
QuickTime Constants Reference

QuickTime



2006-05-23



Apple Inc.
© 2006 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

.Mac is a registered service mark of Apple Inc.

Apple, the Apple logo, Carbon, ColorSync, Logic, Mac, Mac OS, Macintosh, Pixlet, QuickDraw, QuickTime, and SoundTrack are trademarks of Apple Inc., registered in the United States and other countries.

Adobe, Acrobat, and PostScript are trademarks or registered trademarks of Adobe Systems Incorporated in the U.S. and/or other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Times is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

QuickTime Constants Reference 5

Overview	5
Constants	5
Atom ID Codes	5
FCompressImage Values	14
Codec Flags	15
VDSetCompression Values	21
CodecInfo Values	21
CreateMovieFile Values	23
FlattenMovieData Values	24
ICM Preferences and Flags	24
ImageFieldSequenceExtractCombine Values	26
QTSetComponentProperty Values	26
kDataHCanRead	27
QTVRWrapAndConstrain Values	28
Sprite Properties	29
QTSampleTableGetSampleFlags Values	30
movieFileSpecValid	31
MovieImportDataRef Values	31
MovieProgressProc Values	32
New Movie Properties	34
NewMovieController Values	34
QuickTime Preferences Dialog Options	35
Standard Compression Constants	40
SGPanelGetDITLForSize Values	44
Media Identifiers	44
SpriteWorldHitTest Values	45
Text Properties	46
ToneDescription Values	50
Arithmetic and Logical Operator IDs	50
Codec Identifiers	50
Codec Properties	51
Codec Type Constants	52
Color Constants	54
Color Modes	54
Component Call Selectors	54
Component Identifiers	63
Component Property IDs and Flags	64
Error Codes	64
File Types and Creators	70
Graphics Transfer Modes	72

Localization Codes 76

Document Revision History 85

Index 87

QuickTime Constants Reference

Framework:	Frameworks/QuickTime.framework
Declared in	ImageCodec.h ImageCompression.h MacErrors.h Movies.h QuickTimeComponents.h QuickTimeMusic.h QuickdrawTypes.h Script.h

Overview

This reference covers the constants common to multiple QuickTime frameworks.

Constants

Atom ID Codes

Identify the four-character type codes of atoms.

```

enum {
    ConnectionSpeedPrefsType      = 'cspd',
    ConnectionSpeedIsValidPrefsType = 'vspd'
};
enum {
    kEffectNameAtom              = 'name', /* name of effect */
    kEffectTypeAtom              = 'type', /* codec sub-type for effect */
    kEffectManufacturerAtom      = 'manu' /* codec manufacturer for effect */
};
enum {
    kGraphicsExportGroup         = 'expo',
    kGraphicsExportFileType      = 'ftyp',
    kGraphicsExportMIMEType      = 'mime',
    kGraphicsExportExtension     = 'ext ',
    kGraphicsExportDescription   = 'desc'
};
enum {
    kInputMapSubInputID         = 'subi'
};
enum {
    kMovieMediaDataReference      = 'mmdr', /* data reference*/
    kMovieMediaDefaultDataReferenceID = 'ddri', /* atom id*/
    kMovieMediaSlaveTime         = 'slti', /* boolean*/
    kMovieMediaSlaveAudio        = 'slau', /* boolean*/
    kMovieMediaSlaveGraphicsMode = 'slgr', /* boolean*/
    kMovieMediaAutoPlay          = 'play', /* boolean*/
    kMovieMediaLoop              = 'loop', /* UInt8 (0=no loop, 1=loop, 2=palindrome
loop)*/
    kMovieMediaUseMIMEType       = 'mime', /* string indicating the MIME type to
use for the dataref (usually not required)*/
    kMovieMediaTitle             = 'titl', /* string of the media's title (tooltips)*/
    kMovieMediaAltText           = 'altt', /* string of alternate text if media
isn't loaded*/
    kMovieMediaClipBegin        = 'clpb', /* MovieMediaTimeRecord of start time of
embedded media*/
    kMovieMediaClipDuration      = 'clpd', /* MovieMediaTimeRecord of duration of
embedded media*/
    kMovieMediaRegionAtom       = 'regi', /* contains subatoms that describe layout*/
    kMovieMediaSlaveTrackDuration = 'sltr', /* Boolean indicating that media handler
should adjust track and media based on actual embedded movie duration*/
    kMovieMediaEnableFrameStepping = 'enfs', /* boolean. if true stepping on external
movie steps frames within embedded movie.*/
    kMovieMediaBackgroundColor   = 'bkcl', /* RGBColor.*/
    kMovieMediaPrerollTime       = 'prer' /* SInt32 indicating preroll time*/
};
enum {
    kMovieMediaSpatialAdjustment = 'fit ', /* OSType from kMovieMediaFit**/
    kMovieMediaRectangleAtom      = 'rect',
    kMovieMediaTop                = 'top ',
    kMovieMediaLeft               = 'left',
    kMovieMediaWidth              = 'wd ',
    kMovieMediaHeight             = 'ht '
};
enum {
    kQTEventType                 = 'evnt',
    kAction                      = 'actn',
    kWhichAction                 = 'whic',
    kActionParameter              = 'parm',

```

```

kActionTarget                = 'targ',
kActionFlags                 = 'flag',
kActionParameterMinValue    = 'minv',
kActionParameterMaxValue    = 'maxv',
kActionListAtomType         = 'list',
kExpressionContainerAtomType = 'expr',
kConditionalAtomType        = 'test',
kOperatorAtomType           = 'oper',
kOperandAtomType            = 'oprn',
kCommentAtomType            = 'why ',
kCustomActionHandler        = 'cust',
kCustomHandlerID            = 'id ',
kCustomHandlerDesc          = 'desc',
kQTEventRecordAtomType     = 'erec'
};
enum {
    kQTParseTextHREFText      = 'text', /* string*/
    kQTParseTextHREFBaseURL   = 'bur1', /* string*/
    kQTParseTextHREFClickPoint = 'clik', /* Point; if present, QTParseTextHREF
will expand URLs to support server-side image maps*/
    kQTParseTextHREFUseAltDelim = 'altd', /* boolean; if no
kQTParseTextHREFDelimiter, delim is ':'*/
    kQTParseTextHREFDelimiter = 'delm', /* character*/
    kQTParseTextHREFRecomposeHREF = 'rhrf' /* Boolean; if true, QTParseTextHREF
returns recomposed HREF with URL expanded as appropriate*/
};
enum {
    kQTResolutionSettings     = 'reso',
    kQTTargetDataSize         = 'dasz',
    kQTDontRecompress         = 'dntr',
    kQTInterlaceStyle         = 'ilac',
    kQTColorSyncProfile       = 'iccp',
    kQTThumbnailSettings      = 'thum',
    kQTEnableExif             = 'exif', /* UInt8 (boolean)*/
    kQTMetaData               = 'meta'
};
enum {
    kQTSConnectionPrefsType   = 'stcm', /* root atom that all other atoms are
contained in*/
                                /* kQTSNotUsedForProxyPrefsType = 'nopr',
// comma-delimited list of URLs that are never used for proxies*/
    kQTSConnectionMethodPrefsType = 'mthd', /* connection method (OSType that
matches one of the following three)*/
    kQTSDirectConnectPrefsType = 'drct', /* used if direct connect
(QTSDirectConnectPrefsRecord)*/
                                /* kQTSRTSPProxyPrefsType = 'rtsp',
// used if RTSP Proxy (QTSPProxyPrefsRecord)*/
    kQTS SOCKSPrefsType       = 'sock' /* used if SOCKS Proxy
(QTSPProxyPrefsRecord)*/
};
enum {
    kQTSNullNotification      = 'null', /* NULL */
    kQTSErrorNotification     = 'err ', /* QTSErrorParams*, optional */
    kQTSNewPresDetectedNotification = 'newp', /* QTSNewPresDetectedParams* */
    kQTSPresBeginChangingNotification = 'prcb', /* NULL */
    kQTSPresDoneChangingNotification = 'prcd', /* NULL */
    kQTSPresentationChangedNotification = 'prch', /* NULL */
    kQTSNewStreamNotification = 'stnw', /* QTSNewStreamParams* */
};

```

```

kQTSSStreamBeginChangingNotification = 'stcb', /* QTSSStream */
kQTSSStreamDoneChangingNotification = 'stcd', /* QTSSStream */
kQTSSStreamChangedNotification = 'stch', /* QTSSStreamChangedParams* */
kQTSSStreamGoneNotification = 'stgn', /* QTSSStreamGoneParams* */
kQTSPreviewAckNotification = 'pvak', /* QTSSStream */
kQTSPrerollAckNotification = 'pack', /* QTSSStream */
kQTSSStartAckNotification = 'sack', /* QTSSStream */
kQTSSStopAckNotification = 'xack', /* QTSSStream */
kQTSSStatusNotification = 'stat', /* QTSSStatusParams* */
kQTSSURLNotification = 'url ', /* QTSSURLParams* */
kQTSDurationNotification = 'dura', /* QTSDurationAtom* */
kQTSSNewPresentationNotification = 'nprs', /* QTSPresentation */
kQTSPresentationGoneNotification = 'xprs', /* QTSPresentation */
kQTSPresentationDoneNotification = 'pdon', /* NULL */
kQTSSBandwidthAlertNotification = 'bwal', /* QTSSBandwidthAlertParams* */
kQTSSAnnotationsChangedNotification = 'meta' /* NULL */
};
enum {
    kQTSSStatisticsInfo = 'stat', /* QTSSStatisticsParams* */
    kQTSSMinStatusDimensionsInfo = 'mstd', /* QTSDimensionParams* */
    kQTSSNormalStatusDimensionsInfo = 'nstd', /* QTSDimensionParams* */
    kQTSTotalDataRateInfo = 'drtt', /* UInt32*, add to what's there */
    kQTSTotalDataRateInInfo = 'drti', /* UInt32*, add to what's there */
    kQTSTotalDataRateOutInfo = 'drto', /* UInt32*, add to what's there */
    kQTSLostPercentInfo = 'lpct', /* QTSLostPercentParams*, add to what's
there */
    kQTSSNumViewersInfo = 'nviw', /* UInt32* */
    kQTSSMediaTypeInfo = 'mtyp', /* OSType* */
    kQTSSNameInfo = 'name', /* QTSSNameParams* */
    kQTSSCanHandleSendDataType = 'chsd', /* QTSSCanHandleSendDataTypeParams* */
    kQTSSAnnotationsInfo = 'meta', /* QTAtomContainer */
    kQTSSRemainingBufferTimeInfo = 'btms', /* UInt32* remaining buffer time before
playback, in microseconds */
    kQTSSInfo_SettingsText = 'sttx', /* QTSSettingsTextParams* */
    kQTSSInfo_AverageFrameRate = 'fps ' /* UnsignedFixed* */
};
enum {
    kQTSSStreamMediaType = 'strm'
};
enum {
    kQTSTargetBufferDurationInfo = 'bufr', /* Fixed* in seconds; expected, not actual
*/
    kQTSDurationInfo = 'dura', /* QTSDurationAtom* */
    kQTSSSoundLevelMeteringEnabledInfo = 'mtrn', /* Boolean* */
    kQTSSSoundLevelMeterInfo = 'levm', /* LevelMeterInfoPtr */
    kQTSSSourceTrackIDInfo = 'otid', /* UInt32* */
    kQTSSSourceLayerInfo = 'olyr', /* UInt16* */
    kQTSSSourceLanguageInfo = 'olng', /* UInt16* */
    kQTSSSourceTrackFlagsInfo = 'otfl', /* SInt32* */
    kQTSSSourceDimensionsInfo = 'odim', /* QTSDimensionParams* */
    kQTSSSourceVolumesInfo = 'ovol', /* QTSSVolumesParams* */
    kQTSSSourceMatrixInfo = 'omat', /* MatrixRecord* */
    kQTSSSourceClipRectInfo = 'oclp', /* Rect* */
    kQTSSSourceGraphicsModeInfo = 'ogrm', /* QTSSGraphicsModeParams* */
    kQTSSSourceScaleInfo = 'oscl', /* Point* */
    kQTSSSourceBoundingRectInfo = 'orct', /* Rect* */
    kQTSSSourceUserDataInfo = 'oudt', /* UserData */
    kQTSSSourceInputMapInfo = 'oimp', /* QTAtomContainer */
};

```



```

kQTSInfo_DataProc          = 'datp', /* QTSDDataProcParams* */
kQTSInfo_SendDataExtras   = 'dext', /* QTSSendDataExtrasParams* */
kQTSInfo_HintTrackID     = 'htid', /* long* */
kQTSInfo_URL              = 'url ', /* Handle*, cstring in handle */
kQTSInfo_Authentication   = 'aup', /* QTSAuthenticationParams */
kQTSInfo_MediaPacketizer  = 'rmpk' /* ComponentInstance */
};
enum {
kQTVRNodeHeaderAtomType   = 'ndhd',
kQTVRHotSpotParentAtomType = 'hspa',
kQTVRHotSpotAtomType     = 'hots',
kQTVRHotSpotInfoAtomType  = 'hsin',
kQTVRLinkInfoAtomType     = 'link'
};
enum {
kQTVRObjectInfoAtomID     = 1,
kQTVRObjectImageTrackRefAtomID = 1, /* New with 2.1, it adds a track reference
to select between multiple image tracks*/
kQTVRObjectHotSpotTrackRefAtomID = 1 /* New with 2.1, it adds a track reference
to select between multiple hotspot tracks*/
};
enum {
kQTVRStringAtomType       = 'vrsg',
kQTVRStringEncodingAtomType = 'vrse', /* New with 2.1*/
kQTVRPanoSampleDataAtomType = 'pdat',
kQTVRObjectInfoAtomType   = 'obji',
kQTVRImageTrackRefAtomType = 'imtr', /* Parent is kQTVRObjectInfoAtomType.
Required if track ref is not 1 as required by 2.0 format.*/
kQTVRHotSpotTrackRefAtomType = 'hstr', /* Parent is kQTVRObjectInfoAtomType.
Required if track ref is not 1 as required by 2.0 format.*/
kQTVRAngleRangeAtomType   = 'arng',
kQTVRTrackRefArrayAtomType = 'tref',
kQTVRPanConstraintAtomType = 'pcon',
kQTVRTiltConstraintAtomType = 'tcon',
kQTVRFOVConstraintAtomType = 'fcon',
kQTVRCubicViewAtomType    = 'cuvw', /* New with 5.0*/
kQTVRCubicFaceDataAtomType = 'cufa' /* New with 5.0*/
};
enum {
kQTVRWorldHeaderAtomType   = 'vrsc',
kQTVRImagingParentAtomType = 'imgp',
kQTVRPanoImagingAtomType   = 'impn',
kQTVRObjectImagingAtomType = 'imob',
kQTVRNodeParentAtomType    = 'vrnp',
kQTVRNodeIDAtomType        = 'vrni',
kQTVRNodeLocationAtomType  = 'nloc',
kQTVRCursorParentAtomType  = 'vrpc', /* New with 2.1*/
kQTVRCursorAtomType        = 'CURS', /* New with 2.1*/
kQTVRColorCursorAtomType   = 'crsr' /* New with 2.1*/
};
enum {
kSpriteAtomType            = 'sprt',
kSpriteImagesContainerAtomType = 'imct',
kSpriteImageAtomType       = 'imag',
kSpriteImageDataAtomType   = 'imda',
kSpriteImageDataRefAtomType = 'imre',
kSpriteImageDataRefTypeAtomType = 'imrt',
kSpriteImageGroupIDAtomType = 'imgr',

```

```

kSpriteImageRegistrationAtomType = 'imrg',
kSpriteImageDefaultImageIndexAtomType = 'defi',
kSpriteSharedDataAtomType       = 'dflt',
kSpriteNameAtomType             = 'name',
kSpriteImageNameAtomType        = 'name',
kSpriteUsesImageIDsAtomType     = 'uses', /* leaf data is an array of QTAtomID's,
one per image used*/
kSpriteBehaviorsAtomType        = 'beha',
kSpriteImageBehaviorAtomType    = 'imag',
kSpriteCursorBehaviorAtomType   = 'crsr',
kSpriteStatusStringsBehaviorAtomType = 'sstr',
kSpriteVariablesContainerAtomType = 'vars',
kSpriteStringVariableAtomType   = 'strv',
kSpriteFloatingPointVariableAtomType = 'flov'
};
enum {
    kTargetMovie                = 'moov', /* no data */
    kTargetMovieName            = 'mona', /* (PString movieName) */
    kTargetMovieID              = 'moid', /* (long movieID) */
    kTargetRootMovie            = 'moro', /* no data */
    kTargetParentMovie          = 'mopa', /* no data */
    kTargetChildMovieTrackName  = 'motn', /* (PString childMovieTrackName) */
    kTargetChildMovieTrackID    = 'moti', /* (long childMovieTrackID) */
    kTargetChildMovieTrackIndex = 'motx', /* (long childMovieTrackIndex) */
    kTargetChildMovieMovieName  = 'momn', /* (PString childMovieName) */
    kTargetChildMovieMovieID    = 'momi', /* (long childMovieID) */
    kTargetTrackName            = 'trna', /* (PString trackName) */
    kTargetTrackID              = 'trid', /* (long trackID) */
    kTargetTrackType            = 'trty', /* (OSType trackType) */
    kTargetTrackIndex           = 'trin', /* (long trackIndex) */
    kTargetSpriteName           = 'spna', /* (PString spriteName) */
    kTargetSpriteID             = 'spid', /* (QTAtomID spriteID) */
    kTargetSpriteIndex          = 'spin', /* (short spriteIndex) */
    kTargetQD3DNamedObjectName  = 'nana', /* (CString objectName) */
    kTargetCurrentQTEventParams = 'evpa' /* no data */
};
enum {
    kTrackModifierInput        = 0x696E, /* is really 'in'*/
    kTrackModifierType         = 0x7479, /* is really 'ty'*/
    kTrackModifierReference     = 'ssrc',
    kTrackModifierObjectID     = 'obid',
    kTrackModifierInputName    = 'name'
};
enum {
    kTrackPropertyMediaType    = 'mtyp', /* OSType*/
    kTrackPropertyInstantiation = 'inst' /* MovieMediaInstantiationInfoRecord*/
};
enum {
    kTrackReferenceChapterList = 'chap',
    kTrackReferenceTimeCode    = 'tmcd',
    kTrackReferenceModifier    = 'ssrc'
};
enum {
    kTweenEntry                = 'twen',
    kTweenData                  = 'data',
    kTweenType                  = 'twnt',
    kTweenStartOffset          = 'twst',
    kTweenDuration              = 'twdu',

```

```

kTweenFlags                = 'flag',
kTweenOutputMin            = 'omin',
kTweenOutputMax           = 'omax',
kTweenSequenceElement     = 'seque',
kTween3dInitialCondition  = 'icnd',
kTweenInterpolationID     = 'intr',
kTweenRegionData          = 'qdrg',
kTweenPictureData         = 'PICT',
kListElementType          = 'type',
kListElementDataType      = 'daty',
kNameAtom                 = 'name',
kInitialRotationAtom      = 'inro',
kNonLinearTweenHeader     = 'nlth'
};
enum {
    MovieAID                = 'moov',
    MovieHeaderAID          = 'mvhd',
    ClipAID                 = 'clip',
    RgnClipAID              = 'crgn',
    MatteAID                = 'matt',
    MatteCompAID            = 'kmat',
    TrackAID                = 'trak',
    UserDataAID             = 'udta',
    TrackHeaderAID          = 'tkhd',
    EditsAID                = 'edts',
    EditListAID             = 'elst',
    MediaAID                = 'mdia',
    MediaHeaderAID          = 'mdhd',
    MediaInfoAID            = 'minf',
    VideoMediaInfoHeaderAID = 'vmhd',
    SoundMediaInfoHeaderAID = 'smhd',
    GenericMediaInfoHeaderAID = 'gmhd',
    GenericMediaInfoAID     = 'gmin',
    DataInfoAID             = 'dinf',
    DataRefAID              = 'dref',
    SampleTableAID          = 'stbl',
    STSampleDescAID         = 'stsd',
    STTimeToSampAID         = 'stts',
    STSyncSampleAID         = 'stss',
    STSampleToChunkAID     = 'stsc',
    STShadowSyncAID        = 'stsh',
    HandlerAID              = 'hdlr',
    STSampleSizeAID         = 'stsz',
    STChunkOffsetAID        = 'stco',
    STChunkOffset64AID     = 'co64',
    STSampleIDAID           = 'stid',
    STCompositionOffsetAID  = 'ctts',
    STSampleDependencyAID   = 'sdtP',
    STCompositionShiftLeastGreatestAID = 'cslg',
    STPartialSyncSampleAID  = 'stps',
    DataRefContainerAID     = 'drfc',
    TrackReferenceAID       = 'tref',
    ColorTableAID           = 'ctab',
    LoadSettingsAID        = 'load',
    PropertyAtomAID         = 'code',
    InputMapAID             = 'imap',
    MovieBufferHintsAID     = 'mbfh',
    MovieDataRefAliasAID    = 'mdra',

```

```

SoundLocalizationAID          = 'sloc',
CompressedMovieAID            = 'cmov',
CompressedMovieDataAID        = 'cmvd',
DataCompressionAtomAID        = 'dcom',
ReferenceMovieRecordAID        = 'rmra',
ReferenceMovieDescriptorAID    = 'rmda',
ReferenceMovieDataRefAID       = 'rdrf',
ReferenceMovieVersionCheckAID = 'rmvc',
ReferenceMovieDataRateAID      = 'rmdr',
ReferenceMovieComponentCheckAID = 'rmcd',
ReferenceMovieQualityAID       = 'rmqu',
ReferenceMovieLanguageAID      = 'rmla',
ReferenceMovieCPURatingAID     = 'rmcs',
ReferenceMovieAlternateGroupAID = 'rmag',
ReferenceMovieNetworkStatusAID = 'rnet',
CloneMediaAID                 = 'clon',
FileTypeAID                   = 'ftyp',
SecureContentInfoAID          = 'sinf',
SecureContentSchemeTypeAID    = 'schm',
SecureContentSchemeInfoAID    = 'schi'
};
enum {
    MovieResourceAtomType      = 'moov',
    MovieDataAtomType          = 'mdat',
    FreeAtomType               = 'free',
    SkipAtomType               = 'skip',
    WideAtomPlaceholderType    = 'wide'
};
enum {
    quickTimeImageFileImageDescriptionAtom = 'idsc',
    quickTimeImageFileImageDataAtom = 'idat',
    quickTimeImageFileMetaDataAtom = 'meta',
    quickTimeImageFileColorSyncProfileAtom = 'iicc'
};

```

Constants

`kMovieMediaDefaultDataReferenceID`

Atom id.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kMovieMediaSlaveTime`

Boolean.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kMovieMediaSlaveGraphicsMode`

Boolean.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kMovieMediaBackgroundColor`

RGBColor..

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

- `kMovieMediaPrerollTime`
 SInt32 indicating preroll time.
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.
- `kQTParseTextHREFText`
 String.
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.
- `kQTEnableExif`
 UInt8 (Boolean).
 Available in Mac OS X v10.1 and later.
 Declared in `ImageCompression.h`.
- `kTargetChildMovieTrackIndex`
 (long childMovieTrackIndex).
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.
- `kTargetChildMovieMovieName`
 (PString childMovieName).
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.
- `kTargetTrackType`
 (OSType trackType).
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.
- `kTargetTrackIndex`
 (long trackIndex).
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.
- `kTargetSpriteName`
 (PString spriteName).
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.
- `kTargetSpriteID`
 (QTAtomID spriteID).
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.
- `kTargetQD3DNamedObjectName`
 (CString objectName).
 Available in Mac OS X v10.0 and later.
 Declared in `Movies.h`.

kTargetCurrentQTEventParams

No data.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

FCompressImage Values

Constants passed to `FCompressImage`.

```
enum {
    codecFlagUseImageBuffer          = (1L << 0), /* decompress*/
    codecFlagUseScreenBuffer         = (1L << 1), /* decompress*/
    codecFlagUpdatePrevious          = (1L << 2), /* compress*/
    codecFlagNoScreenUpdate          = (1L << 3), /* decompress*/
    codecFlagWasCompressed           = (1L << 4), /* compress*/
    codecFlagDontOffscreen           = (1L << 5), /* decompress*/
    codecFlagUpdatePreviousComp      = (1L << 6), /* compress*/
    codecFlagForceKeyFrame           = (1L << 7), /* compress*/
    codecFlagOnlyScreenUpdate        = (1L << 8), /* decompress*/
    codecFlagLiveGrab                = (1L << 9), /* compress*/
    codecFlagDiffFrame               = (1L << 9), /* decompress*/
    codecFlagDontUseNewImageBuffer   = (1L << 10), /* decompress*/
    codecFlagInterlaceUpdate         = (1L << 11), /* decompress*/
    codecFlagCatchUpDiff             = (1L << 12), /* decompress*/
    codecFlagSupportDisable          = (1L << 13), /* decompress*/
    codecFlagReenable                = (1L << 14) /* decompress*/
};
```

Constants

codecFlagUpdatePrevious

Controls whether your compressor updates the previous image during compression. This flag is only used with sequences that are being temporally compressed. If this flag is set to 1, your compressor should copy the current frame into the previous frame buffer at the end of the frame-compression sequence. Use the source image.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

codecFlagWasCompressed

Indicates to your compressor that the image to be compressed has been compressed before. This information may be useful to compressors that can compensate for the image degradation that may otherwise result from repeated compression and decompression of the same image. This flag is set to 1 to indicate that the image was previously compressed. This flag is set to 0 if the image was not previously compressed.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecFlagUpdatePreviousComp`

Controls whether your compressor updates the previous image buffer with the compressed image. This flag is only used with temporal compression. If this flag is set to 1, your compressor should update the previous frame buffer at the end of the frame-compression sequence, allowing your compressor to perform frame differencing against the compression results. Use the image that results from the compression operation. If this flag is set to 0, your compressor should not modify the previous frame buffer during compression.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecFlagLiveGrab`

Indicates whether the current sequence results from grabbing live video. When working with live video, your compressor should operate as quickly as possible and disable any additional processing, such as compensation for previously compressed data. This flag is set to 1 when you are compressing from a live video source.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecFlagDiffFrame`

Decompress.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecFlagSupportDisable`

Decompress.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

Codec Flags

Constants that represent codec flags.

```

enum {
    codecCompletionSource          = (1 << 0), /* asynchronous codec is done with
source data */
    codecCompletionDest           = (1 << 1), /* asynchronous codec is done with
destination data */
    codecCompletionDontUnshield   = (1 << 2), /* on dest complete don't unshield
cursor */
    codecCompletionWentOffscreen  = (1 << 3), /* codec used offscreen buffer */
    codecCompletionUnlockBits     = (1 << 4), /* on dest complete, call
ICMSequenceUnlockBits */
    codecCompletionForceChainFlush = (1 << 5), /* ICM needs to flush the whole chain
*/
    codecCompletionDropped        = (1 << 6), /* codec decided to drop this frame */
    codecCompletionDecoded        = (1 << 10), /* codec has decoded this frame; if
it is cancelled and rescheduled, set icmFrameAlreadyDecoded in
ICMFrameTimeRecord.flags */
    codecCompletionNotDisplayable = (1 << 11), /* the frame may still be scheduled
for decode, but will not be able to be displayed because the buffer containing it
will need to be recycled to display earlier frames. */
    codecCompletionNotDrawn      = (1 << 12) /* set in conjunction with
codecCompletionDest to indicate that the frame was not drawn */
};
enum {
    codecFlagOutUpdateOnNextIdle  = (1L << 9),
    codecFlagOutUpdateOnDataSourceChange = (1L << 10),
    codecFlagSequenceSensitive    = (1L << 11),
    codecFlagOutUpdateOnTimeChange = (1L << 12),
    codecFlagImageBufferNotSourceImage = (1L << 13),
    codecFlagUsedNewImageBuffer   = (1L << 14),
    codecFlagUsedImageBuffer      = (1L << 15)
};
enum {
    codecInfoDoes1                = (1L << 0), /* codec can work with 1-bit pixels
*/
    codecInfoDoes2                = (1L << 1), /* codec can work with 2-bit pixels
*/
    codecInfoDoes4                = (1L << 2), /* codec can work with 4-bit pixels
*/
    codecInfoDoes8                = (1L << 3), /* codec can work with 8-bit pixels
*/
    codecInfoDoes16               = (1L << 4), /* codec can work with 16-bit pixels
*/
    codecInfoDoes32               = (1L << 5), /* codec can work with 32-bit pixels
*/
    codecInfoDoesDither           = (1L << 6), /* codec can do ditherMode */
    codecInfoDoesStretch          = (1L << 7), /* codec can stretch to arbitrary
sizes */
    codecInfoDoesShrink           = (1L << 8), /* codec can shrink to arbitrary sizes
*/
    codecInfoDoesMask             = (1L << 9), /* codec can mask to clipping regions
*/
    codecInfoDoesTemporal         = (1L << 10), /* codec can handle temporal redundancy
*/
    codecInfoDoesDouble           = (1L << 11), /* codec can stretch to double size
exactly */
    codecInfoDoesQuad             = (1L << 12), /* codec can stretch to quadruple
size exactly */
    codecInfoDoesHalf             = (1L << 13), /* codec can shrink to half size */
};

```



```

    codecInfoDoesQuarter          = (1L << 14), /* codec can shrink to quarter size
    */
    codecInfoDoesRotate           = (1L << 15), /* codec can rotate on decompress */
    codecInfoDoesHorizFlip        = (1L << 16), /* codec can flip horizontally on
    decompress */
    codecInfoDoesVertFlip         = (1L << 17), /* codec can flip vertically on
    decompress */
    codecInfoHasEffectParameterList = (1L << 18), /* codec implements get effects
    parameter list call, once was codecInfoDoesSkew */
    codecInfoDoesBlend            = (1L << 19), /* codec can blend on decompress */
    codecInfoDoesReorder          = (1L << 19), /* codec can rearrange frames during
    compression */
    codecInfoDoesWarp             = (1L << 20), /* codec can warp arbitrarily on
    decompress */
    codecInfoDoesMultiPass        = (1L << 20), /* codec can perform multi-pass
    compression */
    codecInfoDoesRecompress       = (1L << 21), /* codec can recompress image without
    accumulating errors */
    codecInfoDoesSpool            = (1L << 22), /* codec can spool image data */
    codecInfoDoesRateConstrain    = (1L << 23) /* codec can data rate constrain */
};
enum {
    codecLockBitsShieldCursor     = (1 << 0) /* shield cursor */
};

```

Constants

`codecCompletionSource`

The Image Compression Manager is done with the source buffer. The Image Compression Manager sets this flag to 1 when it is done with the processing associated with the source buffer. For compression operations, the source is the uncompressed pixel map you are compressing. For decompression operations, the source is the decompressed data you are decompressing.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecCompletionDest`

The Image Compression Manager is done with the destination buffer. The Image Compression Manager sets this flag to 1 when it is done with the processing associated with the destination buffer.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecCompletionWentOffscreen`

Codec used offscreen buffer.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecCompletionUnlockBits`

On dest complete, call `ICMSequenceUnlockBits`.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecCompletionForceChainFlush`

ICM needs to flush the whole chain.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecCompletionDropped`

Codec decided to drop this frame.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecCompletionDecoded`

Codec has decoded this frame; if it is cancelled and rescheduled, set `icmFrameAlreadyDecoded` in `ICMFrameTimeRecord.flags`.

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

`codecCompletionNotDisplayable`

The frame may still be scheduled for decode, but will not be able to be displayed because the buffer containing it will need to be recycled to display earlier frames..

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

`codecCompletionNotDrawn`

Set in conjunction with `codecCompletionDest` to indicate that the frame was not drawn.

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

`codecFlagUsedImageBuffer`

Indicates to your application that the decompressor used the offscreen image buffer for this frame. If this flag is set to 1, the decompressor used the image buffer. If this flag is set to 0, the decompressor did not use the image buffer.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoes1`

Codec can work with 1-bit pixels.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoes2`

Codec can work with 2-bit pixels.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoes4`

Codec can work with 4-bit pixels.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoes8`

Codec can work with 8-bit pixels.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoes16`

Codec can work with 16-bit pixels.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoes32`

Codec can work with 32-bit pixels.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesDither`

Codec can dither images.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesStretch`

Codec can stretch images to arbitrary sizes.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesShrink`

Codec can shrink images to arbitrary sizes.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesMask`

Codec can mask images to clipping regions.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesTemporal`

Codec can handle temporal redundancy.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesDouble`

Codec can stretch images to exactly double size.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesQuad`

Codec can stretch images to exactly quadruple size.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesHalf`

Codec can shrink images to exactly half size.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

- `codecInfoDoesQuarter`
Codec can shrink images to exactly quarter size.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.
- `codecInfoDoesRotate`
Codec can rotate images during decompression.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.
- `codecInfoDoesHorizFlip`
Codec can flip images horizontally during decompression.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.
- `codecInfoDoesVertFlip`
Codec can flip images vertically during decompression.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.
- `codecInfoHasEffectParameterList`
Codec implements `QTGetEffectsList`.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.
- `codecInfoDoesBlend`
Codec can blend image during decompression.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.
- `codecInfoDoesReorder`
Codec can rearrange frames during compression.
Available in Mac OS X v10.3 and later.
Declared in `ImageCompression.h`.
- `codecInfoDoesWarp`
Codec can warp image arbitrarily during decompression.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.
- `codecInfoDoesMultiPass`
Codec can perform multi-pass compression.
Available in Mac OS X v10.3 and later.
Declared in `ImageCompression.h`.
- `codecInfoDoesRecompress`
Codec can recompress image without accumulating errors.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.

`codecInfoDoesSpool`

Codec can spool image data.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

VDSetCompression Values

Constants passed to `VDSetCompression`.

```
enum {  
    codecLosslessQuality      = 0x00000400,  
    codecMaxQuality           = 0x000003FF,  
    codecMinQuality           = 0x00000000,  
    codecLowQuality           = 0x00000100,  
    codecNormalQuality        = 0x00000200,  
    codecHighQuality          = 0x00000300  
};
```

Constants

`codecLosslessQuality`

Lossless compression or decompression. This special value is valid only for components that can support lossless compression or decompression.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecMaxQuality`

The maximum standard value.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecMinQuality`

The minimum valid value.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecLowQuality`

Low-quality image reproduction. This value should correspond to the lowest image quality that still results in acceptable display characteristics.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecNormalQuality`

Image reproduction of normal quality.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

CodecInfo Values

Constants passed to `CodecInfo`.

```

enum {
    codecInfoDepth1          = (1L << 0), /* compressed data at 1 bpp depth
available */
    codecInfoDepth2          = (1L << 1), /* compressed data at 2 bpp depth
available */
    codecInfoDepth4          = (1L << 2), /* compressed data at 4 bpp depth
available */
    codecInfoDepth8          = (1L << 3), /* compressed data at 8 bpp depth
available */
    codecInfoDepth16         = (1L << 4), /* compressed data at 16 bpp depth
available */
    codecInfoDepth32         = (1L << 5), /* compressed data at 32 bpp depth
available */
    codecInfoDepth24         = (1L << 6), /* compressed data at 24 bpp depth
available */
    codecInfoDepth33         = (1L << 7), /* compressed data at 1 bpp monochrome
depth available */
    codecInfoDepth34         = (1L << 8), /* compressed data at 2 bpp grayscale
depth available */
    codecInfoDepth36         = (1L << 9), /* compressed data at 4 bpp grayscale
depth available */
    codecInfoDepth40         = (1L << 10), /* compressed data at 8 bpp grayscale
depth available */
    codecInfoStoresClut      = (1L << 11), /* compressed data can have custom
cluts */
    codecInfoDoesLossless    = (1L << 12), /* compressed data can be stored in
lossless format */
    codecInfoSequenceSensitive = (1L << 13) /* compressed data is sensitive to
out of sequence decoding */
};

```

Constants

```

codecInfoDepth1
    Compressed data available at 1 bit-per-pixel depth.
    Available in Mac OS X v10.0 and later.
    Declared in ImageCompression.h.

codecInfoDepth2
    Compressed data available at 2 bit-per-pixel depth.
    Available in Mac OS X v10.0 and later.
    Declared in ImageCompression.h.

codecInfoDepth4
    Compressed data available at 4 bit-per-pixel depth.
    Available in Mac OS X v10.0 and later.
    Declared in ImageCompression.h.

codecInfoDepth8
    Compressed data available at 8 bit-per-pixel depth.
    Available in Mac OS X v10.0 and later.
    Declared in ImageCompression.h.

```

`codecInfoDepth16`

Compressed data available at 16 bit-per-pixel depth.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDepth32`

Compressed data available at 32 bit-per-pixel depth.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDepth24`

Compressed data available at 24 bit-per-pixel depth.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDepth33`

Compressed data available at 1 bit-per-pixel monochrome depth.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDepth34`

Compressed data available at 2 bit-per-pixel grayscale depth.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDepth36`

Compressed data available at 4 bit-per-pixel grayscale depth.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDepth40`

Compressed data available at 8 bit-per-pixel grayscale depth.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoStoresClut`

Compressed data can have custom color lookup tables.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`codecInfoDoesLossless`

Compressed data can be stored in lossless format.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

CreateMovieFile Values

Constants passed to `CreateMovieFile`.

```
enum {
    createMovieFileDeleteCurFile = 1L << 31,
    createMovieFileDontCreateMovie = 1L << 30,
    createMovieFileDontOpenFile = 1L << 29,
    createMovieFileDontCreateResFile = 1L << 28
};
```

Constants

`createMovieFileDontOpenFile`

Controls whether the function opens the new movie file. If you set this flag to 1, the Movie Toolbox does not open the new movie file. In this case, the function ignores the `outDataHandler` parameter. If you set this flag to 0, the Movie Toolbox opens the new movie file and returns its reference number into the field referenced by `outDataHandler`.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

FlattenMovieData Values

Constants passed to `FlattenMovieData`.

```
enum {
    flattenAddMovieToDataFork = 1L << 0,
    flattenActiveTracksOnly = 1L << 2,
    flattenDontInterleaveFlatten = 1L << 3,
    flattenFSSpecPtrIsDataRefRecordPtr = 1L << 4,
    flattenCompressMovieResource = 1L << 5,
    flattenForceMovieResourceBeforeMovieData = 1L << 6
};
```

ICM Preferences and Flags

Constants that represent the flags and preferences for ICM sessions.


```

enum {
    icmFrameTimeHasVirtualStartTimeAndDuration = 1 << 0,
    icmFrameAlreadyDecoded                    = 1 << 1,
    icmFrameTimeIsNonScheduledDisplayTime    = 1 << 2,
    icmFrameTimeHasDecodeTime                = 1 << 3,
    icmFrameTimeDecodeImmediately           = 1 << 4,
    icmFrameTimeDoNotDisplay                 = 1 << 5
};
enum {
    kICMGetChainUltimateParent               = 0,
    kICMGetChainParent                      = 1,
    kICMGetChainChild                       = 2,
    kICMGetChainUltimateChild               = 3
};
enum {
    kICMImageBufferNoPreference             = 0,
    kICMImageBufferPreferMainMemory        = 1,
    kICMImageBufferPreferVideoMemory       = 2
};
enum {
    kICMNoDeinterlacing                    = 0,
    kICMDeinterlaceFields                  = 1
};
enum {
    kICMPixelFormatIsPlanarMask            = 0x0F, /* these bits in formatFlags indicate how
many planes there are; they're 0 if chunky*/
    kICMPixelFormatIsIndexed                = (1L << 4),
    kICMPixelFormatIsSupportedByQD         = (1L << 5),
    kICMPixelFormatIsMonochrome            = (1L << 6),
    kICMPixelFormatHasAlphaChannel         = (1L << 7)
};
enum {
    kICMSequenceTaskWeight                  = 'twel', /* data is pointer to UInt32*/
    kICMSequenceTaskName                    = 'tnam', /* data is pointer to OSType*/
    kICMSequenceUserPreferredCodecs        = 'punc' /* data is pointer to
CodecComponentHandle*/
};
enum {
    kICMTempThenAppMemory                   = 1L << 12,
    kICMAppThenTempMemory                   = 1L << 13
};

```

Constants

`icmFrameTimeHasVirtualStartTimeAndDuration`
Indicates that `virtualStartTime` and `virtualDuration` are valid.
Available in Mac OS X v10.0 and later.
Declared in `ImageCompression.h`.

`icmFrameTimeHasDecodeTime`
Indicates that `decodeTime` is valid.
Available in Mac OS X v10.3 and later.
Declared in `ImageCompression.h`.

`kICMPixelFormatIsPlanarMask`

If this flag is 1, the pixel format is a planar mask and `bitsPerPixel[]` represents the bits for each pixel component. If this flag is 0, the pixel format is chunky (not planar) and `bitsPerPixel[0]` represents the bits per pixel. Chunky pixel formats pack the different components together. For example, 3 pixels of 32-bit ARGB is represented in memory as ARGBARGBARGB. Planar formats pack the different components separately. If the pixel format is planar, then `(formatFlags & kICMPixelFormatIsPlanarMask)` is equal to the number of components.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`kICMPixelFormatIsIndexed`

If the pixel format is indexed (which, by definition, means that there are no individual components) then this flag is 1. Generally, color modes of 8 bit per pixel or less are indexed.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`kICMPixelFormatIsSupportedByQD`

If this flag is 1, you can call `QuickDraw` on `Pixmap` structures that store this kind of pixel data. With Macintosh, the classic QD pixel formats will have this set, but not any of the YUV pixel formats. With Windows, more formats will have this set, because the Windows implementation of `QuickDraw` needs to support more pixel formats.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`kICMSequenceUserPreferredCodecs`

Data is pointer to `CodecComponentHandle`.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

ImageFieldSequenceExtractCombine Values

Constants passed to `ImageFieldSequenceExtractCombine`.

```
enum {
    evenField1ToEvenFieldOut    = 1 << 0,
    evenField1ToOddFieldOut     = 1 << 1,
    oddField1ToEvenFieldOut     = 1 << 2,
    oddField1ToOddFieldOut      = 1 << 3,
    evenField2ToEvenFieldOut    = 1 << 4,
    evenField2ToOddFieldOut     = 1 << 5,
    oddField2ToEvenFieldOut     = 1 << 6,
    oddField2ToOddFieldOut      = 1 << 7
};
```

QTSetComponentProperty Values

Constants passed to `QTSetComponentProperty`.

```

enum {
    kComponentPropertyCacheFlagNotPersistent = (1L << 0), /* property metadata should
not be saved in persistent cache*/
    kComponentPropertyCacheFlagIsDynamic = (1L << 1) /* property metadata should not
cached at all*/
};
enum {
    kComponentPropertyClassPropertyInfo = 'pnfo', /* property info class */
                                     /* property info property IDs */
    kComponentPropertyInfoList = 'list', /* array of ComponentPropertyInfo (CFData),
one for each property */
    kComponentPropertyCacheSeed = 'seed', /* property cache seed value */
    kComponentPropertyCacheFlags = 'flgs', /* see kComponentPropertyCache flags */
    kComponentPropertyExtendedInfo = 'meta' /* CFDictionary with extended property
information*/
};

```

Constants

`kComponentPropertyCacheFlagNotPersistent`

Property metadata should not be saved in persistent cache.

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

`kComponentPropertyCacheFlagIsDynamic`

Property metadata should not be cached at all.

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

`kComponentPropertyClassPropertyInfo`

A `QTComponentPropertyInfo` structure that defines a property information class. Also 'pnfo'.

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

`kComponentPropertyInfoList`

An array of `QTComponentPropertyInfo` structures, one for each property. Also 'list'.

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

`kComponentPropertyCacheSeed`

A component property cache seed value. Also 'seed'.

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

`kComponentPropertyCacheFlags`

One of the following two flags: Also 'flgs'.

Available in Mac OS X v10.3 and later.

Declared in `ImageCompression.h`.

kDataHCanRead

Constants grouped with `kDataHCanRead`.

```
enum {
    kDataHCanRead           = 1L << 0,
    kDataHSpecialRead      = 1L << 1,
    kDataHSpecialReadFile  = 1L << 2,
    kDataHCanWrite         = 1L << 3,
    kDataHSpecialWrite     = 1 << 4,
    kDataHSpecialWriteFile = 1 << 5,
    kDataHCanStreamingWrite = 1 << 6,
    kDataHMustCheckDataRef = 1 << 7
};
```

Constants`kDataHCanRead`

Indicates that your data handler can read from the volume.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

`kDataHSpecialRead`

Indicates that your data handler can read from the volume using a specialized method. For example, your data handler might support access to networked multimedia servers using a special protocol. In that case, your component would set this flag to 1 whenever the volume resides on a supported server.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

`kDataHSpecialReadFile`

Reserved for use by Apple.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

`kDataHCanWrite`

Indicates that your data handler can write data to the volume. In particular, use this flag to indicate that your data handler's `DataHPutData` function will work with this volume.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

`kDataHSpecialWrite`

Indicates that your data handler can write to the volume using a specialized method. As with the `kDataHSpecialRead` flag, your data handler would use this flag to indicate that your component can access the volume using specialized support (for example, special network protocols).

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

`kDataHCanStreamingWrite`

Indicates that your data handler can support the special write functions for capturing movie data when writing to this volume.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

QTVRWrapAndConstrain Values

Constants passed to `QTVRWrapAndConstrain`.

```
enum {  
    kQTVRPan                = 0,  
    kQTVRTilt               = 1,  
    kQTVRFieldOfView       = 2,  
    kQTVRViewCenterH       = 4,    /* WrapAndConstrain only*/  
    kQTVRViewCenterV       = 5,    /* WrapAndConstrain only*/  
};
```

Sprite Properties

Constants that represent the properties of sprites.

```

enum {
    kGetSpriteWorldInvalidRegionAndLeaveIntact = -1L,
    kGetSpriteWorldInvalidRegionAndThenSetEmpty = -2L
};
enum {
    kKeyFrameAndSingleOverride = 1L << 1,
    kKeyFrameAndAllOverrides = 1L << 2
};
enum {
    kNoQTIdleEvents = -1
};
enum {
    kOnlyDrawToSpriteWorld = 1L << 0,
    kSpriteWorldPreflight = 1L << 1
};
enum {
    kScaleSpritesToScaleWorld = 1L << 1,
    kSpriteWorldHighQuality = 1L << 2,
    kSpriteWorldDontAutoInvalidate = 1L << 3,
    kSpriteWorldInvisible = 1L << 4,
    kSpriteWorldDirtyInsteadOfFlush = 1L << 5
};
enum {
    kSpritePropertyMatrix = 1,
    kSpritePropertyImageDescription = 2,
    kSpritePropertyImageDataPtr = 3,
    kSpritePropertyVisible = 4,
    kSpritePropertyLayer = 5,
    kSpritePropertyGraphicsMode = 6,
    kSpritePropertyImageDataSize = 7,
    kSpritePropertyActionHandlingSpriteID = 8,
    kSpritePropertyCanBeHitTested = 9,
    kSpritePropertyImageIndex = 100,
    kSpriteTrackPropertyBackgroundColor = 101,
    kSpriteTrackPropertyOffscreenBitDepth = 102,
    kSpriteTrackPropertySampleFormat = 103,
    kSpriteTrackPropertyScaleSpritesToScaleWorld = 104,
    kSpriteTrackPropertyHasActions = 105,
    kSpriteTrackPropertyVisible = 106,
    kSpriteTrackPropertyQTIdleEventsFrequency = 107,
    kSpriteTrackPropertyAllSpritesHitTestingMode = 108,
    kSpriteTrackPropertyPreferredDepthInterpretationMode = 109,
    kSpriteImagePropertyRegistrationPoint = 1000,
    kSpriteImagePropertyGroupID = 1001
};

```

QTSampleTableGetSampleFlags Values

Constants passed to QTSampleTableGetSampleFlags.

```

enum {
    mediaSampleNotSync          = 1 << 0, /* sample is not a sync sample (eg. is
frame differenced */
    mediaSampleShadowSync      = 1 << 1, /* sample is a shadow sync */
    mediaSampleDroppable       = 1 << 27, /* sample is not required to be decoded
for later samples to be decoded properly */
    mediaSamplePartialSync     = 1 << 16, /* sample is a partial sync (e.g., I
frame after open GOP) */
    mediaSampleHasRedundantCoding = 1 << 24, /* sample is known to contain redundant
coding */
    mediaSampleHasNoRedundantCoding = 1 << 25, /* sample is known not to contain
redundant coding */
    mediaSampleIsDependedOnByOthers = 1 << 26, /* one or more other samples depend
upon the decode of this sample */
    mediaSampleIsNotDependedOnByOthers = 1 << 27, /* synonym for mediaSampleDroppable
*/
    mediaSampleDependsOnOthers   = 1 << 28, /* sample's decode depends upon decode
of other samples */
    mediaSampleDoesNotDependOnOthers = 1 << 29, /* sample's decode does not depend
upon decode of other samples */
    mediaSampleEarlierDisplayTimesAllowed = 1 << 30 /* samples later in decode order
may have earlier display times */
};

```

Constants

`mediaSampleNotSync`

Returned for frame-differenced video sample data.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

movieFileSpecValid

Constants grouped with `movieFileSpecValid`.

```

enum {
    pasteInParallel           = 1 << 0,
    showUserSettingsDialog    = 1 << 1,
    movieToFileOnlyExport     = 1 << 2,
    movieFileSpecValid        = 1 << 3
};

```

MovieImportDataRef Values

Constants passed to `MovieImportDataRef`.

```
enum {
    movieImportCreateTrack      = 1,
    movieImportInParallel      = 2,
    movieImportMustUseTrack    = 4,
    movieImportWithIdle        = 16
};
enum {
    movieImportResultUsedMultipleTracks = 8,
    movieImportResultNeedIdles        = 32,
    movieImportResultComplete         = 64
};
```

Constants

movieImportResultNeedIdles

Undocumented

Available in Mac OS X v10.0 and later.

Declared in QuickTimeComponents.h.

MovieProgressProc Values

Constants passed to MovieProgressProc.

```
enum {
    movieProgressOpen          = 0,
    movieProgressUpdatePercent = 1,
    movieProgressClose         = 2
};
enum {
    progressOpFlatten          = 1,
    progressOpInsertTrackSegment = 2,
    progressOpInsertMovieSegment = 3,
    progressOpPaste            = 4,
    progressOpAddMovieSelection = 5,
    progressOpCopy              = 6,
    progressOpCut               = 7,
    progressOpLoadMovieIntoRam  = 8,
    progressOpLoadTrackIntoRam  = 9,
    progressOpLoadMediaIntoRam  = 10,
    progressOpImportMovie       = 11,
    progressOpExportMovie       = 12
};
```

Constants

movieProgressOpen

Indicates the start of a long operation. This is always the first message sent to your function. Your function can use this message to trigger the display of your progress window.

Available in Mac OS X v10.0 and later.

Declared in Movies.h.

`movieProgressUpdatePercent`

Passes completion information to your function. The Movie Toolbox repeatedly sends this message to your function. The `percentDone` parameter indicates the relative completion of the operation. You can use this value to update your progress window.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`movieProgressClose`

Indicates the end of a long operation. This is always the last message sent to your function. Your function can use this message as an indication to remove its progress window.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpFlatten`

Your application has called the `FlattenMovie` or `FlattenMovieData` function.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpInsertTrackSegment`

Your application has called the `InsertTrackSegment` function. The Movie Toolbox calls the progress function that is assigned to the movie that contains the destination track.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpInsertMovieSegment`

Your application has called the `InsertMovieSegment` function. The Movie Toolbox calls the progress function that is assigned to the destination movie.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpPaste`

Your application has called the `PasteMovieSelection` function. The Movie Toolbox calls the progress function that is assigned to the destination movie.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpAddMovieSelection`

Your application has called the `AddMovieSelection` function. The Movie Toolbox calls the progress function that is assigned to the destination movie. The Movie Toolbox calls the progress function that is assigned to the destination movie.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpCopy`

Your application has called the `CopyMovieSelection` function. The Movie Toolbox calls the progress function that is assigned to the destination movie.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpCut`

Your application has called the `CutMovieSelection` function. The Movie Toolbox calls the progress function that is assigned to the destination movie.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpLoadMovieIntoRam`

Your application has called the `LoadMovieIntoRam` function. The Movie Toolbox calls the progress function that is assigned to the destination movie.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpLoadTrackIntoRam`

Your application has called the `LoadTrackIntoRam` function. The Movie Toolbox calls the progress function that is assigned to the destination track.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpLoadMediaIntoRam`

Your application has called the `LoadMediaIntoRam` function. The Movie Toolbox calls the progress function that is assigned to the destination media.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`progressOpImportMovie`

Your application has called the `ConvertFileToMovieFile` function. The Movie Toolbox calls the progress function that is associated with the destination movie file. This flag is also used, as appropriate, for the `PasteHandleIntoMovie` functions.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

New Movie Properties

Constants that represent the properties of new movies.

```
enum {
    newMovieActive                = 1 << 0,
    newMovieDontResolveDataRefs  = 1 << 1,
    newMovieDontAskUnresolvedDataRefs = 1 << 2,
    newMovieDontAutoAlternates   = 1 << 3,
    newMovieDontUpdateForeBackPointers = 1 << 4,
    newMovieDontAutoUpdateClock  = 1 << 5,
    newMovieAsyncOK              = 1 << 8,
    newMovieIdleImportOK        = 1 << 10,
    newMovieDontInteractWithUser = 1 << 11
};
```

NewMovieController Values

Constants passed to `NewMovieController`.

```
enum {
    mcTopLeftMovie          = 1 << 0, /* usually centered */
    mcScaleMovieToFit      = 1 << 1, /* usually only scales down */
    mcWithBadge            = 1 << 2, /* give me a badge */
    mcNotVisible          = 1 << 3, /* don't show controller */
    mcWithFrame           = 1 << 4 /* gimme a frame */
};
```

QuickTime Preferences Dialog Options

Constants that represent options for QuickTime preference dialogs.

```

enum {
    pdActionConfirmDialog          = 1,    /* no param*/
    pdActionSetAppleMenu          = 2,    /* param is MenuRef*/
    pdActionSetEditMenu           = 3,    /* param is MenuRef*/
    pdActionGetDialogValues       = 4,    /* param is QTAtomContainer*/
    pdActionSetPreviewUserItem    = 5,    /* param is long*/
    pdActionSetPreviewPicture     = 6,    /* param is QTParamPreviewPtr;*/
    pdActionSetColorPickerEventProc = 7,  /* param is UserEventUPP*/
    pdActionSetDialogTitle       = 8,    /* param is StringPtr */
    pdActionGetSubPanelMenu      = 9,    /* param is MenuRef* */
    pdActionActivateSubPanel     = 10,   /* param is long */
    pdActionConductStopAlert     = 11,   /* param is StringPtr */
    pdActionModelessCallback     = 12,   /* param is QTParamDialogEventPtr */
    pdActionFetchPreview         = 13,   /* param is QTParamFetchPreviewPtr */
    pdActionSetDialogSettings    = 14,   /* param is QTAtomContainer */
    pdActionGetDialogSettings    = 15,   /* param is QTAtomContainer */
    pdActionGetNextSample        = 16,   /* param is QTAtomContainer with effect
sample to change - createdDialog may be NIL */
    pdActionGetPreviousSample    = 17,   /* param is QTAtomContainer with effect
sample to change - createdDialog may be NIL */
    pdActionCompactSample        = 18,   /* param is QTAtomContainer with effect
sample to compact, - createdDialog may be NIL */
    pdActionSetEditCallout       = 19,   /* param is QTParamPreviewCalloutPtr, can
be NIL */
    pdActionSetSampleTime        = 20,   /* param is QTParamSampleTimePtr, can be
NIL */
    pdActionDoEditCommand        = 21,   /* param is long with menu command (ie,
mcMenuCut etc) */
    pdActionGetSubPanelMenuValue = 22,   /* param is long and returns current
sub-panel value selected by the effect */
    /* Action codes and typedefs used for custom
controls within effects */
    pdActionCustomNewControl      = 23,   /* param is QTCustomControlNewPtr */
    pdActionCustomDisposeControl = 24,   /* param is QTCustomControlNewPtr */
    pdActionCustomPositionControl = 25,  /* param is QTCustomControlPositionControlPtr
*/
    pdActionCustomShowHideControl = 26,  /* param is QTCustomControlShowHideControlPtr
*/
    pdActionCustomHandleEvent     = 27,  /* param is QTCustomControlHandleEventPtr
*/
    pdActionCustomSetFocus        = 28,  /* param is QTCustomControlSetFocusPtr */
    pdActionCustomSetEditMenu     = 29,  /* param is QTCustomControlSetEditMenuPtr
*/
    pdActionCustomSetPreviewPicture = 30, /* param is
QTCustomControlSetPreviewPicturePtr */
    pdActionCustomSetEditCallout  = 31,  /* param is QTCustomControlSetEditCalloutPtr
*/
    pdActionCustomGetEnableValue  = 32,  /* param is QTCustomControlGetEnableValuePtr
*/
    pdActionCustomSetSampleTime   = 33,  /* param is QTCustomControlSetSampleTimePtr
*/
    pdActionCustomGetValue        = 34,  /* param is QTCustomControlGetValue */
    pdActionCustomDoEditCommand   = 35,  /* param is QTCustomControlDoEditCommand
*/
    /* more actions for the dialog */
    pdActionRunInEventLoop       = 36,  /* param is QTEventLoopDescriptionPtr - OS
X only*/
    pdActionConvertSettingsToXML  = 37,  /* param is QTAtomContainer* inbound,

```

```

Handle* outbound contains the XML - createdDialog may be NIL */
pdActionConvertSettingsToXMLWithComments = 38, /* param is QAtomContainer*
inbound, Handle* outbound contains the XML with comments - createdDialog may be
NIL */
pdActionConvertSettingsToText = 39, /* param is QAtomContainer* inbound,
Handle* outbound contains human readable text - createdDialog may be NIL */
pdActionConvertXMLToSettings = 40, /* param is Handle* inbound, QAtomContainer*
outbound contains parameters - createdDialog may be NIL */
pdActionSetPropertyComponent = 41 /* param is QTParamComponentPropertyPtr */
};
enum {
pdOptionsCollectOneValue = 0x00000001, /* should collect a single value
only*/
pdOptionsAllowOptionalInterpolations = 0x00000002, /* non-novice interpolation
options are shown */
pdOptionsModalDialogBox = 0x00000004, /* dialog box should be modal */
pdOptionsEditCurrentEffectOnly = 0x00000008, /* List of effects will not be shown
*/
pdOptionsHidePreview = 0x00000010, /* Preview item will not be shown */
pdOptionsDisplayAsSheet = 0x00000020 /* Dialog will be used as a sheet (on
platforms that support it) */
};
enum {
pdSampleTimeDisplayOptionsNone = 0x00000000
};

```

Constants

pdActionModelessCallback

Parameter is QTParamDialogEventPtr.

Available in Mac OS X v10.0 and later.

Declared in Movies.h.

pdActionFetchPreview

Parameter is QTParamFetchPreviewPtr.

Available in Mac OS X v10.0 and later.

Declared in Movies.h.

pdActionSetDialogSettings

Parameter is QAtomContainer.

Available in Mac OS X v10.2 and later.

Declared in Movies.h.

pdActionGetDialogSettings

Parameter is QAtomContainer.

Available in Mac OS X v10.2 and later.

Declared in Movies.h.

pdActionGetNextSample

Parameter is QAtomContainer with effect sample to change - createdDialog may be NIL.

Available in Mac OS X v10.2 and later.

Declared in Movies.h.

`pdActionGetPreviousSample`

Parameter is `QTAtomContainer` with effect sample to change - `createdDialog` may be `NIL`.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionCompactSample`

Parameter is `QTAtomContainer` with effect sample to compact, - `createdDialog` may be `NIL`.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionSetEditCallout`

Parameter is `QTParamPreviewCalloutPtr`, can be `NIL`.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionSetSampleTime`

Parameter is `QTParamSampleTimePtr`, can be `NIL`.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionDoEditCommand`

Parameter is long with menu command (that is, `mcMenuCut` etc).

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionGetSubPanelMenuValue`

Parameter is long and returns current sub-panel value selected by the effect.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionCustomNewControl`

Parameter is `QTCustomControlNewPtr`.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionCustomDisposeControl`

Parameter is `QTCustomControlNewPtr`.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionCustomPositionControl`

Parameter is `QTCustomControlPositionControlPtr`.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdActionCustomShowHideControl`

Parameter is `QTCustomControlShowHideControlPtr`.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

- `pdActionCustomHandleEvent`
Parameter is `QTCustomControlHandleEventPtr`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionCustomSetFocus`
Parameter is `QTCustomControlSetFocusPtr`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionCustomSetEditMenu`
Parameter is `QTCustomControlSetEditMenuPtr`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionCustomSetPreviewPicture`
Parameter is `QTCustomControlSetPreviewPicturePtr`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionCustomSetEditCallout`
Parameter is `QTCustomControlSetEditCalloutPtr`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionCustomGetEnableValue`
Parameter is `QTCustomControlGetEnableValuePtr`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionCustomSetSampleTime`
Parameter is `QTCustomControlSetSampleTimePtr`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionCustomGetValue`
Parameter is `QTCustomControlGetValue`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionCustomDoEditCommand`
Parameter is `QTCustomControlDoEditCommand`.
Available in Mac OS X v10.2 and later.
Declared in `Movies.h`.
- `pdActionRunInEventLoop`
Parameter is `QTEventLoopDescriptionPtr` - OS X only.
Available in Mac OS X v10.3 and later.
Declared in `Movies.h`.

`pdActionConvertSettingsToXML`

Parameter is `QTAtomContainer` inbound, Handle outbound contains the XML - `createdDialog` may be NIL.

Available in Mac OS X v10.3 and later.

Declared in `Movies.h`.

`pdActionConvertSettingsToXMLWithComments`

Parameter is `QTAtomContainer` inbound, Handle outbound contains the XML with comments - `createdDialog` may be NIL.

Available in Mac OS X v10.3 and later.

Declared in `Movies.h`.

`pdActionConvertSettingsToText`

Parameter is `QTAtomContainer` inbound, Handle outbound contains human readable text - `createdDialog` may be NIL.

Available in Mac OS X v10.3 and later.

Declared in `Movies.h`.

`pdActionConvertXMLToSettings`

Parameter is `Handle` inbound, `QTAtomContainer` outbound contains parameters - `createdDialog` may be NIL.

Available in Mac OS X v10.3 and later.

Declared in `Movies.h`.

`pdActionSetPropertyComponent`

Parameter is `QTParamComponentPropertyPtr`.

Available in Mac OS X v10.3 and later.

Declared in `Movies.h`.

`pdOptionsModalDialogBox`

Dialog box should be modal.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`pdOptionsEditCurrentEffectOnly`

List of effects will not be shown.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

`pdOptionsHidePreview`

Preview item will not be shown.

Available in Mac OS X v10.2 and later.

Declared in `Movies.h`.

Standard Compression Constants

Constants that represent constants for Standard Compression.


```

enum {
    /*
     * Indicates the client is ready to use the ICM compression session
     * API to perform compression operations. StdCompression disables
     * frame reordering and multi pass encoding if this flag is cleared.
     */
    scAllowEncodingWithCompressionSession = 1L << 8,
    /*
     * Indicates the client does not want the user to change the frame
     * reordering setting.
     */
    scDisableFrameReorderingItem = 1L << 9,
    /*
     * Indicates the client does not want the user to change the multi
     * pass encoding setting
     */
    scDisableMultiPassEncodingItem = 1L << 10
};
enum {
    /*
     * Specifies if frame reordering can occur in encoding.
     */
    scVideoAllowFrameReorderingType = 'bfra', /* pointer to Boolean*/
    /*
     * The settings to control multi pass encoding.
     */
    scVideoMultiPassEncodingSettingsType = 'mpes' /* pointer to
    SCVideoMutiPassEncodingSettings struct*/
};
enum {
    scListEveryCodec                = 1L << 1,
    scAllowZeroFrameRate            = 1L << 2,
    scAllowZeroKeyFrameRate        = 1L << 3,
    scShowBestDepth                 = 1L << 4,
    scUseMovableModal               = 1L << 5,
    scDisableFrameRateItem          = 1L << 6,
    scShowDataRateAsKilobits       = 1L << 7
};
enum {
    scOKItem                        = 1,
    scCancelItem                    = 2,
    scCustomItem                    = 3
};
enum {
    scPositionRect                  = 2,
    scPositionDialog                = 3,
    scSetTestImagePictHandle        = 4,
    scSetTestImagePictFile          = 5,
    scSetTestImagePictMap           = 6,
    scGetBestDeviceRect              = 7,
    scRequestImageSettings           = 10,
    scCompressImage                  = 11,
    scCompressPicture                = 12,
    scCompressPictureFile            = 13,
    scRequestSequenceSettings        = 14,
    scCompressSequenceBegin          = 15,
    scCompressSequenceFrame         = 16,
    scCompressSequenceEnd            = 17,
};

```

```

    scDefaultPictHandleSettings    = 18,
    scDefaultPictFileSettings     = 19,
    scDefaultPixMapSettings       = 20,
    scGetInfo                     = 21,
    scSetInfo                     = 22,
    scNewGWorld                   = 23
};
enum {
    scPreferCropping              = 1 << 0,
    scPreferScaling               = 1 << 1,
    scPreferScalingAndCropping    = scPreferScaling | scPreferCropping,
    scDontDetermineSettingsFromTestImage = 1 << 2
};
enum {
    scSpatialSettingsType        = 'sptl', /* pointer to SCSpatialSettings struct*/
    scTemporalSettingsType       = 'tprl', /* pointer to SCTemporalSettings struct*/
    scDataRateSettingsType       = 'drat', /* pointer to SCDataRateSettings struct*/
    scColorTableType             = 'clut', /* pointer to CTabHandle*/
    scProgressProcType           = 'prog', /* pointer to ProgressRecord struct*/
    scExtendedProcsType          = 'xprc', /* pointer to SCExtendedProcs struct*/
    scPreferenceFlagsType        = 'pref', /* pointer to long*/
    scSettingsStateType          = 'ssta', /* pointer to Handle*/
    scSequenceIDType             = 'sequ', /* pointer to ImageSequence*/
    scWindowPositionType         = 'wndw', /* pointer to Point*/
    scCodecFlagsType             = 'cflg', /* pointer to CodecFlags*/
    scCodecSettingsType          = 'cdec', /* pointer to Handle*/
    scForceKeyValueType          = 'ksim', /* pointer to long*/
    scCompressionListType        = 'ctyl', /* pointer to OSType Handle*/
    scCodecManufacturerType      = 'cmfr', /* pointer to OSType*/
    scAvailableCompressionListType = 'avai', /* pointer to OSType Handle*/
    scWindowOptionsType          = 'shee', /* pointer to SCWindowSettings struct*/
    scSoundVBRCompressionOK      = 'cvbr', /* pointer to Boolean*/
    scSoundSampleRateChangeOK    = 'rcok', /* pointer to Boolean*/
    scSoundCompressionType       = 'ssct', /* pointer to OSType*/
    scSoundSampleRateType        = 'ssrt', /* pointer to UnsignedFixed*/
    scSoundInputSampleRateType   = 'ssir', /* pointer to UnsignedFixed*/
    scSoundSampleSizeType        = 'ssss', /* pointer to short*/
    scSoundChannelCountType      = 'sscc' /* pointer to short*/
};
enum {
    scTestImageWidth             = 80,
    scTestImageHeight            = 80
};
enum {
    scUserCancelled              = 1
};
enum {
    scWindowRefKindCarbon        = 'carb' /* WindowRef*/
};

```

Constants

scVideoAllowFrameReorderingType

Pointer to Boolean.

Available in Mac OS X v10.3 and later.

Declared in QuickTimeComponents.h.

- `scSpatialSettingsType`
A video track's `SCSpatialSettings` structure.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeComponents.h`.
- `scTemporalSettingsType`
A video track's `SCTemporalSettings` structure.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeComponents.h`.
- `scDataRateSettingsType`
A video track's `SCDataRateSettings` structure.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeComponents.h`.
- `scCodecSettingsType`
Pointer to Handle.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeComponents.h`.
- `scForceKeyValueType`
Pointer to long.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeComponents.h`.
- `scCodecManufacturerType`
Pointer to `OStype`.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeComponents.h`.
- `scAvailableCompressionListType`
Pointer to `OStype` Handle.
Available in Mac OS X v10.2 and later.
Declared in `QuickTimeComponents.h`.
- `scWindowOptionsType`
Pointer to `SCWindowSettings` struct.
Available in Mac OS X v10.3 and later.
Declared in `QuickTimeComponents.h`.
- `scSoundVBRCompressionOK`
Pointer to Boolean.
Available in Mac OS X v10.2 and later.
Declared in `QuickTimeComponents.h`.
- `scSoundSampleRateChangeOK`
Pointer to Boolean.
Available in Mac OS X v10.2 and later.
Declared in `QuickTimeComponents.h`.

`scSoundCompressionType`

A sound track's compression type constant; see Codec Identifiers.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

`scSoundSampleRateType`

An `UnsignedFixed` value that represents a sound track's sampling rate.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

`scSoundInputSampleRateType`

Pointer to `UnsignedFixed`.

Available in Mac OS X v10.2 and later.

Declared in `QuickTimeComponents.h`.

`scSoundSampleSizeType`

A short integer that represents a sound track's sample size.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

`scSoundChannelCountType`

A short integer that represents a sound track's channel count.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeComponents.h`.

SGPanelGetDITLForSize Values

Constants passed to `SGPanelGetDITLForSize`.

```
enum {
    kSGSmallestDITLSize          = -1,    /* requestedSize h and v set to this to
    retrieve small size*/
    kSGLargestDITLSize           = -2     /* requestedSize h and v set to this to
    retrieve large size*/
};
```

Media Identifiers

Identify media types in QuickTime.

```

enum {
    VideoMediaType           = 'vide',
    SoundMediaType          = 'soun',
    TextMediaType           = 'text',
    BaseMediaType           = 'gnrc',
    MPEGMediaType           = 'MPEG',
    MusicMediaType          = 'musi',
    TimeCodeMediaType       = 'tmcd',
    SpriteMediaType         = 'sprt',
    FlashMediaType          = 'flsh',
    MovieMediaType          = 'moov',
    TweenMediaType          = 'twen',
    ThreeDeeMediaType       = 'qd3d',
    SkinMediaType           = 'skin',
    HandleDataHandlerSubType = 'hndl',
    PointerDataHandlerSubType = 'ptr ',
    NullDataHandlerSubType  = 'null',
    ResourceDataHandlerSubType = 'rsrc',
    URLDataHandlerSubType  = 'url ',
    AliasDataHandlerSubType = 'alis',
    WiredActionHandlerType  = 'wire'
};

```

Constants

SoundMediaType

Sound channel.**Available in Mac OS X v10.0 and later.****Declared in** `Movies.h`.

TextMediaType

Text media.**Available in Mac OS X v10.0 and later.****Declared in** `Movies.h`.**SpriteWorldHitTest Values**Constants passed to `SpriteWorldHitTest`.

```
enum {
    spriteHitTestBounds          = 1L << 0, /* point must only be within sprite's
    bounding box*/
    spriteHitTestImage          = 1L << 1, /* point must be within the shape of
    the sprite's image*/
    spriteHitTestInvisibleSprites = 1L << 2, /* invisible sprites may be hit tested*/
    spriteHitTestIsClick        = 1L << 3, /* for codecs that want mouse events*/
    spriteHitTestLocInDisplayCoordinates = 1L << 4, /* set if you want to pass a
    display coordinate point to SpriteHitTest*/
    spriteHitTestTreatAllSpritesAsHitTestable = 1L << 5 /* set if you want to override
    each sprites hittestable property as true*/
};
```

Text Properties

Constants that represent the properties of text.

```

enum {
    /* set property parameter / get property
parameter*/
    kTextTextHandle = 1, /* Handle / preallocated Handle*/
    kTextTextPtr = 2, /* Pointer*/
    kTextTEStyle = 3, /* TextStyle * / TextStyle **/
    kTextTESelection = 4, /* long [2] / long [2]*/
    kTextBackColor = 5, /* RGBColor * / RGBColor **/
    kTextForeColor = 6, /* RGBColor * / RGBColor **/
    kTextFace = 7, /* long / long **/
    kTextFont = 8, /* long / long **/
    kTextSize = 9, /* long / long **/
    kTextAlignment = 10, /* short * / short **/
    kTextHilite = 11, /* hiliteRecord * / hiliteRecord **/
    kTextDropShadow = 12, /* dropShadowRecord * / dropShadowRecord
**/
    kTextDisplayFlags = 13, /* long / long **/
    kTextScroll = 14, /* TimeValue * / TimeValue **/
    kTextRelativeScroll = 15, /* Point **/
    kTextHyperTextFace = 16, /* hyperTextSetFace * / hyperTextSetFace
**/
    kTextHyperTextColor = 17, /* hyperTextSetColor * / hyperTextSetColor
**/
    kTextKeyEntry = 18, /* short*/
    kTextMouseDown = 19, /* Point **/
    kTextTextBox = 20, /* Rect * / Rect **/
    kTextEditState = 21, /* short / short **/
    kTextLength = 22, /* / long **/
};
enum {
    dfDontDisplay = 1 << 0, /* Don't display the text*/
    dfDontAutoScale = 1 << 1, /* Don't scale text as track bounds grows
or shrinks*/
    dfClipToTextBox = 1 << 2, /* Clip update to the textbox*/
    dfUseMovieBGColor = 1 << 3, /* Set text background to movie's
background color*/
    dfShrinkTextBoxToFit = 1 << 4, /* Compute minimum box to fit the sample*/
    dfScrollIn = 1 << 5, /* Scroll text in until last of text is
in view */
    dfScrollOut = 1 << 6, /* Scroll text out until last of text is
gone (if both set, scroll in then out)*/
    dfHorizScroll = 1 << 7, /* Scroll text horizontally (otherwise
it's vertical)*/
    dfReverseScroll = 1 << 8, /* vert: scroll down rather than up;
horiz: scroll backwards (justification dependent)*/
    dfContinuousScroll = 1 << 9, /* new samples cause previous samples to
scroll out */
    dfFlowHoriz = 1 << 10, /* horiz scroll text flows in textbox
rather than extend to right */
    dfContinuousKaraoke = 1 << 11, /* ignore begin offset, hilite everything
up to the end offset(karaoke)*/
    dfDropShadow = 1 << 12, /* display text with a drop shadow */
    dfAntiAlias = 1 << 13, /* attempt to display text anti aliased*/
    dfKeyedText = 1 << 14, /* key the text over background*/
    dfInverseHilite = 1 << 15, /* Use inverse hiliting rather than
using hilite color*/
    dfTextColorHilite = 1 << 16 /* changes text color in place of hiliting.
*/
}

```

};

Constants

`kTextSelection`

Long [2] long [2].

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextScroll`

The text scroll position.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextRelativeScroll`

Point.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextHyperTextFace`

HyperTextSetFace hyperTextSetFace.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextHyperTextColor`

HyperTextSetColor hyperTextSetColor.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextKeyEntry`

Short.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextMouseDown`

Point.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextTextBox`

Rect Rect.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextEditState`

Short short.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`kTextLength`

Long.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfDontDisplay`

Does not display the specified sample.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfDontAutoScale`

Does not scale the text if the track bounds increase.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfClipToTextBox`

Clips to just the text box. This is useful if the text overlays the video.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfUseMovieBGColor`

Set text background to movie's background color.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfShrinkTextBoxToFit`

Recalculates size of the `textBox` parameter to just fit the given text and stores this rectangle with the text data.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfScrollIn`

Scrolls the text in until the last of the text is in view. This flag is associated with the `scrollDelay` parameter.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfScrollOut`

Scrolls text out until the last of the text is out of view. This flag is associated with the `scrollDelay` parameter. If both `dfScrollIn` and `dfScrollOut` are set, the text is scrolled in, then out.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfHorizScroll`

Scrolls a single line of text horizontally. If the `dfHorizScroll` flag is not set, then the scrolling is vertical.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

`dfReverseScroll`

If set, scrolls vertically down, rather than up. If not set, horizontal scrolling proceeds toward the left rather than toward the right.

Available in Mac OS X v10.0 and later.

Declared in `Movies.h`.

ToneDescription Values

Constants passed to ToneDescription.

```
enum {
    kSoftSynthComponentSubType    = 'ss ',
    kGMSynthComponentSubType     = 'gm ',
};
```

Constants

`kSoftSynthComponentSubType`
Software synthesizer; value is 'ss'.
Available in Mac OS X v10.0 and later.
Declared in QuickTimeMusic.h.

Arithmetic and Logical Operator IDs

Constants that identify arithmetic and logical operations.

```
kOperatorAdd           = 'add '
kOperatorSubtract     = 'sub '
kOperatorMultiply     = 'mult'
kOperatorDivide       = 'div '
kOperatorOr           = 'or  '
kOperatorAnd          = 'and '
kOperatorNot          = 'not '
kOperatorLessThan     = '<  '
kOperatorLessThanEqualTo = '<='
kOperatorEqualTo      = '='
kOperatorNotEqualTo   = '!='
kOperatorGreaterThan  = '>  '
kOperatorGreaterThanEqualTo = '>='
kOperatorModulo       = 'mod '
kOperatorIntegerDivide = 'idiv'
kOperatorAbsoluteValue = 'abs '
kOperatorNegate       = 'neg '
```

Codec Identifiers

Identify codec components and data types in QuickTime.

kAnimationCodecType	= 'rle '
kAVRJPEGCodecType	= 'avr '
kBaseCodecType	= 'base '
kBMPCodecType	= 'WRLE'
kCinepakCodecType	= 'cvid'
kCloudCodecType	= 'clou'
kCMYKCodecType	= 'cmyk'
kComponentVideoCodecType	= 'yuv2'
kComponentVideoSigned	= 'yuvu'
kComponentVideoUnsigned	= 'yuvs'
kDVCNTSCCodecType	= 'dvc '
kDVCPALCodecType	= 'dvcp'
kDVCPProNTSCCodecType	= 'dvpn'
kDVCPProPALCodecType	= 'dvpp'
kFireCodecType	= 'fire'
kFLCCCodecType	= 'flic'
k48RGBCodecType	= 'b48r'
kGIFCodecType	= 'gif '
kGraphicsCodecType	= 'smc '
kH261CodecType	= 'h261'
kH263CodecType	= 'h263'
kIndeo4CodecType	= 'IV41'
kJPEGCodecType	= 'jpeg'
kMacPaintCodecType	= 'PNTG'
kMicrosoftVideo1CodecType	= 'msvc'
kMotionJPEGACodecType	= 'mjpa'
kMotionJPEGBCodecType	= 'mjpb'
kMpegYUV420CodecType	= 'myuv'
kOpenDMLJPEGCodecType	= 'dmb1'
kPhotoCDCCodecType	= 'kpcd'
kPlanarRGBCodecType	= '8BPS'
kPNGCodecType	= 'png '
kQuickDrawCodecType	= 'qdrw'
kQuickDrawGXCodecType	= 'qdgx'
kRawCodecType	= 'raw '
kSGICodecType	= ' .SGI'
k16GrayCodecType	= 'b16g'
k64ARGBCodecType	= 'b64a'
kSorensonCodecType	= 'SVQ1'
kSorensonYUV9CodecType	= 'syv9'
kTargaCodecType	= 'tga '
k32AlphaGrayCodecType	= 'b32a'
kTIFFCodecType	= 'tiff'
kVectorCodecType	= 'path'
kVideoCodecType	= 'rpza'
kWaterRippleCodecType	= 'ripl'
kWindowsRawCodecType	= 'WRAW'
kYUV420CodecType	= 'y420'

Discussion

All codec components of the same type provide the same kinds of services and support a common application programming interface.

Codec Properties

Constants that represent the properties of codecs.

```
codecImageBufferIsInPCIMemory = 1L << 5
codecSupportsOutOfOrderDisplayTimes = 1L << 8
codecSupportsScheduledBackwardsPlaybackWithDifferenceFrames = 1L << 9
codecConditionNewMask = 1L << 6
codecInfoResourceType = 'cdci'
codecInterfaceVersion = 2
codecSuggestedBufferSentinel = 'sent'
codecMinimumDataSize = 32768L
```

`codecImageBufferIsInPCIMemory`

Codec image buffer is across a PCI bus; byte writes are bad.

Available in Mac OS X v10.0 and later.

Declared in `ImageCodec.h`.

`codecSupportsOutOfOrderDisplayTimes`

Codec supports frames queued in one order for display in a different order, for example IPB content.

Available in Mac OS X v10.3 and later.

Declared in `ImageCodec.h`.

`codecSupportsScheduledBackwardsPlaybackWithDifferenceFrames`

Codec can use additional buffers to minimize redecoding during backwards playback.

Available in Mac OS X v10.3 and later.

Declared in `ImageCodec.h`.

`codecInfoResourceType`

Codec info resource type.

Available in Mac OS X v10.0 and later.

Declared in `ImageCodec.h`.

`codecInterfaceVersion`

High word returned in component `GetVersion`.

Available in Mac OS X v10.0 and later.

Declared in `ImageCodec.h`.

`codecSuggestedBufferSentinel`

Codec public resource containing suggested data pattern to put past end of data buffer.

Available in Mac OS X v10.2 and later.

Declared in `ImageCodec.h`.

Codec Type Constants

Constants that represent codec types.

```

kDVCPPro50NTSCCodecType      = 'dv5n'
kDVCPPro50PALCodecType       = 'dv5p'
kDVCPPro100NTSCCodecType     = 'dv1n'
kDVCPPro100PALCodecType      = 'dv1p'
kDVCPROHD720pCodecType       = 'dvhp'
kDVCPROHD1080i60CodecType    = 'dvh6'
kDVCPROHD1080i50CodecType    = 'dvh5'
kSorenson3CodecType          = 'SVQ3'
kMPEG4VisualCodecType        = 'mp4v'
k422YpCbCr8CodecType         = '2vuy'
k444YpCbCr8CodecType         = 'v308'
k4444YpCbCrA8CodecType       = 'v408'
k422YpCbCr16CodecType        = 'v216'
k422YpCbCr10CodecType        = 'v210'
k444YpCbCr10CodecType        = 'v410'
k4444YpCbCrA8RCodecType      = 'r408'
kJPEG2000CodecType           = 'mjp2'
kPxlletCodecType              = 'pxlt'
kH264CodecType                = 'avc1'

```

kSorenson3CodecType

Available in QuickTime 5 and later.

Available in Mac OS X v10.0 and later.

Declared in ImageCompression.h.

k422YpCbCr8CodecType

Component Y'CbCr 8-bit 4:2:2.

Available in Mac OS X v10.0 and later.

Declared in ImageCompression.h.

k444YpCbCr8CodecType

Component Y'CbCr 8-bit 4:4:4.

Available in Mac OS X v10.0 and later.

Declared in ImageCompression.h.

k4444YpCbCrA8CodecType

Component Y'CbCrA 8-bit 4:4:4:4.

Available in Mac OS X v10.0 and later.

Declared in ImageCompression.h.

k422YpCbCr16CodecType

Component Y'CbCr 10,12,14,16-bit 4:2:2.

Available in Mac OS X v10.0 and later.

Declared in ImageCompression.h.

k422YpCbCr10CodecType

Component Y'CbCr 10-bit 4:2:2.

Available in Mac OS X v10.0 and later.

Declared in ImageCompression.h.

k444YpCbCr10CodecType

Component Y'CbCr 10-bit 4:4:4.

Available in Mac OS X v10.0 and later.

Declared in ImageCompression.h.

k4444YpCbCrA8RCodecType

Component Y'CbCrA 8-bit 4:4:4, rendering format. full range alpha, zero biased YUV.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

Color Constants

Identify default colors for a graphics importer component.

blackColor	=33
blueColor	=409
cyanColor	=273
greenColor	=341
magentaColor	=137
redColor	=205
whiteColor	=30
yellowColor	=69

Color Modes

Constants that represent color modes.

useColorMatching	= 4
graphicsModePreWhiteAlpha	= 257
graphicsModePreBlackAlpha	= 258
graphicsModeComposition	= 259
graphicsModePreMulColorAlpha	= 261
graphicsModePerComponentAlpha	= 272
kQTAAlphaMode	= 'almo'
kQTAAlphaModePreMulColor	= 'almp'

kQTAAlphaMode

UInt32; for example, `graphicsModeStraightAlpha` or `graphicsModePreBlackAlpha`.

Available in Mac OS X v10.1 and later.

Declared in `ImageCompression.h`.

kQTAAlphaModePreMulColor

RGBColor; used if `kQTAAlphaMode` is `graphicsModePreMulColorAlpha`.

Available in Mac OS X v10.1 and later.

Declared in `ImageCompression.h`.

Component Call Selectors

Constants that represent selectors for component calls.

kClockGetTimeSelect	= 0x0001
kClockNewCallbackSelect	= 0x0002
kClockDisposeCallbackSelect	= 0x0003
kClockCallMeWhenSelect	= 0x0004
kClockCancelCallbackSelect	= 0x0005
kClockRateChangedSelect	= 0x0006
kClockTimeChangedSelect	= 0x0007
kClockSetTimeBaseSelect	= 0x0008
kClockStartStopChangedSelect	= 0x0009
kClockGetRateSelect	= 0x000A
kClockGetTimesForRateChangeSelect	= 0x000B
kClockGetRateChangeConstraintsSelect	= 0x000C
kSCGetCompressionExtendedSelect	= 0x0001
kSCPositionRectSelect	= 0x0002
kSCPositionDialogSelect	= 0x0003
kSCSetTestImagePictHandleSelect	= 0x0004
kSCSetTestImagePictFileSelect	= 0x0005
kSCSetTestImagePixMapSelect	= 0x0006
kSCGetBestDeviceRectSelect	= 0x0007
kSCRequestImageSettingsSelect	= 0x000A
kSCCompressImageSelect	= 0x000B
kSCCompressPictureSelect	= 0x000C
kSCCompressPictureFileSelect	= 0x000D
kSCRequestSequenceSettingsSelect	= 0x000E
kSCCompressSequenceBeginSelect	= 0x000F
kSCCompressSequenceFrameSelect	= 0x0010
kSCCompressSequenceEndSelect	= 0x0011
kSCDefaultPictHandleSettingsSelect	= 0x0012
kSCDefaultPictFileSettingsSelect	= 0x0013
kSCDefaultPixMapSettingsSelect	= 0x0014
kSCGetInfoSelect	= 0x0015
kSCSetInfoSelect	= 0x0016
kSCNewWorldSelect	= 0x0017
kSCSetCompressFlagsSelect	= 0x0018
kSCGetCompressFlagsSelect	= 0x0019
kSCGetSettingsAsTextSelect	= 0x001A
kSCGetSettingsAsAtomContainerSelect	= 0x001B
kSCSetSettingsFromAtomContainerSelect	= 0x001C
kSCCompressSequenceFrameAsyncSelect	= 0x001D
kSCAsyncIdleSelect	= 0x001E
kSCCopyCompressionSessionOptionsSelect	= 0x001F
kSCAudioInvokeLegacyCodecOptionsDialogSelect	= 0x0081
kTweenInitializeSelect	= 0x0001
kTweenDoTweenSelect	= 0x0002
kTweenResetSelect	= 0x0003
kTCGetCurrentTimeCodeSelect	= 0x0101
kTCGetTimeCodeAtTimeSelect	= 0x0102
kTCTimeCodeToStringSelect	= 0x0103
kTCTimeCodeToFrameNumberSelect	= 0x0104
kTCFrameNumberToTimeCodeSelect	= 0x0105
kTCGetSourceRefSelect	= 0x0106
kTCSetSourceRefSelect	= 0x0107
kTCSetTimeCodeFlagsSelect	= 0x0108
kTCGetTimeCodeFlagsSelect	= 0x0109
kTCsetDisplayOptionsSelect	= 0x010A
kTCgetDisplayOptionsSelect	= 0x010B
kMovieImportHandleSelect	= 0x0001
kMovieImportFileSelect	= 0x0002

kMovieImportSetSampleDurationSelect	= 0x0003
kMovieImportSetSampleDescriptionSelect	= 0x0004
kMovieImportSetMediaFileSelect	= 0x0005
kMovieImportSetDimensionsSelect	= 0x0006
kMovieImportSetChunkSizeSelect	= 0x0007
kMovieImportSetProgressProcSelect	= 0x0008
kMovieImportSetAuxiliaryDataSelect	= 0x0009
kMovieImportSetFromScrapSelect	= 0x000A
kMovieImportDoUserDialogSelect	= 0x000B
kMovieImportSetDurationSelect	= 0x000C
kMovieImportGetAuxiliaryDataTypeSelect	= 0x000D
kMovieImportValidateSelect	= 0x000E
kMovieImportGetFileTypeSelect	= 0x000F
kMovieImportDataRefSelect	= 0x0010
kMovieImportGetSampleDescriptionSelect	= 0x0011
kMovieImportGetMIMETypeListSelect	= 0x0012
kMovieImportSetOffsetAndLimitSelect	= 0x0013
kMovieImportGetSettingsAsAtomContainerSelect	= 0x0014
kMovieImportSetSettingsFromAtomContainerSelect	= 0x0015
kMovieImportSetOffsetAndLimit64Select	= 0x0016
kMovieImportIdleSelect	= 0x0017
kMovieImportValidateDataRefSelect	= 0x0018
kMovieImportGetLoadStateSelect	= 0x0019
kMovieImportGetMaxLoadedTimeSelect	= 0x001A
kMovieImportEstimateCompletionTimeSelect	= 0x001B
kMovieImportSetDontBlockSelect	= 0x001C
kMovieImportGetDontBlockSelect	= 0x001D
kMovieImportSetIdleManagerSelect	= 0x001E
kMovieImportSetNewMovieFlagsSelect	= 0x001F
kMovieImportGetDestinationMediaTypeSelect	= 0x0020
kMovieImportSetMediaDataRefSelect	= 0x0021
kMovieImportDoUserDialogDataRefSelect	= 0x0022
kMovieExportToHandleSelect	= 0x0080
kMovieExportToFileSelect	= 0x0081
kMovieExportGetAuxiliaryDataSelect	= 0x0083
kMovieExportSetProgressProcSelect	= 0x0084
kMovieExportSetSampleDescriptionSelect	= 0x0085
kMovieExportDoUserDialogSelect	= 0x0086
kMovieExportGetCreatorTypeSelect	= 0x0087
kMovieExportToDataRefSelect	= 0x0088
kMovieExportFromProceduresToDataRefSelect	= 0x0089
kMovieExportAddDataSourceSelect	= 0x008A
kMovieExportValidateSelect	= 0x008B
kMovieExportGetSettingsAsAtomContainerSelect	= 0x008C
kMovieExportSetSettingsFromAtomContainerSelect	= 0x008D
kMovieExportGetFileNameExtensionSelect	= 0x008E
kMovieExportGetShortFileTypeStringSelect	= 0x008F
kMovieExportGetSourceMediaTypeSelect	= 0x0090
kMovieExportSetGetMoviePropertyProcSelect	= 0x0091
kTextExportGetDisplayDataSelect	= 0x0100
kTextExportGetTimeFractionSelect	= 0x0101
kTextExportSetTimeFractionSelect	= 0x0102
kTextExportGetSettingsSelect	= 0x0103
kTextExportSetSettingsSelect	= 0x0104
kMIDIImportGetSettingsSelect	= 0x0100
kMIDIImportSetSettingsSelect	= 0x0101
kMovieExportNewGetDataAndPropertiesProcsSelect	= 0x0100
kMovieExportDisposeGetDataAndPropertiesProcsSelect	= 0x0101

kGraphicsImageImportSetSequenceEnabledSelect	= 0x0100
kGraphicsImageImportGetSequenceEnabledSelect	= 0x0101
kPreviewShowDataSelect	= 0x0001
kPreviewMakePreviewSelect	= 0x0002
kPreviewMakePreviewReferenceSelect	= 0x0003
kPreviewEventSelect	= 0x0004
kDataCodecDecompressSelect	= 0x0001
kDataCodecGetCompressBufferSizeSelect	= 0x0002
kDataCodecCompressSelect	= 0x0003
kDataCodecBeginInterruptSafeSelect	= 0x0004
kDataCodecEndInterruptSafeSelect	= 0x0005
kDataCodecDecompressPartialSelect	= 0x0006
kDataCodecCompressPartialSelect	= 0x0007
kDataHGetDataSelect	= 0x0002
kDataHPutDataSelect	= 0x0003
kDataHFlushDataSelect	= 0x0004
kDataHOpenForWriteSelect	= 0x0005
kDataHCloseForWriteSelect	= 0x0006
kDataHOpenForReadSelect	= 0x0008
kDataHCloseForReadSelect	= 0x0009
kDataHSetDataRefSelect	= 0x000A
kDataHGetDataRefSelect	= 0x000B
kDataHCompareDataRefSelect	= 0x000C
kDataHTaskSelect	= 0x000D
kDataHScheduleDataSelect	= 0x000E
kDataHFinishDataSelect	= 0x000F
kDataHFlushCacheSelect	= 0x0010
kDataHResolveDataRefSelect	= 0x0011
kDataHGetFileSizeSelect	= 0x0012
kDataHCanUseDataRefSelect	= 0x0013
kDataHGetVolumeListSelect	= 0x0014
kDataHWriteSelect	= 0x0015
kDataHPreextendSelect	= 0x0016
kDataHSetFileSizeSelect	= 0x0017
kDataHGetFreeSpaceSelect	= 0x0018
kDataHCreateFileSelect	= 0x0019
kDataHGetPreferredBlockSizeSelect	= 0x001A
kDataHGetDeviceIndexSelect	= 0x001B
kDataHIsStreamingDataHandlerSelect	= 0x001C
kDataHGetDataInBufferSelect	= 0x001D
kDataHGetScheduleAheadTimeSelect	= 0x001E
kDataHSetCacheSizeLimitSelect	= 0x001F
kDataHGetCacheSizeLimitSelect	= 0x0020
kDataHGetMovieSelect	= 0x0021
kDataHAddMovieSelect	= 0x0022
kDataHUpdateMovieSelect	= 0x0023
kDataHDoesBufferSelect	= 0x0024
kDataHGetFileNameSelect	= 0x0025
kDataHGetAvailableFileSizeSelect	= 0x0026
kDataHGetMacOSFileTypeSelect	= 0x0027
kDataHGetMIMETypeSelect	= 0x0028
kDataHSetDataRefWithAnchorSelect	= 0x0029
kDataHGetDataRefWithAnchorSelect	= 0x002A
kDataHSetMacOSFileTypeSelect	= 0x002B
kDataHSetTimeBaseSelect	= 0x002C
kDataHGetInfoFlagsSelect	= 0x002D
kDataHScheduleData64Select	= 0x002E
kDataHWrite64Select	= 0x002F

kDataHGetFileSize64Select	= 0x0030
kDataHPreextend64Select	= 0x0031
kDataHSetFileSize64Select	= 0x0032
kDataHGetFreeSpace64Select	= 0x0033
kDataHAppend64Select	= 0x0034
kDataHReadAsyncSelect	= 0x0035
kDataHPollReadSelect	= 0x0036
kDataHGetDataAvailabilitySelect	= 0x0037
kDataHGetFileSizeAsyncSelect	= 0x003A
kDataHGetDataRefAsTypeSelect	= 0x003B
kDataHSetDataRefExtensionSelect	= 0x003C
kDataHGetDataRefExtensionSelect	= 0x003D
kDataHGetMovieWithFlagsSelect	= 0x003E
kDataHGetFileTypeOrderingSelect	= 0x0040
kDataHCreateFileWithFlagsSelect	= 0x0041
kDataHGetMIMETypeAsyncSelect	= 0x0042
kDataHGetInfoSelect	= 0x0043
kDataHSetIdleManagerSelect	= 0x0044
kDataHDeleteFileSelect	= 0x0045
kDataHSetMovieUsageFlagsSelect	= 0x0046
kDataHUseTemporaryDataRefSelect	= 0x0047
kDataHGetTemporaryDataRefCapabilitiesSelect	= 0x0048
kDataHRenameFileSelect	= 0x0049
kDataHGetAvailableFileSize64Select	= 0x004E
kDataHGetDataAvailability64Select	= 0x004F
kDataHPlaybackHintsSelect	= 0x0103
kDataHPlaybackHints64Select	= 0x010E
kDataHGetDataRateSelect	= 0x0110
kDataHSetTimeHintsSelect	= 0x0111
kVDGetMaxSrcRectSelect	= 0x0001
kVDGetActiveSrcRectSelect	= 0x0002
kVDSetDigitizerRectSelect	= 0x0003
kVDGetDigitizerRectSelect	= 0x0004
kVDGetVBlankRectSelect	= 0x0005
kVDGetMaskPixMapSelect	= 0x0006
kVDGetPlayThruDestinationSelect	= 0x0008
kVDUseThisCLUTSelect	= 0x0009
kVDSetInputGammaValueSelect	= 0x000A
kVDGetInputGammaValueSelect	= 0x000B
kVDSetBrightnessSelect	= 0x000C
kVDGetBrightnessSelect	= 0x000D
kVDSetContrastSelect	= 0x000E
kVDSetHueSelect	= 0x000F
kVDSetSharpnessSelect	= 0x0010
kVDSetSaturationSelect	= 0x0011
kVDGetContrastSelect	= 0x0012
kVDGetHueSelect	= 0x0013
kVDGetSharpnessSelect	= 0x0014
kVDGetSaturationSelect	= 0x0015
kVDGrabOneFrameSelect	= 0x0016
kVDGetMaxAuxBufferSelect	= 0x0017
kVDGetDigitizerInfoSelect	= 0x0019
kVDGetCurrentFlagsSelect	= 0x001A
kVDSetKeyColorSelect	= 0x001B
kVDGetKeyColorSelect	= 0x001C
kVDAddKeyColorSelect	= 0x001D
kVDGetNextKeyColorSelect	= 0x001E
kVDSetKeyColorRangeSelect	= 0x001F

kVDGetKeyColorRangeSelect	= 0x0020
kVDSetDigitizerUserInterruptSelect	= 0x0021
kVDSetInputColorSpaceModeSelect	= 0x0022
kVDGetInputColorSpaceModeSelect	= 0x0023
kVDSetClipStateSelect	= 0x0024
kVDGetClipStateSelect	= 0x0025
kVDSetClipRgnSelect	= 0x0026
kVDClearClipRgnSelect	= 0x0027
kVDGetCLUTInUseSelect	= 0x0028
kVDSetPLLFilterTypeSelect	= 0x0029
kVDGetPLLFilterTypeSelect	= 0x002A
kVDGetMaskandValueSelect	= 0x002B
kVDSetMasterBlendLevelSelect	= 0x002C
kVDSetPlayThruDestinationSelect	= 0x002D
kVDSetPlayThruOnOffSelect	= 0x002E
kVDSetFieldPreferenceSelect	= 0x002F
kVDGetFieldPreferenceSelect	= 0x0030
kVDPreflightDestinationSelect	= 0x0032
kVDPreflightGlobalRectSelect	= 0x0033
kVDSetPlayThruGlobalRectSelect	= 0x0034
kVDSetInputGammaRecordSelect	= 0x0035
kVDGetInputGammaRecordSelect	= 0x0036
kVDSetBlackLevelValueSelect	= 0x0037
kVDGetBlackLevelValueSelect	= 0x0038
kVDSetWhiteLevelValueSelect	= 0x0039
kVDGetWhiteLevelValueSelect	= 0x003A
kVDGetVideoDefaultsSelect	= 0x003B
kVDGetNumberOfInputsSelect	= 0x003C
kVDGetInputFormatSelect	= 0x003D
kVDSetInputSelect	= 0x003E
kVDGetInputSelect	= 0x003F
kVDSetInputStandardSelect	= 0x0040
kVDSetupBuffersSelect	= 0x0041
kVDGrabOneFrameAsyncSelect	= 0x0042
kVDDoneSelect	= 0x0043
kVDSetCompressionSelect	= 0x0044
kVDCompressOneFrameAsyncSelect	= 0x0045
kVDCompressDoneSelect	= 0x0046
kVDReleaseCompressBufferSelect	= 0x0047
kVDGetImageDescriptionSelect	= 0x0048
kVDResetCompressSequenceSelect	= 0x0049
kVDSetCompressionOnOffSelect	= 0x004A
kVDGetCompressionTypesSelect	= 0x004B
kVDSetTimeBaseSelect	= 0x004C
kVDSetFrameRateSelect	= 0x004D
kVDGetDataRateSelect	= 0x004E
kVDGetSoundInputDriverSelect	= 0x004F
kVDGetDMA DepthsSelect	= 0x0050
kVDGetPreferredTimeScaleSelect	= 0x0051
kVDReleaseAsyncBuffersSelect	= 0x0052
kVDSetDataRateSelect	= 0x0054
kVDGetTimeCodeSelect	= 0x0055
kVDUseSafeBuffersSelect	= 0x0056
kVDGetSoundInputSourceSelect	= 0x0057
kVDGetCompressionTimeSelect	= 0x0058
kVDSetPreferredPacketSizeSelect	= 0x0059
kVDSetPreferredImageDimensionsSelect	= 0x005A
kVDGetPreferredImageDimensionsSelect	= 0x005B

kVDGetInputNameSelect	= 0x005C
kVDSetDestinationPortSelect	= 0x005D
kVDGetDeviceNameAndFlagsSelect	= 0x005E
kVDCaptureStateChangingSelect	= 0x005F
kVDGetUniqueIDsSelect	= 0x0060
kVDSelectUniqueIDsSelect	= 0x0061
kVDCopyPreferredAudioDeviceSelect	= 0x0063
kVDIIDCGetFeaturesSelect	= 0x0200
kVDIIDCSetFeaturesSelect	= 0x0201
kVDIIDCGetDefaultFeaturesSelect	= 0x0202
kVDIIDCGetCSRDataSelect	= 0x0203
kVDIIDCSetCSRDataSelect	= 0x0204
kVDIIDCGetFeaturesForSpecifierSelect	= 0x0205
kXMLParseDataRefSelect	= 0x0001
kXMLParseFileSelect	= 0x0002
kXMLParseDisposeXMLDocSelect	= 0x0003
kXMLParseGetDetailedParseErrorSelect	= 0x0004
kXMLParseAddElementSelect	= 0x0005
kXMLParseAddAttributeSelect	= 0x0006
kXMLParseAddMultipleAttributesSelect	= 0x0007
kXMLParseAddAttributeAndValueSelect	= 0x0008
kXMLParseAddMultipleAttributesAndValuesSelect	= 0x0009
kXMLParseAddAttributeValueKindSelect	= 0x000A
kXMLParseAddNameSpaceSelect	= 0x000B
kXMLParseSetOffsetAndLimitSelect	= 0x000C
kXMLParseSetEventParseRefConSelect	= 0x000D
kXMLParseSetStartDocumentHandlerSelect	= 0x000E
kXMLParseSetEndDocumentHandlerSelect	= 0x000F
kXMLParseSetStartElementHandlerSelect	= 0x0010
kXMLParseSetEndElementHandlerSelect	= 0x0011
kXMLParseSetCharDataHandlerSelect	= 0x0012
kXMLParseSetPreprocessInstructionHandlerSelect	= 0x0013
kXMLParseSetCommentHandlerSelect	= 0x0014
kXMLParseSetCDATAHandlerSelect	= 0x0015
kSGInitializeSelect	= 0x0001
kSGSetDataOutputSelect	= 0x0002
kSGGetDataOutputSelect	= 0x0003
kSGSetGWorldSelect	= 0x0004
kSGGetGWorldSelect	= 0x0005
kSGNewChannelSelect	= 0x0006
kSGDisposeChannelSelect	= 0x0007
kSGStartPreviewSelect	= 0x0010
kSGStartRecordSelect	= 0x0011
kSGIdleSelect	= 0x0012
kSGStopSelect	= 0x0013
kSGPauseSelect	= 0x0014
kSGPrepareSelect	= 0x0015
kSGReleaseSelect	= 0x0016
kSGGetMovieSelect	= 0x0017
kSGSetMaximumRecordTimeSelect	= 0x0018
kSGGetMaximumRecordTimeSelect	= 0x0019
kSGGetStorageSpaceRemainingSelect	= 0x001A
kSGGetTimeRemainingSelect	= 0x001B
kSGGrabPictSelect	= 0x001C
kSGGetLastMovieResIDSelect	= 0x001D
kSGSetFlagsSelect	= 0x001E
kSGGetFlagsSelect	= 0x001F
kSGSetDataProcSelect	= 0x0020

kSGNewChannelFromComponentSelect	= 0x0021
kSGDisposeDeviceListSelect	= 0x0022
kSGAppendDeviceListToMenuSelect	= 0x0023
kSGSetSettingsSelect	= 0x0024
kSGGetSettingsSelect	= 0x0025
kSGGetIndChannelSelect	= 0x0026
kSGUpdateSelect	= 0x0027
kSGGetPauseSelect	= 0x0028
kSGSettingsDialogSelect	= 0x0029
kSGGetAlignmentProcSelect	= 0x002A
kSGSetChannelSettingsSelect	= 0x002B
kSGGetChannelSettingsSelect	= 0x002C
kSGGetModeSelect	= 0x002D
kSGSetDataRefSelect	= 0x002E
kSGGetDataRefSelect	= 0x002F
kSGNewOutputSelect	= 0x0030
kSGDisposeOutputSelect	= 0x0031
kSGSetOutputFlagsSelect	= 0x0032
kSGSetChannelOutputSelect	= 0x0033
kSGGetDataOutputStorageSpaceRemainingSelect	= 0x0034
kSGHandleUpdateEventSelect	= 0x0035
kSGSetOutputNextOutputSelect	= 0x0036
kSGGetOutputNextOutputSelect	= 0x0037
kSGSetOutputMaximumOffsetSelect	= 0x0038
kSGGetOutputMaximumOffsetSelect	= 0x0039
kSGGetOutputDataReferenceSelect	= 0x003A
kSGWriteExtendedMovieDataSelect	= 0x003B
kSGGetStorageSpaceRemaining64Select	= 0x003C
kSGGetDataOutputStorageSpaceRemaining64Select	= 0x003D
kSGWriteMovieDataSelect	= 0x0100
kSGAddFrameReferenceSelect	= 0x0101
kSGGetNextFrameReferenceSelect	= 0x0102
kSGGetTimeBaseSelect	= 0x0103
kSGSortDeviceListSelect	= 0x0104
kSGAddMovieDataSelect	= 0x0105
kSGChangedSourceSelect	= 0x0106
kSGAddExtendedFrameReferenceSelect	= 0x0107
kSGGetNextExtendedFrameReferenceSelect	= 0x0108
kSGAddExtendedMovieDataSelect	= 0x0109
kSGAddOutputDataRefToMediaSelect	= 0x010A
kSGSetSettingsSummarySelect	= 0x010B
kSGSetChannelUsageSelect	= 0x0080
kSGGetChannelUsageSelect	= 0x0081
kSGSetChannelBoundsSelect	= 0x0082
kSGGetChannelBoundsSelect	= 0x0083
kSGSetChannelVolumeSelect	= 0x0084
kSGGetChannelVolumeSelect	= 0x0085
kSGGetChannelInfoSelect	= 0x0086
kSGSetChannelPlayFlagsSelect	= 0x0087
kSGGetChannelPlayFlagsSelect	= 0x0088
kSGSetChannelMaxFramesSelect	= 0x0089
kSGGetChannelMaxFramesSelect	= 0x008A
kSGSetChannelRefConSelect	= 0x008B
kSGSetChannelClipSelect	= 0x008C
kSGGetChannelClipSelect	= 0x008D
kSGGetChannelSampleDescriptionSelect	= 0x008E
kSGGetChannelDeviceListSelect	= 0x008F
kSGSetChannelDeviceSelect	= 0x0090

kSGSetChannelMatrixSelect	= 0x0091
kSGGetChannelMatrixSelect	= 0x0092
kSGGetChannelTimeScaleSelect	= 0x0093
kSGChannelPutPictureSelect	= 0x0094
kSGChannelSetRequestedDataRateSelect	= 0x0095
kSGChannelGetRequestedDataRateSelect	= 0x0096
kSGChannelSetDataSourceNameSelect	= 0x0097
kSGChannelGetDataSourceNameSelect	= 0x0098
kSGChannelSetCodecSettingsSelect	= 0x0099
kSGChannelGetCodecSettingsSelect	= 0x009A
kSGGetChannelTimeBaseSelect	= 0x009B
kSGGetChannelRefConSelect	= 0x009C
kSGGetChannelDeviceAndInputNamesSelect	= 0x009D
kSGSetChannelDeviceInputSelect	= 0x009E
kSGSetChannelSettingsStateChangingSelect	= 0x009F
kSGInitChannelSelect	= 0x0180
kSGWriteSamplesSelect	= 0x0181
kSGGetDataRateSelect	= 0x0182
kSGAlignChannelRectSelect	= 0x0183
kSGPanelGetDITLSelect	= 0x0200
kSGPanelGetTitleSelect	= 0x0201
kSGPanelCanRunSelect	= 0x0202
kSGPanelInstallSelect	= 0x0203
kSGPanelEventSelect	= 0x0204
kSGPanelItemSelect	= 0x0205
kSGPanelRemoveSelect	= 0x0206
kSGPanelSetGrabberSelect	= 0x0207
kSGPanelSetResFileSelect	= 0x0208
kSGPanelGetSettingsSelect	= 0x0209
kSGPanelSetSettingsSelect	= 0x020A
kSGPanelValidateInputSelect	= 0x020B
kSGPanelSetEventFilterSelect	= 0x020C
kSGPanelGetDITLForSizeSelect	= 0x020D
kSGGetSrcVideoBoundsSelect	= 0x0100
kSGSetVideoRectSelect	= 0x0101
kSGGetVideoRectSelect	= 0x0102
kSGGetVideoCompressorTypeSelect	= 0x0103
kSGSetVideoCompressorTypeSelect	= 0x0104
kSGSetVideoCompressorSelect	= 0x0105
kSGGetVideoCompressorSelect	= 0x0106
kSGGetVideoDigitizerComponentSelect	= 0x0107
kSGSetVideoDigitizerComponentSelect	= 0x0108
kSGVideoDigitizerChangedSelect	= 0x0109
kSGSetVideoBottlenecksSelect	= 0x010A
kSGGetVideoBottlenecksSelect	= 0x010B
kSGGrabFrameSelect	= 0x010C
kSGGrabFrameCompleteSelect	= 0x010D
kSGDisplayFrameSelect	= 0x010E
kSGCompressFrameSelect	= 0x010F
kSGCompressFrameCompleteSelect	= 0x0110
kSGAddFrameSelect	= 0x0111
kSGTransferFrameForCompressSelect	= 0x0112
kSGSetCompressBufferSelect	= 0x0113
kSGGetCompressBufferSelect	= 0x0114
kSGGetBufferInfoSelect	= 0x0115
kSGSetUseScreenBufferSelect	= 0x0116
kSGGetUseScreenBufferSelect	= 0x0117
kSGGrabCompressCompleteSelect	= 0x0118

kSGDisplayCompressSelect	= 0x0119
kSGSetFrameRateSelect	= 0x011A
kSGGetFrameRateSelect	= 0x011B
kSGSetPreferredPacketSizeSelect	= 0x0121
kSGGetPreferredPacketSizeSelect	= 0x0122
kSGSetUserVideoCompressorListSelect	= 0x0123
kSGGetUserVideoCompressorListSelect	= 0x0124
kSGSetSoundInputDriverSelect	= 0x0100
kSGGetSoundInputDriverSelect	= 0x0101
kSGSoundInputDriverChangedSelect	= 0x0102
kSGSetSoundRecordChunkSizeSelect	= 0x0103
kSGGetSoundRecordChunkSizeSelect	= 0x0104
kSGSetSoundInputRateSelect	= 0x0105
kSGGetSoundInputRateSelect	= 0x0106
kSGSetSoundInputParametersSelect	= 0x0107
kSGGetSoundInputParametersSelect	= 0x0108
kSGSetAdditionalSoundRatesSelect	= 0x0109
kSGGetAdditionalSoundRatesSelect	= 0x010A
kSGSetFontNameSelect	= 0x0100
kSGSetFontSelect	= 0x0101
kSGSetTextForeColorSelect	= 0x0102
kSGSetTextBackColorSelect	= 0x0103
kSGSetJustificationSelect	= 0x0104
kSGGetTextReturnToSpaceValueSelect	= 0x0105
kSGSetTextReturnToSpaceValueSelect	= 0x0106
kSGGetInstrumentSelect	= 0x0100
kSGSetInstrumentSelect	= 0x0101
kQTVideoOutputGetDisplayModeListSelect	= 0x0001
kQTVideoOutputGetCurrentClientNameSelect	= 0x0002
kQTVideoOutputSetClientNameSelect	= 0x0003
kQTVideoOutputGetClientNameSelect	= 0x0004
kQTVideoOutputBeginSelect	= 0x0005
kQTVideoOutputEndSelect	= 0x0006
kQTVideoOutputSetDisplayModeSelect	= 0x0007
kQTVideoOutputGetDisplayModeSelect	= 0x0008
kQTVideoOutputSaveStateSelect	= 0x000A
kQTVideoOutputRestoreStateSelect	= 0x000B
kQTVideoOutputGetGWorldSelect	= 0x000C
kQTVideoOutputGetGWorldParametersSelect	= 0x000D
kQTVideoOutputGetIndSoundOutputSelect	= 0x000E
kQTVideoOutputGetClockSelect	= 0x000F
kQTVideoOutputSetEchoPortSelect	= 0x0010
kQTVideoOutputGetIndImageDecompressorSelect	= 0x0011
kQTVideoOutputBaseSetEchoPortSelect	= 0x0012
kQTVideoOutputCopyIndAudioOutputDeviceUIDSelect	= 0x0016

Component Identifiers

Identify the types of components.

clockComponentType	= 'clok'
compressorComponentType	= 'imco'
CreateFilePreviewComponentType	= 'pmak'
DataHandlerType	= 'dhlr'
decompressorComponentType	= 'imdc'
MediaHandlerType	= 'mhlr'
MovieControllerComponentType	= 'play'
MovieExportType	= 'spit'
MovieImportType	= 'eat '
SeqGrabChannelType	= 'sgch'
SeqGrabComponentType	= 'barg'
SeqGrabCompressionPanelType	= 'cmpr'
SeqGrabPanelType	= 'sgpn'
SeqGrabSourcePanelType	= 'sour'
ShowFilePreviewComponentType	= 'pnot'
StandardCompressionSubType	= 'imag'
StandardCompressionSubTypeSound	= 'soun'
StandardCompressionType	= 'scdi'
systemMicrosecondClock	= 'micr'
systemMillisecondClock	= 'mill'
systemSecondClock	= 'seco'
systemTickClock	= 'tick'
videoDigitizerComponentType	= 'vdig'

Discussion

All components of the same type or subtype provide the same kinds of services and support a common application programming interface. Codecs have their own set of types.

Component Property IDs and Flags

Constants that contain the flags and IDs of component properties.

```

uppcallComponentGetComponentPropertyInfoProcInfo = 0x0003FFFF
uppcallComponentGetComponentPropertyProcInfo = 0x0003FFFF
uppcallComponentSetComponentPropertyProcInfo = 0x0000FFFF
uppcallComponentAddComponentPropertyListenerProcInfo = 0x0000FFFF
uppcallComponentRemoveComponentPropertyListenerProcInfo = 0x0000FFFF
kCallComponentExecuteWiredActionSelect = -9
kComponentPropertyFlagCanSetLater = (1L << 0)
kComponentPropertyFlagCanSetNow = (1L << 1)
kComponentPropertyFlagCanGetNow = (1L << 3)
kComponentPropertyFlagHasExtendedInfo = (1L << 4)
kComponentPropertyFlagValueMustBeReleased = (1L << 5)
kComponentPropertyFlagValueIsCTypeRef = (1L << 6)
kComponentPropertyFlagGetBufferMustBeInitialized = (1L << 7)
kQTComponentPropertyListenerCollectionContextVersion = 1
kQTGetComponentPropertyInfoSelect = -11
kQTGetComponentPropertySelect = -12
kQTSetComponentPropertySelect = -13
kQTAddComponentPropertyListenerSelect = -14
kQTRemoveComponentPropertyListenerSelect = -15

```

Error Codes

Identify errors generated while executing QuickTime calls.

// General QuickTime errors	
couldNotResolveDataRef	=-2000
badImageDescription	=-2001
badPublicMovieAtom	=-2002
cantFindHandler	=-2003
cantOpenHandler	=-2004
badComponentType	=-2005
noMediaHandler	=-2006
noDataHandler	=-2007
invalidMedia	=-2008
invalidTrack	=-2009
invalidMovie	=-2010
invalidSampleTable	=-2011
invalidDataRef	=-2012
invalidHandler	=-2013
invalidDuration	=-2014
invalidTime	=-2015
cantPutPublicMovieAtom	=-2016
badEditList	=-2017
mediaTypesDontMatch	=-2018
progressProcAborted	=-2019
movieToolboxUninitialized	=-2020
noRecordOfApp	=-2020
wfFileNotFound	=-2021
cantCreateSingleForkFile	=-2022
invalidEditState	=-2023
nonMatchingEditState	=-2024
staleEditState	=-2025
userDataItemNotFound	=-2026
maxSizeToGrowTooSmall	=-2027
badTrackIndex	=-2028
trackIDNotFound	=-2029
trackNotInMovie	=-2030
timeNotInTrack	=-2031
timeNotInMedia	=-2032
badEditIndex	=-2033
internalQuickTimeError	=-2034
cantEnableTrack	=-2035
invalidRect	=-2036
invalidSampleNum	=-2037
invalidChunkNum	=-2038
invalidSampleDescIndex	=-2039
invalidChunkCache	=-2040
invalidSampleDescription	=-2041
dataNotOpenForRead	=-2042
dataNotOpenForWrite	=-2043
dataAlreadyOpenForWrite	=-2044
dataAlreadyClosed	=-2045
endOfDataReached	=-2046
dataNoDataRef	=-2047
noMovieFound	=-2048
invalidDataRefContainer	=-2049
badDataRefIndex	=-2050
noDefaultDataRef	=-2051
couldNotUseAnExistingSample	=-2052
featureUnsupported	=-2053
unsupportedAuxiliaryImportData	=-2057
auxiliaryExportDataUnavailable	=-2058

samplesAlreadyInMediaErr	== -2059
noSourceTreeFoundErr	== -2060
sourceNotFoundErr	== -2061
movieTextNotFoundErr	== -2062
missingRequiredParameterErr	== -2063
invalidSpriteWorldPropertyErr	== -2064
invalidSpritePropertyErr	== -2065
gWorldsNotSameDepthAndSizeErr	== -2066
invalidSpriteIndexErr	== -2067
invalidImageIndexErr	== -2068
invalidSpriteIDErr	== -2069
// QuickTime Music Architecture errors	
internalComponentErr	== -2070
notImplementedMusicOSErr	== -2071
cantSendToSynthesizerOSErr	== -2072
cantReceiveFromSynthesizerOSErr	== -2073
illegalVoiceAllocationOSErr	== -2074
illegalPartOSErr	== -2075
illegalChannelOSErr	== -2076
illegalKnobOSErr	== -2077
illegalKnobValueOSErr	== -2078
illegalInstrumentOSErr	== -2079
illegalControllerOSErr	== -2080
midiManagerAbsentOSErr	== -2081
synthesizerNotRespondingOSErr	== -2082
synthesizerOSErr	== -2083
illegalNoteChannelOSErr	== -2084
noteChannelNotAllocatedOSErr	== -2085
tunePlayerFullOSErr	== -2086
tuneParseOSErr	== -2087
noExportProcAvailableErr	== -2089
videoOutputInUseErr	== -2090
// Windows-specific errors	
componentDllLoadErr	== -2091
componentDllEntryNotFoundErr	== -2092
qtmlDllLoadErr	== -2093
qtmlDllEntryNotFoundErr	== -2094
qtmlUninitialized	== -2095
unsupportedOSErr	== -2096
unsupportedProcessorErr	== -2097
noVideoTrackInMovieErr	== -2054
noSoundTrackInMovieErr	== -2055
soundSupportNotAvailableErr	== -2056
// QT atom errors	
cannotFindAtomErr	== -2101
notLeafAtomErr	== -2102
atomsNotOfSameTypeErr	== -2103
atomIndexInvalidErr	== -2104
duplicateAtomTypeAndIDErr	== -2105
invalidAtomErr	== -2106
invalidAtomContainerErr	== -2107
invalidAtomTypeErr	== -2108
cannotBeLeafAtomErr	== -2109
// Data access errors	
pathTooLongErr	== -2110
emptyPathErr	== -2111
noPathMappingErr	== -2112
pathNotVerifiedErr	== -2113

unknownFormatErr	=-2114
wackBadFileErr	=-2115
wackForkNotFoundErr	=-2116
wackBadMetaDataErr	=-2117
qfcbNotFoundErr	=-2118
qfcbNotCreatedErr	=-2119
AAPNotCreatedErr	=-2120
AAPNotFoundErr	=-2121
ASDBadHeaderErr	=-2122
ASDBadForkErr	=-2123
ASDEntryNotFoundErr	=-2124
fileOffsetTooBigErr	=-2125
notAllowedToSaveMovieErr	=-2126
qtNetworkAlreadyAllocatedErr	=-2127
urlDataHTTPProtocolErr	=-2129
urlDataHTTPNoNetDriverErr	=-2130
urlDataHTTPURLErr	=-2131
urlDataHTTPRedirectErr	=-2132
urlDataFTPProtocolErr	=-2133
urlDataFTPShutdownErr	=-2134
urlDataFTPBadUserErr	=-2135
urlDataFTPBadPasswordErr	=-2136
urlDataFTPServerError	=-2137
urlDataFTPDataConnectionErr	=-2138
urlDataFTPNoDirectoryErr	=-2139
urlDataFTPQuotaErr	=-2140
urlDataFTPPermissionsErr	=-2141
urlDataFTPFilenameErr	=-2142
urlDataFTPNoNetDriverErr	=-2143
urlDataFTPBadNameListErr	=-2144
urlDataFTPNeedPasswordErr	=-2145
urlDataFTPNoPasswordErr	=-2146
urlDataFTPServerDisconnectedErr	=-2147
urlDataFTPURLErr	=-2148
notEnoughDataErr	=-2149
qtActionNotHandledErr	=-2157
// Digitizing errors	
digiUnimpErr	=-2201
qtParamErr	=-2202
matrixErr	=-2203
notExactMatrixErr	=-2204
noMoreKeyColorsErr	=-2205
notExactSizeErr	=-2206
badDepthErr	=-2207
noDMAErr	=-2208
badCallOrderErr	=-2209
// Codec errors	
codecErr	=-8960
noCodecErr	=-8961
codecUnimpErr	=-8962
codecSizeErr	=-8963
codecScreenBufErr	=-8964
codecImageBufErr	=-8965
codecSpoolErr	=-8966
codecAbortErr	=-8967
codecWouldOffscreenErr	=-8968
codecBadDataErr	=-8969
codecDataVersErr	=-8970

codecExtensionNotFoundErr	=-8971
scTypeNotFoundErr	=-8971
codecConditionErr	=-8972
codecOpenErr	=-8973
codecCantWhenErr	=-8974
codecCantQueueErr	=-8975
codecNothingToBlitErr	=-8976
codecNoMemoryPleaseWaitErr	=-8977
codecDisabledErr	=-8978
codecNeedToFlushChainErr	=-8979
lockPortBitsBadSurfaceErr	=-8980
lockPortBitsWindowMovedErr	=-8981
lockPortBitsWindowResizedErr	=-8982
lockPortBitsWindowClippedErr	=-8983
lockPortBitsBadPortErr	=-8984
lockPortBitsSurfaceLostErr	=-8985
codecParameterDialogConfirm	=-8986
codecNeedAccessKeyErr	=-8987
codecOffscreenFailedErr	=-8988
codecDroppedFrameErr	=-8989
directXObjectAlreadyExists	=-8990
lockPortBitsWrongGDeviceErr	=-8991
codecOffscreenFailedPleaseRetryErr	=-8992
// Sequence Grabber errors	
noDeviceForChannel	=-9400,
grabTimeComplete	=-9401,
cantDoThatInCurrentMode	=-9402,
notEnoughMemoryToGrab	=-9403,
notEnoughDiskSpaceToGrab	=-9404,
couldntGetRequiredComponent	=-9405,
badSGChannel	=-9406,
seqGrabInfoNotAvailable	=-9407,
deviceCantMeetRequest	=-9408,
// Movie Controller errors	
badControllerHeight	=-9994,
editingNotAllowed	=-9995,
controllerBoundsNotExact	=-9996,
cannotSetWidthOfAttachedController	=-9997,
controllerHasFixedHeight	=-9998,
cannotMoveAttachedController	=-9999
// QuickTime VR Errors	
notAQTVRMovieErr	=-30540
constraintReachedErr	=-30541
callNotSupportedByNodeErr	=-30542
selectorNotSupportedByNodeErr	=-30543
invalidNodeIDErr	=-30544
invalidViewStateErr	=-30545
timeNotInViewErr	=-30546
propertyNotSupportedByNodeErr	=-30547
settingNotSupportedByNodeErr	=-30548
limitReachedErr	=-30549
invalidNodeFormatErr	=-30550
invalidHotSpotIDErr	=-30551
noMemoryNodeFailedInitialize	=-30552
streamingNodeNotReadyErr	=-30553
qtvrLibraryLoadErr	=-30554
qtvrUninitialized	=-30555

`noRecordOfApp`

A replica of the `movieToolboxUninitialized` error.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`cantCreateSingleForkFile`

The file to be created already exists.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`componentDllLoadErr`

Windows error returned when a component is loading.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`componentDllEntryNotFoundErr`

Windows error returned when a component is loading.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`qtmDllLoadErr`

Windows error returned when the QuickTime Media Layer is loading.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`qtmDllEntryNotFoundErr`

Windows error returned when the QuickTime Media Layer is loading.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`digiUnimpErr`

Digitizer feature is unimplemented.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`qtParamErr`

Bad input parameter (out of range, for example).

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`matrixErr`

Bad matrix; the digitizer did nothing.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`notExactMatrixErr`

Warning of a bad matrix; the digitizer did its best.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`noMoreKeyColorsErr`

All the key indexes are in use.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`notExactSizeErr`

Can't digitize to the exact size requested.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`badDepthErr`

Can't digitize into the requested pixel depth.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`noDMAErr`

Can't do DMA digitizing; that is, can't go to the requested destination.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

`badCallOrderErr`

A status call was made before being set up first.

Available in Mac OS X v10.0 and later.

Declared in `MacErrors.h`.

Discussion

The Movie Toolbox provides two error values to your application: the current error and the sticky error. The current error is the result code from the last Movie Toolbox function; it is updated each time your application calls a Movie Toolbox function. The sticky error value contains the first nonzero result code from any Movie Toolbox function that you called after having cleared the sticky error with `ClearMoviesStickyError`.

File Types and Creators

Identify the formats of graphics files and the applications that create them.

```

// File types
ftAdobePremiereMovie      ='MooV'
ftAfterDarkModule         ='ADgm'
ftClip3Dgraphic           ='EZ3D'
ftCricketChart            ='CGPC'
ftCricketDrawing          ='CKDT'
ftDesignCADDrawing        ='DCAD'
ftImageStudioGraphic      ='RIFF'
ftKaleidaGraphGraphic     ='QPCT'
ftMacFlowChart            ='FLCH'
ftMacSpinDataSet          ='D2BN'
ftMoviePlayerMovie        ='MooV'
ftPixelPaint              ='PX01'
ftSuper3DDrawing          ='3DBX'
ftSwivel3DDrawing         ='SMDL'
ftVersaCADDrawing         ='2D '
// Creator codes
sigAdobePremiere          ='PrMr'
sigAfterDark               ='ADrk'
sigAldusSuper3D           ='SP3D'
sigAutoCAD                 ='ACAD'
sigClip3D                  ='EZ3E'
sigCricketDraw            ='CRDW'
sigCricketGraph           ='CGRF'
sigDeltagraphPro          ='DGRH'
sigDesign2                 ='DESG'
sigDesignCAD               ='ASBC'
sigDesignStudio           ='MRJN'
sigDigDarkroom            ='DIDR'
sigDreams                  ='PHNX'
sigDynaperspective        ='PERS'
sigGenericCADD             ='CAD3'
sigGraphMaster            ='GRAM'
sigImageStudio            ='FSPE'
sigInfiniD                 ='SI∞D'
sigKaleidaGraph           ='QKPT'
sigKidPix                  ='Kid2'
sigLabVIEW                 ='LBVW'
sigMacDraft                ='MD20'
sigMacDraw                 ='MDRW'
sigMacFlow                 ='MCFL'
sigMacSpin                 ='D2SP'
sigMiniCad                 ='CDP3'
sigModelShop               ='MDSP'
sigMoviePlayer            ='TVOD'
sigMovieRecorder          ='mrcr'
sigOasis                   ='TAOA'
sigOBJECTMASTER           ='BROW'
sigOfoto                   ='APLS'
sigOmnis5                  ='Q2$$'
sigOptix                   ='PIXL'
sigPhotoMac                ='PMAC'
sigPictureCompressor       ='ppxi'
sigPICTViewer              ='MDTS'
sigPixelPaint              ='PIXR'
sigScreenPlay              ='SPLY'
sigSmoothie                ='Smoo'
sigStudio1                 ='ST/1'

```

```

sigStudio32          ='ST32'
sigStudio8           ='ST/8'
sigSwivel3D         ='SWVL'
sigVersaCad          ='VCAD'

```

Discussion

Constant names for creator codes are written as `sig` followed by the application name. Constant names for file types are written as `ft` followed by the document type.

Graphics Transfer Modes

Determine how images will be transferred.

```

// Boolean modes
// src modes are used with bitmaps and text;
// pat modes are used with lines and shapes
srcCopy              =0
srcOr                =1
srcXor               =2
srcBic               =3
notSrcCopy           =4
notSrcOr             =5
notSrcXor            =6
notSrcBic            =7
patCopy              =8
patOr                =9
patXor               =10
patBic               =11
notPatCopy           =12
notPatOr             =13
notPatXor            =14
notPatBic            =15
// Text dimming
grayishTextOr        =49
// Highlighting
hilite               =50
hilitetransfermode   =50
// Arithmetic modes
blend                =32
addPin               =33
addOver              =34
subPin               =35
addMax                =37
adMax                =37
subOver              =38
adMin                =39
ditherCopy           =64
// Transparent mode
transparent           =36

```

`srcCopy`

If the source is black, apply the foreground color to the destination; if the source is white, apply the background color; otherwise apply weighted portions of the foreground and background colors.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`srcOr`

If the source is black, apply the foreground color to the destination; if the source is white, do nothing; otherwise apply weighted portions of the foreground color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`srcXor`

If the source is black, invert the destination (this operation is undefined for a colored destination). Otherwise, do nothing.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`srcBic`

If the source is black, apply the background color to the destination. If the source is white, do nothing. Otherwise, apply weighted portions of the background color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`notSrcCopy`

If the source is white, apply the foreground color to the destination; if the source is black, apply the background color; otherwise apply weighted portions of the foreground and background colors.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`notSrcOr`

If the source is white, apply the foreground color to the destination; if the source is black, do nothing; otherwise apply weighted portions of the foreground color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`notSrcXor`

If the source is white, invert the destination (this operation is undefined for a colored destination pixel). Otherwise, do nothing.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`notSrcBic`

If the source is white, apply the background color to the destination. If the source is black, do nothing. Otherwise, apply weighted portions of the background color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`patCopy`

If the source is black, apply the foreground color to the destination; if the source is white, apply the background color; otherwise apply weighted portions of the foreground and background colors.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

patOr

If the source is black, apply the foreground color to the destination; if the source is white, do nothing; otherwise apply weighted portions of the foreground color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

patXor

If the source is black, invert the destination (this operation is undefined for a colored destination). Otherwise, do nothing.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

patBic

If the source is black, apply the background color to the destination. If the source is white, do nothing. Otherwise, apply weighted portions of the background color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

notPatCopy

If the source is white, apply the foreground color to the destination; if the source is black, apply the background color; otherwise apply weighted portions of the foreground and background colors.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

notPatOr

If the source is white, apply the foreground color to the destination; if the source is black, do nothing; otherwise apply weighted portions of the foreground color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

notPatXor

If the source is white, invert the destination (this operation is undefined for a colored destination pixel). Otherwise, do nothing.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

grayishTextOr

Dim the destination. If in color, replace it with a blend of the foreground and background; if black-and-white, replace it with dithered black and white. This mode is used primarily for text.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

hilite

Replace the background color with the highlight color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

hiliteTransfermode

Replace the background color with the highlight color.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`blend`

Replace the destination with a blend of the source and destination colors. If the destination is a bitmap, this is the same as `srcCopy`.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`addPin`

Replace the destination with the sum of the source and destination, up to a maximum value. If the destination is a bitmap, this is the same as `srcBic`.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`addOver`

Replace the destination with the sum of the source and destination, but if the resulting red, green, or blue value exceeds 65536, then subtract 65536 from it. If the destination is a bitmap, this is the same as `srcXor`.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`subPin`

Replace the destination with the difference between the source and destination, but not less than a minimum value. If the destination is a bitmap, this is the same as `srcOr`.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`addMax`

Compare the source and destination, and replace the destination with the greater value of each of the red, green, and blue components. If the destination is a bitmap, this is the same as `srcBic`.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`adMax`

Compare the source and destination, and replace the destination with the greater value of each of the red, green, and blue components. If the destination is a bitmap, this is the same as `srcBic`.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`subOver`

Replace the destination with the difference between the source and destination, but if the resulting red, green, or blue value is negative, then add 65536 to it. If the destination is a bitmap, this is the same as `srcXor`.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`adMin`

Compare the source and destination, and replace the destination with the lesser value of each of the red, green, and blue components. If the destination is a bitmap, this is the same as `srcOr`.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`ditherCopy`

Replace the destination with a dither mix of the source and destination.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

`transparent`

Replace the destination with the source if the source is not equal to the background.

Available in Mac OS X v10.0 and later.

Declared in `QuickdrawTypes.h`.

See Also

For more information about graphics transfer modes, see *Inside Macintosh: Imaging With QuickDraw*.

Localization Codes

Identify languages, scripts, numbering systems, calendar systems, and geographical regions.

```

// Language codes:
langAfrikaans           =141 // smRoman script
langBreton              =142 // smRoman or smRoman/Celtic script
langAlbanian           =36 // smRoman script
langAmharic            =85 // smEthiopic script
langArabic              =12 // smArabic script
langArmenian           =51 // smArmenian script
langAssamese           =68 // smBengali script
langAymara             =134 // smRoman script
langAzerbaijanAr      =50 // Azerbaijani in smArabic script
langAzerbaijani       =49 // Azerbaijani in smCyrillic script
langBasque             =129 // smRoman script
langBelorussian        =46 // Synonym for langByelorussian
langUzbek              =47 // smCyrillic script
langBengali            =67 // smBengali script
langBulgarian          =44 // smCyrillic script
langBurmese            =77 // smBurmese script
langByelorussian      =46 // smCyrillic script
langCatalan            =130 // smRoman script
langChewa              =92 // synonym for langNyanja
langCroatian           =18 // modified smRoman/Croatian script
langCzech              =38 // smCentralEuroRoman script
langDanish             =7 // smRoman script
langDutch              =4 // smRoman script
langDzongkha          =137 // (Bhutan) smTibetan script
langEnglish            =0 // smRoman script
langEsperanto          =94 // smRoman script
langEstonian           =27 // smCentralEuroRoman script
langFaroese           =30 // smRoman/Icelandic script
langFarsi              =31 // modified smArabic/Farsi script
langFinnish            =13 // smRoman script
langFlemish            =34 // smRoman script
langFrench             =1 // smRoman script
langGalician           =140 // smRoman script
langGeorgian           =52 // smGeorgian script
langGerman             =2 // smRoman script
langGreek              =14 // Greek script using smRoman script
langGreekPoly         =148 // smGreek script
langGreenlandic        =149 // smRoman script
langGuarani           =133 // smRoman script
langGujarati          =69 // smGujarati script
langHebrew            =10 // smHebrew script
langHindi              =21 // smDevanagari script
langHungarian          =26 // smCentralEuroRoman script
langIcelandic         =15 // modified smRoman/Icelandic script
langIndonesian         =81 // smRoman script
langInuktitut         =143 // Inuit using smEthiopic script
langIrishGaelic        =35 // smRoman or smRoman/Celtic script
langIrishGaelicScript =146 // smRoman/Gaelic script
langItalian            =3 // smRoman script
langJapanese          =11 // smJapanese script
langJavaneseRom       =138 // Javanese in smRoman script
langKannada           =73 // smKannada script
langKashmiri          =61 // smArabic script
langKazakh            =48 // smCyrillic script
langKhmer              =78 // smKhmer script
langKinyarwanda       =90 // smRoman script
langKirghiz           =54 // smCyrillic script

```

langKorean	=23	// smKorean script
langKurdish	=60	// smArabic script
langLao	=79	// smLao script
langLatin	=131	// smRoman script
langLatvian	=28	// smCentralEuroRoman script
langLithuanian	=24	// smCentralEuroRoman script
langMacedonian	=43	// smCyrillic script
langMalagasy	=93	// smRoman script
langMalayalam	=72	// smMalayalam script
langMalayArabic	=84	// Malay in smArabic script
langMalayRoman	=83	// Malay in smRoman script
langMaltese	=16	// smRoman script
langManxGaelic	=145	// smRoman or smRoman/Celtic script
langMarathi	=66	// smDevanagari script
langMoldavian	=53	// smCyrillic script
langMongolian	=57	// Mongolian in smMongolian script
langMongolianCyr	=58	// Mongolian in smCyrillic script
langNepali	=64	// smDevanagari script
langNorwegian	=9	// smRoman script
langNyanja	=92	// smRoman script
langOriya	=71	// smOriya script
langOromo	=87	// smEthiopic script
langPashto	=59	// smArabic script
langPersian	=31	// Synonym for langFarsi
langPolish	=25	// smCentralEuroRoman script
langPortuguese	=8	// smRoman script
langPunjabi	=70	// smGurmukhi script
langQuechua	=132	// smRoman script
langRomanian	=37	// modified smRoman/Romanian script
langRuanda	=90	// synonym for langKinyarwanda
langRundi	=91	// smRoman script
langRussian	=32	// smCyrillic script
langSami	=29	// language of the Sami in Scandanavia
langSanskrit	=65	// smDevanagari script
langScottishGaelic	=144	// smRoman or smRoman/Celtic script
langSerbian	=42	// smCyrillic script
langSimpChinese	=33	// Mandarin in smSimpChinese script
langSindhi	=62	// smArabic script
langSinhalese	=76	// smSinhalese script
langSlovak	=39	// smCentralEuroRoman script
langSlovenian	=40	// modified smRoman/Croatian script
langSomali	=88	// smRoman script
langSpanish	=6	// smRoman script
langSundaneseRom	=139	// Sundanese in smRoman script
langSwahili	=89	// smRoman script
langSwedish	=5	// smRoman script
langTagalog	=82	// smRoman script
langTajiki	=55	// smCyrillic script
langTamil	=74	// smTamil script
langTatar	=135	// smCyrillic script
langTelugu	=75	// smTelugu script
langThai	=22	// smThai script
langTibetan	=63	// smTibetan script
langTigrinya	=86	// smEthiopic script
langTongan	=147	// smRoman script
langTradChinese	=19	// Mandarin in smTradChinese script
langTurkish	=17	// modified smRoman/Turkish script
langTurkmen	=56	// smCyrillic script

```

langUighur           =136 // smArabic script
langUkrainian       =45  // modified smCyrillic/Ukrainian script
langUrdu            =20  // smArabic script
langVietnamese     =80  // smVietnamese script
langWelsh          =128 // modified smRoman/Celtic script
langYiddish        =41  // smHebrew script
langUnspecified    =32767
// Script codes
smArabic           =4
smArmenian         =24
smBengali          =13
smBurmese          =19
smCentralEuroRoman =29
smCyrillic        =7
smDevanagari      =9
smEthiopic        =28
smExtArabic       =31 // extended Arabic
smGeez            =28 // Synonym for smEthiopic
smGeorgian        =23
smGreek           =6
smGujarati        =11
smGurmukhi        =10
smHebrew          =5
smJapanese        =1
smKannada         =16 // Kannada/Kanarese
smKhmer           =20 // Khmer/Cambodian
smKorean          =3
smLao             =22
smMalayalam       =17
smMongolian       =27
smOriya           =12
smRoman           =0
smRSymbol         =8 // Right-left symbol
smSimpChinese     =25 // Simplified Chinese
smSinhalese       =18
smTamil           =14
smTelugu          =15
smThai           =21
smTibetan         =26
smTradChinese     =2 // Traditional Chinese
smUnicodeScript   =0x7E // Unicode
smUninterp        =32 // Uninterpreted symbols
smVietnamese      =30
// Calendar codes
calGregorian      =0
calArabicCivil    =1
calArabicLunar    =2
calJapanese       =3
calJewish         =4
calCoptic         =5
calPersian        =6
// Integer format codes
intWestern        =0
intArabic         =1
intRoman          =2
intJapanese       =3
intEuropean       =4
// Region codes

```

verAfrikaans	=102
verArabic	=16
verArmenian	=84
verAustralia	=15
verAustria	=92
verBengali	=60
verBhutan	=83
verBrazil	=71
verBreton	=77
verBritain	=2
verBulgaria	=72
verByeloRussian	=61
verCatalonia	=73
verChina	=52
verCroatia	=68
verCyprus	=23
verCzech	=56
verDenmark	=9
verEngCanada	=82
verEsperanto	=103
verEstonia	=44
verFarEastGeneric	=58
verFaroeIsl	=47
verFinland	=17
verFlemish	=6
verFrance	=1
verFrBelgium	=98
verFrCanada	=11
verFrenchUniversal	=91
verFrSwiss	=18
verGeorgian	=85
verGermany	=3
verGreece	=20
verGreecePoly	=40
verGreenland	=107
verGrSwiss	=19
verGujarati	=94
verHungary	=43
verIceland	=21
verIndiaHindi	=33
verIndiaUrdu	=96
verInternational	=37
verIran	=48
verIreland	=50
verIrishGaelicScript	=81
verIsrael	=13
verItalianSwiss	=36
verItaly	=4
verJapan	=14
verKorea	=51
verLatvia	=45
verLithuania	=41
verMacedonian	=67
verMagyar	=59
verMalta	=22
verManxGaelic	=76
verMarathi	=104
verMultilingual	=74

verNepal	=106
verNetherlands	=5
verNorway	=12
verNunavut	=78
verNynorsk	=101
verPakistanUrdu	=34
verPoland	=42
verPortugal	=10
verPunjabi	=95
verRomania	=39
verRussia	=49
verSami	=46
verScottishGaelic	=75
verScriptGeneric	=55
verSerbian	=65
verSingapore	=100
verSlovak	=57
verSlovenian	=66
verSpain	=8
verSpLatinAmerica	=86
verSweden	=7
verTaiwan	=53
verThailand	=54
verTibetan	=105
verTonga	=88
verTurkey	=24
verTurkishModified	=35
verUkraine	=62
verUS	=0
verUzbek	=99
verVietnam	=97
verWelsh	=79

langIrishGaelic

Irish Gaelic for Ireland (without dot above).

Available in Mac OS X v10.0 and later.

Declared in Script.h.

verIreland

Irish Gaelic for Ireland (without dot above).

Available in Mac OS X v10.0 and later.

Declared in Script.h.

langIrishGaelicScript

Irish Gaelic for Ireland (using dot above).

Available in Mac OS X v10.0 and later.

Declared in Script.h.

verIrishGaelicScript

Irish Gaelic for Ireland (using dot above).

Available in Mac OS X v10.0 and later.

Declared in Script.h.

`langSimpChinese`

Chinese using simplified characters.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`smSimpChinese`

Chinese using simplified characters.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verChina`

Chinese using simplified characters.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`langTradChinese`

Chinese using traditional characters.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`smTradChinese`

Chinese using traditional characters.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verTaiwan`

Chinese using traditional characters.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`smCentralEuroRoman`

Script for Czech, Slovak, Polish, Hungarian, and the Baltic languages.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`smRSymbol`

Right-left symbol for bidirectional scripts (such as Arabic and Hebrew).

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verFarEastGeneric`

Generic Far East system (no language or script).

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verGreece`

Monotonic modern Greek.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verGreecePoly`

Polytonic ancient Greek.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verInternational`

English for international use.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verMultilingual`

No language or script.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verScriptGeneric`

Generic script system (no language or script).

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verSpain`

Spanish for Spain.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

`verSpLatinAmerica`

Spanish for Latin America.

Available in Mac OS X v10.0 and later.

Declared in `Script.h`.

See Also

For more information about localization codes, see *Inside Macintosh: Text*. For general information about localization, see *Guide to Macintosh Software Localization* (Addison-Wesley 1992, ISBN 0-201-60856-1).

Document Revision History

This table describes the changes to *QuickTime Constants Reference*.

Date	Notes
2006-05-23	New document, based on previously published material, that covers constants common to multiple QuickTime frameworks.

REVISION HISTORY

Document Revision History

Index

A

addMax **constant** 75
addOver **constant** 75
addPin **constant** 75
adMax **constant** 75
adMin **constant** 75
Arithmetic and Logical Operator IDs 50
Atom ID Codes 5

B

badCallOrderErr **constant** 70
badDepthErr **constant** 70
blend **constant** 75

C

cantCreateSingleForkFile **constant** 69
Codec Flags 15
Codec Identifiers 50
Codec Properties 51
Codec Type Constants 52
codecCompletionDecoded **constant** 18
codecCompletionDest **constant** 17
codecCompletionDropped **constant** 18
codecCompletionForceChainFlush **constant** 17
codecCompletionNotDisplayable **constant** 18
codecCompletionNotDrawn **constant** 18
codecCompletionSource **constant** 17
codecCompletionUnlockBits **constant** 17
codecCompletionWentOffscreen **constant** 17
codecFlagDiffFrame **constant** 15
codecFlagLiveGrab **constant** 15
codecFlagSupportDisable **constant** 15
codecFlagUpdatePrevious **constant** 14
codecFlagUpdatePreviousComp **constant** 15
codecFlagUsedImageBuffer **constant** 18

codecFlagWasCompressed **constant** 14
codecImageBufferIsInPCIMemory **constant** 52
CodecInfo Values 21
codecInfoDepth1 **constant** 22
codecInfoDepth16 **constant** 23
codecInfoDepth2 **constant** 22
codecInfoDepth24 **constant** 23
codecInfoDepth32 **constant** 23
codecInfoDepth33 **constant** 23
codecInfoDepth34 **constant** 23
codecInfoDepth36 **constant** 23
codecInfoDepth4 **constant** 22
codecInfoDepth40 **constant** 23
codecInfoDepth8 **constant** 22
codecInfoDoes1 **constant** 18
codecInfoDoes16 **constant** 19
codecInfoDoes2 **constant** 18
codecInfoDoes32 **constant** 19
codecInfoDoes4 **constant** 18
codecInfoDoes8 **constant** 18
codecInfoDoesBlend **constant** 20
codecInfoDoesDither **constant** 19
codecInfoDoesDouble **constant** 19
codecInfoDoesHalf **constant** 19
codecInfoDoesHorizFlip **constant** 20
codecInfoDoesLossless **constant** 23
codecInfoDoesMask **constant** 19
codecInfoDoesMultiPass **constant** 20
codecInfoDoesQuad **constant** 19
codecInfoDoesQuarter **constant** 20
codecInfoDoesRecompress **constant** 20
codecInfoDoesReorder **constant** 20
codecInfoDoesRotate **constant** 20
codecInfoDoesShrink **constant** 19
codecInfoDoesSpool **constant** 21
codecInfoDoesStretch **constant** 19
codecInfoDoesTemporal **constant** 19
codecInfoDoesVertFlip **constant** 20
codecInfoDoesWarp **constant** 20
codecInfoHasEffectParameterList **constant** 20
codecInfoResourceType **constant** 52
codecInfoStoresClut **constant** 23

[codecInterfaceVersion constant 52](#)
[codecLosslessQuality constant 21](#)
[codecLowQuality constant 21](#)
[codecMaxQuality constant 21](#)
[codecMinQuality constant 21](#)
[codecNormalQuality constant 21](#)
[codecSuggestedBufferSentinel constant 52](#)
[codecSupportsOutOfOrderDisplayTimes constant 52](#)
[codecSupportsScheduledBackwardsPlaybackWithDifferenceFrames constant 52](#)
Color Constants 54
Color Modes 54
Component Call Selectors 54
Component Identifiers 63
Component Property IDs and Flags 64
[componentD11EntryNotFoundErr constant 69](#)
[componentD11LoadErr constant 69](#)
CreateMovieFile Values 23
[createMovieFileDontOpenFile constant 24](#)

D

[dfClipToTextBox constant 49](#)
[dfDontAutoScale constant 49](#)
[dfDontDisplay constant 49](#)
[dfHorizScroll constant 49](#)
[dfReverseScroll constant 49](#)
[dfScrollIn constant 49](#)
[dfScrollOut constant 49](#)
[dfShrinkTextBoxToFit constant 49](#)
[dfUseMovieBGColor constant 49](#)
[digiUnimpErr constant 69](#)
[ditherCopy constant 76](#)

E

Error Codes 64

F

FCompressImage Values 14
File Types and Creators 70
FlattenMovieData Values 24

G

Graphics Transfer Modes 72
[grayishTextOr constant 74](#)

H

[hilite constant 74](#)
[hilitetransfermode constant 74](#)

I

ICM Preferences and Flags 24
[icmFrameTimeHasDecodeTime constant 25](#)
[icmFrameTimeHasVirtualStartTimeAndDuration constant 25](#)
ImageFieldSequenceExtractCombine Values 26

K

[k422YpCbCr10CodecType constant 53](#)
[k422YpCbCr16CodecType constant 53](#)
[k422YpCbCr8CodecType constant 53](#)
[k4444YpCbCrA8CodecType constant 53](#)
[k4444YpCbCrA8RCodecType constant 54](#)
[k444YpCbCr10CodecType constant 53](#)
[k444YpCbCr8CodecType constant 53](#)
[kComponentPropertyCacheFlagIsDynamic constant 27](#)
[kComponentPropertyCacheFlagNotPersistent constant 27](#)
[kComponentPropertyCacheFlags constant 27](#)
[kComponentPropertyCacheSeed constant 27](#)
[kComponentPropertyClassPropertyInfo constant 27](#)
[kComponentPropertyInfoList constant 27](#)
kDataHCanRead 27
[kDataHCanRead constant 28](#)
[kDataHCanStreamingWrite constant 28](#)
[kDataHCanWrite constant 28](#)
[kDataHSpecialRead constant 28](#)
[kDataHSpecialReadFile constant 28](#)
[kDataHSpecialWrite constant 28](#)
[kICMPixelFormatIsIndexed constant 26](#)
[kICMPixelFormatIsPlanarMask constant 26](#)
[kICMPixelFormatIsSupportedByQD constant 26](#)
[kICMSequenceUserPreferredCodecs constant 26](#)
[kMovieMediaBackgroundColor constant 12](#)

kMovieMediaDefaultDataReferenceID **constant 12**
 kMovieMediaPrerollTime **constant 13**
 kMovieMediaSlaveGraphicsMode **constant 12**
 kMovieMediaSlaveTime **constant 12**
 kQTAlphaMode **constant 54**
 kQTAlphaModePreMulColor **constant 54**
 kQTEnableExif **constant 13**
 kQTParseTextHREFText **constant 13**
 kSoftSynthComponentSubType **constant 50**
 kSorenson3CodecType **constant 53**
 kTargetChildMovieMovieName **constant 13**
 kTargetChildMovieTrackIndex **constant 13**
 kTargetCurrentQTEventParams **constant 14**
 kTargetQD3DNamedObjectName **constant 13**
 kTargetSpriteID **constant 13**
 kTargetSpriteName **constant 13**
 kTargetTrackIndex **constant 13**
 kTargetTrackType **constant 13**
 kTextEditState **constant 48**
 kTextHyperTextColor **constant 48**
 kTextHyperTextFace **constant 48**
 kTextKeyEntry **constant 48**
 kTextLength **constant 48**
 kTextMouseDown **constant 48**
 kTextRelativeScroll **constant 48**
 kTextScroll **constant 48**
 kTextSelection **constant 48**
 kTextTextBox **constant 48**

L

langIrishGaelic **constant 81**
 langIrishGaelicScript **constant 81**
 langSimpChinese **constant 82**
 langTradChinese **constant 82**
 Localization Codes **76**

M

matrixErr **constant 69**
Media Identifiers 44
 mediaSampleNotSync **constant 31**
 movieFileSpecValid **31**
MovieImportDataRef Values 31
 movieImportResultNeedIdles **constant 32**
 movieProgressClose **constant 33**
 movieProgressOpen **constant 32**
MovieProgressProc Values 32
 movieProgressUpdatePercent **constant 33**

N

New Movie Properties 34
NewMovieController Values 34
 noDMAErr **constant 70**
 noMoreKeyColorsErr **constant 70**
 noRecordOfApp **constant 69**
 notExactMatrixErr **constant 69**
 notExactSizeErr **constant 70**
 notPatCopy **constant 74**
 notPatOr **constant 74**
 notPatXor **constant 74**
 notSrcBic **constant 73**
 notSrcCopy **constant 73**
 notSrcOr **constant 73**
 notSrcXor **constant 73**

P

patBic **constant 74**
 patCopy **constant 73**
 patOr **constant 74**
 patXor **constant 74**
 pdActionCompactSample **constant 38**
 pdActionConvertSettingsToText **constant 40**
 pdActionConvertSettingsToXML **constant 40**
 pdActionConvertSettingsToXMLWithComments
 constant 40
 pdActionConvertXMLToSettings **constant 40**
 pdActionCustomDisposeControl **constant 38**
 pdActionCustomDoEditCommand **constant 39**
 pdActionCustomGetEnableValue **constant 39**
 pdActionCustomGetValue **constant 39**
 pdActionCustomHandleEvent **constant 39**
 pdActionCustomNewControl **constant 38**
 pdActionCustomPositionControl **constant 38**
 pdActionCustomSetEditCallout **constant 39**
 pdActionCustomSetEditMenu **constant 39**
 pdActionCustomSetFocus **constant 39**
 pdActionCustomSetPreviewPicture **constant 39**
 pdActionCustomSetSampleTime **constant 39**
 pdActionCustomShowHideControl **constant 38**
 pdActionDoEditCommand **constant 38**
 pdActionFetchPreview **constant 37**
 pdActionGetDialogSettings **constant 37**
 pdActionGetNextSample **constant 37**
 pdActionGetPreviousSample **constant 38**
 pdActionGetSubPanelMenuValue **constant 38**
 pdActionModelessCallback **constant 37**
 pdActionRunInEventLoop **constant 39**
 pdActionSetDialogSettings **constant 37**
 pdActionSetEditCallout **constant 38**

pdActionSetPropertyComponent **constant** 40
 pdActionSetSampleTime **constant** 38
 pdOptionsEditCurrentEffectOnly **constant** 40
 pdOptionsHidePreview **constant** 40
 pdOptionsModalDialogBox **constant** 40
 progressOpAddMovieSelection **constant** 33
 progressOpCopy **constant** 33
 progressOpCut **constant** 34
 progressOpFlatten **constant** 33
 progressOpImportMovie **constant** 34
 progressOpInsertMovieSegment **constant** 33
 progressOpInsertTrackSegment **constant** 33
 progressOpLoadMediaIntoRam **constant** 34
 progressOpLoadMovieIntoRam **constant** 34
 progressOpLoadTrackIntoRam **constant** 34
 progressOpPaste **constant** 33

Q

qtMdl1EntryNotFoundErr **constant** 69
 qtMdl1LoadErr **constant** 69
 qtParamErr **constant** 69
 QTSampleTableGetSampleFlags Values 30
 QTSetComponentProperty Values 26
 QTVRWrapAndConstrain Values 28
 QuickTime Preferences Dialog Options 35

S

scAvailableCompressionListType **constant** 43
 scCodecManufacturerType **constant** 43
 scCodecSettingsType **constant** 43
 scDataRateSettingsType **constant** 43
 scForceKeyValueType **constant** 43
 scSoundChannelCountType **constant** 44
 scSoundCompressionType **constant** 44
 scSoundInputSampleRateType **constant** 44
 scSoundSampleRateChangeOK **constant** 43
 scSoundSampleRateType **constant** 44
 scSoundSampleSizeType **constant** 44
 scSoundVBRCompressionOK **constant** 43
 scSpatialSettingsType **constant** 43
 scTemporalSettingsType **constant** 43
 scVideoAllowFrameReorderingType **constant** 42
 scWindowOptionsType **constant** 43
 SGPanelGetDITLForSize Values 44
 smCentralEuroRoman **constant** 82
 smRSymbol **constant** 82
 smSimpChinese **constant** 82
 smTradChinese **constant** 82

SoundMediaType **constant** 45
 Sprite Properties 29
 SpriteWorldHitTest Values 45
 srcBic **constant** 73
 srcCopy **constant** 72
 srcOr **constant** 73
 srcXor **constant** 73
 Standard Compression Constants 40
 subOver **constant** 75
 subPin **constant** 75

T

Text Properties 46
 TextMediaType **constant** 45
 ToneDescription Values 50
 transparent **constant** 76

V

VDSetCompression Values 21
 verChina **constant** 82
 verFarEastGeneric **constant** 82
 verGreece **constant** 82
 verGreecePoly **constant** 83
 verInternational **constant** 83
 verIreland **constant** 81
 verIrishGaelicScript **constant** 81
 verMultilingual **constant** 83
 verScriptGeneric **constant** 83
 verSpain **constant** 83
 verSpLatinAmerica **constant** 83
 verTaiwan **constant** 82