Application Services / ColorSync Manager

API Collection

ColorSync Manager

Overview

The ColorSync Manager is the API for ColorSync, a platform-independent color management system from Apple. ColorSync provides essential services for fast, consistent, and accurate color calibration, proofing, and reproduction using input, output, and display devices. ColorSync also provides an interface to system-wide color management settings that allows users to save color settings for specific jobs and switch between settings.

You need this reference if your software product performs color drawing, printing, or calculation, or if your peripheral device supports color. You also need this reference if you are creating a color management module (CMM)—a component that implements color-matching, color-conversion, and gamut-checking services.

The Color Picker Manager, documented separately, provides a standard user interface for soliciting color choices.

Carbon supports the majority of the ColorSync Manager programming interface. However, ColorSync 1.0 compatibility calls such as `CWNewColorWorld`, `GetProfile`, and `SetProfile` are not supported.

Nor does Carbon support ColorSync functions used for color management modules (CMMs). These functions aren't supported because macOS uses Bundle Services to implement CMMs.

Some applications use the Component Manager to determine what CMMs are available. You cannot use the Component Manager for this purpose in macOS. Apple has, however, provided the function `CMIterateCMMInfo` to query for available CMMs.

Topics

Working With Universal Procedure Pointers

`NewCMBitmapCallBackUPP`

Creates a new universal procedure pointer (UPP) to a bitmap callback.

`DisposeCMBitmapCallBackUPP`

Disposes of a universal procedure pointer (UPP) to a bitmap callback.

`InvokeCMBitmapCallBackUPP`

Invokes a universal procedure pointer (UPP) to a bitmap callback.

`NewCMConcatCallBackUPP`

Creates a new universal procedure pointer (UPP) to a progress-monitoring callback.

`DisposeCMConcatCallBackUPP`

Disposes of a universal procedure pointer (UPP) to a progress-monitoring callback.

`InvokeCMConcatCallBackUPP`

Invokes a universal procedure pointer (UPP) to a progress-monitoring callback.

`NewCMFlattenUPP`

Creates a new universal procedure pointer (UPP) to a data-flattening callback.

`DisposeCMFlattenUPP`

Disposes of a universal procedure pointer (UPP) to a data-flattening callback.

`InvokeCMFlattenUPP`

Invokes a universal procedure pointer (UPP) to a data-flattening callback.

`NewCMMIterateUPP`

Creates a new universal procedure pointer (UPP) to a progress-monitoring callback for the `CMIterateCMMInfo` function.

`DisposeCMMIterateUPP`

Disposes of a universal procedure pointer (UPP) to a progress-monitoring callback for the `CMIterateCMMInfo` function.

`InvokeCMMIterateUPP`

Invokes a universal procedure pointer (UPP) to a progress-monitoring callback for the CMIterateCMMInfo function.

`NewCMProfileIterateUPP`

Creates a new universal procedure pointer (UPP) to a profile-iteration callback.

`DisposeCMProfileIterateUPP`

Disposes of a universal procedure pointer (UPP) to a profile-iteration callback.

`InvokeCMProfileIterateUPP`

Invokes a universal procedure pointer (UPP) to a profile-iteration callback.

Callbacks

`typealias CMFlattenProcPtr`

Defines a pointer to a data transfer callback function that transfers profile data from the format for embedded profiles to disk file format or vice versa.

Data Types

`struct CM2Profile`

`struct CMDeviceInfo`

`struct CMDeviceProfileArray`

`struct CMDeviceScope`

`struct CMError`

Defines motion errors.

`typealias CMFlattenUPP`

Defines a universal procedure pointer to a data-flattening callback.

`typealias CMMultiFunctLutA2BType`

`struct CMMultiFunctLutType`

`struct CMXYZColor`

Contains values for a color specified in XYZ color space.

`typealias CMXYZComponent`

Constants

Abstract Color Space Constants

Specify values that represent general color spaces.

Channel Encoding Format

Specify an encoding format for sRGB64.

Color Packing for Color Spaces

Specify how color values are stored.

Color Space Signatures

Define four-character-sequences associated with color spaces.

Color Space Masks

Specify masks used for color spaces.

Current Device Versions

Specify the current versions of the data structure containing information on registered devices.

Current Info Versions

Specify current device and profile versions.

Current Major Version Mask

Specifies the current major version number.

Data Transfer Commands

Specify commands for caller-supplied ColorSync data transfer functions.

Data Type Element Values

Specify a data type.

Default CMM Signature

Specifies a signature for the default color management module supplied by Color Sync.

Default IDs

Specify default values for device and profile IDs.

Device Attribute Values for Version 2.x Profiles

Define masks your application can use to set or test bits in the `deviceAttributes` field of the `CM2Header` structure.

`typealias CMDeviceClass`

Define constants to represent a variety of input and output devices.

Device and Media Attributes

Used to set or obtain device or media attributes.

Device States

Specify device states.

Element Tags and Signatures for Version 1.0 Profiles

Define tags and signatures used for version 1.0 profiles.

Embedded Profile Flags

Specify copyright-protection flag options,

Flag Mask Definitions for Version 2.x Profiles

Define masks your application can use to set or test various bits in the `flags` field of the `CM2Header` structure.

ICC Profile Versions

Specify IDD profile version numbers.

Illuminant Measurement Endocings

Specify standard illuminate measurement encodings.

Magic Cookie Number

Specifies a magic cookie number for anonymous file ID.

Maximum Path Size

Specifies the maximum length for a path name.

Measurement Flares

Specify measurement flare encodings.

Measurement Geometries

Specify measurement geometry encodings.

Parametric Types

Specify a parametric curve type enumeration,

Platform Enumeration Values

Specify computer platforms.

Profile Iteration Values

Specify profiles to iterate.

Profile Location Sizes

Specify a location size.

PostScript Data Formats

Specify constants that indicate the format of PostScript data.

Profile Access Procedures

Specify operations used to access profiles.

Profile Classes

Specify profile class enumerations.

Profile Concatenation Values

Specify values to use when concatenating profiles.

Profile Iteration Constants

Define an iteration version.

Profile Location Type

Defines profile location kinds.

Public Tags

Specify tag values available for public use.

Public Type Signatures

Specify signatures for public types.

Quality Flag Values for Version 2.x Profiles

Define the possible values for the quality bits in the `flags` field of the `CM2Header` structure.

Rendering Intent Values for Version 2.x Profiles

Define the four possible values for the rendering intent bits of the `renderingIntent` field of the `CM2Header` structure.

Screen Encoding Tags

Specify tags to use for screen encodings.

Spot Function Values

Specify values for spot functions.

Standard Observer

Standard observer measurement type encodings.

Tag Type Information

Defines a constant for 2.0 tag type information.

Technology Tag Descriptions

Define descriptor tags for technologies.

Use Types

Specify use types.

Video Card Gamma Storage Types

Specify data storage type constants.

Video Card Gamma Tags

Specify video card gamma information.

Video Card Gamma Signatures

Specify signatures used for video card gamma information.

Result Codes

The most common result codes returned by ColorSync Manager are listed below.

`var cmProfileError: Int`

There is something wrong with the content of the profile

`var cmMethodError: Int`

An error occurred during the CMM arbitration process that determines the CMM to use

`var cmMethodNotFound: Int`

CMM not present

`var cmProfileNotFound: Int`

Responder error

`var cmProfilesIdentical: Int`

Profiles are the same

`var cmCantConcatenateError: Int`

Profiles cannot be concatenated

`var cmCantXYZ: Int`

CMM does not handle XYZ color space

`var cmCantDeleteProfile: Int`

Responder error

`var cmUnsupportedDataType: Int`

Responder error

`var cmNoCurrentProfile: Int`

Responder error

`var cmElementTagNotFound: Int`

The tag you specified is not in the specified profile

`var cmIndexRangeErr: Int`

Tag index out of range

`var cmCantDeleteElement: Int`

Cannot delete the specified profile element

`var cmFatalProfileErr: Int`

Returned from File Manager while updating a profile file in response to `CMUpdateProfile`; profile content may be corrupted

`var cmInvalidProfile: Int`

Profile reference is invalid or refers to an inappropriate profile

`var cmInvalidProfileLocation: Int`

Operation not supported for this profile location

`var cmInvalidSearch: Int`

Bad search handle

`var cmSearchError: Int`

Internal error occurred during profile search

`var cmErrIncompatibleProfile: Int`

Unspecified profile error

`var cmInvalidColorSpace: Int`

Profile color space does not match bitmap type

`var cmInvalidSrcMap: Int`

Source pixel map or bitmap was invalid

`var cmInvalidDstMap: Int`

Destination pix/bit map was invalid

`var cmNoGDevicesError: Int`

Begin matching or end matching—no graphics devices available

`var cmInvalidProfileComment: Int`

Bad profile comment during `drawpicture`

`var cmRangeOverFlow: Int`

One or more output color value overflows in color conversion; all input color values will be converted and the overflow will be clipped

`var cmCantCopyModifiedV1Profile: Int`

It is illegal to copy version 1.0 profiles that have been modified

`var cmNamedColorNotFound: Int`

The specified named color was not found in the specified profile

`var cmCantGamutCheckError: Int`

Gamut checking not supported by this color world—that is, the color world does not contain a gamut table because it was built with gamut checking turned off

`var cmDeviceDBNotFoundErr: Int`

Preferences not found or loaded; returned by a CM device integration routine.

`var cmDeviceAlreadyRegistered: Int`

Device already registered; returned by a CM device integration routine.

`var cmDeviceNotRegistered: Int`

Device not found; returned by a CM device integration routine.

`var cmDeviceProfilesNotFound: Int`

Profiles not found; returned by a CM device integration routine.

`var cmInternalCFErr: Int`

CoreFoundation failure; returned by a CM device integration routine.

See Also

Managers

- Apple Event Manager
- Speech Synthesis Manager