vec3MBS, v2 as LCMS2Vec3MBS) 207

**Constructor(v0 as LCMS2Vec3MBS, v1 as LCMS2Vec3MBS, v2 as LCMS2Vec3MBS)**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new matrix with the given three vectors.

See also:

- 2.17.1 Constructor 207
- 2.17.1 Constructor(other as LCMS2Mat3MBS) 207

**value(index as UInt32) as LCMS2Vec3MBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the vector by index.

**Notes:**

Index from 0 to 2.

(Read and Write computed property)

## 2.17.2 Properties

### V0 as LCMS2Vec3MBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Property for first vector.

**Notes:** (Read and Write property)

### V1 as LCMS2Vec3MBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Property for second vector.

**Notes:** (Read and Write property)

### V2 as LCMS2Vec3MBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Property for third vector.

**Notes:** (Read and Write property)

## 2.18 class LCMS2CIExyYMBS

### class LCMS2CIExyYMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a CIE xyY value.

**Example:**

```
dim c as new LCMS2CIExyYMBS(1,2,3)
MsgBox str(c.x)+" "+str(c.y)+" "+str(c.yy)
```

**Notes:** As Real Studio is case insensitive, we have to name the big Y as YY.

## 2.18.1 Methods

### Clone as LCMS2CIExyYMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the color.

### Constructor(other as LCMS2CIExyYMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes the object with values from other object.

See also:

- 2.18.1 Constructor(X as double=0.0, Y as double=0.0, YY as double=0.0) 209

### Constructor(X as double=0.0, Y as double=0.0, YY as double=0.0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new xyY object with the given values.

See also:

- 2.18.1 Constructor(other as LCMS2CIExyYMBS) 209

### TempFromWhitePoint as double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the temperature from using this xyY point as the white point.

**Example:**

```
dim c as new LCMS2CIExyYMBS(1,2,3)
MsgBox str(c.TempFromWhitePoint)
```

**XYZ as LCMS2CIEXYZMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts the xyY value to XYZ.

**Example:**

```
dim c as new LCMS2CIExyYMBS(1,2,3)
dim n as LCMS2CIEXYZMBS = c.XYZ
MsgBox str(n.x)+" "+str(n.y)+" "+str(n.z)
dim x as LCMS2CIExyYMBS = n.xyY
MsgBox str(x.x)+" "+str(x.y)+" "+str(x.yy)
```

**2.18.2 Properties****x as Double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The x value.

**Notes:** (Read and Write property)

**y as Double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The y value.

**Notes:** (Read and Write property)

**YY as Double**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Y value.

**Notes:**

As Real Studio is case insensitive, we have to name the big Y as YY.  
(Read and Write property)

## 2.19 class LCMS2CIELChMBS

### class LCMS2CIELChMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The plugin class for CIE LCh values.

**Example:**

```
dim c as new LCMS2CIELChMBS(1,2,3)
MsgBox str(c.L)+" "+str(c.C)+" "+str(c.h)
```

### 2.19.1 Methods

#### Clone as LCMS2CIELChMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the color object.

#### Constructor(L as double=0.0, C as double=0.0, h as double=0.0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new object with the given values.

See also:

- 2.19.1 Constructor(other as LCMS2CIELChMBS) 211

#### Constructor(other as LCMS2CIELChMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes object with values from other object.

See also:

- 2.19.1 Constructor(L as double=0.0, C as double=0.0, h as double=0.0) 211

### Lab as LCMS2CIELabMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts the LCh value to a Lab value.

**Example:**

```
dim c as new LCMS2CIELChMBS(0.1, 0.2, 0.3)
dim lab as LCMS2CIELabMBS = c.Lab
MsgBox str(lab.L)+" "+str(lab.a)+" "+str(lab.b)
```

## 2.19.2 Properties

### C as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The C value.

**Notes:** (Read and Write property)

### h as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The h value.

**Notes:** (Read and Write property)

### L as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The L value.

**Notes:** (Read and Write property)

## 2.20 class LCMS2CIEXYZMBS

### class LCMS2CIEXYZMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a CIE XYZ values.

**Example:**

```
dim n as new LCMS2CIEXYZMBS(0.1, 0.2, 0.3)
MsgBox str(n.x)+" "+str(n.y)+" "+str(n.z)
```

### 2.20.1 Methods

#### Constructor(x as double=0.0, y as double=0.0, z as double=0.0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new object with the given values.

#### Lab(whitePoint as LCMS2CIEXYZMBS=nil) as LCMS2CIELabMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts the XYZ value to a lab value using the given whitepoint.

**Notes:** Setting WhitePoint to nil forces D50 as white point.

#### xyY as LCMS2CIExyYMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts the XYZ value to a xyY value.

**Example:**

```
dim n as new LCMS2CIEXYZMBS(0.1, 0.2, 0.3)
dim l as LCMS2CIExyYMBS = n.xyY
MsgBox str(l.x)+" "+str(l.y)+" "+str(l.yy)
```

## 2.20.2 Properties

### x as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The x value.  
**Notes:** (Read and Write property)

### y as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The y value.  
**Notes:** (Read and Write property)

### z as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The z value.  
**Notes:** (Read and Write property)

## 2.21 class LCMS2CIExyYTripleMBS

### class LCMS2CIExyYTripleMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a xyY color triple.

### 2.21.1 Methods

#### Clone as LCMS2CIExyYTripleMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the xyY triple.

**Constructor**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes a xyY triple with zero values.

See also:

- 2.21.1 Constructor(other as LCMS2CIExyYTripleMBS) 215
- 2.21.1 Constructor(Red as LCMS2CIExyYMBS, Green as LCMS2CIExyYMBS, Blue as LCMS2CIExyYMBS) 215

**Constructor(other as LCMS2CIExyYTripleMBS)**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes a xyY triple with values from other object.

See also:

- 2.21.1 Constructor 215
- 2.21.1 Constructor(Red as LCMS2CIExyYMBS, Green as LCMS2CIExyYMBS, Blue as LCMS2CIExyYMBS) 215

**Constructor(Red as LCMS2CIExyYMBS, Green as LCMS2CIExyYMBS, Blue as LCMS2CIExyYMBS)**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes a xyY triple with given values.

See also:

- 2.21.1 Constructor 215
- 2.21.1 Constructor(other as LCMS2CIExyYTripleMBS) 215

**2.21.2 Properties****Blue as LCMS2CIExyYMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The blue color.  
**Notes:** (Read and Write property)

**Green as LCMS2CIExyYMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The green color.  
**Notes:** (Read and Write property)

**Red as LCMS2CIExyYMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The red color.  
**Notes:** (Read and Write property)

**2.22 class LCMS2BitmapMBS****class LCMS2BitmapMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a bitmap.

**Notes:**

You can pass bitmap data as raw memoryblock or fill it using a Real Studio picture.  
 In our LCMS 1.x plugin, the bitmap was always 16 bit. This plugin also can also use 8 bit or 32 bit.

**2.22.1 Methods****Constructor**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates an empty bitmap object.

See also:

- 2.22.1 Constructor(p as picture, bits as integer = 8) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer) 217

- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer, data as memoryblock) 218

### Constructor(p as picture, bits as integer = 8)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new bitmap based on the pictures.

**Notes:** Bits can be 8, 16 or 32 bit integers.

See also:

- 2.22.1 Constructor 216
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer, data as memoryblock) 218

### Constructor(width as integer, height as integer, colorspace as integer)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a bitmap object with given size and color space.

See also:

- 2.22.1 Constructor 216
- 2.22.1 Constructor(p as picture, bits as integer = 8) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer, data as memoryblock) 218

### Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a bitmap object with given size and color space.

See also:

- 2.22.1 Constructor 216

- 2.22.1 Constructor(p as picture, bits as integer = 8) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer, data as memoryblock) 218

### Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer, data as memoryblock)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a bitmap object with given size and color space.

#### Notes:

In this variant, you pass the memoryblock with right size.  
 Passing memoryblock of wrong size can lead into crashes.  
 See also:

- 2.22.1 Constructor 216
- 2.22.1 Constructor(p as picture, bits as integer = 8) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer) 217
- 2.22.1 Constructor(width as integer, height as integer, colorspace as integer, RowBytes as integer) 217

### Invert

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Inverts the bitmap data.

**Notes:** Requires correct settings for rowbytes, height and data properties.

### Picture as picture

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a copy of the picture.

#### Example:

```
// get some picture
dim logo as Picture = LogoMBS(500)
```

```

// let us convert it to a 16 bit memory block
dim l as new LCMS2BitmapMBS(logo, 16)

// check bit depth
Title = str(l.Bits)

// convert back
dim p as Picture = l.Picture

// and display
Backdrop = p

```

**Notes:** This method works with 8, 16 and 32 bit integer pictures. Make sure bits property is set.

## 2.22.2 Properties

### Bits as Integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Number of bits.  
**Notes:**

Can be 8, 16 or 32 and is used for conversion to and from Real Studio picture objects.  
 (Read and Write property)

### ColorSpaceType as Integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The color space type.  
**Notes:** (Read and Write property)

### Data as MemoryBlock

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The memoryblock with the actual image data.  
**Notes:**

Data can be stored in 8, 16 or 32 bit Integers or 32bit Floats.  
(Read and Write property)

### Height as Integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The height of the Bitmap in Pixels.

**Notes:** (Read and Write property)

### RowBytes as Integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bytes per row.

**Notes:**

Normally:  $\text{rowbytes} = \text{pixelsize} * \text{width} + \text{padding}$

Where padding is some extra bytes.

(Read and Write property)

### Width as Integer

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The width of the Bitmap in Pixels.

**Notes:** (Read and Write property)

## 2.23 class LCMS2CIECAM02MBS

### class LCMS2CIECAM02MBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Class to evaluated viewing conditions.

**Notes:** Viewing conditions. Please note those are CAM model viewing conditions, and not the ICC tag viewing conditions, which I'm naming LCMS2ICCVueingConditionsMBS to make differences evident. Unfortunately, the tag cannot deal with surround La, Yb and D value so is basically useless to store CAM02

viewing conditions.

### 2.23.1 Methods

#### Constructor(context as variant, VC as LCMS2ViewingConditionsMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a CAM02 object based on given viewing conditions.

**Notes:**

Such object may be used as a color appearance model and evaluated in forward and reverse directions. Viewing conditions is defined by LCMS2ViewingConditionsMBS class. The surround member has to be one of this values: kAVG\_ SURROUND, kDIM\_ SURROUND, kDARK\_ SURROUND or kCUTSHEET\_ SURROUND. Degree of chromatic adaptation (d), can be specified in 0...1.0 range, or the model can be instructed to calculate it by using D\_ CALCULATE constant (-1).

Context: user-defined context cargo.

VC: A structure holding viewing conditions.

#### Forward(value as LCMS2CIEXYZMBS) as LCMS2JChMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Evaluates the CAM02 model in the forward direction: XYZ to JCh

**Notes:**

value: the input XYZ value.

Returns the output JCh value.

#### Reverse(value as LCMS2JChMBS) as LCMS2CIEXYZMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Evaluates the CAM02 model in the reverse direction: JCh to XYZ

**Notes:**

Values: The input JCh value.

Returns the output XYZ value

### 2.23.2 Properties

#### Handle as Ptr

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal handle to the conversion engine.

**Notes:** (Read and Write property)

## 2.24 class LCMS2CIELabMBS

### class LCMS2CIELabMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a Lab color.

### 2.24.1 Methods

#### BFDdeltaE(Other as LCMS2CIELabMBS) as double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates BFD delta E metric.

#### CIE2000DeltaE(Other as LCMS2CIELabMBS, K1 as double, Kc as double, Kh as double) as double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the Delta-E 2000.

**Notes:** Delta-E 2000 is the first major revision of the dE94 equation. Unlike dE94, which assumes that L\* correctly reflects the perceived differences in lightness, dE2000 varies the weighting of L\* depending on where in the lightness range the color falls. dE2000 is still under consideration and does not seem to be

widely supported in graphics arts applications.

### CIE94DeltaE(Other as LCMS2CIELabMBS) as double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the CIE 94 delta e value.

#### Notes:

A technical committee of the CIE (TC1-29) published an equation in 1995 called CIE94. The equation is similar to CMC but the weighting functions are largely based on RIT/DuPont tolerance data derived from automotive paint experiments where sample surfaces are smooth.

It also has ratios, labeled kL (lightness) and Kc (chroma) and the commercial factor (cf) but these tend to be preset in software and are not often exposed for the user (as it is the case in Little CMS).

### Clone as LCMS2CIELabMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the color object.

#### Example:

```
dim l as new LCMS2CIELabMBS(0.1, 0.2, 0.3)
dim k as LCMS2CIELabMBS = l.Clone
MsgBox str(k.L)+" " +str(k.a)+" " +str(k.b)
```

### CMCdeltaE(Other as LCMS2CIELabMBS, l as double, c as double) as double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the CMC delta E.

#### Notes:

In 1984 the CMC (Colour Measurement Committee of the Society of Dyes and Colourists of Great Britain) developed and adopted an equation based on LCH numbers. Intended for the textiles industry, CMC l:c allows the setting of lightness (l) and chroma (c) factors. As the eye is more sensitive to chroma, the default ratio for l:c is 2:1 allowing for 2x the difference in lightness than chroma (numbers). There is also a 'commercial factor' (cf) which allows an overall varying of the size of the tolerance region according to accuracy requirements. A cf=1.0 means that a delta-E CMC value <1.0 is acceptable.

CMC l:c is designed to be used with D65 and the CIE Supplementary Observer. Commonly-used values for

l:c are 2:1 for acceptability and 1:1 for the threshold of imperceptibility.

### Constructor(L as double=0.0, a as double=0.0, b as double=0.0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new Lab color with the given values.

See also:

- 2.24.1 Constructor(other as LCMS2CIELabMBS) 224

### Constructor(other as LCMS2CIELabMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a copy of the color value.

See also:

- 2.24.1 Constructor(L as double=0.0, a as double=0.0, b as double=0.0) 224

### DeltaE(Other as LCMS2CIELabMBS) as double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates delta E.

**Notes:**

You don't have to spend too long in the color management world before you come across the term Delta-E. As with many things color, it seems simple to understand at first, yet the closer you look, the more elusive it gets. Delta-E (dE) is a single number that represents the 'distance' between two colors. The idea is that a dE of 1.0 is the smallest color difference the human eye can see. So any dE less than 1.0 is imperceptible and it stands to reason that any dE greater than 1.0 is noticeable. Unfortunately it's not that simple. Some color differences greater than 1 are perfectly acceptable, maybe even unnoticeable. Also, the same dE color difference between two yellows and two blues may not look like the same difference to the eye and there are other places where it can fall down. It's perfectly understandable that we would want to have a system to show errors. After all, we've spent the money on the instruments; shouldn't we get numbers from them?

Delta-E numbers can be used for:

how far off is a print or proof from the original

how much has a device drifted

how effective is a particular profile for printing or proofing

removes subjectivity (as much as possible)

These functions does compute the difference between two Lab colors, using several difference

spaces.

The  $L^*a^*b^*$  color space was devised in 1976 and, at the same time delta-E 1976 (dE76) came into being. If you can imagine attaching a string to a color point in 3D Lab space, dE76 describes the sphere that is described by all the possible directions you could pull the string. If you hear people speak of just plain 'delta-E' they are probably referring to dE76. It is also known as dE-Lab and dE-ab. One problem with dE76 is that Lab itself is not 'perceptually uniform' as its creators had intended. So different amounts of visual color shift in different color areas of Lab might have the same dE76 number. Conversely, the same amount of color shift might result in different dE76 values. Another issue is that the eye is most sensitive to hue differences, then chroma and finally lightness and dE76 does not take this into account.

### LCh as LCMS2CIELChMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts color to LCh value.

### XYZ(whitePoint as LCMS2CIEXYZMBS=nil) as LCMS2CIEXYZMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts color to XYZ value.

**Notes:** Setting WhitePoint to NULL forces D50 as white point.

## 2.24.2 Properties

### A as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The a value.

**Notes:** (Read and Write property)

### B as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The b value.

**Notes:** (Read and Write property)

## L as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The L value.

**Notes:** (Read and Write property)

## 2.25 class LCMS2ICCDDataMBS

### class LCMS2ICCDDataMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for ICC data.

**Notes:** This is used for some tags where no dedicated class is available. This way you can modify the data directly in the memoryblock.

### 2.25.1 Properties

#### Data as Memoryblock

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The memoryblock with the data.

**Notes:** (Read only property)

#### Flags as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The flags.

**Notes:** (Read and Write property)

#### Size as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The size of the data.

**Notes:** (Read and Write property)

## 2.26 class LCMS2ICCMeasurementConditionsMBS

### class LCMS2ICCMeasurementConditionsMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for ICC measurement conditions.

### 2.26.1 Methods

**Constructor**(Observer as UInt32 = 0, Backing as LCMS2CIEXYZMBS = nil, Geometry as UInt32 = 0, Flare as double = 0.0, IlluminantType as UInt32 = 0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new object with given values.

### 2.26.2 Properties

#### Backing as LCMS2CIEXYZMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Backing value.  
**Notes:** (Read and Write property)

#### Flare as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The flare value.  
**Notes:**

Range 0..1.0.  
(Read and Write property)

**Geometry as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The geometry value.

**Notes:**

0=unknown, 1=45/0, 0/45 2=0d, d/0.  
(Read and Write property)

**IlluminantType as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The illuminant type.

**Notes:** (Read and Write property)

**Observer as UInt32**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The observer value.

**Notes:**

0 = unknown, 1=CIE 1931, 2=CIE 1964  
(Read and Write property)

**2.27 class LCMS2GamutBoundaryDescriptionMBS**

**class LCMS2GamutBoundaryDescriptionMBS**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This class allows you to create gamut boundary, add points, compute and check points.

**Example:**

```
dim n1,n2 as integer
dim h as new LCMS2GamutBoundaryDescriptionMBS
```

```
// Fill all Lab gamut as valid
```

```

for L as integer = 0 to 100 step 10
for a as integer = -128 to 128 step 5
for b as integer = -128 to 128 step 5

dim lab as new LCMS2CIELabMBS(1, a, b)

if h.AddPoint(lab) then
n1 = n1 + 1
else
break
MsgBox "Point not in boundary!"
end if
next
next
next

// Complete boundaries
call h.Compute

// All points should be inside gamut

for L as integer = 10 to 90 step 25
for a as integer = -120 to 120 step 25
for b as integer = -120 to 120 step 25

dim lab as new LCMS2CIELabMBS(1, a, b)

if h.CheckPoint(lab) then
n2 = n2 + 1
else
break
MsgBox "Point not in boundary!"
end if
next
next
next

MsgBox str(n1)+" points added and "+str(n2)+" other points found."

```

**Notes:** Please check LittleCMS API and tutorial documentation for more details.

### 2.27.1 Methods

#### AddPoint(Lab as LCMS2CIELabMBS) as Boolean

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a new sample point for computing the gamut boundary descriptor.

**Notes:**

This function can be called as many times as known points. No memory or other resources are wasted by adding new points. The gamut boundary descriptor cannot be checked until Compute() is called.

Lab: Lab value.

Returns true on success, false on error.

#### CheckPoint(Lab as LCMS2CIELabMBS) as Boolean

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Checks whatever a Lab value is inside a given gamut boundary descriptor.

**Notes:**

Lab: Lab value.

Returns: True if point is inside gamut, false otherwise.

#### Compute(options as UInt32 = 0) as Boolean

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Computes the gamut boundary descriptor using all known points and interpolating any missing sector(s).

**Notes:**

Call this function after adding all known points with AddPoint() and before using CheckPoint().

Flags: reserved (unused). Set it to 0.

Returns true on success, false on error

**Constructor(context as variant = nil)**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new object.

**2.27.2 Properties****Context as Variant**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The context for this profile.

**Notes:**

Error handling uses it, so you can see which part of your application failed.  
(Read and Write property)

**Handle as Ptr**

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The internal object reference.

**Notes:** (Read only property)

**2.28 class LCMS2JChMBS****class LCMS2JChMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a JCh value.

## 2.28.1 Methods

### Clone as LCMS2JChMBS

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of this object.

### Constructor(J as double=0.0, C as double=0.0, h as double=0.0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new JCh value.

See also:

- 2.28.1 Constructor(other as LCMS2JChMBS) 232

### Constructor(other as LCMS2JChMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes new JCh color object with values from existing object.

See also:

- 2.28.1 Constructor(J as double=0.0, C as double=0.0, h as double=0.0) 232

## 2.28.2 Properties

### C as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The C value.  
**Notes:** (Read and Write property)

### h as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The h value.  
**Notes:** (Read and Write property)

### J as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The J value.  
**Notes:** (Read and Write property)

## 2.29 class LCMS2ICCViewingConditionsMBS

### class LCMS2ICCViewingConditionsMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a ICC viewing conditions tag stored in the profile.

### 2.29.1 Methods

**Constructor**(IlluminantXYZ as LCMS2CIEXYZMBS = nil, Backing as LCMS2CIEXYZMBS = nil, IlluminantType as UInt32 = 0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new ICC viewing conditions object.

### 2.29.2 Properties

#### IlluminantType as UInt32

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The illuminant.  
**Notes:** (Read and Write property)

**IlluminantXYZ as LCMS2CIEXYZMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The illuminant.  
**Notes:** (Read and Write property)

**SurroundXYZ as LCMS2CIEXYZMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The surround.  
**Notes:** (Read and Write property)

## 2.30 class LCMS2CurveSegmentMBS

**class LCMS2CurveSegmentMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a curve segment.  
**Notes:** Segmented curves are formed by several segments.

### 2.30.1 Methods

**Constructor(nGridPoints as integer = 0)**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new object and allocates sample points.  
**Notes:** Pass number of sample points you want to fill.

**Params(index as integer) as double**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Parameters if Type != 0.  
**Notes:**

Index from 0 to 9.  
(Read and Write computed property)

### **SampledPoints(index as integer) as Single**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Array of floats if Type = 0.

**Notes:**

Index from 0 to nGridPoints-1.  
(Read and Write computed property)

## **2.30.2 Properties**

### **nGridPoints as UInt32**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Number of grid points if Type = 0.

**Notes:** (Read only property)

### **Type as Integer**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of curve segment.

**Notes:**

Parametric type, Type = 0 means sampled segment.  
Negative values are reserved.  
(Read and Write property)

### **x0 as Single**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The x0 value of the domain.

**Notes:**

Domain; for  $x_0 < x \leq x_1$   
(Read and Write property)

**x1 as Single**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The x1 value of the domain.

**Notes:**

Domain; for  $x_0 < x \leq x_1$   
(Read and Write property)

## 2.31 class LCMS2CIEXYZTripleMBS

**class LCMS2CIEXYZTripleMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a XYZ color triple.

### 2.31.1 Methods

**Clone as LCMS2CIEXYZTripleMBS**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the XYZ triple.

**Constructor**

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes a XYZ triple with zero values.

See also:

- 2.31.1 Constructor(other as LCMS2CIEXYZTripleMBS) 237
- 2.31.1 Constructor(Red as LCMS2CIEXYZMBS, Green as LCMS2CIEXYZMBS, Blue as LCMS2CIEXYZMBS) 237

### Constructor(other as LCMS2CIEXYZTripleMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes a XYZ triple with values from other object.

See also:

- 2.31.1 Constructor 236
- 2.31.1 Constructor(Red as LCMS2CIEXYZMBS, Green as LCMS2CIEXYZMBS, Blue as LCMS2CIEXYZMBS) 237

### Constructor(Red as LCMS2CIEXYZMBS, Green as LCMS2CIEXYZMBS, Blue as LCMS2CIEXYZMBS)

Plugin Version: 12.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes a XYZ triple with given values.

See also:

- 2.31.1 Constructor 236
- 2.31.1 Constructor(other as LCMS2CIEXYZTripleMBS) 237

## 2.31.2 Properties

### Blue as LCMS2CIEXYZMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The blue color.

**Notes:** (Read and Write property)

### Green as LCMS2CIEXYZMBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The green color.

**Notes:** (Read and Write property)

**Red as LCMS2CIEXYZMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The red color.  
**Notes:** (Read and Write property)

**2.32 class LCMS2DateMBS****class LCMS2DateMBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for a date.

**2.32.1 Methods****date as date**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The date as a Real Studio date object.

**2.32.2 Properties****Day as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The day value.  
**Notes:** (Read only property)

**Daylight as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether it is daylight saving time.  
**Notes:** (Read only property)

**DayOfWeek as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The day of the week.

**Notes:** (Read only property)

**DayOfYear as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The day of the year.

**Notes:** (Read only property)

**Hour as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The hour value.

**Notes:** (Read only property)

**Minute as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The minute value.

**Notes:** (Read only property)

**Month as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The month value.

**Notes:** (Read only property)

**Second as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The second value.

**Notes:** (Read only property)

**Year as Integer**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The year value.

**Notes:** (Read only property)

# Chapter 3

## List of all classes

• LCMS2BitmapMBS	216
• LCMS2CIECAM02MBS	220
• LCMS2CIELabMBS	222
• LCMS2CIELChMBS	211
• LCMS2CIExyYMBS	208
• LCMS2CIExyYTripleMBS	214
• LCMS2CIEXYZMBS	213
• LCMS2CIEXYZTripleMBS	236
• LCMS2CurveSegmentMBS	234
• LCMS2DateMBS	238
• LCMS2GamutBoundaryDescriptionMBS	228
• LCMS2ICCDDataMBS	226
• LCMS2ICCMeasurementConditionsMBS	227
• LCMS2ICCViewingConditionsMBS	233
• LCMS2JChMBS	231
• LCMS2Mat3MBS	206
• LCMS2MLUMBS	37
• LCMS2NamedColorListMBS	29

• LCMS2PipelineMBS	32
• LCMS2ProfileMBS	78
• LCMS2ScreeningChannelMBS	106
• LCMS2ScreeningMBS	51
• LCMS2SequenceDescriptionMBS	49
• LCMS2SequenceMBS	62
• LCMS2StageMBS	52
• LCMS2StageSamplerMBS	48
• LCMS2ToneCurveMBS	42
• LCMS2TransformMBS	70
• LCMS2UcrBgMBS	66
• LCMS2Vec3MBS	64
• LCMS2ViewingConditionsMBS	67

## Chapter 4

### List of all modules

- LCMS2MBS

108