



AcroColor API Reference

Technical Note #5425

Version : Acrobat 6.0



ADOBE SYSTEMS INCORPORATED

Corporate Headquarters


345 Park Avenue

San Jose, CA 95110-2704

(408) 536-6000

<http://partners.adobe.com>

May 2003



Copyright 2003 Adobe Systems Incorporated. All rights reserved.

NOTICE: All information contained herein is the property of Adobe Systems Incorporated. No part of this publication (whether in hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the Adobe Systems Incorporated.

PostScript is a registered trademark of Adobe Systems Incorporated. All instances of the name PostScript in the text are references to the PostScript language as defined by Adobe Systems Incorporated unless otherwise stated. The name PostScript also is used as a product trademark for Adobe Systems' implementation of the PostScript language interpreter.

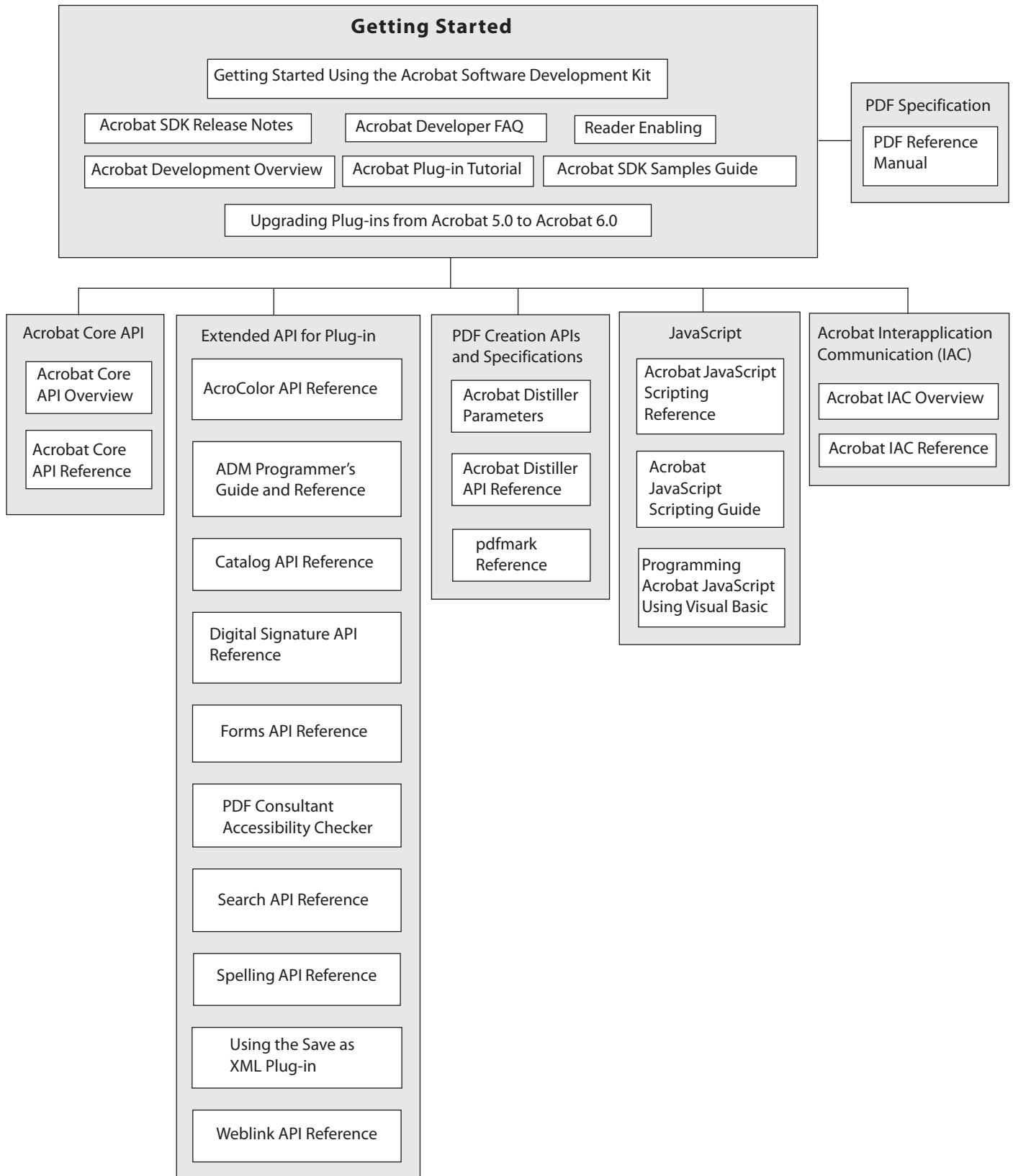
Except as otherwise stated, any reference to a "PostScript printing device," "PostScript display device," or similar item refers to a printing device, display device or item (respectively) that contains PostScript technology created or licensed by Adobe Systems Incorporated and not to devices or items that purport to be merely compatible with the PostScript language.

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Acrobat Capture, Distiller, PostScript, the PostScript logo and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apple, Macintosh, and Power Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. PowerPC is a registered trademark of IBM Corporation in the United States. ActiveX, Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. UNIX is a registered trademark of The Open Group. All other trademarks are the property of their respective owners.

This publication and the information herein is furnished AS IS, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and noninfringement of third party rights.

Acrobat SDK Documentation Roadmap





Contents

Introduction 7

- Contents 7
- Overview 7
- Conventions Used in This Book 9

AcroColor API Declarations 11

- ACCalCMYK 11
- ACCalGray 12
- ACCalLab 13
- ACCalRGB 14
- AC_ColorSpace 15
- AC_Error 17
- AC_FileSpec 19
- AC_LabColor 20
- AC_PackingCode 21
- AC_PresetList 24
- AC_Profile 25
- AC_ProfileCode 26
- AC_ProfileList 28
- AC_RenderIntent 29
- AC_SelectorCode 31
- AC_Settings 33
- AC_SettingsType 34
- AC_SettingKey 35
- AC_String 37
- AC_ToneCurve 38
- AC_Transform 39
- ACWorkingSpace 40
- AC_XYZColor 41

AcroColor API Methods 43

- ACApplyTransform 43
- ACEngineCount 44
- ACEngineInfo 45
- ACGetSettingsProfile 46
- ACGetSettingsString 47
- ACGetSettingsUnsigned32 48



ACGetWorkingSpaceProfile	49
ACLoadSettings	50
ACMakeBufferProfile	51
ACMakeCalGray	52
ACMakeCalLab	53
ACMakeCalRGB	54
ACMakeColorTransform	55
ACMakePresetList	56
ACMakeProfileList	57
ACMakeSettings	58
ACMakeString	59
ACMonitorProfile	60
ACPresetFileToName	61
ACPresetListCount	62
ACPresetListItemFile	63
ACProfileColorSpace	64
ACProfileData	65
ACProfileDescription	66
ACProfileFromCode	67
ACProfileFromDescription	68
ACProfileListCount	69
ACProfileListItemCode	70
ACProfileListItemDescription	71
ACProfileSize	72
ACProfilesMatch	73
ACSetEngine	74
ACStringASCII	75
ACStringLocalized	76
ACStringUnicode	77
ACUnReferencePresetList	78
ACUnReferenceProfile	79
ACUnReferenceProfileList	80
ACUnReferenceSettings	81
ACUnReferenceString	82
ACUnReferenceTransform	83



Introduction

AcroColor is an HFT that allows you to access the AcroColor engine (ACE), which controls color profile management for Adobe Acrobat. Plug-ins can import the AcroColor HFT to use the color management methods.

If you received this technical note without obtaining the entire Acrobat Software Development Kit (SDK), you can get the complete SDK by visiting:

<http://partners.adobe.com/asn/developer/acrosdk/main.html>

Contents

This document provides details of the public API for the AcroColor HFT:

- “[Overview](#)” on [page 7](#) give a general overview of the AcroColor objects and usage model.
- “[AcroColor API Declarations](#)” on [page 11](#) describes the data structures used in the AcroColor API.
- “[AcroColor API Methods](#)” on [page 43](#) describes the methods in the AcroColor API.

Overview

The AcroColor HFT encapsulates color management into a set of convenient objects and functions. The objects represent basic color-management entities:

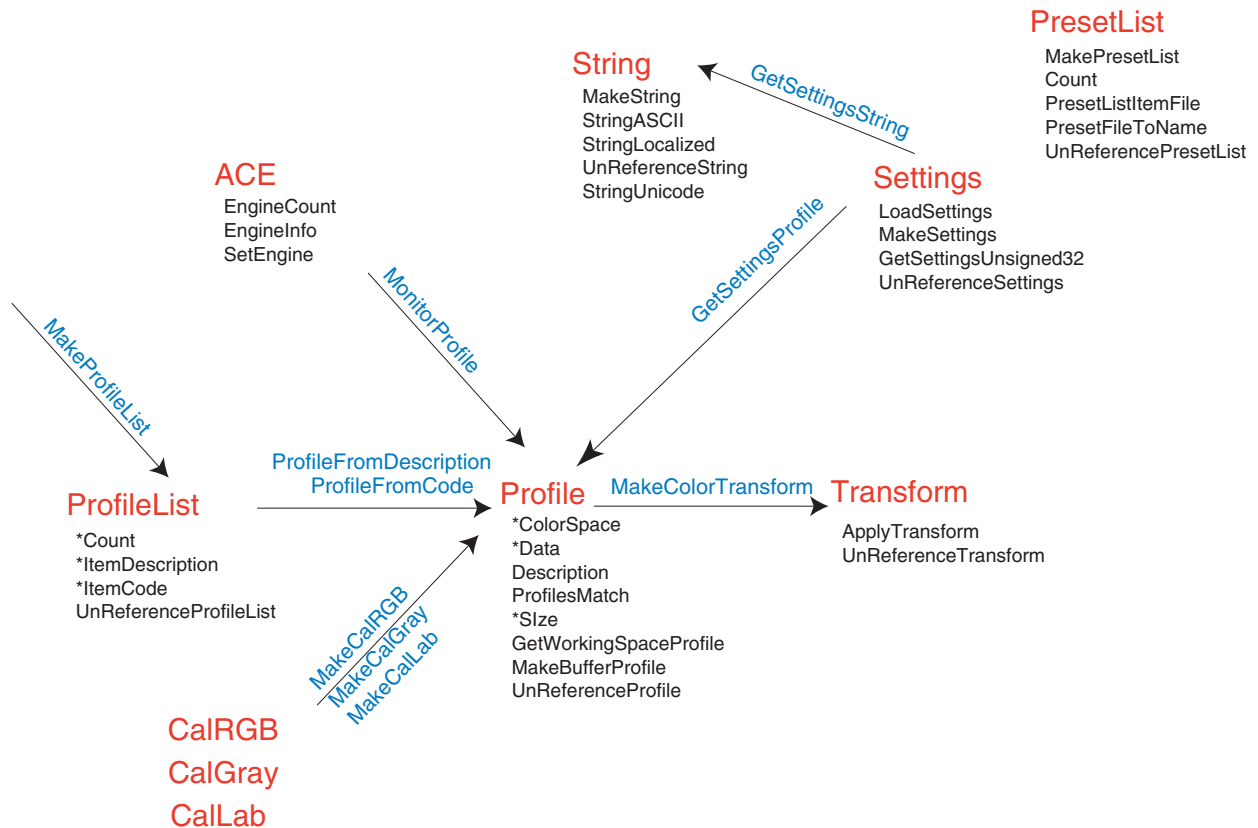
- The color management *engine*, or ACE, which is used by the underlying software to control a color management session.
- Device-specific ICC *color profiles*, which provide specific mapping between standard color specifications and specific values for particular output devices that produce those colors. Additional support objects include *profile lists*.
- *Color spaces* for the different kinds of color production (such as Grayscale, RGB, CMYK). Additional support objects include *calibrated color spaces* for standard color specifications.
- *Transformations* between profiles or color spaces.
- Color *settings*, as listed under **Edit->Preferences->Color Management->Settings** in the Acrobat UI. Color settings files contain, for instance, references to color profiles, and apply across Adobe products. Additional support objects include a *string* object and *preset lists* of settings.

You can create an ICC color profile from available data ([ACMakeBufferProfile](#)), or use profiles that are installed on the system ([ACGetWorkingSpaceProfile](#)), or stored in color settings files ([ACGetSettingsProfile](#)).

You can extract information directly from profiles, such as a string to use in the UI ([ACProfileDescription](#)). However, the most important thing you do with color profiles is use them to make transformations ([ACMakeColorTransform](#)). You can then apply it ([ACApplyTransform](#)) to transform a set of image data from one profile to another, so that it appears with the same colors on a different display device.

AcroColor objects are reference-counted. Each object type has an unreferencemethod (such as [ACUnReferenceProfile](#)). Whenever you create one of these objects, you are responsible for using the corresponding unreferencemethod to release it when you are finished with it.

The following figure illustrates the objects and their relationships.



Conventions Used in This Book

The Acrobat documentation uses text styles according to the following conventions.

Font	Used for	Examples
monospaced	Paths and filenames	<code>C:\templates\mytmpl.fm</code>
	Code examples set off from plain text	These are variable declarations: <code>AVMenu commandMenu,helpMenu;</code>
monospaced bold	Code items within plain text	The <code>GetExtensionID</code> method ...
	Parameter names and literal values in reference documents	The enumeration terminates if <code>proc</code> returns <code>false</code> .
monospaced italic	Pseudocode	<code>ACCB1 void ACCB2 ExeProc(void)</code> <code>{ do something }</code>
	Placeholders in code examples	<code>AFSimple_Calculate(cFunction, cFields)</code>
blue	Live links to Web pages	The Acrobat Solutions Network URL is: http://partners/adobe.com/asn/
	Live links to sections within this document	See Using the SDK .
	Live links to other Acrobat SDK documents	See the Acrobat Core API Overview .
	Live links to code items within this document	Test whether an <code>ASAtom</code> exists.
bold	PostScript language and PDF operators, keywords, dictionary key names	The <code>setpagedevice</code> operator
	User interface names	The File menu
italic	Document titles that are not live links	<i>Acrobat Core API Overview</i>
	New terms	<i>User space</i> specifies coordinates for...
	PostScript variables	<i>filename</i> <code>deletefile</code>



Introduction

Conventions Used in This Book

AcroColor API Declarations

ACCaLCMYK

```
typedef struct _t_ACCaLCMYK
{
    AC_XYZColor w;
    AC_XYZColor k;
    AC_XYZColor c;
    AC_XYZColor m;
    AC_XYZColor y;
    AC_XYZColor cm;
    AC_XYZColor cy;
    AC_XYZColor ck;
    AC_XYZColor my;
    AC_XYZColor mk;
    AC_XYZColor yk;
    AC_XYZColor cmy;
    AC_XYZColor cmk;
    AC_XYZColor cyk;
    AC_XYZColor myk;
    AC_XYZColor cmyk;

    AC_ToneCurve cTRC;
    AC_ToneCurve mTRC;
    AC_ToneCurve yTRC;
    AC_ToneCurve kTRC;

    double opticalGamma;

    AC_XYZColor white;
    AC_XYZColor black;
} ACCaLCMYK;
```

Description

A simple 16-patch calibrated CMYK color space specification.

Header File

AcroColorExpT.h

ACCalGray

```
typedef struct _t_ACCalGray
{
    double gamma;
    AC_XYZColor white;
    AC_XYZColor black;
} ACCalGray;
```

Description

A calibrated grayscale color space specification.

Header File

AcroColorExpT.h

Related Methods

[ACMakeCalGray](#)



ACCalLab

```
typedef struct _t_ACCalLab
{
    AC_XYZColor white;
    AC_XYZColor black;
    struct {
        ASInt32 min;
        ASInt32 max;
    } rangeA;
    struct {
        ASInt32 min;
        ASInt32 max;
    } rangeB;
} ACCalLab;
```

Description

A calibrated LAB color space specification. The usual **min** and **max** values for **rangeA** and **rangeB** are -128 and +127.

Header File

AcroColorExpT.h

Related Methods

[ACMakeCalLab](#)

ACCaIRGB

```
typedef struct _t_ACCaIRGB
{
    double redGamma;
    double greenGamma;
    double blueGamma;

    AC_XYZColor red;
    AC_XYZColor green;
    AC_XYZColor blue;

    AC_XYZColor white;
    AC_XYZColor black;
} ACCaIRGB;
```

Description

A calibrated RGB space specification.

Header File

AcroColorExpT.h

Related Methods

[ACMakeCalRGB](#)



AC_ColorSpace

Description

Constant values for ICC color space signatures.

Values

<code>AC_Space_XYZ</code>	Standard ICC color space signatures.
<code>AC_Space_Lab</code>	
<code>AC_Space_RGB</code>	
<code>AC_Space_Gray</code>	
<code>AC_Space_CMYK</code>	
<code>AC_Space_Luv</code>	
<code>AC_Space_YCbCr</code>	
<code>AC_Space_HSV</code>	
<code>AC_Space_HLS</code>	
<code>AC_Space_CMY</code>	
<code>AC_Space_2Component</code>	ICC Color space signatures for generic color spaces.
<code>AC_Space_3Component</code>	
<code>AC_Space_4Component</code>	
<code>AC_Space_5Component</code>	
<code>AC_Space_6Component</code>	
<code>AC_Space_7Component</code>	
<code>AC_Space_8Component</code>	
<code>AC_Space_9Component</code>	
<code>AC_Space_10Component</code>	
<code>AC_Space_11Component</code>	
<code>AC_Space_12Component</code>	
<code>AC_Space_13Component</code>	
<code>AC_Space_14Component</code>	
<code>AC_Space_15Component</code>	
<code>AC_Space_PhotoYCC</code>	
<code>AC_Space_Null</code>	A null color space used to represent "spot-only" color spaces.

Header File

`AcroColorExpT.h`



Related Methods

[ACProfileColorSpace](#)



AC_Error

Description

Error codes returned by AcroColor functions.

Values

AC_Error_None	No error.
AC_Error_General	Other error.
AC_Error_Param	Bad parameters to an API call.
AC_Error_Version	Application and ACE library mismatch.
AC_Error_UserAbort	The user aborted the operation. Returned by ACE when the client progress callback returns false.
AC_Error_Memory	Out of memory.
AC_Error_StackFull	Out of stack space.
AC_Error_ScratchFull	Client callback ran out of scratch space.
AC_Error_StringOverflow	String does not fit in supplied buffer.
AC_Error_NoASCII	String does not contain ASCII data.
AC_Error_NoUnicode	String does not contain Unicode data.
AC_Error_NoLocalized	String does not contain localized data.
AC_Error_BadAlignment	Data is not correctly byte aligned.
AC_Error_BadDescription	Invalid profile description.
AC_Error_BadConcat	Unable to concatenate transforms.
AC_Error_BadMerge	Unable to merge transforms.
AC_Error_BadProfile	Invalid profile.
AC_Error_UnsupCMS	Unsupported CMS.
AC_Error_UnsupOption	Unsupported ACE option.
AC_Error_UnsupPacking	Unsupported packing code.
AC_Error_UnsupProfile	Unsupported profile version.

AC_Error_UnsupProfileCode	Unsupported profile code.
AC_Error_UnsupSpace	Unsupported color space.
AC_Error_UnsupQuery	Unsupported query code.
AC_Error_MissingProfile	A profile was missing from the disk.
AC_Error_ModifiedProfile	The profile on disk has been modified.
AC_Error_FileNotFound	File is missing from disk.
AC_Error_EOF	End of file error.
AC_Error_FileLocked	File locked error.
AC_Error_DiskIO	Disk I/O error.
AC_Error_ColorSync	A problem using ColorSync.
AC_Error_ICM	A problem using ICM.
AC_Error_MissingKey	The color settings do not contain this key.
AC_Error_InvalidSettings	The color settings file is invalid.
AC_Error_SettingsVersion	The color settings file is an incompatible version.
AC_Error_NotImplemented	The function is not implemented (subsetted library).

Header File

AcroColorExpT.h



AC_FileSpec

```
typedef struct _t_AC_FileSpec
{
    #ifdef MAC_ENV
        FSSpec spec;
    #else
        char path [kACMaxPathLength];
    #endif
} AC_FileSpec;
```

Description

Contains a platform-specific version of a file specification. This is an **FSSpec** on the Macintosh, or a full path name on other platforms.

Header File

AcroColorExpT.h

Related Methods

[ACLoadSettings](#)
[ACPresetFileToName](#)
[ACPresetListItemFile](#)

AC_LabColor

```
typedef struct _t_AC_LabColor
{
    double L;
    double a;
    double b;
} AC_LabColor;
```

Description

A floating-point Lab color type. Pure white is encoded as 100.0, 0.0, 0.0. The structure supports any **a** and **b** range, but the usual range is -128.0 to +127.0.

Header File

AcroColorExpT.h

Related Methods

[ACMakeCalLab](#)



AC_PackingCode

Description

Constants that specify the packing used in a color transformation.

Values

AC_Packing_pRGB8	<p>8-bit RGB (or grayscale destination), with a leading pad byte. When grayscale is output in this format, the gray value is replicated into the R, G, and B values.</p> <p>R, G, B = 0 is black.</p> <p>R, G, B = 255 is white.</p> <p>Data must be 32-bit aligned.</p>
AC_Packing_RGB8	<p>Same as AC_Packing_pRGB8, without the leading pad byte.</p> <p>Data must be 8-bit aligned.</p>
AC_Packing_pRGB15	<p>15-bit RGB (or grayscale destination), with a leading pad word. When grayscale is output in this format, the gray value is replicated into the R, G, and B values.</p> <p>R, G, B = 0 is black.</p> <p>R, G, B = 32768 is white.</p> <p>Values greater than 32768 are invalid.</p> <p>Data must be 64-bit aligned.</p>
AC_Packing_CMYK8_Black0	<p>8-bit CMYK.</p> <p>C, M, Y, K = 0 is 100% ink.</p> <p>C, M, Y, K = 32768 is 0% ink.</p> <p>Values greater than 32768 are invalid.</p> <p>Data must be 32-bit aligned</p>
AC_Packing_CMYK15_Black0	<p>15-bit CMYK.</p> <p>C, M, Y, K = 0 is 100% ink.</p> <p>C, M, Y, K = 32768 is 0% ink.</p> <p>Values greater than 32768 are invalid.</p> <p>Data must be 64-bit aligned</p>

AC_Packing_CMYK8_White0	<p>8-bit CMYK. C, M, Y, K = 32768 is 100% ink. C, M, Y, K = 0 is 0% ink. Values greater than 32768 are invalid. Data must be 32-bit aligned</p>
AC_Packing_pLab8	<p>8-bit LAB, with a leading pad byte. L = 0 -> L* = 0.0 L = 255 -> L* = 100.0 a, b = 0 -> a*, b* = -128.0 a, b = 128 -> a*, b* = 0.0 a, b = 255 -> a*, b* = +127.0 Data must be 32-bit aligned.</p>
AC_Packing_Lab8	<p>Same as AC_Packing_pLab8, without the leading pad byte. Data must be 8-bit aligned.</p>
AC_Packing_pLab15	<p>15-bit LAB, with a leading pad word. L = 0 -> L* = 0.0 L = 32768 -> L* = 100.0 a, b = 0 -> a*, b* = -128.0 a, b = 16384 -> a*, b* = 0.0 a, b = 32768 -> a*, b* = +128.0 Values greater than 32768 are invalid. Data must be 64-bit aligned</p>
AC_Packing_Gray8_Black0	<p>8-bit Grayscale or gamut test results, no padding. G = 0 is 100% ink or black or fully out of gamut. G = 255 is 0% ink or white or fully in gamut. When used for gamut test results, any value ≥ 128 should be considered to be in gamut.</p>
AC_Packing_Gray8_White0	<p>Same as AC_Packing_Gray8_Black0 with inverse encoding.</p>
AC_Packing_Gray15_Black0	<p>15-bit Grayscale or gamut test results, no padding. G = 0 is 100% ink or black or fully out of gamut. G = 32768 is 0% ink or white or fully in gamut. Values greater than 32768 are invalid. Data must be 16-bit aligned.</p>

<code>AC_Packing_pXYZ16</code>	16-bit XYZ, with a leading pad word. X, Y, Z = 0 -> X, Y, Z = 0.0 X, Y, Z = 32768 -> X, Y, Z = 1.0 X, Y, Z = 65535 -> X, Y, Z = 1.9999694824. Data must be 64-bit aligned.
<code>AC_Packing_pABC8</code>	Generic padded 3-component 8-bit packing. Data must be 32-bit aligned.
<code>AC_Packing_ABC8</code>	Same as <code>AC_Packing_pABC8</code> , without the leading pad byte. Data must be 8-bit aligned.
<code>AC_Packing_pABC15</code>	Generic padded 3-component 15+ bit packing. Data must be 64-bit aligned.
<code>AC_Packing_ABCD8</code>	Generic 4-component 8 bit packing. Data must be 32-bit aligned.
<code>AC_Packing_ABCD15</code>	Generic 4-component 15+ bit packing. Data must be 64-bit aligned.
<code>AC_Packing_CS64_Gray</code> <code>AC_Packing_CS64_RGB</code> <code>AC_Packing_CS64_CMYK</code> <code>AC_Packing_CS64_Lab</code> <code>AC_Packing_CS64_XYZ</code> <code>AC_Packing_CS64_ABC</code> <code>AC_Packing_CS64_ABCD</code>	Packing codes for native 64-bit ColorSync formats (type "CMColor"). ICM2 also uses these packings formats (type "COLOR"). See the Apple ColorSync documentation for the details of these formats. These are mostly intended for internal use by ACE, and are not intended for use by most ACE clients. Data must be 16-bit aligned
<code>AC_Packing_Null</code>	Null data, for use with data in <code>AC_Space_Null</code>
<code>AC_Packing_General</code>	None of the above, use general packing specification.

Header File`AcroColorExpT.h`**Related Methods**[`ACApplyTransform`](#)

AC_PresetList

Description

An opaque object representing a preset list. A preset list is a list of predefined color settings that specifies the source and destination working color profiles to be used for color conversion.

Header File

`AcroColorExpT.h`

Related Methods

`ACMakePresetList`
`ACPresetListCount`
`ACPresetListItemFile`
`ACUnReferencePresetList`



AC_Profile

Description

An opaque object representing a device color profile.

Header File

AcroColorExpT.h

Related Methods

ACGetSettingsProfile
ACGetWorkingSpaceProfile
ACMakeBufferProfile
ACMakeCalGray
ACMakeCalLab
ACMakeCalRGB
ACMakeColorTransform
ACMonitorProfile
ACProfileColorSpace
ACProfileData
ACProfileDescription
ACProfileFromCode
ACProfileFromDescription
ACProfileSize
ACProfilesMatch
ACUnReferenceProfile

AC_ProfileCode

Description

Constants that describe the type of a device color profile.

Values

<code>AC_Profile_Null</code>	A null result, indication that a profile is not a built-in profile.
<code>AC_Profile_Lab_D50</code>	Adobe's standard LAB profile. It has a white point of D50, and exactly matches the ICC's LAB PCS space.
<code>AC_Profile_PCS_XYZ</code>	An XYZ profile that exactly matches the ICC's XYZ PCS space.
<code>AC_Profile_Flat_XYZ</code>	An XYZ profile with a "flat" white point encoding ($X = Y = Z = 1.0$). Photoshop uses this as an intermediate space in its display loop.
<code>AC_Profile_sRGB</code>	HP's sRGB profile. The default Windows monitor profile.
<code>AC_Profile_AppleRGB</code>	Default RGB space used by Photoshop 4.0 and earlier. The default Macintosh monitor profile.
<code>AC_Profile_AdobeRGB1998</code>	A wider gamut RGB space recommended by Adobe.
<code>AC_Profile_ColorMatchRGB</code>	A simplified version of Radius' ColorMatch RGB space, without Radius' non-zero black point.
<code>AC_Profile_Gamma18</code> <code>AC_Profile_Gamma22</code>	Grayscale display profiles with various gammas.
<code>AC_Profile_DotGain10</code> <code>AC_Profile_DotGain15</code> <code>AC_Profile_DotGain20</code> <code>AC_Profile_DotGain25</code> <code>AC_Profile_DotGain30</code>	Grayscale printer profiles with various dot gains.
<code>AC_Profile_MonitorRGB</code>	The system "Monitor RGB" profile.



<code>AC_Profile_SystemRGB</code>	The system default profiles for the color spaces, and for the input, output, and proofer devices. (ColorSync 3.0 only).
<code>AC_Profile_SystemCMYK</code>	
<code>AC_Profile_SystemGray</code>	
<code>AC_Profile_SystemInput</code>	
<code>AC_Profile_SystemOutput</code>	
<code>AC_Profile_SystemProofer</code>	
<hr/>	
<code>AC_Profile_WorkingRGB</code>	The application working color space profiles. (For use as abstract values only, since ACE does not keep track of these profiles, and thus cannot make profiles from these codes.)
<code>AC_Profile_WorkingCMYK</code>	
<code>AC_Profile_WorkingGray</code>	

Header File

`AcroColorExpT.h`

Related Methods

[ACProfileFromCode](#)

[ACProfileListItemCode](#)

AC_ProfileList

Description

An opaque object representing an list of device color profiles.

Header File

AcroColorExpT.h

Related Methods

`ACMakeProfileList`
`ACProfileListCount`
`ACProfileListItemCode`
`ACProfileListItemDescription`
`ACUnReferenceProfileList`



AC_RenderIntent

Description

Constants that specify a standard ICC rendering intent for a device color profile. The rendering intent specifies the color translation method for colors that are outside the gamut of the color profile.

Values

AC_Perceptual	Tries to preserve the visual relationship between colors in a way that is perceived as natural to the human eye, although the color values themselves may change. This is the same as the Image intent in Adobe PageMaker and Illustrator. This option is suitable for photographic, continuous tone images.
AC_RelColorimetric	The same as absolute colorimetric, except that it compares the white point of the source color space to that of the destination color space and shifts all other colors accordingly, rather than comparing individual colors.
AC_Saturation	Tries to create vivid color at the expense of accurate color. It scales the source gamut to the destination gamut, but preserves relative saturation instead of hue, so that when scaling to a smaller gamut, hues may shift. This is the same as the Graphics intent in Adobe PageMaker and Illustrator. This option is suitable for business graphics and charts, where the exact relationship between colors is not as important as having bright saturated colors.
AC_AbsColorimetric	Tries to maintain color accuracy at the expense of preserving relationships between colors. Leaves colors that fall inside the destination gamut unchanged. When translating to a smaller gamut, two colors that are distinct in the source space may be mapped to the same color in the destination space. This type of rendering can be more accurate than relativeColorimetric if the color profile of the image contains correct white point (extreme highlight) information.
AC_UseProfileIntent	Use the source profile's rendering intent.

Header File

AcroColorExpT.h



Related Methods

[ACMakeCalGray](#)
[ACMakeCalLab](#)
[ACMakeCalRGB](#)



AC_SelectorCode

Description

Constants that specify types of device profiles to include in a profile list.

Values

AC_Selector_RGB_Standard	Standard, recommended RGB profiles. These profiles are always bi-directional.
AC_Selector_RGB_OtherInputCapable	RGB profiles that can be used as a source. These profiles may or may not be useable as a destination. Does not include profiles selected by AC_Selector_RGB_Standard .
AC_Selector_RGB_OtherOutputCapable	RGB profiles that can be used as a destination. These profiles may or may not be useable as a source. Does not include profiles selected by AC_Selector_RGB_Standard .
AC_Selector_CMYK_StandardInput	Standard, recommended CMYK profiles that can be used as a source. These profiles may or may not be useable as a destination.
AC_Selector_CMYK_StandardOutput	Standard, recommended CMYK profiles that can be used as a destination. These profiles are also useable as a source.
AC_Selector_CMYK_OtherInputCapable	CMYK profiles that can be used as a source. These profiles may or may not be useable as a destination. Does not include profiles selected by AC_Selector_CMYK_StandardInput or AC_Selector_CMYK_StandardOutput .
AC_Selector_CMYK_OtherOutputCapable	CMYK profiles that can be used as a destination. These profiles are also useable as a source. Does not include profiles selected by AC_Selector_CMYK_StandardOutput .
AC_Selector_Gray_Standard	Standard, recommended grayscale profiles. These profiles are always bi-directional.
AC_Selector_Gray_OtherInputCapable	Grayscale profiles that can be used as a source. These profiles may or may not be useable as a destination. Does not include profiles selected by AC_Selector_Gray_Standard .

AC_Selector_Gray_OtherOutputCapable	Grayscale profiles that can be used as a destination. These profiles are also useable as a source. Does not include profiles selected by AC_Selector_Gray_Standard .
AC_Selector_DotGain_Standard	Standard dot gain profiles. Used by Photoshop to represent a single ink's dot gain curve. Stored as an ICC gray output profile.
AC_Selector_DotGain_Other	Other grayscale printer profiles. Does not include profiles selected by AC_Selector_DotGain_Standard , and does not include grayscale display profiles.
AC_Selector_PhotoYCC_InputCapable	PhotoYCC profiles that can be used as a source.

Header File

AcroColorExpT.h

Related Methods

[ACMakeProfileList](#)



AC_Settings

Description

An opaque object representing the settings for the AcroColor engine (ACE).

Header File

AcroColorExpT.h

Related Methods

[ACGetSettingsProfile](#)
[ACGetSettingsString](#)
[ACGetSettingsUnsigned32](#)
[ACLoadSettings](#)
[ACMakeSettings](#)
[ACUnReferenceSettings](#)

AC_SettingsType

Description

Constant values that determine the type of an [AC_Settings](#) object.

Values

AC_SettingsType_Color	Holds global color settings, such as working spaces.
AC_SettingsType_Proof	Specifies the parameters for proofing a document. The Proof Setup Files generally control a per-window setting.

Header File

AcroColorExpT.h

Related Methods

[ACMakePresetList](#)
[ACMakeSettings](#)

AC_SettingKey

Description

Constant key values for an [AC_Settings](#) object.

Values

AC_Key_Name	Settings file name string (if different that file name).
AC_Key_Description	Settings file description string.
AC_Key_WriterName	Name of application to write this settings file.
AC_Key_WorkingRGB	Working RGB profile.
AC_Key_WorkingCMYK	Working CMYK profile.
AC_Key_WorkingGray	Working gray profile.
AC_Key_WorkingSpot	Working spot profile.
AC_Key_PolicyRGB	RGB color management policy (AC_Policy enum).
AC_Key_PolicyCMYK	CMYK color management policy (AC_Policy enum).
AC_Key_PolicyGray	Gray color management policy (AC_Policy enum).
AC_Key_MismatchAskOpening	Ask about profile mismatches when opening (0 = no, 1 = yes).
AC_Key_MismatchAskPasting	Ask about profile mismatches when pasting (0 = no, 1 = yes).
AC_Key_MissingAskOpening	Ask about missing profile when opening (0 = no, 1 = yes).
AC_Key_EngineCMS	Conversion engine CMS code (4-char code, stored as unsigned32).
AC_Key_EngineCMM	Conversion engine CMM code (4-char code, stored as unsigned32).
AC_Key_Intent	Conversion intent (standard ICC integer encoding).
AC_Key_KPC	Conversion black point compensation (0 = no, 1 = yes).

AC_Key_Dither	Dither conversions between 8-bit color spaces (0 = no, 1 = yes).
AC_Key_CompressionEnabled	Enable/disable monitor compression (0 = off, 1 = on).
AC_Key_CompressionPercent	Monitor compression percent (in percent).
AC_Key_BlendGammaEnabled	Enable/disable RGB blending gamma (0 = off, 1 = on).
AC_Key_BlendGammaValue	RGB blending gamma value (100 = gamma 1.00).
AC_Key_ProofType	Proof type (AC_ProofType enum).
AC_Key_ProofProfile	Proof profile.
AC_Key_Simulate	Display simulation (AC_Simulate enum).

Header File

AcroColorExpT.h

Related Methods

[ACGetSettingsProfile](#)
[ACGetSettingsString](#)
[ACGetSettingsUnsigned32](#)



AC_String

```
typedef struct ACString *AC_String;
```

Description

A string value type for use with AcroColor functions.

Header File

```
AcroColorExpT.h
```

Related Methods

```
ACEngineInfo  
ACGetSettingsString  
ACMakeCalGray  
ACMakeCalLab  
ACMakeCalRGB  
ACMakeString  
ACPresetFileToName  
ACProfileDescription  
ACProfileFromDescription  
ACProfileListItemDescription  
ACStringASCII  
ACStringLocalized  
ACStringUnicode  
ACUnReferenceString
```

AC_ToneCurve

```
typedef struct _t_ACToneCurve
{
    ASUns32 bytesPerValue;
    ASUns32 count;
    void *data;
} AC_ToneCurve;
```

Description

A tone curve value for use in a calibrated CMYK color space specification.

Members

bytesPerValue	The number of bytes per value, 1 or 2.
count	The number of samples. Often 256.
data	A pointer to the samples.

Header File

AcroColorExpT.h

Related Data Types

[ACCalCMYK](#)



AC_Transform

Description

An opaque object representing a color transformation.

Header File

`AcroColorExpT.h`

Related Methods

[ACApplyTransform](#)

[ACMakeColorTransform](#)

[ACUnReferenceTransform](#)

ACWorkingSpace

Description

Constants that specify the color space of working profiles.

Values

<code>kACWorkingGray</code>	Grayscale profile.
<code>kACWorkingRGB</code>	RGB profile.
<code>kACWorkingCMYK</code>	CMYK profile

Header File

`AcroColorExpT.h`

Related Methods

[ACGetWorkingSpaceProfile](#)



AC_XYZColor

```
typedef struct _t_ACXYZColor
{
    double X;
    double Y;
    double Z;
} AC_XYZColor;
```

Description

A floating point XYZ color type. A pure black would be encoded as 0.0, 0.0, 0.0, while a D50 based pure white would be encoded as 0.9642, 1.0, 0.8249.

Header File

AcroColorExpT.h

Related Data Types

[ACCalCMYK](#)
[ACCalGray](#)
[ACCalLab](#)
[ACCalRGB](#)



AcroColor API Methods

ACApplyTransform

```
AC_Error ACApplyTransform (AC_Transform transform,  
const void* srcData, void* dstData, ASUns32 count,  
AC_PackingCode srcPacking, AC_PackingCode dstPacking);
```

Description

Applies a color conversion or gamut test transformation. Processes the number of colors specified by **count**, using the memory formats for the source and destination data specified in **srcPacking** and **dstPacking**. The source data and destination data can point to the same block of memory if the source and destination packing formats use the same number of bits per color.

Parameters

transform	The color conversion or transformation to apply.
srcData	The source data to transform.
dstData	The destination for the transformed data.
count	The number of colors to transform.
srcPacking	The packing type used in the source data.
dstPacking	The packing type to use in the destination data.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeColorTransform](#)

ACEngineCount

```
AC_Error ACEngineCount (ASUns32 *count);
```

Description

Gets the number of Color Management System/Color Management Module (CMS/CMM) choices available for the AcroColor engine (ACE).

Parameters

count	<i>(Filled by the method)</i> A pointer to the count value.
--------------	---

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACEngineInfo](#)
[ACSetEngine](#)



ACEngineInfo

```
AC_Error ACEngineInfo (ASUns32 index, AC_String *name,
ASUns32 *cmsID, ASUns32 *cmmID );
```

Description

Gets information for a CMS/CMM in the AcroColor engine (ACE) by index. The CMS and CMM identifiers specify an engine to the AcroColor engine (ACE). Engine names should only be used as the text for popup menus. It is better to store the identifiers in settings files (rather than names), because they are independent of localization.

Parameters

index	The 0-based index of the CMS/CMM. The highest legal value is (AC_EngineCount)-1.
name	<i>(Filled by the method)</i> Optional. If non-NULL, returns the name of the CMS/CMM.
cmsID	<i>(Filled by the method)</i> Returns the CMS identifier.
cmmID	<i>(Filled by the method)</i> Returns the CMM identifier.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACEngineCount](#)

[ACSetEngine](#)

ACGetSettingsProfile

```
AC_Error ACGetSettingsProfile (AC_Settings settings,
AC_SettingKey key, AC_Profile *profile );
```

Description

Gets the current color profile for a given key from the AcroColor engine (ACE) settings object.

- If the settings file contains a profile entry with the specified key, returns that profile.
- If the settings file contains a special **NULL** entry with the key, a **NULL** profile is returned.
- If the settings file contains a string with this key rather than an embedded profile, returns an installed profile whose description matches the string, if found.
- Otherwise, returns **AC_Error_MissingKey**.

The method does not check for known keys or legal key values. It is up to the client to write only legal key values, and to verify key values when reading.

Parameters

settings	The settings object from which the profile is obtained.
key	The value key constant.
profile	<i>(Filled by the method)</i> A pointer to the current color profile value of the given key.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsString](#)
[ACGetSettingsUnsigned32](#)
[ACLoadSettings](#)
[ACMakeSettings](#)
[ACUnReferenceProfile](#)
[ACUnReferenceSettings](#)



ACGetSettingsString

```
AC_Error ACGetSettingsString (AC_Settings settings,  
AC_SettingKey key, AC_String *string);
```

Description

Gets the current string value for a given key from the AcroColor engine (ACE) settings object.

- if the settings file contains a string entry with the specified key, returns the entry.
- If the settings file contains a special **NULL** entry with the key, returns a **NULL** string
- Otherwise, returns **AC_Error_MissingKey**.

Parameters

settings	The settings object from which the string is obtained.
key	The value key constant.
string	<i>(Filled by the method)</i> A pointer to the current string value of the given key.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetSettingsUnsigned32](#)
[ACLoadSettings](#)
[ACMakeSettings](#)
[ACUnReferenceSettings](#)
[ACUnReferenceString](#)

ACGetSettingsUnsigned32

```
AC_Error ACGetSettingsUnsigned32 (AC_Settings settings,
AC_SettingKey key, ASUns32 *value);
```

Description

Gets the current numeric value for a given key from the AcroColor engine (ACE) settings object.

- if the settings file contains an unsigned 32-bit numeric entry with the specified key, returns the entry.
- Otherwise, returns **AC_Error_MissingKey**.

Parameters

settings	The settings object from which the value is obtained.
key	The value key constant.
value	<i>(Filled by the method)</i> A pointer to the current numeric value of the given key.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetSettingsString](#)
[ACLoadSettings](#)
[ACMakeSettings](#)
[ACUnReferenceSettings](#)



ACGetWorkingSpaceProfile

```
AC_Error ACGetWorkingSpaceProfile (ACWorkingSpace space,  
AC_Profile *workingProfile);
```

Description

Gets a working color profile in a specified color space.

Parameters

space	The type of color space of the profile to obtain.
workingProfile	<i>(Filled by the method)</i> A pointer to the working profile. When done with this object, dereference it using ACUnReferenceProfile .

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACProfilesMatch](#)
[ACUnReferenceProfile](#)

ACLoadSettings

```
AC_Error ACLoadSettings (AC_Settings settings,  
AC_FileSpec *file);
```

Description

Loads the AcroColor engine (ACE) settings from a file.

Reads the settings entries from the specified file and stores them in the settings object, including entries with unknown keys or data formats. As a general rule, the client should keep the settings object around so these unknown keys are preserved when the settings are saved out. The only time the client should start with a fresh settings object is when performing another settings load.

Parameters

settings	<i>(Filled by the method)</i> The settings object.
file	A pointer to the file specification for the file containing the settings.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetSettingsString](#)
[ACGetSettingsUnsigned32](#)
[ACMakeSettings](#)
[ACPresetListItemFile](#)
[ACUnReferenceSettings](#)



ACMakeBufferProfile

```
AC_Error ACMakeBufferProfile (AC_Profile *profile, void *data,  
ASUns32 dataSize );
```

Description

Creates a device color profile object from a data buffer containing the raw ICC profile data. Copies the data, so the client can dispose of the source data. The client should call [ACUnReferenceProfile](#) when done with the profile.

Parameters

profile	<i>(Filled by the method)</i> The device profile.
data	The buffer containing the device profile data.
dataSize	The size in bytes of the data buffer.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

- [ACGetSettingsProfile](#)
- [ACGetWorkingSpaceProfile](#)
- [ACMakeCalGray](#)
- [ACMakeCalLab](#)
- [ACMakeCalRGB](#)
- [ACMonitorProfile](#)
- [ACProfileFromCode](#)
- [ACProfileFromDescription](#)
- [ACUnReferenceProfile](#)

ACMakeCalGray

```
AC_Error ACMakeCalGray (AC_Profile *profile, ACCalGray *spec,
AC_RenderIntent intent, AC_String description );
```

Description

Creates a device color profile object from a calibrated grayscale color space, with the specified rendering intent and descriptive string. The client should call [ACUnReferenceProfile](#) when done with the profile.

Parameters

profile	<i>(Filled by the method)</i> A pointer to the new device color profile.
spec	A pointer to the calibrated grayscale color space specification.
intent	The rendering intent.
description	The description of the new profile. If non- NULL , must contain ASCII data, and may contain a Unicode data also. If NULL , a hard-coded default description is used.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetWorkingSpaceProfile](#)
[ACMakeBufferProfile](#)
[ACMakeCalLab](#)
[ACMakeCalRGB](#)
[ACMonitorProfile](#)
[ACProfileFromCode](#)
[ACProfileFromDescription](#)
[ACUnReferenceProfile](#)



ACMakeCalLab

```
AC_Error ACMakeCalLab (AC_Profile *profile, ACCalLab *spec,
AC_RenderIntent intent, AC_String description);
```

Description

Creates a device color profile object from a calibrated LAB color space, with the specified rendering intent and descriptive string. The client should call [ACUnReferenceProfile](#) when done with the profile.

Parameters

profile	<i>(Filled by the method)</i> A pointer to the new device color profile.
spec	The calibrated LAB color space specification.
intent	The rendering intent.
description	The description of the new profile. If non- NULL , must contain ASCII data, and may contain a Unicode data also. If NULL , a hard-coded default description is used.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetWorkingSpaceProfile](#)
[ACMakeBufferProfile](#)
[ACMakeCalGray](#)
[ACMakeCalRGB](#)
[ACMonitorProfile](#)
[ACProfileFromCode](#)
[ACProfileFromDescription](#)
[ACUnReferenceProfile](#)

ACMakeCalRGB

```
AC_Error ACMakeCalRGB (AC_Profile *profile, ACCalRGB *spec,
AC_RenderIntent intent, AC_String description);
```

Description

Creates a device color profile object from a calibrated RGB color space, with the specified rendering intent and descriptive string. The client should call [ACUnReferenceProfile](#) when done with the profile.

Parameters

profile	<i>(Filled by the method)</i> A pointer to the new device color profile.
spec	The calibrated RGB color space specification.
intent	The rendering intent.
description	The description of the new profile. If non- NULL , must contain ASCII data, and may contain a Unicode data also. If NULL , a hard-coded default description is used.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetWorkingSpaceProfile](#)
[ACMakeBufferProfile](#)
[ACMakeCalGray](#)
[ACMakeCalLab](#)
[ACMonitorProfile](#)
[ACProfileFromCode](#)
[ACProfileFromDescription](#)
[ACUnReferenceProfile](#)



ACMakeColorTransform

```
AC_Error ACMakeColorTranform(AC_Transform *transform,
AC_Profile srcProfile, AC_Profile dstProfile,
AC_RenderIntent intent);
```

Description

Creates a color transformation object.

The client can dispose of the source and destination profiles as soon as the transform is created. If the source profile is a device link or abstract profile, then the destination profile must be **NULL**; otherwise it must be non-**NULL**.

Parameters

transform	<i>(Filled by the method)</i> A pointer to the new color transformation object.
srcProfile	The source profile from which to transform color data.
dstProfile	The destination profile to which to tranfrom color data.
intent	The rendering intent for colors outside the gamut of the destination profile.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACApplyTransform](#)
[ACUnReferenceTransform](#)

ACMakePresetList

```
AC_Error ACMakePresetList (AC_PresetList *list,  
AC_SettingsType type );
```

Description

Creates a list of preset AcroColor engine (ACE) settings, of the specified type. Clients should call [ACUnReferencePresetList](#) when done with the preset list.

A preset list is a list of predefined color settings that specifies the source and destination working color profiles to be used for color conversion.

Parameters

list	<i>(Filled by the method)</i> A pointer to the new preset list object.
type	The settings type, AC_SettingsType_Color or AC_SettingsType_Proof .

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACPresetListCount](#)
[ACPresetListItemFile](#)
[ACUnReferencePresetList](#)



ACMakeProfileList

```
AC_Error ACMakeProfileList (AC_ProfileList *list,  
AC_SelectorCode selector );
```

Description

Creates a list of device color profiles of a given type.

Builds a list of those profiles from the database that meet the criterion of the specified selector. If the profile database has never been built, it will be automatically built, without a progress callback. Clients should call [ACUnReferenceProfileList](#) when done with the profile list.

Parameters

list	<i>(Filled by the method)</i> A pointer to the new profile list object.
selector	The code for the type of device profile to include in the list.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACProfileListCount](#)
[ACProfileListItemCode](#)
[ACProfileListItemDescription](#)
[ACUnReferenceProfileList](#)

ACMakeSettings

```
AC_Error ACMakeSettings (AC_Settings *settings,  
AC_SettingsType type);
```

Description

Creates an AcroColor engine (ACE) settings object of a given type, with no entries. When done with all operations, call [ACUnReferenceSettings](#) to free the settings object.

Parameters

settings	<i>(Filled by the method)</i> A pointer to the new settings object.
type	The settings type, AC_SettingsType_Color or AC_SettingsType_Proof .

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetSettingsString](#)
[ACGetSettingsUnsigned32](#)
[ACLoadSettings](#)
[ACUnReferenceSettings](#)

ACMakeString

```
AC_Error ACMakeString (AC_String* string, const char* ascii,
const ASUns16* unicode );
```

Description

Creates an AcroColor string from a **NULL**-terminated ASCII string or a **NULL**-terminated Unicode string, or both. If both ASCII and Unicode data are specified, the **AC_String** object keeps track of both in parallel, returning the ASCII data when asked for ASCII, and the Unicode data when asked for Unicode.

These dual-encoded strings are useful as description strings for ICC profiles, which can store both ASCII and Unicode data in their description tags. The ICC profile standard requires that ASCII version of the description string be limited to 7-bit ASCII characters. The AcroColor engine requires only the Unicode descriptions to be unique among profile descriptions.

Parameters

string	<i>(Filled by the method)</i> A pointer to the new string object.
ascii	The ASCII data. Should be limited to 7-bit ASCII characters for use in profile descriptions.
unicode	The Unicode data. All Unicode characters are two-byte characters, in native byte order, including the trailing NULL .

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACProfileFromDescription](#)
[ACProfileListItemDescription](#)
[ACStringASCII](#)
[ACStringLocalized](#)
[ACStringUnicode](#)
[ACUnReferenceString](#)

ACMonitorProfile

```
AC_Error ACMonitorProfile(AC_Profile *profile,
void *monitorID);
```

Description

Gets a device color profile for a specific monitor device. The returned profile may be either RGB or grayscale. If no profile is specified by the system, returns a default platform profile (sRGB on Windows, Apple RGB on Macintosh). The client should call [ACUnReferenceProfile](#) when done with the returned profile.

Parameters

profile	<i>(Filled by the method)</i> A pointer to the profile object.
monitorID	A pointer to the platform-specific monitor device identifier. On the Macintosh, this is the monitor's AVID . On Windows, this is a NULL -terminated ASCII string containing the monitor's device name (for example, "Display").

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetWorkingSpaceProfile](#)
[ACMakeBufferProfile](#)
[ACMakeCalGray](#)
[ACMakeCalLab](#)
[ACMakeCalRGB](#)
[ACProfileFromCode](#)
[ACProfileFromDescription](#)
[ACUnReferenceProfile](#)



ACPresetFileName

```
AC_Error ACPresetFileName (const AC_FileSpec* file,  
AC_String* name);
```

Description

Translates a pre-set settings file specification to a name ready to be displayed in menus (with directory paths and file extensions removed). The client should call [ACUnReferenceString](#) when done with the name.

- If the file contains an internal name tag, the returned string is created from the internal name.
- If the file does not contain an internal name tag, the returned string is built from the file name. In this case, the method assumes that the file name is in the local script code, and the ASCII data of the returned string is also in the local script code.

Parameters

file	A pointer to the preset file specification.
name	<i>(Filled by the method)</i> A pointer to the display name string.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACPresetListItemFile](#)
[ACUnReferenceString](#)

ACPresetListCount

```
AC_Error ACPresetListCount (AC_PresetList list,  
ASUns32 *count);
```

Description

Gets the number of predefined color settings, as listed under **Edit->Preferences->Color Management->Settings** in the Acrobat UI.

Parameters

list	The preset list object.
count	<i>(Filled by the method)</i> A pointer to the number of settings in the list.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakePresetList](#)
[ACPresetListItemFile](#)
[ACUnReferencePresetList](#)



ACPresetListItemFile

```
AC_Error ACPresetListItemFile (AC_PresetList list,  
ASUns32 index, AC_FileSpec* file);
```

Description

Gets the file specification for a preset settings item in a preset list.

Parameters

list	The preset list object.
index	The item index in the list.
file	<i>(Filled by the method)</i> A pointer to the file specification for the item.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACLoadSettings](#)
[ACMakePresetList](#)
[ACPresetFileToName](#)
[ACPresetListCount](#)
[ACUnReferencePresetList](#)

ACProfileColorSpace

```
AC_Error ACProfileColorSpace (AC_Profile profile,  
AC_ColorSpace *space);
```

Description

Gets the color space for a device profile.

Parameters

profile	The device color profile.
space	<i>(Filled by the method)</i> A pointer to the color space.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACProfileData](#)
[ACProfileDescription](#)



ACProfileData

```
AC_Error ACProfileData (AC_Profile profile, void *data);
```

Description

Gets the data for a device profile. Copies the raw ICC profile data into a supplied buffer.

Parameters

profile	The device color profile.
data	<i>(Filled by the method)</i> A pointer to the color profile data.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACProfileColorSpace](#)
[ACProfileDescription](#)
[ACProfileSize](#)

ACProfileDescription

```
AC_Error ACProfileDescription (AC_Profile profile,  
AC_String *description);
```

Description

Gets the description of a device profile. The returned description string contains both ASCII and Unicode data, even if the profile itself only contains ASCII data.

Parameters

profile	The device color profile.
description	<i>(Filled by the method)</i> A pointer to the descriptive string.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

```
ACMakeString  
ACProfileColorSpace  
ACProfileData  
ACProfileListItemDescription  
ACProfileFromDescription
```



ACProfileFromCode

```
AC_Error ACProfileFromCode (AC_Profile *profile,  
AC_ProfileCode code);
```

Description

Creates a device profile from a device profile type code. The client should call [ACUnReferenceProfile](#) when done with the profile.

Parameters

profile	<i>(Filled by the method)</i> A pointer to the device color profile object.
code	The profile type code.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetWorkingSpaceProfile](#)
[ACMakeBufferProfile](#)
[ACMakeCalGray](#)
[ACMakeCalLab](#)
[ACMakeCalRGB](#)
[ACMonitorProfile](#)
[ACProfileFromDescription](#)
[ACUnReferenceProfile](#)

ACProfileFromDescription

```
AC_Error ACProfileFromDescription (AC_Profile *profile,
AC_String description);
```

Description

Finds a profile in the database given its description string. The client should call [ACUnReferenceProfile](#) when done with the profile.

- If the description string contains Unicode text, the Unicode text is used to find the profile.
- If the description string contains only ASCII text, the method tries to find a match. However, the AcroColor engine requires only Unicode descriptions to be unique, so this might return the wrong profile in some rare cases. Use ASCII-only description strings only when Unicode description strings are unavailable.

Parameters

profile	<i>(Filled by the method)</i> A pointer to the device color profile object.
description	The descriptive string.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetWorkingSpaceProfile](#)
[ACMakeBufferProfile](#)
[ACMakeCalGray](#)
[ACMakeCalLab](#)
[ACMakeCalRGB](#)
[ACMakeString](#)
[ACMonitorProfile](#)
[ACProfileFromCode](#)
[ACProfileListItemDescription](#)
[ACUnReferenceProfile](#)



ACProfileListCount

```
AC_Error ACProfileListCount (AC_ProfileList list,  
ASUns32 *count);
```

Description

Gets the number of profiles in a device color profile list.

Parameters

list	The profile list.
count	<i>(Filled by the method)</i> A pointer to the number of profiles in the list.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeProfileList](#)
[ACProfileListItemCode](#)
[ACProfileListItemDescription](#)

ACProfileListItemCode

```
AC_Error ACListItemCode (AC_ProfileList list, ASUns32 index,  
AC_ProfileCode *code);
```

Description

Gets the profile code of a specified profile in a profile list. If the specified profile does not have a code, returns **AC_Profile_Null**.

While this routine is not absolutely required, since the description string is always a unique reference, profile codes have the advantage that they are localization-independent.

Parameters

list	The profile list.
index	The index for the profile in the list.
code	<i>(Filled by the method)</i> A pointer to the profile code.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeProfileList](#)
[ACProfileListCount](#)
[ACProfileListItemDescription](#)



ACProfileListItemDescription

```
AC_Error ACListItemDescription (AC_ProfileList list,
ASUns32 index, AC_String *description);
```

Description

Returns the description string of a specified profile in a list. The returned description string always contains both ASCII and Unicode data, even if the profile itself only contains an ASCII version. You can store only the Unicode data in settings files if you wish; the [ACProfileFromDescription](#) method finds the correct profile when passed the Unicode-only string.

Parameters

list	The profile list.
index	The index for the profile in the list.
description	<i>(Filled by the method)</i> A pointer to the profile descriptive string.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeProfileList](#)
[ACMakeString](#)
[ACProfileFromDescription](#)
[ACProfileListCount](#)
[ACProfileListItemCode](#)

ACProfileSize

```
AC_Error ACProfileSize (AC_Profile profile, ASUns32 *size);
```

Description

Gets the size, in bytes, of the raw ICC profile data in a device profile.

Parameters

profile	The device color profile object.
size	<i>(Filled by the method)</i> A pointer to the profile data size.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACProfileData](#)



ACProfilesMatch

```
AC_Error ACProfilesMatch (AC_Profile workingProfile,  
AC_Profile documentProfile, ASBool *match);
```

Description

Compares the working device profile with the document device profile to determine if they are the same. This comparison ignores rendering intents, and is "fuzzy," allowing very close, but not exactly the same, profiles to match. Equivalent profiles always match, but some non-equivalent profiles may also match.

Parameters

workingProfile	The working device color profile.
documentProfile	The document's device color profile.
match	<i>(Filled by the method)</i> A pointer to the result: true if the profiles match, false otherwise.

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACGetSettingsProfile](#)
[ACGetWorkingSpaceProfile](#)
[ACUnReferenceProfile](#)

ACSetEngine

```
AC_Error ACSetEngine (ASUns32 cmsID, ASUns32 cmmID);
```

Description

Sets the AcroColor engine (ACE) for the system, changing the global default CMS/CMM choice.

This method rebuilds all existing transforms using the new engine.

If the user aborts the process, or if the ACE runs out of resources during the rebuilding process, an error code is returned and some existing transforms may still use the previous engine choice. Everything will still work, since multiple engines can be used at once. Call the method again to restart the transform rebuilding process.

Parameters

cmsID	The Color Management System (CMS) identifier for the new engine default, as returned by ACEngineInfo .
cmmID	The Color Management Module (CMM) identifier for the new engine default, as returned by ACEngineInfo .

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACEngineCount](#)

[ACEngineInfo](#)

ACStringASCII

```
AC_Error ACStringASCII (AC_String string, char* buffer,
ASUns32* count, ASUns32 maxCount);
```

Description

Copies the ASCII version of a string into a supplied buffer. Either the buffer or the count can be **NULL**.

The ICC profile standard requires that ASCII version of the profile description string be limited to 7-bit ASCII characters.

Depending on the API, OS, file contents, and so on, the method can return strings in the local script code (8-bit single byte or 8-bit encoded multi-byte). To be safe, clients should always assume that the ASCII data is in the local script code (of which the 7-bit ASCII characters are a subset).

Parameters

string	The AcroColor string containing ASCII data. If the string does not contain an ASCII version, the method returns AC_Error_NoASCII .
buffer	<i>(Filled by the method)</i> A buffer to contain a copy of the ASCII data.
count	<i>(Filled by the method)</i> A pointer to the size of buffer in bytes, including the trailing NULL character.
maxCount	The maximum size of buffer in bytes. If the length of the string is longer than maxCount , the method copies a truncated string to the buffer and returns AC_Error_StringOverflow .

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeString](#)
[ACStringLocalized](#)
[ACStringUnicode](#)
[ACUnReferenceString](#)

ACStringLocalized

```
AC_Error ACStringLocalized (AC_String string, ASUns16* buffer,
ASUns32* count, ASUns32 maxCount);
```

Description

Copies the localized Unicode version of a string into a supplied buffer. Either the buffer or the count can be **NULL**.

The settings file format, and ICC profiles later than version 2, can contain text in multiple languages or countries. When the AcroColor engine (ACE) create strings from these files or profiles, it uses the current language and country codes to create strings with a third fork--a localized Unicode version. These localized versions are intended for user display only, and should not be stored in preferences files or action scripts, since they vary from country to country and are not portable.

Parameters

string	The AcroColor string containing localized Unicode data. If the string does not contain a localized Unicode version. the method returns AC_Error_NoLocalized
buffer	<i>(Filled by the method)</i> A buffer to contain a copy of the localized Unicode data.
count	<i>(Filled by the method)</i> A pointer to the size of buffer in bytes, including the trailing NULL character.
maxCount	The maximum size of buffer in bytes. If the length of the string is longer than maxCount , the method copies a truncated string to the buffer and returns AC_Error_StringOverflow .

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeString](#)
[ACStringASCII](#)
[ACStringUnicode](#)
[ACUnReferenceString](#)



ACStringUnicode

```
AC_Error ACStringUnicode (AC_String string, ASUns16* buffer,
ASUns32* count, ASUns32 maxCount);
```

Description

Copies the Unicode version of a string into a supplied buffer. Either the buffer or the count can be **NULL**.

Parameters

string	The AcroColor string containing localized Unicode data. If the string does not contain a Unicode version, the method returns AC_Error_NoUnicode .
buffer	<i>(Filled by the method)</i> A buffer to contain a copy of the Unicode data.
count	<i>(Filled by the method)</i> A pointer to the size of buffer in bytes, including the trailing NULL character.
maxCount	The maximum size of buffer in bytes. If the length of the string is longer than maxCount , the method copies a truncated string to the buffer and returns AC_Error_StringOverflow .

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeString](#)
[ACStringASCII](#)
[ACStringLocalized](#)
[ACUnReferenceString](#)

ACUnReferencePresetList

```
AC_Error ACUnReferencePresetList (AC_PresetList list);
```

Description

Decrements the reference count of a preset list object. If this causes the object's reference count to reach zero, the method deletes it.

Parameters

list	The preset list object.
-------------	-------------------------

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakePresetList](#)



ACUnReferenceProfile

```
AC_Error ACUnReferenceProfile (AC_Profile profile);
```

Description

Decrements the reference count of a device color profile object. If this causes the object's reference count to reach zero, the method deletes it.

Parameters

profile	The profile object.
----------------	---------------------

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

```
ACGetSettingsProfile  
ACGetWorkingSpaceProfile  
ACMakeBufferProfile  
ACMakeCalGray  
ACMakeCalLab  
ACMakeCalRGB  
ACMonitorProfile  
ACProfileFromCode  
ACProfileFromDescription  
ACProfileFromDescription
```

ACUnReferenceProfileList

```
AC_Error ACUnReferenceProfileList (AC_ProfileList list);
```

Description

Decrements the reference count of a device color profile list object. If this causes the object's reference count to reach zero, the method deletes it.

Parameters

list	The profile list object.
-------------	--------------------------

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeProfileList](#)



ACUnReferenceSettings

```
AC_Error ACUnReferenceSettings (AC_Settings settings);
```

Description

Decrements the reference count of an AcroColor engine settings object. If this causes the object's reference count to reach zero, the method deletes it.

Parameters

settings	The settings object.
-----------------	----------------------

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeSettings](#)

ACUnReferenceString

```
AC_Error ACUnReferenceString (AC_String string );
```

Description

Decrements the reference count of a string object. If this causes the object's reference count to reach zero, the method deletes it.

Parameters

string	The string object.
---------------	--------------------

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeString](#)



ACUnReferenceTransform

```
AC_Error ACUnReferenceTransform (AC_Transform transform);
```

Description

Decrements the reference count of a color transformation object. If this causes the object's reference count to reach zero, the method deletes it.

Parameters

transform	The transform object.
------------------	-----------------------

Return Value

0 if successful, an non-zero error code otherwise.

Header File

AcroColorCalls.h

Product

TBD

Related Methods

[ACMakeColorTransform](#)

